A Cross-layer Middleware Architecture for Time and Safety Critical Applications in MANETs

by

Sarogini Grace Pease

A Doctoral Thesis

Submitted in partial fulfilment
of the requirements for the award of

Doctor of Philosophy
of
Loughborough University

21st August 2013

Abstract

Mobile Ad hoc Networks (MANETs) can be deployed instantaneously and adaptively, making them highly suitable to military, medical and disaster-response scenarios. Using real-time applications for provision of instantaneous and dependable communications, media streaming, and device control in these scenarios is a growing research field. Realising timing requirements in packet delivery is essential to safety-critical real-time applications that are both delay- and loss-sensitive. Safety of these applications is compromised by packet loss, both on the network and by the applications themselves that will drop packets exceeding delay bounds. However, the provision of this required Quality of Service (QoS) must overcome issues relating to the lack of reliable existing infrastructure, conservation of safety-certified functionality. It must also overcome issues relating to the layer-2 dynamics with causal factors including hidden transmitters and fading channels. This thesis proposes that bounded maximum delay and safety-critical application support can be achieved by using cross-layer middleware. Such an approach benefits from the use of established protocols without requiring modifications to safety-certified ones. This research proposes ROAM: a novel, adaptive and scalable cross-layer Real-time Optimising Ad hoc Middleware framework for the provision and maintenance of performance guarantees in self-configuring MANETs. The ROAM framework is designed to be scalable to new optimisers and MANET protocols and requires no modifications of protocol functionality. Four original contributions are proposed: (1) ROAM, a middleware entity abstracts information from the protocol stack using application programming interfaces (APIs) and that implements optimisers to monitor and autonomously tune conditions at protocol layers in response to dynamic network conditions. The cross-layer approach is MANET protocol generic, using minimal imposition on the protocol stack, without protocol modification requirements. (2) A horizontal handoff optimiser that responds to time-varying link quality to ensure optimal and most robust channel usage. (3) A distributed contention reduction optimiser that reduces channel contention and related delay, in response to detection of the presence of a hidden transmitter. (4) A feasibility evaluation of the ROAM architecture to bound maximum delay and jitter in a comprehensive range of ns2-MIRACLE simulation scenarios that demonstrate independence from the key causes of network dynamics: application setting and MANET configuration; including mobility or topology. Experimental results show that ROAM can constrain end-to-end delay, jitter and packet loss, to support real-time applications with critical timing requirements.

Acknowledgements

This work is sponsored by CASE award of the Engineering and Physical Science Research Council (EPSRC) and BAE Systems, UK. I would like to thank:

- My supervisors: Dr Iain Phillips, Dr Lin Guan, Dr Alan Grigg for their invaluable and inspirational guidance and support during the course of my PhD.

- Dr Olaf Maennel for his expert advice and feedback during annual review meetings.

- The staff in the Department of Computer Science at Loughborough University and at the Systems Engineering and Innovation Centre, in particular Dr George Oikonomou, Judith Poulton, Christine Bagley, Jo Mcouat for their kind support.

- The students in my department and fellow research student, Xin, for their friendship and advice.

- Finally, my parents for always keeping me motivated and my husband, David for his unwavering support and positivity.

Publications

- Sarogini Grace Pease. ROAM: Supporting Safety Critical Applications in MANETs with Cross-layer Middleware, *In Proceedings of IEEE 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Madrid, Spain, Jun. 2013*

- Sarogini Grace Pease, Iain Phillips, Lin Guan, Alan Grigg. Cross-Layer Signalling and Middleware: A Survey for Inelastic Soft Real-time Applications in MANETs. *Elsevier Journal of Network and Computer Applications, pp. 1928-1941, Vol. 34, Issue 6, 2011*

Contents

List of Figures

List of Tables

Glossary of Acronyms

ABR Associativity Based Routing

ACK Acknowledgement

AFDX Avionics Full DupleX switched ethernet

AIMD Additive Increase Multiplicative Decrease

AMRIS Ad hoc Multicast Routing protocol utilising increasing ID numberS

AMRoute Ad hoc Multicast Routing

AODV Ad hoc on-demand distance vector routing

AP Access Point (See PoA)

API Application Programming Interface

AQM Active Queue Management

AQOR Ad hoc QoS Aware On-demand Routing

ARP Address Resolution Protocol

ARQ Automatic Repeat reQuest

ASAAC Allied Standard Avionics Architecture Council

ATM Asynchronous Transfer Mode

BAG Bandwidth Allocation Gap

BCET Best-case Execution Time

BDP Bandwidth Delay Product

bps bits per second

CACP Contention-aware Admission Control Protocol

CBR Constant Bit Rate

CDF Cumulative Distribution Function

CGSR Cluster Gateway Switch Routing

COIN Common Optimisation Interface

COOL Common Optimisation Layer

COP Common Optimisation Protocol

COTS Commercial Off The Shelf

CQI Channel Quality Information

CQMP Mesh-based multicast routing protocol with Consolidated Query Packets

CRMAC Congestion Reducing Medium Access Control

CSMA Carrier Sense Multiple Access

CTS Clear To Send
CWND Congestion window
D-TUBA Differentiated-Time Urgency Based Algorithm
DBTMA Dual Busy Tone Multiple Access
DCSPT Dual Carrier Sensing with Parallel Transmission awareness
DCF Distributed Coordination Function
Delay-EDD Delay-Earliest Due Date
DFS Distributed Fair Scheduling
DHCP Dynamic Host Configuration Protocol
DiffServ Differentiated Services
DII Defence Information Infrastructure
DRR Deficit Round Robin
DS Data Sending
DSDV Destination-Sequenced Distance Vector
DSR Dynamic Source Routing
DVMRP Distance Vector Multicast Routing Protocol
DWRR Deficit Weighted Round Robin
ECN Explicit Congestion Notification
EuQoS End-to-end QoS over Heterogeneous Networks
EWMA Exponentially Weighted Moving Average
FEC Forward Error Correction
FIFO First In First Out
FPV Flight Path Vector
FQMM Flexible QoS Model for MANET
GOP Group Of Pictures
HARQ Hybrid ARQ protocol
HF High Frequency
HN Handoff Notification
HOL Head Of Line
HR Handoff Report
HRR Hierarchical Round Robin
HRT Hard RT
HUD Head Up Display
ICMAN Intermittently Connected Mobile Ad hoc Network
IEEE Institute of Electrical and Electronic Engineering
IETF Internet Engineering Task Force
IFQ InterFace Queue
IGP Interior Gateway Protocol
IMS Integrated Modular Systems

IN Intermediate Node
IN Intermediate Node (Forwarding Node)
IntServ Integrated Services
IP Internet Protocol
IPCP Internet Protocol Control Protocol
ISP Interlayer Signalling Pipe
ISRT Inelastic SRT
J-EDD Jitter-Earliest Due Date
JSCC Joint Source-Channel Coding
JSCPC Joint Source Coding-Power Control
kbps kilo bits per second
LAN Local Area Network
LAN Local Area Network
LATP Link Adaptive Transport Protocol
LGF Location-Based Geocasting and Forwarding
LLC Logical Link Control
LRU Line Replaceable Unit
LUNAR Lightweight Underlay Network Ad hoc Routing
MAC Medium Access Control
MACA Multiple Access with Collision Avoidance
MACAW Multiple Access with Collision Avoidance for Wireless
MACMAN Multi-path Admission Control for Mobile Ad hoc Networks
MAI Multiple Access Interference
MANET Mobile Ad hoc NETwork
MAODV Multicast Ad hoc On-Demand Distance Vector
Mbps Mega bits per second
MIP Mobile IP
MLT Minimum Laxity Threshold
MN Mobile Node
MT Mobile Transmitter
MTU Maximum Transmission Unit
NRT Non-RT
OS Operating System
OSPF Open Shortest Path First
OSS Optimising SubSystem
PAC Perceptive Admission Control protocol
PDR Packet Delivery Ratio
PLR Packet Loss Ratio
PO Protocol Optimiser

PoA Point of Attachment

POEM Performance-OriEnted Model

PPP Point-to-Point Protocol

PPS Packets Per Second

PPRCH Packet Probing with RTS/CTS Handshake

PR$_{rx}$ Receive Power (of packet)

PS Packet Size

PSINR Perceived SINR

PSNR Perceived SNR

PSP Prediction Service Proxy

QLT Queue Length Threshold

QMPR QoS-aware Multicast Routing Protocol

QoS Quality of Service

R Transmission Rate

RCVWND Receive Window

RD Rate Distortion

RED Random Early Detection

RescACK Reservation ACK

ResvCTS Reservation CTS

ResvRTS Reservation RTS

RF Radio Frequency

RM Redundancy Management

RMU RM Unit

ROMANT Robust Multicasting in Ad hoc Network using Tree

RR Receiver Report

RREP Route REPly

RREQ Route REQuest

RSS Received Signal Strength

RSSI RSS Indicator

RSVP Resource reSerVation Protocol

RT Real-Time

RTCP RT Control Protocol

RTMAC RT MAC

RTO Recovery Time Objective

RTP RT Protocol

RTS Request To Send

RTT Round Trip Time

SD Standard Deviation

SCFQ Self-Clocked Fair Queueing

SGQ Stop and Go Queueing
SINR Signal and Interference to Noise Ratio
SNR Signal to Noise Ratio
SPBM Scaleable Position-Based Multicast
SRT Soft RT
SSR Signal Stability Routing
TBP Ticket Based Probing
TCP Transmission Control Protocol
TTL Time To Live
TORA Temporally Ordered Routing Algorithm
UAV Unmanned Air Vehicle
VBR Variable Bit Rate
VL Virtual Link
WAN Wide Area Network
WCET Worst Case Execution Time
WCI Wireless Channel Information
WDRR Weighted Deficit Round Robin
WEH Wireless Extension Header
WFQ Weighted Fair Queueing
WiMAX Worldwide Interoperability for Microwave Access
WLAN Wireless LAN
WRP Wireless Routing Protocol
WRR Weighted Round Robin

Chapter 1

Introduction

Mobile Ad hoc Networks (MANETs) are self-organising, infrastructureless networks. MANET protocols work on a self-configuring basis to adaptively create network connections, without centralised management. This makes them ideal for media streaming and communications in military or disaster-response scenarios, which are limited by insufficient telephony and cellular infrastructure. For example, in a military scenario, a UAV out of range of a base station could still transmit mission critical video data on friendly vehicles in its vicinity to the base station via a network of other mobile vehicles.

Each node in a MANET discretely acts as mobile transmitter, receiver or router along an end-to-end (E2E) path. In the latter case the node is referred to as an intermediate node (IN), responsible for forwarding packets to a receiver. Mobility in some or all MANET nodes leads to dynamic and frequent setup and teardown of connections and paths through the network. Link quality and availability changes as nodes move farther from each other or into interference range [143, 172]. Wireless channels are also subject to environmental interference, hidden nodes, multipath fading or attenuation and Doppler effects [54, 67, 87, 88, 118, 129, 138]. These create multiple component factors at layers 1 and 2 that contribute to high loss and variable E2E delay.

Time critical network applications are delay- and jitter-sensitive, in contrast to non-real-time (NRT) applications that are not time or constraint driven. If these applications are also safety-critical, timeliness of packet delivery can influence both the usefulness of data and safety of a system. Support for these applications is a growing research field that extends to military scenarios, search and rescue, disaster response, media streaming, online communications and gaming. All these types of applications can be divided into three groups: hard real-time (HRT), elastic soft real-time (SRT) and inelastic soft real-time (ISRT) [70]. In wired networks, over-provisioning and resource management or predetermined routing can be used to provide the HRT stipulations of fixed delivery deadlines and zero loss guarantees [72].

In dynamic wireless networks absolute guarantees cannot be provided. Therefore, widely used elastic SRT applications for Voice Over Internet Protocol (VoIP) and multimedia streaming have been designed to tolerate loss and delay. However, safety critical applications are rigorously tested to certify reliability. For example when transmitting video feeds of friendly and non-friendly vehicles in a military operational scenario, regular loss and delay of video frames would not be acceptable. These applications therefore operate within the remit of ISRT, that instead tolerates loss, delay and jitter within acceptable and guaranteed bounds. This thesis, therefore, considers the support of ISRT applications. Safety of these applications is compromised by excessive packet loss, both on the network and by the applications themselves, that will drop packets exceeding delay bounds. Quality of Service (QoS) provisioning that ensures timeliness is provided cannot therefore rely on a tradeoff for unbounded E2E loss. Therefore, for MANETs to be viable solutions to ISRT applications, service provisioning must be responsive to the layer-2 conditions that contribute to high loss and variable E2E delay in these networks.

It has been widely concluded that cross-layer responsiveness to dynamic network conditions enables higher layer protocols to distinguish between causes of packet losses and errors [21, 52, 83, 121, 144, 144, 147, 147, 163]. The static services provided by oblivious OSI layers create increased loss and delay in ad hoc networks, due to reliance on centralised control and a lack of coordination between the efforts of coexisting protocol layers [28, 118, 160, 163, 175]. Schemes that follow this paradigm of information or signal sharing sit under the title of cross-layer optimisation or design. Many cross-layer approaches to QoS improvement have been proposed for NRT applications in MANETs and for SRT applications in wireless networks, with a few straddling both of these fields. Most have been developed to meet highly specialised network performance goals such as improvement in video quality [4, 37, 65, 67, 77, 169, 184, 185] or TCP fairness [16, 53, 126, 151, 168, 173]. To implement these approaches, various cross-layer architectures have been proposed along with mixed combinations of protocols to be optimised [40, 65, 82, 83, 115, 147, 163, 168]. Many of these designs have therefore introduced levels of complexity in protocol modifications and interactions that reduce their scalability and reusability [76].

Withholding internal layer parameters from other layers has long been used to facilitate the fast development of interoperable systems. Managing cross-layer optimisation with independent middleware can be used to preserve this successful functionality by monitoring protocol parameters and responsively tuning these from a single, external location [30, 58, 82, 124, 165]. Protocol-independence enables generic support of contemporary and safety-certified network protocols by modifying parameters that a protocol accesses rather than the protocol functionality itself. The corollary of this is that future MANET protocols can continue to be interoperable and transparent as they need not be developed with complex interlayer interactions.

Current proposals for non-real-time applications have conceptualised but not tested middleware that uses API access to protocol data structures [30, 58, 124]. These designs can provide generic solutions for contemporary and legacy network protocols, but require the addition of optimising functionality to meet the requirements of maximum delay and jitter sensitive ISRT. A wider variety of QoS provisioning methods that do not utilise local middleware have been proposed for SRT support [4, 24, 38, 65, 71, 78, 93, 98, 108, 108, 140, 159, 178] and for MANET performance improvement [34, 40, 40, 59, 63, 88, 104, 118, 121, 134, 138].

There is still a need to constrain delay and jitter, with due weight given to the safety critical, loss-sensitive nature of scenarios where inelastic SRT applications are used in MANETs. In light of these considerations a dynamic middleware approach is required to improve ISRT performance. Given the dynamic MANET conditions it must function within, this research aims to develop a generic approach that functions independent of these conditions, which are primarily caused by application transmission settings and MANET configurations

1.1 Research Motivation

The aim of this research project is to develop a cross-layer approach that compensates for ad hoc changes to resource availability in MANETs in order to provide performance guarantees to inelastic SRT applications that require bounded E2E delay, jitter and packet loss. In particular, it considers the performance deterioration under mobility induced handoff and hidden node contention.

The key motivation of this thesis is to provide mobile and dependable media communications in military or disaster scenarios that are subject to certain wireless network issues: lack of reliable existing infrastructure, requirements to conserve safety-certified functionality and also layer-2 dynamics with major influences including hidden transmitters and fading links. Few research contributions in the field of real-time networking and MANETs overlap both of these fields simultaneously, or consider the stringent timing requirements of the safety-critical ISRT domain. MANETs are ideal for media streaming and communications in military or disaster scenarios [21]. However, the use of shared channels and time-varying or complex network topologies create multiple factors at layers-1 and -2 that contribute to high and variable E2E delay. Previous research in these fields has therefore concluded the importance of cross-layer design in providing wireless network performance guarantees [144, 163]. Cross-layer design can also eliminate the need for protocol modification. This can then provide continuing support to safety-certified technologies alongside contemporary approaches. Additionally, this avoids modification of military or commercial hardware or MAC layer firmware that is often not possible or creates issues of reduced interoperability and transparency. There is still a requirement to ensure that E2E delay and jitter can be bounded for MANETs to be viable solutions to real-time applications in these scenarios, where timeliness of packet delivery influences both usefulness and safety.

The objectives of this research project are therefore to:

- Evaluate the performance degradation experienced by ISRT transmissions in a MANET that result from contention between hidden transmitters and around horizontal handoff as a transmitter moves through the network.
- Develop a lightweight middleware solution and method of cross-layer information exchange that
  - Requires minimal imposition on the protocol stack, accessing protocol parameters only using generic APIs
  - Utilises information from, but does not optimise, MAC or physical layers that are generally inaccessible to developers
- Ensure that this middleware solution manages network optimisation with a methodology that is:
  - Autonomously responsive to network dynamics: primarily caused by application, transmission and MANET configurations
  - Stateless and independent of application, transmission and MANET configurations
  - Improves performance by providing bounded E2E delay, jitter and loss when shared link contention and horizontal handoff requirement appear

1.2 Major Contributions

This thesis proposes, tests and validates an architecture containing ROAM a novel, adaptive and scalable Real-time Optimising Ad hoc Middleware and cross-layer framework for the provision and maintenance of performance guarantees in MANETs. ROAM is designed to support ISRT applications used in military, medical and emergency scenarios that have safety-critical as well as timing guarantee requirements.

Through compensation for changes in resource availability, the ROAM framework provides bounded E2E delay, jitter and reduced packet loss. ROAM is also designed to support heterogeneity of contemporary and immutable safety-certified protocols. Further detail is given in Chapter 4, with an overview of the design in figure 4.1. ROAM can support multiple optimising functions and two optimisers have been developed for the management of optimised handoff between channels of the same technology and load control under hidden node contention conditions.

The first optimising function avoids suboptimal link use when resource conditions are reduced, through accessing lower layer information and optimising the network and application layers. This optimiser also prevents the use of links that are not robust, due to high speed or highly variable node mobility. The second is a distributed contention reduction optimiser that responds to detection of the presence of a hidden transmitter. The optimiser ensures that resource use is reduced in conditions of increased link contention; following detection by ROAM of the presence of hidden transmitters.

ROAM has been designed to seamlessly support heterogeneous systems without imposing novel modifications on protocols or complex stack or interlayer interactions. ROAM uses generic API to abstract performance information held in protocol layer parameters. ROAM then uses API access above layer-2 to tune parameter data structures in an adaptive and scalable manner, providing responsiveness to dynamic network conditions. By accessing but not exploiting lower layer parameter data structures, ROAM maintains transparency and interoperability by avoiding firmware or hardware modification.

The feasibility of the ROAM architecture is validated in Chapters 5 and 6 in simulation scenarios that demonstrate independence from the causes of network dynamics: application type, transmission setting and MANET conditions, such as mobility or topology. ROAM provides better performance, in the form of bounded maximum delay and jitter and reduced packet loss guarantees, than can be provided by the unoptimised MANET protocol stack.

1.3 Research Goals

The goals of a new network framework or adaptation of an existing model should be independent of the specific implementation but related to its future potential. The characteristics of a framework, how it is managed and the specific control measures implemented will have an impact on the achievement of the following overarching design goals:

- Adaptable and rapid prototyping: through the development of an adaptable and scalable network control concept. Although the OSI model was developed for a wired network it provides strong benefits to developers in providing modularity: the ability to exchange protocols at one layer without considering or impacting on other layers. The wireless ad hoc medium is subject to numerous vulnerabilities not seen in the wired medium that manifest as variations in resource availability, complicating the achievement of QoS performance targets. As a result a dynamic and potentially tunable approach is desirable that can maintain QoS with minimal intrusion on or independence of the protocol stack design. Such an approach should be easily implemented in a small initial prototype but also scalable to more complex deployments in terms of topology, network loading and mobility or traffic characteristics. In this sense the scaling of processing and communication overheads of a desired implementation must be considered, for example cross-layer approaches that rely on E2E messaging can incur high overheads over large or dynamic topologies, negating their benefit in an ad hoc network.

- Transparency and portability: between the hardware and software implementation, ensuring that heterogeneous nodes can interact with the proposed control framework and that the framework can be ported to multiple systems. The design may therefore benefit from a distributed rather than centralised implementation that is recommended in ad hoc environments where failure of a single point of responsibility is highly probable. When a distributed per-hop approach is selected, high node mobility should also have a lower impact on QoS control.

- Lightweight design and efficiency: of collaboration between control measures and therefore protocols. The proposed implementation should minimise messaging and processing requirements in light of the high-speed requirements of RT implementations; the reduced resource availability in the shared medium as well as the low memory and processing capacity within nodes themselves.

1.4 Thesis Structure

The rest of the thesis is organised as follows:

A full survey of current approaches to cross-layer design and schemes for the maintenance of QoS in MANETs and for wireless real-time support is given in Chapter 2. Projects investigating real-time communications in military networks are studied and a case-study taken into a current network solution for wired military traffic. This comprehensive literature review outlines the open research areas in the field of delay and loss-sensitive applications in MANETs. This provides the grounding and motivation for this thesis. A delay, jitter and loss analysis in a variety of MANETs is presented in Chapter 3. This investigates the key factors that cause peak delay and packet loss under a range of network configurations. In particular the effects of repeated handoff and the presence of multiple transmitters are analysed. These results are also used to compare the output of the ns-2 simulator and the newer ns2-MIRACLE addon. The analysis demonstrates the requirement for cross-layer responsiveness to channel conditions in order to provide an adaptive method of bounding peak delay and loss in a MANET. The simulation methodology, including outlines of the experimental setup: scenarios, topologies, envrionments and configurations is also provided.

Chapter 4 presents a cross-layer middleware architecture that implements two optimisers that tune parameters at the network and application layers in response to node mobility and channel contention. This is to constrain E2E delay, jitter and packet loss. The proposed architecture consists of layer-associated API, cross-layer messages and ROAM, the proposed middleware entity. The horizontal handoff optimiser, implemented by ROAM, is simulated and validated using the ns2-MIRACLE simulator in Chapter 5. Validation and analysis of performance with the contention control optimiser is conducted in Chapter 6. Each optimiser has been rigourously tested under a wide range of traffic and network configurations to demonstrate the feasiblity of implementation in the dynamic conditions that appear in a MANET. Finally, Chapter 7 concludes on the findings of this thesis, and considers the extensibility of the ROAM architecture. It provides a summary of how the ROAM architecture reflects the thesis motivation and aims considered in this chapter and the goals of supporting delay and loss-sensitive applications in MANETs.

Chapter 2

Approaches to MANET Performance Improvement

2.1 Introduction

This chapter provides a comprehensive literature review of QoS control schemes for MANETs and real-time applications and performance improvement approaches that use cross-layer design. Real-time applications [70], depending on flow characteristics, require diverse levels of performance from a network. Applications that control onboard aircraft equipment, such as with Avionics Full DupleX switched ethernet(AFDX) [2], will require assurances that flows arrive at the other end of the network through deterministic delay, jitter and loss guarantees. In contrast, applications like Voice over Internet Protocol (VoIP), may operate within tolerable maximum constraints. Within an ad hoc network that consists of rapidly variable links, all QoS provisioning must occur without reliance on infrastructure support or centralised management [56, 87]. In order to accommodate safety critical applications any QoS control approach must also support heterogeneity in order to benefit from contemporary protocols without exploiting the functionality of safety-certified protocols.

Many protocols have developed to meet stringent real-time QoS requirements on wireless channels or compensate for MANET channel quality variation. Without centralised control these rely on single-hop guarantees that are then extended along the E2E path of a packet from transmission to receipt through the implementation of QoS control measures:

- Traffic shaping: used to control the volume of traffic entering a network in a certain period (bandwidth throttling) or the maximum rate at which traffic can be sent (rate limiting) through the introduction of delay.
- Traffic policing: more robust technique than shaping, where non-conforming

packets exceeding a bandwidth allocation are marked to be dropped or dropped immediately.

- Traffic conditioning: uses service differentiation and admission control to manage entry of traffic flows to a network followed by traffic shaping and traffic policing to prevent over subscription of services.
- Resource reservation: implements bandwidth sharing, traffic shaping, policing or conditioning and packet scheduling to allocate and maintain the allocation of resources to flows (Section 2.3.1).
- Collision avoidance: reduces packet loss rate by ensuring fair sharing of available resources (Section 2.3.3).
- Congestion control: detects and avoids network congestion also using traffic shaping, policing or conditioning (Section 2.3.3).
- Scheduling: manages enqueueing and dequeueing of packets for fair and maximised use of resources and support of deadline achievement (Section 2.3.4).
- Routing and addressing: selects appropriate E2E packet routes to maximise throughput or minimise delay, while supporting load balancing (Section 2.3.2).
- Service differentiation and admission control: classifies flows into prioritised sets to support differentiated treatment according to QoS requirements of a class. Flows can be provided access to resources according to class-specific allocations (Sections 2.3.5 and 2.3.1).

Protocol development has long followed the practice of withholding internal parameters from other layers to facilitate the fast development of interoperable systems. QoS control schemes are therefore combined to form a network QoS model or framework in which each functions discretely. It has been widely concluded that in wireless and ad hoc networks this leads to low levels of protocol performance [53, 56, 95, 115, 160, 175, 176]. For example, the interaction between bursty real-time traffic and variable MANET configurations creates network dynamics that oblivious higher layer protocols do not respond to. They cannot distinguish between possible causes of packet losses and errors, or ensure fair distribution of bandwidth. As a result, wireless networking is being extended to include the communication of signals between layers and cross-layer interaction whereby several control measures may act as a single collaborative operation within the QoS model [83, 144, 147, 163]. Various methods of cross-layer signalling and protocol tuning have thus been proposed. These must be examined in terms of their improvements to QoS control measures and ability to maintain useful levels of interoperability and transparency.

The rest of the chapter is organized as follows: Section 2.2 provides a brief description of the application, network and scenario requirements of the project; Sections 2.3 and 2.5 give an overview of existing and proposed QoS control schemes and approaches to improving these through cross-layer optimisation; and lastly, a summary of this literature review is discussed in Section 2.6.

2.2 Scope of the Project

This section provides a brief background of the literature to be discussed in this chapter. An overview of requirements engendered by time-critical applications and the nature of MANETs is given alongside discussion of their potential for implementation in safety-critical military scenarios.

2.2.1 An Overview of MANETs

Wireless networking architectures currently span a spectrum from networks of fixed location devices or nodes, such as WLANS, to Intermittently Connected Mobile Ad hoc Networks (ICMANs) often referred to as ad hoc networks. In the latter type, as a result of node mobility, no E2E paths may exist from time to time. Mobile Ad hoc Networks (MANETs) are within the class of ICMANs benefiting from not relying on a pre-existing infrastructure. In a MANET a node will connect to the rest of the network as required, when in range of a connected node or access point (AP).

A MANET node can be referred to as a router, end node, intermediate node, access point or mobile node, as a single node can discretely act as transmitter, receiver or intermediate relay at one point in time. In ad hoc networks all QoS guarantees are limited by the high packet loss rates resulting from propagation in a low frequency spectrum across poor diffusion environments and the unpredictable nature of the network topology. Transmissions are therefore subject to neighbour and environmental interference, multipath fading or attenuation and Doppler effects. The high packet loss rates common to MANET communications are firstly a result of the weaknesses of the wireless medium [54, 129]:

- Limited bandwidth
- High E2E delay
- Time varying delay and throughput.

They are also compounded by dynamic node mobility that introduces new problems of object shadowing and in identification of hidden and exposed nodes [17, 19, 67, 88]:

- Link quality variation with time
- Regular signal outages
- Limited power availability in nodes.

All of these factors contribute to performance reductions in the form of increased packet loss, delay and jitter. As a result service disciplines for MANETs must be optimised to maximise the use of resources as they become available while also ensuring the provision of fair access to the applications sharing these resources. When these applications are transmitting to RT deadlines, E2E delay and throughput must be carefully controlled to ensure that no single flow may over consume resources on an already unreliable network that operates in a close to congested state.

2.2.2 Real-time (RT) Applications

Timeliness is key to RT flows, for which QoS depends strongly upon deadline achievement and high packet arrival rates. All application processes and transmitted packets can be categorised as RT or non-real-time (NRT). RT processes are time-triggered, based on an internal system schedule or event-triggered by environmental stimuli, and explicitly use global physical completion time constraints, or deadlines, to manage their resources. While QoS for RT packets is often expressed primarily in terms of deadline achievement and worst-case execution time (WCET), there is no benefit in delivering RT packets early. This in fact may be detrimental to the system by introducing scheduling problems. Additionally, the consumption of buffer resources by the storage of early arriving packets and can be potentially dangerous to systems or personnel in critical control or mission critical scenarios. Specification of a best-case execution time (BCET) is therefore also necessary [101, 166]. NRT processes may also perform computations which satisfy their timing requirements but resource management is not time or constraint driven. The definition of RT is divided into hard real-time (HRT) and soft real-time (SRT) [64] and the latter has further been subdivided to elastic and inelastic SRT [70] (ISRT):

- **HRT processes** have strict E2E delay requirements, and late packets are considered unusable. This is because the completion of a related computation after its deadline will impede a systems ability to operate correctly or have a critical impact on the system. A deterministic deadline, or constant execution time, is therefore required in these safety critical systems to guarantee no damage to equipment or personnel. HRT packet deadlines are fixed and must always be realised for minimum QoS guarantees to be met. For example all directions for the remote operation of a medical device must arrive at the time specified by the deadline. HRT systems are also highly loss intolerant, and thus reliant on underutilisation, static resource management and predetermined fixed routing.

- **Elastic SRT processes** require constraints on E2E delay in light of a computational deadline but can tolerate packet arrival at a suboptimal time to differing degrees. Certain SRT applications, such as multimedia streams, may be able to compensate for delayed completion, translating this to a lower level of user service. Soft deadlines are generally used to ensure an optimally efficient rather than fixed reaction to a trigger. A SRT deadline has an explicit BCET and WCET, between which the usefulness of the output decreases. A WCET can also be missed occasionally, typically with an upper bound on the number of misses within a defined interval. For example VoIP applications have an interval between packet arrivals and buffer these prior to playback to compensate for jitter in the stream.

- **ISRT processes** will have more stringent delay and jitter requirements and low tolerance to packet loss. In comparison to traditional SRT packets these require a low WCET and thus a smaller difference between BCET and WCET, a low upper bound on acceptable WCET misses is also necessary. In this way a high level of deadline achievement is stipulated, without the hard requirement for scheduling to a constant execution time. ISRT also requires guaranteed bounded packet delivery, or maximum packet loss. For example, in transmission of a mission critical video stream from a surveying UAV to a military aircraft, partial data loss would always be preferable to total data loss.

HRT transmissions strongly depend on the provision of predictable and bounded network jitter and delay and low packet loss. Flow jitter can result within a node, due to variable queueing and processing delays at multiple protocol layers, as a result of over-subscription of resources. It can also appear during packet transit across the network, as a result of multi-hop multi-path routing. Resource reservation disciplines, rate control, admission control, routing or priority-based scheduling can be implemented to ensure that network resources are not over-subscribed. In addition, collision avoidance or congestion control can be used to reduce packet losses and transmission delay along the E2E route.

The requirements from a network that will provide HRT support are stringent and traditional wired OSI architectures rely on underutilisation and static management of resources in combination with predetermined fixed routing to provide QoS. Notwithstanding the cost to flexibility and scalability of such implementations, when the medium of transmission consists of a dynamic topology with variable resource availability, the static definition of scheduling and resource management will negatively impact deadline achievement. A requirement still exists to transmit packets that would be treated as HRT in a deterministic wired network, in wireless networks and applications must therefore be modified, treating HRT packets as ISRT.

As evinced by the need for over-provisioning in wired HRT support and the high loss-tolerance required of SRT, the layered network's simple forwarding services do not support good RT performance. The primary role of the ad hoc network must be extended to deal with the possibility that no E2E path may exist at any one time. Although the practice under OSI of withholding internal layer parameters from other layers facilitates the fast development of interoperable systems, conversely this limits performance due to a lack of coordination between the efforts of coexisting protocol layers. Oblivious layers are unable to distinguish between possible causes of packet losses and errors, or to estimate and fairly distribute available E2E bandwidth.

Wireless networks have not yet been able to meet the deterministic QoS requirements of ISRT, to provide zero or negligible packet loss and guaranteed fixed deadline achievement. A great amount of research effort has, however, concentrated on elastic SRT support, particularly in the area of wireless multimedia streaming [3, 29, 77, 112, 123, 140, 153, 184]. However it is still essential to provide support to ISRT packets, though perhaps only by treating these as elastic SRT but with stringent delay and jitter requirements and low tolerance to packet loss such as to ensure a high level of deadline achievement. The support of ISRT applications in MANETs is still an open research issue therefore this section will investigate the predominant approaches HRT support in safety-critical and military scenarios. Section 2.3 the discusses proposals for to elastic SRT support in wireless networks and NRT in MANETs and evaluates the possibility of implementing these for the provision of QoS to ISRT applications.

2.2.2.1 Measuring RT Performance

QoS metrics quantify the fulfilment of application performance requirements and therefore the value of control or optimisation measures that are implemented in the network. The following is a list of control metrics generally used in wireless and ad hoc network performance evaluation, though the majority are also common to wired networking. To RT applications that are delay or jitter-intolerant, these two metrics become the most important in measuring application performance. However, metrics such as packet loss and goodput also provide a measure of the level of support that is provided by the network.

End to end delay: is the time taken for a packet to move from source to destination and is made up of one-hop transmission, processing, propagation, contention and queuing delays. Transmission and contention delays decide the time for the packet to arrive on the link. The former is the time taken to push an entire packet onto the communication link and is therefore dependent on packet length, while the latter depends on the number of sources competing for the same collision domain and the collision avoidance mechanism in use. Queueing and processing delays occurs within nodes, the first is within the processor and is therefore dependent on the total load and the speed of the processor; the second is the waiting time prior to processing. If as a result of high traffic load or processing requirements the packet arrival rate is faster than the node processing speed packets are enqueued until they can be dealt with on a FIFO basis. Propagation delay is the transmission time across a single hop and is therefore highly dependent on the type of medium. In a wireless network this delay component is generally negligible. The nodal delay is therefore composed of all of these types of packet delay for a single hop and E2E delay is calculated as an aggregation of all of the per-hop delays from source to destination. As an example of RT delay guarantees, one-way delay should not exceed 150ms for a VoIP connection [66].

Jitter: is the variation in delay experienced between different packets on the network as a result of queueing, contention, processing or congestion along the packet path. When an application produces a stream of packets each with a RT deadline that must be reconstructed in sequence with a specific temporal separation it is considered jitter sensitive. Buffering can be implemented at the receiver in order to minimize jitter as long as E2E delay can still be bounded.

Throughput and Goodput: Throughput measures the average rate of successful message delivery through a node or over a physical or logical link, expressed in bps or packets per second or per time slot. The aggregate throughput is the total of all rates delivered to all terminals in a network. As a result of wireless processing overheads, collisions, congestion and interference, achievable throughput is less than available bandwidth and the application transmission rate, or datarate. However, the number of useful bps that traverse the source-destination path, excluding protocol overhead and retransmitted packets is the goodput experienced by the application. This value is lower than the throughput and a better quantifier of user-perceived network performance.

Bandwidth: or Bandwidth capacity is a measure of the maximum bitrate or throughput of a logical or physical link expressed in bps. The calculation of available bandwidth is requisite to resource reservation and admission control mechanisms.

Packet Loss Ratio: is a metric that measures the number of packets that did not arrive at the destination as a percentage of the number sent. For a good connection this value should be as small as possible, although certain applications such as multimedia streaming can tolerate a degree of packet loss at the expense of reduced user perceived video quality. Over allocation of bandwidth, signal outages, congestion and interference can all result in packet loss.

SINR: this metric is an engineering concepts utilised in interference and noise limited systems and increasingly being used in the evaluation of channel strength for multiple access wireless systems where interference and noise impact on the received signal [52]. The SINR is the quotient between the average received power, C, and the average received powers of co-channel interference, I, and noise, N, given by Equation 2.1.

$$SINR = \frac{C}{N + I} \tag{2.1}$$

Sequencing: many applications such as SRT multimedia streaming rely on the sequential delivery of packets, therefore some method of packet identification and possibly re-sequencing at the receiver may be monitored.

2.2.3 Safety Critical Military Requirements

The following section considers the implementation requirements of this project in terms of QoS provision and the communication service required in the military domain [39, 114]. This includes a case study of AFDX [2], a wired network model optimised specifically for high-speed HRT support. The UK MOD project, Network Enabled Capability (NEC) [114], aims to improve communications through the creation of collaborative operational systems and architectures. Timeliness of information provision is a primary objective of NEC, notably when this information is intelligence that will support the political process and must be extracted from numerous sources and rapidly disseminated. Essentially the aim is to enable networks of system entities with multi-dimensional motion to cooperate through high-speed interactions relating to communications, information sharing and operational procedure as illustrated in figure 2.1. All of these communication systems as well as many other safety critical systems are developed in line with the Integrated Modular Systems (IMS) software architecture, which is used in both the civil and military domains and specified by the ASAAC Standard [42]. Modularity of both software and hardware is key to the IMS concept, used in real-time

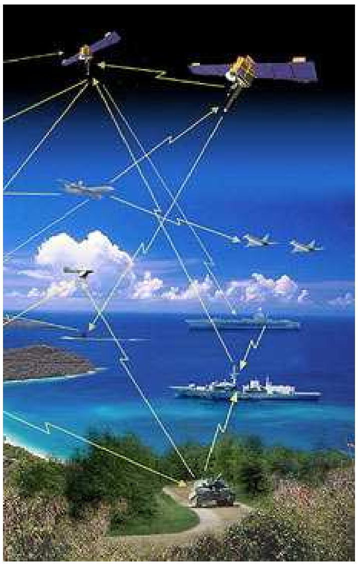


Figure 2.1: NEC Scenario [149]

onboard avionic systems. IMS uses three layers of software abstraction supporting transparency through the use of common hardware APIs, Line Replaceable Units (LRU) and virtual channel schemes that are topologically transparent.

In choosing the type of network timeliness, reconfigurability, safety, security and fault management are considered the key performance indicators. Reconfigurability of communication channels is introduced at a high level of abstraction via system blueprint documents that are used to configure the system state. Reconfiguration is static as blueprints are created and validated at design time but at run time the system may react to dynamic stimuli through the selection of appropriate configurations. The efficient use of bandwidth is identified by the Standards as a metric that may be sacrificed in order to achieve predictability of delay. The Standards do not stipulate but require system designers to define interfaces and technologies that are technology transparent. This definition must include generic functionality that is applicable to all networks and specified interaction with the networked IMS systems: the definition of logical interfaces to lower software layers and physical interfaces to the processing Common Functional Modules of the system. Among those architectures developed under the vision of NEC are Bowman HF radio, the DII, Falcon and Skynet 5. Bowman HF radio is one of the main media currently used for long-range tactical ground-based military communications.

HF radio is generally preferred over VHF due to the extended LOS coverage. Bowman radio provides the capacity for a subscriber or mobile node (MN) to switch between long and short-range communications: a Ground Wave VHF subnet and the Sky Wave subnet, based on a 3rd generation NATO STANAG 4358 protocol. A MN may move from one subnet to another without manually informing other subscribers of an address change. In addition, two levels of service are provided on each subnet voice only combat radio and voice or data networks. Bowman radio alongside IP has been suggested as a network solution to allow easier integration with other IP based systems [86]. IP provides a best effort service that enforces minimal constraints on transmission and simple conventions for addressing and routing. IP is therefore a good candidate to sit between various device drivers and physical media to create a flexible and modifiable network stack.

Bowman radios act as gateway routers interfaced to the PPP based wired network as well as providing a RF network interface. In order to manage the IP routing and addressing for dual interfaces, a bridging radio is used to connect different media within the same IP subnet, for example RF and PPP. This eliminates the need for radio inter-subnet routing and the same IP address is therefore assigned by the IPCP to the RF IP address of the radio as well as the PPP IP address. However, with timing sensitive data being transmitted over best-effort IP, it is necessary to manage QoS carefully in order to ensure that applications that take timeliness as a key requirement can still perform efficiently.

All of these communications mechanisms have been designed to support the service requirements of numerous military applications, particularly in terms of their timing requirements. While some of these applications may be able to operate efficiently over a best-effort network others require stringent controls on resource sharing when intolerant to variations in jitter or delay to the stream of traffic. All of these applications and their processes can be categorised as RT, NRT or heterogeneous RT. All RT processes are time-triggered, based on an internal system schedule or event-triggered by environment stimuli and QoS for these is expressed in terms of deadline achievement, but there is no benefit in delivering HRT packets early and this in fact may be detrimental to the system as early packets consume buffer resources.

Military onboard networks have traditionally streamlined QoS requirements by treating all transmitted data as having HRT deadlines, due to the operational safety impact of data losses. All critical command and control military data has the characteristics of HRT. For example Hawk AJT HUD parameters in flight control software are updated at a fixed frequency and must be received to deadline in order to avoid the implementation of HUD blanking as a safety precaution. Offboard communication packets could hold mission critical multimedia data such as video, audio or radar or control information that ideally should be treated as HRT data due to their critical importance. However, in actuality all of these applications will be supported as ISRT, with BCET and WCET.

2.2.3.1 Case Study: Wired AFDX

Support of HRT deadline achievement requires wired network architectures to rely on over-provisioning and careful control of resources in combination with predetermined fixed routing. Although these architectures cannot be directly applicable to the MANET where resource quality and availability are dynamic, the policies that have been implemented must be considered in order to evaluate an appropriate wireless solution for HRT support.

AFDX [2] is an HRT optimised switched network model based on Ethernet and Asynchronous Transfer Mode (ATM) [8] and has been designed specifically for the transfer of data between avionics subsystems, providing fixed maximum E2E delay and throughput guarantees. It is a good example of a network that introduces deterministic RT constraints and has been implemented on the Airbus A380, the Boeing B787 Dreamliner, the ACAC ARJ21 Xiangfeng and the Sukhoi Superjet 100 to date. The physical layer of onboard military communication networks utilises a range of new and legacy technologies, including 1553 bus and ARINC 629. ATM relies on a combination of connection oriented communication, traffic shaping, synchronous time division multiplexing and fixed packet size to support heterogeneous traffic requirements. It was developed to unify support in telecommunications and computer networks. The use of ATM is being investigated but AFDX, as a deterministic model, has also been suggested for use in place of ATM.

The AFDX network stack consists of an avionics specific physical layer of twisted pair or optical fibre traditional Ethernet cabling, a VL concept in the MAC layer, End Systems which are subsystems that must be embedded in each avionics node and AFDX switches. The network has a star topology and each switch connects up to 24 End Systems that can be cascaded to construct a larger network. AFDX switches are responsible for establishing point-to-point connections between a sender and several receivers with MAC routing as well as checking frame integrity and maintaining QoS guarantees. Each switch supports static reconfiguration for example with run-time blueprints, but not physical reconfiguration such as with Ethernet. In order to bound E2E delay [131] a switch uses very simple transmission and receipt packet buffering with FIFO store and forward transmission. The architecture enables TCP/IP or UDP use for data transmission, though generally the lower overheads resulting from UDP are preferred as a result of the smaller packet header and lack of acknowledgement.

In order to avoid packet collisions a number of control schemes are implemented. Permissible network loading is fixed far below capacity, link scheduling is carefully profiled and the redundancy management (RM) parameter is used to specify that a VL should transmit along dual lines of communication for transmission and receipt by independent AFDX ports; providing dual redundant full duplex communications. Rather than sending redundant packets across the same collision domain, two packets separated by a skew time are transmitted to two independent receiver ports via two independent switches in order to increase reliability and time determinism. Although not actually a point-to-point network, AFDX emulates one through the VL concept: the use of unidirectional multicast command and data paths for flows entering the network from each individual End System.

The network is carefully profiled with a fixed bandwidth allocation reserved for each VL and parameters for all End Systems and switches defined in static configuration tables loaded into each at startup. Each VL is defined by four values, an identifier, a BAG that is the minimum delay between the source emission of two consecutive frames on the VL and the minimum and more importantly maximum frame length, Lmax. Based on predefined configurations the software and network collaborate to define active VLs. There is therefore a potential path between any of the networked modules and in the event of failure the software and network can reconfigure VLs quickly in predetermined ways.

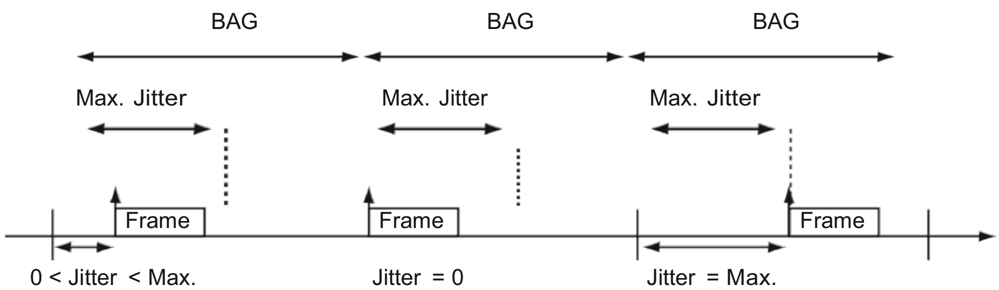
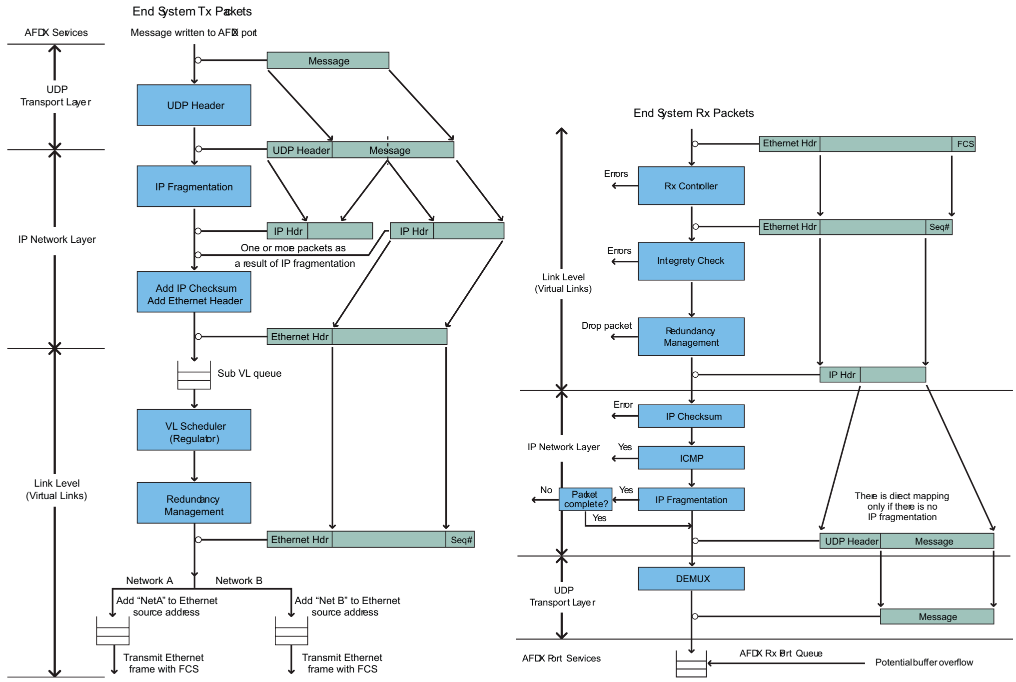


Figure 2.2: AFDX VL Regulation (Jitter) [2]

Traffic shaping is implemented in order to provide determinism in AFDX [150, 181] with predefined control of the timing of packet emission onto the network for each VL and E2E bandwidth reservation.

Bandwidth control is achieved with queue management and multiple bandwidth use strategies. The restriction of BAG and Lmax also prevents interference between VLs using the same physical link. In order to maintain traffic flows at a constant deterministic rate the data stream is distributed into timed slots with the BAG value determining the time interval between packets. This interval is restored at each switch in order to maintain guaranteed QoS, as described by figure 2.2.

The AFDX switch is also responsible for frame filtering and traffic policing based on the token bucket algorithm ensuring that arriving traffic is compliant with VL restrictions. However, each switch port has no scheduling or regulating



(a) Transmission (b) Receipt

Figure 2.3: AFDX Protocol Stack [2]

responsibilities, only performing simple integrity checks. As a result processing delay is kept to a minimum and frame transmission is kept at line speed, respecting the inter-frame gap. Error control is achieved with a CRC at the receiver with out-of-sequence packet rejection communicated to the network manager. Each switch and End System also performs link level RM after integrity checking.

AFDX supports true HRT performance and E2E delay control through careful scheduling in combination with accurate time stamping in End Systems. As a result of the static control of network resources, service guarantees can be provided concurrently to discrete classes of data with entirely independent and stringent transmission requirements while also providing the ability to constantly monitor the payload for each of these services. AFDX does still have a number of vulnerabilities relating to packet loss recovery and delay. As the network does not provide guaranteed delivery, applications must be responsible for their own retransmissions but if TCP/IP is not utilised packet losses cannot be recovered.

Scheduling may be implemented to stagger frames when many VLs are likely to coincide but this is at the expense of E2E delay. Ultimately it is the AFDX switch that is responsible for solving the contention caused by collisions between simultaneous packets from multiple asynchronous end systems, buffering and transmitting these after a brief queuing delay. Therefore overloading a switch with many concurrent VLs will lead to repeated packet loss due to buffer limitations; as a result

AFDX networks are generally over provisioned in order to reduce packet loss ratios.

The other problem introduced by AFDX relates to the fact that in spite of bounding delay and jitter, large E2E delays can be seen. AFDX frame management is vulnerable to faults such as network babbling that can trigger unwarranted system resets. Anand et al [12] therefore proposed the integration of redundancy management and integrity checking with a priority queue and duplication of reset messages to alleviate this problem, but at the expense of increased E2E delay.

If the network is not carefully profiled, congestion is also in fact shifted to the switch output ports, and traffic overload at one port as a result of traffic bursts can lead to variable E2E delays or even frame losses due to queue overflow. Finally, jitter is still introduced if a message arrives at a non-empty VL queue or in multiplexing all VL queues into RMU and transmission onto physical links, therefore VL regulators must introduce delay between frames according to the BAG to maintain maximum bounded delay. Although this traffic shaping improves jitter guarantees it may not provide ideal delay guarantees for high-speed requirements. As a result a number of proposals have been made to improve E2E delay guarantees [31, 136, 181].

2.2.4 Key Findings

The projects outlined in Section 2.2.3 indicate the service requirements of future military networks. The move towards improving support for ad hoc networking in the commercial domain has been mirrored in the military domain following the proliferation of COTS components supporting RT QoS requirements. In the commercial domain innovation is still concentrated around SRT multimedia support, particularly video streaming and VoIP rather than RT control applications or heterogeneous RT support but many of the novel frameworks developed are being adapted for use in critical systems.

The distinction between the requirements of SRT and HRT applications must be carefully considered when transitioning these COTS components to networks where timeliness can have an impact on safety. HRT systems have stringent E2E delay requirements, treating packet arrivals following a deadline as redundant or even dangerous in terms of system operations and the protection of equipment and personnel. SRT systems span a spectrum across which late packet arrivals may be tolerated and output value degenerated to decreasing degrees between the BCET and WCET. The network must guarantee SRT QoS to ensure an optimally efficient rather than a fixed reaction to a trigger. While the requirements of zero packet loss and constant absolute deadline achievement necessary for HRT support are still not feasible in ad hoc networks, these may be approached in order to meet the

QoS requirements of SRT applications with more stringent deadline and packet arrival requirements.

The capabilities of wired networks such as AFDX that support true HRT performance with deterministic characteristics and appropriate QoS must be considered in developing future network solutions. Future military networks such as those being developed under the NEC vision of collaborative operational systems and architectures aim to provide the QoS characterised by AFDX and other wireline frameworks but instead in ad hoc networks. The performance of these ad hoc frameworks must therefore be gauged in terms of the QoS metrics applied to all wired and wireless networks, such as E2E delay or available bandwidth, as discussed in Section 2.2.2.1.

When developing a model that provides QoS to dependable applications it is packet delay and its aggregation E2E delay that are the strongest indicators of performance. Many QoS control approaches implement traffic conditioning and corresponding packet dropping in order to reduce average delay. However, in MANETs based on an IEEE 802.11 MAC layer, uncontrolled packet loss has the greatest impact on maximum E2E delay. Excessive retransmission and backoff, as a result of channel noise, interference or collisions, create localised delay increase and ultimately reduce deadline achievement. Therefore, in order to provide bounded delay, such loss should be avoided.

A cross-layer approach has been embraced in the domain of ad hoc networking based on an increased understanding that frameworks of oblivious layers underperform in highly dynamic topologies and lossy conditions. It is only when protocols collaborate and react to gathered information on network conditions that E2E QoS provisioning can be achieved. Section 2.5 therefore evaluates approaches to interlayer collaboration that can be implemented in order to improve ad hoc network performance and achieve RT QoS.

2.3 Network QoS Control

2.3.1 Admission Control and Resource Reservation

This section reviews wireless network QoS control mechanisms for delay and loss control from the range of QoS control measures that do not explicitly implement a cross-layer approach. These mechanisms contribute to either delay or loss control or both. This review includes solutions for time critical applications and loss reduction in wireless networks and MANETs.

Load management for heavy load heterogeneous flows must be dynamic as traffic is bursty therefore at a given instant traffic rates can be disproportionate to

QoS allocations and it is necessary to ensure that the burstiness of one flow does not interfere with the performance of other flows. As a RT service requires more than best effort, a resource reservation protocol must control the traffic entering a network to ensure that bandwidth and maximum permitted delay are allocated to streams according to their tolerances. The network must make per-flow decisions on admission to a channel and how available bandwidth is reserved for streams based on QoS requirements.

Generally allocation may be (1) static: accessible only to a single flow until released, or (2) dynamic: allocated on a flexible basis to aggregated traffic according to QoS stipulations. Reservation must be extended to the entire transmission path to prevent deterioration of QoS. When this process is static, resources may be underutilised as a flow may not require all of the resources allocated but these cannot be transferred to another flow. At the same time dynamic allocation does not permit the provision of per-flow QoS guarantees. Admission control does not explicitly require reservation but many approaches have suggested the joint allocation of capacity and flow, via exchange of link capacity and flow requirements between the network and MAC layers results in greater link utilisation and reduced congestion.

Resource reservation schemes incur high overheads in multi-hop networks and many current methods of evaluating available bandwidth overlook the impact of contention. A MAC-layer reservation approach is therefore preferable as this takes into account contention for the shared medium. However, these schemes introduce new inefficiencies through the over conservative distribution of resources: the inability to consider spatial reuse and consequential throttling of parallel transmissions.

The high overheads and poor utilisation associated with resource reservation can translate to increased packet delay under the characteristics of the shared wireless medium. Both admission control and bandwidth reservation rely on bandwidth estimation techniques, which can be highly inaccurate under network dynamics [161]. Therefore, when these are implemented at the transport layer, they benefit from the receipt of signalling information from the MAC layer. In spite of the throttling of parallel transmissions, in a limited-frequency spectrum shared between many users load control must be implemented to prevent existing flows being induced to violate their QoS requirements.

In some wired networks, applications requesting a connection provide the network with a flowspec detailing the traffic characteristics of any packets sharing the same QoS requirements for storage in a traffic contract. The network refers to this contract when performing admission control to ensure that requesting flows are not permitted even when resources are available, if sufficient resources must be reserved for prioritised flows or handoff events or to prevent significant degradation in other connections. These requirements are then mechanised by the routers that perform packet scheduling [119].

Centralised admission control is not possible in ad hoc networks therefore collaboration between devices is required to ensure that each flow satisfies its bandwidth requirements. Although the admission control process does not guarantee QoS, the path discovery phase of routing is key to providing timing guarantees and is highly reliant on the stipulation of QoS requirements at this point. Admission control depends on accurate estimation of available resources, and in wireless ad hoc networking the available bandwidth metric must be calculated in light of contention within the carrier sense range of the node. The contention range of a radio reaches further than its communication range and carrier-sensing thresholds are tuned conservatively in order to avoid interfering with neighbouring receivers.

The performance of admission control is highly dependent on the bandwidth estimation method implemented. Approaches that rely on analytical modelling for collision prediction are topology dependent. They do not perform well in ad hoc networks due to the difficulty in each node maintaining information for the required period on all receiving nodes, the transmission probabilities and the traffic within carrier sense but outside of transmission range or hidden from the node. Routing is extremely important to the admission of flows as it is through appropriate path discovery that the QoS agreements of admission control can be upheld.

2.3.2 Ad hoc Routing

Routing protocols are responsible for the selection of the most appropriate from multiple available paths from source to destination over which traffic flows can be transmitted [22]. Load control can only take place once a route has been selected as E2E bandwidth and delay are highly dependent on the channel quality and node capacity at each of the hops along the chosen route. In ad hoc networks routing protocols must compensate for the lack of centralised control of resource management while also dealing with the exposed and hidden node problem [88]. Source based routing is not possible in large dynamic networks where the source cannot know the whole topology, therefore distributed routing is usually implemented with each mobile node selecting the next hop from among its one-hop neighbours; possibly a subset of these when flooding or multi-copy routing are implemented.

In networks providing service differentiation, routing protocols must refer to quantitative metrics, commonly delay and throughput, to satisfy application QoS requirements under the constraints of available resources [41]. Resource reservation is then used to guarantee these constraints. In these QoS-aware networks bandwidth and network delay are the most commonly used metrics followed by hop count, jitter, energy, loss probability and signal strength or distance. Depending on the protocol used one or several metrics are calculated for each discovered path and then each path is compared to identify the best one.

Metric selection can have an increasing impact on performance, depending on whether simpler additive or more complex multiplicative or concave calculations are used. Additive computations aggregate the metric for all links, such as with delay, jitter and hop number calculations, whereas multiplicative computations such as reliability and packet loss probability multiply the per-link metrics. Calculations of bandwidth are concave metrics as minimum and maximum values are required for each link [18].

Link and MAC layer metrics can also affect the QoS of a session and several protocols exist to jointly optimise these lower layer metrics with the network layer. The per-nodal MAC delay; frame delivery ratio, a statistical measure of arrival probability; predicted link lifetime, or link stability; normalised MAC load, the ratio of transmitted control frame bits to user data frame bits and relative node mobility-stability ratio of neighbouring nodes can all be used to gauge performance. When paths are selected according to the requirements of the new flow as well as channel conditions and in order to avoid existing flow requirement violation, packet loss can be controlled more effectively. QoS routing is congestion aware, providing smooth performance degradation but not explicitly reserving resources for flows.

For ad hoc networks, routing protocols can be subdivided into two classes: proactive and table driven or reactive and on-demand [134]. Proactive protocols maintain tables of routing information that are periodically redistributed throughout the network as a result of self-replicating updates, triggered according to topological changes. Responsiveness to re-configuration of the network under node mobility is therefore low and a large proportion of bandwidth must be attributed to routing table maintenance. Destination-Sequenced Distance Vector (DSDV), Wireless Routing Protocol (WRP) and Cluster-head Gateway Switch Routing (CGSR) are examples of proactive protocols that provide greater efficiency through reduced spatial diversity of updates. In contrast, reactive routing is initiated only when a source requires a path to a destination that it cannot itself generate, necessitating a path-discovery mechanism with an associated delay.

The reactive routing schemes that are best effort have been developed to include QoS awareness: Dynamic Source Routing (DSR) [74], Ad hoc On-Demand Distance Vector (AODV) [46, 117] and Temporally-Ordered Routing Algorithm

(TORA) [138], where all nodes in range compete for the medium. These are also the most widely used MANET routing algorithms that rely on on shortest path routing. TORA discovers several E2E paths maintaining these until they have all failed [138]. However, route maintenance depends on reliable, ordered receipt of control packets. TORA, DSR and AODV protocols all use query or Route REQuest (RREQ) control packets that are rebroadcast to all neighbours. The target node responds with a unicast update or Route REPly (RREP) via the broadcasting nodes to the sender, establishing an E2E path. Optional local repair can also be implemented with HELLO messages, repeatedly broadcast for link maintenance. However, the initial flooding of RREQs as well as the use of HELLO messaging can congest the network.

In mesh networks, link metrics such as Expected Transmission Count (ETX), Medium Time Metric (MTM), Weighted Cumulative Expected Transmission Time (WCETT) have been utilised for path selection but in MANETs node mobility, link breakages and the underlying process of the MAC layer such as repeated backoff introduce complexity in path selection and performance [111, 121]. In order to minimise E2E delay, ad hoc protocols have become QoS-aware through the introduction of delay estimation into the protocol based on constituents of E2E delay. The IEEE802.11 transmission and backoff mechanisms have been used to estimate the forwarding delay of a route [107, 145], although increased contention and rerouteing in MANETs can result in elevated queueing delays that are not considered in this estimation.

Delay Aware AODV (DA-AODV) [51] requires nodes to record accumulated delay along the E2E in routing tables and a route is only selected if it can fulfil application delay requirements. Maximum delay was therefore constrained with this approach. Similarly, power and Delay aware TORA (PDTORA) [68] extended TORA to eliminate the selection of paths using nodes that do not satisfy maximum delay and minimum power requirements. PDTORA showed up to a 60% improvement on E2E delay and packet delivery at high node speeds, when compared to TORA.

Initial control packet flooding is used by Ad hoc QoS On-demand Routing (AQOR) [174] to identify routes that can meet bandwidth requirements, with the lowest delay. The RREQ flood filters along paths that can fulfil QoS requirements until arrival at the destination. Source-based route selection then occurs based on RTT of RREPs returned along each path. The approach does not address the variable link capacities of MANETs, as RTT depends on symmetric links and E2E information can become rapidly unreliable in dynamic network conditions.

Social structure based proactive routing [162] has also been suggested for networks where topology dynamics are not known in advance with the concept that these dynamics are not wholly unpredictable as devices are likely to follow a deterministic schedule of movement. The protocol is self-boosting as nodes collect trace data from the network while running and circulate routing information when dramatic changes to their routing tables are discovered. However, high overheads are incurred by the regular, network-wide updates required that increase with the rate of topology change and rapidly degrade QoS provisioning when proactive routing is implemented in MANETs.

In ad hoc networks route discovery and path selection are highly dependent on the level of node mobility. When nodes continually move across and between channels through access point or forwarding node handoffs the network topology changes dynamically and as a result service users will experience varying levels of channel quality. The loss of a physical link as a result of mobility requires the network to manage complex changes in channel access and early notification can aid this process.

Notification can also enable applications to optimise their output according to available channel conditions or loss events. Handoffs between channels can be of two types: (1) Vertical, where a node moves between APs of different technologies or (2) Horizontal, between APs of the same technology. In both cases early notification can ensure the seamless adaptation of higher layer services to underlying wireless technologies, while also exploiting their properties [52]. It is also the provision of this information from lower to higher protocol layers that can benefit the management of congestion and packet loss recovery. If handoff notification is provided to the transport layer, this can be used to prevent the initiation of retransmission timeout and exponential backoff and ensure the commencement of a fast retransmit [19] to improve retransmission delay and throughput by up to 75% and 25% respectively.

Once notified of a signal loss, the network must manage node handoff to the next appropriate link, selecting a new interface, channel and route, updating the traffic contract and removing the old route. For RT or continuous streaming applications this entire process must be seamless. Bellavista et al [19] divided node handoff into two phases: evaluation, where information is gathered on interfaces and connectors in current use and available, and continuity management, where the evaluation result is used to select a new interface-connector pair and perform a handoff to a new access point. The first phase is implemented in interface firmware and is based on the RSSI or Signal and Interference to Noise Ratio (SINR) at the origin and destination, when these values are low it is assumed that network performance is limited. The second phase largely depends on the results of the first, though connector-signalling messages are used to update on the node location and Mobile IP can be involved. Network performance is maximised when the optimal access point, primarily for the requesting node but also for the application is selected.

An alternative to the aforementioned routing approaches is multicast routing that has been widely implemented in wireless networks. Multicast is gradually being used in ad hoc networks due to the efficient utilisation of bandwidth [36, 106, 109]. A multicast sender transmits to a subset of network hosts via a single address on the basis of one of three categories of algorithm: flooding (multicast and broadcast), source based or core based routing. Epidemic and multi-copy routing are examples of flooding mechanisms and are useful for reliability provision in dynamic networks, but waste bandwidth, incur high processing overheads and increase the probability of congestion. Distance Vector Multicast Routing Protocol (DVMRP) is an example of a source based protocol that requires sources to maintain multicast trees of all group participants, routing is very efficient but regular updates lead to large increases in overhead.

Multicast Ad hoc On-Demand Distance Vector (MAODV) [133], Ad hoc Multicast Routing (AMRoute) [102], STAMP and Ad hoc Multicast Routing protocol utilising increasing ID numbers (AMRIS) [167] are examples of core-based protocols where one core host maintains the multicast tree. However, core based routing results in increased traffic on links to the core and promotion of the core node to become a single point of failure with ramifications for the rest of the multicast tree. Mesh-based multicast routing protocol with Consolidated Query Packets (CQMP) is a mesh based multicast protocol [47] that supports a high delivery ratio under mobility, high throughput and low overhead with multiple sources. Here each core distributes multicast mappings to one or more core addresses. CQMP is reliant on routing information, distances and beacon-basted localisation from an underlying unicast routing protocol.

Robust Multicasting in Ad hoc Network using Tree (ROMANT) [156] in contrast avoids dependence on unicast routing, while providing all of the benefits of CQMP. Unfortunately ROMANT relies on node synchronisation that while also useful in RT communications is not feasible for implementation in multi-hop networks, as discussed earlier. A number of location based multicast protocols such as Location-Based Geocasting and Forwarding (LGF) or Scalable Position-Based Multicast (SPBM) have been suggested, requiring senders to utilise geographical localisation information from GPS, Bluetooth or another source to determine the location of the receiver. Location or query flooding or a pre-defined quorum on node-associations can then be used for path distribution [106] [84].

2.3.3 Collision and Congestion Control

Collisions and associated retransmissions are one of the key causes of congestion and therefore collision avoidance schemes have a large impact on congestion avoidance. Transmissions between two nodes in a shared medium consume bandwidth not only from those two nodes but also from all neighbouring nodes that are within carrier sense range. Some QoS mechanisms have therefore been designed to prevent contention for the shared medium and corresponding collisions. The aim of this is to maintain QoS guarantees during packet transmission. CSMA [152] is a media access protocol that ensures a channel is free prior to transmission based on receiver feedback, or carrier sense and recovers from packet loss with random backoff and retransmission. Collision detection (CSMA/CD) extends CSMA to immediately halt transmission when a collision is detected. However, radio broadcast nodes rely on a single antenna for transmission and receipt. Therefore, CSMA/CD cannot be implemented in a wireless network as collision detection cannot occur at the same time as data transmission. Additionally, use of a long-range receiver can affect the ability of the MAC layer to detect the channel state, as a result of signal fading [54, 67, 88, 129, 132].

Without the functionality of CSMA/CD, collisions and difficulties of media access control, such as the the hidden and exposed node problems occur [79, 164]. These are illustrated in figure 2.4 where node B is in range of A and C which are not in each others range. If C and A transmit packets away from B, the exposed node B may be unnecessarily throttled. If A transmits to B it is unable to sense the presence of the hidden node, C. The partial solution in IEEE 802.11 networks is to use MAC layer handshaking of RTS/CTS packets, also known as virtual carrier sensing or CSMA with collision avoidance (CSMA/CA). However, RTS/CTS relies on consistent traffic rates and has also been shown to reduce network performance as a result of interference errors or even cause blocking multiple nodes, leading to congestion [127, 146, 172].

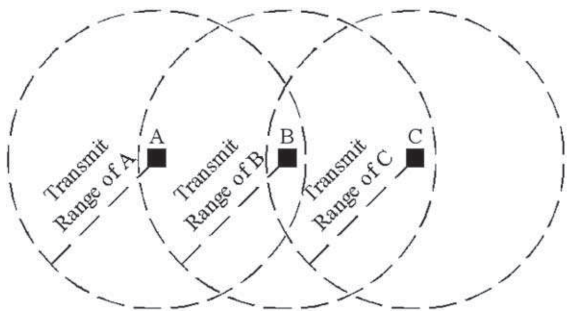


Figure 2.4: The Hidden and Exposed Node Problems

Network Congestion occurs when data packets must wait for service at a resource bottleneck due to traffic flows exceeding the available capacity of buffers over the data path. This may be as a result of nonconforming sources increasing flow rates beyond allocations, multiple collisions and retransmissions on shared links or bottlenecks between high and ow bandwidth links. As buffer occupancy increases as does the queueing delay experienced by neighbouring flows. When limited buffer resources are exhausted network reliability decreases in terms of service provisioning as packets are dropped. Loss recovery schemes may then be implemented at the transport layer. When loss rates occur between 10^{-3} and 10^{-1} in wireless networks the retransmissions utilised by these schemes can reach rates at which they greatly reduce network utilisation and increase the delay experienced by data packets.

MANET traffic is commonly bursty and is therefore prone to congestion and corresponding packet loss, the high bit-rate requirements of SRT and HRT streams can also compound this problem. High protocol processing and communication overheads can result in increased jitter, as nodal processing is no longer possible at line speed. Packet bursts may lead to extended queueing delays for all flows sharing the link. If congestion goes uncontrolled network congestion collapse is possible, resulting in worst-case E2E delay and goodput, as well as repeated and escalating packet loss. Therefore congestion control is generally implemented to detect and prevent congestion or congestion avoidance can be implemented if early notification is available.

Increases in RTT can be taken as an early implicit notification of congestion and packet loss as a late notification. Hop-by-hop congestion control schemes allow a node to feedback its congestion status to the previous node so that it can adjust its transmission rate, if congestion is persistent the source node will eventually receive the congestion data and throttle its transmission rate. However, the congestion data must travel through a congested network to reach the source. Like Explicit Congestion Notification (ECN) [126], this method of notification that relies on increasing backpressure is considered explicit notification and unlike RTT or packet loss is a more reliable identifier of congestion in networks where packet loss and delay can have numerous causes.

Overhearing neighbouring node packet receipt has been suggested as an implicit backpressure congestion control mechanism [137]. This Cooperative cross-layer Congestion Control (CXCC) approach therefore responds to changes in local conditions before forwarding the next packet, by throttling the downstream node under congestion.

In wired networks congestion control is either based in the source or routers, the latter is generally implemented at the transport layer most commonly with TCP which maintains a transmit window of a size proportional to the available

Bandwidth Delay Product. Wired TCP congestion control assumes that congestion is the sole cause of packet loss and on a loss event drops its transmit window. The normal operation of TCP involves the gradual increase of a CWND variable for each RTT. Initial transmission rates increase gradually and exponentially with each RTT up to a threshold after which they increase more rapidly. On a loss event the transmission window drops rapidly, as a result the entire process is considered AIMD. TCP has numerous drawbacks in wireless networks that have been considered by Wang et al [160] including:

- High overheads associated with connection establishment
- Discrimination against nodes with poor connectivity
- Higher packet losses as a result of the delayed reaction of E2E congestion control
- Energy inefficiency in E2E retransmissions.

Improvements to TCP for underlying wireless technologies have included modifications for further notification of the causes of packet loss at the transport layer [94, 104] and lower layers using reliable link-layer protocols with FEC and ARQ [93, 168]. FEC [60] relies on redundant encoding bits to detect errors within a message. ARQ is the use of acknowledgement of correctly received frames and timeout before an acknowledgement is expected, for error control. Lower layer signalling has also been used to improve the fairness of TCP with Random Early Detection (RED), an AQM and congestion avoidance algorithm that drops packets once the average queue size exceeds a threshold value. LRED [53] (Loss ratio based RED) utilises MAC layer signalling on the average number of transmission retries rather than average queue length to calculate the probability that a packet is dropped. The congestion threshold is therefore defined as a minimum number of MAC layer retry attempts.

NRED (Neighbourhood RED) in contrast is a distributed algorithm designed for ad hoc networking where nodes sense channel utilisation or probe neighbouring nodes for this information. The utilisation is then used to gauge average neighbouring queue lengths and if higher than a threshold, to provide an early indication of congestion. When congestion is notified, each node calculates the probability at which packets will be dropped. RED-RT [38] extends RED to provide differential treatment for RT and NRT, with RT packets only dropped according to probability calculation when queue lengths exceed a maximum threshold. All NRT packets are dropped at the maximum threshold, but dropped according to probability above the minimum threshold.

RED has also been modified for ad hoc networks, AHRED [1] (Ad hoc Hazard RED) that drops packets using a probabilistic Weibull failure rate to reduce packet loss and delay. Under bursty wireless traffic conditions, a fixed buffer size can create unnecessary delays and underutilisation of resources. A$*$ has, therefore, been proposed by the authors in [96], along with the ALT (Adaptive Limit Tuning) algorithm to improve TCP throughput and queuing delay. This is addressed by adjusting the buffer size when the link is idle or busy, according to maximum QoS requirements, but does not provided bounded delay or loss guarantees.

All NRT packets are dropped at the maximum threshold, but dropped according to probability above the minimum threshold. However, even with lower protocol adaptation TCP use still results in high link error rates of up to $10^{-6} bits/s$ [116]. TCP provides improved performance by implementing congestion control and handshaking for reliability. A number of congestion avoidance algorithms exist for TCP, which respond to a lack of ACKs with traffic rate throttling of up to 50%. Excessive reduction in traffic rate introduces increasing delay and jitter to flows therefore TCP is not appropriate for delay-sensitive applications. RT applications generally use UDP at the transport layer but still have a congestion avoidance requirement.

DCCP [85, 92] has been proposed as a transport layer protocol for applications with timing constraints where packets are useless to the receiver if reliable sequenced transport combined with congestion avoidance is implemented. Like UDP, DCCP provides unreliable transmission but adds congestion control that is not implemented at the application layer. One of two mechanisms can be selected for congestion control according to application requirements: CCID-2 that utilises a CWND and AIMD or CCID-3 that provides TCP Friendly Rate Control that adjusts transmission rates according to packet loss ratio. However, in wireless networks, DCCP is also subject to the same performance faults as TCP.

Networks can implement traffic conditioning, including classification, policing and shaping to maintain the conformity of flows to the predefined traffic contract thus avoiding network congestion. Traffic policing is administered on a per packet basis where offending packets are dropped or, if resources are not currently overutilised, marked as non-conforming to be dropped with the highest priority. Traffic shaping uses bandwidth throttling or the introduction of delay to ensure per-flow conformity to the contract, though the latter is generally avoided when dealing with RT streams. The leaky bucket algorithm is a common method of throttling transmission rates of received bursty traffic to create a constant stream of outgoing traffic.

Collision avoidance mechanisms alone cannot prevent contention-related packet loss but the unmodified carrier sense capability of radios is still important in ensuring the fair allocation of bandwidth and collision avoidance to nodes within carrier range. Transmissions between two nodes in a shared medium consume bandwidth not only from those two nodes but also from all neighbouring nodes that are within carrier sense range. Contention aware QoS control can then be used to avoid oversubscription of resources.

2.3.4 Packet Scheduling

When a packet is dequeued for transmission it is scheduling algorithms that make the decision of which packet is the head of line based on priority, delay requirements, nodal congestion or other QoS requirements. Queues exist at multiple protocol layers inside a node and must be serviced appropriately in order to ensure maximum and fair resource utilisation and that QoS guarantees are upheld. Wired network schedulers rely on traffic and queueing statuses, as a basis for prioritising packets but in wireless networks with varying channel capacities this is insufficient. Packet scheduling is utilised to ensure efficient link utilisation; provision of delay bound guarantees, smooth service degradation and protection from non-conforming sessions; fair redistribution of resources across sessions and guaranteed short-term and long-term throughput [50].

Scheduling disciplines are either rate-controlled or rate-allocating: the former permits flow rates higher than the guaranteed rate for a connection, providing that guarantees can still be met for other connections, the latter ensures packets never exceed the guaranteed rate. Rate allocating disciplines are work-conserving and are never idle if a packet is awaiting transmission. A bandwidth-preserving server that optimises use of all background time or algorithms such as Weighted Fair Queueing (WFQ), Delay-Earliest Due Date (Delay-EDD), Self-Clocked Fair Queueing (SCFQ), Weighted Round Robin (WRR) and Deficit Round Robin (DRR) are all work-conserving.

A constant utilisation server and Hierarchical Round Robin (HRR) and Jitter-EDD algorithms are rate-controlled, implementing algorithms that are non-work-conserving and may be idle, even if packet backlog occurs, if a higher priority packet is expected. These disciplines may utilise round robins, timestamping or frames. Round robin schedulers are easy to implement as they service queues according to some predefined order but cannot provide timing guarantees.

Timestamping may be applied before packets are placed in queues so that head of line packets are sorted in increasing timestamp order and the packet with the lowest is scheduled first. Timestamping incurs processing delay but provides good QoS guarantees. Finally frame-based schedulers divide time into fixed or variable frames to which a portion is ascribed to each session: examples of the latter are WRR and DRR and the former, HRR and Stop and Go Queueing (SGQ). When the frame size is fixed the scheduler will remain idle if a packet does not utilise a full reservation. Such non-work-conserving implementations are subject to higher packet delay and generally to not utilise bandwidth efficiently but are more appropriate for flows with high dependence on bounded jitter [101].

The simplest scheduling algorithm to implement is FIFO but this does not provide QoS guarantees. WFQ is an algorithm that differentiates packets from different connections into different queues, each of which is serviced according to a round robin. Each connection is assigned a weight according to which a level of service is allocated, therefore higher priority flows can be allocated more bandwidth and suffer shorter delays. However, delay cannot be bounded with this method of prioritisation. The WRR algorithm similarly separates flows into individual queues, serviced by round robin according to weight and mean packet size. Deficit Round Robin (DRR) or Deficit Weighted Round Robin (DWRR) is a modified WRR scheme that serves the queue with a deficit counter greater than the size of its head of line packet and then decrements the counter; if this clause is not true the deficit counter is incremented. This simple weighted prioritisation cannot distinguish between desired timing and relative importance therefore several algorithms use relative deadlines to service packets, such as Delay-EDD. Jitter-EDD extends Delay-EDD to further support jitter-sensitive packets through the inclusion of minimum jitter requirements in the scheduling decision, thereby ensuring that both the single hop and E2E delay are bounded [101].

Several QoS provisioning approaches consider optimising the source bit rate according to channel conditions in order to minimise congestion and delay. This is generally only possible in a select group of SRT applications including media streaming, video conferencing and interactive network gaming. If the multimedia transmission rate selected is lower than the optimal transmission rate along a path a large amount of jitter is introduced into the stream and if it is higher packets will be dropped at intermediate nodes and the receiver. In order to reduce distortion in video streaming a select set of parameters may be optimised. The application layer participates as it contributes parameters relating to per-packet loss distortion effects and can adapt the source rate according to network capability information provided by the data link or physical layer.

Packet scheduling is key to meeting deadline requirements and MAC layer scheduling has the added benefit of being contention aware. QoS aware scheduling for multi-hop networks is still an area that requires further performance analysis and testing [50] as multiple priority queueing requirements will result in increased packet delay. Of the QoS aware approaches, priority-type scheduling and timestamping for separate packet and slot queues combined together provide the best assurances to delay and jitter-sensitive applications. This is at the expense of reduced response rates with higher node computation and processing requirements. Therefore, while the provision of bounded delay through QoS aware scheduling of RT packets is desirable, it will also result in increased E2E delay.

2.3.5 Service Differentiation

QoS service requirements from a network are typically application specific as different flows require different network services and can be divided into three types: best effort, guaranteed service delivery for a session duration and an approach that aims to support both schemes within a session but does not provide guarantees for the entire session. QoS provisioning begins with the separation of traffic into classes according to requirements, scheduling is then used to control the interaction between these classes and to fairly distribute services. Service differentiation models are therefore extremely popular in providing support to heterogeneous networks, while also maximising resource use.

The IntServ model [89] has been widely implemented in wired networks to provide per flow session guarantees through static resource reservation with RSVP [23] across an entire path, but suffered from a lack of flexibility and scalability. Real Time DSR Protocol with Delay constraints (RTD-DSR) [69] combines delay-sensitive routing and IntServ-based admission control to ensure QoS achievement of flows. New flows are admitted based on delay counter incremented at each hop during route discovery compared to their deadlines and those of existing flows. The usefulness of service differentiation depends on the lower level hop-by-hop schemes implemented, for example, different QoS requirements may be serviced at the link layer with prioritisation according to minimum delay requirements or with an appropriate FEC or ARQ scheme according to reliability requirements. This is where DiffServ has gained popularity in IP networks in providing differential treatment and link layer services to classified traffic flows according to priorities assigned by the network operator, particularly in terms of multi-queue processing, without explicit recourse to resource reservation, although still including RSVP.

DiffServ, while sufficient in wired networks does not compensate reliably for the drawbacks of the wireless environment and in fact has been shown to drastically increase processing overhead, elicit unpredictable E2E behaviour and detrimentally result in the global synchronisation of streams practising congestion avoidance [5, 153]. Service differentiation in Stateless Wireless Ad hoc Network (SWAN) [5] is a stateless QoS mechanism based on the DiffServ model that provides traffic classification and servicing with different priorities without requiring per-flow information or signalling. Rate control is automatically configured based on measurements of MAC delay and the bandwidth available to new RT connections is estimated according to neighbouring flow rates. Admission control is administered to UDP RT flows at the sender but SWAN stops rather than throttles a RT session under congestion making it inapplicable to highly dynamic networks as high overheads would be incurred by regular session re-establishment.

EuQoS [182] is another QoS framework for E2E guarantees over heterogeneous networks without recourse to application specific signalling protocols. Users first select the QoS requirements of each application according to predefined classes, which are then serviced depending on the provisioning model in use. Within the Loose model path selection resides in the routing protocol while a signalling protocol dynamically reserves resources along the path rather than on an E2E basis. The Hard model assumes known QoS across any one link. Although resources are not explicitly reserved on a per-flow basis, they are reserved per class and allocated dynamically at flow admission based on the state of the shared medium. A Control Plane associated with a network technology dependent and a network technology independent layer is responsible for reserving all connections in both directions for a user. The level of overhead associated with EuQoS is high due to the signalling required for dynamic binding of resources to a specific path, while this also leads to underutilisation of available resources.

Tian et al [153] adapted the class based provisioning seen in DiffServ to provide a routing protocol independent cross-layer adaptation appropriate for RT delay requirements, CLA-QOS. Here the application layer in the source node dynamically adapted the class of emitted packets according to periodic updates on the loss ratio from the destination application layer. In using the loss ratio rather than a running average of E2E delay packets dropped by the scheduler for deadline overrun counted towards the overall computation of channel quality. The network layer was also responsible for E2E delay measurement for use in link layer scheduling and differentiation according to timing requirements. These schemes have been important in terms of QoS provisioning to RT traffic as guarantees are provided with minimal overhead and maximal link utilisation. However, the delay requirements of HRT and some SRT streams could not be met by service differentiation models in a wireless environment due to the oblivious implementation of control measures such as routing and rate allocation.

2.3.6 Key Findings

The control mechanisms discussed within this section indicate that many approaches can be taken to delay and packet loss control in wireless or ad hoc networks. Although discussed individually, many combinations of these mechanisms have been implemented. These form frameworks designed to achieve specific performance goals such as high quality multimedia playback, improved TCP congestion control (in spite of high packet loss ratios due to MN mobility) or the prevention of resource oversubscription. Research in the field of ad hoc RT support has predominantly concentrated on multimedia applications, such as video streaming and VoIP, that are not subject to the vital safety or mission critical delay requirements seen in the military domain. The frameworks and control mechanisms discussed in Section 2.3 outline a lack of provision for dependable applications. It is only through cross-layer optimisation of control mechanisms such as admission control, resource reservation or error recovery that overall QoS can be satisfied in an ad hoc network, which by its nature frustrates the QoS requirements of RT applications.

A number of models rely on bandwidth reservation to guarantee QoS to traffic flows and to prevent packet errors resulting from overloading of the shared medium. Congestion control and traffic shaping are then responsible for ensuring that flows adhere to their original specifications. However, rather than controlling delay the high overheads of messaging and poor utilisation of bandwidth associated with resource reservation can translate to increased packet delay. Load control cannot be ignored altogether in ad hoc networks where a limited-frequency spectrum is being shared between large numbers of users. The unmanaged admission of new flows can induce existing flows to violate their QoS requirements, corresponding to reduced resource availability and increased interference. A centralised approach to admission would required the same unacceptably high level of signalling as resource reservation therefore distributed admission control must be relied upon. Participating MNs must honour the requirements of other nodes and cooperate to provide high utilisation as well as reduced collisions. Control of traffic loading has ramifications for packet loss resulting from congestion and collisions and as a result bounded delay cannot be provided without its implementation, but the result of the admission control phase is still only the communication but not galvanisation of QoS.

Admission control and bandwidth reservation rely on current bandwidth estimation techniques, which can be highly inaccurate. Therefore, when these are implemented at the transport layer, they benefit from the receipt of signalling information from the MAC layer. With this information load control may be contention aware but can still lead to the over conservative distribution of resources: the inability to consider spatial reuse and consequential throttling of parallel transmissions. However, in a limited-frequency spectrum that is being shared between large numbers of users, some degree of load control must be implemented. The unmanaged admission of new flows can induce existing flows to violate their

QoS requirements, leading to reduced resource availability and increased interference. To ensure bounded delay, admission control can be implemented through distributed cooperation between nodes, preventing congestion and collisions. In contrast, bandwidth reservation subjects nodes to high overheads in maintaining E2E reservations in dynamic topologies. Rather than supporting QoS, the high overheads of messaging and poor utilisation of bandwidth associated with resource reservation can translate to increased packet delay.

Routing is also extremely important to QoS control as it is through appropriate path discovery that the QoS agreements of admission control can be upheld. When paths are selected according to the requirements of the new flow as well as channel conditions and in order to avoid existing flow requirement violation, performance can be controlled more effectively. Ultimately, if routing is based solely on topological characteristics, allocations are fair but resource use is not maximised. Signalling between routing and admission control enables optimal path selection because alternatives to the shortest path may be identified if it is congested. QoS aware routing can therefore be congestion aware: providing smooth performance degradation but not explicit reservation for new flows. The congestion awareness of all of these approaches has been emphasised in light of the limitations of traditional congestion control mechanisms, both in RT support as well as ad hoc networking. There have been many attempts to improve TCP congestion control in order to prevent flow-throttling reactions to packet loss that results not from congestion, but from ad hoc path tear down. However, when congestion control is implemented at the transport layer, TCP-like reliability mechanisms must still be provided. Therefore, mechanisms that access information from the MAC or network layers can provide better support for UDP-like unreliable traffic than that realised by transport layer congestion control.

MANETs are subject to high and time varying packet error and loss rates as a result of node mobility and resulting local resource overloading, congestion and contention for the shared medium. In ad hoc networks node mobility contributes to packet loss ratios due to variation in local loading or physical link loss as a node moves from one AP to another. Increases in network load due to mobility can also result in increased network congestion. All wireless transmissions are subject to the problems of the shared medium: packets in the same neighbourhood contend for a low available bandwidth, with a high probability of collision and interference induced packet loss. The implementation of control measures developed for wired networks can introduce significant variation in packet delay, particularly with the use of congestion control, error recovery and traffic shaping. These approaches, that rely upon bandwidth throttling and the direct or indirect introduction of per-hop delays, are generally not implemented with RT applications but are also frustrated in ad hoc networks by the lack of centralised control.

Error recovery mechanisms have been suggested as a means of recuperating packet losses but even when implemented on a hop-by-hop basis the retransmission of packets and sequencing utilised introduce unacceptable delay jitter into RT streams. The poor channel conditions in ad hoc networks can also result in multiple retransmissions and correspondingly increased channel delay. RT traffic is bursty and burst errors introduce the problem of association of multiple retransmissions. Control messages have been used in WiMAX networks to create a relationship between packets in bursty retransmissions but [52] have shown that these can occupy up to 60% of available bandwidth. The bounded delay requirements of RT transmission therefore limit the extent and effectiveness of all retransmission based error correction and link layer retransmission schemes.

Packet scheduling is key to meeting deadline requirements and is therefore key to providing QoS guarantees to RT traffic. Furthermore MAC layer scheduling has the added benefit of being contention aware. QoS aware scheduling for multi-hop networks is still an area that requires further performance analysis and testing [50] as multiple priority queueing requirements will result in increased packet delay. Of the QoS aware approaches, timestamping for separate packet and slot queues provides the best assurances to delay and jitter-sensitive applications. This is at the expense of reduced response rates with higher MN computation and processing requirements. Therefore, while the provision of bounded delay through QoS aware scheduling of RT packets is desirable, it will also result in increased E2E packet delay.

2.4 Wireless, MANET and Safety Critical Protocols

This section provides a list of protocols that are currently in use in the wireless, MANET and safety critical domains (Tables 2.1-2.2). This list is not exhaustive, but includes examples that have been referred to in this Chapter. The sections in which these protocols are discussed are given in the final columns of each of the tables.

Table 2.1: Widely used Wireless Protocols

QoS Mechanism	Load Control	Addressing/Routing	Collision/Congestion	Packet Scheduling	Service Differentiation	Jitter/Loss Recovery	**Section**
Wireless Protocol							
Automatic Repeat reQuest (ARQ)						*	2.3.3
Carrier Sense Multiple Access (CSMA)						*	2.3.3
CSMA with Collision Detection (CSMA/CD)			*			*	2.3.3
CSMA with Collision Avoidance (CSMA/CA)			*			*	2.3.3
Datagram Congestion Control Protocol (DCCP)			*				2.3.3
Differentiated Services (DiffServ)	*				*		2.3.5
Explicit Congestion Notification (ECN)				*			2.3.3, 2.5.2.1
Forward Error Correction (FEC)						*	2.3.3
Integrated Services (IntServ)	*				*		2.3.5
Internet Protocol (IP)		*					2.2.3
Random Early Detection (RED)			*	*			2.3.3
Real-time Protocol (RTP)						*	2.5.2.1
Real-time Control Protocol (RTCP)						*	2.5.2.1
Resource ReSerVation Protocol (RSVP)	*						2.3.5
Transport Control Protocol (TCP)			*			*	2.3.3
Wireless Routing Protocol (WRP)		*					2.3.2

Table 2.2: Widely used MANET and Safety Critical Protocols

QoS Mechanism	Load Control	Addressing/Routing	Collision/Congestion	Packet Scheduling	Service Differentiation	Jitter/Loss Recovery	**Section**
MANET Protocol							
Ad hoc Multicast Routing (AMRoute)		*					2.3.2
Ad hoc Multicast Routing utilising increasing Id numberS (AMRIS)		*					2.3.2
Ad hoc On-Demand Distance Vector (AODV)		*					2.3.2
Dynamic Source Routing (DSR)		*					2.3.2
Multicast AODV (MAODV)		*					2.3.2
Temporally-Ordered Routing Algorithm (TORA)		*					2.3.2
Safety Critical Protocol							
Avionics Full DupleX switched ethernet (AFDX)	*	*	*	*	*	*	2.2.3.1
Asynchronous Transfer Mode (ATM)	*		*	*	*	*	2.2.3.1

2.5 Implementing Cross-layer Optimisation

2.5.1 Definition of Cross-layer Optimisation

It has been widely concluded that layered, OSI-type architectures perform poorly with a wireless physical layer [56,175]. This performance further deteriorates with the independent mobility and intermittent connectivity of nodes in MANETs [10, 144,163]. The practice of withholding internal layer parameters from other layers facilitates the fast development of interoperable systems. Conversely, this limits performance, due to a lack of coordination between the efforts of coexisting protocol layers. Oblivious layers are unable to distinguish between possible causes of packet losses and errors, or to estimate and fairly distribute available E2E bandwidth. Under unreliable conditions, oblivious layer use results in the inefficient use of network resources and duplication of efforts at multiple layers. For example, data transmission rates cannot be dynamically tuned according to varying channel quality. As a result, wireless networking is being extended to include the communication of signals between layers. This usually involves increased responsibilities at the lower layers that represent these scarce resources. The variety of QoS provisioning schemes that follow this paradigm sit under the title of cross-layer optimisation or design.

QoS considers the ability of a network to provide a range of services, each suited to a certain class of flow. It is evaluated in terms of metrics such as bandwidth, delay, and jitter. QoS control measures, including those discussed in Section 2.3, are used to improve network performance in order to meet particular goals of the traffic flows being serviced. Section 2.3 considered oblivious QoS control approaches as it is the QoS control measures, that make up a network service model, that guarantee a minimum level of performance. When cross-layer interaction is added to this model, QoS control becomes more sophisticated, but potentially more complex. These control measures are then mutually tuned, to provide increased capabilities to guarantee a certain level of performance to a flow.

In order to reflect this trade-off between performance and complexity in cross-layer designs, Kliazovich et al [82] therefore proposed the classification of all cross-layer schemes into two categories (1) weak cross-layering and (2) strong cross-layering. The former generalises the traditional interaction between adjacent layers of the protocol stack to non-adjacent interaction and the latter takes a joined up approach to algorithm design which may be implemented in any entity at any layer; possibly resulting in the loss of individual features of the different layers. While strong cross-layering provides higher performance this is always at the expense of flexibility in deployment scenarios and the reduced cost and complexity that is offered by its weak counterpart. This is because increased interlayer interaction, in increasing the complexity of the network stack [76], undermines a significant factor that has supported innovation and upkeep of communication networks, the ability to exchange protocols at one layer without considering other layers. Therefore, while moving away from the OSI model it is still preferable for new architectures to use a weak cross-layer approach, placing minimal constraints on future modifications and maintaining the flexibility to support unmodified areas of the protocol stack.

One, of a number of approaches, may be implemented for the propagation of signalling information, depending on whether this information is to be passed across the protocol stack within a node or from a protocol layer in one node to another node. One of the simplest methods of direct signalling implemented in a single node is the use of packet headers or structures to encapsulate signalling information this is propagated across the protocol stack providing accessibility to subsequent layers along the processing path. Signalling data can also be inserted into a defined section of the packet structure, allocated to each layer, on transmission or receipt. Only layers that implement the cross-layer modification can edit and access their corresponding packet structure segments, while non-participating layers need not perform any of these tasks. For direct interlayer communications out-of-band control protocols, have been utilised for upper layer notification, allowing non-neighbouring layers to exchange messages without bypassing adjacent layers, enabling high-speed signalling and reduced overhead.

Alternatively, local profiles have been applied for the storage of periodically updated information within the node. These profiles can be created on a per layer basis, for access by interested layers and contain parameters that are abstracted and stored. However, even if a node did have sufficient memory capacity, the local storage of parameters would not provide the direct access necessary in high-speed networks. Distributed servers are also candidates for gathering, abstraction and management of signalling information from protocol layers within a single node to a database that is then accessed by interested layers without the requirement of protocol layer tuning. Callback functions can be added to improve abstracted signalling architectures, providing the opportunity for event based signalling, allowing one protocol layer to register these with a second layer for execution on a specific events occurrence at the second layer. These functions can also be defined and installed by the protocol, registered to the library at one point in time and invoked by a cross-layer entity. Such an interlayer middleware entity must be responsible for the abstraction of signal parameters and the coordination of their usage and is also known as an optimiser or interaction scheduler [83,147]. In taking this responsibility away from the protocols themselves and avoiding inter-protocol entanglement, this provides a trade-off between performance and network-stack complexity in the architecture of a cross-layer approach.

The management middleware entities proposed to date can be divided into four classes, as defined by Foukalas et al [52]:

- **External centralised middleware** that is hosted by a single external network node that abstracts and manages all cross-layer signalling and optimisation.

- **External decentralised middleware** that is one of several management entities acting in concert, each within a different node.

- **Internal intralayer middleware** that is implemented within the node protocol stack, sitting between the application layer and the OS to coordinate the operation of all layers.

- **Internal interlayer middleware** that is implemented within the node protocol stack, associated with a single protocol layer and acting in concert with other interlayer entities.

There is a common format to all cross-layer optimisation processes. This involves taking a set of parameter values from one or a subset of protocol layers and returning optimised parameter values to the same or other protocol layers. This commonality has enabled Khan et al [77] to define an overarching three-stage method of cross-layer optimisation, illustrated in figure 2.5.

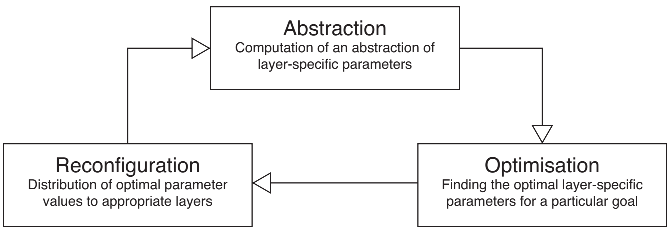


Figure 2.5: Stages of Cross-layer Interaction defined in [77]

The first, abstraction, stage is the most critical to the reduction of communication and processing overheads. It decides whether a reduced number of parameters are to be circulated, and underlying technologies veiled. Optimisation and reconfiguration then enable protocol adaptation to current network conditions and QoS requirements, in order to maximise network performance. This is through the tuning of the abstracted, or other related parameters that are then returned to the network stack. These three steps can be repeated according to changing QoS requirements and resource capabilities. However, cross-layer proposals have spanned a multitude of descriptive signal transfer methods and QoS control methods. While the former concentrate on how layers collaborate, the latter place more importance on which layers collaborate.

2.5.2 Cross-layer Signalling Methods

Cross-layer design aims to introduce a joined up approach to network management but in order to maintain the benefits of a layered architecture, the merging of all layers has generally been avoided. Cross layer signalling has had increasing interest in wireless networking and numerous proposals exist. The authors in [147] have presented a survey of the methods of signal propagation across cross-layer interfaces in static wireless networks. New designs have also abstracted the responsibility of cross-layer optimisation away from the protocol stack into kernel space middleware also known as a cross-layer management entity, plane or optimiser. This vertical entity avoids the use of direct interlayer coupling or communication. As many of these designs can be combined to create more complex or sophisticated cross-layering, those that would be relevant to providing QoS guarantees to ISRT applications in MANETs are evaluated here.

The methods of signal propagation proposed to date have included the use of packet headers and structures, external profiles and even network servers. In all of these approaches cross-layer parameters are abstracted from modified layers and accessed by other interested layers for the appropriate optimisation of their services. Prior to access and depending on the implementation, signals may be stored in memory on the mobile host, in a local hard disk in the node or in a third party server.

However, abstracted signalling information can only be of two types: locally or globally abstracted, though these are not always mutually exclusive. With the former, parameters are abstracted and optimised within a single node. The latter uses non-local, network-wide distribution of information from neighbouring nodes. The following section compares mechanisms that fall under these two categories and evaluates their performance in supporting RT applications in MANETs. These designs are discussed in detail and a representative list of examples is provided.

2.5.2.1 Network-Wide Signalling

The earliest approaches to cross-layer design have used packets for signal transmission. Packet headers are used by ECN [126], a network-wide cross-layer approach that is in commercial use. It is used to mark TCP packet headers, to provide an early notification of congestion. Increases in RTT can also be taken as an early implicit notification of congestion, in contrast to packet loss, that is a late notification.

ECN is a more reliable form of explicit notification than RTT, especially in MANETs where loss and delay can have numerous causes. Its use is appropriate for applications that are sensitive to packet loss. Unfortunately, ECN's hop-by-hop scheme requires notification to travel by increasing back-pressure through a congested network to reach the source. ECN is also not applicable with UDP, a protocol commonly used by RT applications. This is due to the requirement of application layer based congestion control, as well as API constraints on the appropriate header bits. The alternatives, Datagram Congestion Control Protocol (DCCP), Stream Control Transmission Protocol (SCTP) and TCP are highly inefficient in wireless networks [160], with connection oriented overheads, delayed congestion control and energy inefficiency.

Alim and Mohamed [9] implemented a leaky bucket algorithm with optimised burst parameters to regulate traffic at the network interface in combination with the token bucket algorithm to guarantee E2E deadlines for HRT flows. The second algorithm searches for the maximum number of tokens, permitting bursts, but only up to this threshold and therefore improving connection admission probability. Leaky bucket implements static allocations and this inflexibility is not generally appropriate for bursty RT flows. As a result the Token bucket algorithm is often used to provide dynamic control of the transmission rates of bursty flows through packet dropping, delay or marking of non-conforming packets.

Service differentiation through congestion control and traffic conditioning is proposed by Kim et al [81] where RT traffic is passed through a token bucket to provide a controlled traffic rate. Under network congestion RT traffic is prioritised over best effort traffic on the basis of backpressure notification to upstream nodes to concede bandwidth allocated to the latter in order to support QoS guarantees to the former. Use of the token bucket algorithm can lead to resource underutilisation as at a given point in time, the total token buckets and hence total bandwidth and buffer requirements of all flows can be greater than the network resource capability, when the actual resource utilisation is less than the token bucket total.

Packet structures have also been used in the detection and signalling of network conditions. Numerous approaches have used packet probing in the estimation of available bandwidth and E2E delay [5, 24, 48, 120, 151]. While accurate measurements of propagation delay cannot be obtained under heavy traffic loads, bandwidth estimation has been more successful. Ahn et al [5] used UDP control packets and Taleb et al [151] used low-priority dummy RTP packets, marked within the unused header bits to probe new network capability. Each intermediate node updated the packet if its available bandwidth was lower than that requested in the packet. The destination could then relay the minimum available bandwidth to the source. This approach required the modification of multiple nodes, to ensure recognition of the probes at end nodes. Dummy RTP packets were sent at a maximum streaming rate of the multimedia data for a fixed period of less than $1s$, to which the receiver responded with reception quality feedback in a Real Time Control Protocol (RTCP) packet. RTP uses timestamps and sequence numbers to provide timely delivery, relying on the RTCP to specify QoS requirements and synchronises streams.

Nodes may also utilise globally distributed MAC layer information to estimate resources in their localisation of the network. For example, CACP [180] uses discovery of neighbouring nodes' available resources, as a result of broadcast querying or carrier sensing of idle nodes. Queried nodes respond if they measure resources to be insufficient and the source will attempt to transmit again after a backoff period. CACP is difficult to implement in MANETs, due to the requirement of periodic querying along a static E2E path. Perceptive Admission Control (PAC) [27] therefore avoids the use of queries and estimates bandwidth according to channel utilisation, for distances up to which two flows can be transmitted simultaneously, without collision. Multipath Admission Control for MANETs (MACMAN) [100] avoids the flow throttling common in admission control schemes, that often result in resource underutilisation. It extends PAC with multipath routing: enabling senders to transmit on whichever path has sufficient resources. However, the methods of resource estimation used by CACP, PAC and MACMAN require frequent signalling, to ensure that transmissions are delayed or stopped if a threshold value is reached.

Other approaches have used passive monitoring in bandwidth estimation, such as that suggested by Vanhatup et al [157]. This begins with a node measuring utilisation and signal strength or throughput of neighbouring nodes, based on their periodic beacon signals. If a terminal receives a beacon from a neighbour, that neighbour is assumed to be in the collision domain of the terminal. This estimation assumes that the interference range is less than the carrier sense range but incorporates nodal distance into the throughput calculation. The available throughput of node AP_x is of inverse proportion to the total activity of AP_i nodes, which is in turn a value calculated relative to the channel activity of AP_x and weighted according to the channel distance to the other node. Ergin et al [48] express the aggregate link utilisation ρ_{aggr} in proportion to the number of nodes in contention range, N_{cont} in Equation 2.2.

$$\rho_{aggr} = \sum_{i=1}^{N_{cont}+1} R \cdot \left[\frac{L}{B_i} + T_{oh} \right] \tag{2.2}$$

Here R is the inverse of the packet generation frequency, L the length of the data packet, B is the transmitter link rate and T_{oh} is the link-rate/frame-size independent portion of the single-hop channel occupation duration. Such an implementation is subject to the exposed node problem [59], shown in figure 2.6, where node C is throttled if it detects a CTS from F to E, as F is on the boundary of C's carrier sense range. This overly conservative bandwidth throttling leads to unused opportunities for spatially and temporally parallel transmissions.

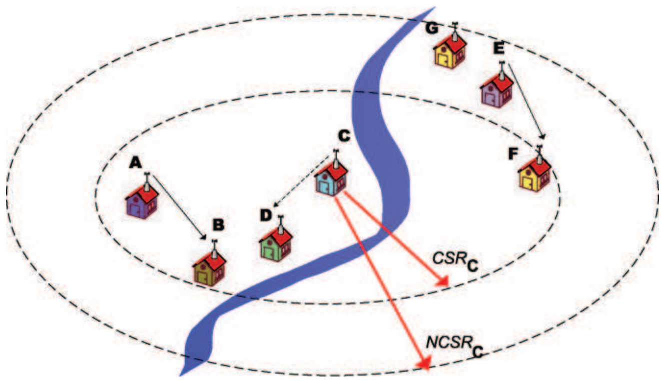


Figure 2.6: Self-interfering Flows [48]

Dual Carrier Sensing with Parallel Transmission awareness (DSCPT) and Packet Probing with RTS/CTS Handshake (PPRCH) [48] have been proposed to support parallel transmissions. DCSPT is an extension of PAC and CACP that relies on dynamic adjustment between higher ($NCSR_C$) and lower (CSR_C) carrier sense thresholds. As indicated in figure 2.6, DSCPT use allows C to transmit to D, while considering the E to F transmission in its available bandwidth estimation. This approach gives throughput gains of up to 80%. However, in order to extend the carrier sense range of nodes, DCSPT assumes range modification capability in radio hardware. PPRCH was therefore suggested as this is not possible in most existing devices.

PPRCH utilises handshaking of only probe packets in order to avoid the hidden node problem. Two probe packets are sent out at highest and lowest priority respectively, so that the second does not compete with data packets and the dispersion of the probes is used to give the available bandwidth. However, this process utilises repeat probe packets with adjusted delays between pairs, in order to find a probe datarate that is equal to the available bandwidth. The problem is that this heuristic approach may take several RTTs to reach the correct measurement, which is not ideal for delay sensitive flows.

Packet probes have also been used in Ticket Based Probing (TBP), a QoS routing protocol. TBP uses tickets to narrow down possible paths and established routes in light of E2E delay requirements [35]. When a source needs a QoS path to a receiver, probes holding tickets are used for route discovery. Each probe accumulates the path delay, which is updated by intermediate nodes on the path. The total number of tickets available limits the possible paths probed. When a probe with N tickets arrives at a mobile node, the node will split the probe into $\frac{N}{x}$ probes, based on local state information, with a subset of the tickets. These are then forwarded onto x destinations. If delay restrictions are exceeded the remaining tickets are invalidated. Therefore, if multiple valid probes arrive at the destination the path with the least cost is selected and other paths kept as backup. A confirmation is then relayed to the source, using the list of mobile nodes along the path that is carried in the probe. Processing overheads are low for this mechanism. However, the accumulated link delay resulting from the use of probes can impact on performance in highly loaded networks. In addition, each node must store state information for each neighbour, entailing increased memory requirements for more complex network topologies [18].

As MANET nodes commonly have limited memory capacity, storage of signalling information in external servers can improve the performance of a cross-layer approach. The Wireless Channel Information (WCI) [80] network service and CrossTalk [165] externally store signals, giving accessibility to the rest of the network. WCI abstracts neighbours' data link and physical parameters, forwarding these via a proxy server. In contrast, CrossTalk [165], shown in figure 2.8, combines local with network-wide signalling. Each modified node coordinates parameter exchange in the local stack, which is then aggregated to a database of global network conditions. Jiang et al [73] proposed an architecture for IP-based CDMA single-hop networks using a centralised cross-layer scheduler at a base station that would exchange QoS provisioning parameters with mobile nodes. Under this model, multimedia frames would be compressed into packet batches according to priority in the form of the batch class and size communicated to the scheduler by the node. The scheduler would also optimise backoff values in light of the maximum delay timeout and channel state parameters that could also be optimised. These parameters include the threshold at which a channel is considered good or bad, the good/bad threshold, F, defined according to channel quality and used by nodes that transmit when channel gain is FdB less than the average.

Lin et al [99] implemented a decentralized scheduler to simultaneously schedule multiple wireless links with the benefit of being able to minimise multi-channel interference. However, similar entities have not been implemented widely in ad hoc networks due to their high connectivity requirements. In using network-wide abstraction, processing overheads impact on E2E delay. Packet delay is also increased by the requirement for processing of cross-layer parameters in intermediate nodes [128,135], reducing the capability for RT support. The authors in [130] have also suggested a novel architecture that maintains global network status information and uses this to select and modify protocol parameters on a network-wide basis. The architecture is yet to be validated, but proposes the use of CTS packets to piggyback channel quality estimates from the receiver, triggering optimisation along the path to the sender.

2.5.2.2 Local Node Signalling

A local approach abstracts cross-layer parameters from modified layers, storing them for access by other interested layers, for the appropriate optimisation of their services. In contrast with network-wide designs, these parameters can be associated with a particular packet or flow. As previously discussed, the earliest methods of local signalling used packet headers or structures for the encapsulation of parameters, providing accessibility to subsequent layers along the processing path. Packet headers are used by the Interlayer Signalling Pipe (ISP) [168], that modifies the IPv6 Wireless Extension Header (WEH) for in-band parameter propagation. The ISP does not require any add-on messaging protocols, as interested nodes can access signals if WEH aware. However, processing overheads increase when successive layers are required to access the network layer header. As packet based signal transfer is continual, the signalling benefit does not compensate for the resultant long-term increase in propagation delay.

Alternatively, signalling data can be inserted into a section of the packet structure, allocated to each layer, on transmission or receipt. Only layers that implement the cross-layer modification can edit and access their corresponding packet structure segments, while non-participating layers need not perform any of these tasks. Similarly to ISP, the use of data packet structures limits the exchange of information to neighbouring layers in the direction of packet flow. In contrast, local out-of-band signalling shortcuts, using dedicated API, enable direct interlayer signalling between non-neighbouring layers. Both Internet Control Message Protocol (ICMP) and Real-Time Transport Control Protocol (RTCP), out-of-band control protocols, can be used for notification up the stack [110,148,151,168]. IP uses ICMP messages for the same purposes as RTP uses RTCP: for the transfer of control information. These control messages are generated at any layer when convenient. For every parameter change, beyond a predefined threshold, a new control message must be generated, greatly increasing the competition with data packets for bandwidth.

ICMP messages must be encapsulated in IP packets, and RTCP in UDP packets. As a result, all signals are forced to pass the network layer, if ICMP is used, and transport layer, if RTCP is. This is even if the sending and receiving layers are not divided by those layers. CLASS [163] allows bidirectional message exchange between non-neighbouring layers. Signalling is higher speed, without the need to bypass adjacent layers. CLASS uses ICMP for general messaging and TCP/IP headers for shorter notifications. These out-of-band methods lose the benefit of signal association with a particular packet. They are not ideal for MANETS as they rely on control packet generation capability in all intermediate nodes, and are inflexibly limited to request-response procedures [82]. The result of the latter is increased network load, due to the large number of control packets on the network. Increased processing delay is also imposed by the heavy headers and checksum requirements as well as the storage of parameters in memory or a local hard disk in the node. Even if a node did have sufficient memory capacity, this local storage would not provide the direct access necessary in high-speed RT networks.

To avoid the drawbacks associated with external storage implementations, local profiles as well as functionality to manage their use, have been introduced to the field of cross-layer optimisation. These profiles are also known as parameter databases, cross-layer servers or planes. El Defrawy et al [43] have developed the Cross-layer Server, to provide signal accessibility to all layers. Clients are used to communicate with layer protocols, requesting, optimising and controlling internal parameters. Similarly, the Central Cross-layer Plane [33] uses local profiles of abstracted parameters, stored by an XML based mechanism, that are created on a per-layer basis for access by interested layers. The use of Callback functions by these profiles enables event based signalling: allowing one protocol layer to register these functions with a second layer for execution on a specific event's occurrence at the second layer. The benefit of an event-based method is that signalling traffic is reduced by at least half, as the need to request parameters is removed. Callback functions are defined and installed by a protocol, registered to the library at one point in time and only invoked when a parameter reaches a certain threshold value.

These functions have been used to enable protocols to transparently access information in a related data repository, as used in the MobileMAN [40]. MobileMAN has been implemented in an experimental testbed. However, to reduce protocol modifications, callback functions can further be extended to contain instructions for encoding asynchronous private protocol data into a related abstraction with a local middleware entity. Such a local middleware entity can be made responsible for the abstraction of signal parameters, as well as the coordination of their usage. This entails a weakly cross-layered solution [82] that ensures protocol reusability but that can reap the performance benefits of strong cross-layer interaction by tuning parameters at multiple layers. The local middleware entities proposed to date, can be divided into two classes, as defined by Foukalas et al [52]:

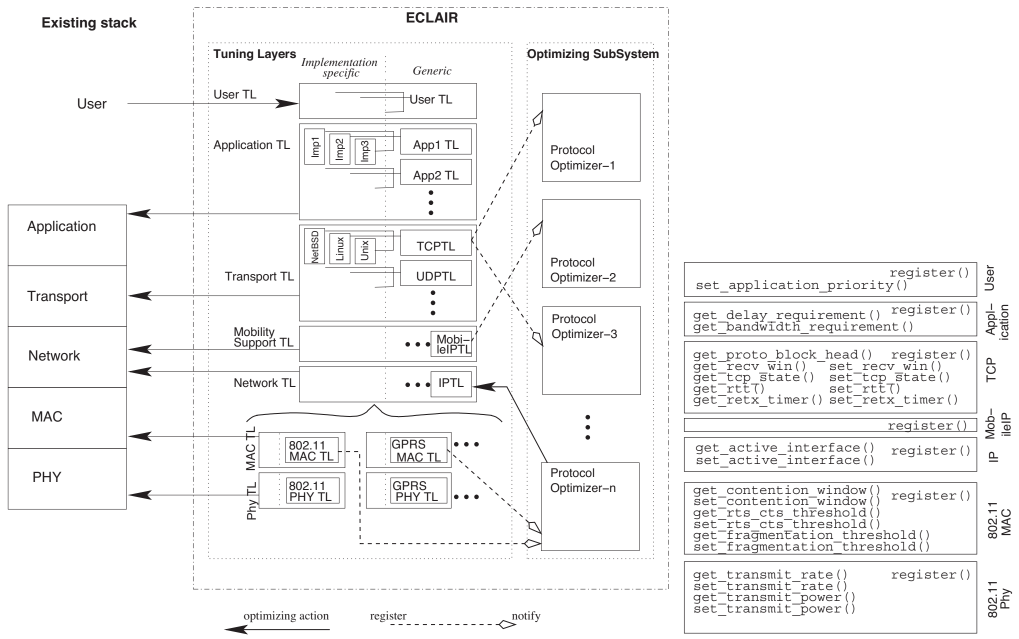
- **Internal interlayer entity:** associated with an individual protocol layer and acting in concert with other interlayer entities, associated with other layers.

- **Internal intralayer entity:** sitting between the application layer and the Operating System to coordinate signal propagation at and between all layers.

Carneiro et al [25] have used interlayer entities that are aware of the state of each protocol layer at any time, through notifications of layer specific events. These cross-layer entities, or coordination managers, abstract parameters from protocol layers, thereby enabling interaction between heterogeneous technologies. Calculations are then made, based on comparison between a minimal number of parameters shared, to identify the optimised values for a particular function. For example the average Perceived Signal to Noise Ratio (PSNR) between the video stream at input and output, as well as the rate distortion factor can be calculated. These are then distributed to protocol layers, for comparison and amendment of their own related parameters [77]. Communication overhead is incurred in the transfer of parameters to and between middleware optimisers and packet delay is increased. This is due to distributed calculations and protocol layer reconfigurations incurring high processing overheads.

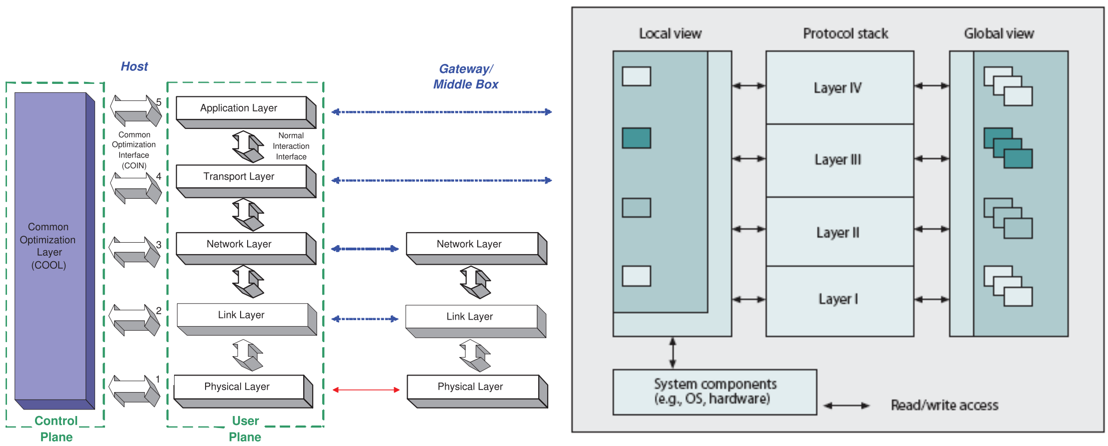
When optimisation is centralised in intralayer middleware, loops and conflicts between layers and processing and communication overheads of a distributed approach are avoided. A centralised middleware optimiser has been used in the Interaction Control Middleware Plane [30]. This plane coordinates multiple optimisers operating simultaneously in different protocol layers. Intralayer middleware controls the multiple interlayer entities. The Interaction Control Middleware Plane uniquely uses in-band signal propagation, which limits the exchange of information to neighbouring layers in the direction of packet flow. It is therefore subject to the same drawbacks as its earlier in-band counterpart, the ISP.

Two conceptual architectures, ECLAIR [122–125] and Performance-Oriented Model [58] (POEM) are key models as they introduce interface access to protocol parameters, without protocol modification. Though neither has been performance tested, they can advise the development of middleware for ISRT in MANETs. POEM has been conceptualised to use an internal intralayer entity that does not compromise normal protocol layer functionality. POEM is made up of two conceptual planes, or optimisers, as in figure 2.8. The first permits normal non-optimised data flows and the second optimises interactions. An common interface between multiple protocol layers and an Optimisation Layer provide self-optimising services through a control protocol, the Common Optimisation Protocol (COP). POEM



(a) Architecture (b) Tuning Layer APIs

Figure 2.7: The ECLAIR Architecture and APIs [124]



(a) POEM Model (b) CrossTalk Architecture

Figure 2.8: POEM [58] and CrossTalk [165]

aims to execute concurrently with normal protocol interactions, without modification of the protocols.

ECLAIR is also a theoretical, but partly validated architecture (figure 2.7) that aims to use internal intralayer middleware: the Optimizing Sub-System (OSS). ECLAIR introduces cross-layer interfaces, alongside the OSS, acting as tuning layers to support and control communication between the plane and layers. The tuning layers can manipulate protocol data structures at generic or operating system specific levels. The proof of concept for ECLAIR has shown that the TCP modifications and cross-layer approach are able to be implemented, however they have not been validated together as a single approach. Protocols themselves are not modified in ECLAIR, but API functions are exported to layers allowing read and write access to protocol control and data structures. Some examples of the APIs are presented in figure 2.7.

Protocol Optimisers then use these APIs to manipulate protocol runtime behaviour. This optimisation is intelligently based on input from other layers and devices. The individual ECLAIR optimisers are protocol code independent. Function call use does incur communication overheads but the overall processing overhead in the stack is negligible as the optimiser executes at the same time as the stack. ECLAIR has been used to calculate and tune receive windows and hence application bandwidth, according to allocated application priorities. A similar entity was also used by Kwon et al [90] in an OFDMA network: a MAC scheduler and resource controller together increased achievable throughput according to Channel Quality Information from the physical layer. This showed the same benefit as ECLAIR of reduced processing delay. XIAN [6, 7] is the first full linux kernel and testbed implementation of an optimiser that provides access to MAC and physical layer parameters. The architecture has been extended to include multiple, interconnected kernel- and user-space API and a middleware consisting of a range of parameter storage and management components. The authors demonstrated the capability of XIAN to broadcast ETX information to neighbouring nodes in order to improve routing decisions. However, specific optimisers have not been implemented and tested in the architecture and the validity of link metrics such as ETX is influenced by node mobility, link breakages [121] and the underlying process of the MAC layer, such as repeated backoff. Their usage then introduces complexity in path selection and performance.

2.5.3 Adaptive QoS Control through Protocol Tuning

The ability of cross-layer middleware to improve network performance depends on its functionality: at a high level, which QoS-control measures are to be optimised and specifically which values in protocol data structures will be accessed and modified. The challenge lies, more specifically, in the appropriate selection of protocol parameters that can signal relevant information to other layers or be dynamically tuned to enable improved application performance. Signalling delay and, proportionately, packet delay are also dependent on the number of parameters abstracted and optimised. Monitoring and optimising a large number of parameters can unduly increase both processing and signalling overheads and it follows that limiting the number of parameters abstracted is an aim of RT support. Additionally, the performance of a cross-layer design is dependent on the selection of QoS control measures and appropriate parameters, in line with the optimisation goal. Establishing which should be optimised is not straightforward as there are indirect as well as direct linkages between protocol parameters. Khan et al [77] therefore defined the four key categories of parameters that can be abstracted by a cross-layer model as:

- **Directly tunable parameters:** that can be reconfigured by cross-layer optimisation, e.g. time slot assignment in TDMA.

- **Indirectly tunable parameters:** that cannot be themselves reconfigured but change as a result of reconfiguration of a directly tunable parameter, e.g. BER that depends on the coding and modulation scheme used.

- **Descriptive parameters:** that can be read but not tuned, e.g. channel quality estimates or picture size in video streams.

- **Abstracted parameters:** that are computed from the two types of tunable parameter but do not actually occur within the protocol stack, e.g. net transmission rate.

Many cross-layer models have been proposed in the fields of wireless SRT performance and NRT performance in MANETs. Each of these existing models has been holistically aimed at a specific performance target, such as improved TCP congestion control, delay or jitter reduction or fair sharing of available bandwidth. Research in the field of cross-layer QoS guarantees to RT traffic in MANETs has been limited particularly to the support of high-quality multimedia. In order to meet the goal of guaranteed performance to ISRT in MANETs, the commonalities and learning points from the two aforementioned fields are of particular interest.

Cross-layer approaches for MANETs that access the MAC layer, predominantly deal with the bursty and congestion-prone nature of MANET traffic and corresponding packet losses. Rather than implementing congestion control, congestion may be avoided through the implementation of appropriate load control. Collaboration between devices is required to ensure that each flow satisfies its bandwidth requirements and does not surpass its allocation. Although the admission control process does not guarantee QoS, timing guarantees to applications are dependent on the path discovery phase of routing. This phase is, in turn, highly reliant on the stipulation of QoS requirements during admission control. The efficiency of distributed admission control depends on the accurate estimation of available resources, which is traditionally performed at the transport layer.

Models for wireless SRT have been particularly directed at the the need to guarantee and limit packet delay and loss. Common packet loss rates in wireless networks are between 10^{-3} and 10^{-1}. Compounded by the high bit-rate requirements of RT streams, when a number of streams contend for limited available bandwidth, packet loss can be increased by congestion and interference-related bit errors. Congestion occurs when data packets wait for service at a resource bottleneck, due to traffic flows exceeding the available capacity of buffers over the data path. As buffer occupancy increases, as does queueing delay experienced by neighbouring flows. When buffer resources are exhausted, uncontrolled packet dropping can result in network congestion collapse, with worst-case E2E delay and repeated, escalating packet loss.

Three types of parameter optimisation can be implemented in a cross-layer model: network-adaptive or QoS-adaptive and what will be referred to as hybrid adaptive. In the first, higher layer protocols are tuned in light of variation in network resource conditions. The second is instead a top-down approach whereby lower layers are tuned to meet application-specified QoS requirements. Hybrid approaches combine both of these types of adaptation, although, such a functionally complex approach will often be less scalable.

2.5.3.1 Network-adaptive Tuning

Network adaptive tuning is used to increase protocol responsiveness to network contention and congestion as well as wireless link problems of fading and interference. In wireless networks, congestion control can be implemented to detect and prevent congestion. However, congestion avoidance, if early notification is available, provides better network and RT performance. As previously discussed, both congestion control and notification are traditionally provided by TCP, a protocol that performs poorly in wireless networks and RT support [116, 160]. RT applications predominantly use UDP or RTP at the transport layer, but still have a congestion avoidance requirement.

With synchronous contention-based schemes, collisions can occur but mechanisms for avoidance are implemented. For example, the use of control packet messaging, alone or in combination with carrier sensing. Multiple Access with Collision Avoidance for Wireless (MACAW) [20] employs a five-part control packet exchange, RTS-CTS-DS-DATA-ACK that exhibits high throughput and fast error recovery through use of the ACK. However, these extended control packet dialogues introduce a large amount of competition for resources and do not fully solve the hidden node problem. Control packets may still collide, leading also to eventual data packet collision. Some schemes that combine control packet messaging and carrier sensing utilise service differentiation [108, 159] or reservation [97] while others implement fair scheduling [155].

MACA with Piggyback Reservation (MACA/PR) [97] is a cross-layer QoS-routing scheme with collision avoidance. It is designed for RT traffic and relies on MAC layer bandwidth reservation. Under this scheme the first data packet of a flow is used for MAC level reservation along the path. Source initiated control packet messaging is used to set up a reservation, with data packet transmission immediately following the receiver CTS and containing a piggybacked reservation for the subsequent packet. Data packet receipt is followed up by an ACK response, used solely to refresh the reservation. The protocol does not initiate loss recovery. Neighbouring nodes sensing data and ACK packets use these to maintain a reservation table of transmit and receive windows, for nodes in signalling range to backoff accordingly. If ACK receipt exceeds a timeout at the source, the channel is assumed to be insufficient for bandwidth requirements. The QoS routing protocol at the network layer is signalled accordingly.

The collision avoidance in MACA/PR achieves lower E2E delays than synchronous schemes but also lower aggregate throughput. Reduced throughput results from the communication overheads incurred with reservation table update and exchange and the requirement of the source node to consult these prior to transmission [88]. Collision avoidance alone cannot replace congestion avoidance. At the same time the unmodified carrier sense capability of radios is still important in ensuring that nodes within carrier range gain timely and reasonable access to good quality channels.

Transmissions between two nodes consume bandwidth from all neighbouring nodes. The inference of this is that contention-aware load control must be implemented in order to avoid over-subscription of resources. Admission control does not explicitly use bandwidth reservation but many approaches have suggested the joint allocation of capacity and flow. This is via the exchange of link capacity and flow requirements between network and link layers. It can result in greater link utilisation and reduced congestion. Lower layer signalling has been proposed to adjust transmission rates to individual link capacity and combined with resource reservation increased supported data rates [139–141,186]. However, resource reservation relies on a large amount of control information being passed between and maintained by nodes, and as such is not applicable in MANETs.

Supported datarates have been shown to increase when cross-layer signalling is employed [140] at the link layer to adjust transmission rates according to the capacity of individual links. One such approach is INSIGNIA [95] that encapsulates QoS signalling information: service indicator, payload type, bandwidth indicator and bandwidth request fields in the options field of the IP header. These control signals are then used for resource reservation as well as restoration and adaptation. The scheme operates solely at the network layer and can therefore be combined with most MAC layer mechanisms. Reservation requests are sent to the destination and used by mobile nodes to carry out admission control the result of which is added to the header. If a flow is accepted a soft-state table is updated in order to schedule the rest of the flow otherwise only best-effort services are provided. Resource allocation is lost when a flow becomes inactive, this is judged on the basis of a timer and the last update to the soft-state table. INSIGNIA has good mobility support as signal loss provokes a restoration mode that manages rerouting followed by admission control and resource reservation on a new link. The receiver also sends regular QoS reports to the sender in order for it to adapt its transmissions.

Other models such as Flexible QoS Model for MANET (FQMM) [171] have combined per-flow reservations for high priority traffic with service differentiation and per-class guarantees for all other traffic. FQMM uses hybrid provisioning to allocate resources according to priority classes of traffic and traffic conditioning at the source to ensure conformance to the traffic profile. But FQMM is not appropriate for lossy MANET links. In contrast some solutions work at both the network and link layers, several only use link layer information for routing while others develop a joined up approach.

SWAN [5] is a stateless mechanism that adds AIMD rate control to INSIGNIA, providing traffic classification and servicing with different priorities without requiring per-flow information or signalling. Service differentiation with this wireless network model provides traffic classification and servicing with different priorities, without requiring per-flow signalling. Rate control is automatically configured based on MAC delay, and available bandwidth is estimated according to neighbouring flow rates. However, RT sessions are stopped, not throttled, under congestion so high overheads would be incurred with its use in highly dynamic networks requiring regular session re-establishment.

Scheduling also plays a key role in the support of jitter and delay sensitive wireless applications, given its impact on packet deadline achievement. When packets are dequeued for transmission it is scheduling algorithms that make the decision of which packet is the head of line, based on priority, delay requirements, nodal congestion and other QoS requirements. Queues exist at multiple protocol layers inside a node and must be serviced appropriately in order to ensure maximum and fair resource utilisation and that QoS guarantees are upheld. Wired network schedulers rely on traffic and queueing statuses, as a basis for prioritising packets, but in wireless networks with varying channel-capacities this is insufficient.

The network layer can also select from available interfaces according to application requirements and capabilities of the client nodes and the connector state as one channel may provide lower delays or higher throughput than another. Bellavista et al [19] suggest a double-layered selection of interface firstly discarding unsuitable connectors based on reliability, RSSI or BER of the channel and relative mobility and requirements of the node, to provide a list of suitable connectors from which a channel is selected according to application-specific requirements. In the architecture introduced by Taleb et al [151] the fact that the physical and data link monitor signal strength is exploited in order to anticipate impending handoff and notify the application layer to locate a new AP. While packets continue to be transmitted over the old path, information on the new AP is transferred to the TCP or RTP at the transport layer that probes the new network capability using low-priority dummy packets.

RTP is predominantly used in the Internet for SRT multimedia streaming but does not employ congestion control; instead RTCP enables receivers to communicate perceived QoS to senders via Receiver Reports (RR). However, RTCP is used stringently occupying only 5% of session bandwidth therefore RR transmission is too slow to efficiently notify MANET senders of handoff events. This cross-layer scheme and the use of RTCP Handoff Notification packets is necessary for the notification of senders and receivers respectively of impending handoff. After a defined period of dummy packet transmission the receiver transmits a RTCP Handoff Report with reception quality feedback over the new link. The resulting packet loss ratio is significantly lower as a result of fast throughput adjustment; RTP alone still achieves a higher throughput than this scheme but at the expense of considerable packet drops.

Wireless packet scheduling is utilised to ensure efficient link-utilisation; provision of delay bound guarantees, smooth service degradation and protection from non-conforming sessions. The fair redistribution of resources across sessions and guaranteed short-term and long-term throughput [50] may also be provided. Policies that dynamically adjust priority based on deadline information, such as Minimum Laxity Threshold (MLT) and Queue Length Threshold (QLT) have also been shown to provide good RT performance when dealing with bursty RT traffic [91]. MLT only prioritises RT packets when the time until their deadline (minimum laxity) is less than a threshold value while QLT prioritises NRT traffic only when the length of the NRT queue exceeds a threshold. Distributed Fair Scheduling (DFS) [155] has been proposed to ensure that flows are allocated bandwidth according to their weights or priorities. Each packet is assigned a start and finish time stamp, with higher priority packets given earlier finish times and shorter backoff periods resulting in higher throughput guarantees. Timestamps are calculated according to the Self-Clocked Fair Queueing (SCFQ) algorithm that requires each node to maintain a virtual clock in order for the timestamps to be calculated in proportion to the average reserved throughput for the flow.

In combination with exponential mapping of backoff intervals DFS has been shown to achieve high throughput but DFS alone does not meet the delay requirements of RT packets or overcome the hidden terminal problem [88, 129]. Proportional Differentiated Services / Proportional Delay Differentiation (PDS/PDD) provide differentiation of services in terms of queueing delay to different classes of traffic and the ratio of average queueing delays is controlled between classes. These scheduling algorithms may select the next queue for servicing according to the normalised average delay, normalised head of line waiting time or both depending on whether Proportional Average Delay (PAD), Waiting Time Priority (WTP) or Hybrid Proportional Delay (HPD) is implemented. However, as these decisions are based on relative rather than absolute delay requirements there is no way of providing delay bounds or throughput guarantees to a class of traffic.

A number of cross-layer designs incorporate dynamic schedulers at the transport or MAC layers and occasionally, when optimising video transmissions, at the application layer. Kwon et al [90] used a MAC scheduler and resource controller that together increased achievable throughput according to Channel Quality Information from the physical layer. However, this required a Hybrid ARQ (HARQ) scheme, to support the selection of modulation and coding and improve throughput guarantees, at the expense of high retransmission overheads.

Several scheduling approaches have considered optimising the source bit rate according to channel conditions in order to minimise congestion and delay [3, 77, 112, 140, 184]. If the transmission rate selected is lower than the optimal transmission rate along a path, a large amount of jitter is introduced into the stream. If it is higher, packets will be dropped at intermediate nodes and the receiver. Additionally, the validity of E2E feedback rapidly degrades when node mobility and channel quality create dynamic variance between multiple paths, which has a greater impact on performance when constant tuning of application rates is implemented. This adaptation is generally only useful to a select group of elastic SRT applications including media streaming, video conferencing and interactive network gaming, but has shown an increase in video quality of between 0.63dB [184] and 2dB [77]. In order to reduce distortion in video streaming a select set of parameters may be optimised. The application layer participates, as it contributes parameters relating to per-packet loss distortion effects. It can also adapt the source rate according to network capability information provided by the data link or physical layer.

Khan et al [77] utilised physical and data link layers to estimate transmission capabilities and adapt the time slot allocation and modulation scheme accordingly.

The expected receiver PSNR metric is utilised to quantify QoS achievement at the application layer in terms maximising the average PSNR of all users. Four parameters are abstracted at the radio link layer, these are transmission rate, transmission packet error rate, packet size and channel coherence time; the latter influenced by user velocity and the interference environment. These parameters are then used to compute probability of transition from a good to bad state, p, or bad to good state, q, for each user and to retransmit prioritised packets if bandwidth allocation is higher than the source rate. When distortion information was transmitted alongside Groups Of Pictures (GOPs) the framework offered an increased PSNR of 2dB. Chan and Modestino [29] also suggested the exchange of parameters between source and channel coding in the application and physical layers for optimisation.

Optimised application level scheduling can reduce the number of quality drops seen in wireless transmissions when packets are absent at the decoder, predominantly caused by broken links. Navaratnam et al [112] proposed the Link Adaptive Transport Protocol (LATP) for multimedia streaming with source rate adaptation based on receiver feedback. The sending rate was initially transmitted in the IP header options and updated at each intermediate node if the maximum permissible transmission rate was less than this value. The permissible sending rate was calculated based on measured channel utilisation and MAC layer feedback at each node and aggregated to the maximum transmission rate for the path when arriving at the receiver. In multipath routing this approach would require cooperation with the routing algorithm in order to ensure that the correct rate adaptation were implemented and in ad hoc networking E2E rate feedback may not represent the current conditions by the time of arrival at the source.

2.5.3.2 QoS-adaptive Tuning

QoS-adaptive Tuning of lower layers according to application requirements is a more common proposal for MANETs, with reliance on the MAC layer for bandwidth estimation and formalised protocols for QoS adaptive routing [88]. Many load control approaches for wireless RT do not directly address the effect of MAC layer contention or neighbour interference on bandwidth estimation. Therefore some QoS-adaptive models have moved to take a MAC layer approach to bandwidth reservation, combining this with QoS routing to support wireless SRT. It has been suggested [3, 140] that the MAC layer should perform dynamic capacity assignment, determining resource allocations for different flows that have undergone congestion-optimised multipath routing at the network layer. However, this requires all participating nodes to implement the same MAC layer or employ a bridging device between heterogeneous devices, which may introduce certain security issues.

QoS Protocol for Ad hoc Real-time Traffic (QPART) [179] consists of two components that span the network and MAC layers: a QoS-aware scheduler and a QoS Manager. The scheduler relies on modification of the MAC layer and use of the DCF RTS/CTS mechanism. QPART moves contention window calculation to the network layer, that can access per-flow delay requirements and returns a per-flow backoff period value to the MAC layer. The manager is then responsible for admission control, reliant on prioritisation to reject flows under congestion, which is detected solely based on reduction in channel idle time. While the requirement to modify the MAC layer limits the scalability of QPART, repetitive tuning of MAC layer backoff values can introduce artificial delay into flows.

Synchronous MAC schemes such as TDMA have been traditionally used in wired networks, these aim to provide determinism by dividing the channel into time slots each allocated to a flow during route discovery in order to meet bandwidth requirements. Each node can then transmit within the allocated time slot while other nodes sharing the medium must wait for their allocation. Much like Cluster Gateway Switch Routing (CGSR) [45], Cluster TDMA elects a cluster-head to manage slot allocation to nodes within the cluster [55]. Synchronisation is therefore required between nodes though Cluster TDMA reduces this requirement to the cluster heads. The problems with schemes that require synchronisation are twofold. Synchronisation accuracy depends on regular communication sessions between nodes that introduce high overheads [108] and compete with data packets for resources [49]. This high frequency of communication can be difficult to implement in large multi-hop networks or ad hoc networks where links are setup and torn down dynamically.

In wireless ad hoc networking, contention within carrier sense range of a node must be considered in the bandwidth estimation before load control is performed, given the dynamic topology. When admitting a flow, the transport layer cannot provide an accurate estimate of current network conditions but lower layers can [65]. Yuhe and Jie [182] suggested the joint control of the physical and MAC layers for the estimation and prediction of channel variation based on packet error rates (PER). Modified RTS packets carry required PERs and datarates as well as training bits used by cross-layer middleware in estimation and prediction. The MAC layer can then access physical layer parameters including available transmission rate that, combined with the SINR, can be used to improve scheduling decisions. As a result, higher rate transmissions can be prioritised on links of degrading quality.

The MAC layer continually monitors instantaneous signal strength (ISS) changes, and can provide better information on available resources in a MANET, where link performance is particularly dependent on SINR. A tunable MAC protocol, Congestion Reducing Medium Access Control (CRMAC), was proposed by Bag and Bassiouni [13], which could be adapted to the requirements of the application, on the basis of buffer status data from the network layer. A combination of the recent collision history and number of collisions for that node was then used to calculate an appropriate backoff value, prioritising congested nodes above others in the same collision domain. Although RT delay guarantees cannot be provided with a random backoff, the calculation utilises the useful concept of a weighted collision history to calculate the probability of a collision occurring. This is expressed through the sum of the collisions since last transmission α and the collision history β given by Equation 2.3. The calculation of collision history then utilises the constant μ to ensure that earlier collisions are weighted more than recent ones.

$$\beta = \mu\beta + (1 - \mu)\alpha \tag{2.3}$$

CDAC [113] similarly combines the percentage of slots that are idle, successful or contain collisions with the transmission probability to calculate the probability of the next slot containing a collision. However, the equations proposed are non-linear and contain multiple unknown variables making them difficult to solve and also assume flows with constant transmission probabilities.

IETF OSPF-MANET routing is one of the only commercially used cross-layer designs [115] that includes a MANET-specific cross-layer interface for signalling from the data link layer to the network layer. This implementation reduces packet loss, resulting from signal loss, by 60%. The cross-layer interface tracks incoming frames and then receiving-link quality is assessed for use by the routing protocol. This enables a distinction to be made between physical link failure and congestion, for signalling to upper layers. This scheme enables the assignment of higher priorities to higher link qualities, reducing the rerouting delay that results from link failure. OSPF-MANET also relies on flooding and hop-by-hop acknowledgement and exploits the broadcast efficiency of the underlying radios. However, a cross-layer processing overhead is incurred for each signal, as the routing protocol must use an address mapping function to map the MAC address to an interface IP.

QoS-adaptive scheduling has been implemented in wireless networks for RT application support. Differentiated-Time Urgency Based Algorithm (D-TUBA) [153] utilises a cross-layer scheduler to schedule packets according to class and global delay information abstracted from the network layer in participating nodes. A modified Weighted Deficit Round Robin (WDRR) is then used to decide to where packets are de-queued. WDRR was modified using a counter flag to indicate whether a queue remainder was larger than the next packet so that at most one packet would be serviced whenever a queue was polled. This scheme adopts IP packet header signalling. The next packet is enqueued according to the estimated remaining time of the packet (the delay bound less the estimated time to destination). Using a scheduler to estimate time to destination in lieu of more precise measurement avoids the introduction of large communication overheads, but this assumes that the traffic en route is of a uniform distribution.

An and Song [11] have also developed a priority-based wireless scheme, with routing and MAC scheduling working to meet RT delay requirements. The concept of packet urgency is used to give greater priority to packets where the accumulated delay to required maximum delay ratio is larger. With packet priority at one node dependent on the priorities of packets at other nodes, implementation of this network-wide approach in a MANET would entail high bandwidth consumption and increased delay overheads.

2.5.3.3 Hybrid Network and QoS-adaptive Tuning

A few cross-layer approaches have adopted hybrid adaptation, with parameters shared from and tuned at multiple layers in order to provide QoS guarantees without overloading the network. For example the congestion minimisation scheme proposed by Setton et al [140] supports the highest datarates and yields minimum E2E delay, by guaranteeing a given datarate between RT transmitter-receiver pairs. This requires the MAC and network layers to identify the set of network flows that minimize congestion, through the iterative exchange of possible suboptimal solutions. However, such an heuristic approach requires extensive signalling and would not be ideal for transmissions with critical timing requirements operating over a MANET.

The authors in [142] have investigated the effect of jointly tuning application layer packet size, physical modulation and MAC retry limits with reference to received multimedia performance. This has been done offline, without implementation of a particular signalling method. Over a single-hop with low SINR, lowering modulation and increasing the retry limit reduced delay, but at the expense of greatly lowered throughput. Lowering the packet size according to reduction in channel conditions was also suggested in order to provide minimum goodput and delay guarantees.

Chen et al [34] implement congestion control, routing and distributed scheduling based on backpressure notification of the congestion price at neighbouring nodes and corresponding adjustment of capacity allocation. Congestion price is calculated at the source node according to queue length information that is periodically broadcast or transmitted in response to a query packet. Each node also broadcasts a pilot signal to neighbours upon receipt of which the local SNR is calculated and returned for use in estimation of the current channel conditions. Utilising only local channel state and queue length information ensures that the complexity of the distributed scheduling algorithm is reduced.

The QPS scheduling algorithm [183] continually evaluates physical layer channel quality and MAC layer packet delay to regulate throughput. QPS estimates the probably delay and calculates the cost of transmitting the packet and the required delay. The authors propose packet queueing according to a cost function incorporating a weight parameter; the normalised average delay rate, to indicate delay satisfaction, and the number of excessively delayed packets. The first packet in the queue is therefore the packet with the least cost and a timeout is implemented to drop packets whose waiting time exceeds requirements as a result of poor channel conditions.

Congestion-aware physical rate selection and allocation has been suggested in [185] and [105]. [185] uses network-wide updates of video source rate and link congestion price in local rate allocation. Loiacono et al [105] suggest that such approaches fail in attributing packet loss ratio to channel conditions rather than collisions. Instead they propose consideration of the application codec type, collision probability and physical channel conditions to estimate received video quality. Tuning physical rate according to this estimate results in increased throughput of up to 2.4Mbps.

Liu et al [103] proposed a RT scheduler utilising a per-connection priority function that is updated dynamically according to wireless channel quality as well as QoS requirements. The scheme offers the highest priority and guaranteed QoS to CBR connections, such as VoIP, and a lower priority with some packet loss to RT traffic that can tolerate it. Generally the larger the delay satisfaction the lower the priority, but if it drops below a threshold packets are sent immediately. The use of Channel Quality Information (CQI) ensures that, within a single class, a large normalised received SNR translates to a higher priority, but that channels experiencing severe fading are not serviced at all. However, even when channel quality is low, if the delay satisfaction is low, the connection will still be serviced with a relatively high priority.

The model developed by Chen et al [32] controls packet loss rate, resulting from link errors, using local channel conditions to determine transmission power level and media encoding rate. In the situation of buffer overflow, non-local coordinated scheduling is also initialised. Overall, this allowed for a 70% increase in parallel session support and reduced delay and packet loss when implemented in a collision-free network.

In MANETs, QoS-adaptive routing protocols have also taken into account the durability of the channel. That is, when channel durability is highly likely, a node can be offered a better connection with a low coverage range. When it is unlikely, the channel should be offered to connectors with larger coverage ranges, or ones that move with the same speed and direction as the node. Associativity Based Routing (ABR) and Signal Stability Routing (SSR) route reactively while considering link quality. The former prefers hop stability and the latter chooses routes based on the Received Signal Strength Indication (RSSI). Hybrid proactive-reactive protocols have also been developed. ZRP uses local proactive routing and non-local reactive routing, while Lightweight Underlay Network Ad hoc Routing (LUNAR) [154] combines reactive path discovery and proactive path rebuilding.

LUNAR is a low complexity hybrid protocol that combines reactive path discovery and proactive path rebuilding at a three second frequency in order to deal with topology changes and remove the need for link repair notification with beaconing. The authors suggest that there is an ad hoc horizon of three hops beyond which a routing protocol becomes ineffective in handling topology changes due to decay in the freshness of routing information and the masking of poor transmission locations by local repair. They further suggest that beyond this horizon control information flooding disturbs neighbouring nodes to a greater degree than it benefits transmitting nodes, as a result LUNAR is limited to three hops. LUNAR uses a subnet illusion emulating a LAN within these three hops to the IP layer in the sender. All IP control traffic is translated into LUNAR specific traffic within this emulated LAN for example and IP ARP request is translated to a Route REQuest (RREQ) rebroadcast with a unique ID to all neighbours. The target node responds with a unicast Route REPly (RREP) via the broadcasting node to the sender that establishes a data delivery path along its route. DHCP messages are similarly translated to the protocol specific address allocation and resource discovery mechanisms.

A number of approaches [62], [177], [169], also including the Extremely Opportunistic Routing (ExOR) protocol, have suggested different degrees of coordination between MAC and routing. ExOR is a routing protocol that selects the best next hop, after each per-hop transmission. It uses an average of one-hop link metric information to do this, therefore the performance improvement is limited until sufficient link metric measurements have been received. Wu and Wu [169] have also proposed the joint use of QoS requirements, MAC queue length and physical SINR to distribute traffic over multiple paths to the receiver. In this network-wide framework, routing decisions are made at each hop, based on link status calculated from SINR and queue length information that is added to RREPs by intermediate nodes. At each hop the modulation mode is tuned to adapt transmission rate in proportion to SINR: a PSNR increase of around 1dB results. Wu and Wu further develop their protocol in [170] to include network congestion-awareness. This is implemented via global signalling of MAC layer utilisation over multiple paths, providing up to a 1.7dB PSNR increase over a protocol without the cross-layer signalling.

Hong et al [62] proposed further merging between MAC and routing with the use of virtual links to avoid processing delays between these layers. They require the link layer to both select the next hop and re-encapsulate packets. This resulted in a 7-10% throughput improvement and 50% reduction in processing time. Similarly, in [177] the MAC layer selects and prioritises paths based on physical link quality and route information. Yamao et al [177] state that the minimum hop-count route chosen by AODV results in the use of long low SINR links that fail under fading conditions. They suggest the use of multi-hop path selection, using shortcut paths and novel control messages to prevent transmission redundancy. For node distances of less that 250m this does result in a transmission delay reduction. However, in moving traditional network functionality to the lower layers the modularity and re-usability of such an approach is low.

2.5.4 Key Findings

Many Cross-layer models exist either for the provision of QoS to SRT streams in static wireless networks or to improve the performance of MANETs. This Section has surveyed the common ground and lessons learned from structural models of cross-layer signalling and then protocol-tuning approaches in these two fields. This has been with a view to advising the open research area of high performance service provision to loss and delay intolerant real-time applications in MANETs.

Predominant network-wide and local-parameter based signalling methods were compared in Section 2.5.2. Table 2.3 notes the conclusions made on their relative overheads when applied in support of ISRT applications in MANETs. Network-wide models are not highly appropriate to MANETs due to the reliance on maintained signalling contact with all intermediate nodes. Higher bandwidth overheads and transmission delay result from the addition of signalling traffic to network load. Cumulative signal transmission and computation delays also impact on packet delay. It is local signalling that better suits MANET nodes that are prone to intermittent loss of contact with other nodes and that are likely to be highly and randomly spatially distributed.

Packet headers and options have been used for signalling within the MN stack but subject the receiving node to high processing overheads. The frequent signalling to higher layer protocols required in these implementations also leads to high communication overheads due to the number of applications and sockets involved. In-band methods where cross-layer parameters move in the direction of packet flow incur lower overheads than out-of-band, as a control path must be maintained in the latter. However, out-of-band signalling does not compete with application flows for resources and is ideal in contention based architectures. The limitation of signalling to packet paths alone also introduces a RTT notification delay. Such delays could be avoided by use of ICMP or similar control packets to enable signal passing between specified nodes. As a result, methods such as ECLAIR and CLASS show a good level of performance for both ad hoc networking and RT support.

Local middleware models from POEM to the Control Middleware Plane outperform packet based approaches. This is because the latter incur increased processing and communication overheads that commute to increased jitter, as per-node processing is no longer possible at line speed. The adverse side of local middleware lies outside of run-time: in being a non-standard kernel component both implementation and porting can be complex. Conversely, localisation in the kernel enables high-speed, execution-concurrent optimisation. Middleware also avoids the resultant packet bursts and corresponding queueing delays that are not ideal for ISRT flows or mobile nodes with limited storage capacity. Among the higher-performance middleware schemes, intralayer optimisers, such as ECLAIR, that use event-based signalling also leave the protocol stack intact. This enables adaptable rapid prototyping, transparency, portability and lightweight design. The lower overheads mean better packet-timeliness guarantees can be provided and in optimising the stack from a single, external location, signalling loop errors are avoided.

It is the protocol parameter abstraction and tuning of a cross-layer design that provides QoS guarantees and table 2.4 indicates some of the parameters available at each layer. With event-based optimisation, when a protocol parameter arrives at a pre-specified threshold value it is abstracted. The pre-optimisation threshold value must therefore correspond to minimum QoS requirements or network provisioning and the aim of optimisation process is to ensure that this value is always exceeded. Tuning models that use either or both of network-adaptive and QoS-adaptive approaches have been considered in Section 2.5.3.

QoS-aware routing is commonly implemented in wireless and ad hoc networks. However QoS-adaptive scheduling for ISRT flows and for MANETs is an area which requires further research. The deterministic QoS guarantees required by these flows depend on timely, ordered and guaranteed packet arrival, to which packet scheduling is key. QoS-aware cross-layer scheduling approaches can benefit

Table 2.3: Comparison of Cross-layer Signalling Mechanisms

Signal Scope	Signalling Mechanism		Transmission Delay	Communication Overhead	Processing Overhead	In-band or Out-of-band
Local to node	ISP	IP packet header	Medium	High	Medium	In-band
		Packet structure	Medium	High	Medium	In-band
	Direct interlayer signalling	CLASS	Low	Medium	High	Out-of-band
		ICMP packets	Low	Medium	High	Out-of-band
	Callback functions		Low	Low	Low	Out-of-band
	POEM		Low	Medium	Low	Out-of-band
	Cross-layer Server		Low	Medium	Low	Out-of-band
	Central Cross-layer Plane		Low	Low	Low	Out-of-band
	ECLAIR		Low	Low	Low	Out-of-band
	Control Middleware Plane		Low	Low	Low	In-band
Network-wide	Packet header		High	Low	Medium	In-band
	ICMP		High	High	High	Out-of-band
	WCI		High	Low	Low	Out-of-band
	CrossTalk		High	Low	Low	Out-of-band

from the use of priority scheduling and timestamping for separate packet and slot queues. These scheduling methods, combined, generally provide the best assurances to delay and jitter-sensitive applications. However, this is at the expense of reduced response rates due to higher node computation and processing requirements, with a corresponding impact on packet delay. For multi-hop networks this is an area that requires further performance analysis and testing as multiple priority queueing requirements result in increased packet delay.

The major difference between traditional wireless networks and the MANET is the dynamically varying resource conditions seen in the latter. Responsively, many cross-layer designs have elected parameters from lower layers, such as received packet power or optimal transmission rate to signify these conditions to higher layers or moved resource allocation to these layers. Such notification and tuning is useful but not always possible. Tuning the MAC layer or requiring it to access physical parameters entails modifications to radio firmware or hardware that have limited accessibility, primarily to vendors. This can reduce the transparency and interoperability of modified nodes. However, network-adaptive routing and admission control based on purely MAC layer information is key to the support of ISRT applications in MANETs.

The high packet loss rates, jitter and varying delay common in a MANET must be addressed rather than compensated for. The low and time varying SINR conditions of a MANET adversely effects routing and admission control. Using MAC layer, transmission and retransmission rates, PER, channel coherence time and packet or ACK timestamps cross-layer middleware can characterise these varying conditions. Managed collaboration between QoS-adaptive routing and admission control that respond to this information can then provide dynamic optimal path selection (identifying alternatives to a congested shortest path) and smooth performance degradation. While the tuning of these parameters can improve performance, through opportunistic resource use, it is also essential that a cross-layer approach should be forward thinking, in terms of re-usability and modularity. This is highly dependent on the signalling method implemented.

Due to the intermittent connectivity experienced in MANETs, network-wide interlayer signalling is difficult to implement and centralised control is not possible. This is due to a number of factors such as the memory and processing constraints within nodes, the fact that links are frequently set up and torn down and also the low link capacities for which data and signals must compete in the wireless medium. In addition a large number of MNs must contain cross-layer capability in order for network-wide schemes to be viable, unlike those implemented within the MN. In-band signalling implemented within the MN stack benefits from the association of information with particular packets. However, in-band mechanisms exhibit higher overheads than out-of-band and their signals also compete with data packets for scarce bandwidth. Table 2.3 outlines the signalling mechanisms considered in Section 2.5.2 and indicates that approaches incorporating callback functions, which provide event-based rather than fixed frequency signalling, have the lowest resource requirements of the out-of-band mechanisms. When cross-layer middleware is added to such a scheme these functions can then be registered with the entity, all signalling managed from a single location and actual protocol layers need not be modified.

External centralised and decentralised middleware cannot be relied upon for wide-scale management functionality when based in MANET nodes that are prone to intermittent loss of contact with some or all other network nodes. The signalling overhead required by distributed decentralised schemes, while manageable for small network topologies can increase exponentially with number and spatial distribution of nodes. While the overheads for interlayer managers and optimisers is null or incurred at the signalling stage only, functionality is limited to application oriented and not scalable to heterogeneous technologies. Intralayer schemes generally incur high overheads due to the use of function calls and control messages but these also reduce the frequency of signalling required, as communication can be event based. Use of an intralayer entity also limits optimisation to a single location, avoiding the conflicts or loops that can plague schemes of multiple interlayer entities.

With event based signalling, an event such as the reduction of channel conditions to below an optimal standard for video streaming, represented by the arrival of a particular parameter at a predefined threshold value induces cross-layer optimisation to commence. When the parameter arrives at this threshold value it is abstracted from its protocol layer for optimisation.

Evidently an appropriately QoS aware framework for ad hoc networks requires the implementation of not only an appropriate method of cross-layer signalling support but also the selection of effective and contention aware QoS control schemes. The high error rates of ad hoc networking in combination with the stringent delay and jitter requirements of RT flows prove to be best optimised through the selection of appropriate admission control; scheduling and routing. Through cross-layer control these can be used to combat the causes of congestion and avoid a control or recovery requirement. It is also through the use of only these approaches that overheads can be minimised in order to ensure that the delay and jitter requirements of RT flows can be met.

The indication of an event such as imminent handoff is taken as a threshold parameter value. Therefore parameters are abstracted and passed to optimising middleware from single layers following arrival at this threshold value. The complexity of a cross-layer implementation is proportionate to the number of parameters abstracted and optimised. It follows that minimising the number abstracted by either subtracting those that may be indirectly tuned, or enabling an optimiser to infer a higher layer parameter from an abstracted lower layer parameter can make a scheme more flexible. Ultimately the selection of parameters for abstraction depends on the QoS control mechanisms to be implemented and therefore the participating protocol layers. Section 2.5 compares control mechanisms that may be implemented in order to meet the QoS requirements of RT applications in ad hoc networks.

In line with these suggestions a cross-layer design for ISRT in safety-critical MANETs should avoid spaghetti design, being instead modular, transparent and reusable. However, given the singular characteristics of an ad hoc network this does not necessitate re-usability in other types of wireless network. Therefore MANET-specific conditions must be taken into account in developing a design, for example, scheduling must not assume uniform traffic distribution or routing, symmetrical links. As learnt from the development of internet protocols, a codified approach to development as well as clear publication of the conditions under which a cross-layer design will fail can better ensure its sustainability.

Table 2.4: Parameter Tuning for SRT in MANETs

Protocol Layer	Abstractable Parameters	Tunable Parameters
Application Layer	Delay tolerance, acceptable delay, acceptable jitter, required bandwidth, acceptable packet loss ratio **Use:** Avoiding low SINR triggered congestion response, QoS-adaptive routing and scheduling	Source rate, encoding format, compression **Use:** Network-adaptive rate control
Transport Layer	RTT, Recovery Time Objective, MTU, total packet loss and actual throughput **Use:** Avoiding low SINR triggered congestion response, QoS-adaptive routing and scheduling	Sending rate, MTU **Use:** Network-adaptive rate control
Network Layer	Timestamps of mobility events, route and network interface used **Use:** Network-adaptive admission control	Route selected, network interface selected **Use:** Network-adaptive routing, QoS-adaptive routing and scheduling
MAC Layer	FEC scheme, retransmission totals, frame lengths, time stamps of transmission and handoff events, transmission rate and PER, ISS, nodes in transmission and carrier sense range, channel coherence time **Use:** Network-adaptive admission control; Network-adaptive routing	TDMA time slots, FEC scheme **Use:** QoS-adaptive scheduling

2.6 Summary and Critical Analysis

The goal of developing future military wireless networks such as that considered by the Network Enabled Capability project discussed in Section 2.2.3 hinges on the need to address deterministic HRT support in wireless MANETs. This type of support has previously been characterised by wired networks such as AFDX. The dynamically varying links of MANETs are subject to numerous non-deterministic factors, including interference, multipath fading, shadowing and problems of hidden node identification.

Delay-intolerant, loss-intolerant HRT provisioning relies on synchronisation and stringent scheduling that is undermined by these unpredictable MANET link qualities. Over-provisioning of bandwidth is used in wired networks. Conversely, over-provisioning reduces performance when devices and channels are already resource-constrained. Therefore, these applications can only feasibly be supported as loss-sensitive, timing-critical ISRT. Therefore, loss and delay must be bounded, due to the safety-critical and military scenarios in which they are to be used.

When developing a model that addresses the loss and delay requirements of bursty RT traffic, it is per-hop packet delay and loss that are the strongest indicators of performance, as these aggregate to E2E delay and loss. Per hop delay can only be bounded if each of its components (discussed in Section 2.2.2.1) are bounded; in particular contention and queueing delay, which are generally the most substantial delay components. Consequently, a local approach implemented at each node can meet E2E ISRT requirements if loss and delay are bounded at each hop.

Wireless and MANET Protocols, working oblivious of resource availability or application requirements, have been shown to under-perform in dynamic channel conditions. Cross-layer signalling has thus been embraced in ad hoc networking (Section 2.5). Cross-layer design is divided between models that selectively tune certain protocol functions (Subsection 2.5.3) and architectures intended to reduce cross-layer signalling overheads (Subsection 2.5.2). In order to provide successful ISRT MANET support, both components must ensure that timing and loss guarantees are not invalidated.

The first component is dependent on the MANET and SRT protocols or control mechanisms selected, and analysis in Section 2.3 identified the impact of contention, load control and path discovery. Contention detection combined with load control is of significant importance in wireless networks where hidden nodes can induce existing flows to violate their QoS requirements (Subsection 2.3.1). Localised reduction of contention for a shared link can be used to reduce collisions and the congestion that results from heightened loss recovery requirements. By avoiding extended queueing and contention delays, E2E delay can then also be constrained. In a MANET, control packet-based maintenance leads to unacceptably high levels of overhead and competition for bandwidth, therefore, load control must be distributed.

Selective tuning of these protocols based on MAC or network layer information, as discussed in Subsection 2.5.3.1, enables optimisation of resource use. Therefore, the causes of congestion can be avoided for RT traffic (Subsection 2.3.1). Appropriate path discovery is also important to the assurance that QoS agreements can be upheld (Subsection 2.3.2). If routing is based solely on topological characteristics, allocations are fair but resource use is not maximised. When paths are selected according to MAC layer signalling of channel conditions, high SINR channels can be avoided. Links may also have low levels of coherence due to node mobility, which can be selected depending on the requirements of the new flow.

Many novel designs have been proposed to improve existing wireless and ad hoc protocols. A number of these previously implemented approaches have shown reductions in mean delay or packet loss, however, E2E delay and packet loss have not been bounded with RT MANET provisioning (Section 2.3). The reason for this is that short-term contention induced delays and collisions, channel quality variation, the complex functioning of the wireless MAC layer and rapid topology changes have not all been addressed. For example, traffic shaping combined with bandwidth reservation has been used to guarantee packet delivery, the former to ensure that flows adhere to their original specifications [140]. However, node mobility, link breakages and the underlying process of the MAC layer such as repeated backoff introduce complexity and cause violation of reservation agreements. Resource reservation also relies on signalling that competes for resources, causing flows to be extensively queued. For MANETs to be viable solutions to ISRT applications there is a requirement for load control to be responsive to multiple layer-1 and -2 factors that contribute to increased loss and delay.

Self-configuring and dynamic path discovery is a necessary basis of successful MANET functioning, therefore many ad hoc routing protocols have been developed (Subsection 2.3.2). The majority rely on path setup using repeated control packet exchanges. Delay-sensitivity is then addressed in ad hoc routing, using measurements or estimation of delay based on these exchanges or based on novel control packet transmission [151,174]. Routes that satisfy delay requirements may then be selected, if available, avoiding paths that cease to uphold these requirements. This means that delay cannot be bounded if path quality changes rapidly, as this leads to regular use of suboptimal channels and frequent rerouting. Therefore, QoS-aware routing protocols alone are only able to reduce delay but not provide maximum delay guarantees.

As previously discussed, upholding bounded delay and loss requirements is also dependent on the cross-layer architecture design. From the perspective of military and safety-critical networks, such a design should also be scalable to new and legacy protocols, when the latter undergoes extensive safety testing (Subsection 2.5.2). Subsections 2.5.2.2 and 2.5.4 discussed the value of local middleware above other signalling methods. These middleware architectures ensure tuning is not limited to particular protocols, packet paths, or maintenance of signalling contact with all nodes. The adverse side of local middleware lies outside of run-time: in being a non-standard kernel component both implementation and porting can be complex.

In accessing all or multiple protocol layers, optimisation with middleware can be both QoS-aware and network adaptive (Subsection 2.5.3). Intralayer middleware, such as the ECLAIR architecture that was proposed to improve TCP performance [124], use event-based signalling and leave the protocol stack intact. Without direct protocol modification the framework can be adaptable and scalable, as well as lightweight. Packet-timeliness can be supported in such frameworks due to minimal processing overheads and in optimising the stack from a single, external location, signalling loop errors are avoided. Protocol-independence enables generic support of contemporary and safety-certified network protocols by modifying parameters that a protocol accesses rather than the protocol functionality itself.

Cross-layer middleware approaches have been conceptually designed to rely on local information in order to conserve MANET bandwidth and to support both contemporary and extensively tested legacy equipment. These designs have not been validated in MANETs and have not addressed the need to limit the impact of cross-layer signalling on E2E packet delay and loss. Using middleware to bound delay and jitter for RT applications is still an open research area. Therefore a requirement has been defined for an intralayer middleware architecture that is demonstrated to be scalable to novel and safety-certified protocols, without reliance on global signalling. Investigation must therefore be made into the conditions experienced by ISRT flows in MANETs. This investigation is done in the following chapter, in order to define how network-adaptive and QoS-aware routing and contention control might also be used to fulfil the bounded timing requirements of ISRT.

Chapter 3

Analysis of Real-time Performance in MANETs

3.1 Introduction

In this Chapter, end-to-end (E2E) delay, jitter and packet loss of CBR applications have been analysed in small MANET topologies. The previous review of the related literature has established that node mobility and wireless channel quality are significant factors that influence the performance of ISRT applications in MANETs. Correspondingly, this analysis aims to investigate these factors and demonstrate that cross-layer responsiveness to conditions at lower layers can be used to constrain delay, jitter and packet loss. In MANETs, resource quality and availability fluctuates over time due to neighbour and environmental interference as well as signal fading and attenuation, this necessitates more dynamic delay, jitter and loss control over the network. Additionally, in comparison to wired networks, this results in lower and less predictable supported data-rates and link capacities. Evaluating performance at a node level is requisite to the development of a cross-layer scheme for RT applications in MANETs, in particular both the best and worst-cases of performance should be identified. Therefore, to begin with, RT performance in simple ad hoc networks of two to three nodes has been analysed to identify a baseline scenario of best-case MANET performance.

The aim of this project has been to develop an approach to improve RT performance that is independent of application transmission setting and MANET configuration and capable of providing guarantees on bounded delay, jitter and packet loss ratio, thus supporting time-critical and safety-certified applications. Therefore, the simple topology configurations have then been expanded to a range of transmission setting and MANET scenarios that resulted in horizontal handoff and shared channel contention. These simulation configurations also emulate the safety-critical scenarios of RT in MANETs. Two performance scoping experiments have been conducted into the effects of handoff and contention. Additionally, to identify the specific lower layer statistics available that determine the cross-layer tuning response possibly when conditions deteriorate. The topologies and scenarios used have then provided a foundation for testing the performance improvement approach detailed in Chapters 5–6.

Network simulation has been used as the basis for investigating both MANET protocol and cross-layer middleware performance as it supports the appraisal of a cross-layer approach without the overheads of a real-world implementation. The complexity of cross-layer design can be fully represented in a network simulator and would need to be simplified for investigation using analytical models. However, investigation of sensitivity to parameters requires simulated models to be tested in a large number of scenarios. Network simulation enables testing of a MANET proposal under high datarates and in large topologies: the only limitation on scale is to reduce computational overheads and runtime. In a real world implementation it is not often possible to develop an ad hoc network with inter-nodal distances of hundreds of metres or high node speeds.

Simulation provides an optimal option for analysis of a military based scenario with larger distances between communicating vehicles. It was however, essential to ensure that the simulated design could work under simulation of realistic environmental conditions. The network simulator, ns-2 is the most popular tool utilised in network research [61, 75]. However, the ns2-MIRACLE libraries added to this simulator provide the ability to simulate randomly generated environmental interference conditions and, through restructuring of protocol classes into individual modules, more realistic lower layer protocols than ns-2 [15]. Compared to ns-2, the functions of the lower layers are appropriately attributed to each separate protocol.

The majority of IEEE 802.11 WLAN cards respond to channel noise and interference with a multirate auto-fallback mechanism, this is supported in ns2-MIRACLE, but not in ns-2 [14]. Additionally, the structure of ns-2 only enables imitation of a cross-layer middleware approach by piggybacking control messages in data packets [158]. Simulating the out-of-band signalling of cross-layer middleware in this way would give erroneous results as cross-layer messages should be sent in an approach that is asynchronous to data transmission. ns2-MIRACLE is therefore the only network simulator specifically designed to simulate asynchronous messaging outside of the protocol stack. ns2-MIRACLE is being implemented in a wide range of wireless network projects and is largely based on ns-2 but ns-2 has been more rigorously validated. Therefore, in Section 3.4 the performance of MANET protocols, under the same settings has been investigated in both simulators to validate the similarities between ns2-MIRACLE and ns-2.

3.2 Baseline Ad hoc Network Performance

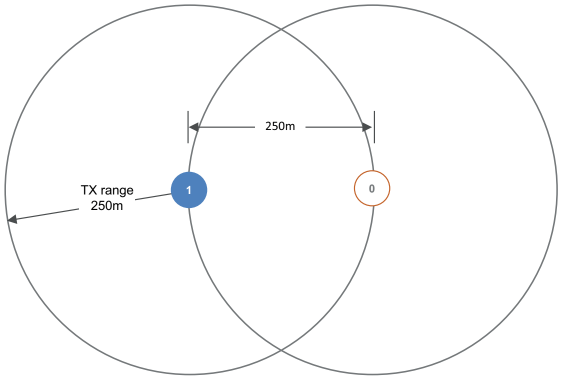
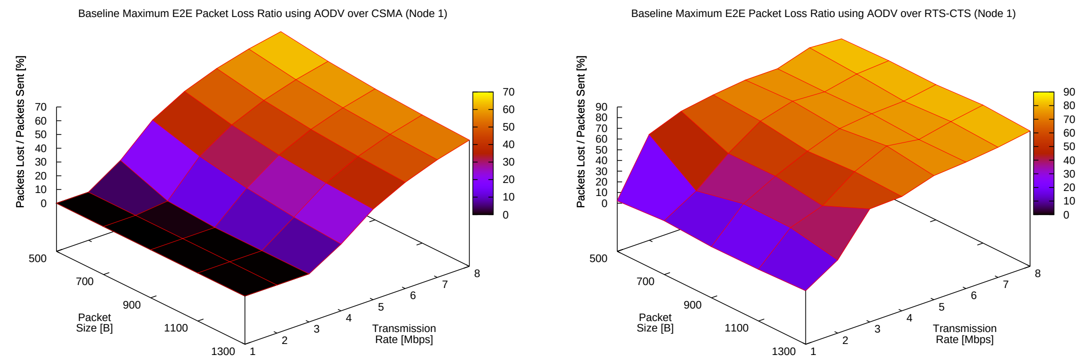


Figure 3.1: Baseline Simulation Network Topology

Chapter 2 showed that ISRT applications require bounded guarantees on delay, jitter and packet loss in order to meet minimum performance requirements. There are several factors influencing these performance metrics, which are analysed in an ideal static wireless scenario in ns2-MIRACLE to provide a baseline of best-case performance. The simulation topology is shown in figure 3.1 and consists of a single static transmitter-receiver pair, transmitting CBR packets over AODV-UU and IEEE802.11, with node 1 transmitting to node 0. The nodal configuration discussed in the following section and given in table 3.1 is based on the default configuration in the ns-2 simulator, providing a transmission range of 250m.

In order to ensure the validity of the data, all results are means collated from 10 runs of each simulation. The transmission rate and packet size were varied to investigate the impact on packet loss and timing. Transmission rates (TR) of 1-8 Mbps were implemented, with increments of 1Mbps and packet size was varied between 300-1300B.

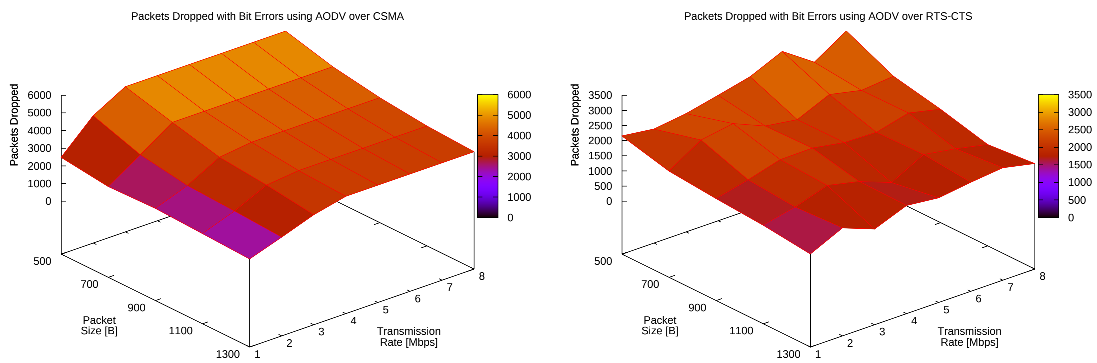
Figure 3.2(a) demonstrates that when CSMA alone was implemented, packet loss remained below 20% until the transmission rate approached 4Mbps. At higher datarates, even though the destination was only 1-hop away, packet loss increased rapidly. When the packet size is small, more packets are transmitted per second for the same datarate. With only one transmitter, collisions between data packets did not occur. However, as a result of receiver control packets, the incidence of packet dropping due to interference errors was the main cause of packet loss with CSMA (figure 3.3(a)). When smaller packet sizes were implemented, packet errors became more prevalent. This was a result of the higher number of packets transmitted onto the link per second, for the same datarate.



(a) CSMA

(b) RTS/CTS

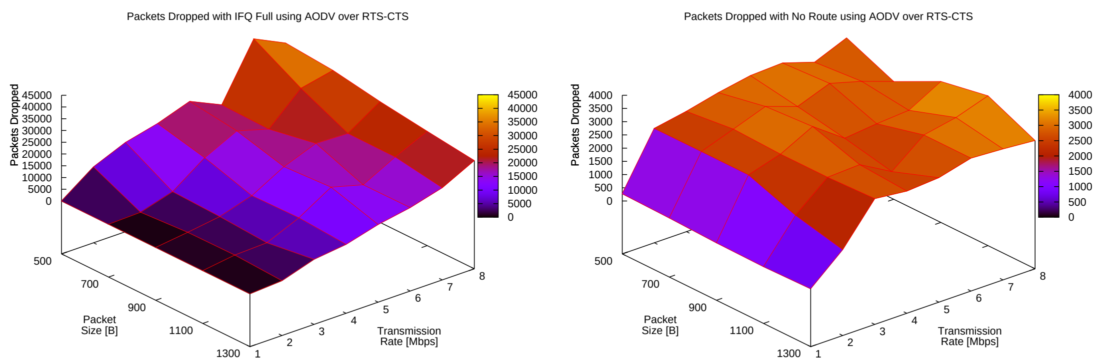
Figure 3.2: Packet Loss Ratio in both Baseline Scenarios



(a) CSMA

(b) RTS/CTS

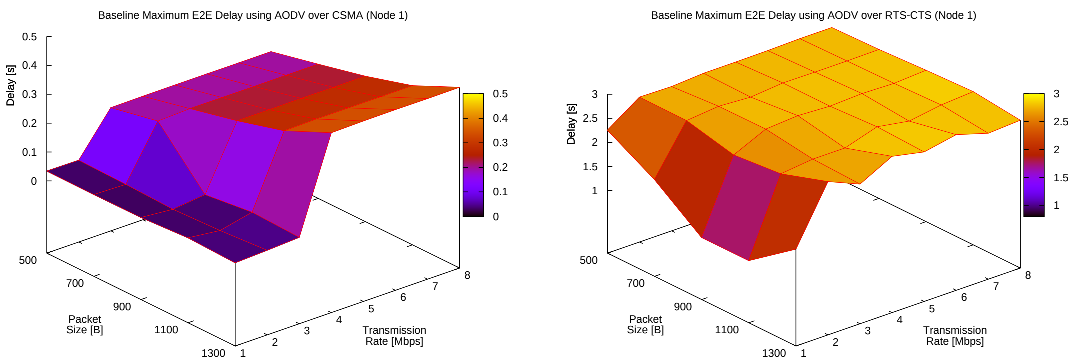
Figure 3.3: Incidence of Packet Errors in both Baseline Scenarios



(a) Incidence of IFQ Overflow

(b) Incidence of No Route Found

Figure 3.4: Other Packet Drops with RTS/CTS in Baseline Scenario

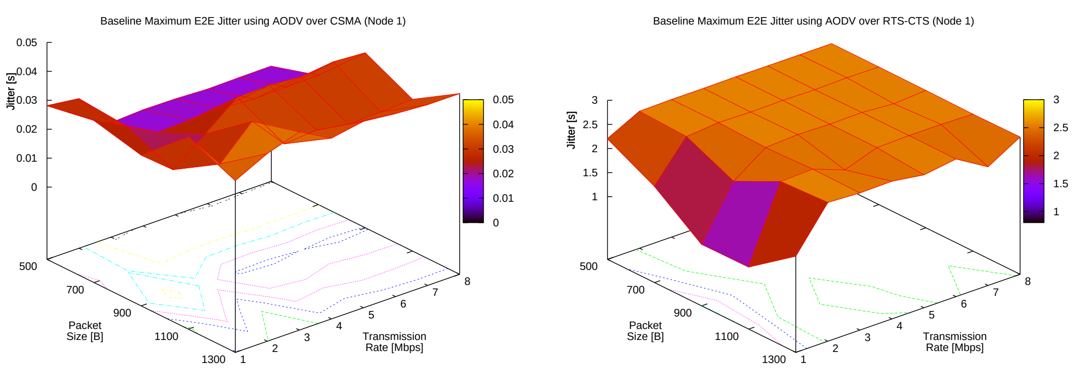


(a) CSMA (b) RTS/CTS

Figure 3.5: Maximum E2E Delay in both Baseline Scenarios

With CSMA, maximum E2E delay and jitter in the stream are influenced by wireless packet loss, as a result of error recovery and increased buffering requirements, as well as the intrinsic mechanisms of the MAC layer. At higher application rates, packets may be enqueued faster than they are dequeued as a result of the multirate auto-fallback aspect of IEEE 802.11. The auto-fallback mechanism reduces datarates in response to noise detected on a channel. Figure 3.5 shows that delay rose rapidly at datarates above 2Mbps due to increased queuing requirements, although the IFQ did not overflow. Conversely, high packet size and datarate resulted in the highest delays.

The IEEE 802.11 RTS/CTS mechanism is implemented as virtual carrier sense in order to prevent hidden transmitter collisions and interference as well as avoiding the exposed node problem. RTS/CTS use resulted in lower performance in



(a) CSMA (b) RTS/CTS

Figure 3.6: Maximum E2E Jitter in both Baseline Scenarios

this scenario than with CSMA alone (figure 3.2(b)). At 2Mbps, the packet loss ratio exceeded 20%, rising rapidly to 90%. RTS/CTS, or virtual carrier sense, enables MAC layer detection that the channel is busy. The mechanism introduces artificial delay into the stream by buffering packets during the exchange of RTS/CTS control packets. The increased presence of control packets on the link also results in loss of both RTS/CTS and routing control packets, requiring further buffering during path maintenance and RTS/CTS exchange (figure 3.4(b)). This eventually results in IFQ overflow (figure 3.4(a)).

Figures 3.3(a)-(b) indicate that packet errors were reduced by up to 50% through the use of RTS/CTS as the receiver will not transmit control packets while waiting for a data transmission. Maximum delay and jitter guarantees cannot be provided by RTS/CTS, which is used to reduce collision incidence but can increase other causes of packet dropping.

Protocol mechanisms at multiple layers contribute to network jitter, such as the MAC layer error recovery and rate control mechanisms that respond to increased signal fading. Figure 3.6(a) shows that maximum jitter was highest when the largest packet sizes were implemented, and the lowest datarates, with CSMA. Whereas, with RTS/CTS, maximum jitter exceeded 1s when the datarate was as low as 1Mbps.

These simulations demonstrate the benefits of overprovisioning to maximum delay and jitter-sensitive applications in a wireless network. The introduction of multiple alternative network paths can also reduce queueing requirements on the E2E path, but introduce new problems such as non-robust channel selection, contention and bottleneck links. These issues are further investigated in the following sections. However, these baseline simulations provide some examples of best-case MANET performance. These are further used to evaluate the success of the middleware implementations in Chapters 5–6.

3.3 Simulation Configurations

3.3.1 Topology and Setup

The configuration of each node used in the simulations is described in table 3.1. The default ns-2 queue discipline, droptail, was used. The simulation module `Mac/802_11/Multirate` supports multi rate transmission, which is a function of most wireless LAN cards. When signal strength on a channel deteriorates, the MAC layer datarate is adapted proportionately in order to avoid related packet loss. A basic transfer rate of 12Mbps was used in the simulations. This was the maximum datarate and meant that the MAC transmission rate could also be

Table 3.1: Simulation Configuration

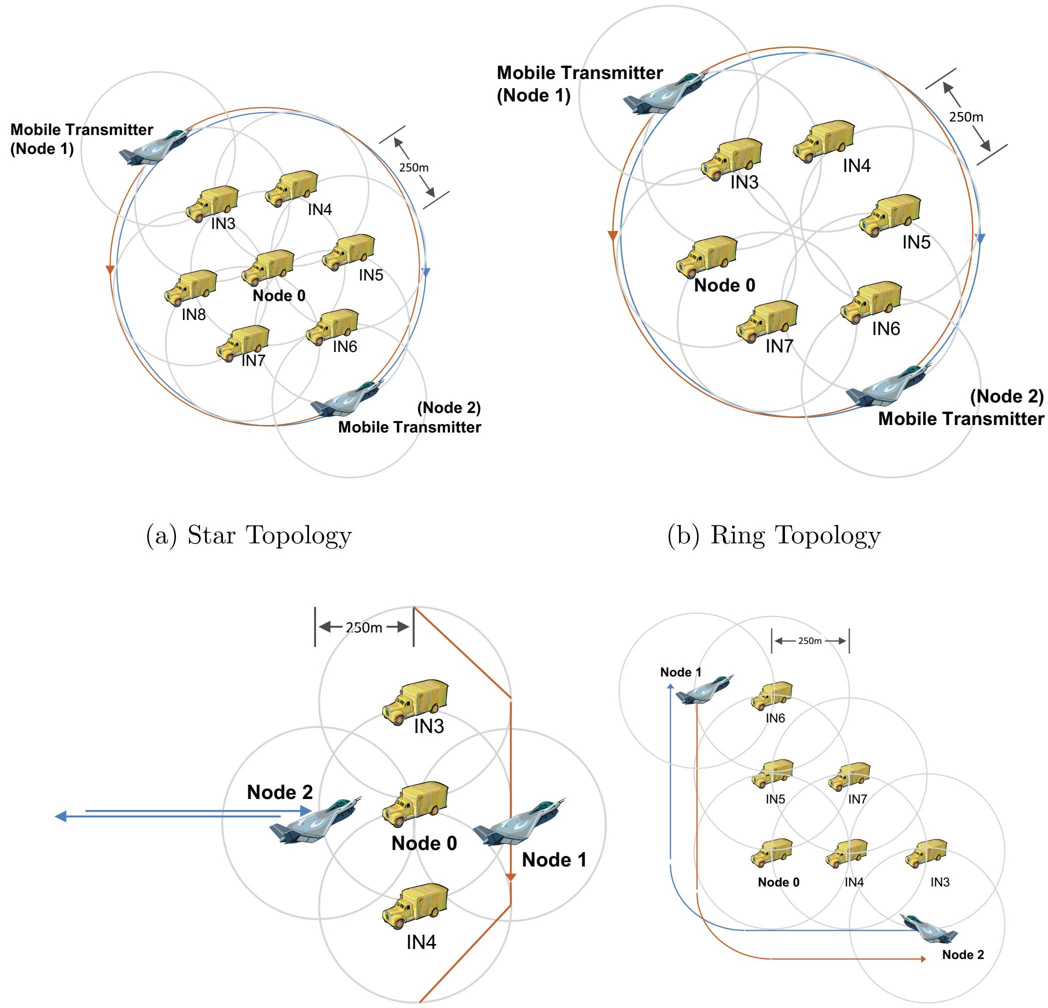
Parameter	Configuration
Radio Propagation Model	TwoRayGround
VoIP Codec	G.729A
Routing Protocol	AODV-UU
Wireless Mode	IEEE802.11b
Virtual Carrier Sensing	OFF (unless stated)
Interface Queue	DropTail/PriQueue
Transmission Power	24.5dBm
Interface Queue Length	100 packets
Transmission Range	250m
Carrier Sense Range	550m
Carrier Sense Range (hidden node)	250m
Transmission Data Rate	12Mbps (with auto-fallback)

stepped between to 1, 2, 6 or 12Mbps using this auto-fallback mechanism.

CBR traffic was sent between the transmitter, node 1, and the receiver, `node0`). AODV-UU is the only available ad hoc routing protocol in the ns2-MIRACLE simulator. AODV-UU is the most widely recommended AODV implementation, developed for the Linux operating system. The developers of AODV-UU have ported the implementation to both ns-2 and ns2-MIRACLE simulators. AODV-UU adds unidirectional link detection and multiple interface support to improve the throughput of AODV [26, 57], however as both of these settings have not been implemented in these simulations, the terms AODV-UU and AODV are used interchangeably from here.

The topology shown in figure 3.7(a) has been used for initial investigation of real-time performance in ns-2 and ns2-MIRACLE. In this topology one transmitter, node 1, orbits an association of forwarding intermediate nodes (IN) surrounding a central receiver, node 0. A second CBR source, node 2, is added to the topology which competes for channel access with the first. This topology was used to both investigate performance of RT applications in MANETs and to validate the horizontal handoff performance improvement approach developed by this project, as given in Chapter 5.

Figure 3.7(b) demonstrates a similar ring topology with the receiver no longer equidistant from all forwarding INs, creating an increasing E2E path length as the transmitters orbit the network. The bus and tree topologies, used to evaluate the network performance under shared channel contention with hidden and



(a) Star Topology (b) Ring Topology

(c) Bus Topology (d) Tree Topology

Figure 3.7: Network Simulation Topologies and Setup

exposed transmitters, are shown in figure 3.7(c) and (d). In order to ensure that link sharing did not coincide with vertical handoff or initial route setup, the two transmitters travelled across the network to a location where they competed for the channel to a mutual end receiver (node 0). Each transmitting node in the simulations moved at 10m/s, unless otherwise stated.

3.3.2 Simulation Scenarios used for Validation

In order to test the middleware approach developed by this project in line with the objectives discussed in Chapter 1, it has been validated in ns2-MIRACLE within a range of MANET topologies and mobility scenarios and under varying application configuration aspects. The topologies used are shown in Figures 3.7(a)-(d) and ns2-MIRACLE simulation configurations in table 3.1. Application transmission settings and MANET configurations create dynamic network conditions at layers-1 and 2 that contribute to high loss and variable E2E delay. The purpose of this validation is thus to ensure that the performance optimisation approach supports RT applications within, and independent of, these network dynamics. Applications that are specifically RT, which are subject to completion time constraints to manage their resources, may transmit CBR or VBR traffic, depending on the application design. The number of test variables used in this thesis has been deliberately limited, based on the feasibility of analysis. Therefore these traffic types, rather than specific application output examples, were selected for use in validation.

The ns2-MIRACLE simulator was installed and simulations were run and analysis carried out on two laptops. Firstly a VirtualBox running Ubuntu 11.04 on a host MacBook Pro running OS X that was equipped with a 2.4GHz Intel Core i5 processor and 4GB RAM. The second machine was a Toshiba Tecra with 2.20GHz Intel Core 2 Duo T7500 and 2GB RAM. For 10 simulation runs with each variation within the scenarios outlined in the following section, 960 simulations were run in total. A simulation with two mobile CBR sources took approximately 2 minutes to run. Performance was analysed using the key metrics of packet loss ratio, maximum delay and maximum jitter. One Perl script and six awk scripts were used to calculate instantaneous delay, jitter, goodput, packet loss ratio, next hop selected and receive signal strength from the simulation trace output and separate the statistical data relating to individual traffic sources. Bash shell scripts were then used to compute maximum delay and jitter and packet loss ratio from the awk and Perl output.

3.3.2.1 Testing Transmission Setting Independence

Two simulation scenarios provided a range of application transmission conditions:

Scenario 1 was used to validate an ability to react to the network dynamics caused by variation in CBR transmission rate (TR) and packet size (PS). Transmission rate was incrementally increased for all transmitters between 1-2Mbps at increments of 0.25Mbps, packet size was also varied between 500-1300Mbps with increments of 200Mbps.

Scenario 2 utilised heterogeneous packet size and transmission rate values at each of three CBR sources. All traffic configurations, used in Chapter 5, are shown in table 3.2. In Chapter 6, rates of 1-5Mbps were combined with packet size of 5-1500B, with the second node transmitting at either 1Mbps more than node 1 or with packets 200B larger than node 1. When each node transmits with different packet size and transmission rate, bandwidth requirements differ between nodes and network jitter is likely to increase as packet delays in enqueueing, dequeueing and transmission become more varied. Additionally, when nodes transmit with different packet size and transmission rate, RTS/CTS handshaking cannot function correctly in avoiding the hidden node problem as one node may not overhear the handshake of its neighbours.

Table 3.2: Transmission Settings for Scenario 3

Sub-Scenario	Transmission Rate[Mbps]			Packet Size [B]		
	Node 1	Node 2	Node 3	Node 1	Node 2	Node 3
2.1	0.3	0.1	0.7	500	900	1200
2.2	0.3	0.6	0.2	800	1000	600
2.3	0.3	0.7	0.1	600	1300	1000
2.4	0.2	0.4	0.5	900	1000	1500
2.5	0.4	0.6	0.4	1000	1100	1300

3.3.2.2 Testing MANET Scenario Independence

Three simulation scenarios provided a range of MANET configurations:

Scenario 1 added an increasing number of extra CBR or bidirectional VoIP connections (up to five) to the simulation. This was to test the ability to improve performance while all of these transmitters competed for channel access with each other, providing dynamic levels of channel contention. With multiple transmitters moving in the pattern of the military scenario, each subsequently added transmitter followed the same route as node 1, but with a separation more than or equal to the maximum transmission distance (250m). In scenario 1(a) CBR flows were transmitted to node 0. Scenario 1(b) then compared results when bidirectional VoIP traffic sources are instead implemented at the application layer, with 0.5Mbps CBR background traffic. The ns2-MIRACLE VoIP sources use a VBR traffic pattern that consists of alternating periods of talk and silence [44]. The length of each period is determined according to a random selection from a Weibull distribution. The developers of the VoIP source have ported the implementation to both ns-2 and ns2-MIRACLE simulators. The source simulates a jitter buffer and generates VoIP traffic patterns for different codecs, the default of G.729A was used. The G.729A codec compresses audio data into packets of 10ms duration, and operates at a bit rate of 8Kbps.

Scenario 2 utilised different topologies that had varied mean shortest hop counts (HC). This was to demonstrate scalability to improve performance without reference to global conditions. A star topology with HC = 2.1 (figure 3.7(b)); a tree topology with a HC of 2.2 (figure 3.7(c)); a ring topology with a HC of 2.3 and linear bus topology with HC = 2.5 (figure 3.7(b)) were implemented. The protocol configuration of nodes was the same as in previous scenarios.

Scenario 3 was used to identify the impact of increased speed of mobile nodes on performance. A CBR transmitter moving at speeds of 10-50m/s, at increments of 10m/s, was implemented to validate an ability to adapt quickly to rapid, non-uniform network changes.

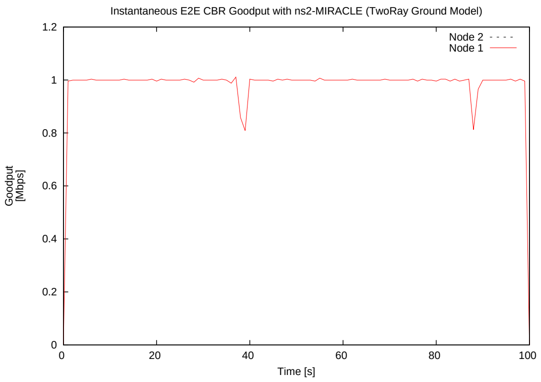
3.4 ns2-MIRACLE Evaluation and Investigation of MANETs

The baseline simulations in Section 3.2 indicated the impact of lower layers on real-time performance in a static wireless network. The mobility pattern of nodes in a MANET changes these dynamics and introduces a requirement for more regular flooding of control packets by ad hoc routing protocols to enable rapid and dynamic route detection and maintenance. Therefore, the following Section conducts an analysis of CBR performance in terms of E2E delay, jitter and packet loss in the MANET. The star topology shown in figure 3.7(a) has been used to investigate the impact of node mobility with a single CBR source in Section 3.4.1 and the bus topology in figure 3.7(c) to demonstrate hidden node contention in Section 3.4.2. This analysis has been conducted in two simulators: ns-2 and ns2-MIRACLE, utilising the same nodal configurations (table 3.1), with the secondary purpose of comparing the results of the two simulators.

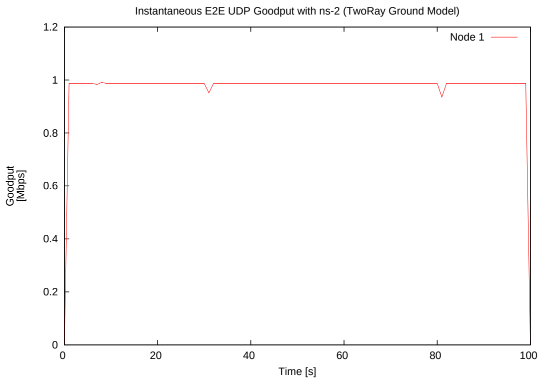
ns2-MIRACLE aims to provide a realistic MANET simulation environment than ns-2, through the implementation of interference models at layer-1 and packet error models at layer-2, as well as a more complex two-ray ground propagation model [15]. As a result, it is expected that packet loss and dynamic variation in goodput is generally higher in the former, for the two-ray ground propagation model. However, ns-2 also provides a comprehensive shadowing propagation model, which has also been investigated.

3.4.1 Single CBR Source Orbiting MANET

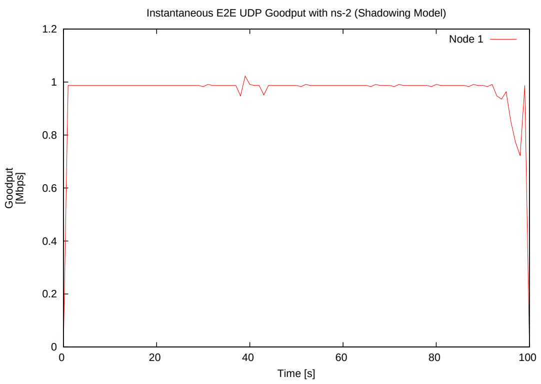
The TwoRayGround radio propagation model includes both line of sight and multi-path, ground-reflected signal transmissions. The Shadowing model extends the effects of multi-path propagation to signal fading due to the presence of obstacles. Figure 3.8 shows the goodput for CBR transmissions from node 1 to node 0, in ns2-MIRACLE with the TwoRayGround model; ns-2 with the TwoRayGround



(a) CBR Goodput with ns2-MIRACLE (TwoRay Ground Model)



(b) UDP Goodput with ns-2 (TwoRay Ground Model)



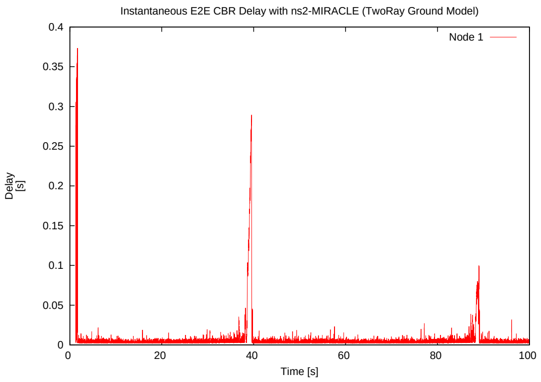
(c) UDP Goodput with ns-2 (Shadowing Model)

Figure 3.8: Goodput Comparison of ns-2 with ns2-MIRACLE (Single Mobile Transmitter)

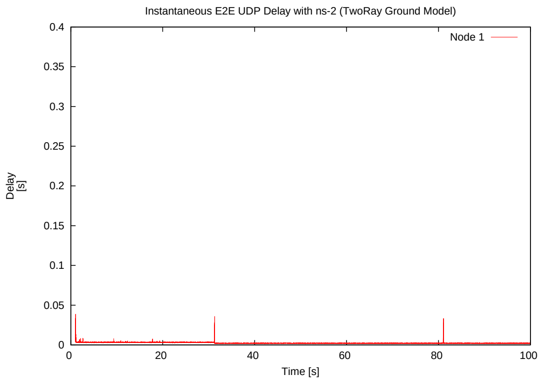
model and ns-2 with the Shadowing model, respectively. At 2 seconds into the simulation, node 1 joined the MANET and began transmitting a CBR stream of 500B packets at 1Mbps. The intitial path setup cost of an ad hoc routing protocol are high as the E2E path is setup on demand using flooding of RREQ packets and dependant of return of RREPs on the best path from the receiver.

As node 1 orbited the star topology, handoff from one forwarding IN occurred approximately every 50s. From 30s, a new IN (node 3) became available to node 1 as the previously used link to node 2 began to degrade. Correspondingly, goodput dropped as a result of MAC autofallback responding to the signal fading, increased packet errors and gradual rerouting by AODV-UU to the new path. When the ns-2 Shadowing model was implemented, this degradation was more evident and similar to results with the ns2-MIRACLE TwoRayGround model.

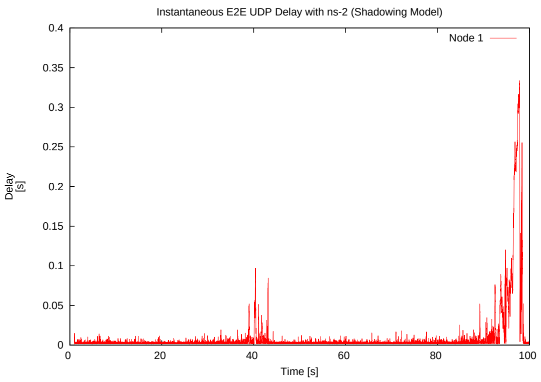
Correspondingly, increases in CBR delay of up to 0.3s resulted during handoff



(a) CBR Delay with ns2-MIRACLE (TwoRay Ground Model)



(b) UDP Delay with ns-2 (TwoRay Ground Model)

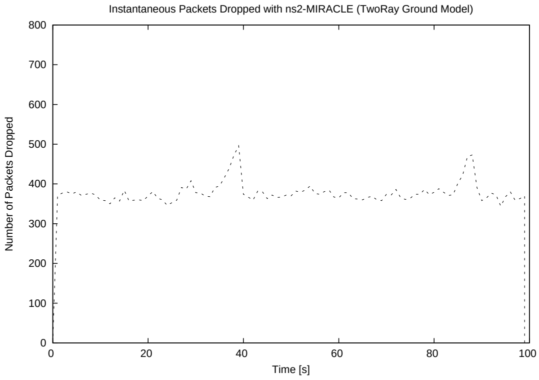


(c) UDP Delay with ns-2 (Shadowing Model)

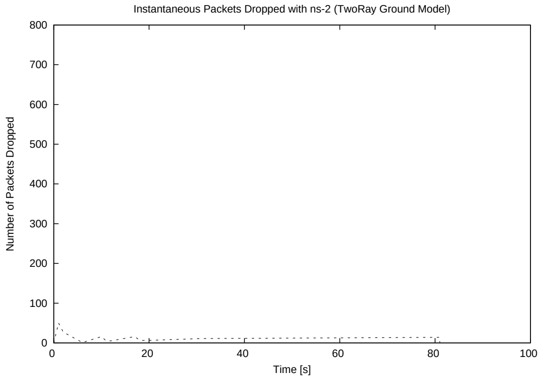
Figure 3.9: Delay Comparison of ns-2 with ns2-MIRACLE (Single Mobile Transmitter)

in both simulators. However, while the instance of peak delay coincides between the TwoRayGround models in both simulators, the simplicity of the ns-2 model results in a lower impact of slow handoff from a fading channel on peak delay (figure 3.9). Although, notably, handoff created packet delays that were similar to those resulting from initial setup of the E2E path. As expected, the number of packets dropped per second with ns2-MIRACLE was higher than ns-2, but both simulators indicated that peak loss occurred during horizontal handoff.

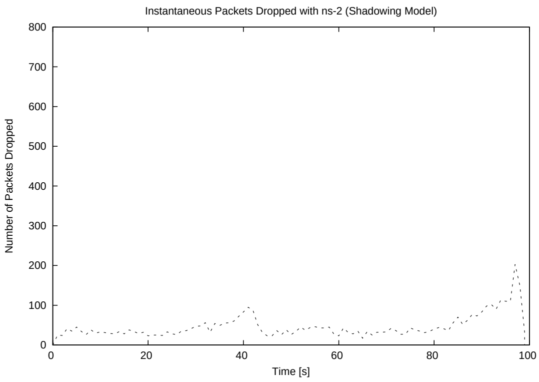
Figure 3.11a) demonstrates the control packet received signal strength (RSS) at node 1 from each of the forwarding INs and the receiver. While the RSS of packets from node 4 is higher than those from node 3 at approximately 30s, the next hop implemented by the ad hoc routing protocol in node 1 only changes after a further 10s and switches between nodes 3 and 4. This is repeated with the second handoff.



(a) CBR Packet Loss with ns2-MIRACLE (TwoRay Ground Model)



(b) UDP Packet Loss with ns-2 (TwoRay Ground Model)

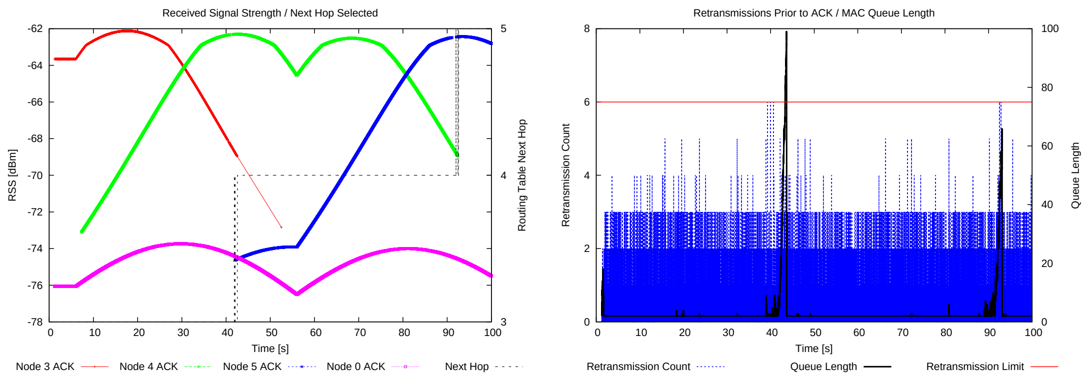


(c) UDP Packet Loss with ns-2 (Shadowing Model)

Figure 3.10: Packet Loss Comparison of ns-2 with ns2-MIRACLE (Single Mobile Transmitter)

As shown in table 3.3, a similar packet loss ratio and maximum delay were produced by ns2-MIRACLE and ns-2, when the shadowing model was implemented in the latter. With ns-2, packet loss was primarily a result of collisions, whereas with ns2-MIRACLE, packet errors and loss of routing packets occurred. However, it is evident from figure 3.11(b) that the buffering required as a result of continued use of a fading channel almost reached the queue limit. Additionally, retransmissions exceeded the threshold during periods of delayed handoff.

It can be seen from the results of this simulation that any motion of the transmitter, causing the previously used channel to fade and that moved it into range of another IN resulted in low goodput and maximal delay during this handoff process. The path discovery process of reactive ad hoc routing protocols, both intially and on rerouting resulted in a degradation in network performance. However, the results in this section highlight the time lag between the identification of the new



(a) Next Hop Selected

(b) Queue Length and Retransmission Count

Figure 3.11: Impact of Handoff on MAC and Routing Protocols

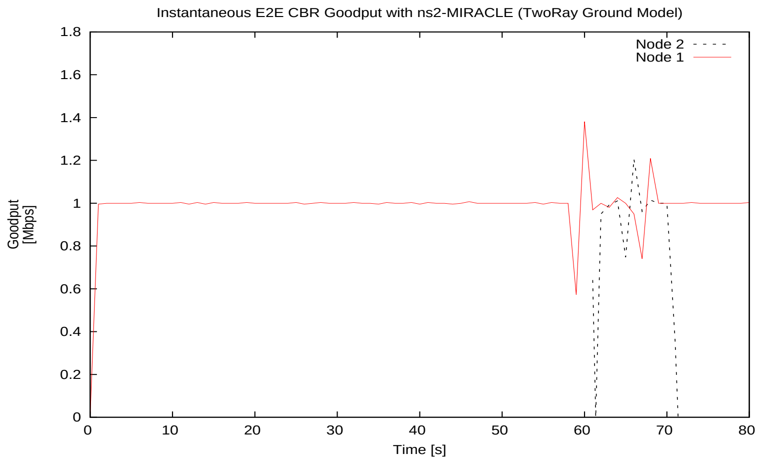
Table 3.3: Overall Performance with Single Mobile CBR Source

	ns-MIRACLE	ns-2	
	TwoRayGround Model	TwoRayGround Model	Shadowing Model
	N1 $\rightarrow$ N0	N1 $\rightarrow$ N0	N1 $\rightarrow$ N0
Packet Loss Ratio (%)	0.57	0.09	0.78
Maximum Delay (s)	0.37	0.04	0.33
Collision Count	5	146	162
Full IFQ Drop	0	0	0
Packet Errors	10707	0	0
AODV No Route	144	0	0

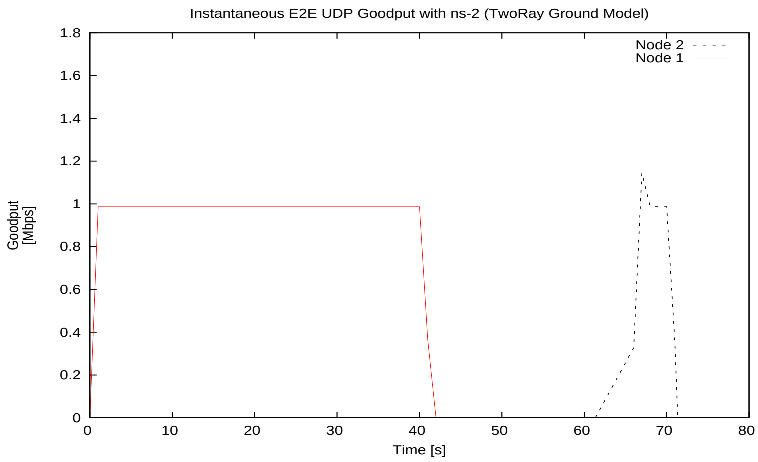
path and the transmission of packets along this path. There was an almost 20s lag between a transmitter coming into direct range of another IN that was in range of the receiver and packets being forwarded via this IN, which included repeated routing packet exchange. This lag occurred as both the better path available and the previous fading path were being interchangeably selected as the next hop. Avoiding the use of fading channels is a key factor in providing bounded delay and loss.

3.4.2 Hidden CBR Sources in MANET

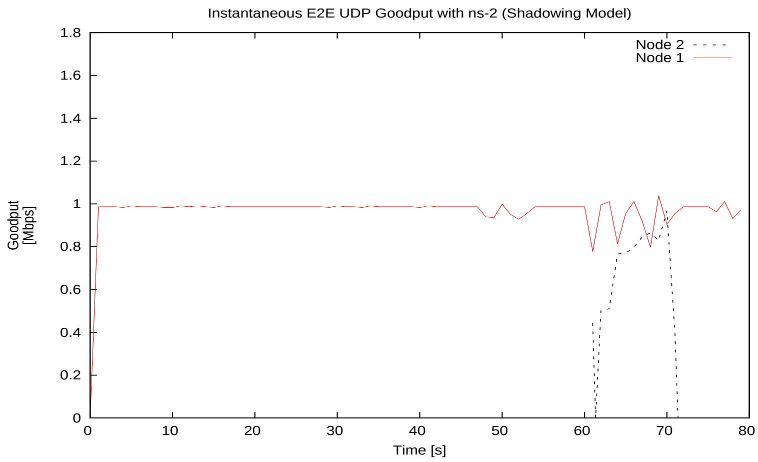
In order to assess the impact of hidden transmitters, a bus topology was utilised and the carrier sense threshold reduced to the the receive threshold. The simula-



(a) CBR Goodput with ns2-MIRACLE (TwoRay Ground Model)

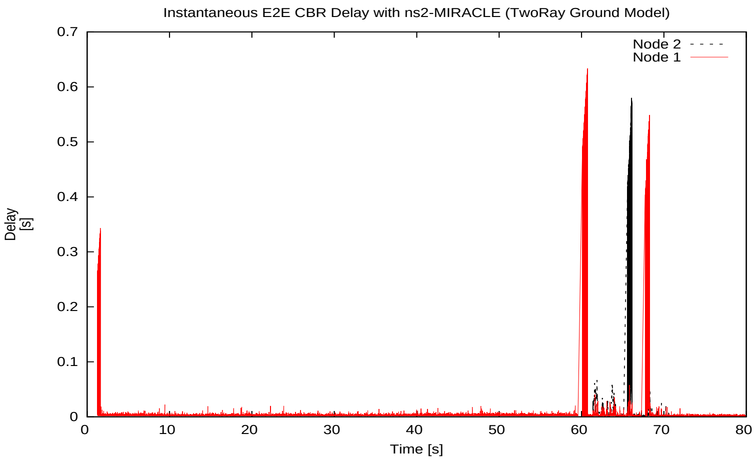


(b) UDP Goodput with ns-2 (TwoRay Ground Model)

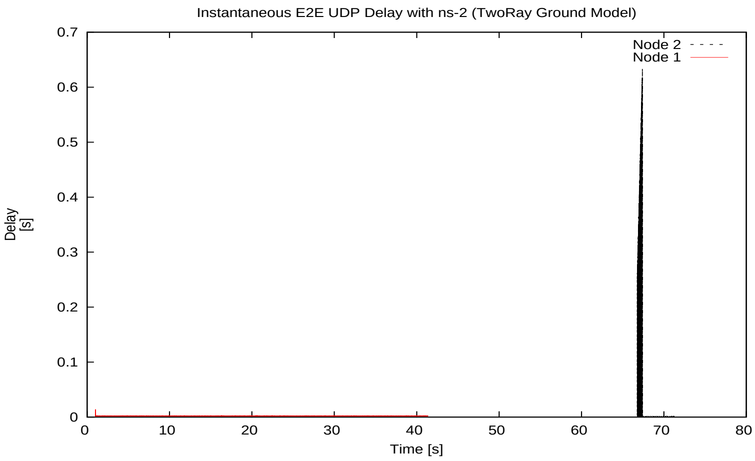


(c) UDP Goodput with ns-2 (Shadowing Model)

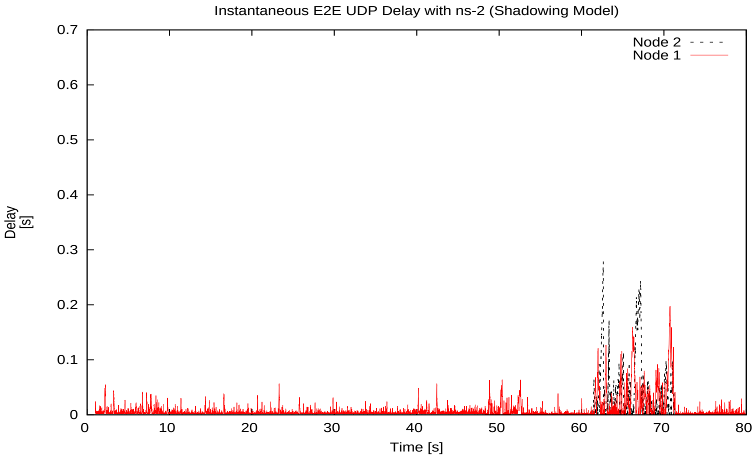
Figure 3.12: Goodput Comparison of ns-2 with ns2-MIRACLE (Two Mobile CBR Sources)



(a) CBR Delay with ns2-MIRACLE (TwoRay Ground Model)

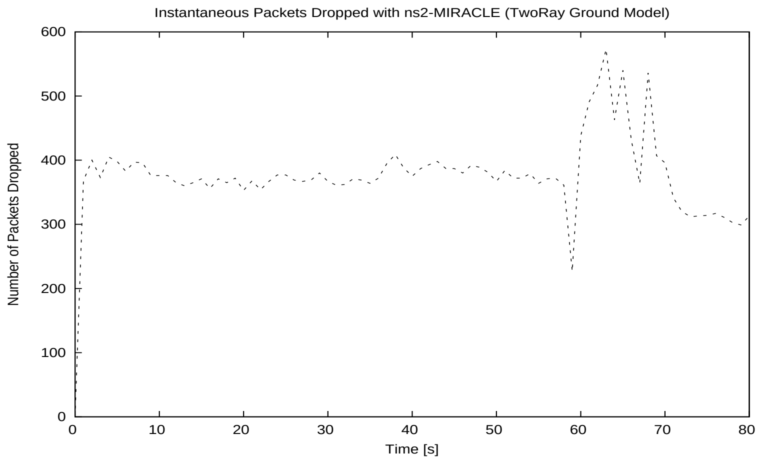


(b) UDP Delay with ns-2 (TwoRay Ground Model)

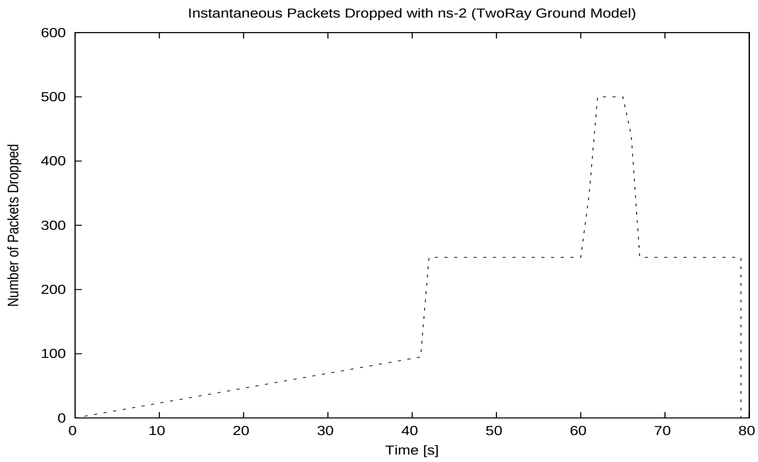


(c) UDP Delay with ns-2 (Shadowing Model)

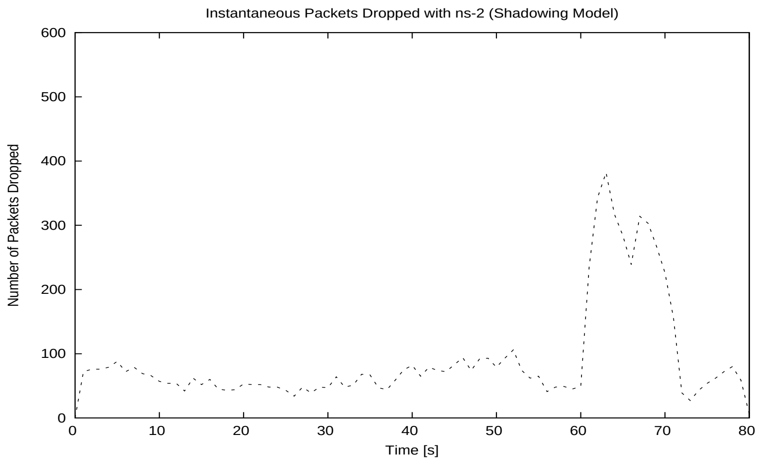
Figure 3.13: Delay Comparison of ns-2 with ns2-MIRACLE (Two Mobile CBR Sources)



(a) CBR Packet Loss with ns2-MIRACLE (TwoRay Ground Model)



(b) UDP Packet Loss with ns-2 (TwoRay Ground Model)



(c) UDP Packet Loss with ns-2 (Shadowing Model)

Figure 3.14: Packet Loss Comparison of ns-2 with ns2-MIRACLE (Two Mobile CBR Sources)

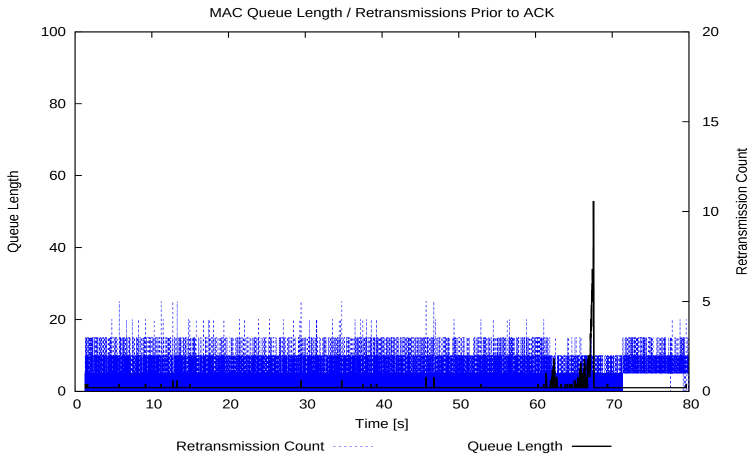


Figure 3.15: Impact of Hidden Node on Queue Length and Retransmission Count

Table 3.4: Overall Performance with Two Mobile CBR Sources

	ns-MIRACLE		ns-2			
	TwoRayGround Model		TwoRayGround Model		Shadowing Model	
	N1 → N0	N2 → N0	N1 → N0	N2 → N0	N1 → N0	N2 → N0
Packet Loss Ratio (%)	46.00	0.01	47.90	0.01	0.84	10.24
Maximum Delay (s)	0.63	0.59	0.64	0.01	0.21	0.29
Collision Count	33		0		913	
Full IFQ Drop	0		0		0	
Packet Errors	17024		0		0	
AODV No Route	205		23771		0	

tion added another transmitter (node 2) to the topology that joined and left the network during the course of the simulation schedule. Both nodes 1 and 2 travel towards the receiver, node 0, and are in range between 60-70s, during which time node 2 transmits a 1 Mbps CBR stream. At this point the two transmitters were equidistant from the receiver and hidden from each other.

Figure 3.12 shows that, using the ns-2 TwoRayGround model, the presence of hidden node 2 resulted in loss of connectivity for node 1, even though the traffic rate was as low as 1Mbps. In the remaining two simulations, the goodput of both transmitters on the MANET dropped rapidly. In ns2-MIRACLE the goodput of node 1 dropped by almost 50% when node 2 was transmitting.

Packets are increasingly buffered when the channel quality is poor, but the MAC rate increases when node 2 stops transmitting, resulting in goodput bursts as the queue is allowed to drain. As the traffic rate was still low, the queue did not overflow (figure 3.15). With ns-2's Shadowing model the main cause of packet loss was collisions, whereas with ns2-MIRACLE and the TwoRayGround model, packet errors were more prevalent. The performance results produced with ns2-MIRACLE do not vary widely from the more rigorously tested ns-2 simulator and demonstrate common performance anomalies under hidden terminal contention.

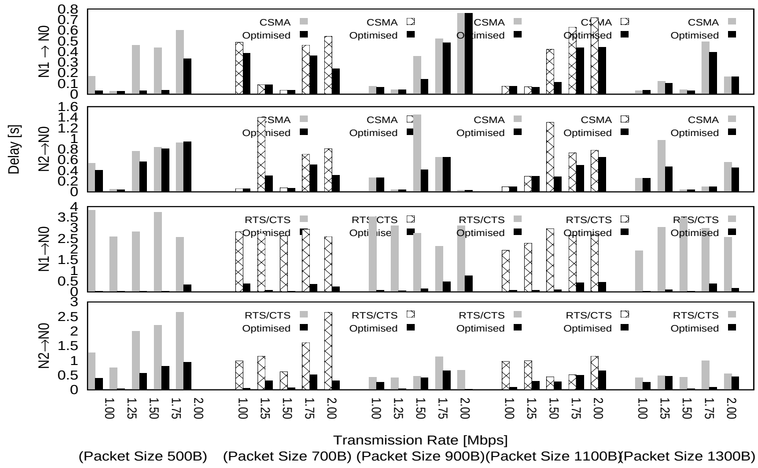
Transmitter contention and increased channel busy time results in high packet delay for both nodes, up to 0.6s when the TwoRayGround model is utilised in both simulators. Contention from the second transmitter caused a greater degradation in performance than seen from path setup and maintenance. Increasing the number of transmitters moving through the network and the corresponding freshness of routing information, ameliorated handoff degradation; reducing timespan of connection loss and lowering peak E2E delay related with handoff.

Investigation has highlighted that cross-layer awareness and control of contention is a promising approach for the provision of bounded packet loss, E2E delay and jitter. The influence of exposed transmitters has not been investigated in this section, modifying the performance of protocols which already respond to the presence of exposed transmitters is outside the scope of this project. The aim of providing an optimisation approach to detect and respond to the presence of a hidden transmitter, outlined in Chapter 1, is therefore extended in the next section where alternative optimisation approaches have been compared to identify the most appropriate.

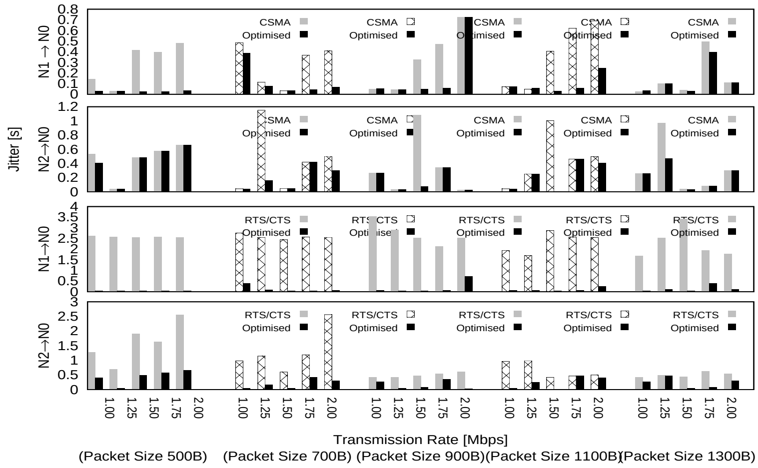
3.5 Comparison of Approaches to Contention Control

In order to identify a beneficial optimisation approach that can alleviate shared channel contention, the effect on performance of reducing transmitted load following detection of a hidden transmitter was investigated. To compare the potential approaches to load reduction, approaches considered were: reduction of load and packet transmission rate (TR) ameliorated by increase in packet size (optimisations 1-10) and maintenance of load through reduction of transmission rate traded off with increase in packet size (optimisations 11-14). Given the difference in interactions between RT sources with varied traffic load, each of these approaches were tested with two hidden CBR sources and packet sizes from 500-1300B and transmission rates of 1-2Mbps. A performance improvement with both variations in settings would be desirable.

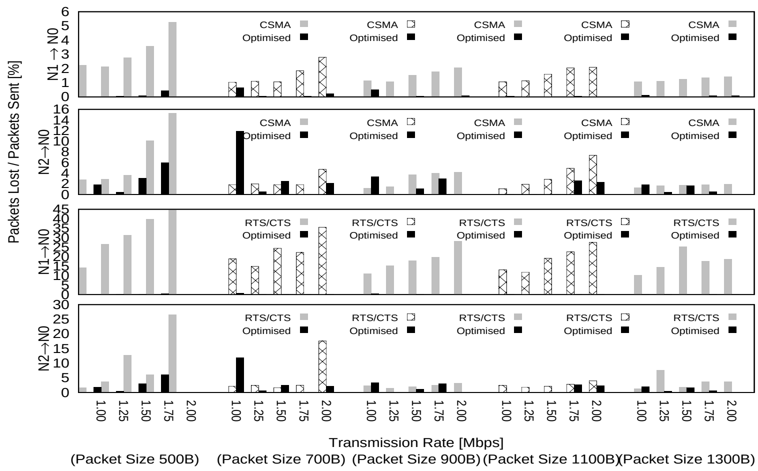
For the hidden node simulations, the 14 optimisation approaches (shown in



(a) Maximum E2E CBR Delay Under Contention

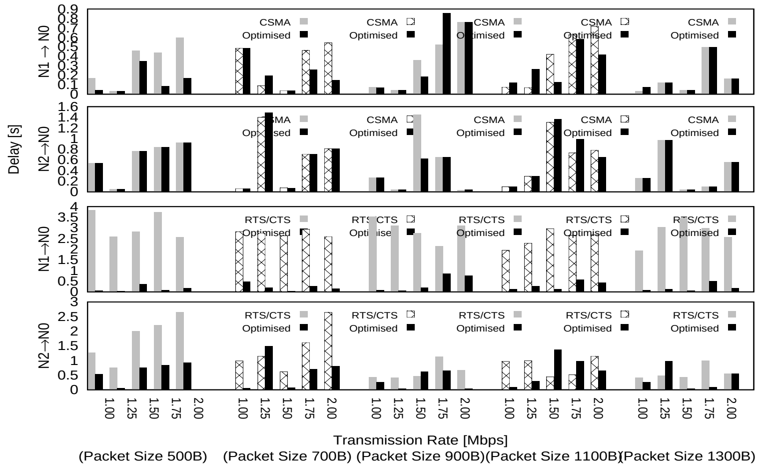


(b) Maximum E2E CBR Jitter Under Contention

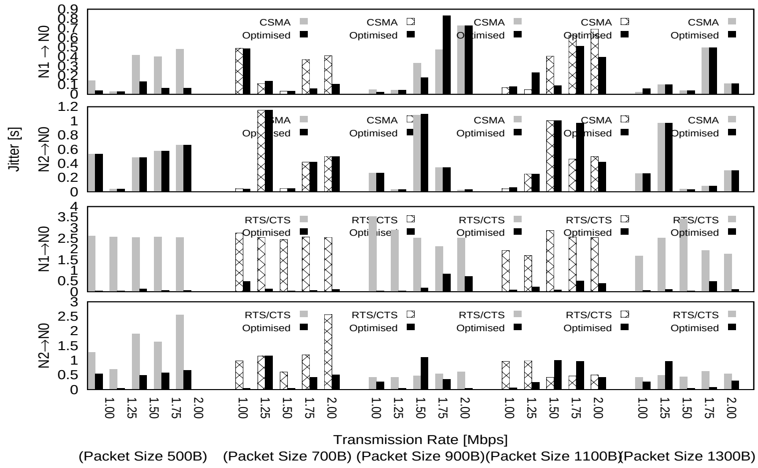


(c) Overall CBR Packet Loss Ratio

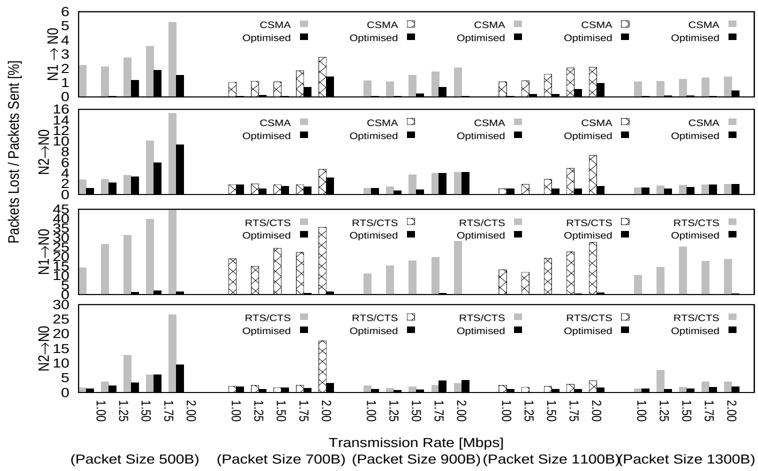
Figure 3.16: Overall Performance Comparison (Optimisation 1)



(a) Maximum E2E CBR Delay Under Contention



(b) Maximum E2E CBR Jitter Under Contention



(c) Overall CBR Packet Loss Ratio

Figure 3.17: Overall Performance Comparison (Optimisation 2)

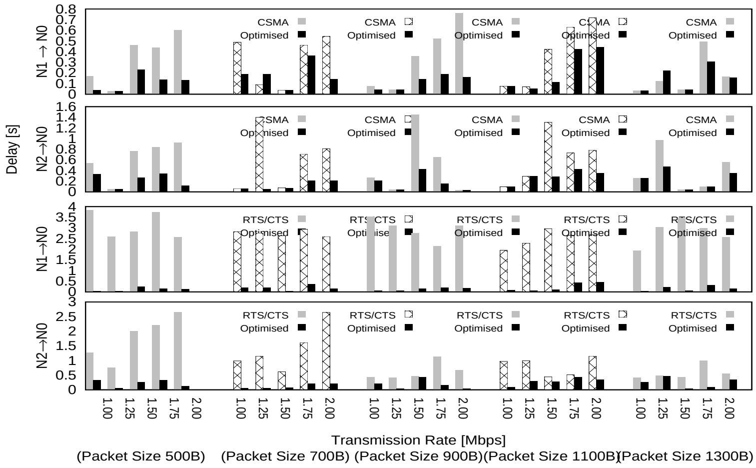
Table 3.5: Investigation of Load Reduction Optimisation Scenarios

Optimisation	PS Increase (%)	*TR* Decrease (%)	Load Reduction (%)	Optimisation	PS Increase (%)	*TR* Decrease (%)	Load Reduction (%)
1	200	25	50	**8**	150	35	30
2	200	30	40	**9**	150	40	20
3	200	35	30	**10**	150	45	10
4	200	40	20	**11**	200	50	0
5	200	45	10	**12**	250	40	0
6	150	25	50	**13**	330	30	0
7	150	30	40	**14**	500	20	0

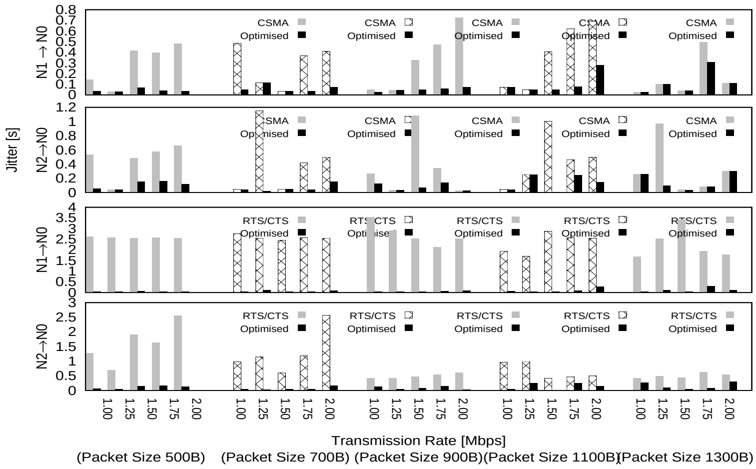
table 3.5) were implemented in all transmitting nodes in the simulation to investigate the impact on performance. Network performance for the optimisation approaches was compared to original performance when both CSMA and RTS/CTS were implemented. The goal of reducing the application transmission rate is to lower contention for the shared channel and related packet losses, without the requirement of excessive control packet handshaking. As fewer packets are transmitted per second by the application layer this reduces the backlog of packets in the buffer, and the frequency of random backoffs, collisions and interference corrupted packets. As fewer packets are lost and fewer retransmissions required, E2E latency and jitter are also reduced. Increasing the data packet size also reduces collision probability when a constant datarate is used. As nodes compete less frequently for channel access with larger packets, they also backoff less frequently and the interface queue backlog is reduced.

The three optimisation approaches providing the best performance were 1-3, where total load was reduced by 50, 40 and then 30%, respectively, therefore these are investigated in detail here. When CSMA alone was implemented, most packet loss in the scenario occurred as a result of packet errors at the receiver (node 0) and collisions at the two transmitters (nodes 1 and 2), as shown for a packet size of 500B in table 3.6. These events were reduced by both RTS/CTS implementation and Optimisation approaches 1-3 (table 3.7). However, while collisions were reduced to zero in almost all cases with RTS/CTS, this was at the expense of interference with routing packets and extended packet buffering. Therefore, losses due to buffer overflow and route errors resulted.

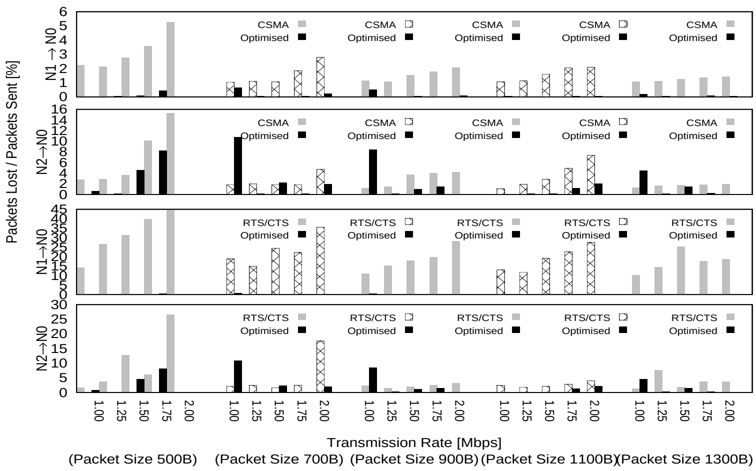
Corresponding to the packet error reductions, the packet loss ratio for all transmitters was reduced under most traffic configurations tested with optimisations 1-3 (figures 3.16–3.18). Optimisation approach 2, provided the greatest performance improvement when compared to both RTS/CTS and CSMA, for packet loss ratio alone. However, figure 3.18 demonstrates that delay was bounded when compared to the two traditional 802.11 implementations with the third approach. This resulted in maximum delay and jitter reduction for both transmitters under all tested traffic configurations. Maximum delay and jitter were not consistently



(a) Maximum E2E CBR Delay Under Contention

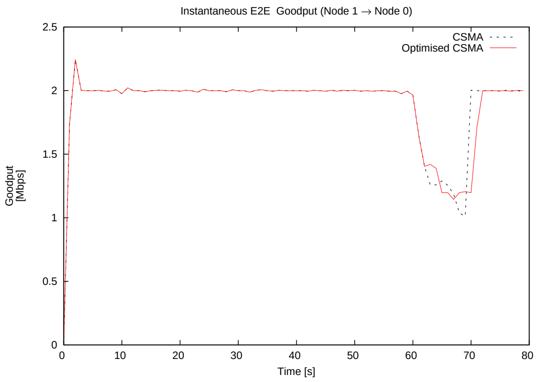


(b) Maximum E2E CBR Jitter Under Contention

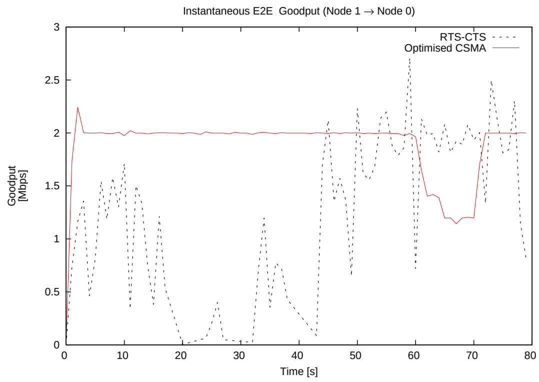


(c) Overall CBR Packet Loss Ratio

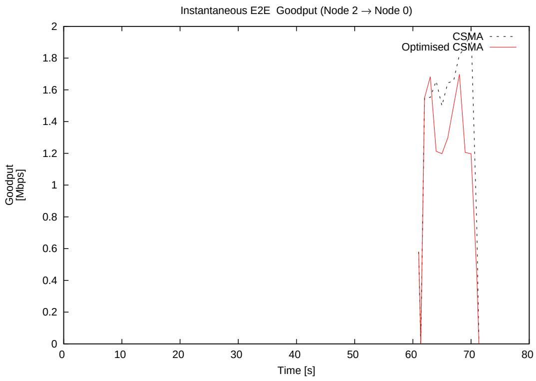
Figure 3.18: Overall Performance Comparison (Optimisation 3)



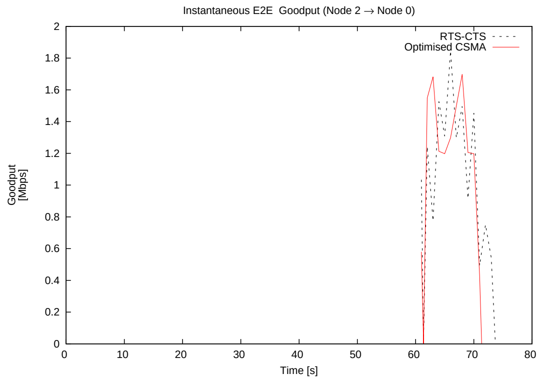
(a) CSMA (N1 → N0)



(b) RTS/CTS (N1 → N0)



(c) CSMA (N2 → N0)

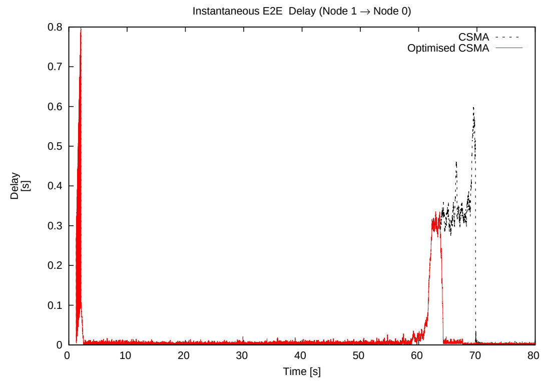


(d) RTS/CTS (N2 → N0)

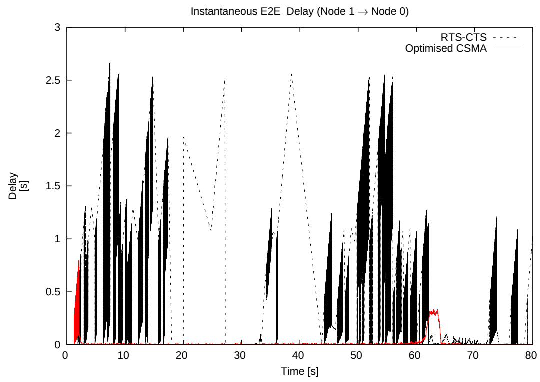
Figure 3.19: Instantaneous E2E CBR Goodput Comparison (Optimisation 3, TR=2Mbps, Packet Size 500B)

reduced in the other optimisations.

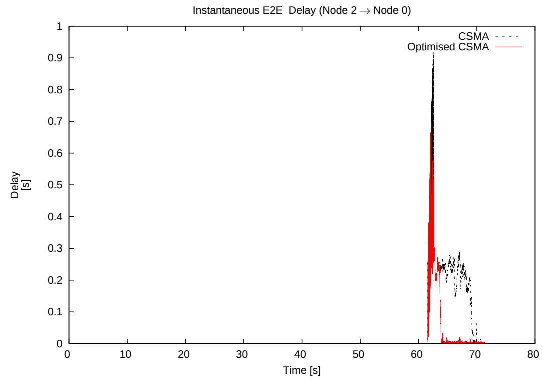
The two CBR sources contended for the channel after 60s into the simulation time, causing goodput to drop noticeably when unoptimised CSMA was implemented (figure 3.19). Increased packet buffering and repeated collision related backoffs also impact on E2E delay. When the two nodes were hidden the lack of collision or congestion control resulted in greedy and oblivious transmissions of packets. In comparison to CSMA, goodput, or the number of packets successfuly transmitted, was not significantly reduced by the 30% reduction in load implemented by optimisation 3. While the MAC layer may also detect channel noise, lowering the frame rate, this is compensated for by the use of larger frames. However, the result of lowering the application traffic rate during this period resulted in lower pressure on the IFQ and reduced contention for the shared receiver. This resulted in significantly lowered delay once the IFQ was able to empty (figure 3.20). In constrast, RTS/CTS flooding resulted in drastically lowered goodput and peak



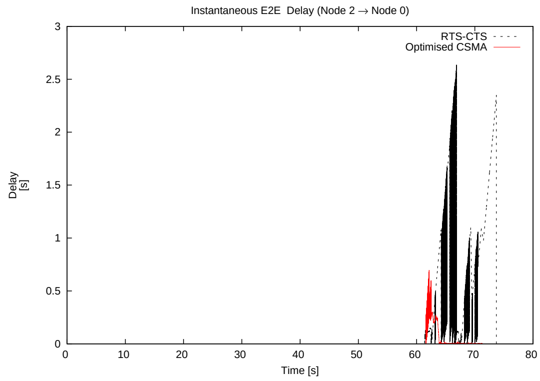
(a) CSMA (N1 → N0)



(b) RTS/CTS (N1 → N0)



(c) CSMA (N2 → N0)



(d) RTS/CTS (N2 → N0)

Figure 3.20: Instantaneous E2E Delay Comparison (Optimisation 3, TR=2Mbps, Packet Size 500B)

delay of 2s. This was particularly as a result of the interaction between RTS/CTS and ad hoc routing control packets, which impeded the maintenance of the E2E path.

As the load control is only tested in these simulations, and not based on any approach for detection of the hidden transmitters, it does not necessarily conincide precisely with the highest E2E delay. Additionally, node 2 only began transmitting when in range of the receiver, this resulted in a high delay as routing path setup occurred simmulaneously. Correspondingly, figure 3.20(c) shows that the load reduction outside of path setup is capable of providing bounded delay guarantees.

Table 3.6: CSMA and RTS/CTS: Selected Packet Drop Comparison (Packet Size = 500B)

CSMA		No	Packet	IFQ	Collision
TR	PS	Route	Errors	Full	Count
1	500	67	8016	0	67
1.25	500	7	9194	0	38
1.5	500	99	10744	0	185
1.75	500	99	14260	0	0
2	500	99	13948	0	486

RTS/CTS		No	Packet	IFQ	Collision
TR	PS	Route	Errors	Full	Count
1	500	1136	5468	709	0
1.25	500	2615	5590	3080	0
1.5	500	2758	6127	5876	0
1.75	500	4065	6098	5233	1
2	500	7007	6180	10374	0

Table 3.7: Optimisation Approaches: Selected Packet Drop Comparison (Packet Size = 500B)

Opt. 1		No	Packet	IFQ	Collision
TR	PS	Route	Errors	Full	Count
1	500	20	7405	0	30
1.25	500	7	8601	0	36
1.5	500	5	9156	0	102
1.75	500	3	9445	0	191
2	500	7	9345	0	301

Opt. 2		No	Packet	IFQ	Collision
TR	PS	Route	Errors	Full	Count
1	500	23	6578	0	33
1.25	500	7	7776	0	35
1.5	500	98	7368	0	132
1.75	500	99	7694	0	237
2	500	99	7338	0	347

Opt. 3		No	Packet	IFQ	Collision
TR	PS	Route	Errors	Full	Count
1	500	21	6508	0	29
1.25	500	7	6615	0	35
1.5	500	11	7183	0	45
1.75	500	8	7485	0	51
2	500	8	3251	0	66

3.6 Summary and Discussion

This section has conducted a detailed analysis of the performance of RT applications in MANETs and laid the groundwork for testing the cross-layer middleware approach. From this analysis it is evident that RT performance is, in a best-case scenario, subject to wireless interference, causing bit errors and slowing E2E path maintenance and resulting in increased queueing backlogs. In the worst-case, network dynamics as a result of non-detection of contention for forwarding nodes and suboptimal link selection result in reduced RT performance. The QoS metrics used to evaluate the network performance in the simulations were E2E throughput, latency, jitter and packet loss ratio. These results have also demonstrated that ns-2 and ns2-MIRACLE provide a similar pattern of MANET performance in the same topology configurations. ns2-MIRACLE uses more complex propagation models for the simulation of environmental conditions in order to simulate channel interference and noise.

The horizontal handoff investigation demonstrated that the path discovery phase of rerouting has a strong influence on packet delay, to a greater degree with successive reroutes. At rerouting an increased number of control packets circulate the network, some broadcast to all nodes in range and other unicast packets returned to transmitting nodes. Additionally, in-use but fading channels are often selected preferentially over more robust channels, based on the receipt of MAC acknowledgements. E2E delay is therefore increased due to data losses, repeated backoffs and retransmissions and jitter as a result of variable queuing times along multiple paths. Additionally, MAC and network layer control packets on these multiple paths compete for bandwidth. While improvement of the routing protocol an alternative. However, this would require a more sophisticated network monitoring and, without MAC layer information, path selection would not be robust.

The contention control investigation demonstrated that when transmitters are hidden from each other there is room to improve the responsiveness of MANET protocols to their interactions. Two hidden transmitters in a MANET, if sharing the same forwarding IN, will gradually experience reduced performance. Both the transmission rate and size of packets sent by each transmitter onto a single channel influence channel access opportunities and interference. As contention control does not have an innate solution, a metaheuristic approach must be taken. Therefore, comparison between alternative optimisation approaches was investigated in Section 3.5. The results therein demonstrated that tuning both transmission rate and packet size when multiple sources contend for a channel is an approach capable of bounding maximum delay, jitter and packet loss ratio for all transmitters.

The factors that are identified to influence variable network conditions exist at multiple layers of the network stack and, individually, are difficult to control. These include number of transmitters, node speed and topology, traffic load and configuration and specific application requirements. Introducing dynamic responsiveness of higher layer protocols to lower layer information, such as channel quality and usage information, can improve network performance. Through cross-layer signalling this information can be used efficiently and without functional modification of established wireless protocols, instead, creating a response to channel conditions. This reactive tuning can then manipulate the higher layers, reducing overheads and ensuring that delay and jitter requirements of RT flows are met. Therefore the key findings and outcomes of this analysis have been used to develop the architecture proposed in Chapter 4.

Chapter 4

Real-time Optimising Ad hoc Middleware Architecture

4.1 Introduction

This chapter describes the functionality and structure of, and parameters optimised by the Real-time Optimising Ad hoc Middleware (ROAM) architecture proposed by this project. The proposed architecture (figure 4.1) consists of ROAM, multiple layer-specific API and associated cross-layer messages for the abstraction of protocol parameters. ROAM is a middleware entity that manages the implementation of optimisers to tune protocols in order to reduce maximum E2E delay, packet loss and jitter. The purpose of this optimisation is to provide support to RT applications operating in MANETs.

The aim of the project has been to develop a performance improvement approach, appropriate to the support of both contemporary, established wireless protocols and safety-certified military or disaster-response protocols. In the latter, functionality of the protocols should not be modified as these have been extensively tested. Consequently, ROAM creates a response to lower layer conditions when parameter information is readily available to higher layer protocols, without modifying their original functions.

Application and network layer protocols under-perform in MANETs without recourse to information from lower layers that relates to stochastic variation in channel quality and contention. Chapter 3 investigated this problem in the specific situations of horizontal handoff and shared channel contention. In both situations this decline related to the non-communication of useful information: that a higher SINR channel is available in the former and that a hidden transmitter is present in the latter case. Lower layer information can particularly be used to avoid large increases in delay and jitter in these scenarios, that impact negatively on RT

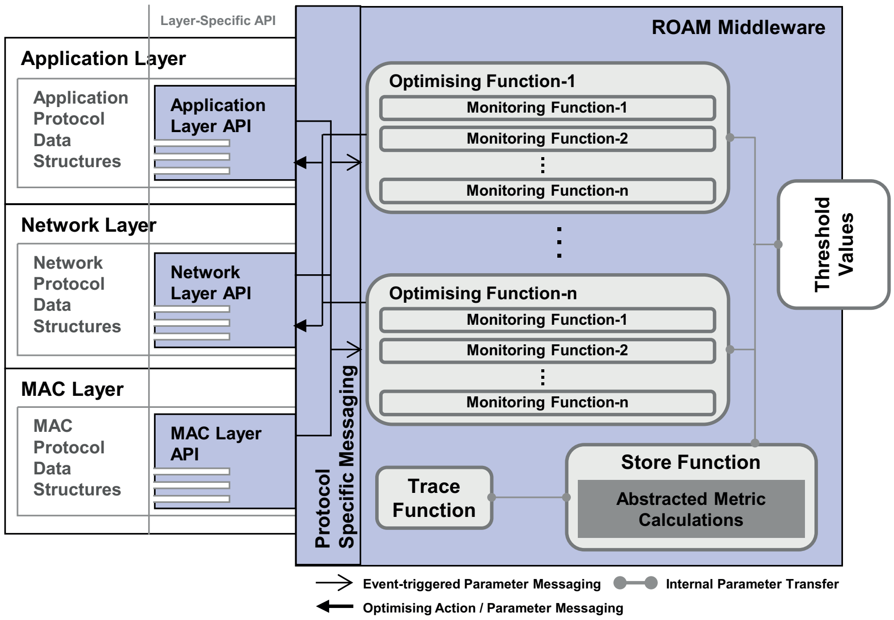


Figure 4.1: The ROAM architecture

applications with time and safety critical deadlines.

The implementation of the ROAM architecture in the ns2-MIRACLE simulator is detailed in this chapter. The middleware contains layer-specific optimisers, which depend on the transfer of protocol parameters by messages transferred across the API. While the structure and function of the middleware is not protocol specific, the optimisers are MANET protocol specific. This is because assumptions are made that reactive ad hoc protocols as well as MAC layer acknowledgement are utilised to set up paths and connections. The results of the ns-2 and ns2-MIRACLE MANET performance simulations in Chapter 3 have provided the foundations for design of the ROAM optimiser functionality.

The middleware has been designed to be scalable, supporting the addition of multiple optimisers. In particular, the API functions allow protocol data structures to read and write to these API, which are layer-specific but can be exported to any MANET protocol. ROAM therefore overhears lower layer information to improve performance, without the need to modify protocols that have been rigorously tested and certified. This provides a solution that is modular and reusable and that maintains protocol transparency, because the protocol stack and middleware function independently.

A second benefit of the ROAM design is that the optimisers and middleware structure are not interdependent. In terms of future usability, each can be modified without changing the other. Information does not pass directly between protocols, the sole interaction across the stack is via API through which the middleware abstracts and returns parameters to and from particular layers. The export of an

API to a protocol increases complexity of the network stack but provides beneficial performance improvement while allowing the protocols to function as normal. This is important in contexts where network components are safety certified and provides for the extensibility of ROAM to other ad hoc networking protocols.

The rest of the modular architecture is generic: cross-layer messaging, storage and trace functions that make up the rest of the ROAM structure are not dependant on interaction with any particular protocol or layer. The specific parameters accessed by ROAM are stipulated by the API function calls. When these reach a predefined threshold or their value changes, this value is abstracted across the interface. Abstracted parameters from the application, network and MAC layers are then passed to the related optimiser.

ROAM implements a local approach that optimises the protocol stack within a single node, on a distributed basis, thus the overheads associated with inter-nodal, or global messaging are avoided. The ROAM entity executes simultaneously with the protocol stack, to ensure reduced overheads of cross-layer communication and processing. Unlike previous cross-layer approaches such as ECLAIR or the Control Middleware Plane this cross-layer framework specifically supports RT protocols as well as RT QoS. This framework therefore proposes optimisation of a three-layer stack (application, network and MAC layers) for parameter abstraction and a two-layer stack (application and network layers) for the return of modified parameters rather than full protocol stack tuning. This is based on an understanding that efficient cross-layer tuning of the contention control and routing handoff processes can provide guaranteed QoS to heterogeneous RT devices.

4.2 Middleware, Messaging and API Design

4.2.1 Real-time Optimising Ad hoc Middleware (ROAM) Optimisers

Two optimisers have been developed to manage the use of lower layer information to improve overall ISRT performance. The first ensures that the highest quality link available is always in use, through tuned horizontal handoff. The second identifies the presence of hidden transmitters, optimising transmission settings to implement contention control. Increased delay and jitter, due to failed transmission, results when handoff to a higher quality link is not performed or a routing protocol switches back to a fading path. While repeated collisions and interference related errors result if a contending mobile transmitter is not detected. The scoping experiments carried out in Chapter 3 demonstrated the manner in which network performance diminishes in these situations. Each optimiser consists of multiple functions and OnEvent() callback functions. These callback functions respond when a parameter abstracted from a protocol layer arrives at a threshold value.

4.2.1.1 Horizontal Handoff Optimiser

As a transmitting node moves through a MANET it may use different INs in the network to forward packets as they come into and go out of range. Increased delay and jitter, due to transmission errors and collisions, results when handoff to a higher quality link is not performed or a routing protocol re-selects a fading path. Horizontal handoff without reference to channel conditions can result in localised increases in packet loss, jitter and delay as packets are transmitted over fading links. In receiving information from multiple layers, ROAM is therefore able to identify a fading link earlier than a routing protocol alone. ROAM can then monitor for better links and using information gathered from control packets received at multiple layers, prioritise paths. Finally, the routing protocol can be optimised to select the most appropriate path and ROAM can ensure that packets are not transmitted over a fading link.

The optimiser utilises five event-triggered, OnEvent() subfunctions. Figure 4.2 shows these subfunctions and indicates that the API utilise callback functions at each layer to abstract parameter values when these change at that layer. For

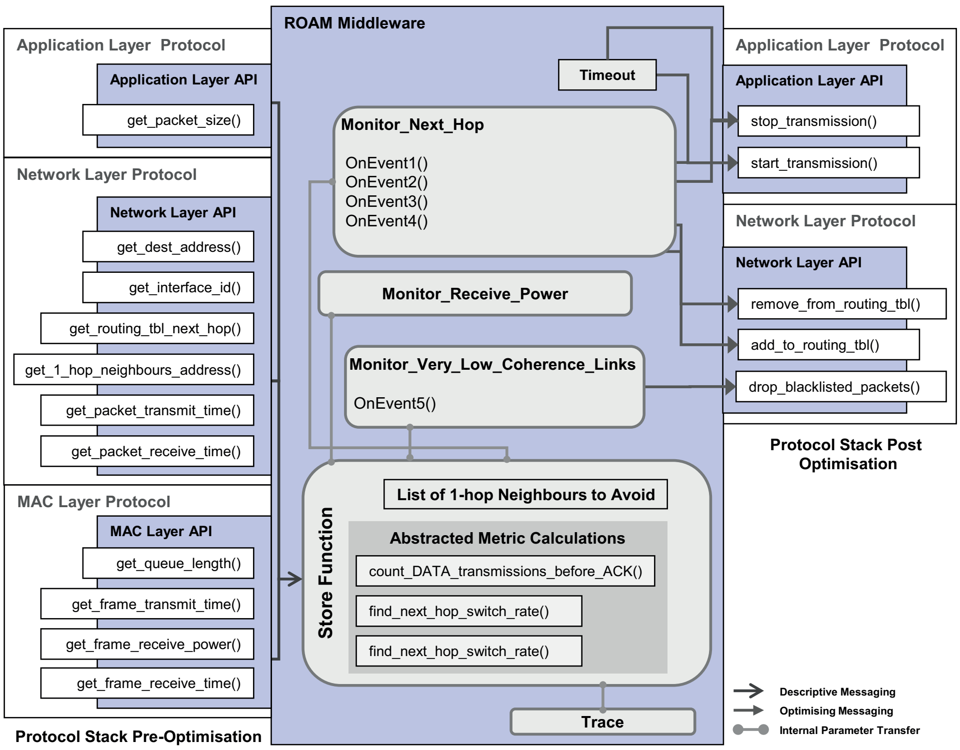


Figure 4.2: Optimised Horizontal Handoff with ROAM

example the get_packet_size() function will abstract the packet size from the application layer at the start of transmission and whenever this value changes at the application layer. The algorithm for tuning fast handoff to non-mobile nodes begins with the Monitor_Receive_Power function, which will continue to monitor packet receipt while ROAM is in use. The Monitor_Next_Hop function can be called by the Monitor_Receive_Power function and this commences a sequence of callback functions: OnEvent1() - OnEvent4(). Only the Monitor_Receive_Power() function runs continually. Thus at any stage this function can stop the sequence of callback functions, if handoff has already occurred. The algorithm runs as follows:

Monitor_Receive_Power
This function monitors the receive power, or received signal strength (RSS) of MAC control frames or higher layer control frames (seen by the MAC layer as data frames, but identified by the optimiser as smaller than the application packet size) that are received from neighbouring nodes. The RSS of control packets received from the current next hop (CNH), to which the MAC layer is sending frames, is continually monitored. If any other neighbouring nodes have a RSS above a threshold, the two remaining functions are called (Monitor_Next_Hop and Monitor_Very_Low_Coherence_Links) so that these neighbouring nodes are monitored by ROAM. The threshold is set at the RSS at which data packets are passed by the MAC protocol to the routing protocol, rather than dropped at the MAC layer (although they may be dropped at the network layer).

If at any point the RSS of control packets from the next hop fades and drops below the threshold, this node is assumed to be out of range. The series of OnEvent functions can add neighbouring nodes to a list of INs to avoid, therefore, this list is cleared as it is assumed that handoff has already occurred as the previous next hop is out of range.

Monitor_Next_Hop
This function is called by the Monitor_Receive_Power function if any other neighbouring nodes have a RSS above the aforementioned threshold. This commences the following sequence of callback functions:

- **OnEvent1()** monitors the next hop selected by the routing protocol and the MAC queue length. The OnEvent1() function monitors for the occurrence of one of the following two conditions: (1) The next hop switches more than a threshold number of times per second between two different neighbour nodes or (2) MAC queue length is increasing. In either case, ROAM takes the action to continue monitoring control packet RSS for both nodes. (The threshold number of next hop switches has been set to three, to ignore normal routing table changes.)

- **OnEvent2()** monitors for the occurrence of one of the following two conditions: (1) RSS on one path reduces and on the second path increases rapidly to within 5dBm of the fading next hop (if next hop switching occurred at OnEvent1()) or (2) RSS on the second path increases to more than or equal to the CNH (if MAC queue length was increasing at OnEvent1()). The intermediate node (IN) on the fading path is then removed from the routing table and added as a parameter to the routing protocol blacklist. This blacklist is a list of broken links that the reactive routing protocol avoids. Network control packets from the fading IN are temporarily dropped at the routing interface.

- **OnEvent3()** triggers the application layer API to call the stop_ transmission() event within the application. This causes the application transmission to be paused, as an imminent handoff is required and OnEvent3() simmultaneously begins a timer. Data acknowledgement is used by ad hoc routing protocols to maintain an E2E path, therefore temporarily pausing the application is necessary to ensure the old path is not restored during handoff. The OnEvent3() function then monitors abstracted information on intercepted and transmitted data, ACK and network layer control packets. The application will be triggered to restart (by calling a start_transmission() event) after either one of the following conditions is met: (1) 2s have elapsed on the timer or (2) a RREP from a neighbour has passed from the MAC layer and arrived at the network layer.

- **OnEvent4()** monitors the comparative RSS (abstracted at the MAC layer) of the previously blacklisted node and the node from which a RREP has been received. If the RSS of this RREP is more than or equal to the fading path the more robust path is added to the routing table.

Monitor_Very_Low_Coherence_Links

This function is called by the Monitor_Receive_ Power function, if any other neighbouring nodes have a RSS above the aforementioned threshold, and calls the OnEvent5() callback function:

- **OnEvent5()**: Monitors the rate at which the retransmission limit is exceeded and the rate of change of control packet RSS for each neighbouring node. The retransmission limit is the maximum number of possible retransmissions of a frame, set by the MAC layer. The rate at which control packet RSS rises as the node comes into range is predicted to be the rate at which the channel will fade as the node goes out of range. This is because nodes that are moving at a high relative velocity are assumed to maintain their speed, if not their direction of motion. The OnEvent5() function monitors for both of the following two conditions to be met: (1) retransmission limit is exceeded at an increasing rate and (2) rate of change in RSS for a neighbouring node is more than that of the path to the CNH in use, as notified by the Store function. If all of these these conditions are met, this node is removed from the routing table and added to the blacklist. This causes the routing interface to drop control packets from these INs. If these attributes no longer appear, as node speed has changed, this link previously identified as a low-coherence link is removed from the routing protocol blacklist by ROAM.

4.2.1.2 Contention Control Optimiser

If more than one transmitter uses a network, at certain locations multiple transmitters may share the same forwarding node. Repeated collisions and interference related errors result if a contending node is not detected. While a hidden transmitter cannot be identified directly, distributed responsiveness to changes in channel availability, path delay, queue length and overheard ACKs enable more efficient use of available resources. In receiving this information from multiple layers, ROAM is able to identify and respond to the presence of a hidden transmitter. ROAM improves RT performance for all flows by optimising application transmission settings: reducing application transmission rate and minimally increasing packet size to avoid randomly throttling flows. As neither transmitter may move away from the IN and handoff to an alternative path is not always possible, each transmitter must independently optimise its use of the shared medium to increase network performance. The aim of this optimiser is to provide distributed load control when more than one transmitter is using the same IN to relay traffic.

The ROAM Contention Control Optimiser utilises three functions: Monitor_RTS-CTS, Find_Hidden_Node and Find_Exposed_Node. These will also access abstracted parameters from the Store function. The Find_Exposed_Node function uses two event-triggered, OnEvent() callback subfunctions. The Find_Hidden_Node function uses three callback subfunctions. Figure 4.3 shows these as well as the API callback functions at each layer, used to abstract parameter values when these change at that layer. For example the get_dest_address() function will abstract the destination of data packets at the network layer at the start of transmission and whenever this value changes. The algorithm for tuning contention control in response to the presence of a hidden terminal begins with

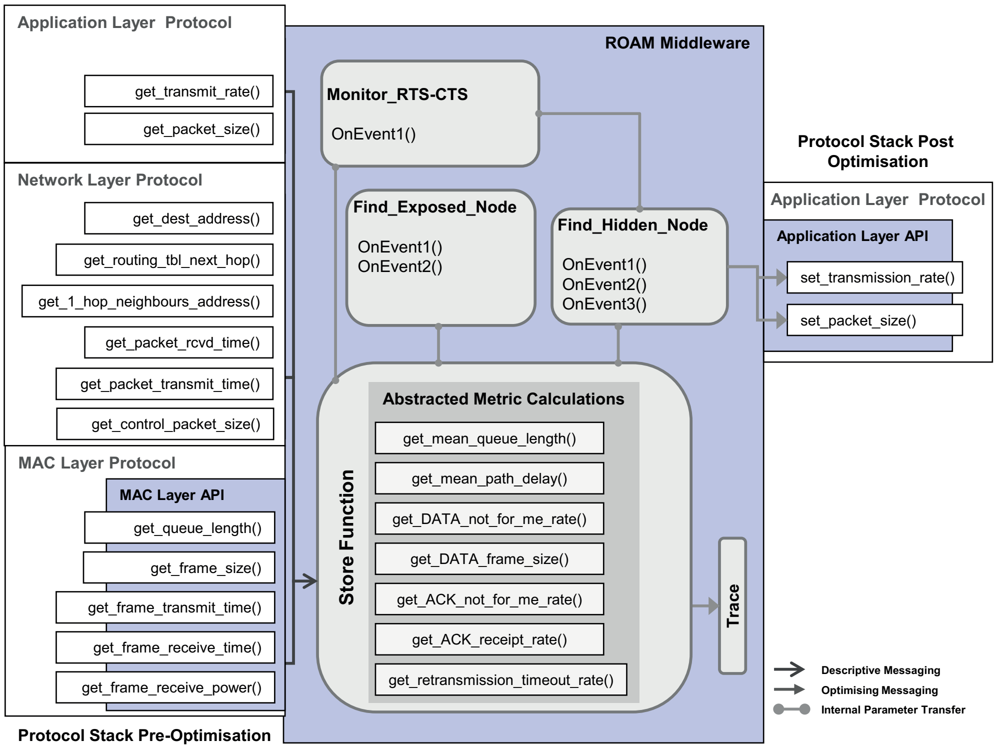


Figure 4.3: Optimised Contention Control with ROAM

the Find_Hidden_Node function and will continue unless obstructed by the Monitor_RTS-CTS or Find_Exposed_Node functions, which will continually monitor for RTS/CTS packets and exposed terminals while ROAM is in use. This algorithm runs as follows:

Monitor_RTS-CTS

This function runs continually and monitors for transmission of RTS packets to evaluate whether RTS/CTS handshaking is in use. If these are detected the Find_Hidden_Node and Find_Exposed_Node functions and related callback functions cannot or can no longer be called and all original application transmission settings are restored.

Find_Hidden_Node

This function runs continually, unless obstructed by the Monitor_RTS-CTS or Find_Exposed_Node functions. This commences the following sequence of callback functions:

- **OnEvent1()** monitors the rates at which MAC retransmission limits are exceeded, of MAC data acknowledgement receipt for the ROAM node and of ACKs from neighbouring nodes. It also triggers the Store function to record mean MAC queue length (Q_{MAC}) and mean path delay (D_p), which will be described in the next function. The maximum MAC frame retransmission limit is set by the MAC layer protocol. The OnEvent1() function monitors for two conditions to be met: (1) the retransmission limit exceeded rate is higher than the data acknowledgement rate for the ROAM node (2) the MAC layer intercepts an increasing number of ACK packets per second from the current next hop (CNH) that are intended for another node (provided that the RSS of these ACKs is not fading). ACKs received at the ROAM node, but intended for a neighbour will be referred to as an ACK_{NFM}: ACKs Not For Me. If all of these aforementioned conditions are met, the ACK_{NFM} count is recorded.

- **OnEvent2()** monitors mean MAC queue length (Q_{MAC}) and mean path delay (D_p). D_p is the delay between requesting and responsive control packet pairs, such as routing requests and replies or MAC layer DATA and ACK. The OnEvent2() function monitors for both of the following two conditions to be met: (1) the D_p and Q_{MAC} have increased more times than they have decreased, since the OnEvent1() subfunction was called and (2) the rate of ACK_{NFM} is higher than the rate of ACKs intended for the ROAM node. If all of these conditions are met, the presence of a hidden contending node is identified and ROAM tunes the application layer to constrain transmission rate by 20% and increase the packet size by 5%. These tuning parameters have been set according to the performance investigations undertaken in Chapter 3, with the aim of reducing transmitted load without throttling the application.
- **OnEvent3()** continues to monitor mean MAC queue length (Q_{MAC}) and mean path delay (D_p). The OnEvent3() function monitors for any of these three conditions to be met: (1) the Q_{MAC} is reducing more than increasing or (2) the D_p is reducing more than increasing or (3) the rate of ACK_{NFM} are no longer received at a rate higher than the rate of CNH ACKs. If any of these conditions are met, the original application transmission settings are restored.

Find_Exposed_Node
This function runs continually and consists of the following callback functions, which run concurrently:

- **OnEvent1()** monitors the RSS of data packets intercepted at the MAC layer from neighbouring nodes and the receipt of ACK_{NFM}. These data packets are identified as not being passed to the routing protocol, which are therefore not routing control packets. The OnEvent1() function monitors for two conditions to be met in sequence: (1) data packets are intercepted from another node that are within 5% RSS of any packets received from the CNH and (2) ACK_{NFM} intercepted are intended for this node (not the CNH). If these conditions are met, an exposed transmitter is identified. All rates and counts recorded by the Store function are then zeroed and the exposed node is monitored, but the Find_Hidden_Node function is obstructed: this function and its related callback functions cannot then be called.

- **OnEvent2()** monitors the RSS of data packets intercepted at the MAC layer from neighbouring nodes and the receipt of ACK_{NFM}. The OnEvent2() function monitors for the following two conditions to be met: (1) data packets are intercepted from another node at less than 5% of the RSS of packets received from the CNH (this node is almost out of range and an exposed transmitter is not identified) and (2) ACK_{NFM} are not intercepted (the node is not sharing the same E2E path). OnEvent2() allows the Find_Hidden_Node function to be called and no longer obstructed: this function and its related callback functions can now be called.

4.2.1.3 Storage and Trace Functions

The Store and Trace functions within ROAM distribute, prioritise and output to a tracefile the values of parameters received by the middleware. The Store function enables ROAM to keep received parameter values for short periods and access these as required. The Trace function records the abstracted parameters generated by the middleware as well as handoff or contention events identified, for performance monitoring purposes. When a monitored value exceeds a threshold specified by ROAM, this is also recorded as an event by the Trace function.

All of the ROAM optimiser functions can access the list of MAC to IP address translations from the Store function. When control packets or data packets are received at the MAC layer, the senders MAC address is stored by ROAM. When these packets are subsequently received at the network layer, ROAM is then able to translate the MAC address to the IP address of this sender. As ACK and DATA packets are received by the MAC layer, the Store() function in ROAM calculates the number of retransmissions prior to an ACK receipt.

These functions, alongside the two optimisers, make up the monolithic middleware. They have not been abstracted to a separate module, as in other cross-layer implementations discussed in Chapter 2, in order to minimise the time required to retrieve parameter values. When an Interface Message is received from the API, ROAM checks the metric type contained and if appropriate sends this value to the Store function. When the ROAM optimiser requires a tuned metric value to be sent to a protocol layer, in an Optimisation Message, it will directly access the prioritised value stored. For example after a horizontal handoff event is identified by the middleware as imminent, the best next hop is identified by the middleware. This parameter is taken from the Store function which receives the details of neighbours in range and prioritises these according to channel quality information.

4.2.2 Cross-layer Messaging and API

ROAM uses cross-layer messages to request and return specific parameter values to and from protocol layers via layer specific API. These two parts are essential to the structural goal of ROAM to ensure minimal interference with the traditional network stack structure and non-modification of protocol functionality. While parameters pass to and from the middlware via the API, there are no direct linkages between protocols. Three types of message are used by the middleware to manage information abstraction and optimisation of the network:

- *Query Message*, a message broadcast by ROAM to the entire network stack, used to locate participating layers (containing the API) in the protocol stack. The receipt of a QUERYMSG initiates parameter monitoring at that layer and event-triggered messaging when a parameter exceeds a preset threshold. The API at each participating layer then automatically returns the synchronous Query Message to the middleware after adding its own layer identifier to the message.

- *Interface Message* is a generic, unicast message that can be sent by any protocol layer API to the middleware. Sending is triggered when either a specified parameter reaches a threshold value or changes from its previous value. The Interface Message for each parameter type is identified by two values, the first is a unique identifier by which the middleware, on receipt, can recognise the parameter and utilise its value. The second identifier of the parameter is the sending layer identifier. These messages are asynchronous as the message is not automatically returned by the receiving module.

- *Optimisation Message* is a unicast messages sent by ROAM following the processing of parameters that identify optimisation is required. An appropriate optimised parameter value is returned to a specific layer in this message in order to improve network performance. For example, if horizontal handoff is identified as necessary, ROAM will transmit the best next hop to the network layer. The aim of sending this message is to indirectly restore the parameters values that triggered optimisation to outside the preset thresholds.

Each of these messages has a role in the optimisation process and can be potentially sent by any participating module. However, if an API receives a message intended for a different receiving entity, the message is dropped. Messages are generic and, having the format shown in figure 4.4, can be exported with the API to any new protocol.

```
------------------------------------------------------------------------
|| SourceId | MetricId | MType | MValueStruct | DestType | DestId ||
------------------------------------------------------------------------
```

Figure 4.4: ROAM Cross-layer Message Format

The `SourceId` contains the identifier of the sending entity (protocol layer or ROAM) and the `MetricId` is used by the receiving entity to identify the parameter contained in the message. `MType` identifies the message as one of the aforementioned three message types. The Message value structure (`MValueStruct`) enables the transfer of different subtypes of tunable parameters (further discussed in Section 4.3).

API, using the aforementioned messaging and the callback functions they contain, can access and abstract parameters contained within protocol data structures to ROAM. While the API are generic and can be exported to any layer, the callback functions they contain are layer specific. The callback functions used by each of the ROAM optimisers are indicated in figures 4.2 and 4.3. When a layer API receives a cross-layer message it checks that the destination ID (DestId) matches its own ID. The API accesses parameters specified in its callback functions and if a parameter changes or reaches a threshold value its value is passed in a message to the middleware. The API callback functions therefore handle monitoring and messaging of parameter values. Messages are not sent to other layers in the stack, but directly to the middleware, therefore each layer interface only links that layer and the middleware.

4.3 ROAM Parameters

4.3.1 Horizontal Handoff Parameters

The parameters directly tuned by ROAM are:

- **One-hop Neighbour IP address:** received from the Network layer API, from the routing table, in order to optimise early handoff ROAM will add and remove one-hop neighbours to the routing table.

- **Application Layer Transmission Start Time:** this is tuned to prevent use of a fading channel while ROAM optimises early handoff.

- **Application Layer Transmission Stop Time:** this is tuned to prevent use of a fading channel while ROAM optimises early handoff.

- **Control Packet Receive Time:** this is tuned by ROAM at the Routing Interface to prevent use of a fading channel while ROAM optimises early handoff.

The parameters indirectly tuned by the ROAM middleware are:

- **MAC Layer Queue Length:** provides an indication of the queue utilisation and correspondingly the traffic loading on the channel. If the queue arrival rate is higher than the transmission rate this parameter increases until packet dropping occurs.

- **Number of One Hop Neighbours:** collected at the Network layer from the routing table, this is a count of the number of neighbours within one hop of the node for in the interval since the count changed.

The descriptive parameters that are used but not tuned by the ROAM middleware are:

- **IP Address of Neighbour Added:** is received from the Network layer API on an addition to the routing table. For AODV this is based on the receipt of a control packet from a neighbour not currently stored in the routing table.

- **One-hop Neighbours:** received from the Network layer API in the routing table

- **Application Data Packet Size:** is the size of a packet at the Application layer.

- **RREQ Receive Time:** is the time at which an RREQ is received from a neighbouring node and is received from the Network layer API. This is used in the identification of nodes within transmission range.

- **RREQ Transmission Time:** is the time at which a RREQ is transmitted at the Network layer. This is used in the measurement of delay for a given path, not necessarily the current path in use. The RREQ transmission time is used in the computation of the abstracted parameter AODV Path Delay.

- **RREP Receive Time:** is the time at which the RREP is received from a neighbouring node. This is used in the measurement of delay for a given path, not necessarily the current path in use. The RREP receive time is used in the computation of the abstracted parameter AODV Path Delay.
- **Destination IP Address:** is intercepted by the Network layer API from any transmitted or received data or control packet. It is used for identification of neighbouring nodes, in combination with the RSS at the MAC layer.
- **Source IP Address:** is intercepted by the Network layer API from any transmitted or received data or control packet. It is used for identification of neighbouring nodes, in combination with the RSS at the MAC layer.
- **Destination MAC Address:** is intercepted by the MAC layer API from any transmitted or received data or control packet. It is used by ROAM for identification of neighbouring nodes, in combination with the RSS at the MAC layer.
- **Source MAC Address:** is intercepted by the MAC layer API from any transmitted or received data or control packet. It is used by ROAM for identification of neighbouring nodes, in combination with the RSS at the MAC layer.
- **Packet Receive Signal Strength (RSS):** is passed to the MAC layer by the Physical header, on receipt of a packet and is measured in dBm. This indicates the quality of the path it has been transmitted on, between the source and the receiving node. The SINR is directly proportional to the Received Signal Strength, Noise Power and Interference Power. Only the first parameter can be measured at a single node, therefore this provides a proportional indicator of the channel quality. It is used for early identification of a degrading receiver in combination with the IP address of the source that is intercepted by the Network layer API. The ACK RSS is used as a late indicator of a degrading channel and RREP RSS as an early indicator.
- **Network Interface Index:** is the unique identifier of the network interface in use. This is used by the middleware to return an optimised routing parameter to AODV.
- **Data Packet UID:** is the unique sequence number of a data packet transmitted.

- **Data Packet Transmission Timestamp:** is time at which a data packet transmitted. This is used in the computation of the abstracted parameters Propagation Delay and the packet sending rate. These parameters indicate the quality of the channel in use.

The abstracted parameters that do not exist in the protocol stack but that are computed by the ROAM middleware are:

- **Current Next Hop:** is the IP address to which each data packet is transmitted, abstracted from the Network layer.

- **Map of IP Address to MAC Address:** is identified by ROAM based on the IP address intercepted at the Network layer API and MAC address taken from 802.11 for any packet transmitted or received.

- **Map of IP Address to MAC Address for Current Next Hop:** is identified by ROAM based on the IP address intercepted at the Network layer and MAC address taken from 802.11 when a data packet is transmitted.

- **Number of Neighbouring Nodes:** abstracted from the Network layer through an addition to the routing table. AODV adds to the routing table on the receipt of a control packet from a neighbour not currently stored in the routing table.

- **Best next hop:** the first of a list of current one-hop neighbours that is prioritised according to link quality

- **Degrading neighbour:** a neighbour separated by a link that is of reducing quality that is identified by several factors such as current AODV Path Delay and the RSS of RREPs received.

- **Propagation Delay(D_{prop}):** is calculated based on the time at which an ACK is received at the MAC layer (ACK_t) and the time of the first data packet transmission (DAT_t), where ${}_tP = ACK_t - DAT_t$. This includes time taken by retransmission and gives an indicator of the link quality.

- **AODV Path Delay(${}_tR$):** is calculated based on the time at which an AODV RREP is received at the Network layer ($RREP_t$) from a specific IP address and the time of the most recent RREQ transmission ($RREQ_t$), where ${}_tR = RREP_t - RREQ_t$ for each source IP. This indicates the quality of the path to each IP address from which an RREP has been received.

- **RREP Count:** The count of RREPs received with a RSS below a pre-specified threshold. This provides an early indication of a degrading channel.

ROAMs Store() function records the RSS of control packets (non-data packets) received at the MAC layer, if these have a RSS of more than (-74dBm). The reason that the RSS threshold is set at this value is that the MAC layer does not pass packets with RSS lower than -74dBm to the network layer. As these are considered out of range by the MAC layer, they are also by ROAM, therefore if these packets are from a link previously marked as incoherent, this link is removed from the list of INs to avoid. Parameters such as packet size, dataframe size and control packet size are abstracted by ROAM from protocol layers.

4.3.2 Contention Control Parameters

The parameters directly tuned by ROAM are:

- **Application Layer Transmission Rate:** this is tuned to prevent over-subscription of available bandwidth on a shared channel.

- **Application Layer Packet Size:** is increased in order to reduce the overheads associated with each packet.

The parameters indirectly tuned by the ROAM middleware are:

- **MAC Layer Queue Length:** if the busy time of the receiver is reduced, the number of packets that need to be queued at the MAC layer will decrease.

The descriptive parameters that are used but not tuned by the ROAM middleware are:

- **One-hop Neighbours:** received from the Network layer API as recorded in the routing table

- **Packet RSS:** is passed to the MAC layer by the Physical header, on receipt of a packet and is measured in dBm. This indicates the quality of the path it has been transmitted on, between the source and the receiving node. The SINR is directly proportional to the Received Signal Strength, or RSS, therefore this provides a proportional indicator of the channel quality. It is used for early identification of a degrading receiver in combination with the IP address of the source.

- **Destination IP Address:** is intercepted by the Network layer API from any transmitted or received data or control packet. It is used for identification of neighbouring nodes, in combination with the RSS at the MAC layer.

- **Source IP Address:** is abstracted at the Network layer from any transmitted or received data or control packet. It is used for identification of neighbouring nodes, in combination with the RSS at the MAC layer.

- **Destination MAC Address:** is intercepted by the MAC layer API from any transmitted or received data or control packet. It is used by ROAM for identification of neighbouring nodes, in combination with the RSS at the MAC layer.

- **Source MAC Address:** is intercepted by the MAC layer API from any transmitted or received data or control packet. It is used by ROAM for identification of neighbouring nodes, in combination with the RSS at the MAC layer.

- **Data Packet UID:** is the unique sequence number of a data packet transmitted, used to identify retransmissions and unacknowledged transmissions.

- **Data Packet Transmission Timestamp:** is time at which a data packet transmitted. This is used in the computation of the abstracted parameters Propagation Delay and the packet sending rate. These parameters indicate the quality of the channel in use.

- **Control Packet Receive Timestamp:** is the time at which an ACK, RTS, CTS, or network layer control packet is received from a neighbouring node. This is used in the measurement of delay for a given path, not necessarily the current path in use. The control packet receive time is used in the computation of the abstracted parameter Running Average Path Delay.

- **MAC Dataframe Size:** is the size of a packet at the MAC layer.

The abstracted parameters that do not exist in the protocol stack but that are computed by the ROAM middleware are:

- **Current Next Hop:** is the IP address to which each data packet is transmitted, abstracted from the Network layer.

- **Map of IP Address to MAC Address:** is identified by ROAM based on the IP address intercepted at the Network layer API and MAC address taken from 802.11 for any packet transmitted or received.

- **Map of IP Address to MAC Address for Current Next Hop:** is identified by ROAM based on the IP address intercepted at the Network layer API and MAC address taken from 802.11 when a data packet is transmitted.

- **Number of Transmitting Neighbours:** calculated by ROAM using the received rate of DATA packets not from the current next hop and RSS of packets from these neighbours in comparison to those from the current next hop.

- **Number of Neighbouring Nodes:** abstracted from the Network layer through an addition to the routing table. AODV adds to the routing table on the receipt of a control packet from a neighbour not currently stored in the routing table.

- **Network Control Packet Size range:** utilised by ROAM to observe received packets but that are smaller than the size of a data packet. This is calculated using the Application Data Packet Size and MAC Dataframe Size.

- **Running Average Path Delay (D_p):** calculated as the time between related data and control packet transmission and receipt e.g. between RTS and CTS, RREQ and RREP or DATA and ACK.

- **Running Average Number of Retransmission Limit Exceeded Events per Second (RTX_R):** the running average number of times that the retransmission of a packet ceases without the receipt of an ACK. This is due to the maximum number of retransmissions (set at the MAC layer) being reached and the packet being dropped by the 802.11 protocol. Retransmission limit exceeded events occur when a packet is repeatedly lost due to collisions or packet errors at a receiving node and an ACK is therefore not returned by that node.

- **Running Average MAC Layer Queue Length:** calculated by ROAM to identify whether the MAC queue length tended to increase or decrease over time. This is used to identify periods during which the channel is increasingly in use and packets are therefore buffered.

- **Running Average Data Rate:** the running average rate of data packets received that do not have this node as a source or destination.

- **ACK Receipt Rate:** the rate of ACK packets received that have this node as destination.

- **Not For Me ACK (NFM ACK) Receipt Rate:** the rate of ACK packets received that have the current next hop as the destination.

4.4 ns2-MIRACLE Implementation of ROAM

This section details the development of the ROAM middleware and messaging approach in the ns2-MIRACLE simulator. ns2-MIRACLE has been designed to enable simulation of interaction between a cross-layer, kernel-based entity (ROAM) and network protocol modules [15]. The ROAM middleware was therefore developed as a monolithic class, as an extension of the Plugin class provided by the simulator. The API exported to the network protocol layers were also created in the simulator in order to function alongside the protocol modules. AODVUU is the sole ad hoc routing protocol provided by the ns2-MIRACLE simulator.

The ns2-MIRACLE implementation of ROAM therefore consisted of:

- Development of ROAM in the format of a ns2-MIRACLE PlugIn
- Creating cross-layer messages through extension of the ns2-MIRACLE ClMessage class
- Development of layer-specific API, able to access appropriate layer parameters
- Export of the API to the application (CBR, VoIP), network (AODVUU) and MAC (MiracleMac802_11) modules

4.4.1 ClMessage Module

A new cross-layer message, ClMessage class was created that defined three new asynchronous cross-layer message formats through extension of the ns2-MIRACLE reference class `ClMessage`: `QUERYMSG`, `INTFCMSG` and `OPTMSG`. Three related classes: `QueryMsg()`, `IntMsg()` and `OptMsg()`, and their related tracer classes: and `Query Tracer()`, `IntTracer()` and `OptTracer()` were also developed. Two structures, `RStats` and `OptStats`, were then used for the transmission of protocol parameters. The first held parameter values abstracted from the protocol modules and the second held optimised parameter values returned from the middleware to the protocols.

```
--------------------------------------------------------------------
|| verbosity | ClMessage_t type | DestinationType dtype  | value ||
--------------------------------------------------------------------
```

Figure 4.5: ns2-MIRACLE ClMessage Format

The generic structure of a `ClMessage` followed the form shown in figure 4.5. The integer `verbosity` indicated the degree to which the message was received by modules as it passed across the SAPs, the lateral connections between the simulator, plugins and simulation modules. The `ClMessage_t type` was the unique identifier of the message, such as `QUERYMSG` and the `DestinationType dtype` was either set to `BROADCAST` or `UNICAST`, depending on which modules were to receive the message. Finally, if the message was sent `UNICAST` then the integer `int value` was set to the identifier of the receiving module: layer or plugin. The `QUERYMSG ClMessages` used in ROAM were broadcast to all modules in order to locate participating modules. All other `ClMessages` were sent unicast to reduce messaging overheads and avoid the situation of several modules overwriting the same message.

Therefore the three clmessages were set as shown in figure 4.6 and the related tracer message fields were similarly formed as in figure 4.7. Each of the subclasses for the cross-layer messages and their associated tracer classes were then defined, firstly in a header file, inheriting from the generic ns2-MIRACLE class `ClMessage`, shown in figure 4.8. The private structures, such as the `OptStats` created could be modified by their class in order to store and forward parameter values between modules and the ROAM middleware. The message handling functions `getStats()` and `setStats()` were then used to access stored parameters from and write to this structure.

The developed cross-layer messages and tracers were declared both within the `Plugin` and exported API (using e.g. `extern ClMessage_t QUERYMSG;`) and the `initlib.cc` file with the code in figure 4.9, in order for these messages to be added to the ns2-MIRACLE `ClMessage` list.

```
QueryMsg::QueryMsg(int verbosity, DestinationType dtype,
          int source, int value):
          ClMessage(verbosity, QUERYMSG, dtype, source, value){}

IntMsg::IntMsg(int verbosity, DestinationType dtype,
          int source, int value):
          ClMessage(verbosity, INTFCMSG, dtype, source, value){}

OptMsg::OptMsg(int verbosity, DestinationType dtype,
          int source, int value):
          ClMessage(verbosity, OPTMSG, dtype, source, value){}
```

Figure 4.6: ROAM ClMessage Module: Message Definitions

4.4.2 Middleware Plugin

A new `Plugin` class was developed as shown in figure 4.10. This class was defined to be set up and added to a node as well as take commands from the Tcl simulation

```
QueryTracer::QueryTracer() : ClMessageTracer(QUERYMSG){}

IntTracer::IntTracer() : ClMessageTracer(INTFCMSG){}

OptTracer::OptTracer() : ClMessageTracer(OPTMSG){}
```

Figure 4.7: ROAM ClMessage Module: Tracers

```
class OptMsg : public ClMessage
{
        public:
                OptMsg();
                OptMsg(OptMsg *m);
                OptMsg(int verbosity, DestinationType dtype,
                                        int source, int value);

                ClMessage* copy();
                OptStats getStats();
                void setStats(OptStats s);

        private:
                OptStats optstats_;
};Module

class OptTracer : public ClMessageTracer
{
        public:
                OptTracer();
                void format(ClMessage *m, ConnectorTrace *sap);
};
```

Figure 4.8: ROAM ClMessage Module: Header File

```
QUERYMSG = ClMessage::addClMessage();
ClSAP::addTracer(new QueryTracer);
```

Figure 4.9: ROAM ClMessage Module: Declaration of Messages

script to attach a new plugin to a particular node. These commands can also start ROAM optimisation and `ClMessage` transmission from a specified simulation time (figure 4.11).

The `Plugin` sent the `Clmessages` discussed in the previous section to an appropriate protocol module (`dest`) by creating a new `Clmessage`, using the `Plugin ID` as the source using the ns2-MIRACLE command `getId()` (figure 4.12).

```
static class PlugInClass : public TclClass {
 public:
        PlugInClass() : TclClass("roamPlugIn") {}
        TclObject* create(int, const char*const*) {
                return (new roamPlugIn());

        }
} class_roamplugin;
```

Figure 4.10: ROAM Middleware Plugin Class

```
   set plg       [new roamPlugIn]
   set Plugin    [$node addPlugin $plg 1 "ROAM"]
                 $ns at 100.0 "$plg start_ROAM"
```

Figure 4.11: Simulation Command: Middleware Inclusion

```
 ClMessage *c = new QueryMsg(DEFAULT_CLMSG_VERBOSITY,
                              BROADCAST, getId(), dest);
  sendAsyncClMsg(c, 0);
```

Figure 4.12: ROAM Middleware Plugin: Use of Clmessages

For parameter values to be enclosed in the `ClMessage`, a structure, such as `OptStats`, was accessed using the `getStats()` function and the appropriate value written to the structure and sent in the message (figure 4.13).

On receipt of a `ClMessage`, the receiving module would then access the message and parameters contained by first identifying the message type (e.g. using `if (m->type()==QUERYMSG)`). The parameters stored in the structure were then extracted (e.g. using `int roam_Plugin_ID = m->getSource();`) and the message deleted (`delete m;`).

4.4.3 Protocol Layer API

The ROAM API were developed as generic interfaces to specific protocol parameters that could be exported to any protocol module. They were formed of four functions:

```
OptStats b = ((OptMsg*)c)->getStats();
b.Value = parameter_value;
```

Figure 4.13: ROAM Middleware Plugin: Accessing Parameters

- `OnEvent()` for the identification and export of a parameter value
- `sendAsyncClMsg()` to create and send a `ClMessage` containing this parameter value
- `recvAsyncClMsg()` to receive, add the layer ID and return a `QUERYMSG`
- `recvAsyncOptMsg()` to receive a `ClMessage` containing an optimised parameter value

When the API received a `QUERYMSG` from the `Plugin` it commenced collecting parameter values and sending these to the `Plugin`, either when they reached a threshold or each time they changed in value, depending on the callback functions defined in the API for that layer. For an identified parameter, calling the `OnEvent()` function then abstracted these values. Figure 4.14 gives an example of the `OnEvent()` function call. Here two parameters are abstracted: an integer (`Parameter_Value1`) and a double (`Parameter_Value2`), alongside a unique Parameter Id and the current simulation time, used to timestamp (`T_stamp`) the message.

```
OnEvent(Parameter_Id, Parameter_Value1,
                      T_stamp, Parameter_Value2);
```

Figure 4.14: Protocol Layer API: OnEvent Function

```
ClMessage *c = new IntMsg(DEFAULT_CLMSG_VERBOSITY,
                          UNICAST, getId(), roamAddr);
RStats b = ((IntMsg*)c)->getStats();
b.LayerId = getId();
b.MetId = Parameter_Id;
b.MetValue = Parameter_Value1;
b.rxpower = Parameter_Value2;
b.timestamp = Timestamp;
((IntMsg*)c)->setStats(b);
sendAsyncClMsg(c, 0);
```

Figure 4.15: Protocol Layer API: Clmessage Transmission

A `ClMessage` would then be transmitted to the ROAM middleware for each updated parameter value: a new `INTMSG` created and the contained structure,

```
int Parameter_Id = ((OptMsg *)m)->getStats().MetId;
```

Figure 4.16: Protocol Layer API: Clmessage Receipt

RStats, accessed. The module layer, parameter ID, values of the associated parameters and timestamp would be included in a structure, RStats, and the message transmitted (see figure 4.15).

On receipt of a ClMessage the message type would be examined (if (m->type() ==OPTMSG)) and the Parameter_Id identified (figure 4.16). This enabled the API to access the appropriate parameter that had been tuned by the Plugin. The optimised Parameter_Value would then be substituted for the current value of that parameter by the API accessing the parameter within the protocol module.

4.5 Challenges to Extensibility

The middleware architecture has been designed to manage optimisation of many MANET protocols, however, assumptions made in the design of the optimisers place certain limits on their operation. The optimisers monitor network and MAC layer control packets and IP addresses for identification and detection of neighbouring nodes and path quality. Therefore, IP addressing in combination with the use of control packet exchange at the network and MAC layers must be in use. The implementation is specific to MANET protocols as the API is used to manipulate and monitor particular protocol data structures, for example the transmission and receipt of routing control packets is assumed to be only for the purpose of maintaining or setting up a new path.

ROAM does not provide direct delay control through traffic conditioning. Therefore, if a requirement to provide optimal horizontal handoff or contention control does not occur, ROAM is incapable of providing bounded delay, jitter or packet loss ratio. For example, in the former if a more robust channel is not available or, in the latter, if an exposed node contends for the channel ROAM will not tune any protocol parameters.

The horizontal handoff optimiser assumes that ROAM is utilised by a MANET transmitter and thus that dynamic routing, commonly used in MANETs, rather than statically predefined routing is in use. If the aforementioned conditions are not fulfilled, ROAM will monitor, but not tune network protocols or network performance. ROAM relies on RSS as part of evaluation of channel quality by the horizontal handoff optimiser. However, this is used as a relative measurement that compares control packets intended for the ROAM node from two links. Therefore ROAM is not dependant on inter-nodal distance.

The distributed Contention Control Optimiser does not directly distinguish between a hidden transmitter on the network or an IN that is forwarding packets. In the latter situation, if the available bandwidth on the channel to the current next hop is reduced by traffic from a hidden node, that node will be seen as a hidden transmitter and contention control optimisation will commence. Finally, the contention control optimiser functions on a distributed basis, thus requiring the middleware to be implemented in all transmitters on the network. This is a requirement to ensure that load reduction has a fair result for all competing flows.

4.6 Summary and Discussion

This chapter lays out the form and functionality of the middleware architecture designed to optimise the QoS provided to time-critical applications in an ad hoc network (ROAM). The purpose of the middleware is to reduce jitter and delay without constraining overall performance and packet delivery.

ROAM utilises API exported to protocol layers, without modifying the protocols themselves, to abstract selected parameters. These API can send and receive parameter information using messages provided by ROAM. On the basis of information from the MAC and network layers, such as transmitters in range, routing table changes, ACK rates or delay between data transmission and ACK receipt, the need to call either the Horizontal Handoff or Contention Control Optimiser is identified. The aim of the middleware has not been to preempt ordinary protocol functioning, but to execute concurrently with the stack. Additionally, where protocols such as RTS/CTS queue packets until handshaking is complete, ROAM has been developed to not introduce artificial delays. Therefore, for example, while ROAM is able to detect neighbouring nodes before the routing protocol, fast horizontal handoff is implemented only when the optimal neighbour has been added to the routing table, although not necessarily as the current next hop.

ROAM employs these two optimisers to improve the network performance provided to the transmitter that it operates in. By acting on a distributed basis, within each single transmitter, ROAM tunes the application and network layers to instantaneous network conditions. The result prevents the inefficiencies that occur within a transmitter that were identified through the scoping experiments in Chapter 3. The optimisers have also been designed to not selfishly impact on the performance of other nodes in range.

As considered in the previous chapter, there are a number of factors that influence QoS dynamics in a MANET. The middleware architecture therefore must be tested under a range of QoS conditions, including changes in load through variations in transmission rate and packet size. Varying distance to the receiver and the number of transmitters using the MANET should also be used to confirm scalability. Finally, ROAM has been designed to improve the performance of time-critical applications and demonstration of this generic approach requires analysis of the interaction between ROAM and different application layer protocols. Therefore, the results of detailed evaluation of the middleware in a simulation environment are discussed in the following chapter.

Chapter 5

Performance of ROAM Handoff Optimiser

5.1 Introduction

The self-organising, self-configuring nature of MANETs is dependent on the discovery of appropriate end-to-end (E2E) paths by ad hoc routing protocols that do not rely on statically predefined routes. These protocols entail the regular exchange of route request (RREQ) packets and route reply (RREP) control packets by nodes, in order to discover nodes that are in range and to set up new paths and maintain existing paths to the receiver. As mobile transmitters move through a MANET of nodes using the same technologies, they will handoff horizontally from one forwarding IN to another. The investigative simulations discussed in Chapter 3 demonstrated that horizontal handoff without reference to channel conditions can result in localised increases in packet loss, jitter and delay. Packets may be repeatedly transmitted over fading links during the process of selecting a new E2E path. Buffering and retransmission enable IEEE 802.11 to recover from packet loss on suboptimal channels, but result in increased E2E packet delays. Additionally, switching between fading and coherent links also results in repeated flooding of RREQ packets, increasing contention delays and congestion.

The cross-layer optimiser validated in this chapter is implemented in the ROAM architecture in order to reduce the performance anomalies associated with horizontal handoff in a MANET. In order to avoid frequent performance drops, horizontal handoff should be fast and prevent switching to a fading path if one of better and increasing coherence is available. The purpose of the optimiser is to ensure that maximum delay, jitter and loss are not associated with handoff. Therefore, these statistics should be lower at handoff than measured during the initial path setup phase when the nodes join the network (Chapter 3).

ROAM introduces the adaptive tuning of application and network parameters and requires minimal configuration only prior to runtime. Optimisation is implemented within the protocol layers of a single mobile transmitter without global signalling of network conditions. Instead, MAC layer information on gradual and rapid changes in channel quality is used to ensure that optimal local links are selected to form the E2E path. Both the ROAM architecture and contention control algorithm have been discussed in detail in Chapter 4.

Section 5.2 evaluates simulation results with the ROAM horizontal optimiser enabled. These are compared to MANET performance with handoff reliant on a reactive ad hoc routing protocol, without access to cross-layer information. A simulated implementation of the AODV routing protocol has been used for this purpose. Validation is then extended to a comparison against a best-case, or baseline MANET performance scenario in Section 5.3 in a single-hop wireless network. This is to evaluate the limitations of the optimiser in conveying performance in a MANET with complex network dynamics closer to that of a network with low levels of resource variation.

Variable MANET configurations and application transmission settings create network dynamics that subject flows to increased packet delay and jitter. Bounded E2E delay and jitter are vital to the provision of QoS to inelastic soft real-time (ISRT) applications, as discussed in Chapter 2. Therefore, the middleware has been rigorously tested for its independence of firstly, transmission setting and secondly, MANET configuration and in both these cases for the ability to constrain E2E delay and jitter. The simulation design and configuration details are given in Section 3.3 of Chapter 3. The test scenarios have been developed in light of the variation in transmitted load and contention of RT applications and that MANETs may self-configure in a range of topologies, with variation in number of traffic sources, node mobility and node speed.

5.2 Simulation Results

The ROAM Horizontal Handoff Optimiser has been validated from various aspects in ns2-MIRACLE. Each sub-case of these simulation scenarios has been tested through 10 simulation runs and means collated. The use of packet error and environmental propagation models by the simulator, and corresponding variations in packet dropping, introduce stochasticity between simulation runs, when the time at which application transmission begins is changed. The ROAM architecture was implemented only in Node 1 in these simulations, providing heterogeneous network comparison in the multi-transmitter simulations. Any resultant impact on performance in other transmitters on the network was, therefore, also investigated. The results with ROAM are compared to those with the AODV-UU implementation of the AODV routing protocol to provide a comparison of normal MANET performance and because the middleware is designed to function alongside reactive ad hoc routing protocols.

Table 5.1: Scenario 1: Overall Packet Drop Comparison

		AODV			ROAM		
TR (Mbps)	PS (B)	Collision Count	IFQ Full	No Route	Collision Count	IFQ Full	No Route
1	500	416	16582	1142	282	15537	495
1.25	500	498	17009	1198	78	15239	840
1.5	500	344	17313	1215	69	15664	821
1.75	500	288	16388	198	2	16257	91
2	500	331	17949	390	7	17190	25
1	700	56	7689	7	0	7725	3
1.25	700	57	10477	79	0	10013	11
1.5	700	145	9882	21	13	10043	8
1.75	700	240	12282	72	0	12103	23
2	700	316	14086	221	5	13502	90
1	900	202	13235	632	101	11323	159
1.25	900	204	12155	301	59	10570	116
1.5	900	79	8442	271	20	8093	8
1.75	900	195	10037	159	0	9202	10
2	900	371	12488	151	51	11807	52
1	1100	91	12480	213	30	12317	49
1.25	1100	190	12862	299	47	12804	102
1.5	1100	79	7773	113	52	7112	3
1.75	1100	123	8874	53	0	8662	7
2	1100	238	11455	92	2	11437	37
1	1300	59	10635	274	16	10392	28
1.25	1300	68	11176	95	14	10957	27
1.5	1300	72	8435	89	0	8139	9
1.75	1300	115	8944	90	0	8794	13
2	1300	205	10199	92	0	10056	28

5.2.1 Testing Transmission Setting Independence

5.2.1.1 Scenario 1: Transmit Rate and Packet Size Variation

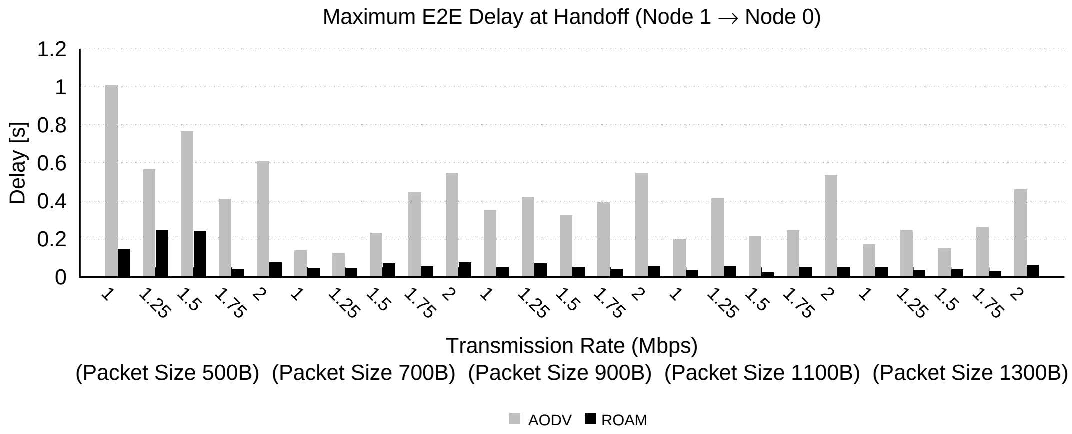
In this scenario, the performance of ROAM has been validated in a star topology (detailed in Chapter 3) with one CBR source. Transmitted load was varied through an increase in CBR application transmission rate (TR) and packet size (PS). The period between transmissions and packet size are variables implemented in simulation, therefore, the packet transmission rate in packets per second is dependent on transmission bitrate and packet size. The transmission rate was varied, at intervals of 0.25Mbps, between 1-2Mbps and packet size, at intervals of 200B, between 500-1300B. This is to demonstrate capability to constrain E2E packet loss, delay and jitter under the dynamic network conditions caused by load variation and interaction between protocol process and network configuration with varying packet size and that the optimiser is independent of particular flow settings.

The MAC layer transmits packets at a specific datarate, which is determined according to noise on the channel. Therefore, with a smaller packet size more packets are transmitted under the same datarate. MAC layer random backoff and packet collisions, therefore, occur more frequently. Suboptimal channels are repeatedly selected as the next hop during a horizontal handoff due to reliance of ad hoc routing protocols on RREP receipt for path maintenance. This leads to preferential selection of longer, established hops (figure 5.6).

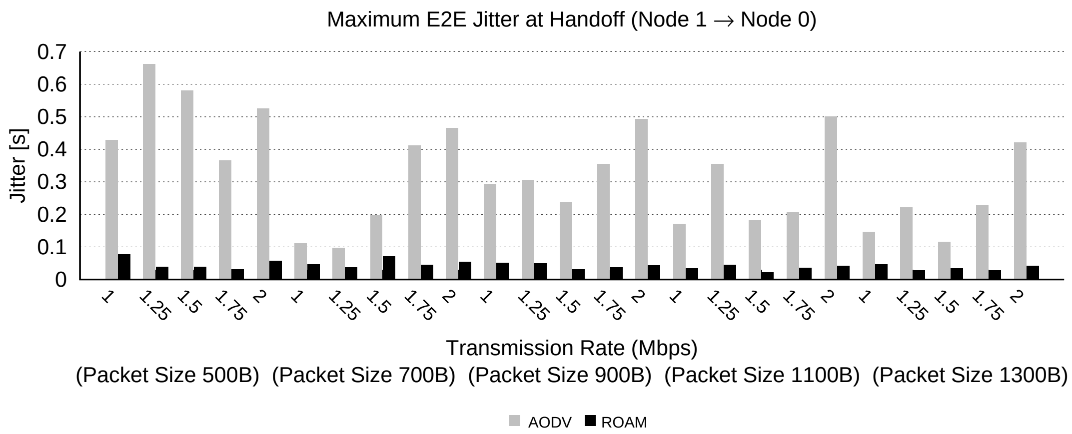
Link fading during MAC layer backoff periods affects the ability of the MAC layer to detect the channel state, and by reducing the use of these links ROAM reduces transmissions that are subject to interference. Figure 5.1(c) show that this reduced overall packet loss, with improved performance at higher packet sizes.

Table 5.1 shows the causes of packet dropping in the two sub-cases: with AODV alone and with ROAM optimisation of horizontal handoff. Total packet drops were reduced in all scenarios by ROAM. Collisions and routing errors were higher with AODV than ROAM. This is because ROAM improves routing protocol performance by monitoring and identifying optimal links prior to handoff. The middleware optimiser then tunes protocol parameters to ensure that the routing protocol selects these links rather than switching intermittently to suboptimal links, where packet loss will increase due to destructive signal fading. Reactive ad hoc routing utilises link layer information for path maintenance, therefore, increasing traffic rates and corresponding ACK rates improves performance.

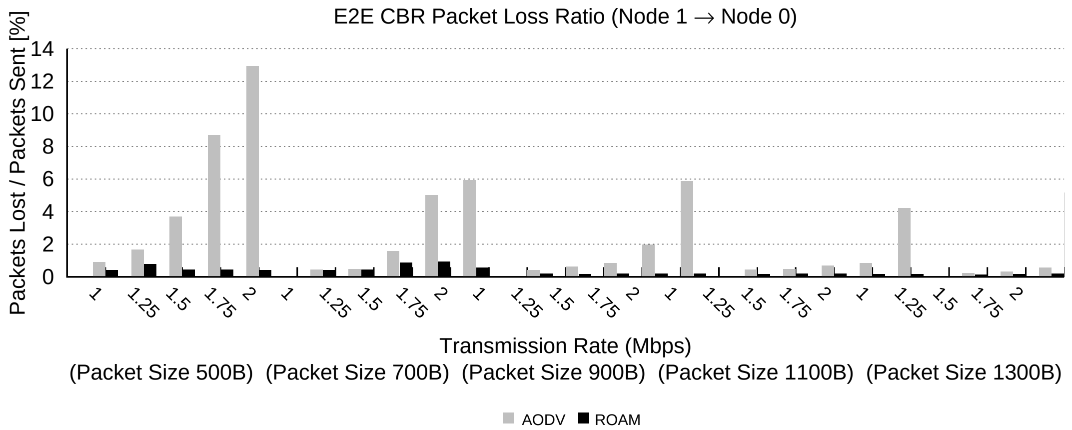
At higher transmission rates IFQs are enqueued faster and, depending on channel busy time, may not empty at the same rate. As all CBR streams shared a single receiver, increasing traffic rate can result in a bottleneck at the final link, however, routing on multiple paths relieves this pressure. In a multi-hop network,



(a) Maximum E2E Delay at Handoff ($N1 \rightarrow N0$)

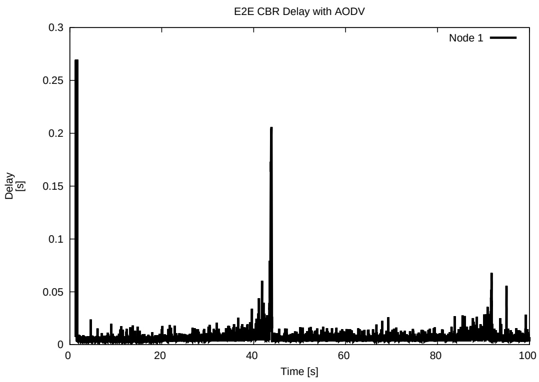


(b) Maximum E2E Jitter at Handoff ($N1 \rightarrow N0$)

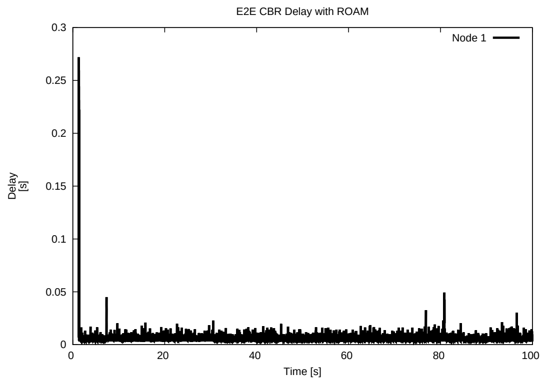


(c) E2E CBR Packet Loss Ratio ($N1 \rightarrow N0$)

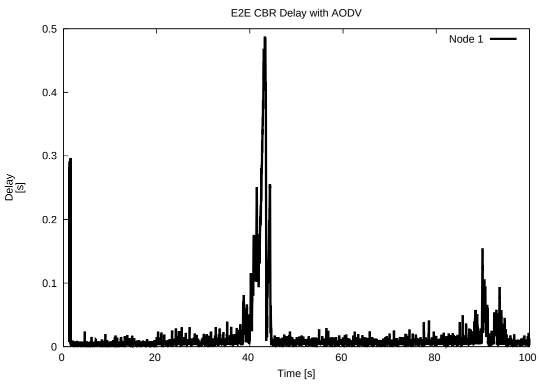
Figure 5.1: Scenario 1: Performance Comparison (AODV versus ROAM)



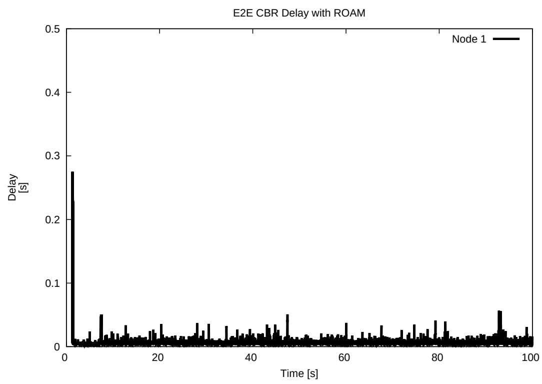
(a) AODV (TR = 1Mbps, PS = 1300B)



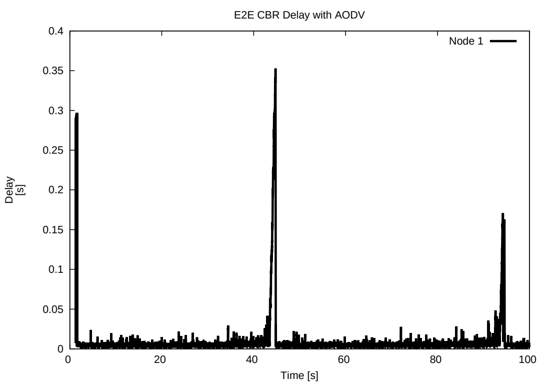
(b) ROAM (TR = 1Mbps, PS = 1300B)



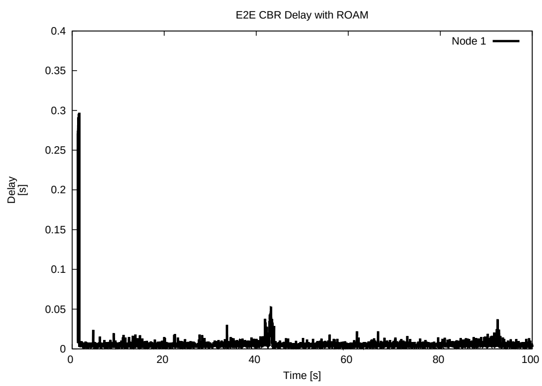
(c) AODV (TR = 1.25Mbps, PS = 1100B)



(d) ROAM (TR = 1.25Mbps, PS = 1100B)

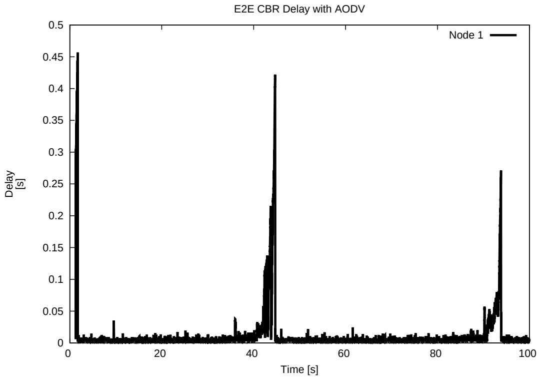


(e) AODV (TR = 1.5Mbps, PS = 900B)

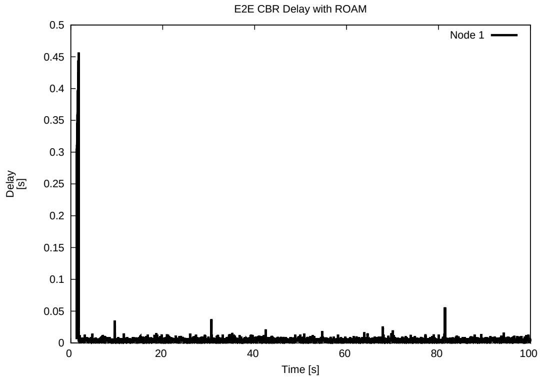


(f) ROAM (TR = 1.5Mbps, PS = 900B)

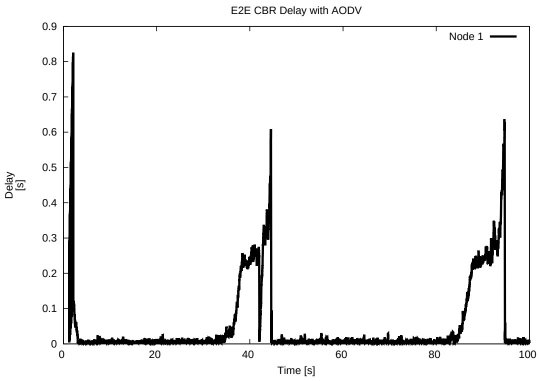
Figure 5.2: Scenario 1: Instantaneous E2E Delay (AODV versus ROAM)



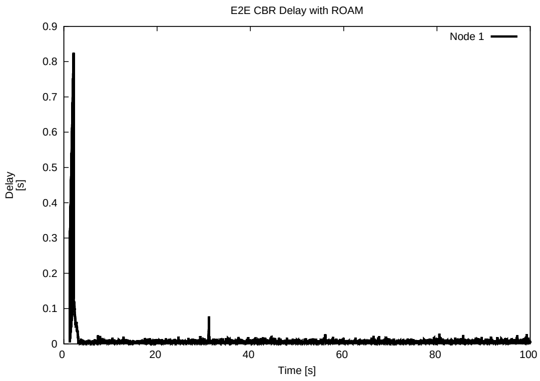
(a) AODV (TR = 1.75Mbps, PS = 700B)



(b) ROAM (TR = 1.75Mbps, PS = 700B)



(c) AODV (TR = 2Mbps, PS = 500B)



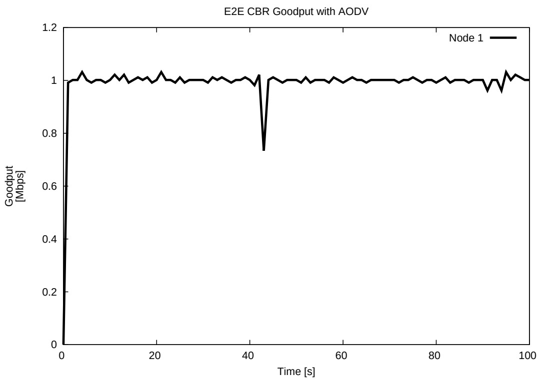
(d) ROAM (TR = 2Mbps, PS = 500B)

Figure 5.3: Scenario 1: Instantaneous E2E Delay (AODV versus ROAM)

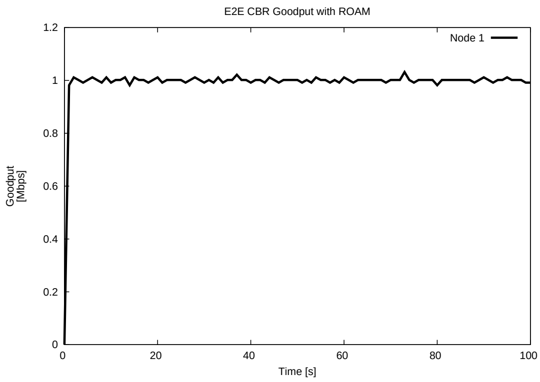
retransmission of packets is used by the MAC layer at each forwarding and transmitting node for loss recovery. Thus total MAC layer packet transmissions and packets dropped are likely to exceed the application transmission rate.

Loss of control packets results from interference of neighbours that can be at a distance much greater than a nodes transmission range, but is also topology dependent. While the incidence of full queue drop was not significantly reduced by ROAM, reduced retransmission requirements and packet queueing during path maintenance resulted in lower delays during handoff. As previously discussed, initial path setup by the routing protocol creates startup peak delay as packets are buffered. Therefore, maximum delay values were measured following this initial period. Figure 5.1(a)-(b) demonstrate that in all scenarios delay and jitter were bounded to below 0.3s and to below 0.1s when the packet size was larger than 500B. The reduction in delay is related to the reduction of packet dropping on a fading link and related recovery.

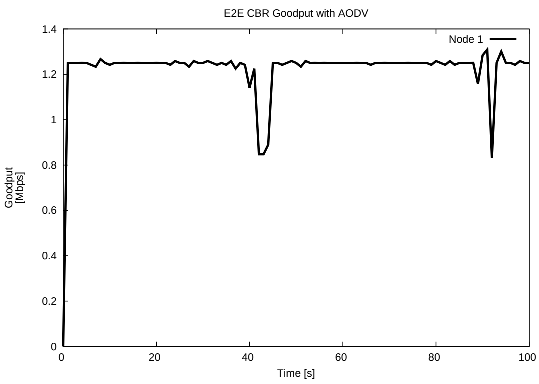
Reactive MANET protocols, such as AODV, rely on the use of continual net-



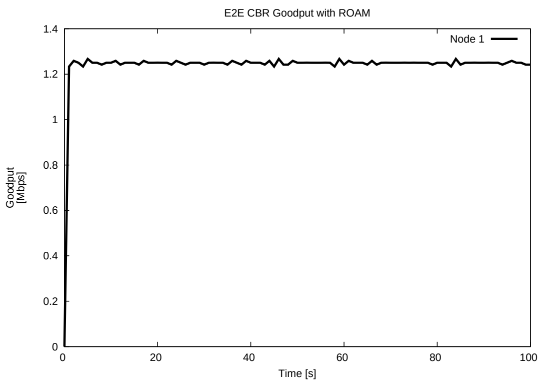
(a) AODV (TR = 1Mbps, PS = 1300B)



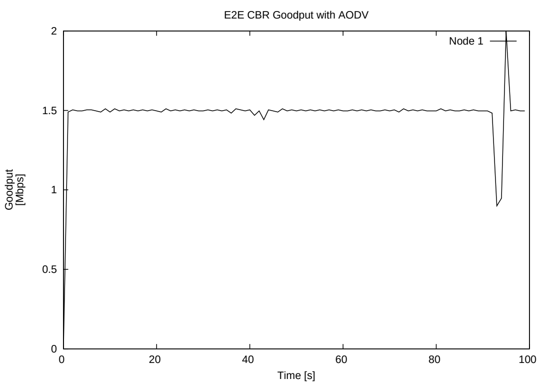
(b) ROAM (TR = 1Mbps, PS = 1300B)



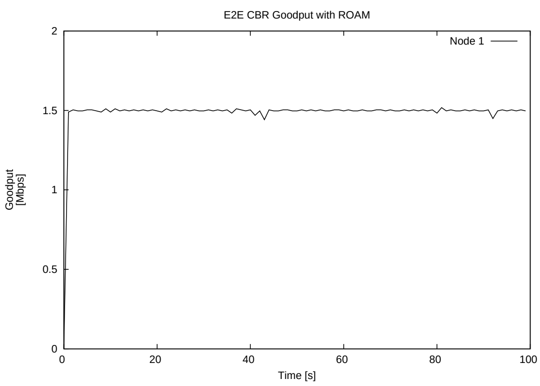
(c) AODV (TR = 1.25Mbps, PS = 1100B)



(d) ROAM (TR = 1.25Mbps, PS = 1100B)

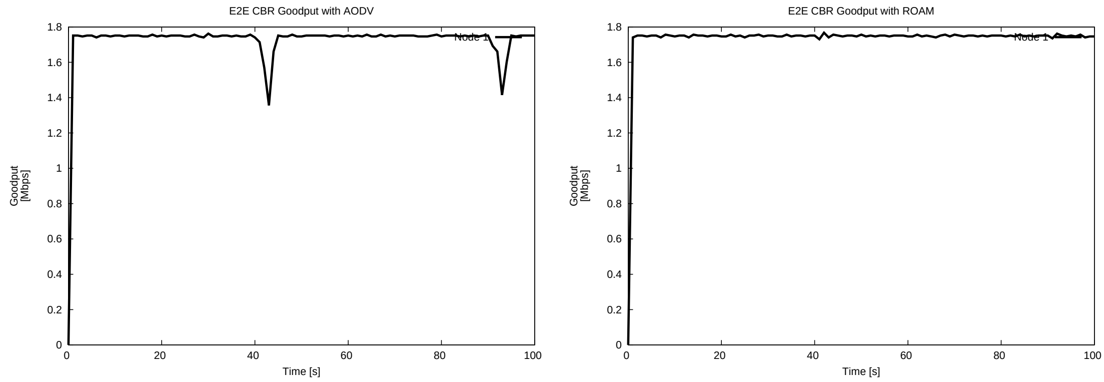


(e) AODV (TR = 1.5Mbps, PS = 900B)

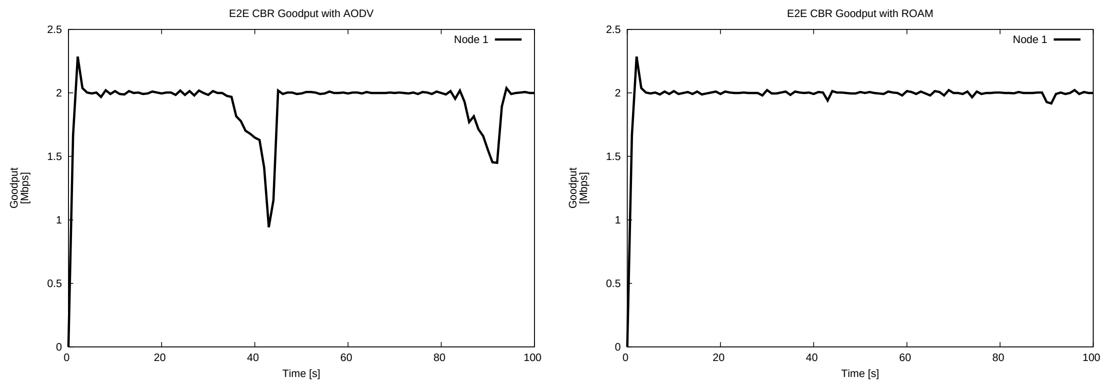


(f) ROAM (TR = 1.5Mbps, PS = 900B)

Figure 5.4: Scenario 1: Instantaneous E2E CBR Goodput (AODV versus ROAM)



(a) AODV (TR = 1.75Mbps, PS = 700B) (b) ROAM (TR = 1.75Mbps, PS = 700B)



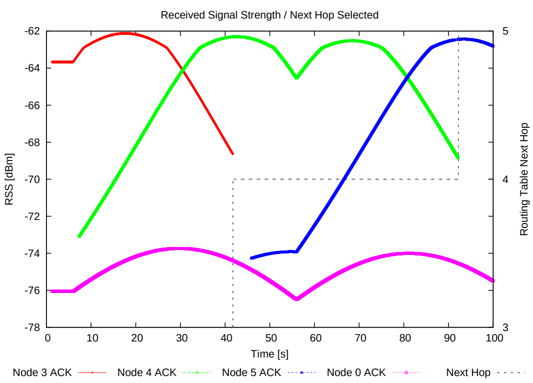
(c) AODV (TR = 2Mbps, PS = 500B) (d) ROAM (TR = 2Mbps, PS = 500B)

Figure 5.5: Scenario 1: Instantaneous E2E CBR Goodput (AODV versus ROAM)

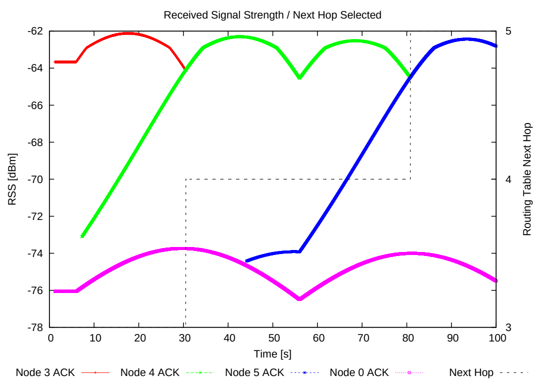
work and link layer control frame exchange to maintain E2E paths. As a result, if control packets are received on a fading channel to a receding node or a non-robust channel to a highly mobile node these can be used as the next hop. MAC layer retransmission as well as retransmission of routing control packets then lead to elevation in collisions and errors. The auto-fallback mechanism of IEEE 802.11 is designed to step down MAC transmission rates if noise on a channel increases. Correspondingly, packets begin to be enqueued at a higher rate than they are dequeued, resulting in a queueing backlog. Notably, with AODV the period of degraded performance and increased delay surrounding handoff was longer at higher datarates. With increased traffic pressure on the IFQ, queueing and retransmissions due to fading link usage more regularly resulted in buffer overflow.

With small packets sent at high application transmission rates, the queueing backlog exceeds buffer provisioning causing packets to be dropped. Additionally, when no route is found by AODV, packets are dropped. This occurs more frequently under higher transmitted load as a result of more collisions with routing control packets on the fading link. With transmission rates of 1-1.5Mbps and a packet size of 700B, the packet loss ratio was unchanged. In this scenario AODV performs well as the low enqueueing rate puts less pressure on the queue in spite of MAC retransmissions. However, AODV does not provide this level of performance under different network dynamics.

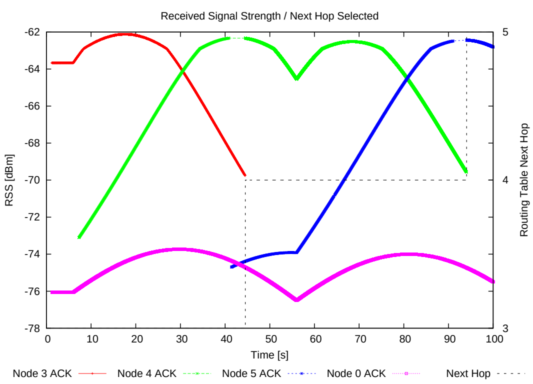
Figures 5.2–5.5 show the instantaneous E2E performance during one run of a simulation. These demonstrate that ROAM is capable of constraining maximum E2E delay and jitter during handoff, but not during initial path setup. This is the period during which AODV floods RREQ and RREP packets through the network to set up an E2E path. This corresponds to an initial peak in E2E delay. With nodes orbiting the star topology at a speed of 1m/s, horizontal handoff occurred every 40s. Even though the receiver is only two hops away, protracted handoff from a receding forwarding node resulted in high packet delay. The period of associated performance degradation extended with lower packet size and high traffic rate.



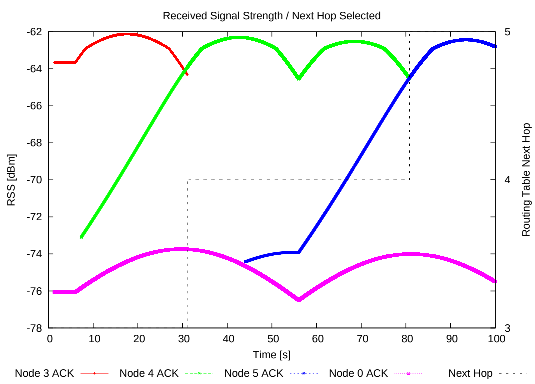
(a) AODV (TR = 1Mbps, PS = 1300B)



(b) ROAM (TR = 1Mbps, PS = 1300B)



(c) AODV (TR = 2Mbps, PS = 500B)



(d) ROAM (TR = 2Mbps, PS = 500B)

Figure 5.6: Scenario 1: Next Hop Selected by Node 1 (AODV versus ROAM)

Table 5.2: Scenario 2: Overall Packet Drop Comparison

	AODV			ROAM		
Sub-Scenario	Collision Count	IFQ Full	No Route	Collision Count	IFQ Full	No Route
2.1	64	12669	326	10	12513	200
2.2	41	12020	426	27	12272	327
2.3	45	12284	218	22	12051	101
2.4	16	10988	367	20	10982	363
2.5	35	11536	489	21	11326	418

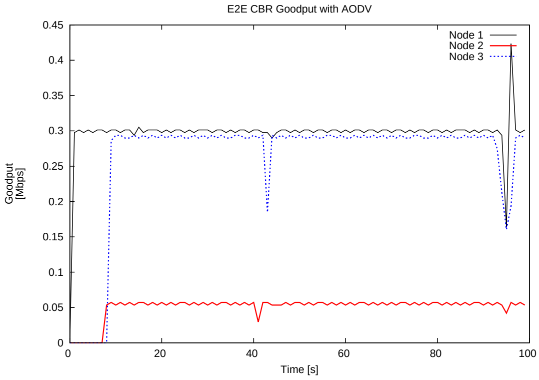
5.2.1.2 Scenario 2: Heterogeneous CBR Traffic

Future military and disaster response networks are likely to require high performance under varied traffic loads, therefore, this scenario considers the impact of configuring CBR transmitters with mixed initial sizes and transmission rates. Three CBR sources transmitted heterogeneous streams to the same receiver, node 0 in sub-scenarios 2.1-2.5. The detailed configuration of nodes in this scenario is given in Section 3.3.2 of Chapter 3.

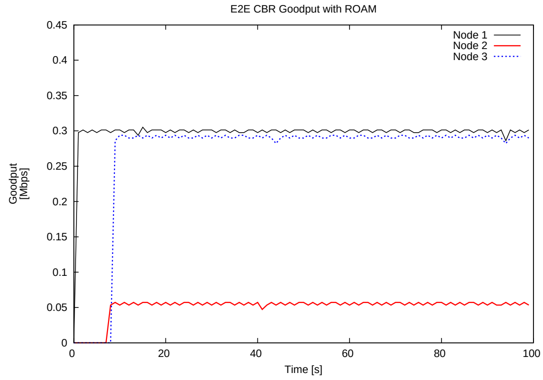
Under heterogeneous conditions, network dynamics become more complex as channel usage differs between nodes and packet delays in enqueueing, dequeueing and transmission become more variable. With multiple flows forwarded through the network, suboptimal link selection can result in increased jitter and routing information and channel quality on the E2E path will change rapidly and abruptly. The purpose of this scenario was to validate that ROAM is stateless, scalable and not reliant on continuous conditions across a network.

Handoff for each CBR source occurred every 50s and figures 5.7–5.8 indicate that with mixed traffic rates, as expected from the previous simulations, AODV performance at handoff differed between nodes. Degradation in goodput prior to handoff resulted, with almost complete signal loss in sub-scenarios 2.2 and 2.5. When the ROAM optimiser was implemented over AODV, nominal performance degradation resulted. Table 5.2 demonstrates the key performance improvements, by which packet dropping was reduced, were reduction in collisions and routing errors.

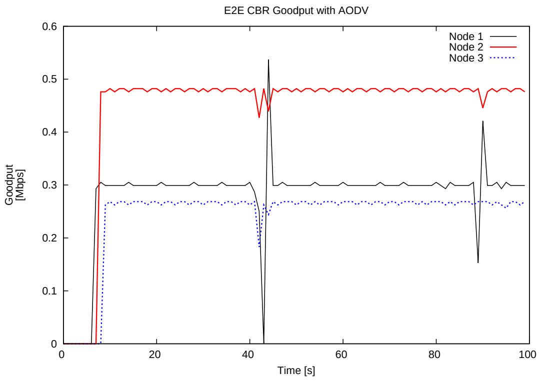
Correspondingly, figure 5.10 demonstrates that when fast handoff was implemented with ROAM, overall packet loss decreased for Node 1. Through reduction in unnecessary transmissions on shared paths, the performance of unoptimised nodes is also improved. However, in sub-scenario 2.3 AODV provided better performance for Node 2 and 3 than ROAM. Packet loss, delay and jitter were increased for Node 2 and loss was also increased for Node 3, although this did not impact on



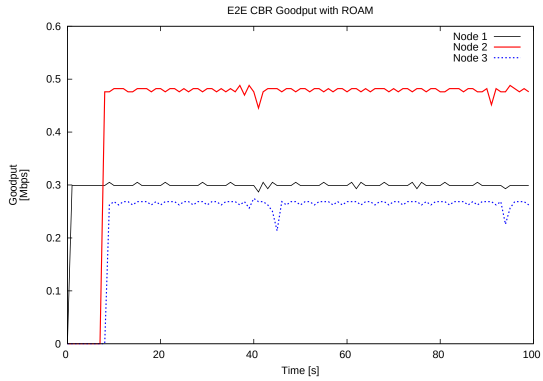
(a) AODV (Sub-scenario 2.1)



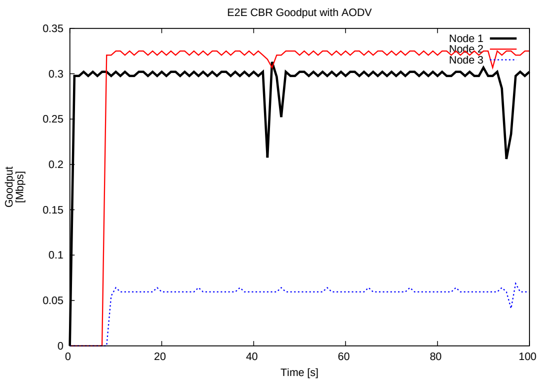
(b) ROAM (Sub-scenario 2.1)



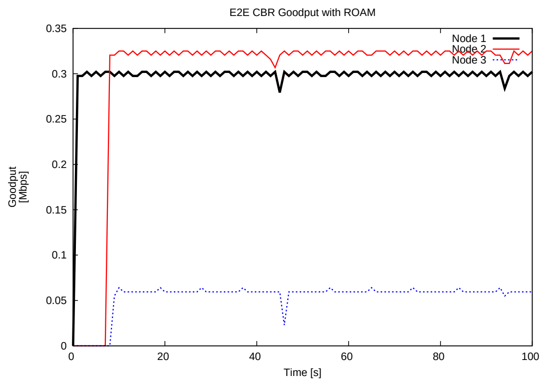
(c) AODV (Sub-scenario 2.2)



(d) ROAM (Sub-scenario 2.2)

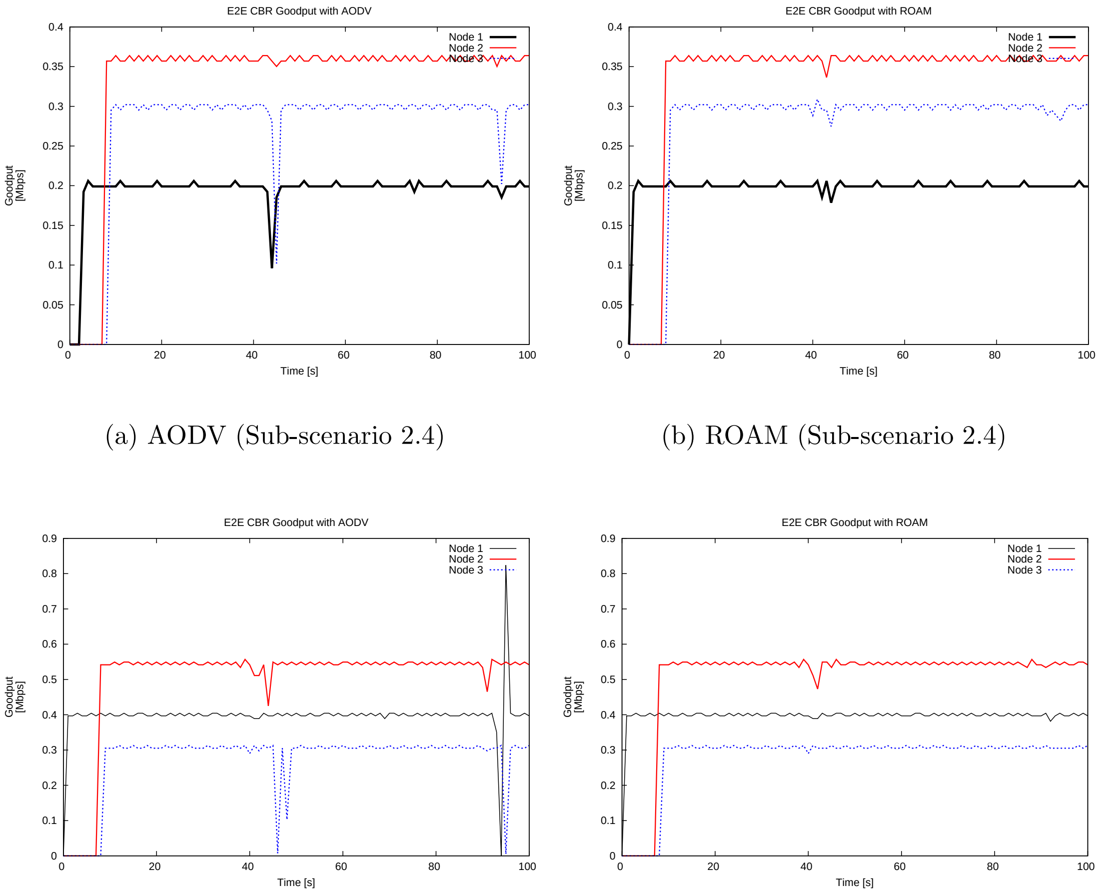


(e) AODV (Sub-scenario 2.3)



(f) ROAM (Sub-scenario 2.3)

Figure 5.7: Scenario 2: Instantaneous E2E CBR Goodput (AODV versus ROAM)



(a) AODV (Sub-scenario 2.4) (b) ROAM (Sub-scenario 2.4)

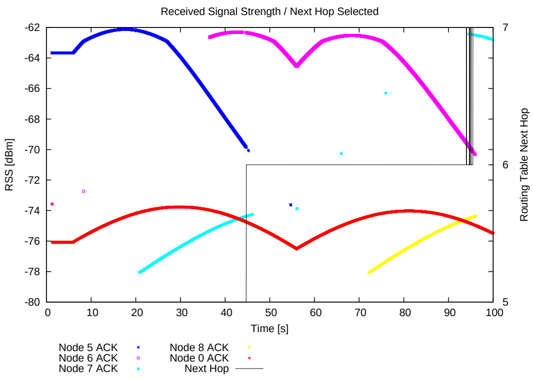
(c) AODV (Sub-scenario 2.5) (d) ROAM (Sub-scenario 2.5)

Figure 5.8: Scenario 2: Instantaneous E2E CBR Goodput (AODV versus ROAM)

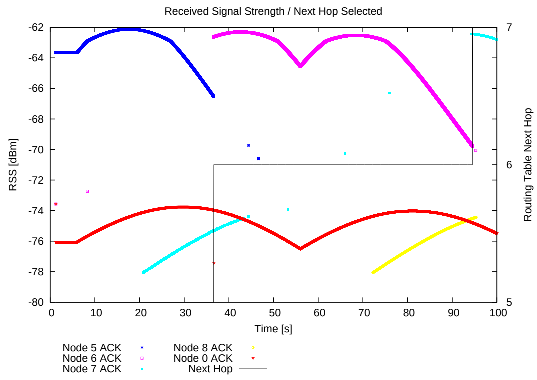
maximum delay for that node. With multiple nodes present, optimal handoff for one stream can result in earlier incidence of contention for a shared channel with a neighbouring transmitter, creating congestion. Therefore, success of AODV is coincidental to the topology in the scenario. If multiple nodes can contend for a fading link with AODV then this would not give the best system performance.

When flows of differing packet size and transmission rate traverse a MANET, bandwidth requirements differ across the network and network jitter is elevated as enqueueing, dequeueing and transmission delays become more varied. The addition of competing CBR flows changes the levels of contention for shared E2E paths. Contention induced delay is a key component of E2E delay, which must be bounded to provide guaranteed performance to ISRT traffic in MANETs.

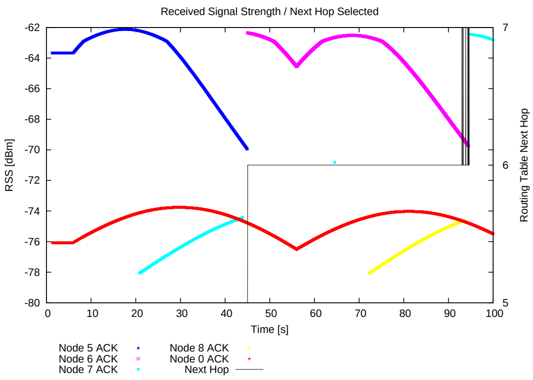
With multiple nodes in range of each other, increased retransmissions and greater channel busy time force transmitters to repeatedly backoff and negotiate wireless channel access before transmitting. By ensuring rapid handoff, when nodes are not competing for the same optimal channels, ROAM is capable of



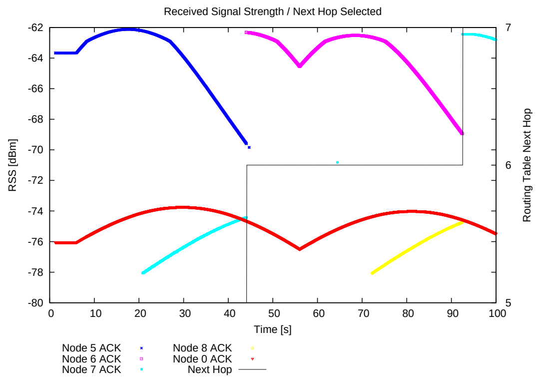
(a) AODV (Sub-scenario 2.1)



(b) ROAM (Sub-scenario 2.1)

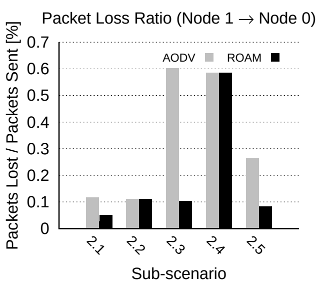


(c) AODV (Sub-scenario 2.4)

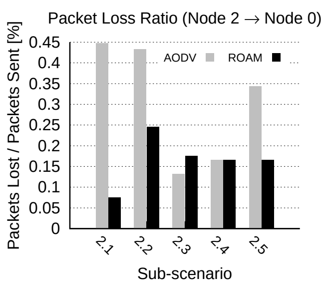


(d) ROAM (Sub-scenario 2.4)

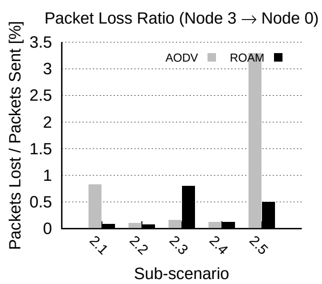
Figure 5.9: Scenario 2: Next Hop Selected by Node 1 (AODV versus ROAM)



(a) N1 → N0

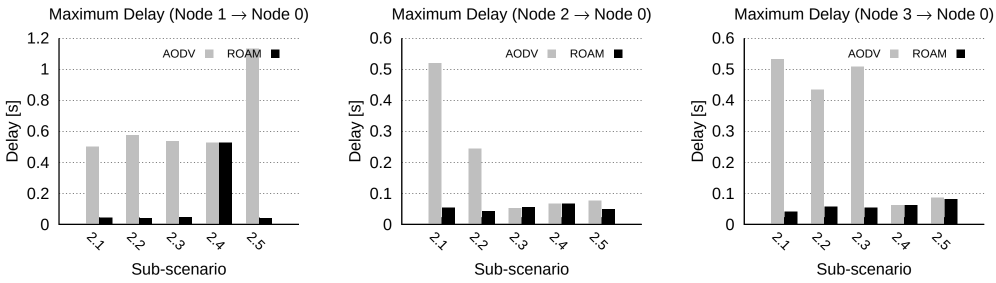


(b) N2 → N0



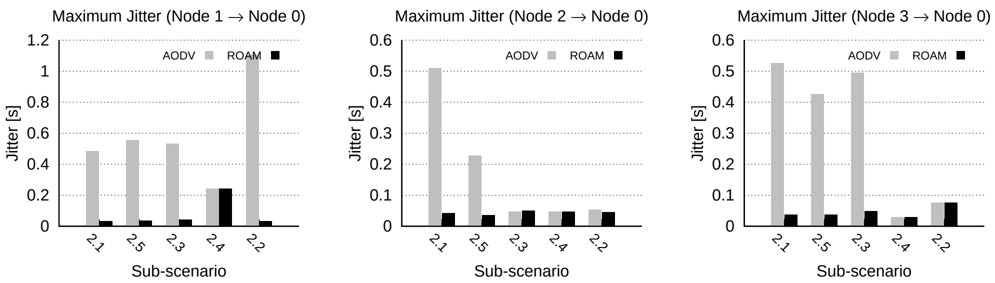
(c) N3 → N0

Figure 5.10: Scenario 2: CBR Packet Loss Ratio (AODV versus ROAM)



(a) N1 → N0 (b) N2 → N0 (c) N3 → N0

Figure 5.11: Scenario 2: Maximum E2E Delay at Handoff (AODV versus ROAM)



(a) N1 → N0 (b) N2 → N0 (c) N3 → N0

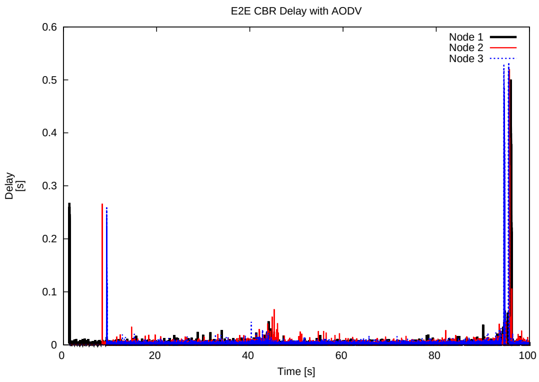
Figure 5.12: Scenario 2: Maximum E2E Jitter at Handoff (AODV versus ROAM)

reducing maximum E2E delay and ensuring that peak delay is only associated with path setup rather than link handoff. Figures 5.13–5.14 show this capability in four of the five scenarios.

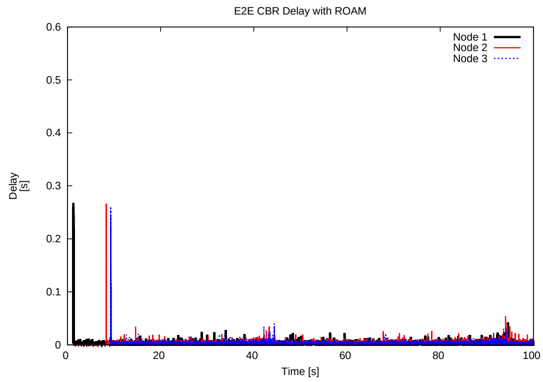
In sub-scenario 2.4 the performance of AODV and ROAM were similar, where Node 1, containing ROAM, had the lowest transmission rate, of 0.2Mbps. This result indicates the limitation of ROAM handoff. At very low datarates, the optimiser cannot gather sufficient information on channels in range in order to institute handoff that is faster than with AODV. Therefore, the level of performance does not significantly improve on that of the underlying routing protocol.

Figures 5.11–5.12 show that with multiple transmitters and varying total network load, maximum delay and jitter were generally constrained to below 0.1s for Node 1, for the remaining scenarios. For all CBR sources in the majority of scenarios, delay and jitter were also constrained or a negligible performance difference was observed. Nodal delay is bounded as a result of adaptive link selection and fast handoff when ROAM is enabled.

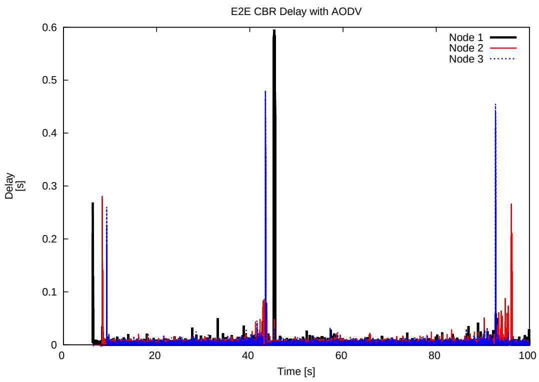
Figure 5.9(a) compares the Received Signal Strength (RSS) of RREPs and link layer control frames at the ROAM node. When the ROAM optimiser is not



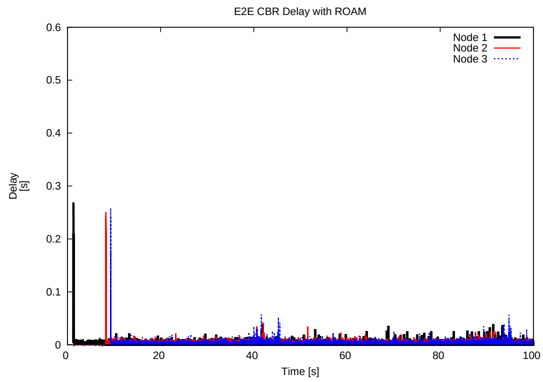
(a) AODV (Sub-scenario 2.1)



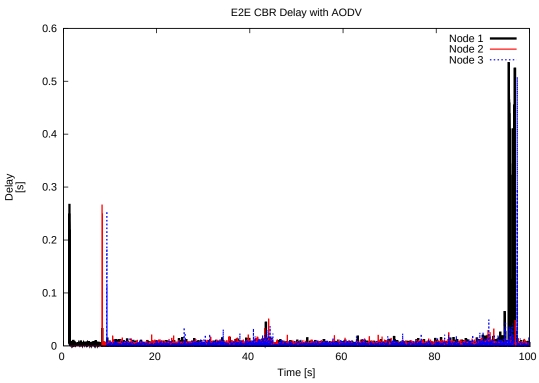
(b) ROAM (Sub-scenario 2.1)



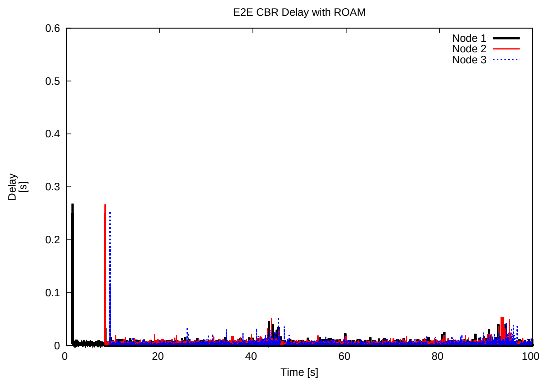
(c) AODV (Sub-scenario 2.2)



(d) ROAM (Sub-scenario 2.2)

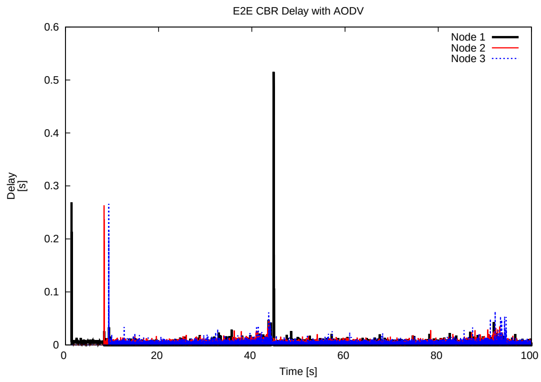


(e) AODV (Sub-scenario 2.3)

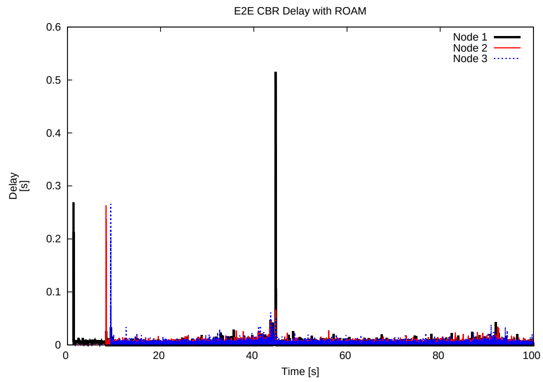


(f) ROAM (Sub-scenario 2.3)

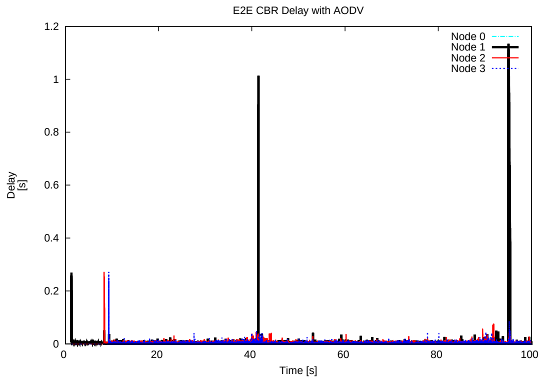
Figure 5.13: Scenario 2: Instantaneous E2E Delay (AODV versus ROAM)



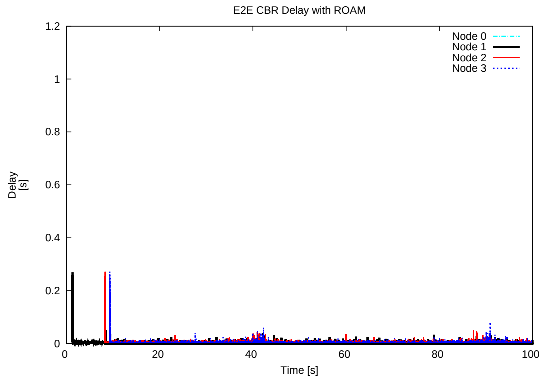
(a) AODV (Sub-scenario 2.4)



(b) ROAM (Sub-scenario 2.4)



(c) AODV (Sub-scenario 2.5)



(d) ROAM (Sub-scenario 2.5)

Figure 5.14: Scenario 2: Instantaneous E2E Delay (AODV versus ROAM)

enabled, packets are received from node 6 at 35s into the simulation, however, the routing protocol continues to transmit packets via Node 5. Additionally, during the handoff from forwarding node 6 to 7, the current next hop is repeatedly changed and packets are sent to both nodes 6 and 7. This is in spite of the fact that node 6 is almost out of transmission range. With ROAM, optimal handoff occurs and the fading link is marked as out of range. In contrast, in scenario 2.4, while switching of next hop selection is avoided by ROAM, handoff timing is not significantly altered.

Maximum jitter results if contiguous packets experience significantly different delay, as is common in a multi-hop, multi-path network. MANET variation in enqueueing and dequeueing of packets and busy time on shared channels leads to increased jitter and the likelihood of packet dropping due to excess of the TTL. ROAM can improve local channel selection but does not influence network-wide path selection decisions. Channel quality therefore varies on a hop-by-hop basis but, irrespective of these flow states, ROAM instead refers to routing protocol path selection and MAC control packet receipt to identify low quality channels and avoid these, removing isolated increases in jitter at handoff.

5.2.2 Testing MANET Scenario Independence

ROAM has been developed in order to improve the performance of safety-critical applications in MANETs. The design and development of these applications has informed the structure of the ROAM architecture. However, in being ISRT, these applications have specific requirements from a MANET which should be provided irrespective of the conditions on the network. The previous sections have validated the independence and scalability of the optimiser under the range of conditions created by variation in application settings.

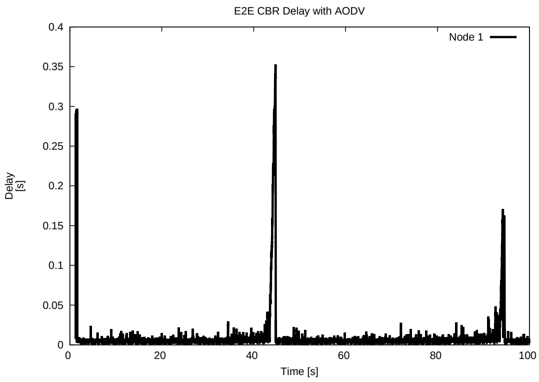
The following simulations demonstrate that ROAM is also capable of constraining maximum delay and jitter when the MANET itself varies due to changes in topology, number of sources on the network and node speeds. All of these contributing factors change the nodal requirements from handoff as well as the contention and interference levels on different links. Multiple CBR and VoIP sources were added to the MANET in scenario 1, with CBR background traffic. ROAM is also validated in different topologies and under varying mobile node speeds, creating rapid topology changes, so performance is reliant on low levels of processing delay.

5.2.2.1 Scenario 1(a): Variable Number of Sources: CBR

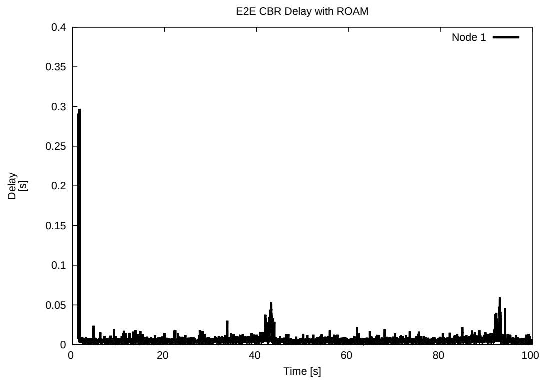
In Scenario 1(a) an increasing number of nodes transmitting CBR flows were added to the network. The same simulation topology and configuration was used as in previous scenarios with subsequent transmitters orbiting the network with a separation of 250m. In order to fully examine the influence of CBR source count on the network and ROAM, total network load was maintained at 1.5 Mpbs in all

Table 5.3: Scenario 1(a): Overall Packet Drop Comparison

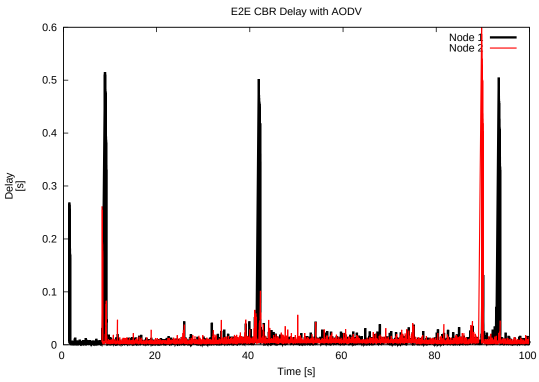
	AODV			ROAM		
N. CBR Sources	Collision Count	IFQ Full	No Route	Collision Count	IFQ Full	No Route
1	80	8460	7	21	8405	2
2	92	10053	469	28	10009	280
3	62	11598	668	23	11318	368
4	90	13251	764	47	12553	657
5	58	15376	799	60	15880	643



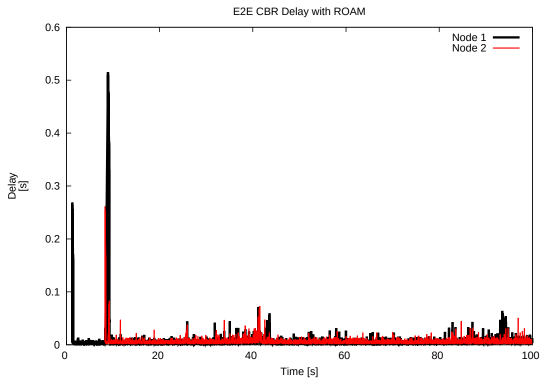
(a) AODV (CBR Sources = 1)



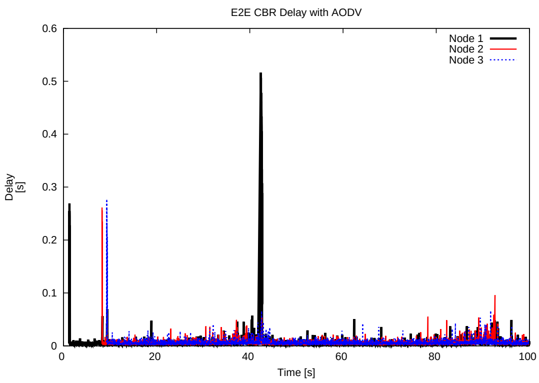
(b) ROAM (CBR Sources = 1)



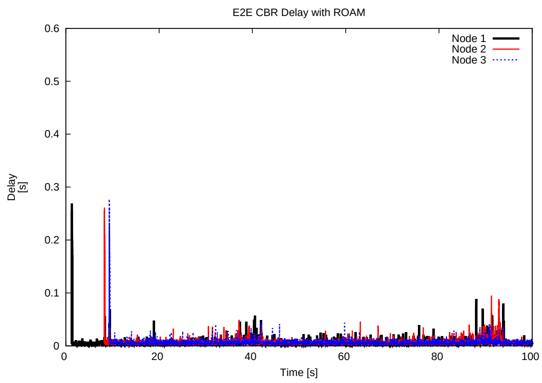
(c) AODV (CBR Sources = 2)



(d) ROAM (CBR Sources = 2)

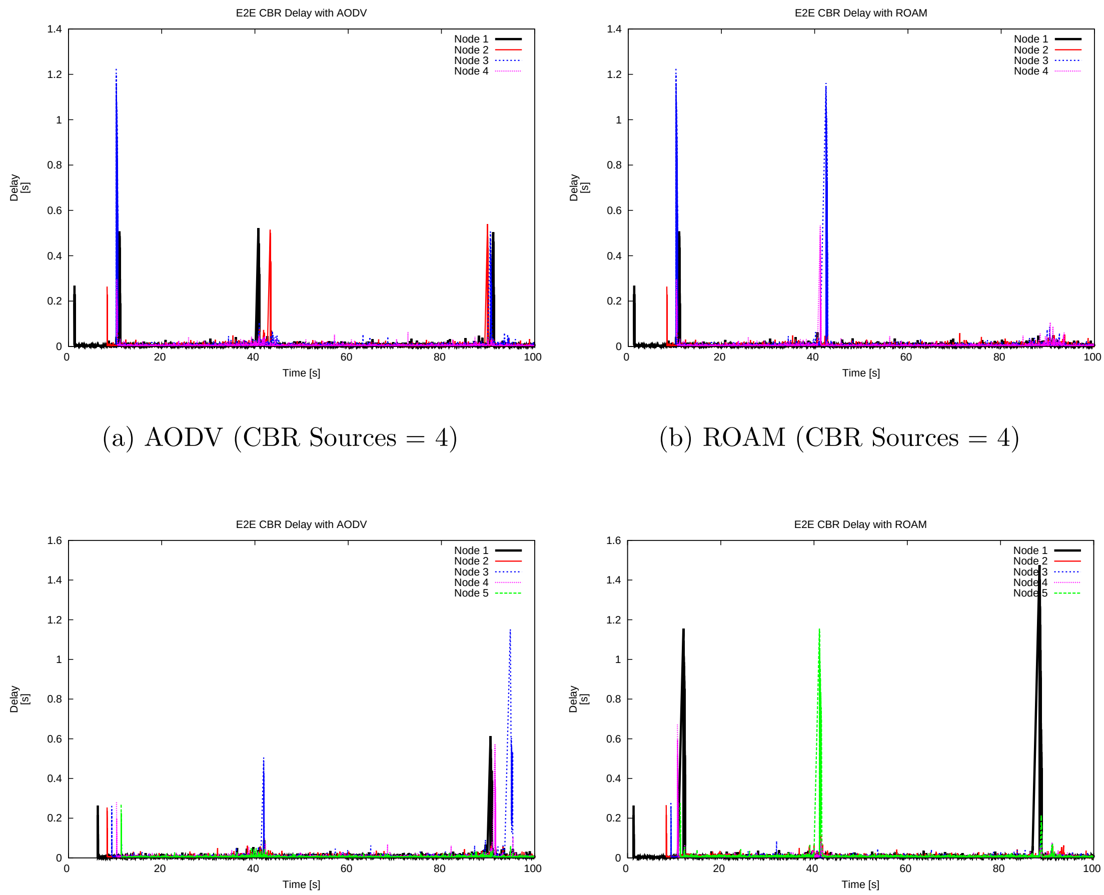


(e) AODV (CBR Sources = 3)



(f) ROAM (CBR Sources = 3)

Figure 5.15: Scenario 1(a): Instantaneous E2E Delay (AODV versus ROAM)



(a) AODV (CBR Sources = 4) (b) ROAM (CBR Sources = 4)

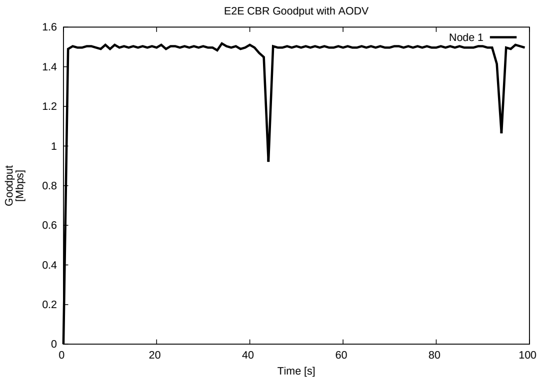
(c) AODV (CBR Sources = 5) (d) ROAM (CBR Sources = 5)

Figure 5.16: Scenario 1(a): Instantaneous E2E Delay (AODV versus ROAM)

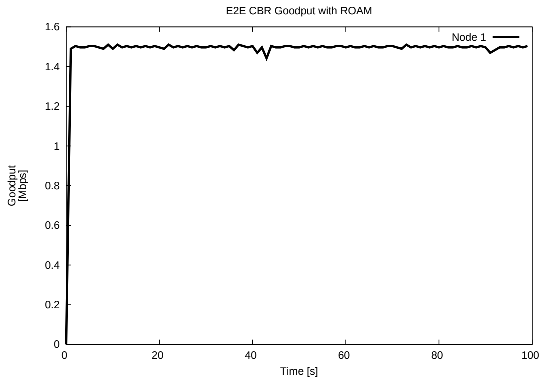
sub-scenarios.

The peripheral topology of the MANET is dynamically changed by each handoff required by a CBR source, as these can also act as forwarding nodes and the simulation details are given in Chapter 3. The purpose of this scenario is to demonstrate that ROAM improves performance while all of these transmitters compete for channel access and E2E paths to the receiver. CBR flows require the most stringent QoS from a network by both transmitting and requiring receipt of a consistent stream of packets. As the network tends towards saturation, with control and data packets, the need to handoff in a timely manner increases in order to avoid congestion.

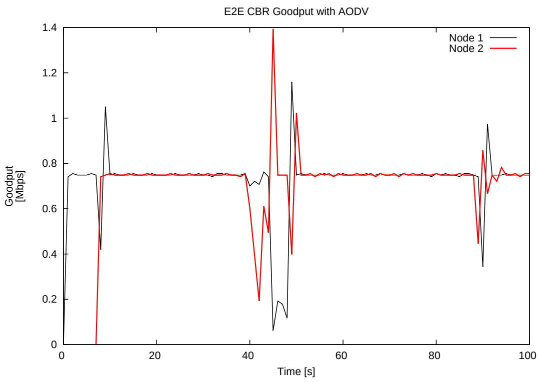
Each additional CBR flow increased competition for E2E paths, and by transmitting to a single receiver this increased the potential for bottlenecks at the last hop. Due to the load reduction for each node as a new source was added, overall collisions reduced in each subsequent sub-scenario, when CSMA was implemented (table 5.3) as a result of the lower traffic load at each source, but overall packet



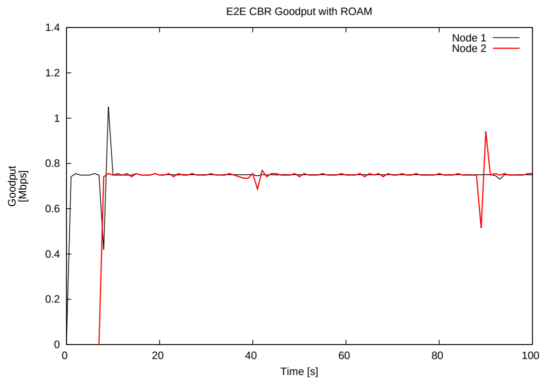
(a) AODV (CBR Sources = 1)



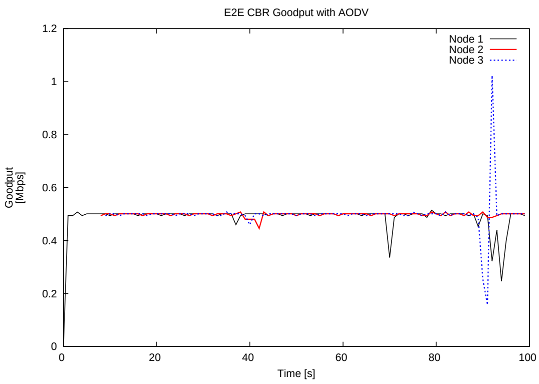
(b) ROAM (CBR Sources = 1)



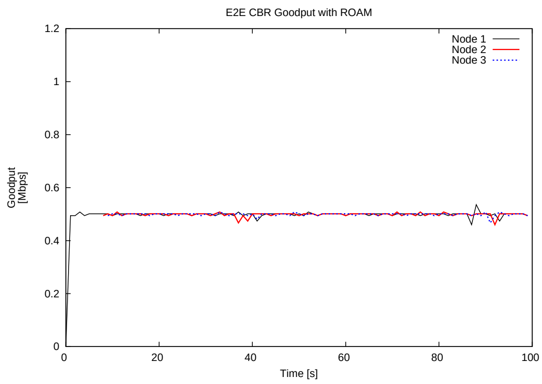
(c) AODV (CBR Sources = 2)



(d) ROAM (CBR Sources = 2)

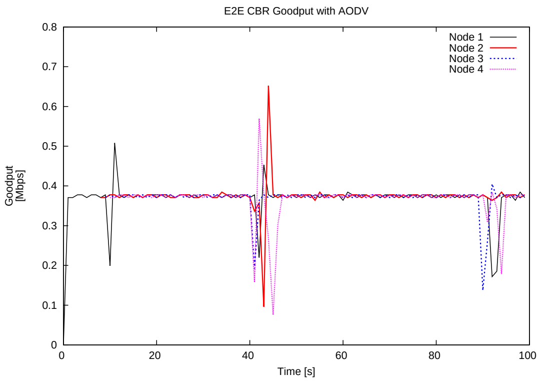


(e) AODV (CBR Sources = 3)

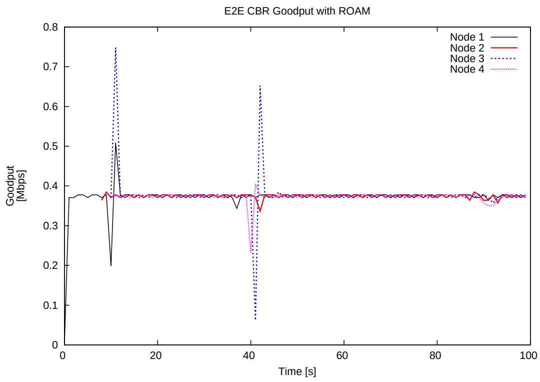


(f) ROAM (CBR Sources = 3)

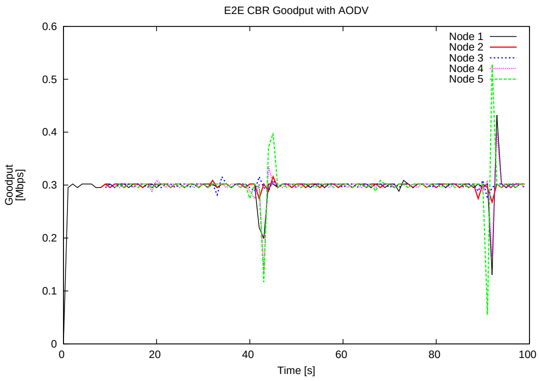
Figure 5.17: Scenario 1(a): Instantaneous E2E CBR Goodput (AODV versus ROAM)



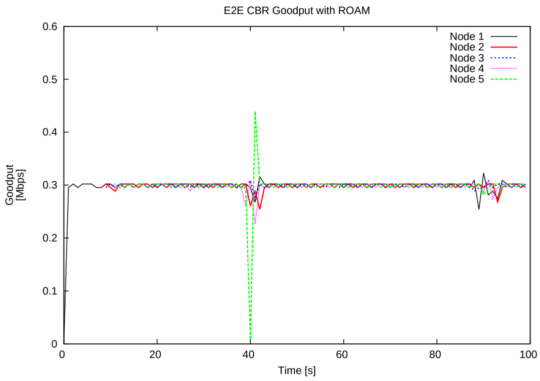
(a) AODV (CBR Sources = 4)



(b) ROAM (CBR Sources = 4)



(c) AODV (CBR Sources = 5)



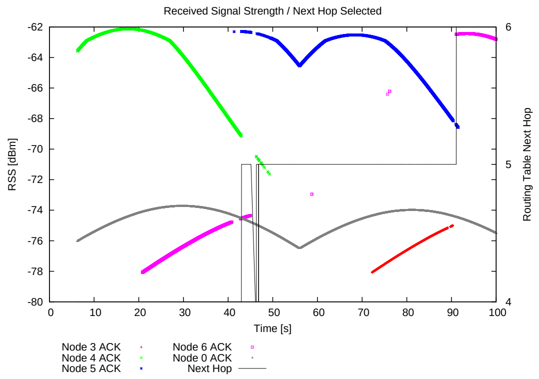
(d) ROAM (CBR Sources = 5)

Figure 5.18: Scenario 1(a): Instantaneous E2E CBR Goodput (AODV versus ROAM)

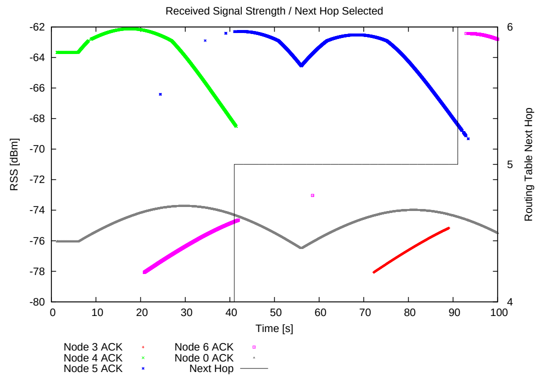
loss ratios increased. This corresponded to the rise in channel contention and the flood of routing control packets on the network.

With between 1-3 CBR sources present, performance improved for the ROAM node and other transmitters as rapid handoff curbed the rise in collisions by avoiding receding INs. Collisions and packet drops due to route incoherence were less prevalent for these scenarios. However, with more than three sources, AODV provided comparable or better performance. As more discrete transmitters are added to the network and with ROAM solely implemented in Node 1, punctual handoff increased the likelihood of two sources sharing the same forwarding node.

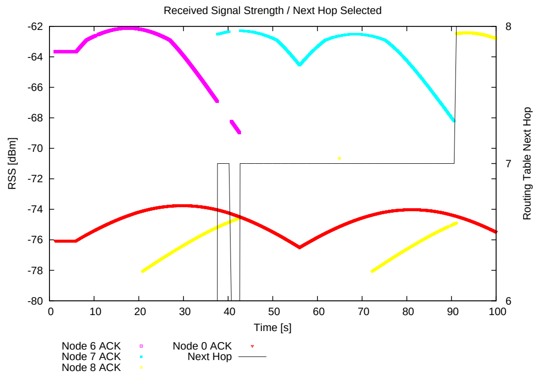
Therefore, when handing off to a link that was suboptimal for CBR sources 4-5, packet loss was increased for these nodes. As a result, collision counts increased with transmitter number. The varying E2E paths used by CBR flows also converge due to the shared receiver, increasing collisions in this locality. Figures 5.17–5.18 demonstrate the periods of low goodput surrounding AODV handoff that



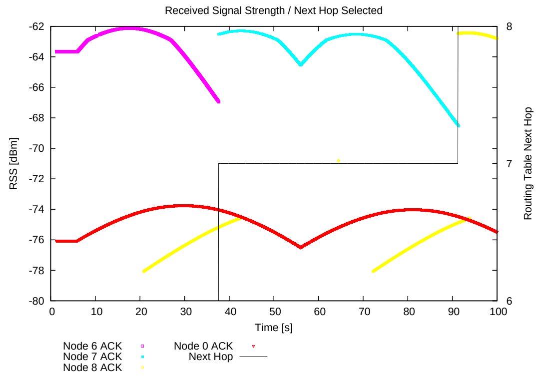
(a) AODV (CBR Sources = 2)



(b) ROAM (CBR Sources = 2)



(c) AODV (CBR Sources = 4)



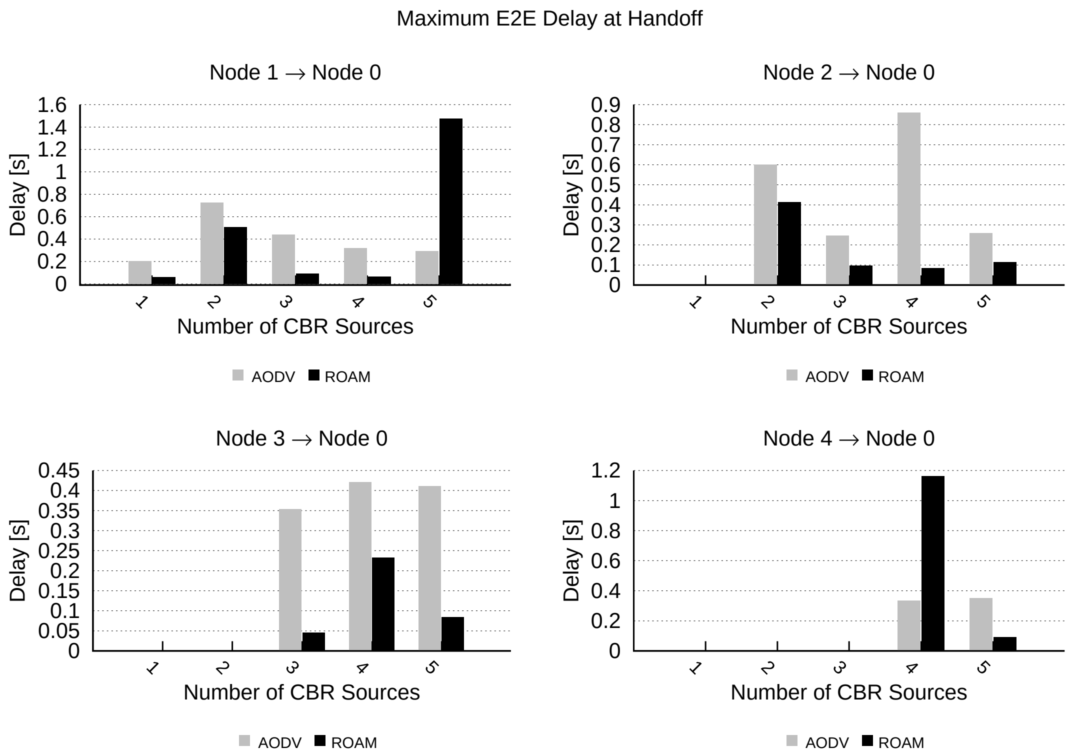
(d) ROAM (CBR Sources = 4)

Figure 5.19: Scenario 1(a): Next Hop Selected by Node 1 (AODV versus ROAM)

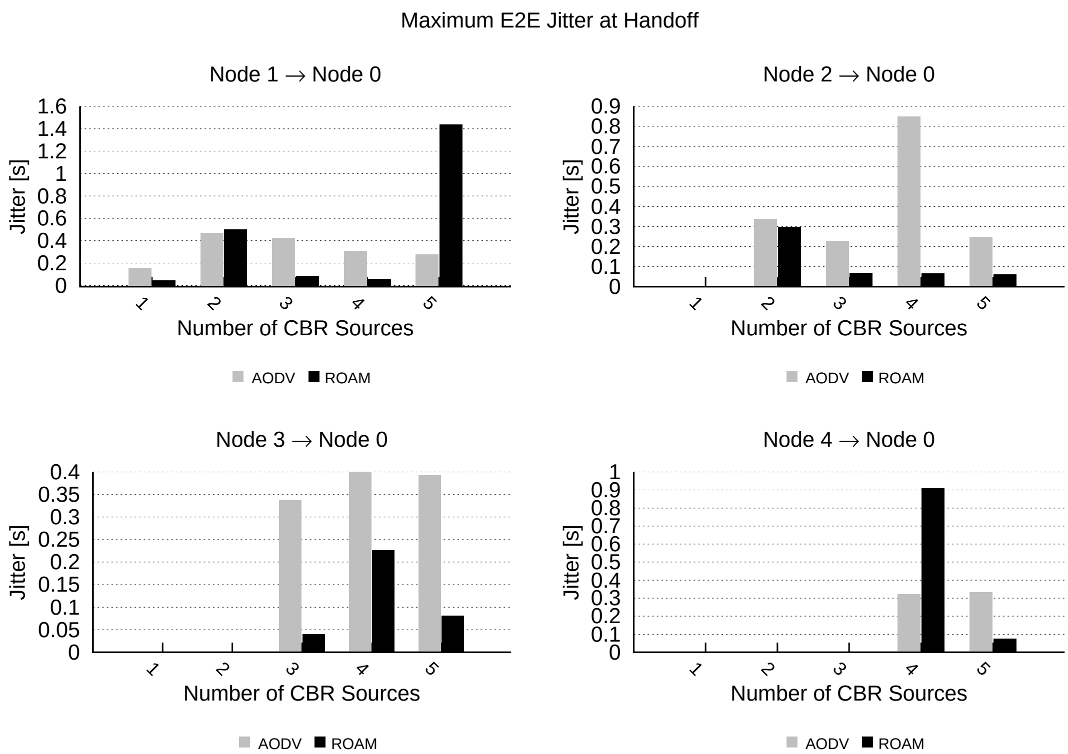
were significantly reduced for the ROAM node and through reduction in queueing and MAC layer loss recovery, this resulted in lower packet delay during handoff (figures 5.15–5.16).

With more flows on the network, as shown in figure 5.19, AODV in Node 1 repeatedly switched between fading and optimal paths before complete handoff occurred. This is a result of the control packet exchange characteristics of wireless ad hoc protocols. Greater circulation of control packets filtering through the network from multiple sources should ideally improve the freshness of routing information when all transmitting nodes use the same receiver. However, if link layer frames are then intercepted on a suboptimal link, this can still induce AODV to update the routing table with this node as the current next hop. While handoff was marginally faster with ROAM than AODV, switching in next hop selection was prevented.

Corresponding to reduced fading path usage, figure 5.21 shows that packet loss ratios were lower with ROAM than AODV even with multiple CBR streams

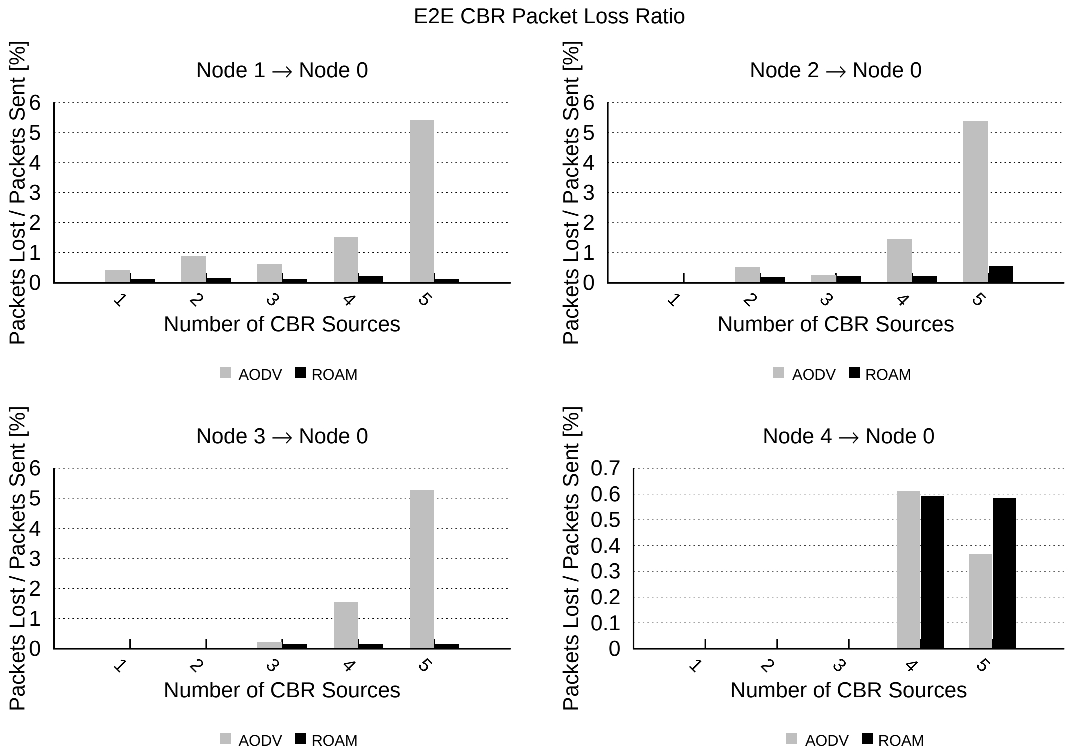


(a) Maximum E2E Delay at Handoff



(b) Maximum E2E Jitter at Handoff

Figure 5.20: Scenario 1(a): Delay and Jitter Comparison (AODV versus ROAM)



(a) E2E CBR Packet Loss Ratio

Figure 5.21: Scenario 1(a): Packet Loss Ratio Comparison (AODV versus ROAM)

traversing the network in sub-scenarios 1-3. Nodes 4 and 5 were subject to packet loss increases as each subsequently added transmitter was located nearest to the ROAM node. Maximum E2E delay and jitter were relatively consistent for each transmitter in all sub-cases (figure 5.20), in spite of variation in the number of CBR sources, as the result of the consistent network load and 2-hop E2E path for each source.

However, when five transmitters were present on the network, congestion at the receiver resulted in increased enqueueing of packets from node 1 as a result of improved handoff management. This corresponded to an overall rise in collisions

Table 5.4: Scenario 1(a): Performance for CBR Source 5

Metric	N5 → N0	
	AODV	ROAM
Maximum E2E Delay (s)	0.513	1.157
Maximum E2E Jitter (s)	0.484	1.143
CBR Packet loss ratio (%)	0.396	0.669

and IFQ overflow outside of the local 1-hop neighbourhood in which ROAM is capable of providing improved performance. The ability to bound maximum delay for multiple nodes with ROAM was demonstrated with less than four transmitters, when the network is not saturated.

Table 5.5: Scenario 1(b): Overall Packet Drop Comparison

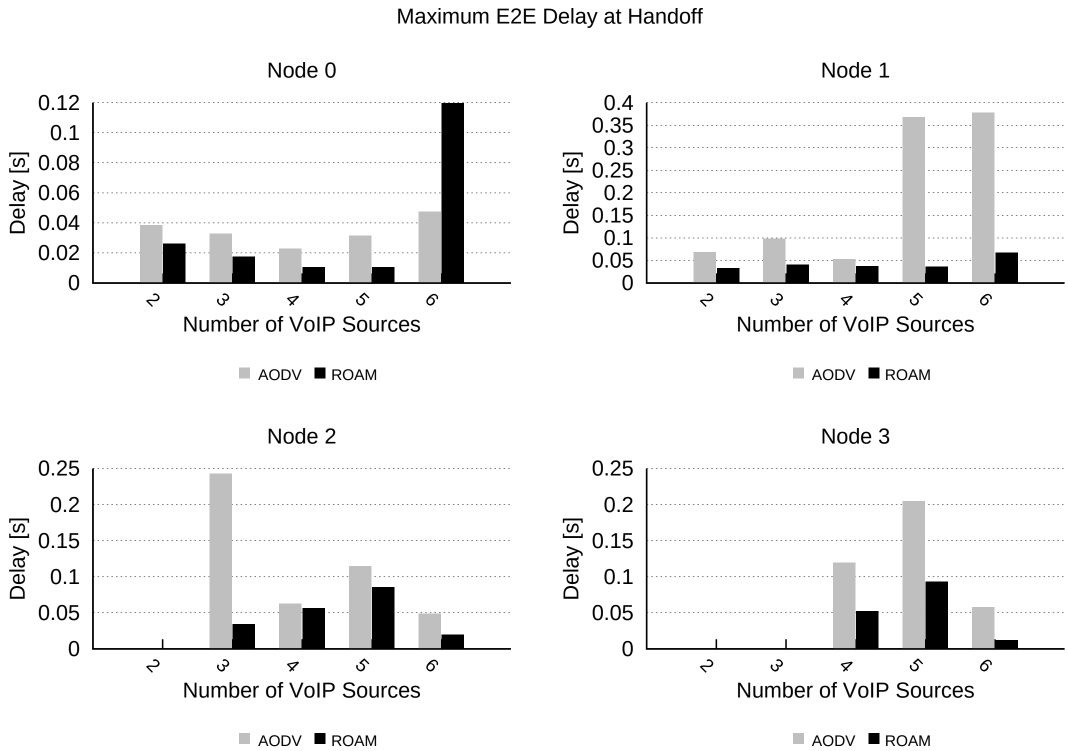
	AODV			ROAM		
N. VoIP Sources	Collision Count	IFQ Full	No Route	Collision Count	IFQ Full	No Route
1	34	8219	40	8	8026	7
2	85	10202	60	42	10197	34
3	153	13113	124	139	13839	116
4	201	15136	232	250	12978	157
5	311	16391	304	327	15265	249

5.2.2.2 Scenario 1(b): Variable Number of Sources: VoIP

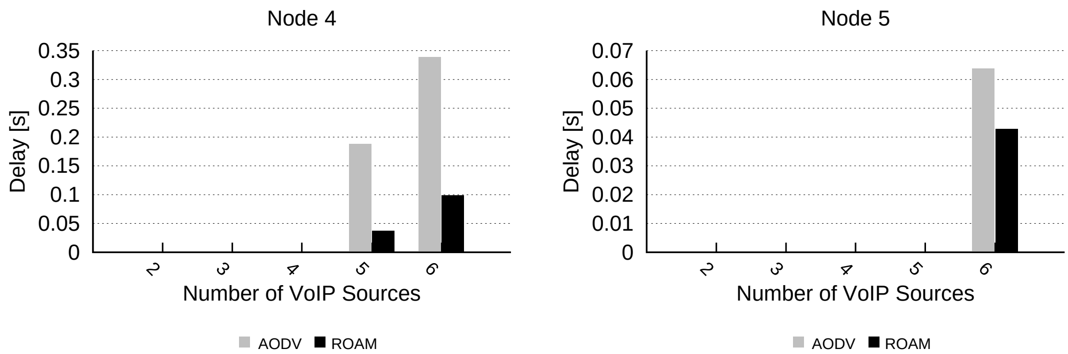
Multiple applications in disaster response and military network scenarios will be considered to be high priority, therefore, while CBR QoS requirements from a network are stringent it is expected that bounded delay and loss guarantees are provided to concurrent VoIP streams. After considering the simulation case with multiple CBR flows, this scenario investigates the performance outcomes with a heterogeneous network of multiple bidirectional VoIP flows over RTP and CBR background traffic of 0.5Mbps over UDP. This was to validate the overarching nature of the previous results and demonstrate a capacity to be ported to network protocol stacks using different application layer technologies.

These one-to-one VoIP sources use a variable traffic pattern model that differs from CBR through the inclusion of intervals of uplink and / or downlink silence amid bursts of VBR transmissions. The number of VoIP sources was increased from one to five to evaluate performance under increased channel load and contention.

With AODV the occurrence of collisions, IFQ overflow and routing errors increased with number of transmitters, to a greater degree than in Scenario 1(a), due to the increased competition for resources with bidirectional traffic (table 5.5). Competition for medium access is more complex when traffic is bursty, arriving at inconsistent rates at forwarding nodes, and backoff and retransmission can have a greater impact on E2E delivery. A sudden increase in traffic rate is more likely to overload IFQs in forwarding INs. Therefore, in spite of the low bandwidth requirements of the VoIP sources, when compared to the results with CBR flows



(a)

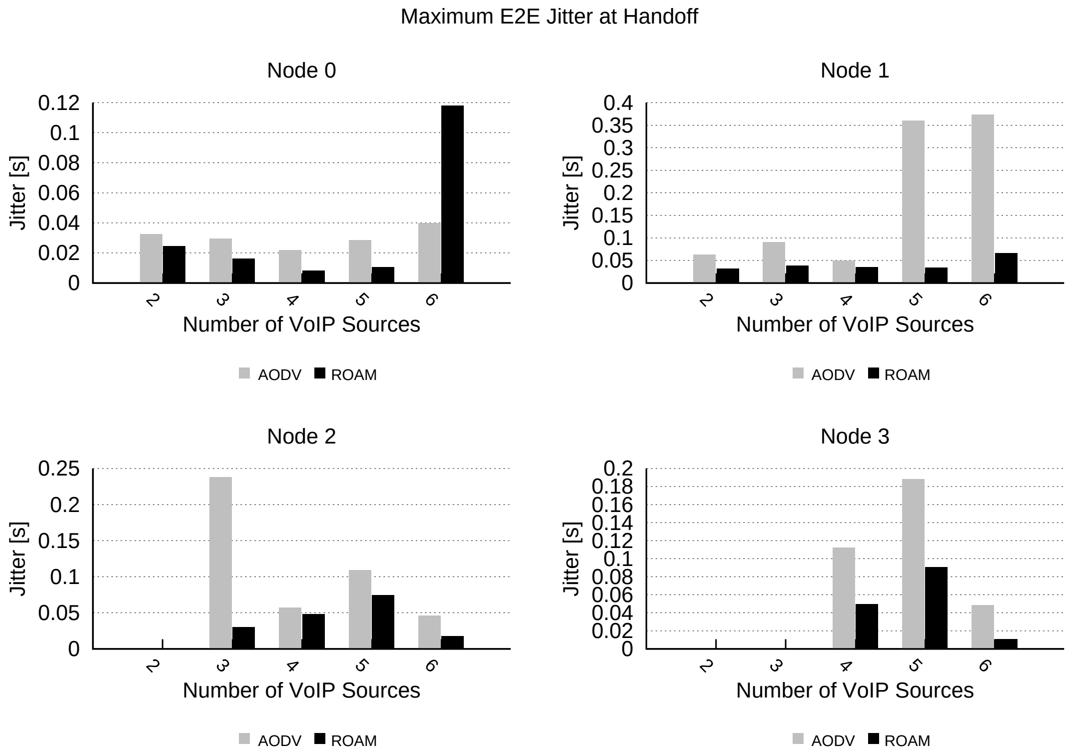


(b)

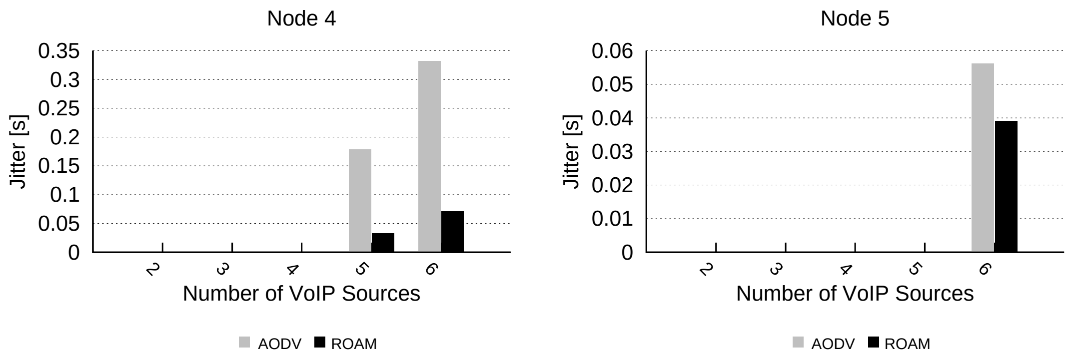
Figure 5.22: Scenario 1(b): Maximum E2E Delay at Handoff (AODV versus ROAM)

over AODV, buffer overflow was similar but total packet loss for all transmitters was much higher.

Packet loss ratios rose rapidly with increasing bidirectional transmissions, although with two transmitters, the results of Scenario 1(a) were similar (figure 5.24). Network-wide collisions and routing errors were reduced by ROAM, but this had the most significant impact for Node 1, for which overall packet delivery was in-



(a)



(b)

Figure 5.23: Scenario 1(b): Maximum E2E Jitter at Handoff (AODV versus ROAM)

creased by up to 20%. However, packet delivery performance for the remaining VoIP sources was generally similar or less promising with ROAM implemented, than with AODV alone. This was due to the implementation of ROAM in only Node 1. This provided optimal handoff for this node but increased the opportunities for bursty Node 1 flows to compete for resources with other receding transmitters (figure 5.27). Therefore, goodput for Node 1 increased with ROAM

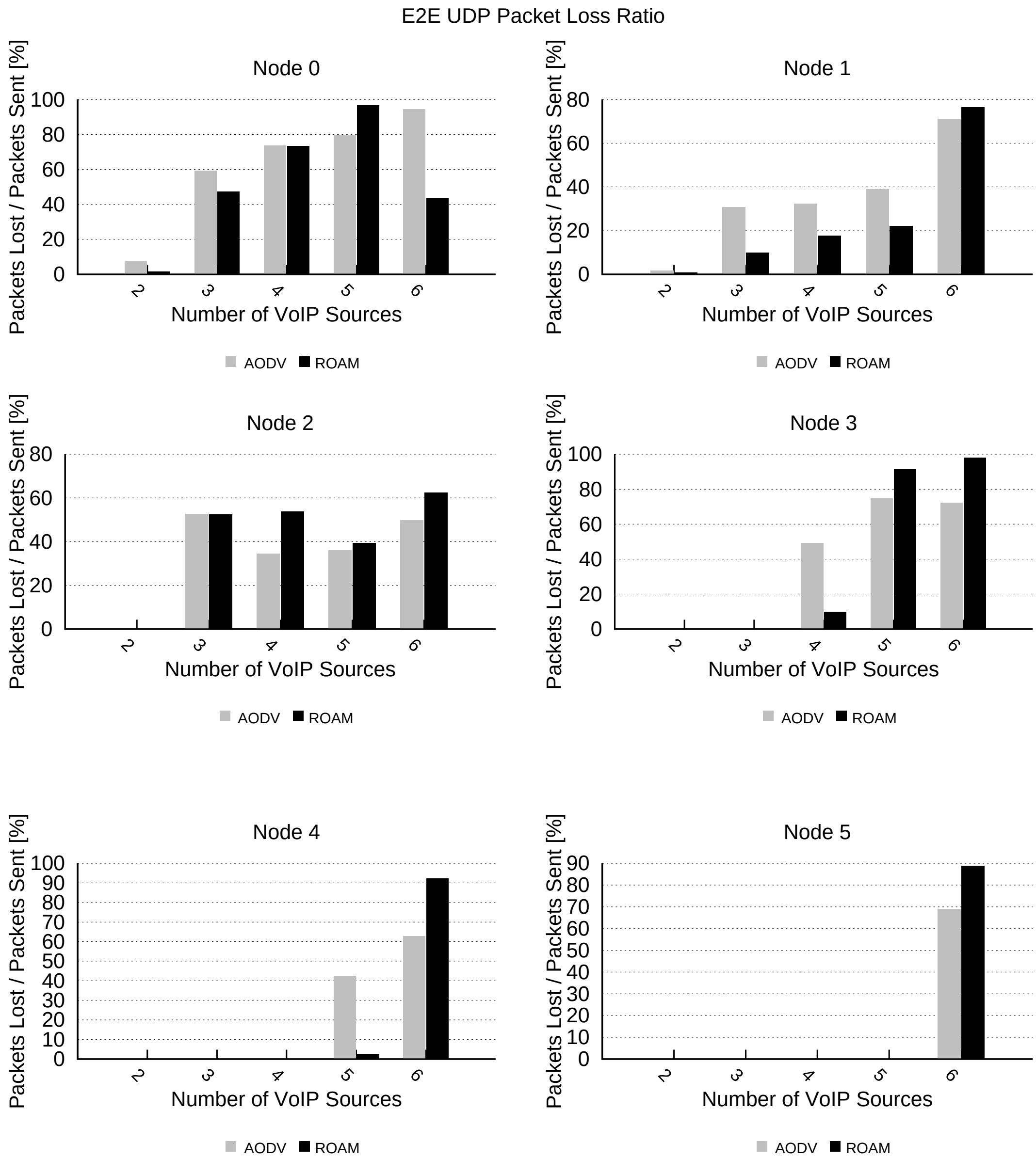
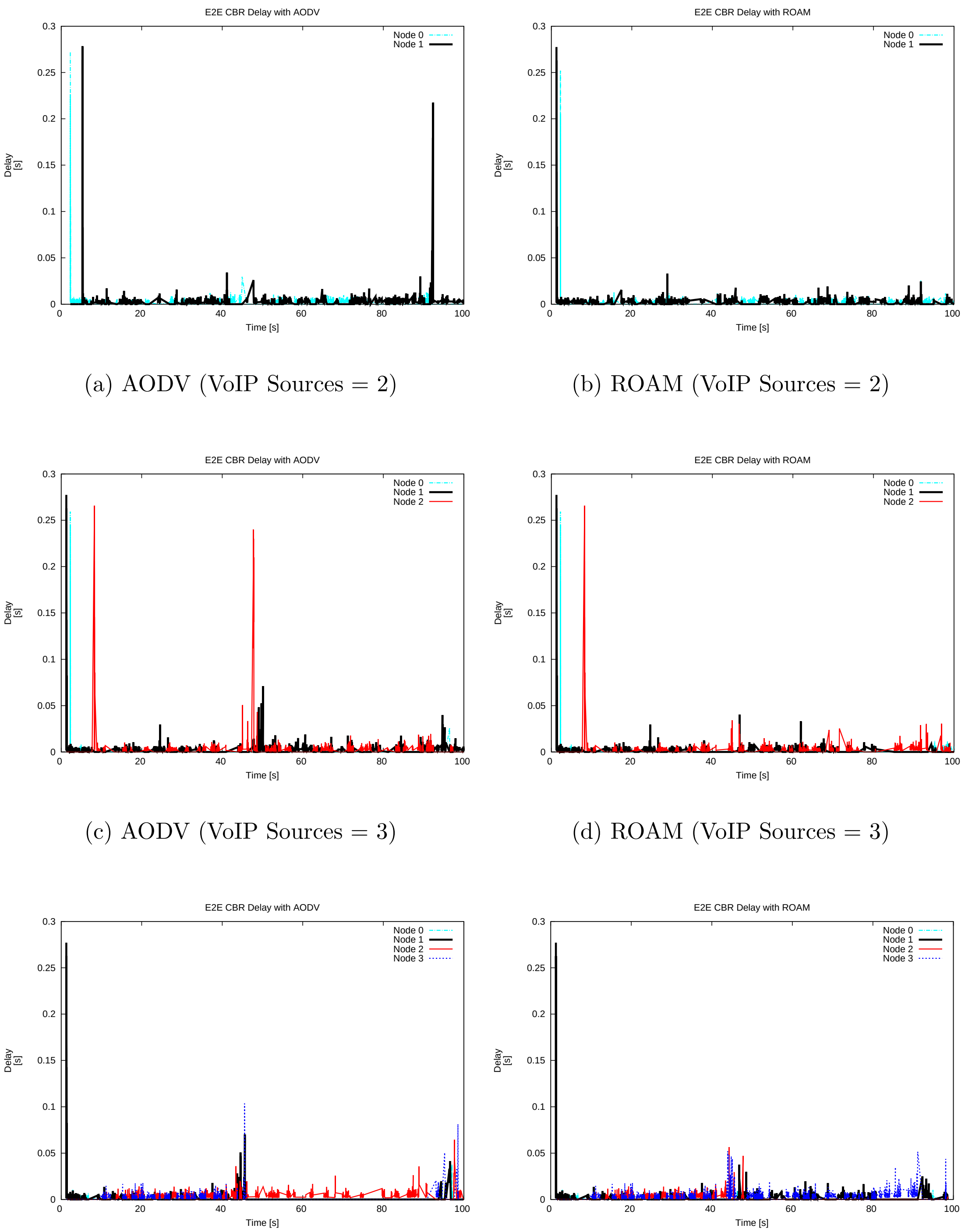


Figure 5.24: Scenario 1(b): UDP Packet Loss Ratio Comparison for AODV versus ROAM

in use, but was reduced E2E for other flows (figure 5.26).

In a multi-hop network, VBR flows are subject to variable contention delays, with related timeouts and backoff, which varies the time for which packets remain in IFQs along the E2E path. When six sources were present, ROAM handoff increased network congestion around the receiver. Repeated backoff and channel access competition results in throttling of bandwidth for multiple flows. However, figures 5.22–5.23 show that maximum

The higher power packets received from neighbouring transmitter Node 2 can be observed in figure 5.27(a), and from Nodes 4 and 5 in figure 5.27(b). Notably,



(a) AODV (VoIP Sources = 2)

(b) ROAM (VoIP Sources = 2)

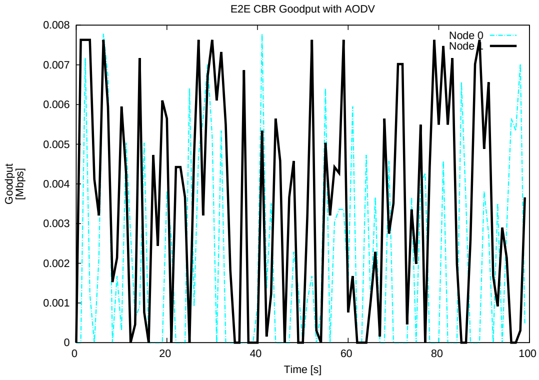
(c) AODV (VoIP Sources = 3)

(d) ROAM (VoIP Sources = 3)

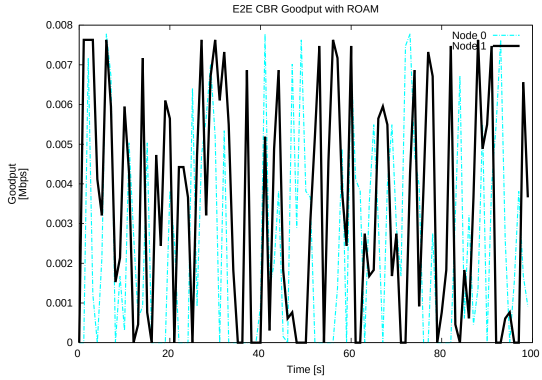
(e) AODV (VoIP Sources = 4)

(f) ROAM (VoIP Sources = 4)

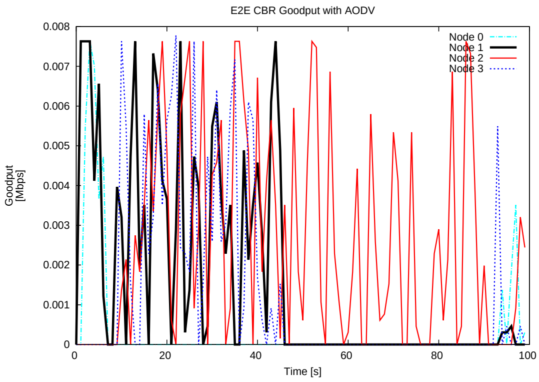
Figure 5.25: Scenario 1(b): Instantaneous E2E Delay (AODV versus ROAM)



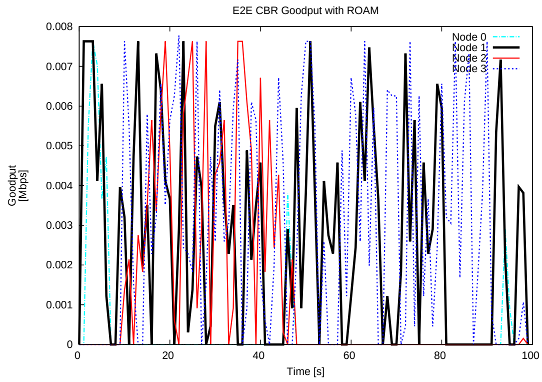
(a) AODV (VoIP Sources = 2)



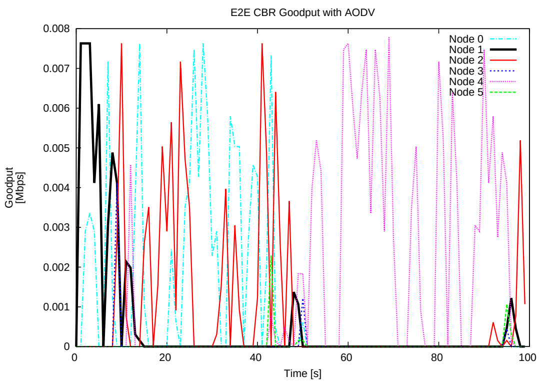
(b) ROAM (VoIP Sources = 2)



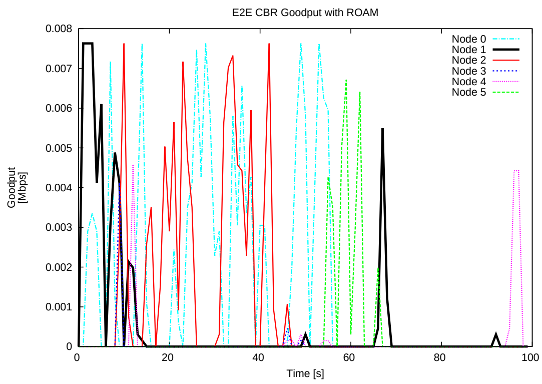
(c) AODV (VoIP Sources = 4)



(d) ROAM (VoIP Sources = 4)

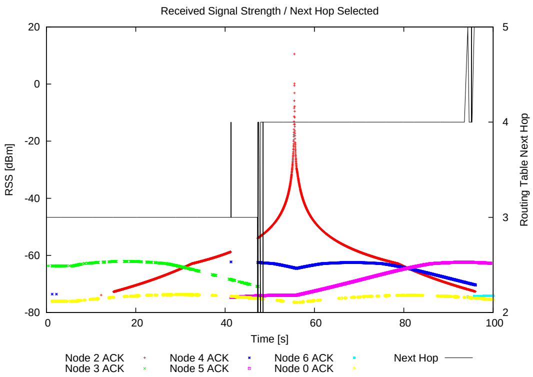


(e) AODV (VoIP Sources = 6)

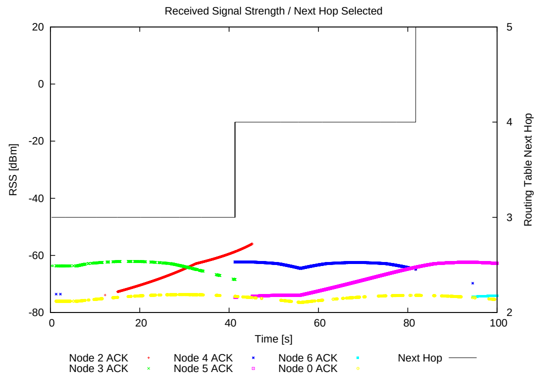


(f) ROAM (VoIP Sources = 6)

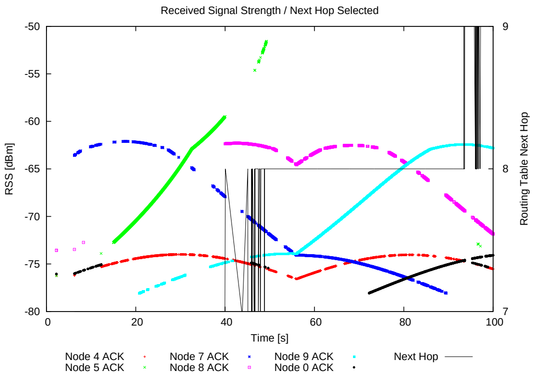
Figure 5.26: Scenario 1(b): Instantaneous E2E CBR Goodput (AODV versus ROAM)



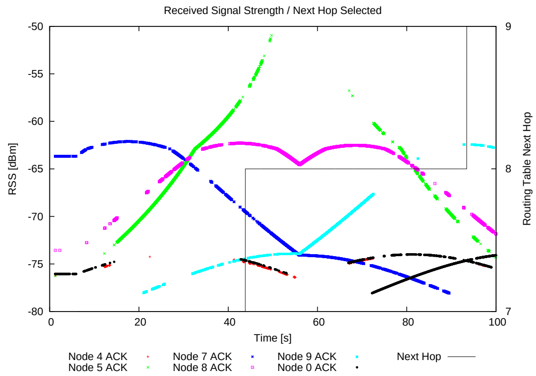
(a) AODV (VoIP Sources = 3)



(b) ROAM (VoIP Sources = 3)



(c) AODV (VoIP Sources = 6)



(d) ROAM (VoIP Sources = 6)

Figure 5.27: Scenario 1(b): Next Hop Selected by Node 1 (AODV versus ROAM)

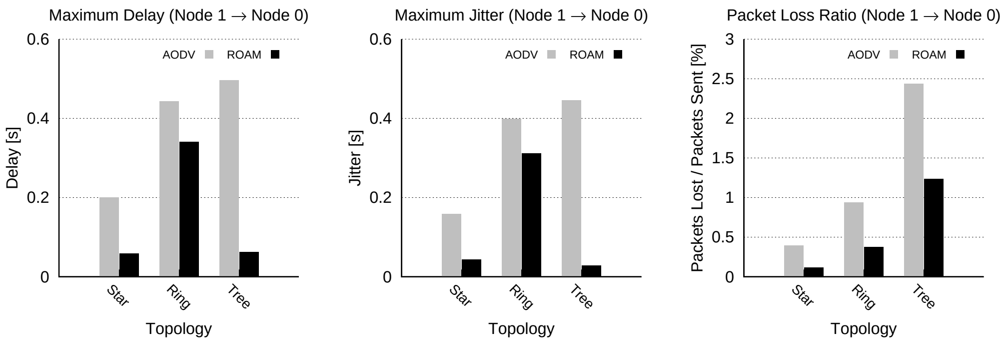
in figure 5.27(a), AODV in Node 1 repeatedly selects Node 2 as a forwarding next hop, creating increased packet delay as Node 2 continues to utilise Node 3 as a forwarding node that it is moving away from. ROAM ensures that a routing protocol does not select a link to a highly mobile node that is likely to have a low coherence time, based on the rate of change of RSS for that node. Therefore, in addition to ensuring rapid link selection and preventing next hop switching, delay is reduced for both the ROAM node and VoIP source 2 (figures 5.25).

5.2.2.3 Scenario 2: Different Topologies

ROAM utilises adaptive protocol parameter monitoring of local links in order to bound delay, packet loss and jitter, without reliance on particular topological arrangements of nodes. The middleware avoids network-wide signalling of global information, which becomes rapidly invalid in a dynamically changing MANET. Instead relative local information is acquired from control packets intercepted, and conditions at the MAC layer, for example RSS and MAC queue length are used

Table 5.6: Scenario 2: Overall Packet Drop Comparison

	AODV			ROAM		
Topology	Collision Count	IFQ Full	No Route	Collision Count	IFQ Full	No Route
Star	80	8460	7	21	8405	2
Ring	197	12213	78	81	9522	48
Tree	361	6772	25	113	6971	11



(a) Maximum E2E Delay at Handoff

(b) Maximum E2E Jitter at Handoff

(c) E2E CBR Packet Loss Ratio

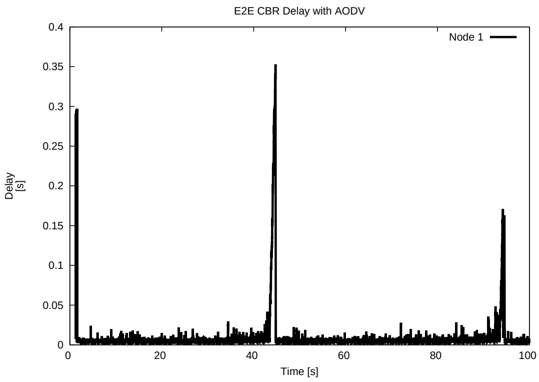
Figure 5.28: Scenario 2: Performance Comparison in Different Topologies

as part of the evaluation of link fading. RSS is used only to compare packets from multiple paths. Additionally, nodes with rapidly rising or falling RSS are assumed to be moving at high speed.

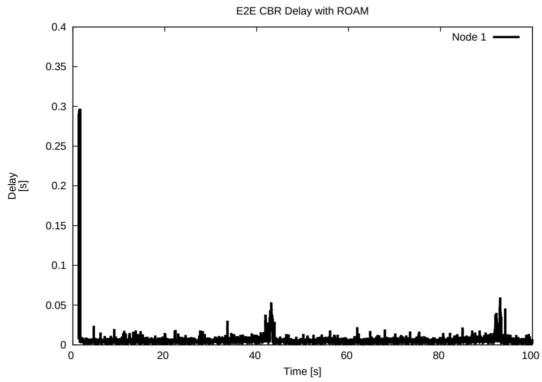
In this scenario ROAM has been evaluated with two novel topologies, not implemented in previous scenarios, that have varied mean shortest hop counts (HC) in order to show the scalability of the middleware architecture. The results are also compared to those with the star topology used in previous scenarios. These novel topologies are a tree topology (HC = 2.2) and ring topology (HC = 2.3), the configuration details of which are given in Chapter 3. A single mobile CBR source transmitted packets of 900B, with a traffic rate of 1.5Mbps.

Due to the size and structure of the tree topology, collisions were elevated when compared to the ring and star topologies. Within a star or ring topology more varied available paths exist and each node will have multiple 1-hop neighbours, raising the network congestion threshold and, depending on the distance between hops, resulting in fewer packet collisions.

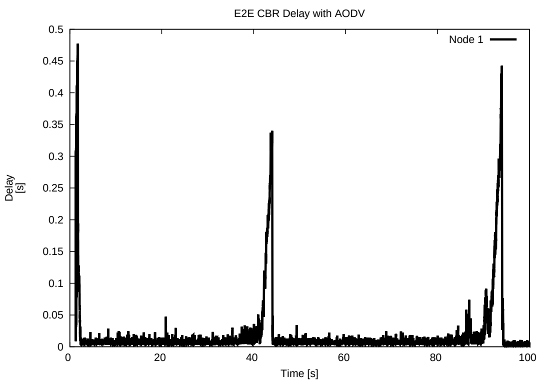
However, the ring topology has a drawback for a MANET transmitter: with a converging E2E path of increasing length, the performance of ad hoc routing degrades with each extra hop and the last hop becomes a bottleneck. Corre-



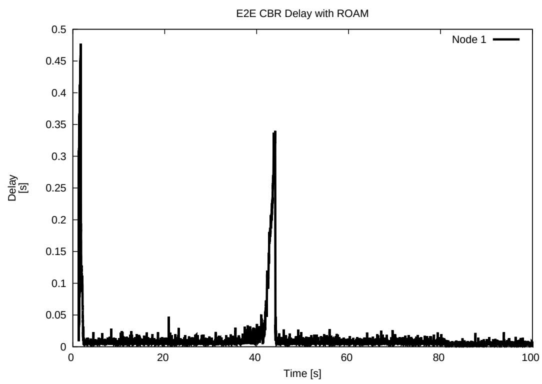
(a) AODV (Star Topology)



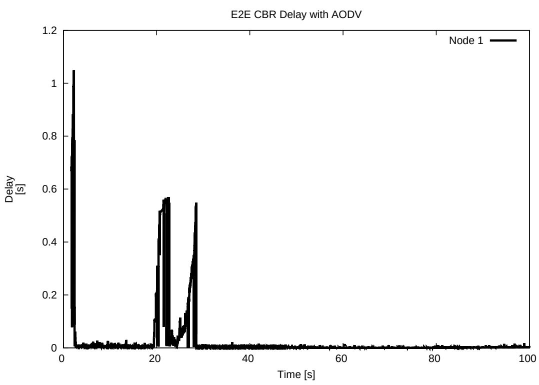
(b) ROAM (Star Topology)



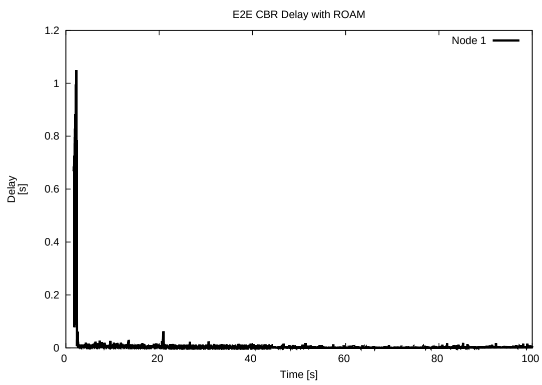
(c) AODV (Ring Topology)



(d) ROAM (Ring Topology)

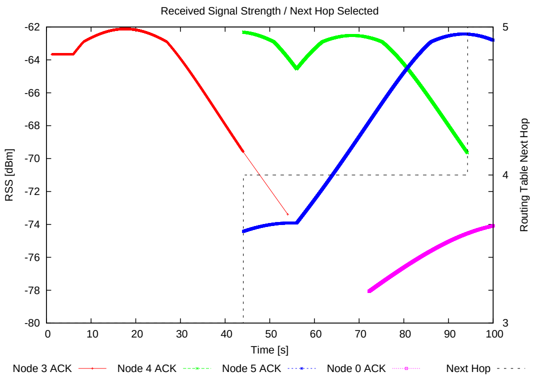


(e) AODV (Tree Topology)

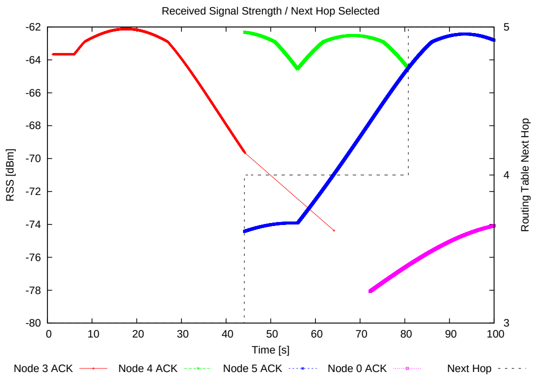


(f) ROAM (Tree Topology)

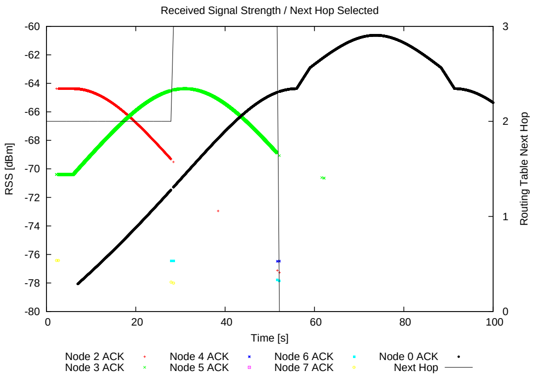
Figure 5.29: Scenario 2: Instantaneous E2E Delay (AODV versus ROAM)



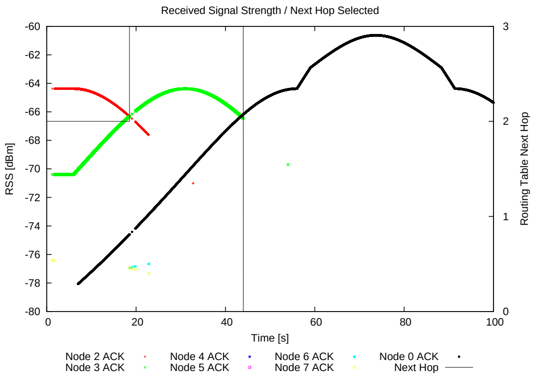
(a) AODV (Ring Topology)



(b) ROAM (Ring Topology)



(c) AODV (Tree Topology)

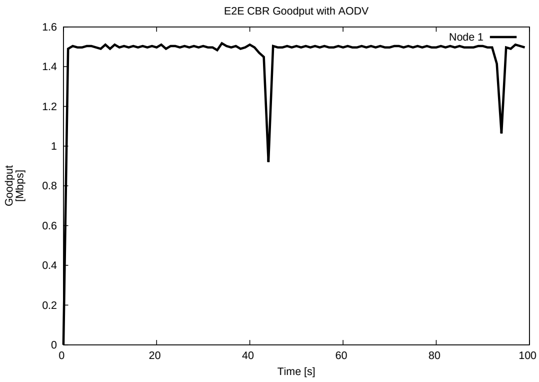


(d) ROAM (Tree Topology)

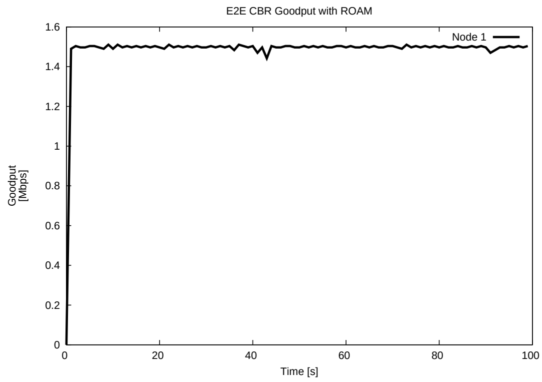
Figure 5.30: Scenario 2: Next Hop Selected by Node 1 (AODV versus ROAM)

spondingly, IFQ overflow at forwarding nodes is more prevalent in this topology. Table 5.6 and figure 5.28(c) show that collision counts and total packet loss with ROAM were reduced when compared to AODV, but that ROAM provided the best improvement in the tree topology, particularly in terms of collision reduction.

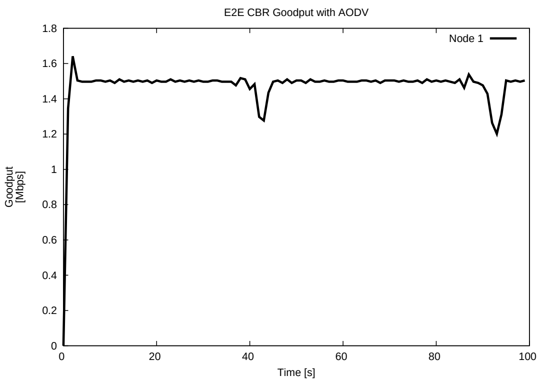
Figure 5.31 indicates that in the tree topology, goodput dropped almost to zero for a period of 10s, before rising to almost double the 1.5Mbps application rate, following a delayed handoff to node 3 (figure 5.30). Whereas, when handoff was expedited by almost 10s with ROAM, this degradation and rise did not occur. The high peak goodput seen with AODV is caused by increased buffering requirements following rise in collisions on the fading link. Subsequent handoff to a link with lower noise levels results in the MAC layer increasing the traffic rate and allowing the queue to drain rapidly. As a consequence, E2E delay was elevated for an extended period (figure 5.29).



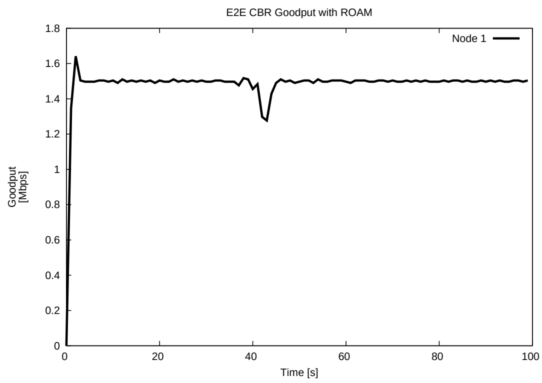
(a) AODV (Star Topology)



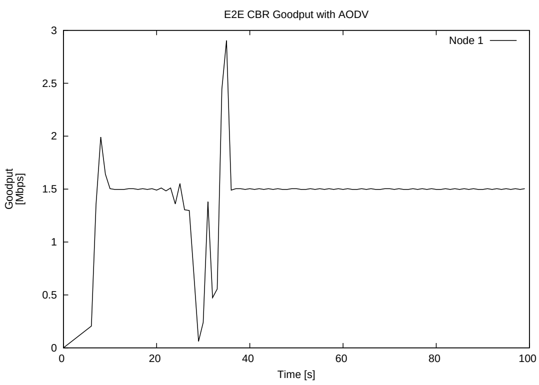
(b) ROAM (Star Topology)



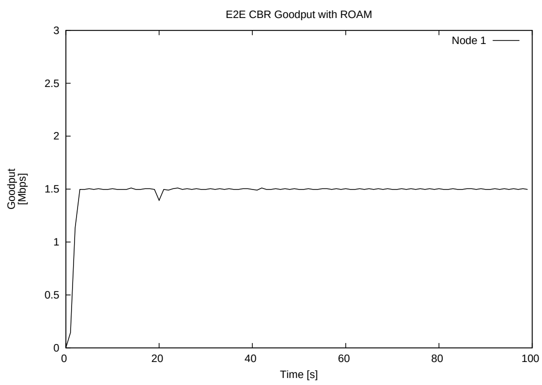
(c) AODV (Ring Topology)



(d) ROAM (Ring Topology)



(e) AODV (Tree Topology)



(f) ROAM (Tree Topology)

Figure 5.31: Scenario 2: Instantaneous E2E CBR Goodput (AODV versus ROAM)

Table 5.7: Scenario 3: Overall Packet Drop Comparison

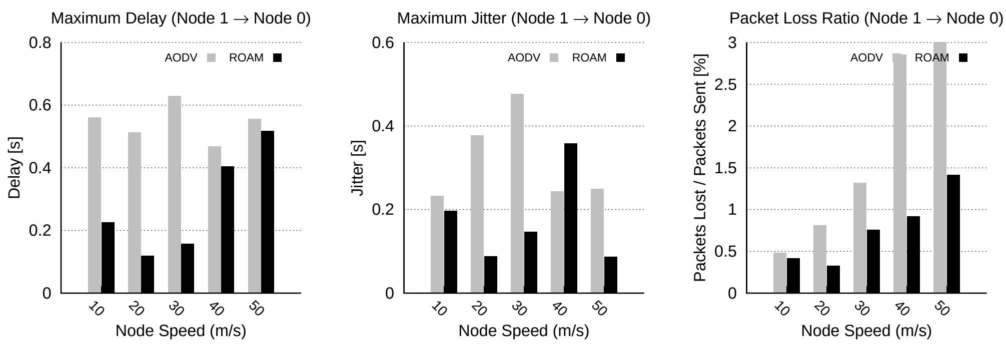
	AODV			ROAM		
Mobile Node Speed (m/s)	Collision Count	IFQ Full	No Route	Collision Count	IFQ Full	No Route
10	80	8460	7	21	8405	2
20	161	8437	9	67	8411	3
30	225	8708	8	161	8693	8
40	291	8796	9	195	8101	11
50	322	8483	12	299	8256	7

5.2.2.4 Scenario 3: Different Mobile Node Speed

This scenario is to demonstrate that ROAM is able to constrain network delay and jitter when channel quality and forwarding IN availability changes rapidly, as node speeds are increased. The star topology and simulation configuration used in previous scenarios has been implemented. Mobile transmitter speed was varied between 10-50m/s at increments of 10m/s, under the same traffic configurations.

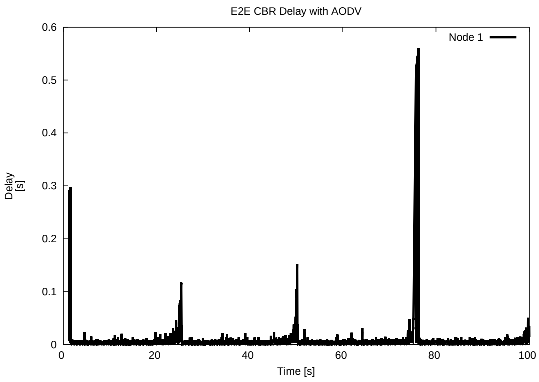
Figure 5.34 shows that goodput was lowered at each IN handoff with AODV at higher speeds. This can be explained by the increased number of collisions on fading channels, as a result of raised handoff frequency. Therefore, while the MAC layer attempts to recover from previous loss, emptying the IFQ (seen as a short goodput burst) the current channel quality has already begun to fall.

However, under the same traffic rate, when nodes move in and out of range of each other at a higher speed, fewer packets are transmitted on each link before handoff. Correspondingly, the occurrence of queue overflow was similar for all sub-scenarios (table 5.7).

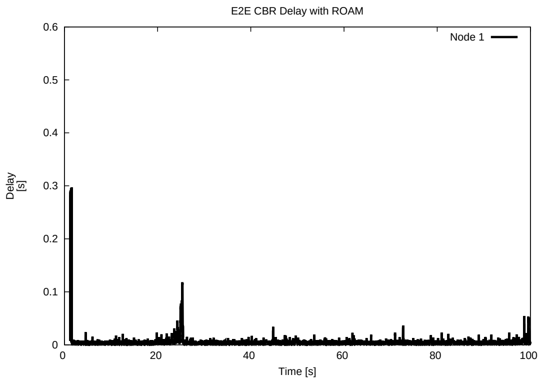


(a) Maximum E2E Delay at Handoff (b) Maximum E2E Jitter at Handoff (c) E2E CBR Packet Loss Ratio

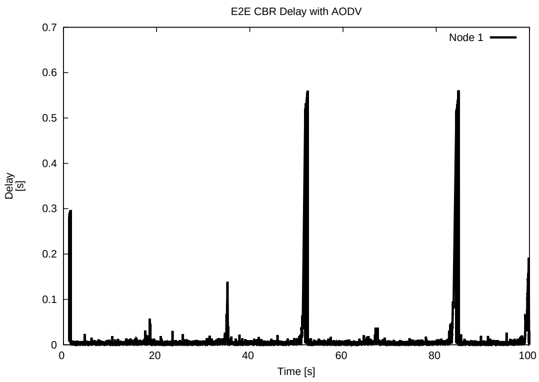
Figure 5.32: Scenario 3: Performance Comparison with Different Node Speed



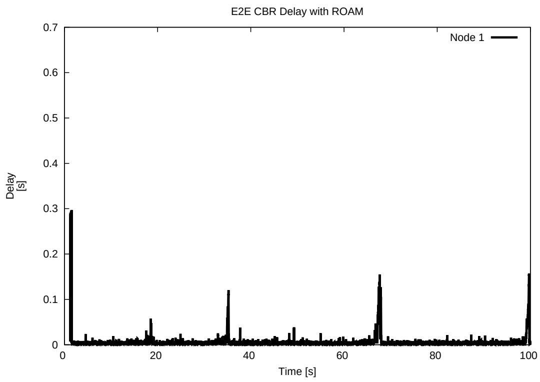
(a) AODV (MN Speed = 20m/s)



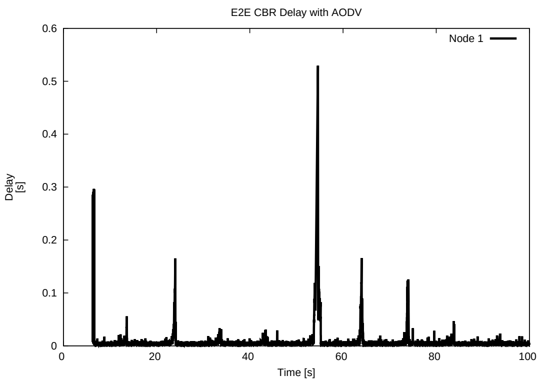
(b) ROAM (MN Speed = 20m/s)



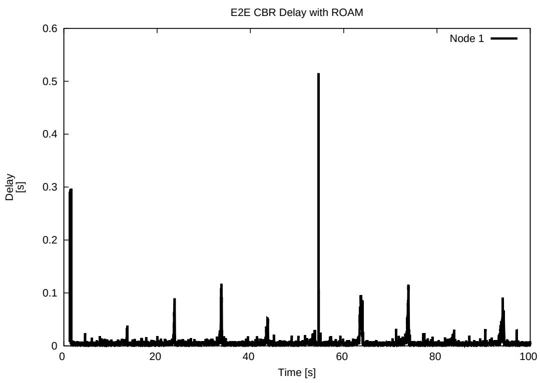
(c) AODV (MN Speed = 30m/s)



(d) ROAM (MN Speed = 30m/s)

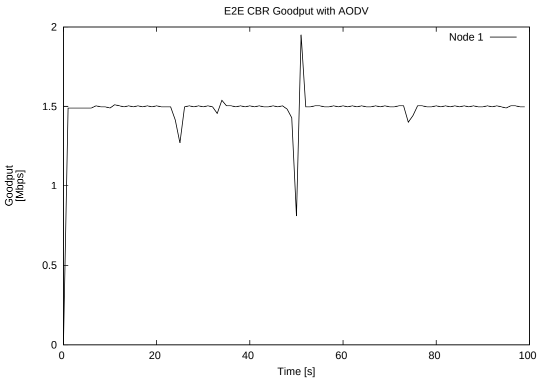


(e) AODV (MN Speed = 50m/s)

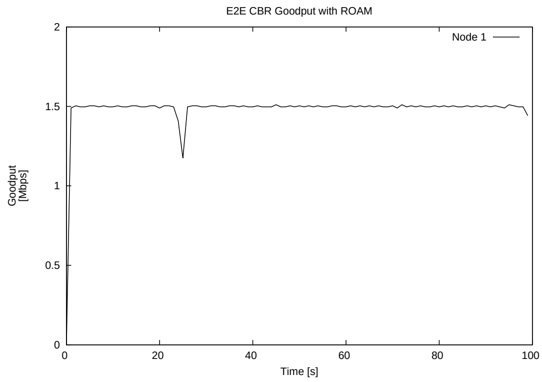


(f) ROAM (MN Speed = 50m/s)

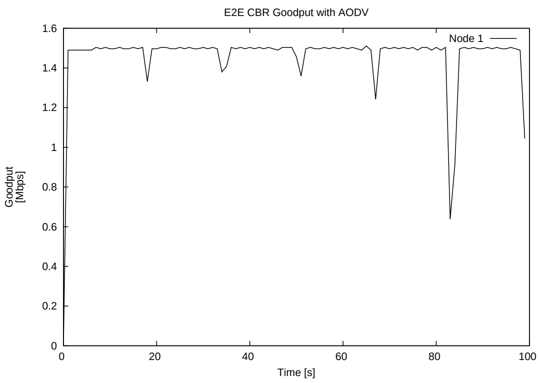
Figure 5.33: Scenario 3: Instantaneous E2E Delay (AODV versus ROAM)



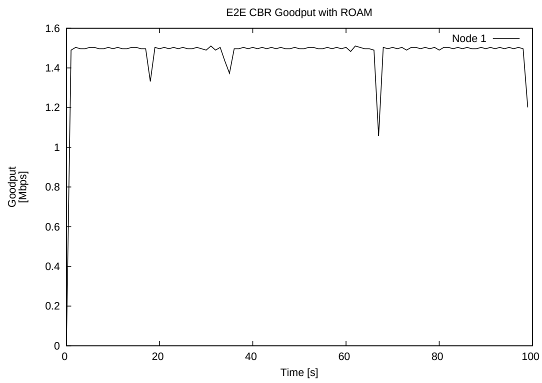
(a) AODV (MN Speed = 20m/s)



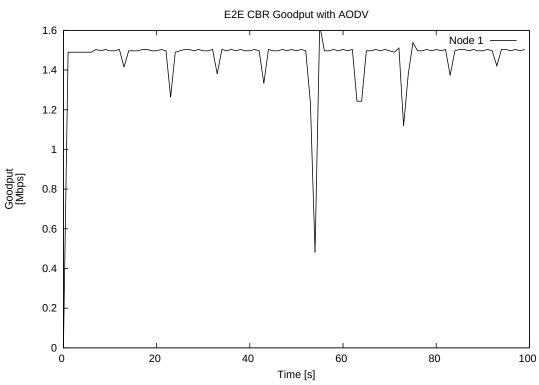
(b) ROAM (MN Speed = 20m/s)



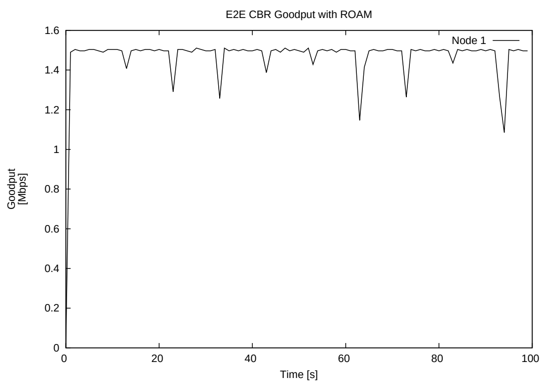
(c) AODV (MN Speed = 30m/s)



(d) ROAM (MN Speed = 30m/s)

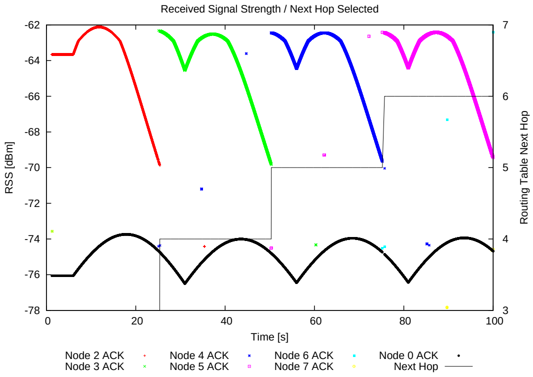


(e) AODV (MN Speed = 50m/s)

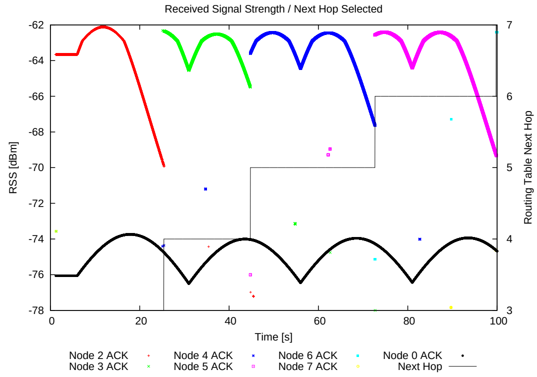


(f) ROAM (MN Speed = 50m/s)

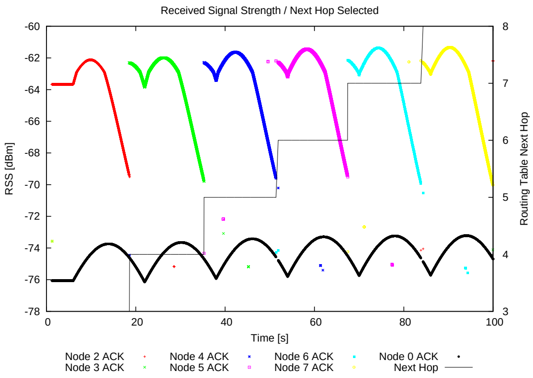
Figure 5.34: Scenario 3: Instantaneous E2E CBR Goodput (AODV versus ROAM)



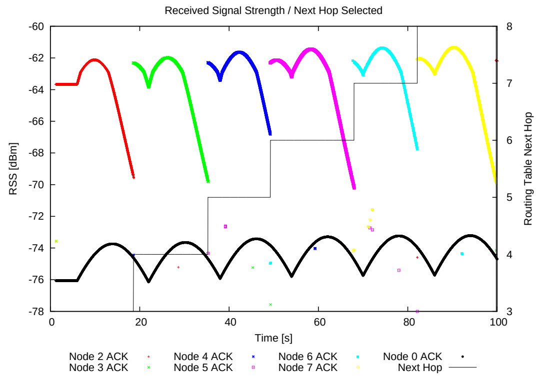
(a) AODV (MN Speed = 20m/s)



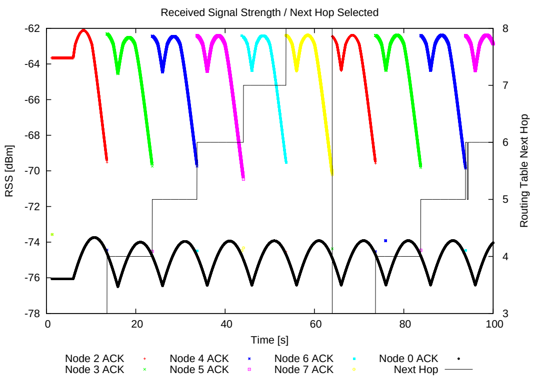
(b) ROAM (MN Speed = 20m/s)



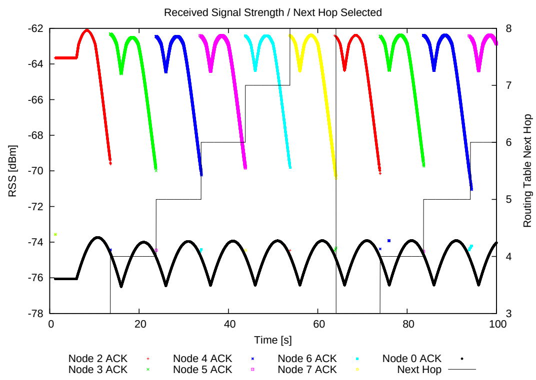
(c) AODV (MN Speed = 30m/s)



(d) ROAM (MN Speed = 30m/s)



(e) AODV (MN Speed = 50m/s)



(f) ROAM (MN Speed = 50m/s)

Figure 5.35: Scenario 3: Next Hop Selected by Node 1 (AODV versus ROAM)

Packet loss increased with each increment in nodal speed, whereas maximum delay was relatively similar between each sub-case. The primary cause of packet dropping was IFQ overflow in all scenarios. Queueing and contention delay are key factors of E2E delay, and ROAM has a more significant impact on the former.

The efficiency of multi-hop routing processes has a greater impact on E2E delay when high speed, repeated handoffs are required. Therefore, at node speeds of 50m/s both AODV and ROAM provided unsatisfactory levels of performance. While ROAM was capable of reducing packet losses at this speed, this had little impact on peak E2E delay as the middleware could not gather sufficient information on the channel to facilitate any large timing difference in handoff (figure 5.35). Therefore, figure 5.32(a) shows that with ROAM, E2E delay rose more regularly at each period of handoff as speed increased.

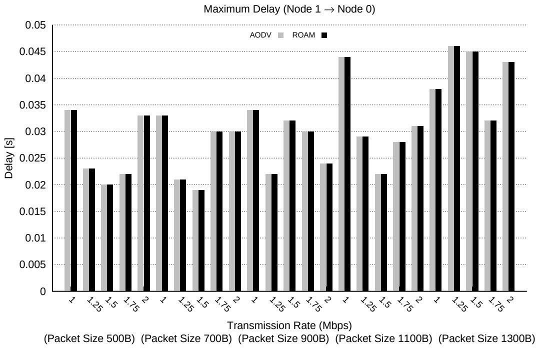
Repeated reconfiguration of topology leads to greater rates of change in channel quality. With transmitter velocity expediting path changes, there was a reduction in the timespan for which local information gathered by the ROAM, based on control packet receipt, was relevant. In this scenario, with node speeds of 40m/s, this had the effect of unnecessarily introducing jitter into the data stream (figure 5.32(b)).

Overall, ROAM performed well under the highly variable conditions that result in a MANET of high speed mobile nodes. A capability to reduce maximum delay and packet loss at handoff has been demonstrated, enabling the provision of guarantees to timing-sensitive applications that the maximum delay occurs during initial path setup.

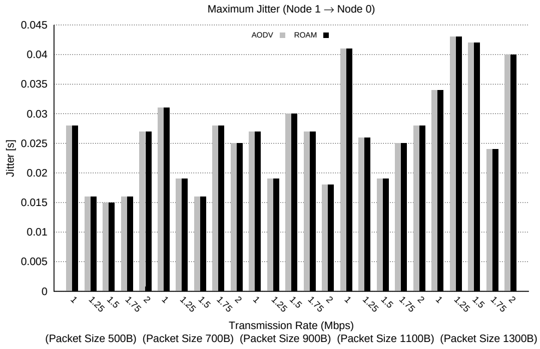
5.3 Validation of Results against Baseline Performance

The ROAM horizontal handoff optimiser is a standalone cross-layer scheme that relies on API intercepted protocol layer parameters to monitor conditions on the channel to the current next hop. If this channel quality begins to deteriorate and a higher power link is detected, handoff is expedited and the fading link avoided. Additionally, ROAM prevents the use of suboptimal channels to highly mobile nodes. The optimiser affects only local link selection, using information currently stored in the routing table.

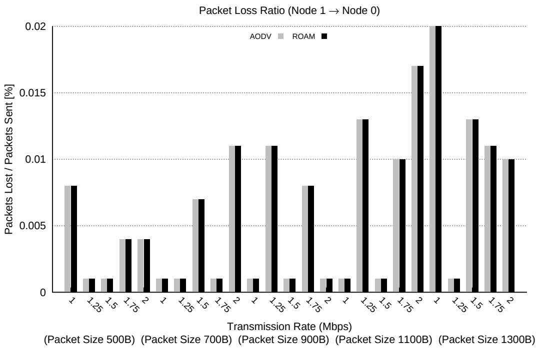
Previous simulation results demonstrated that if a higher power link is not detected early enough, or handoff is required more rapidly than ROAM can intercept protocol parameters, fast handoff is not implemented. This is also true if nodes are static. This limitation has been investigated by returning to the baseline scenario



(a) Maximum Delay at Handoff



(b) Maximum Jitter at Handoff



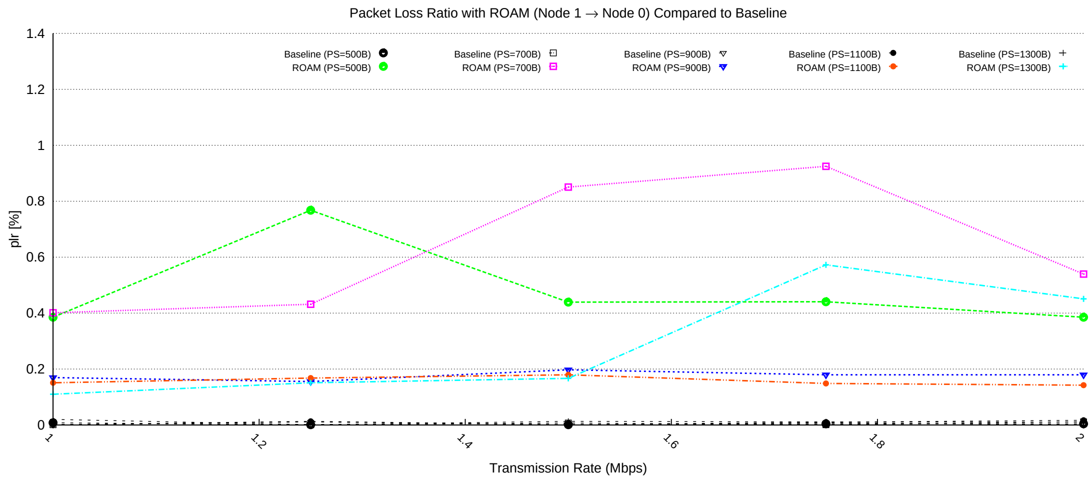
(c) Packet Loss Ratio

Figure 5.36: Performance in Baseline Simulation Scenario (AODV versus ROAM)

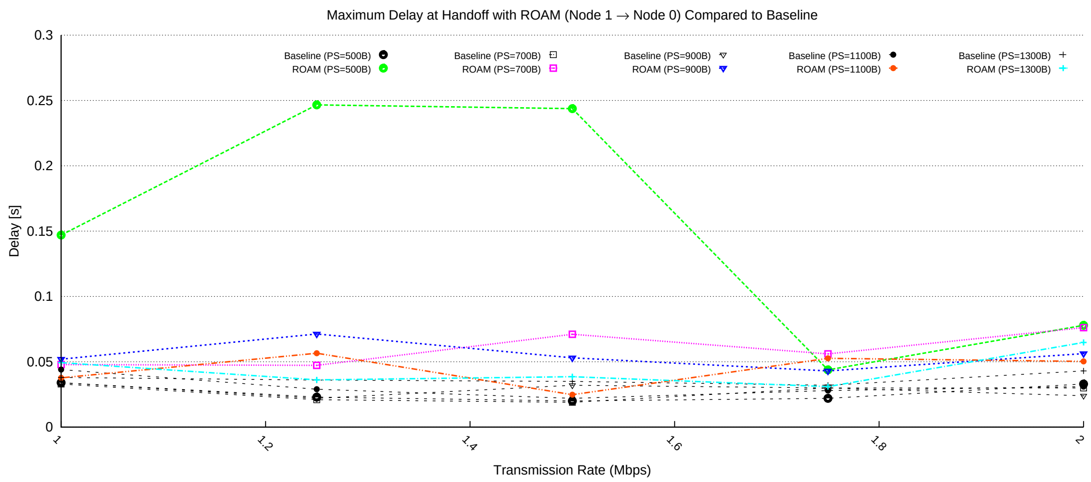
investigated in Chapter 3 and implementing the middleware in this simulation. Without node mobility requiring handoff, ROAM did not change E2E routing of AODV and there was a 0% or negligible change in RT performance, in terms of maximum delay, jitter and packet loss ratio (figure 6.71).

In Chapter 3, static transmitter-receiver ad hoc network simulations were investigated and results collated to form this baseline, to demonstrate best-case performance in a wireless ad hoc network. The ROAM optimiser tested in this chapter has been developed to provide bounded delay and loss guarantees to applications during horizontal handoff. This has been considered in comparison to results with an oblivious ad hoc routing protocol, AODV, which can be considered to be the baseline for worst-case performance.

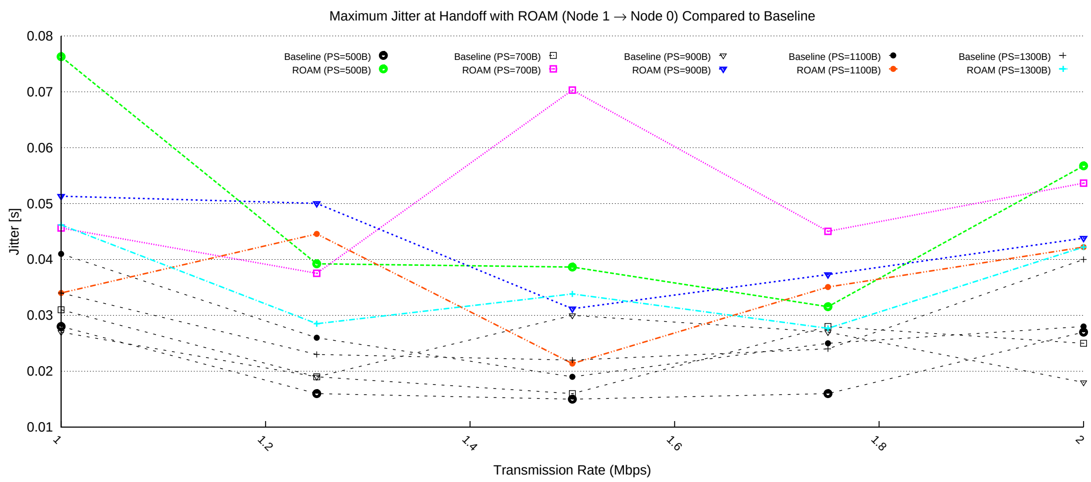
In Section 5.2.1.1, the optimiser was implemented in a single mobile transmitter with a range of CBR traffic configurations, orbiting a star topology. These results are now, therefore, compared to results under the same configurations in the baseline scenario. This is to evaluate the ability of ROAM to reduce the dis-



(a) Packet Loss Ratio



(b) Maximum Delay at Handoff



(c) Maximum Jitter at Handoff

Figure 5.37: Performance of ROAM compared to Baseline Performance

parity between performance in MANETs with complex network dynamics, and those with low levels of available resource variation.

Figure 5.37(a) shows that ROAM was not capable of reducing packet loss to the level of the baseline scenario and also provided a more variable level of performance. The lowest capabilities resulted when packet sizes of below 700B were utilised by the CBR application. All ROAM packet loss ratio results were thus within the limits of best and worst performance.

ROAM was capable of reducing maximum delay at handoff to close to best-case performance, as shown in figure 5.37(b) when the packet size was greater than 500B. These results excluded initial ad hoc path setup delay and jitter, which are low in the single-hop baseline scenario, but elevated in the multi-hop scenario that the optimiser was implemented in.

Maximum jitter was, under certain traffic settings, reduced to within 0.01s of the baseline, but was generally much higher than the baseline (figure 5.37(c)). Multiple approaches utilised by the optimiser can introduce jitter into a flow, including pausing application transmission during the time taken for RREP to travel from the MAC to network layer and a new link to be added to the routing table. However, the result in Section 5.2.1.1 demonstrated that maximum jitter with ROAM was lower than with AODV alone.

At the same time, only one CBR source was used in these simulations and, as demonstrated in Subsection 5.2.2.1, increasing the number of CBR sources resulted in reduced performance for all CBR flows. These results demonstrate the benefits of over-provisioning to support of ISRT in MANETs. The results are promising in terms of ISRT performance and show the capability of cross-layer middleware in bounding E2E delay, jitter and packet loss in comparison to an ad hoc routing protocol alone.

5.4 Summary and Discussion

This chapter presented the results of rigorous testing of a local horizontal handoff optimiser, implemented within a cross-layer middleware architecture (ROAM). The goal of ROAM is to provide guarantees of bounded E2E delay and jitter for time-critical applications in a MANET and to reduce packet loss to meet safety-critical requirements.

MANET routing protocols select paths on the basis of routing and link layer control packet receipt, without consideration of link quality. This results in sub-optimal link selection and repeated switching between multiple paths. Elevated collisions, retransmission timeouts and excess of buffering requirements on fading links, incite violations of QoS when these flows are ISRT, due to their delay and jitter-intolerances (see Chapter 3). The scenarios implemented in this chapter utilised varied traffic, mobility and topology configurations, creating dynamic layer-1 and 2 conditions and packet loss ratios and delay that was elevated in comparison to the baseline, best-case performance.

These scenarios show that ROAM is able to constrain maximum delay and jitter at handoff, in spite of increasingly dynamic network conditions, without reliance on specific per-flow states or topologies. This enables the provision of guarantees to timing-sensitive applications that maximum delay occurs only during initial path setup. This is the period during which maximum E2E delay is highest as packets are necessarily queued while AODV sets up initial E2E paths by flooding RREQ and RREP packets through the network. In providing improved ISRT performance under these conditions, the middleware architecture is demonstrated to be scalable and adaptable to the varied conditions resulting from multiple flows of differing requirements traversing through the MANET.

The optimiser abstracts protocol parameters relating to intercepted control packets and the MAC IFQ, in order to detect link conditions and institute fast handoff. The optimiser is flexible and autonomous as it does not need to be configured during runtime. This is because hidden transmitter indication is based on relative comparison between control packets that are and are not intended for the optimised node.

The ns2-MIRACLE simulation results have shown that ROAM is able to adapt to fluctuation in link quality, to constrain E2E delay, jitter and loss of wireless nodes. ROAM also reduces the gap between best and worst-case performance, with peak delays similar to those observed in a static ad hoc network.

Traffic and network configurations are major factors causing these complex network dynamics. Under variation of these settings, ROAM has demonstrated a capability to bound handoff delay and jitter for all flows, by reducing the preferential use of short, low-quality channels and the associated loss recovery requirements of this usage. However, reduced performance improvement capabilities were apparent when the lowest packet size and traffic rate were implemented, which increased the time taken by the optimiser to detect a fading channel. Additionally, it was observed that non-ROAM transmitters closest to the ROAM node were subject to increased loss and delay. A ROAM node will handoff faster to a link that may be fading for another traffic source. As a result, a short term increase in collisions will impact more on the node that continues to use this fading link.

Contention delay, introduced as the MAC layer arbitrates for the shared medium, is a key stochastic component of E2E delay. Responsiveness to link quality variation and competing traffic can therefore reduce requirements for retransmission and random backoff and, correspondingly, the magnitude of this component.

Therefore, the following chapter investigates the second contention reduction optimiser developed for the ROAM architecture.

Chapter 6

Performance of ROAM Contention Control Optimiser

6.1 Introduction

In a shared medium, transmissions that are spatially exposed to each other contend for bandwidth. However, radio broadcast nodes rely on a single antenna for transmission and receipt, thus collision detection (CSMA/CD) cannot occur at the same time as data transmission. Without functional CSMA/CD, media access control problems, such as the the hidden node problem, occur and when two mobile transmitting nodes that are hidden from each other share the same forwarding node, the contention for resources is hidden and bandwidth use becomes intrinsically selfish. This results in increased collision at the forwarding node and higher retransmission and buffering requirements for both transmitters, as indicated by the investigative simulations in Chapter 3.

IEEE 802.11 networks use RTS/CTS handshaking, also known as virtual carrier sensing or CSMA/CA, to alleviate this problem, but this introduces artificial delay and jitter into flows. The scoping simulations also demonstrated that the rise in control traffic and interference errors in RTS/CTS and routing packets, increased MAC layer retransmission timeouts; repeated backoff and overflow of buffer resources. This impeded the provision of delay and per-packet jitter guarantees. The key cause of this being flow admission that is not responsive to changing channel and contention conditions.

The ROAM architecture implements a distributed contention control optimiser in order to improve network performance in a hidden node situation. Optimisation is locally implemented, only within the protocol layers of a single mobile transmitter, without intercommunication between other instances of the middleware. In order to avoid long term performance loss, nodes should be able to identify and respond quickly to increased channel contention. With a hidden node scenario, it is still possible to provide responsive admission with the aim of reducing contention when certain network characteristics appear. Increasing path delay, ACKs sent to a hidden node by the forwarding node, queue length and MAC layer packet dropping all demonstrate reduced resource conditions. While they do not lead to the direct detection of a hidden node, distributed responsiveness to reduced channel availability enables RT applications to make more efficient use of available resources. Both the ROAM architecture and contention control algorithm are illustrated in Chapter 4.

Section 6.2 evaluates simulation results with the ROAM contention control optimiser enabled. These are compared to MANET performance when, firstly, CSMA alone and secondly, CSMA/CA (virtual carrier sensing with RTS/CTS) are used. ROAM itself is implemented alongside CSMA as this is the default setting for IEEE 802.11. Validation of the middleware is then extended to a comparison against a best-case, or baseline MANET performance scenario in Section 6.3 in a single-hop wireless network. This is to evaluate the limitations of the optimiser in conveying performance in a MANET with complex network dynamics closer to that of a network with low levels of resource variation.

The reduction of E2E delay and jitter is vital to inelastic soft real-time (ISRT) applications, as discussed in Chapter 2. The goal of the optimiser is flexible and autonomous provision of improved performance for RT applications in a MANET. Variation in MANET configurations and the traffic settings of applications create network dynamics that subject flows to increased packet delay and jitter. Correspondingly, the middleware has been rigorously tested for its independence of firstly, transmission setting and secondly, MANET condition and for the ability to constrain E2E delay and jitter. The simulation design and configuration details have been discussed in Section 3.3 of Chapter 3.

6.2 Simulation Results

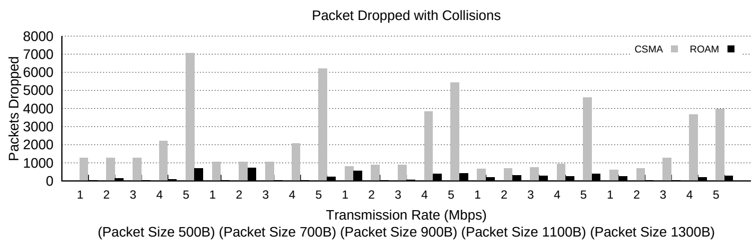
The ROAM Contention Control Optimiser has been validated from various aspects in ns2-MIRACLE. In order to ensure the validity of the data, unless otherwise stated, all results are means collated from 10 runs of each simulation. Given that the optimiser has been developed to alleviate hidden node contention, to which the current solution in IEEE 802.11 networks is to use RTS/CTS handshaking, it has also been tested both with RTS/CTS handshaking enabled and CSMA. This is to validate the optimiser against normal MANET performance. As the contention control optimiser manages resource usage on a distributed basis it has been implemented in all transmitters using the network.

6.2.1 Testing Transmission Setting Independence

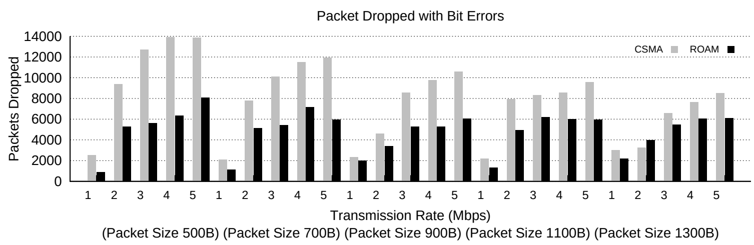
6.2.1.1 Scenario 1: Transmit Rate and Packet Size Variation

ROAM has been validated in a bus topology (detailed in Chapter 3) with differing transmitted load in this scenario, as a result of increasing application transmission rate (TR) and packet size (PS). However, the transmission settings were consistent between the two CBR traffic sources. This is to show that the contention control optimiser is able to constrain E2E packet loss, delay and jitter under the dynamic conditions introduced by increasing traffic rates and that the optimiser is independent of particular flow settings.

Therefore, transmission rate was varied between 1-5Mbps, at intervals of 1Mbps. This was done for each packet size between 500-1300B at increments of 200B. At higher transmission rate settings interface queues will fill faster and, depending on channel busy time, may not empty at the same rate. As both CBR streams shared a similar E2E path to the same receiver, increasing traffic rate creates a bottleneck. Additionally, in a hidden node scenario transmission rate will influence the number of collisions that occur on a shared channel. Depending on maximum queue lengths, larger packet size will generally increase the delay resulting from

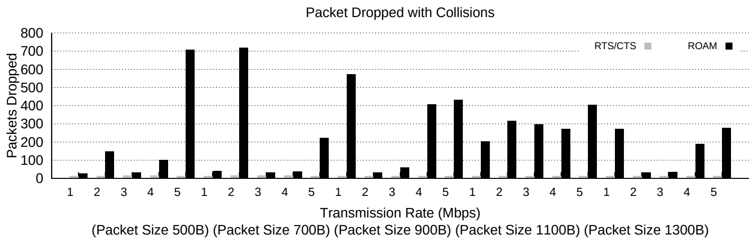


(a) Collision Count

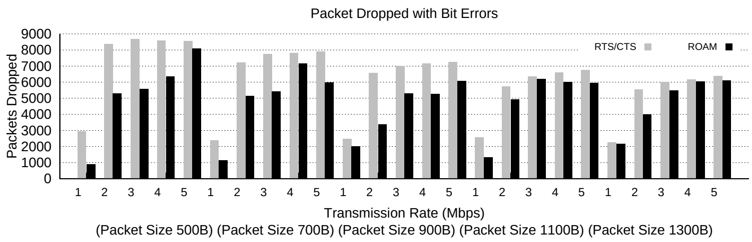


(b) Packet Bit Errors

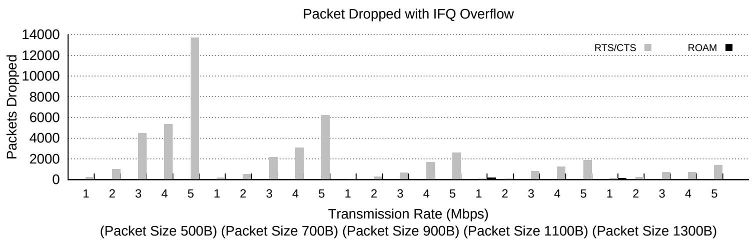
Figure 6.1: Scenario 1: Overall Causes of Packet Dropping (CSMA versus ROAM)



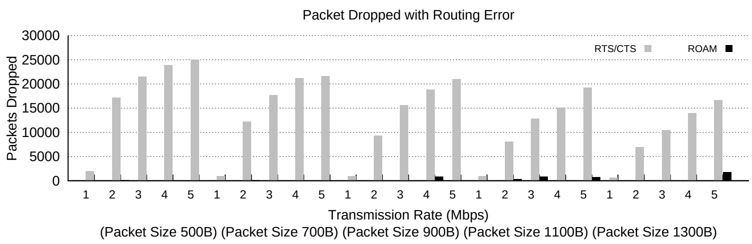
(a) Collision Count



(b) Packet Bit Errors

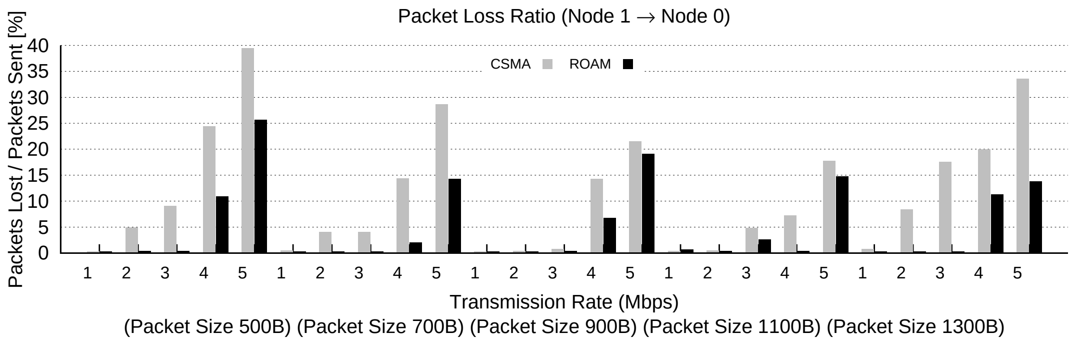


(c) IFQ Full

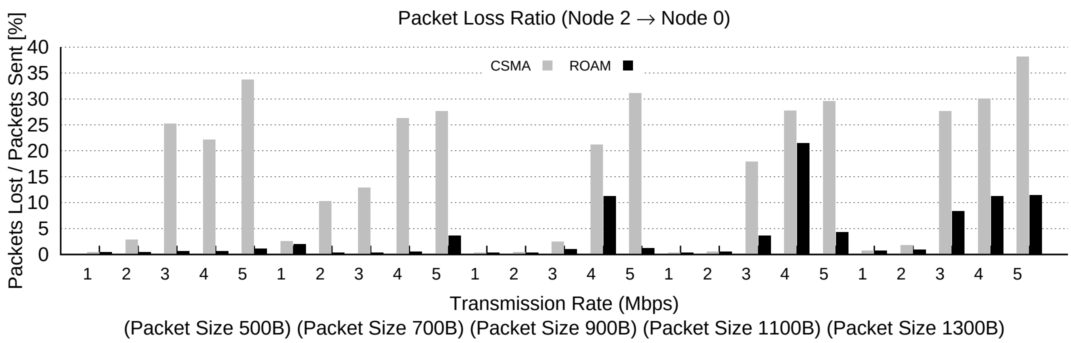


(d) Routing Errors

Figure 6.2: Scenario 1: Overall Causes of Packet Dropping (RTS/CTS versus ROAM)



(a) E2E CBR PLR ($N1 \rightarrow N0$)



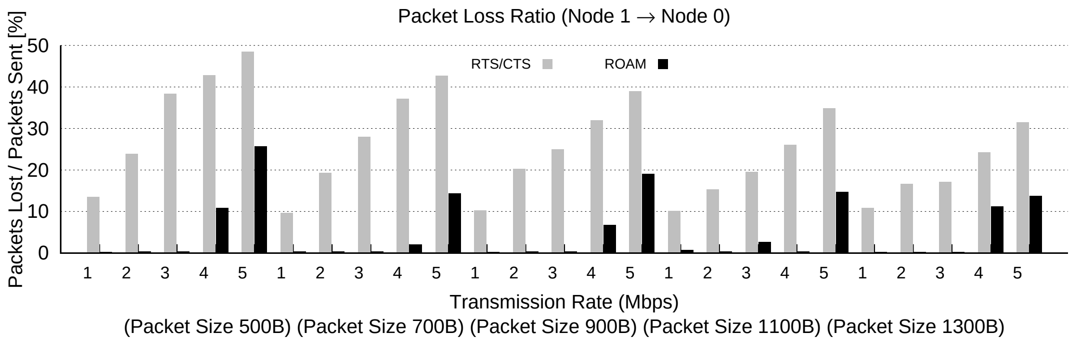
(b) E2E CBR PLR ($N2 \rightarrow N0$)

Figure 6.3: Scenario 1: Packet Loss Ratio Comparison (CSMA versus ROAM)

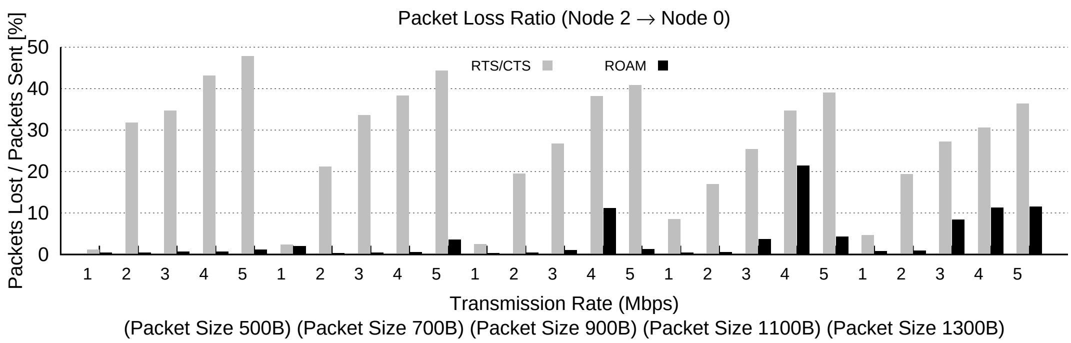
enqueueing and dequeueing packets as well as transmission delay.

Figure 6.1 compares packet dropping in two of the three test cases: with either CSMA or ROAM enabled. In a multi-hop network, retransmission of packets is used by the MAC layer at each forwarding and transmitting node for error recovery. Thus total MAC layer packet transmissions and packets dropped are likely to exceed the application transmission rate. Errors in these packets also result from interference of neighbours that can be at a distance much greater than a nodes transmission range. MAC and routing control packets are also capable of interfering with nodes out of transmission range.

The results show that collisions and packet errors were the most prevalent cause of packet loss with CSMA, while ROAM provides the greatest reduction in packet errors. This reduction is to be expected, as distributed load control creates a corresponding lowering in interference from hidden nodes. Figure 6.2



(a) E2E CBR PLR ($N1 \rightarrow N0$)



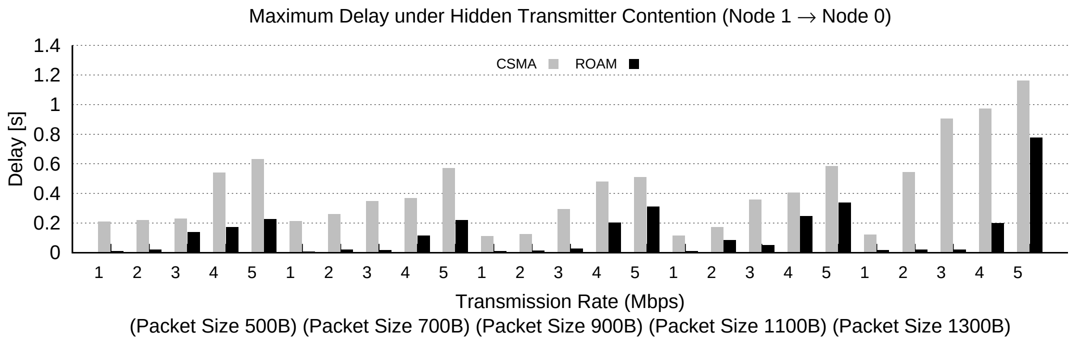
(b) E2E CBR PLR ($N2 \rightarrow N0$)

Figure 6.4: Scenario 1: Packet Loss Ratio Comparison (RTS/CTS versus ROAM)

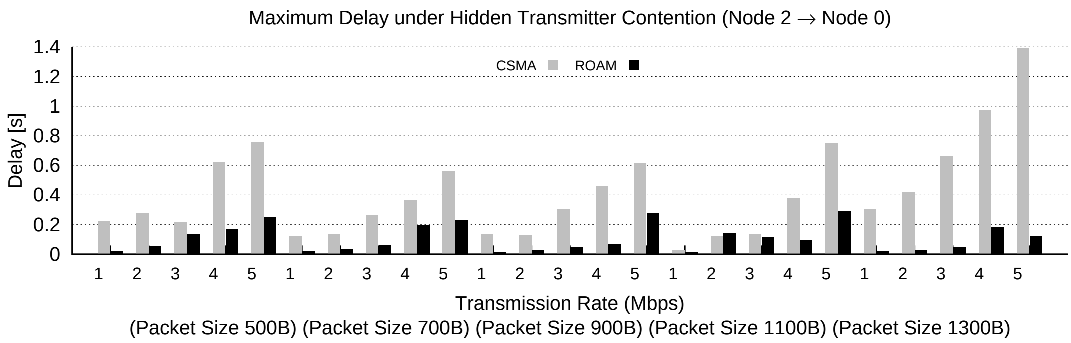
shows that ROAM performance of collision reduction is comparative to RTS/CTS, however, the use of control handshaking packets provides RTS/CTS with a greater improvement on CSMA alone. This is at the expense of packet delay as packets are required to wait in IFQs while handshaking takes place.

If RTS or CTS packets are then dropped due to interference errors, queueing time increases until the IFQ backlog eventually overflows. When the transmitted load was low, packet dropping due to full IFQs did not occur. RTS/CTS induced IFQ overflow was a greater issue when smaller packet sizes were implemented, as more control and data packets are being sent per second for the same datarate.

Additionally, routing errors also became an issue, as handshaking packets interfered with the routing handshake exchange, increasing path coherence times. Whereas with CSMA or ROAM implemented, these packet drops were negligible to zero in all cases. In reducing transmitted load following indication of the hidden



(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)



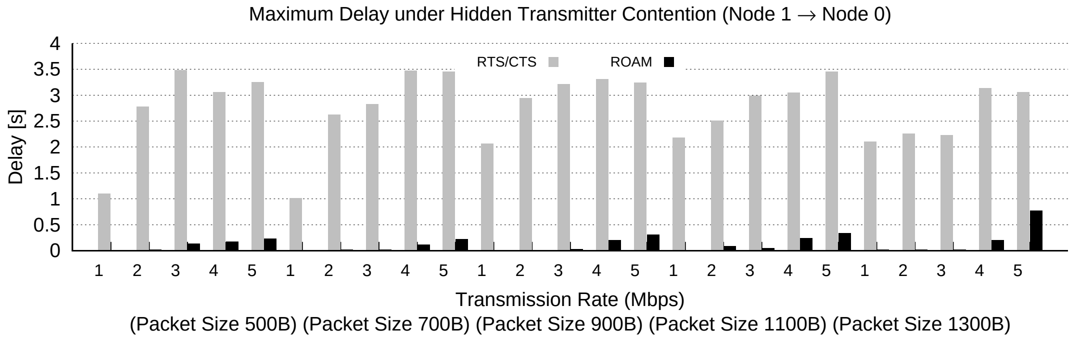
(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

Figure 6.5: Scenario 1: Maximum Delay Comparison (CSMA versus ROAM)

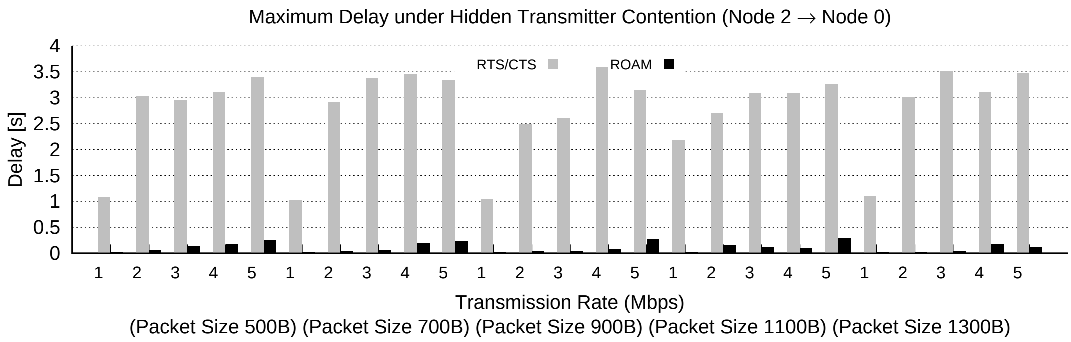
transmitter, packet errors resulting from interference were more greatly reduced with ROAM and collisions were reduced.

Correspondingly, figures 6.3–6.4 demonstrate the related lessening of packet loss ratio both in comparison to RTS/CTS and CSMA for both transmitters. While the application bitrate was CBR, the MAC layer transmission rate is multi-rate and is stepped to avoid packet loss. As a result, goodput in all of these simulations is variable and occasionally rises above the application transmission rate. In comparison to CSMA, ROAM provided reduced packet loss ratio and greater reductions in loss as traffic rate increased.

MAC layer retransmissions reduce the impact of packet error drops on E2E loss. Loss was more significantly influenced by traffic rate than packet size, and with loads of 4-5Mbps, packet loss with CSMA was extremely high. Without any load control, both transmitters continue to transmit to a shared forwarding



(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)



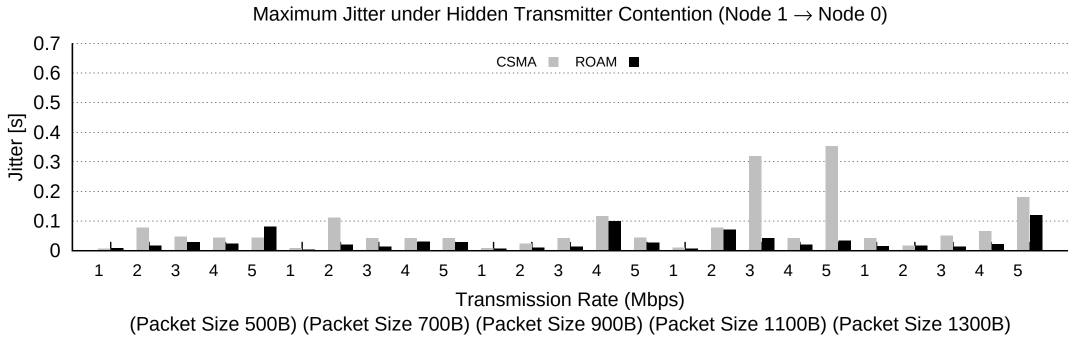
(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

Figure 6.6: Scenario 1: Maximum Delay Comparison (RTS/CTS versus ROAM)

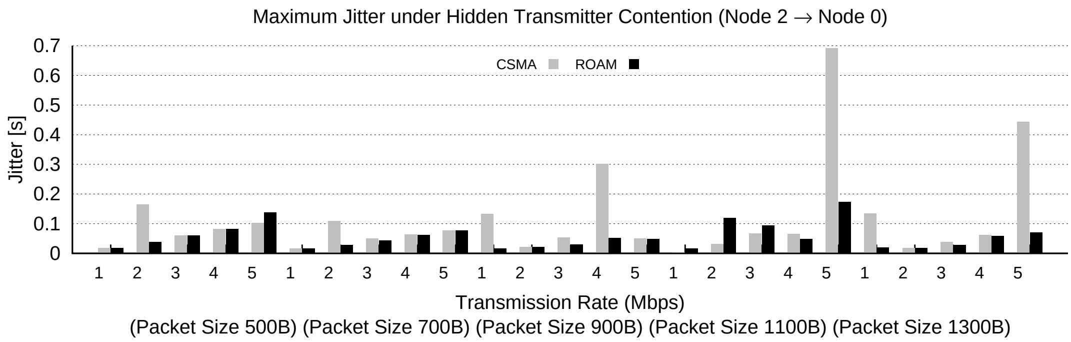
node, oblivious of whether it is in use. At high CBR rates, with RTS/CTS, loss still results as packets are repeatedly enqueued, and while the MAC layer rate steps down, the application does not. Packets that are successfully transmitted are therefore subject to very high delays.

Figures 6.5–6.6 show that maximum E2E delay and jitter were also lower for all tested transmission rates during the period of hidden node contention. With CSMA and RTS/CTS, both metrics increase with increasing load. At this point, localised MAC layer packet dropping resulted in reduced pressure on IFQs further along the E2E path, which could empty and reduce the backlog of packets.

RTS/CTS detection of a busy channel enables collision reduction, but the control packet flood results in increased control packet errors. However, ROAM responds to hidden node contention with the result of reducing both queueing and retransmission requirements to constrain peak values along the E2E path. ROAM



(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)



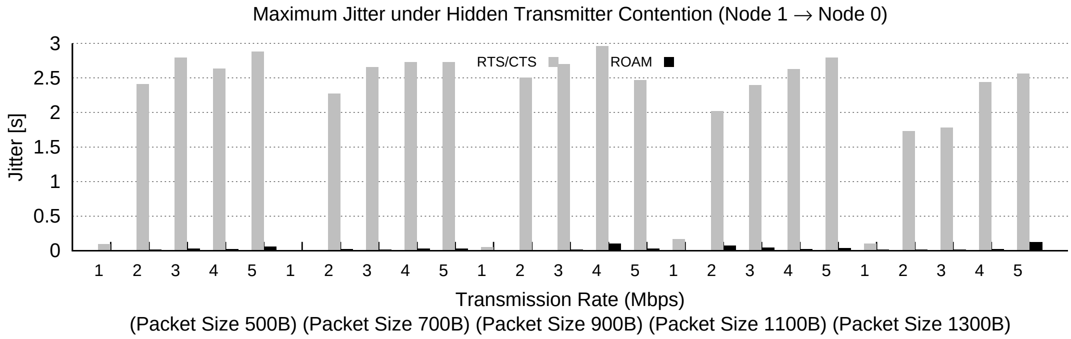
(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

Figure 6.7: Scenario 1: Maximum Jitter Comparison (CSMA versus ROAM)

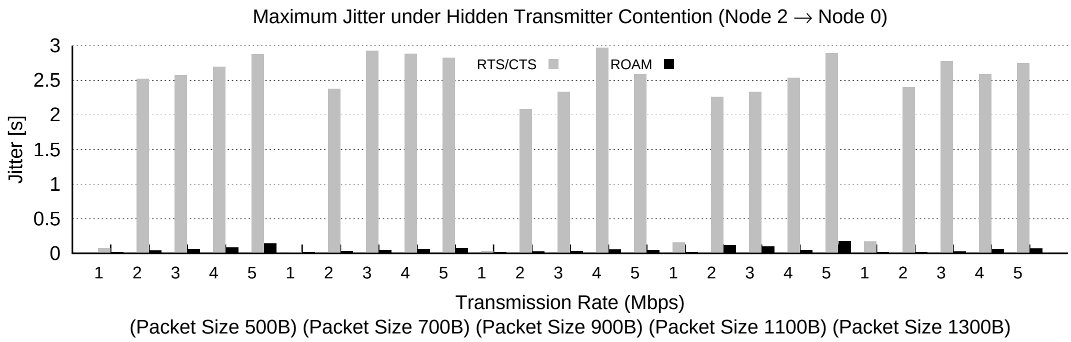
shows positive results in providing guaranteed maximum delay for all scenarios. In comparison to CSMA or RTS/CTS alone, delay was reduced for both transmitters.

Multiple factors have a strong impact on maximum jitter, including variation in queue length due to packet dropping and collisions and the influence of mobility on the length and quality of the E2E path. Increased packet loss at one node may also result in reduced jitter at the next along the path. Maximum jitter, therefore, tends to increase with transmission rate, but also fluctuates between the different sub-scenarios. With ROAM enabled, maximum jitter was also constrained, similar to maximum delay. During the period when two hidden transmitters compete for a forwarding node, with load of less than 3Mbps, ROAM showed a capability to constrain maximum delay and jitter to below 0.3s.

When RTS/CTS was implemented, lower packet loss was a tradeoff for increased peak delay and jitter for all transmitters. When larger packets are trans-



(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)



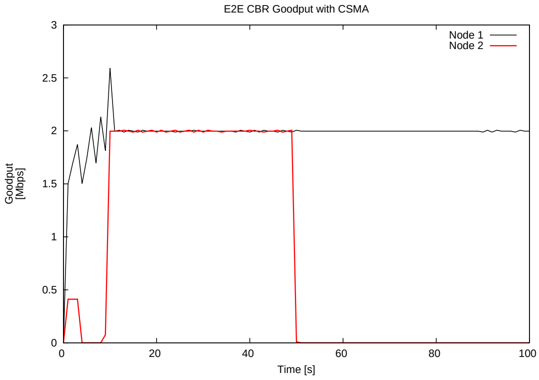
(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

Figure 6.8: Scenario 1: Maximum Jitter Comparison (RTS/CTS versus ROAM)

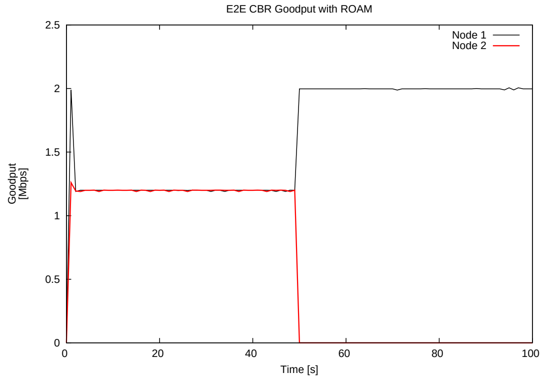
mitted across the network, contention delay increases for neighbouring nodes. As ROAM was able to reduce both collision counts and error induced packet dropping, the peak delay resulting from queueing and retransmission overheads along the E2E path, were also reduced.

Figures 6.9–6.10 show the instantaneous E2E goodput for a single run of the sub-scenarios with the same traffic settings for both transmitters. Large variations in CBR goodput appeared when both transmitters were within two hops of each other, resulting in increased collisions. Due to the close range of the receiver the IFQ is rapidly emptied when the MAC layer detects low noise on the link.

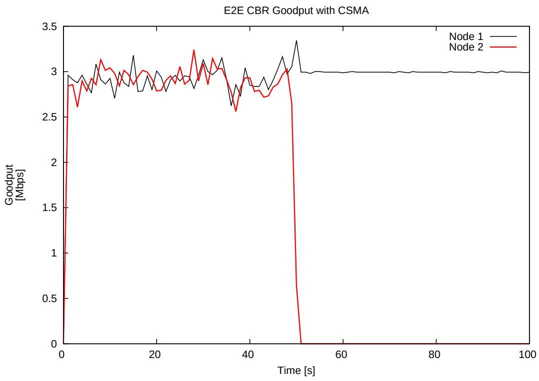
At low loads, the reduced transmission rate implemented by the contention reduction optimiser corresponds to a generally reduced goodput on the 1-hop link, when compared to CSMA. At the same time, low transmission rate with oblivious CSMA resulted in periods of negligible goodput under increased collision



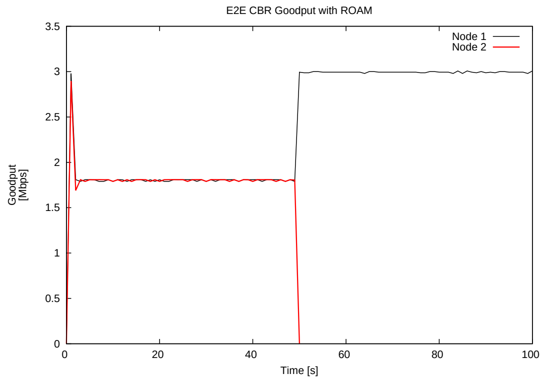
(a) CSMA (TR = 2Mbps, PS = 1100B)



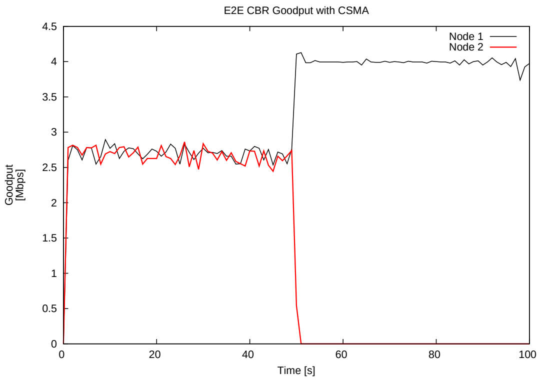
(b) ROAM (TR = 2Mbps, PS = 1100B)



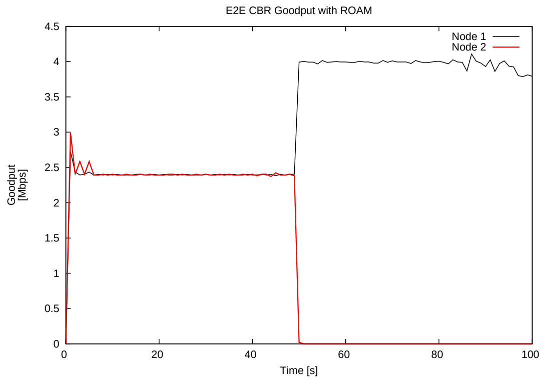
(c) CSMA (TR = 3Mbps, PS = 900B)



(d) ROAM (TR = 3Mbps, PS = 900B)

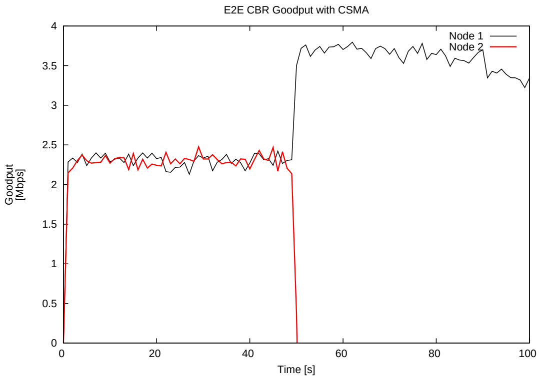


(e) CSMA (TR = 4Mbps, PS = 700B)

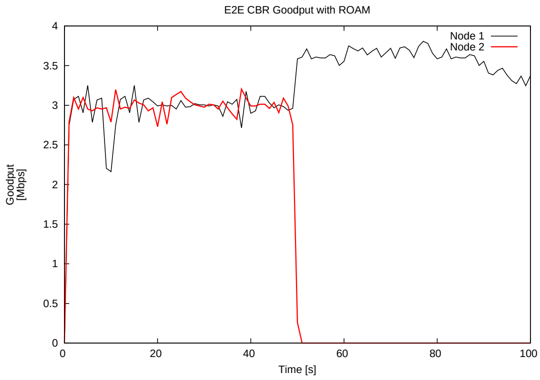


(f) ROAM (TR = 4Mbps, PS = 700B)

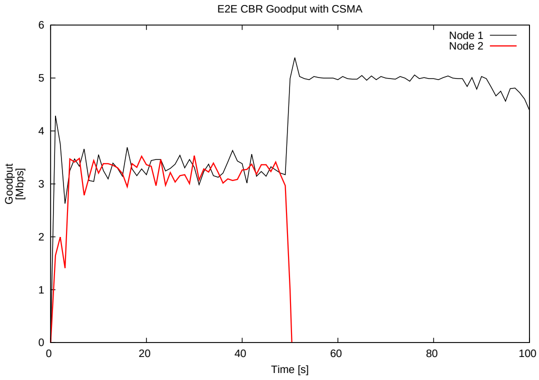
Figure 6.9: Scenario 1: Instantaneous E2E CBR Goodput (CSMA versus ROAM)



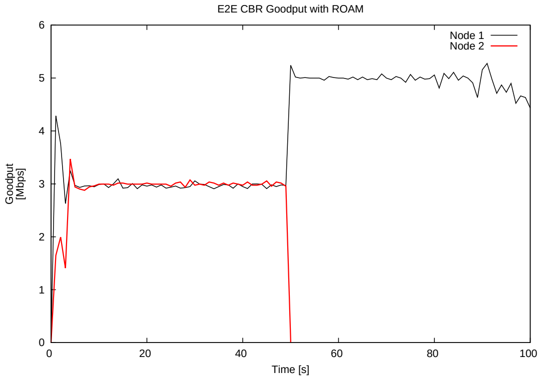
(a) CSMA (TR = 5Mbps, PS = 500B)



(b) ROAM (TR = 5Mbps, PS = 500B)

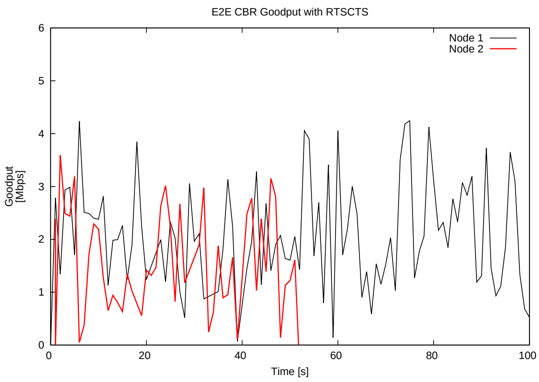


(c) CSMA (TR = 5Mbps, PS = 1300B)

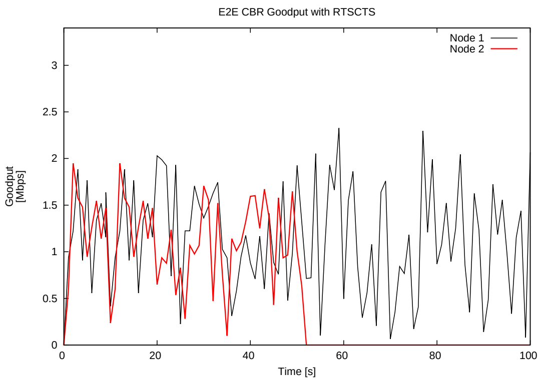


(d) ROAM (TR = 5Mbps, PS = 1300B)

Figure 6.10: Scenario 1: Instantaneous E2E CBR Goodput (CSMA versus ROAM)

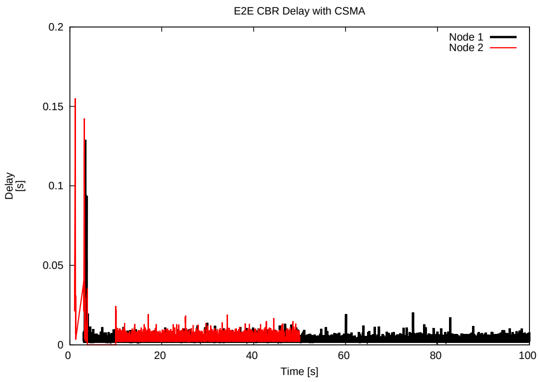


(a) RTS/CTS Goodput (TR = 5Mbps, PS = 1300B)

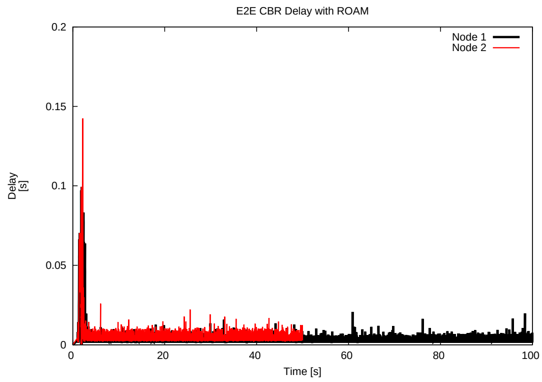


(b) RTS/CTS Goodput (TR = 5Mbps, PS = 500B)

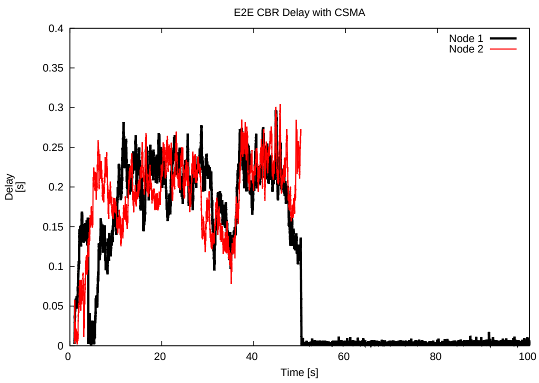
Figure 6.11: Scenario 1: Instantaneous E2E Performance (RTS/CTS)



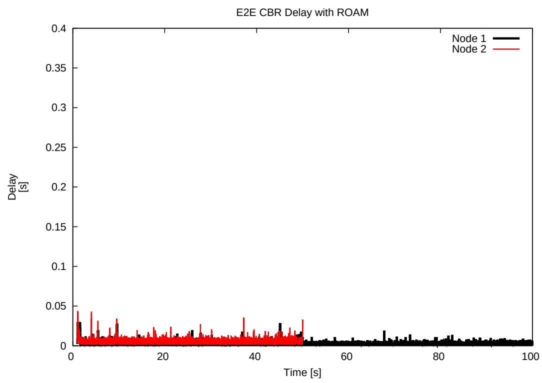
(a) CSMA (TR = 2Mbps, PS = 1100B)



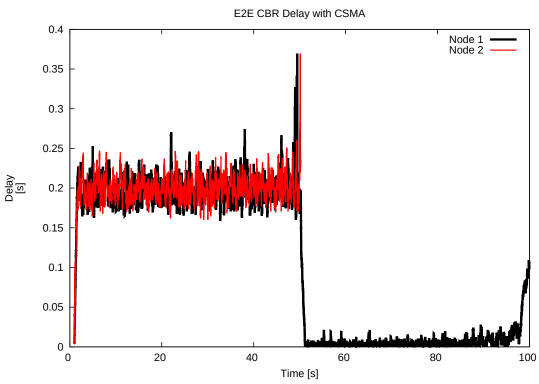
(b) ROAM (TR = 2Mbps, PS = 1100B)



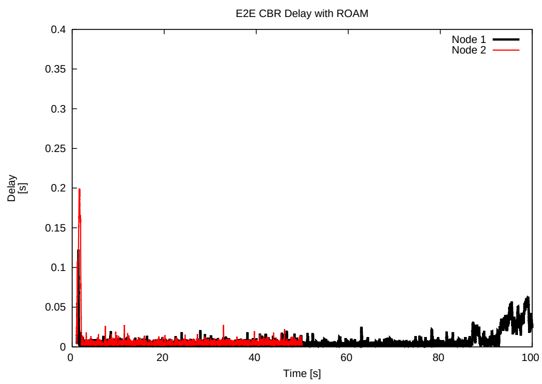
(c) CSMA (TR = 3Mbps, PS = 900B)



(d) ROAM (TR = 3Mbps, PS = 900B)

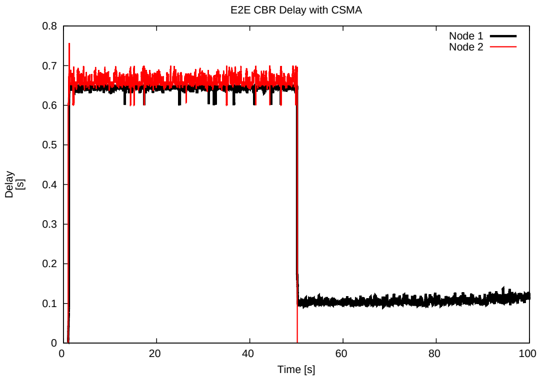


(e) CSMA (TR = 4Mbps, PS = 700B)

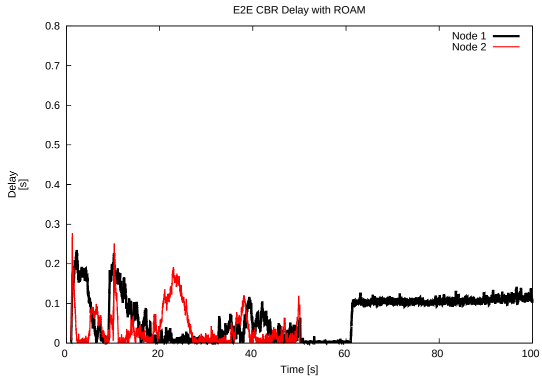


(f) ROAM (TR = 4Mbps, PS = 700B)

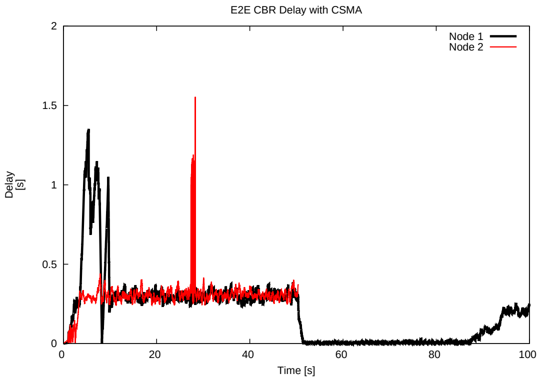
Figure 6.12: Scenario 1: Instantaneous E2E Delay (CSMA versus ROAM)



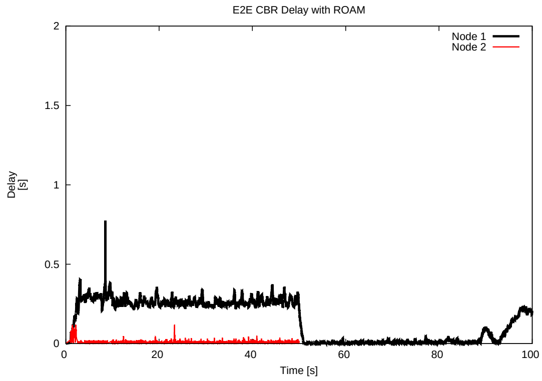
(a) CSMA (TR = 5Mbps, PS = 500B)



(b) ROAM (TR = 5Mbps, PS = 500B)

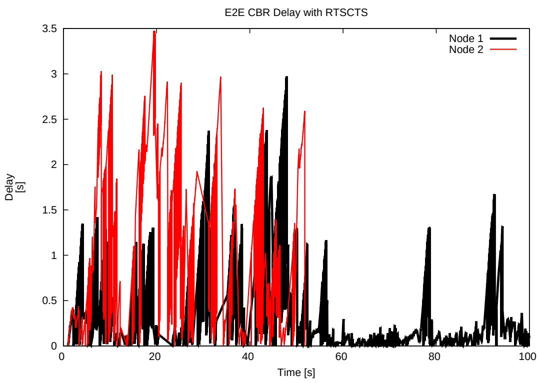


(c) CSMA (TR = 5Mbps, PS = 1300B)

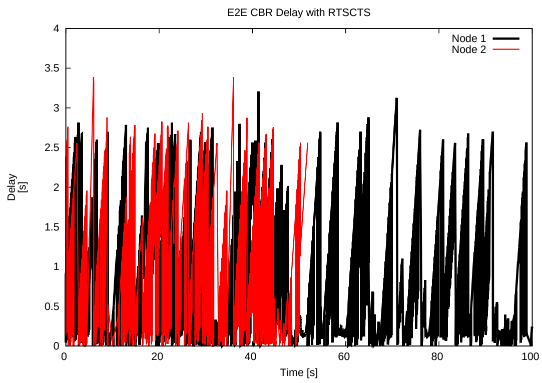


(d) ROAM (TR = 5Mbps, PS = 1300B)

Figure 6.13: Scenario 1: Instantaneous E2E Delay (CSMA versus ROAM)



(a) RTS/CTS Delay (TR = 5Mbps, PS = 1300B)



(b) RTS/CTS Delay (TR = 5Mbps, PS = 500B)

Figure 6.14: Scenario 1: Instantaneous E2E Performance (RTS/CTS)

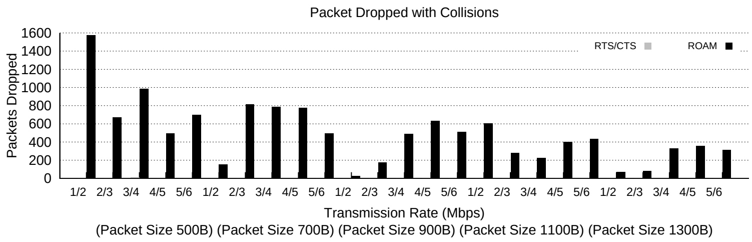
rates. However, at loads of 3Mbps and above, goodput was much lower and more widely variable with CSMA than ROAM during the period of forwarding node contention. The competition for resources between the two hidden transmitters resulted in unfair distribution of available bandwidth and corresponding delays. When node 2 ceased transmitting, ROAM in node 1 returned the application rate to the previous setting and the performance with ROAM was similar to CSMA.

In comparison, RTS/CTS resulted in lower goodput than CSMA when the CBR sources compete, as packets wait in IFQs for resolution of handshaking (figure 6.11). Instantaneous peak goodput was periodically higher for CSMA and RTS/CTS as a result of ROAM load control implementing a blanket reduction on transmitted load, even when the receiver was the 1-hop neighbour. Whereas, when the channel is free, the buffer is able to drain with RTS/CTS. This does not mean that CSMA and RTS/CTS perform better than ROAM, as packets reaching the receiver were subject to much higher delays, as seen in figures 6.12–6.13. Instantaneous delay and jitter were constrained with ROAM enabled as a result of reduced resource requirements following load control. Easing of transmitted load avoids the higher peak delay and jitter that occur as a result of repeated packet drops on the overloaded channel. As expected, figure 6.14 shows that RTS/CTS injects large artificial delays into flows, with significant resulting jitter both under hidden node contention and when only a single CBR flow is transmitted on the network.

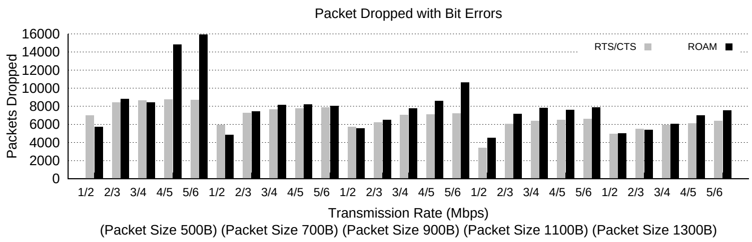
When ROAM is enabled, the stream of control packets intercepted as well as localised performance reductions are monitored for indications of undue contention. If the presence of a hidden CBR source is identified, both packet size and transmission rate of the node are adjusted to reduce the pressure on the shared channel. Following detection, each detecting transmitter reduced its load. When the nodes were out of range of each other, after 50s, ROAM tunes transmitted load to return to its previous value. By reducing the traffic load on the network, competition for the shared receiver was reduced after this point and the queue backlog was allowed to empty. As a result these metrics, as well as their peak values, were demonstrably reduced. In terms of RT performance improvement, a more valuable gain was provided by ROAM through the reduction in both E2E maximum packet loss, delay and jitter.

6.2.1.2 Scenario 2(a): Heterogeneous CBR Transmission Rate

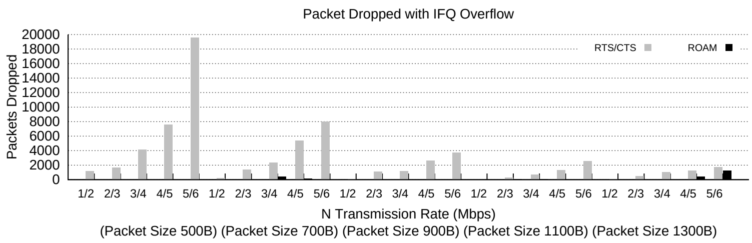
The previous simulation scenarios have considered the influence of homogeneous initial packet size and transmission rate. As the contention control optimiser tunes both of these settings to improve performance, this scenario considers the impact



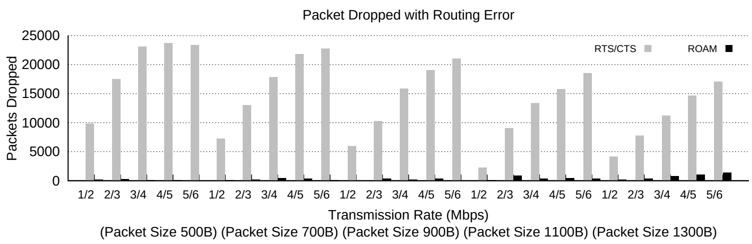
(a) Collision Count



(b) Packet Bit Errors

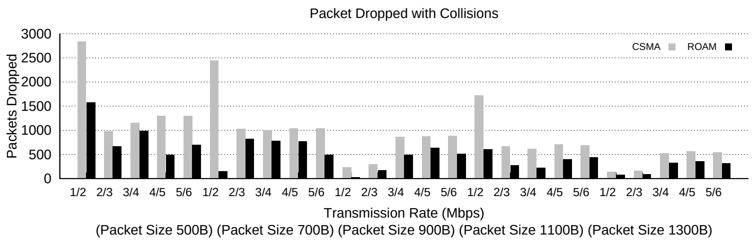


(c) IFQ Full

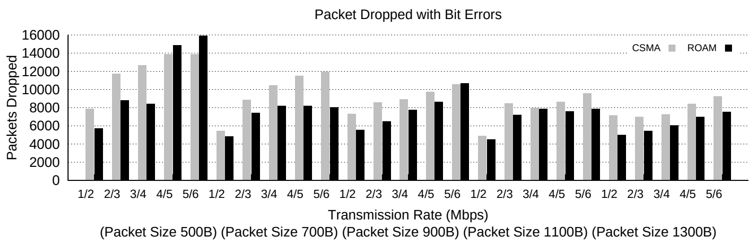


(d) Routing Errors

Figure 6.15: Scenario 2(a): Overall Causes of Packet Dropping (RTS/CTS versus ROAM)



(a) Collision Count



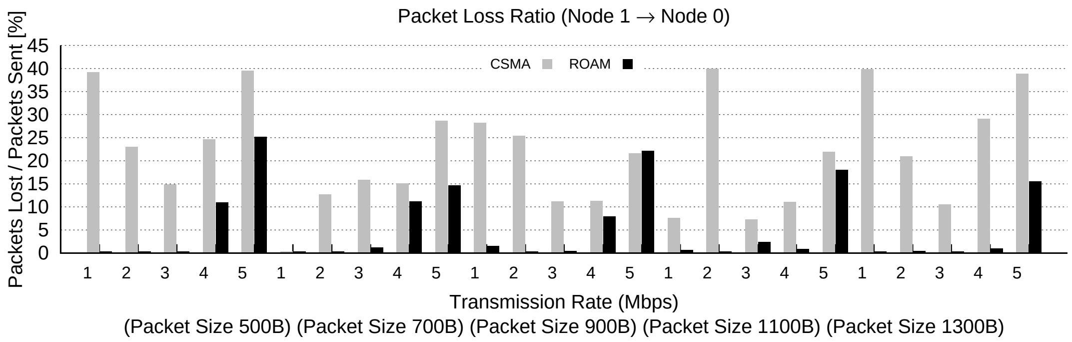
(b) Packet Bit Errors

Figure 6.16: Scenario 2(a): Overall Causes of Packet Dropping (CSMA versus ROAM)

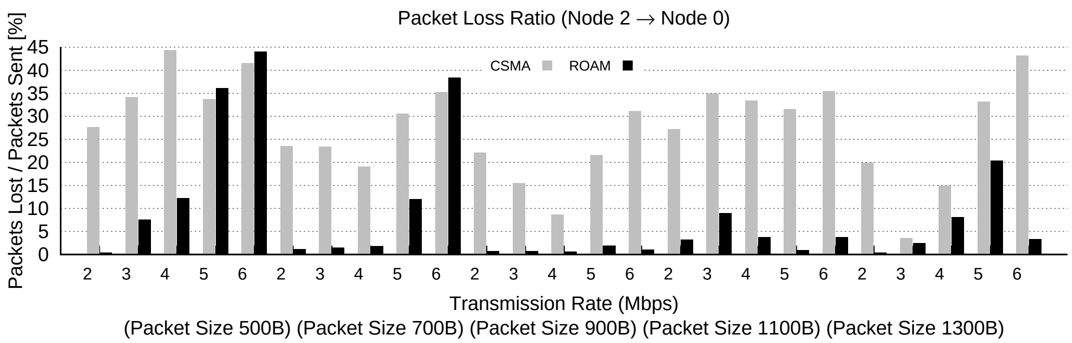
of configuring transmitters with mixed initial transmission rate. The differing packet sizes (N1/N2) can be observed in figure 6.15. Under heterogeneous conditions, load control is essential as bandwidth requirements differ between nodes and network jitter increases as packet delays in enqueueing, dequeueing and transmission become more varied.

In military or search and rescue scenarios, all data is likely to require high priority treatment. At the same time, support of mixed traffic configurations is essential in order to provide a scalable, flexible network. When transmission rates vary between sources, in the absence of efficient load control, bandwidth distribution can become unfair as higher rate flows selfishly overload a shared channel. However, ROAM is not dependant on particular traffic configurations or continuous conditions across a network, in order to identify and respond to the presence of a hidden transmitter.

Figure 6.16 demonstrates the key performance improvements, by which packet dropping was reduced in comparison to CSMA, were reduction in collisions and errors. Collision counts were not as significantly reduced by ROAM as RTS/CTS. However, figure 6.15 shows that this was again at the expense of extremely high



(a) E2E CBR PLR ($N1 \rightarrow N0$)



(b) E2E CBR PLR ($N2 \rightarrow N0$)

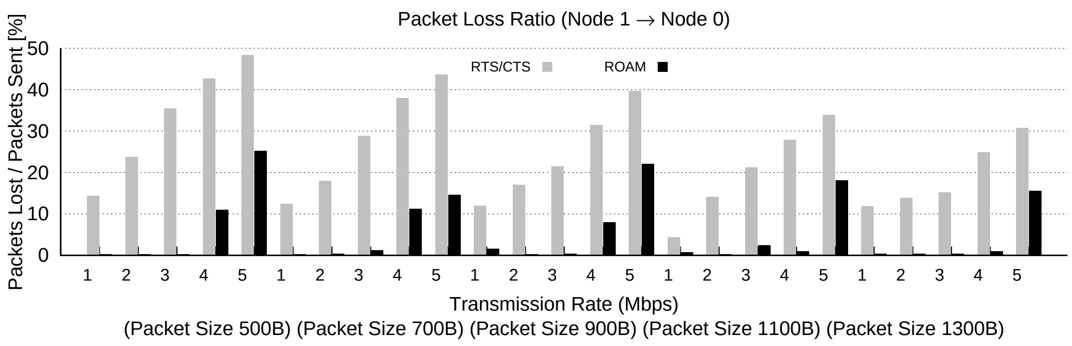
Figure 6.17: Scenario 2(a): Packet Loss Ratio Comparison (CSMA versus ROAM)

levels of buffer overflow and route loss.

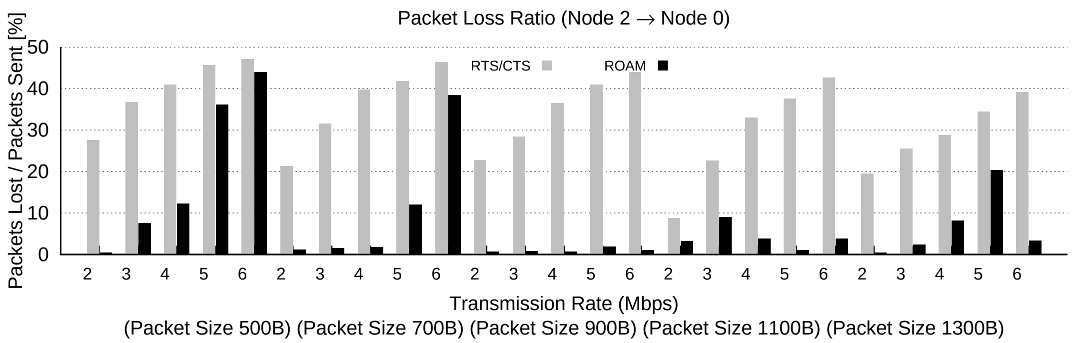
RTS/CTS packets repeatedly interact with ad hoc routing control packets on the channel. Loss of both of these types of control packets has a high impact on E2E delay, as it is only when handshaking and route repair, setup or maintenance are successfully completed that a head of line packet can be dequeued. The transmission rate of RTS/CTS packets is also dependant on traffic rate.

When a low packet size of 500B is implemented, at high traffic rates of 4-6Mbps, more packets are transmitted per second, resulting in increased collisions on a shared link. ROAM demonstrates a similar pattern of increase in packet error rate with traffic rate as seen with CSMA, therefore at the lowest level of performance, packet errors were higher with ROAM than RTS/CTS.

Mixed traffic rates reduce the effectiveness of RTS/CTS, which relies on consistent transmission rates and synchronisation between nodes, therefore packet



(a) E2E CBR PLR ($N1 \to N0$)

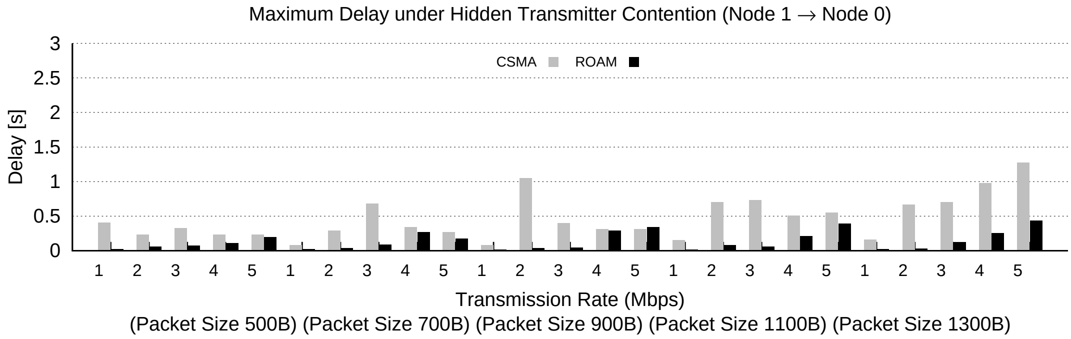


(b) E2E CBR PLR ($N2 \to N0$)

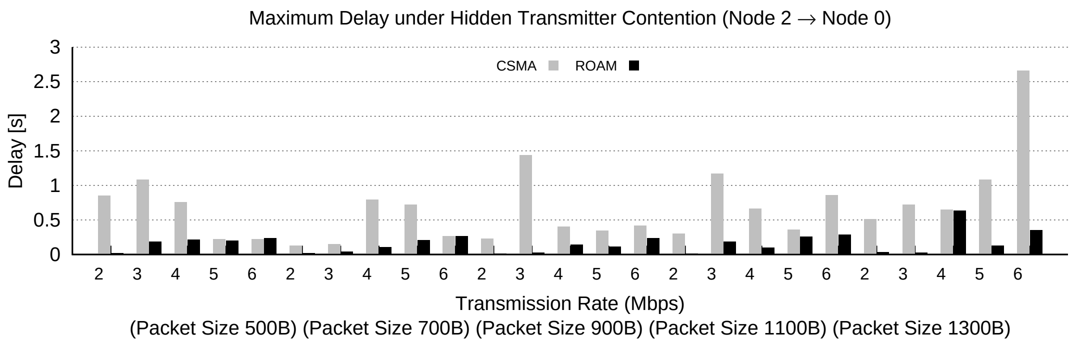
Figure 6.18: Scenario 2(a): Packet Loss Ratio Comparison (RTS/CTS versus ROAM)

errors will still occur. Packet errors were generally low across all traffic loads with RTS/CTS, while collisions, IFQ overflow and routing errors increased consistently with increasing load. Unlike RTS/CTS, ROAM does not prevent transmissions on a busy channel, resulting in lower performance at very high bitrates.

Correspondingly, packet loss ratio was lower with ROAM than both CSMA and RTS/CTS in all scenarios with mixed transmission rates of lower than 5Mbps (figures 6.17–6.18). ROAM monitors performance at the MAC layer as well as control packets intercepted from neighbouring nodes in order to identify the presence of a hidden node. By tuning the application to create a response to channel conditions, competition for the channel is reduced without requiring extensive buffering. This has the effect of constraining peak E2E delay through its components, contention and queueing delay.



(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)

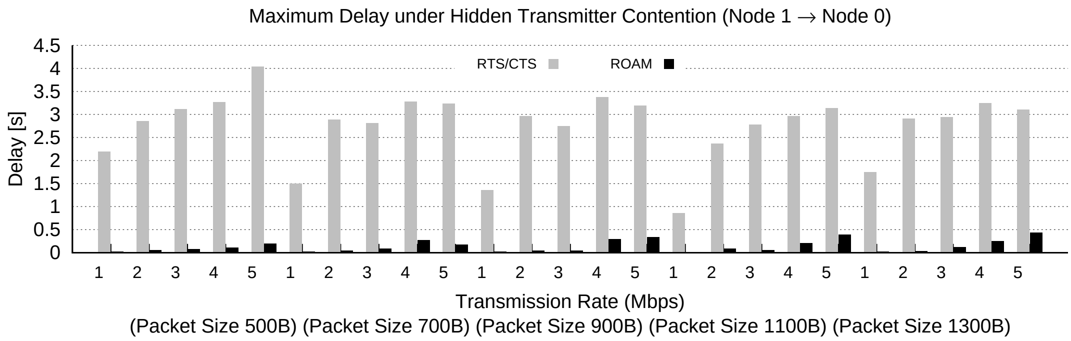


(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

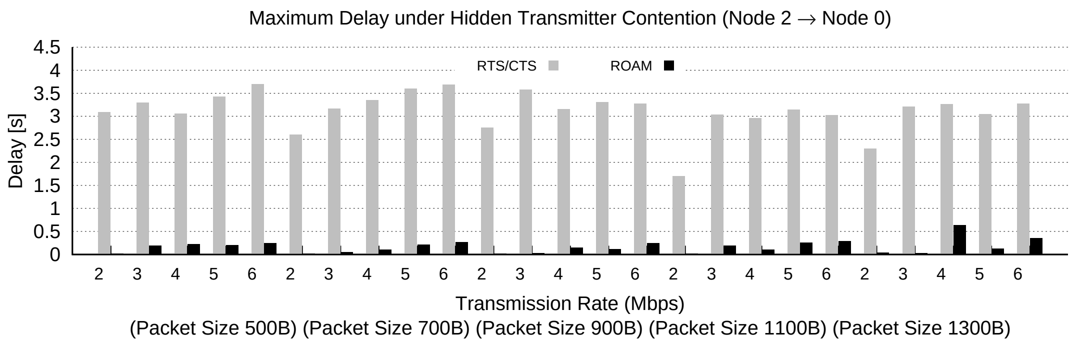
Figure 6.19: Scenario 2(a): Maximum Delay Comparison (CSMA versus ROAM)

With mixed transmission rates, peak delay varied widely between the two transmitters and was generally higher for the higher rate flows from node 2 (figure 6.19–6.20). In a wired network, it would be expected that higher rate corresponds to the lowest E2E delay. In contrast, on a shared channel, a lower level of performance is provided by IEEE 802.11 as a result of employment of both a multi-rate mechanism and distributed coordination function (DCF). As collisions and packet errors increase, nodes must repeatedly backoff and retransmit, introducing random delays into the stream. Additionally, if noise is detected on the channel, the auto-fallback mechanism will step down in response and packets are increasingly buffered.

By reducing errors and queueing requirements ROAM is able to constrain peak delay. Variable jitter and delay along the E2E path also result in increased timeouts on these control packets. In reducing contention without a MAC hand-



(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)

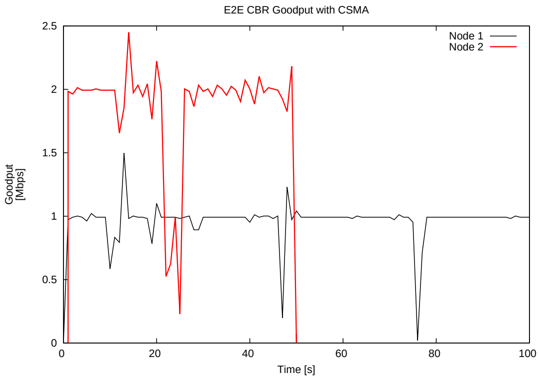


(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

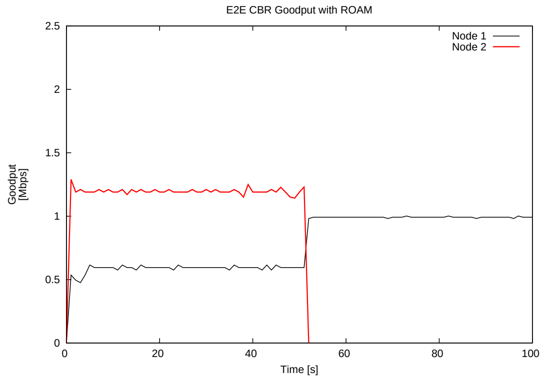
Figure 6.20: Scenario 2(a): Maximum Delay Comparison (RTS/CTS versus ROAM)

shaking mechanism, ROAM makes a greater contribution to overall ISRT performance. Figure 6.21 shows that by reducing transmission rate and tuning packet size ROAM provides more consistent goodput during periods of contention for the same forwarding node. This is at the expense of reduction in peak goodput when the packet size is large. Notably, the extreme packet loss when small packets are transmitted at 5-6Mbps results in goodput that is more than 50% less than the application rate, with CSMA. Although ROAM reduces load by 30%, this is not sufficient to significantly reduce collisions on the busy channel. Figure 6.22 provides an example of the low and widely varying goodput provided with RTS/CTS under these conditions. With mixed traffic rates and frequent busy channel detection, RTS/CTS exchange reduces efficient use of available bandwidth.

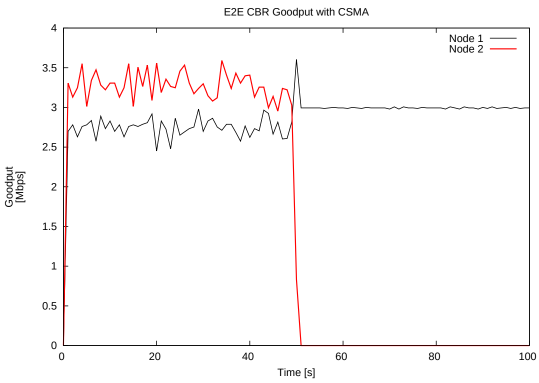
ROAM demonstrates a consistent capability to reduce maximum delay in fig-



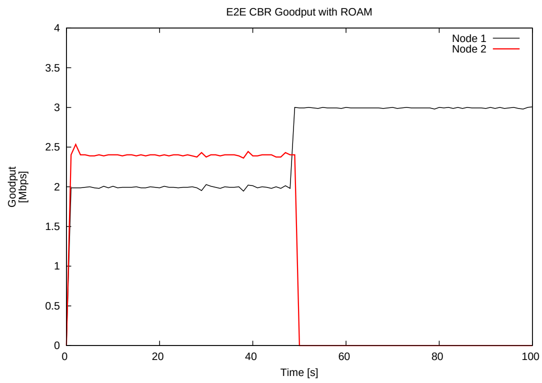
(a) CSMA (TR_1 = 1Mbps, TR_2 = 2Mbps, PS = 1300B)



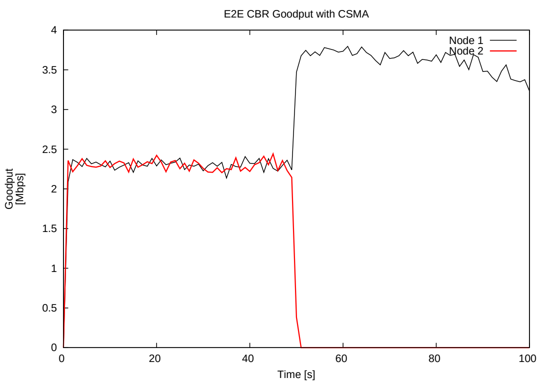
(b) ROAM (TR_1 = 1Mbps, TR_2 = 2Mbps, PS = 1300B)



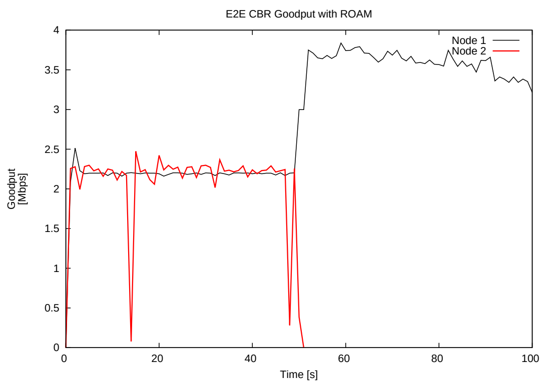
(c) CSMA (TR_1 = 3Mbps, TR_2 = 4Mbps, PS = 900B)



(d) ROAM (TR_1 = 3Mbps, TR_2 = 4Mbps, PS = 900B)

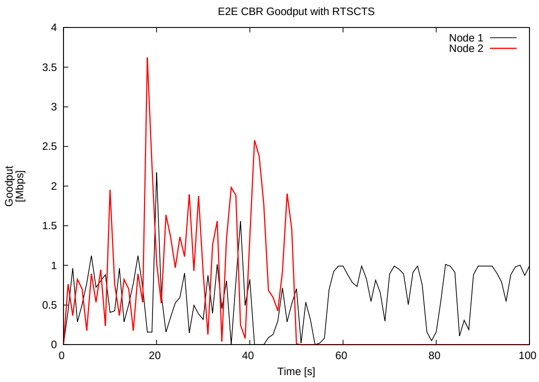


(e) CSMA (TR_1 = 5Mbps, TR_2 = 6Mbps, PS = 500B)

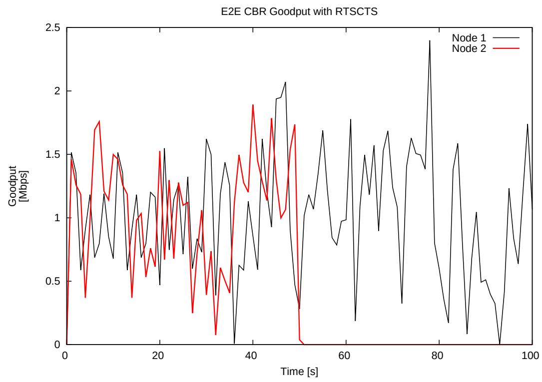


(f) ROAM (TR_1 = 5Mbps, TR_2 = 6Mbps, PS = 500B)

Figure 6.21: Scenario 2(a): Instantaneous E2E CBR Goodput (CSMA versus ROAM)

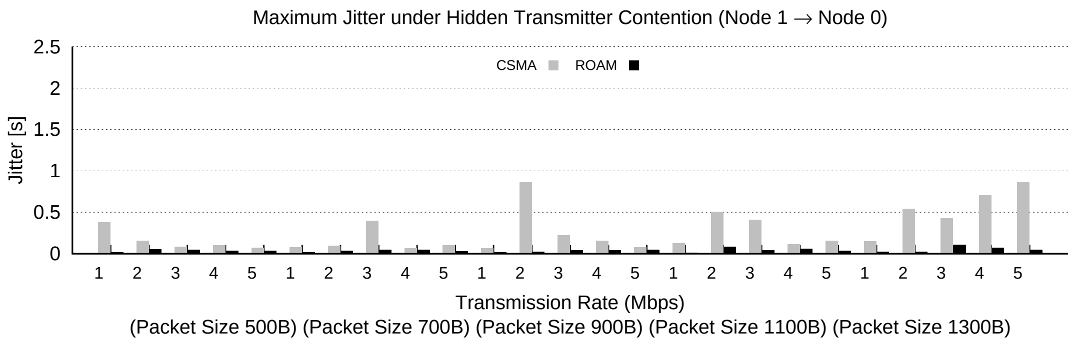


(a) RTS/CTS (TR_1 = 1Mbps, TR_2 = 2Mbps, PS = 1300B)

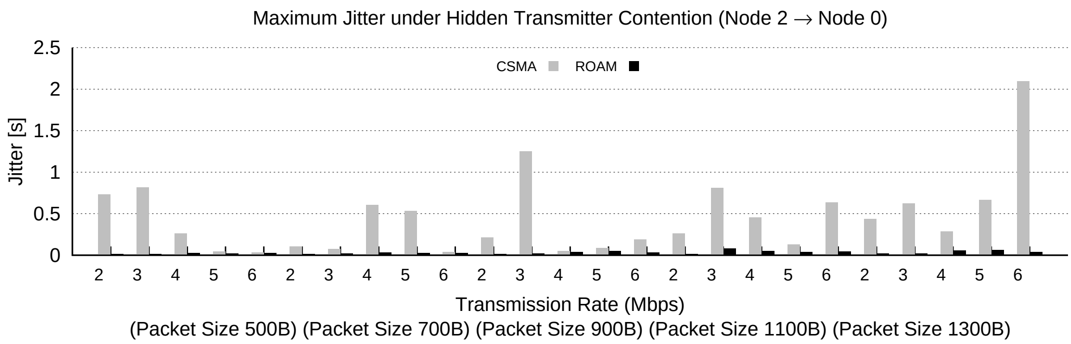


(b) RTS/CTS (TR_1 = 5Mbps, TR_2 = 6Mbps, PS = 500B)

Figure 6.22: Scenario 2(a): Instantaneous E2E CBR Goodput (RTS/CTS versus ROAM)

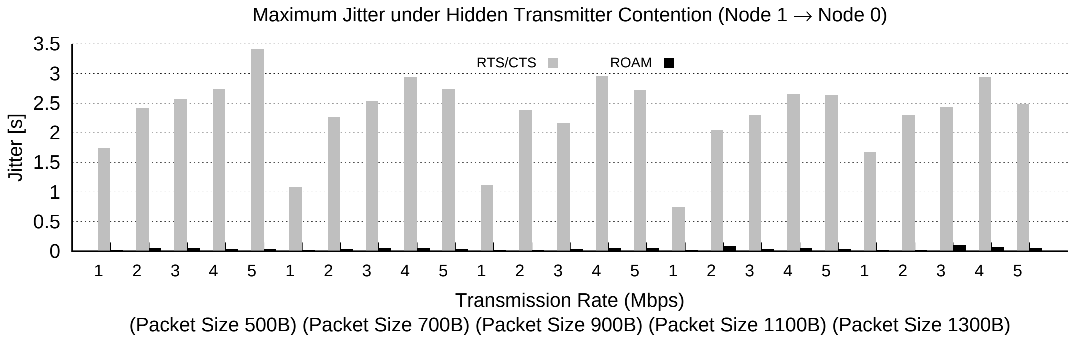


(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)

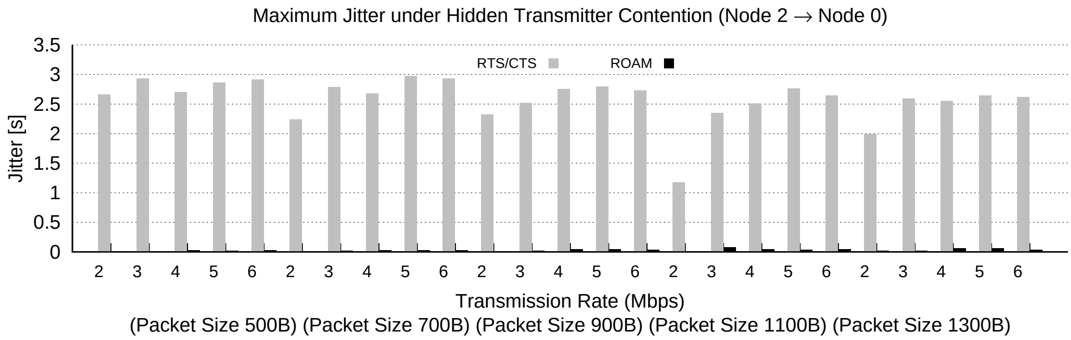


(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

Figure 6.23: Scenario 2(a): Maximum Jitter Comparison (CSMA versus ROAM)

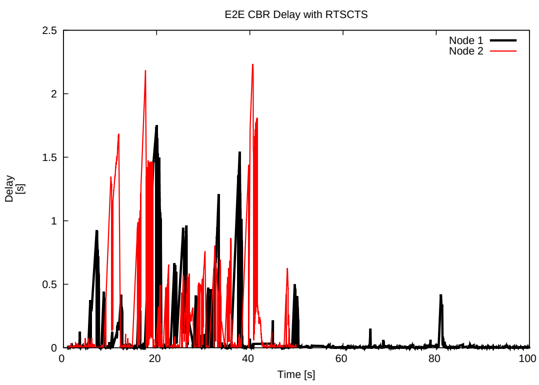


(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)

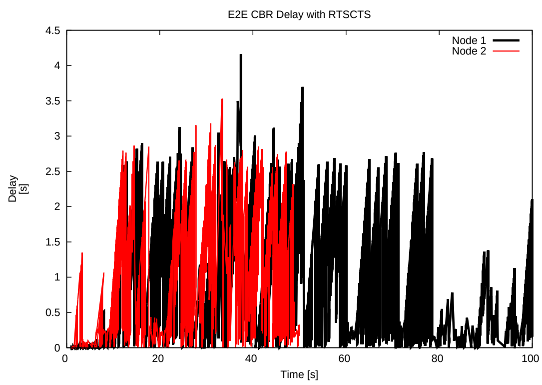


(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

Figure 6.24: Scenario 2(a): Maximum Jitter Comparison (RTS/CTS versus ROAM)

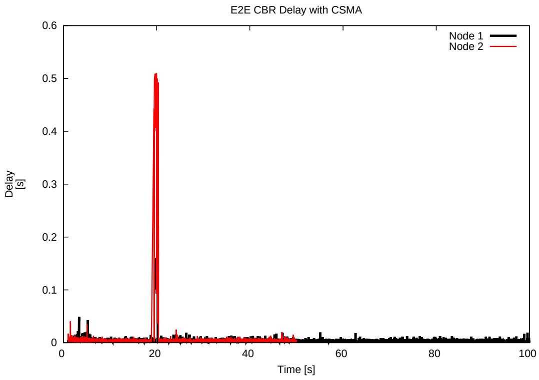


(a) RTS/CTS (TR_1 = 1Mbps, TR_2 = 2Mbps, PS = 1300B)

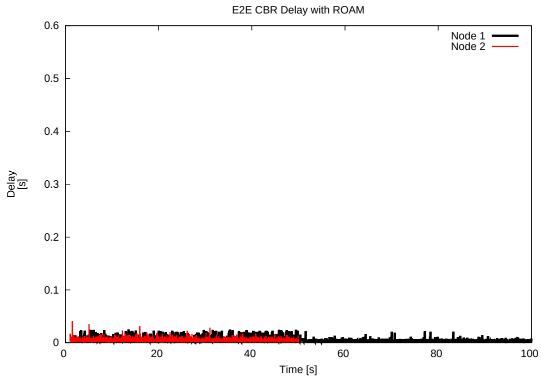


(b) RTS/CTS (TR_1 = 5Mbps, TR_2 = 6Mbps, PS = 500B)

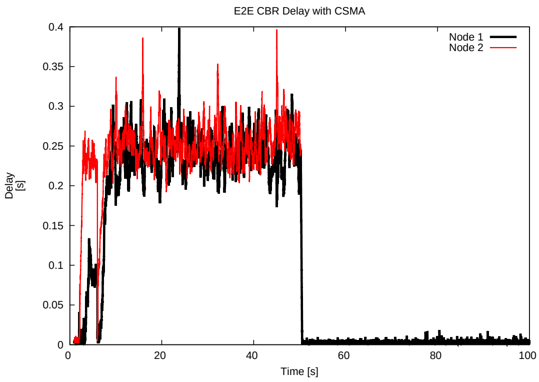
Figure 6.25: Scenario 2(a): Instantaneous E2E Delay (RTS/CTS versus ROAM)



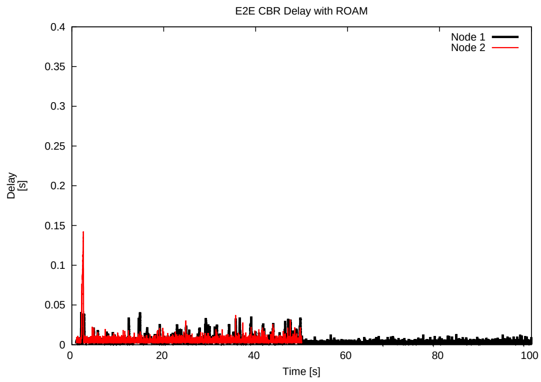
(a) CSMA (TR_1 = 1Mbps, TR_2 = 2Mbps, PS = 1300B)



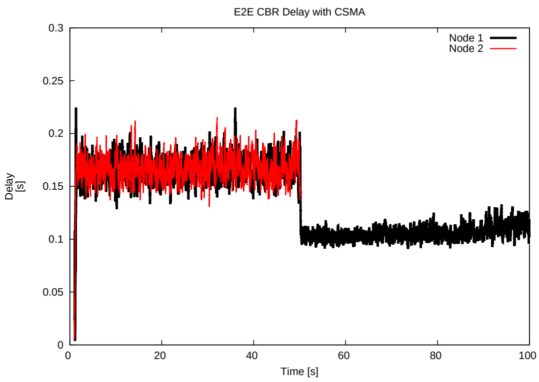
(b) ROAM (TR_1 = 1Mbps, TR_2 = 2Mbps, PS = 1300B)



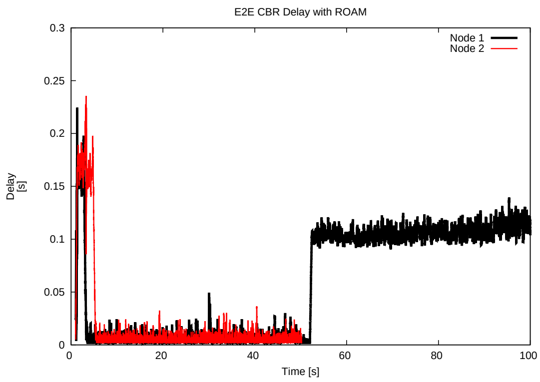
(c) CSMA (TR_1 = 3Mbps, TR_2 = 4Mbps, PS = 900B)



(d) ROAM (TR_1 = 3Mbps, TR_2 = 4Mbps, PS = 900B)



(e) CSMA (TR_1 = 5Mbps, TR_2 = 6Mbps, PS = 500B)



(f) ROAM (TR_1 = 5Mbps, TR_2 = 6Mbps, PS = 500B)

Figure 6.26: Scenario 2(a): Instantaneous E2E Delay (CSMA versus ROAM)

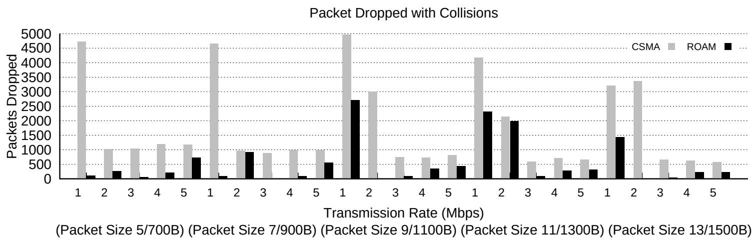
ure 6.26, when compared to CSMA. Maximum delay and delay variation increase drastically with increasing rates and decreasing packet size, using CSMA. While figure 6.26(a) shows that even when the channel is not overloaded, increasing retransmissions on a busy channel ultimately result in increased queueing delay and eventual IFQ overflow and loss.

With mixed transmission rates, the performance of ROAM varies more between nodes as there is more variation in the time taken to identify a hidden transmitter. While this leads to varying levels of performance between nodes, this is still better performance than with CSMA alone.

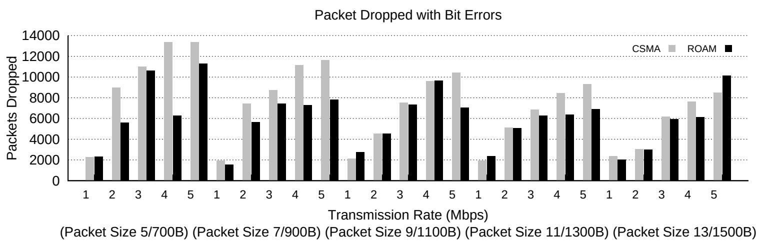
6.2.1.3 Scenario 2(b): Heterogeneous CBR Packet Size

Following consideration of the influence of mixed traffic rates on network performance, this scenario investigates the impact of configuring transmitters when each CBR source utilises a differing packet size. For N1/N2 the packet sizes implemented can be observed at the x-axis in figure 6.27(a).

Figure 6.27 shows the packet dropping performance for CSMA in comparison to ROAM. IFQ overflow and routing errors were negligible with both approaches.

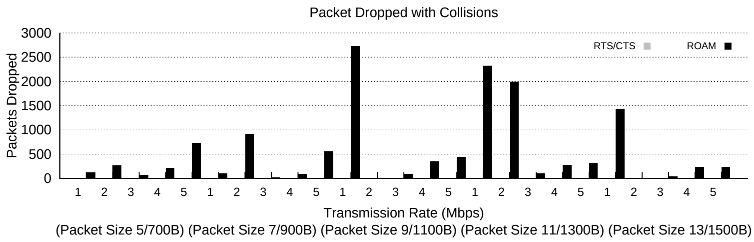


(a) Collision Count

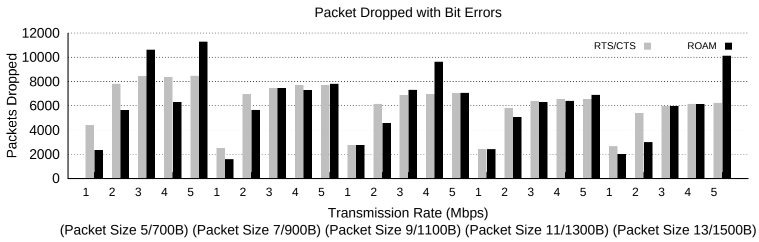


(b) Packet Bit Errors

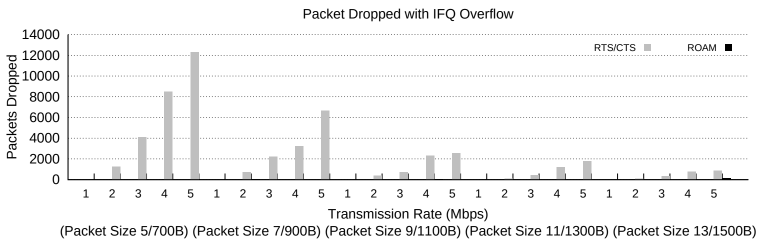
Figure 6.27: Scenario 2(b): Overall Causes of Packet Dropping (CSMA versus ROAM)



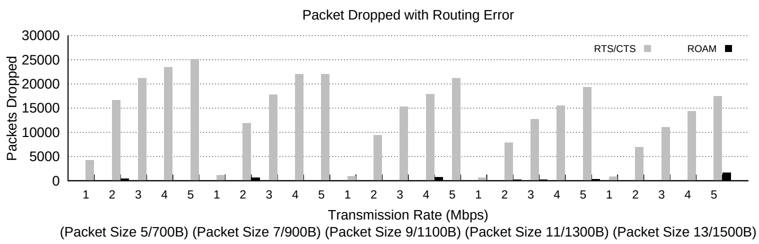
(a) Collision Count



(b) Packet Bit Errors

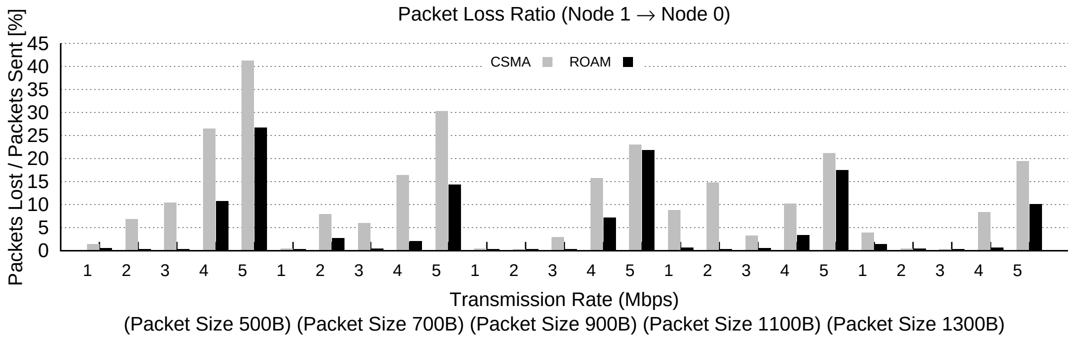


(c) IFQ Full

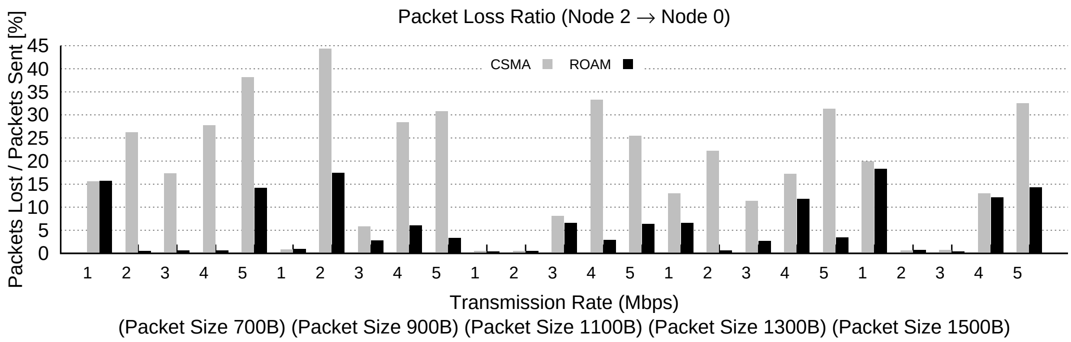


(d) Routing Errors

Figure 6.28: Scenario 2(b): Overall Causes of Packet Dropping (RTS/CTS versus ROAM)



(a) E2E CBR PLR ($N1 \rightarrow N0$)



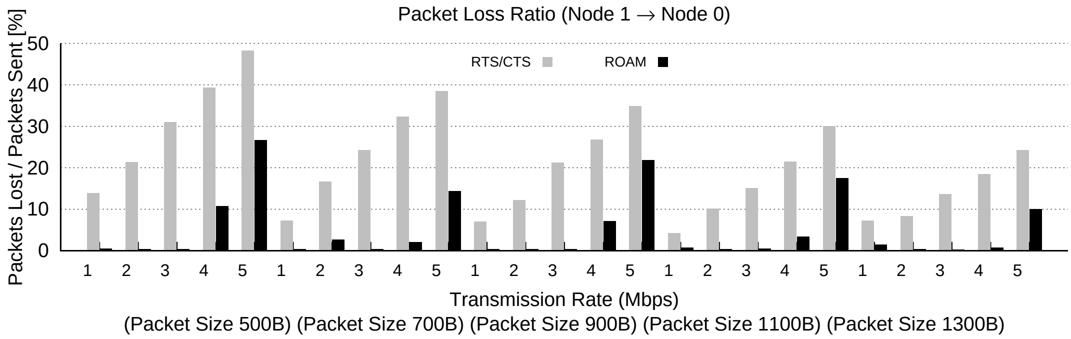
(b) E2E CBR PLR ($N2 \rightarrow N0$)

Figure 6.29: Scenario 2(b): Packet Loss Ratio Comparison (CSMA versus ROAM)

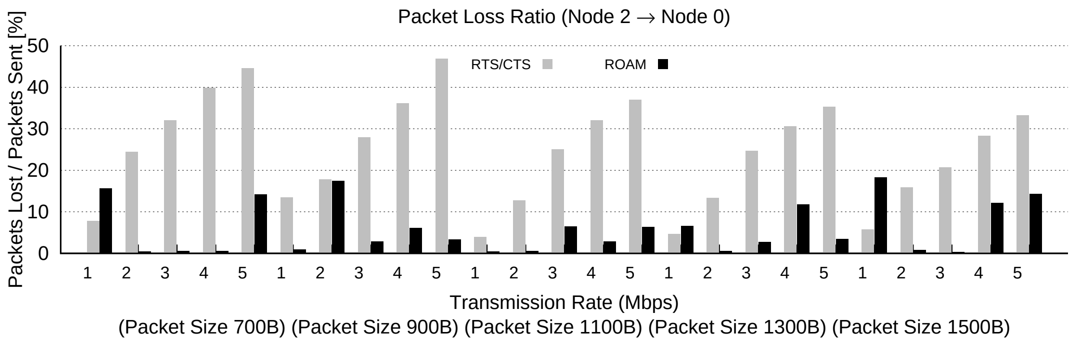
ROAM optimisation significantly reduced collision counts and produced lower packet error incidence than CSMA in all scenarios. As previously shown, ROAM is not capable of reducing collisions to RTS/CTS levels, but bit errors were lower in the majority of scenarios. Overall ROAM was able to produce lower packet dropping than CSMA and comparable loss reduction to RTS/CTS (figure 6.28).

As a result, packet loss ratio was reduced in all scenarios with mixed packet size, when compared to CSMA (figure 6.29). Similar to the results in scenario 1, when channel loading was low and buffer overflow did not occur, ROAM induced marginally (less than 5%) more E2E packet loss than RTS/CTS (figure 6.30). However, in almost all scenarios these packets were more delayed and subject to increased jitter as a result of the increased and variable E2E buffering required by RTS/CTS (figure 6.37).

ROAM produces higher packet loss than RTS/CTS when the lowest traffic rate



(a) E2E CBR PLR ($N1 \rightarrow N0$)

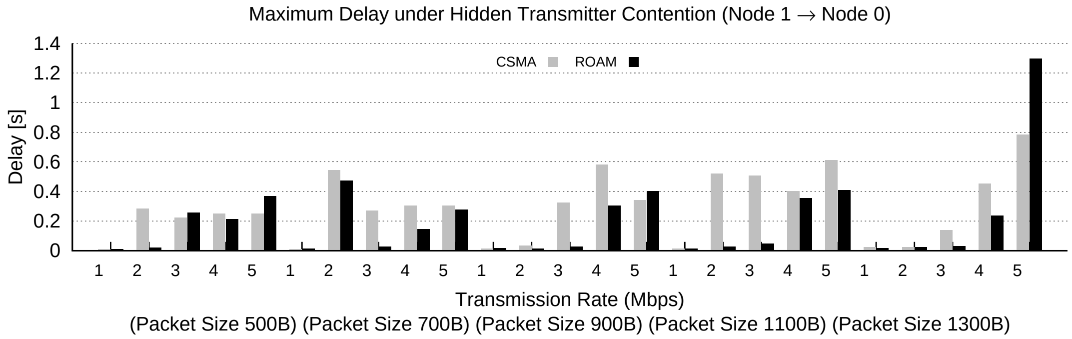


(b) E2E CBR PLR ($N2 \rightarrow N0$)

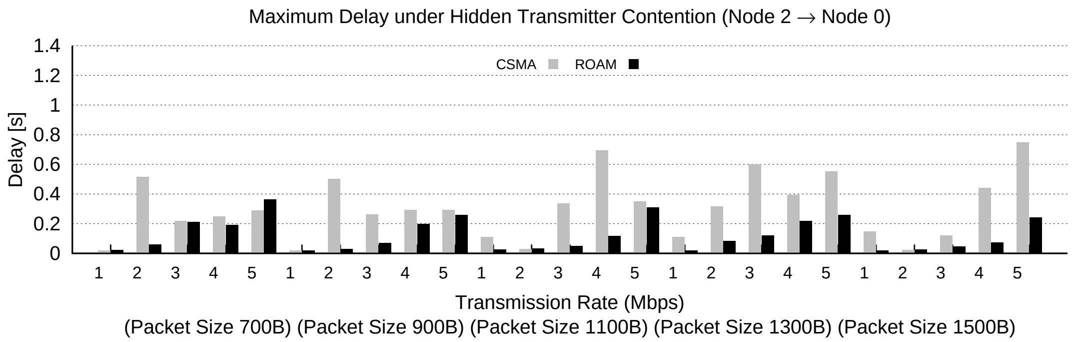
Figure 6.30: Scenario 2(b): Packet Loss Ratio Comparison (RTS/CTS versus ROAM)

of 1Mbps was implemented. It is coincidental that virtual carrier sense performs well within these simulation scenarios, as the remaining results demonstrate optimal results may not be received if more nodes or increased transmission rate are implemented. As a result adaptive cross-layer responsiveness to these conditions is necessary.

Figures 6.31–6.32 demonstrate that maximum delay and jitter were reduced in most scenarios with ROAM in comparison to CSMA. Only with the highest traffic rate, when node 2 implemented the largest packet size of 1500B, was peak delay significantly increased by the optimisation of the network. However, ROAM is capable of providing a comparable level of performance to RTS/CTS in terms of packet loss reduction and a significant improvement in bounding E2E delay (figures 6.37–6.38).



(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)



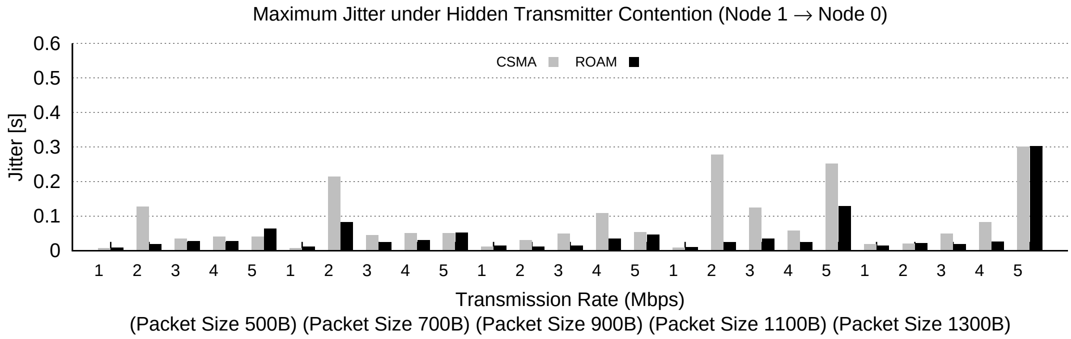
(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

Figure 6.31: Scenario 2(b): Maximum Delay Comparison (CSMA versus ROAM)

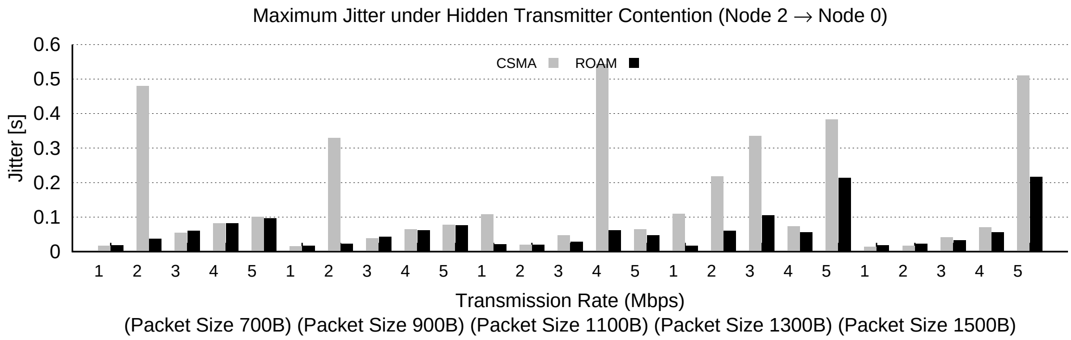
Peak delay is elevated by virtual carrier sense in a MANET, as packets spend longer in an IFQ when the handshaking function is impeded by increased control packet induced errors. By utilising information from ordinary MAC and routing control packets to observe channel conditions, ROAM does not increase competition for scarce resources.

While all nodes transmitted at the same rate in this scenario, the node utilising a lower packet size was subject to higher overheads, reducing the chance for successful transmission (figure 6.33). While the goodput with ROAM is lower than with CSMA during periods of contention, this unfairness of bandwidth distribution can be avoided. In comparison, RTS/CTS ensures more fair distribution of bandwidth but overall degraded performance.

CSMA does not restrain transmission under contention, resulting in increased packet loss but also the highest levels of goodput when transmitter and receiver



(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)



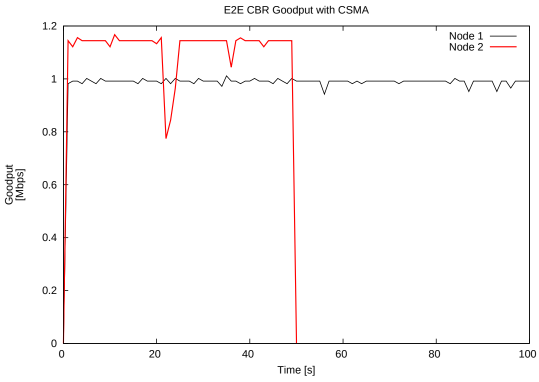
(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

Figure 6.32: Scenario 2(b): Maximum Jitter Comparison (CSMA versus ROAM)

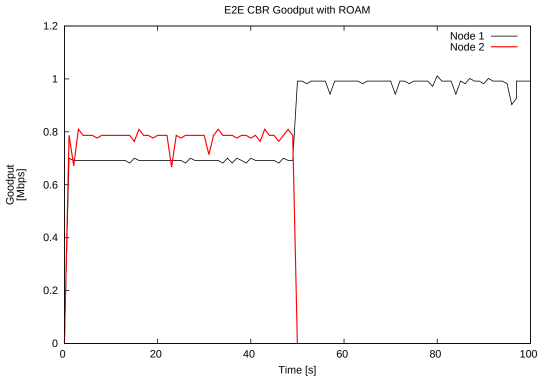
are in range. Both RTS/CTS and ROAM have the effect of limiting goodput with the effect of the former reducing packet collisions and the latter reducing both loss and E2E delay (figure 6.34).

While it has been shown that maximum delay and jitter were reduced by ROAM, figure 6.33 indicates that this was not as a tradeoff for an overall increase in instantaneous delay. With CSMA and RTS/CTS, peak delay coincides with lowest goodput and occurs as a result of increased contention for the single forwarding node. Nodes repeatedly backoff and retransmit during this period as a result of collisions and errors. By reducing traffic during these periods, ROAM is capable of constraining peak delay by reducing retransmission and IFQ requirements.

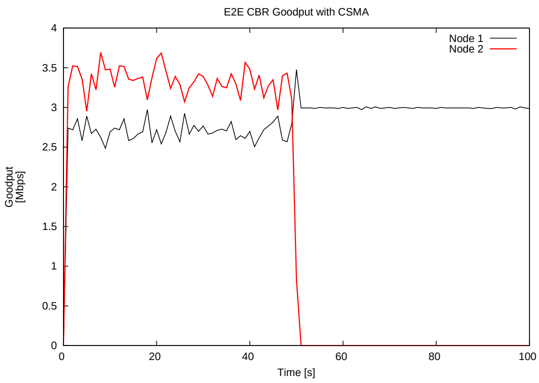
ROAM operates over CSMA and does not tune the DCF or auto-fallback functions. Thus, a comparable goodput and instantaneous delay is provided by both when the application requirements of the network are extreme. ROAM institutes a



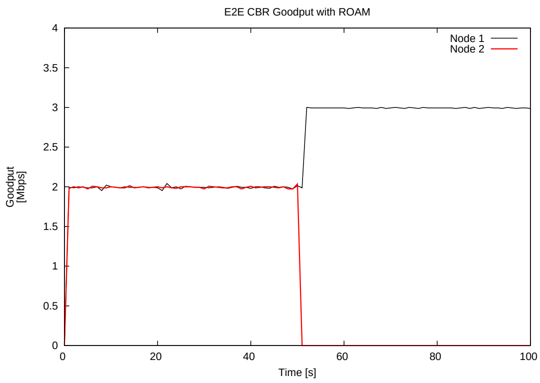
(a) CSMA (TR = 1Mbps, PS_1 = 1300B, PS_2 = 1500B)



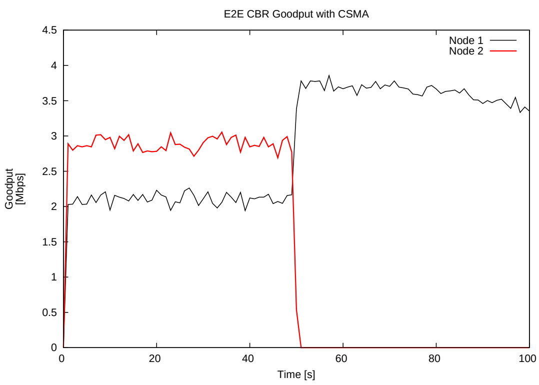
(b) ROAM (TR = 1Mbps, PS_1 = 1300B, PS_2 = 1500B)



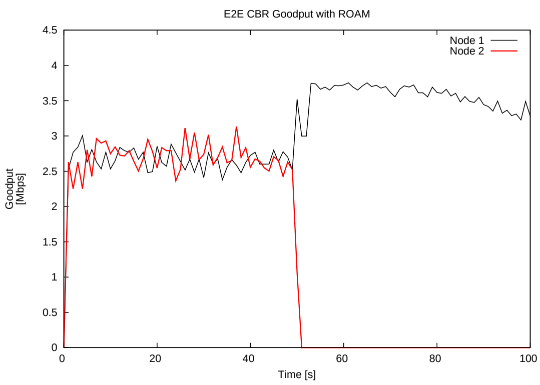
(c) CSMA (TR = 3Mbps, PS_1 = 900B, PS_2 = 1100B)



(d) ROAM (TR = 3Mbps, PS_1 = 900B, PS_2 = 1100B)

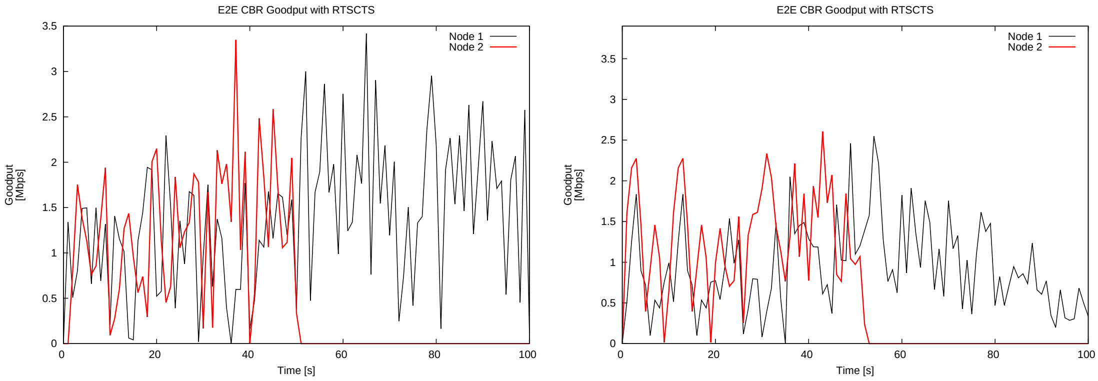


(e) CSMA (TR = 5Mbps, PS_1 = 500B, PS_2 = 700B)



(f) ROAM (TR = 5Mbps, PS_1 = 500B, PS_2 = 700B)

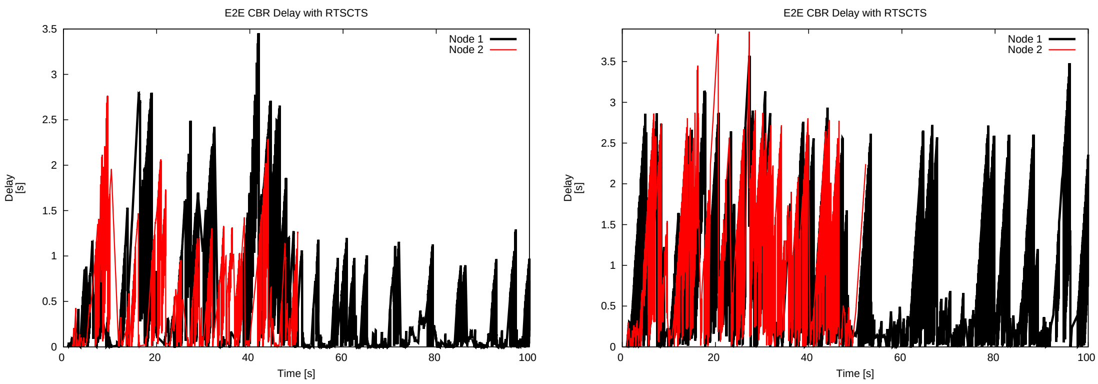
Figure 6.33: Scenario 2(b): Instantaneous E2E CBR Goodput (CSMA versus ROAM)



(a) RTS/CTS (TR = 3Mbps, PS_1 = 900B, PS_2 = 1100B)

(b) RTS/CTS (TR = 5Mbps, PS_1 = 500B, PS_2 = 700B)

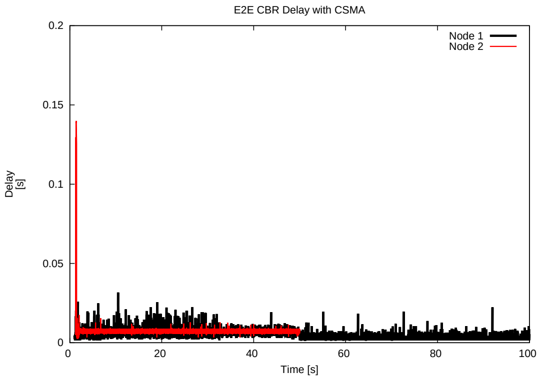
Figure 6.34: Scenario 2(b): Instantaneous E2E CBR Goodput (RTS/CTS versus ROAM)



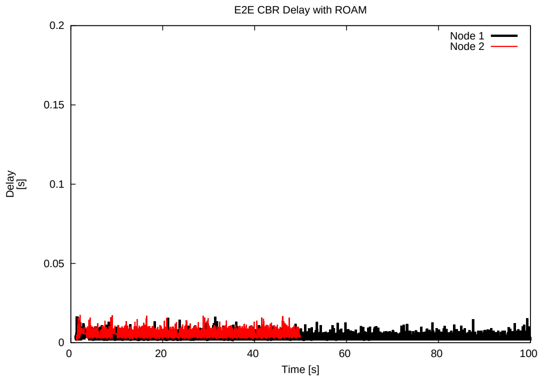
(a) RTS/CTS (TR = 3Mbps, PS_1 = 900B, PS_2 = 1100B)

(b) RTS/CTS (TR = 5Mbps, PS_1 = 500B, PS_2 = 700B)

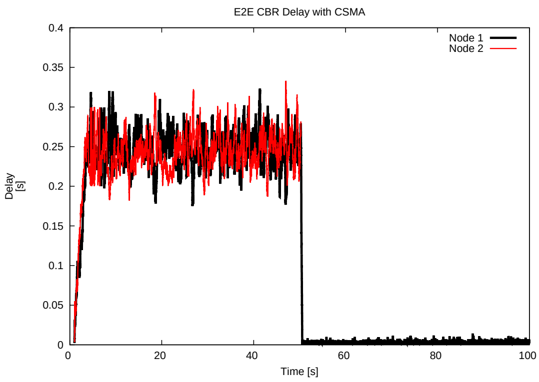
Figure 6.35: Scenario 2(b): Instantaneous E2E Delay (RTS/CTS)



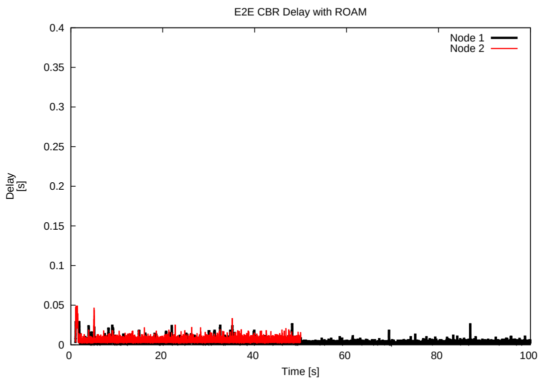
(a) CSMA (TR = 1Mbps, PS_1 = 1300B, PS_2 = 1500B)



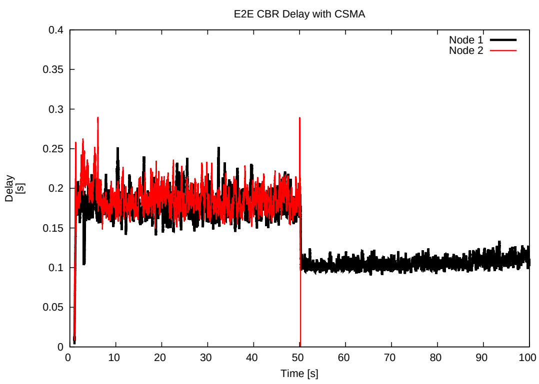
(b) ROAM (TR = 1Mbps, PS_1 = 1300B, PS_2 = 1500B)



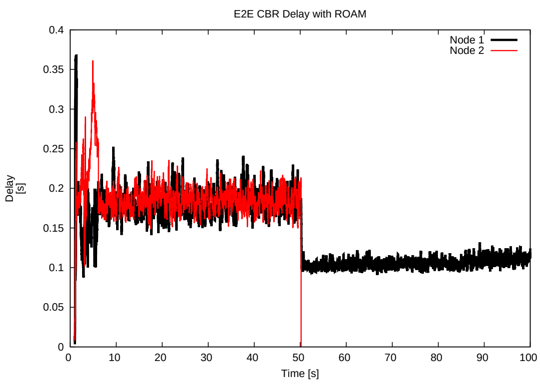
(c) CSMA (TR = 3Mbps, PS_1 = 900B, PS_2 = 1100B)



(d) ROAM (TR = 3Mbps, PS_1 = 900B, PS_2 = 1100B)

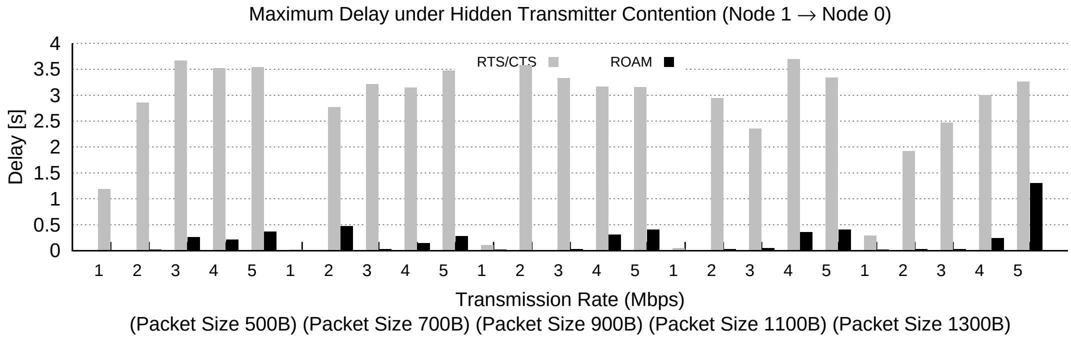


(e) CSMA (TR = 5Mbps, PS_1 = 500B, PS_2 = 700B)

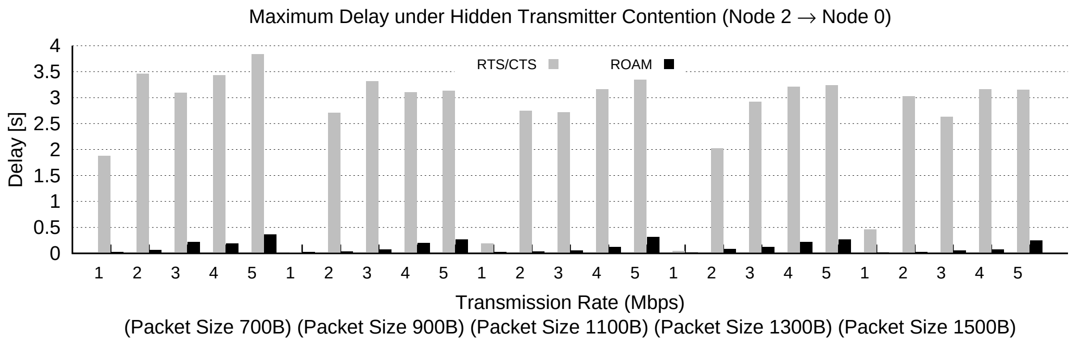


(f) ROAM (TR = 5Mbps, PS_1 = 500B, PS_2 = 700B)

Figure 6.36: Scenario 2(b): Instantaneous E2E Delay (CSMA versus ROAM)



(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)

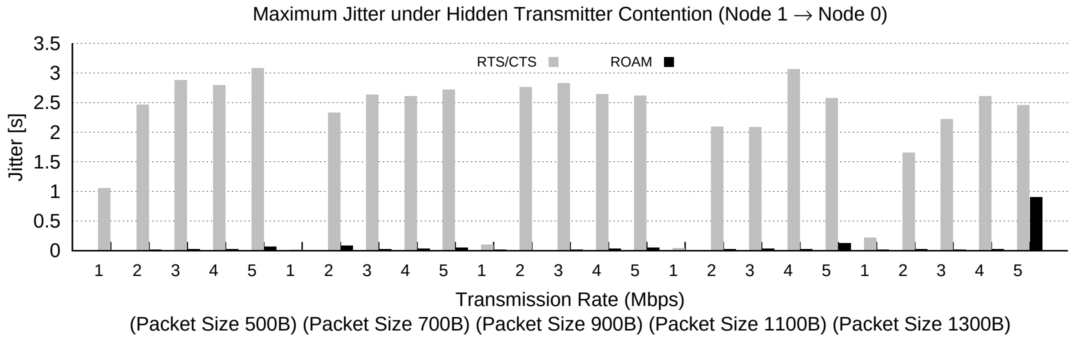


(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

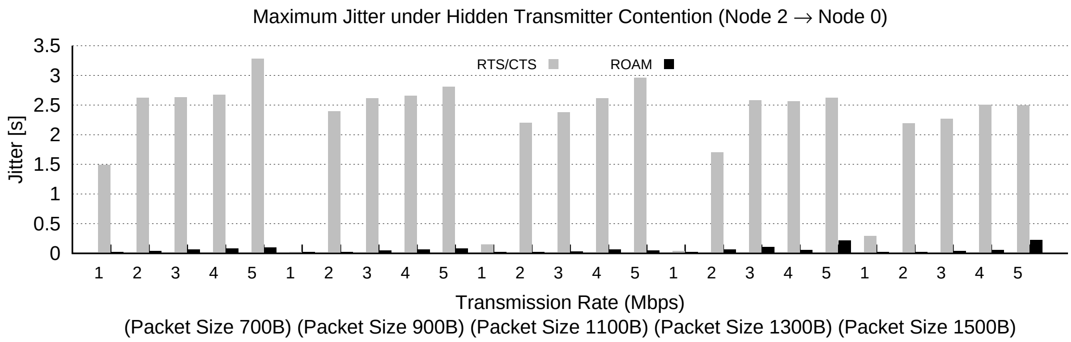
Figure 6.37: Scenario 2(b): Maximum Delay Comparison (RTS/CTS versus ROAM)

blanket 30% load reduction in the presence of a hidden transmitter, but this is not tuned to channel conditions and does not prevent transmissions on a busy channel. As a result, with small packets transmitted at 5-6Mbps, both CBR sources will regularly transmit on a busy channel. While this does not provide a significant performance improvement in comparison to CSMA, ROAM still outperforms traditional collision control with RTS/CTS. Figure 6.34 shows that collision control with virtual carrier sense is not an optimal solution and one that results in very high E2E delay.

Compared to the results in Section 6.2.1.2, ROAM provides greater performance improvements with mixed transmission rate than mixed packet size. When very large packet sizes are implemented, and then increased by ROAM contention control, this increases the likelihood of packet corruption for a given bit error rate.



(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)



(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

Figure 6.38: Scenario 2(b): Maximum Jitter Comparison (RTS/CTS versus ROAM)

The results in Scenario 2 have demonstrated that ROAM is capable of bounding E2E delay under mixed transmission settings, when compared to CSMA and RTS/CTS. However, the middleware is limited by the processes of the underlying protocols, in particular the performance of IEEE 802.11, which is highly dependant on packet size.

6.2.2 Testing MANET Scenario Independence

The previous sections validated the ability of ROAM to provide RT support under the conditions created by fluctuation in transmitted load. The following simulations demonstrate that ROAM is also capable of constraining maximum delay and jitter under the dynamic conditions created by differing number of transmitters on the network as well as changes in topology and node speed. These variables strongly influence MAC layer delay caused by contention, neighbour interference and link quality variation. Multiple CBR sources were added to the MANET in Scenario 1. CBR QoS requirements are the most stringent amongst ISRT applications and provide a representative benchmark for other ISRT scenarios. It is, therefore, assumed that validation with CBR will entail a functional level of support with other RT traffic patterns. Scenario 2 tests ROAM performance with two transmitters in two novel topologies. Finally, ROAM is validated with transmitters moving at higher speeds, creating rapid topology changes and relying on low levels of processing delay.

6.2.2.1 Scenario 1: Variable Number of CBR Sources

In this scenario an increasing number of mobile nodes transmitted CBR streams to a single receiver. This was to test the ability of ROAM to improve performance under variable channel contention from data and control packets and dynamic changes in available bandwidth. In order to fully examine the influence of CBR source count on the network and ROAM, each source transmitted 2Mbps in all sub-scenarios (PS = 700B), therefore total network load increased with each subsequently added source.

The same topology was used as in previous scenarios and, as previously, the mobility of the transmitters led to dynamic topology changes on the periphery of the network. The transmitters were located to be hidden from each other as follows: node 1 hidden from node 2, node 3 from node 4. Each pair and node 5 were out of transmission and CS range of the remaining sources. Nodes 2, 3 and 4 transmitted only for the first 50s of the simulation time, while nodes 1 and 5 transmitted flows for 100s. All of these flows shared common E2E paths to the receiver, therefore, addition of new streams created a bottleneck and increased congestion.

While the addition of nodes 3-5 increased congestion along the E2E path, these transmitters were also further from the receiver and subject to low available bandwidth as a result of the bottleneck link. Therefore, as competition for available resources increased, so did packet loss for each node. Nodes 3 and 4 were subject to the highest loss rates as they were farther from the receiver than nodes 1 and 2 and contended for the same forwarding node.

Overall collision counts were lower with ROAM than CSMA alone. Load reduction under contention tuned by the middleware also provided comparable performance to RTS/CTS with only two CBR flows present, but RTS/CTS provided more consistent reduction even as network congestion increased (Tables 6.1, 6.2,

Table 6.1: Scenario 1: Packets Dropped with CSMA

CSMA	n. CBR Sources			
	2	3	4	5
No Route	277	11809	1745	4882
Packet Error	7786	11691	3282	7128
IFQ Full	0	41	34	642
Collision Count	2081	2097	1558	2444

Table 6.2: Scenario 1: Packets Dropped with ROAM

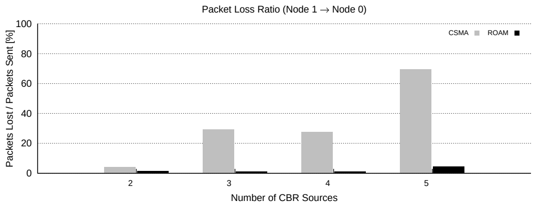
ROAM	n. CBR Sources			
	2	3	4	5
No Route	166	8667	1870	4537
Packet Error	7838	7304	6998	9191
IFQ Full	0	52	37	738
Collision Count	118	1195	961	1063

Table 6.3: Scenario 1: Packets Dropped with RTS/CTS

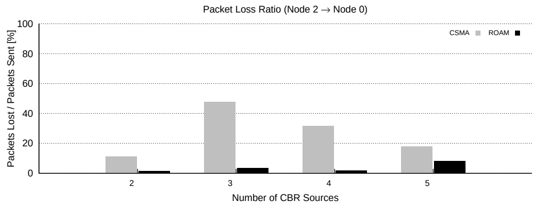
RTS/CTS	n. CBR Sources			
	2	3	4	5
No Route	12260	24278	28822	32646
Packet Error	7222	8752	8451	8910
IFQ Full	551	1383	1904	2319
Collision Count	4	135	392	399

Table 6.4: Scenario 1: Overall Performance for Node 5

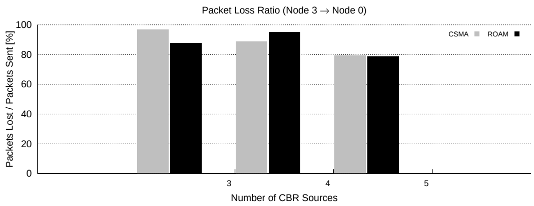
	CSMA	ROAM	RTS/CTS
Maximum Delay (s)	0.606	0.607	1.383
Maximum Jitter (s)	0.406	0.406	1.165
Packet Loss Ratio (%)	25.5	24.8	72.2



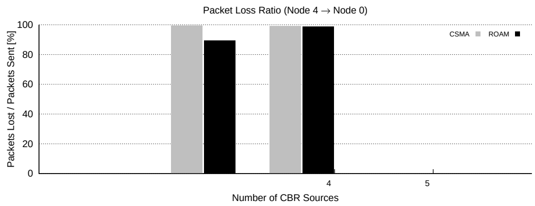
(a) E2E CBR PLR ($N1 \rightarrow N0$)



(b) E2E CBR PLR ($N2 \rightarrow N0$)

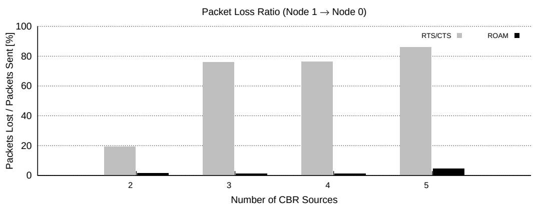


(c) E2E CBR PLR ($N3 \rightarrow N0$)

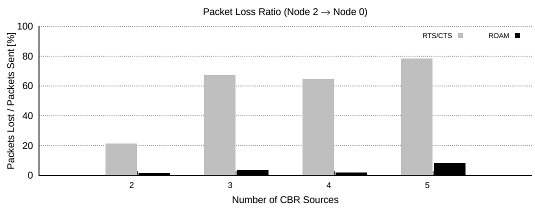


(d) E2E CBR PLR ($N4 \rightarrow N0$)

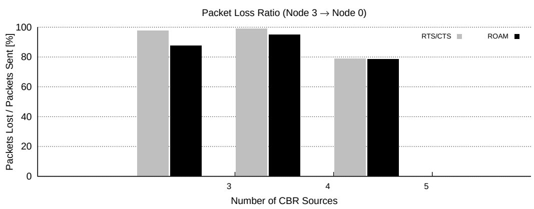
Figure 6.39: Scenario 1: Packet Loss Ratio Comparison (CSMA versus ROAM)



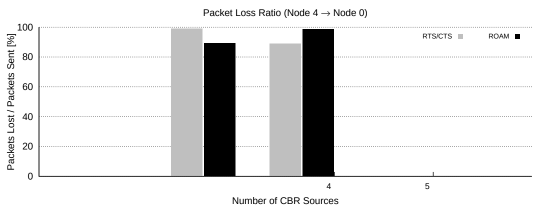
(a) E2E CBR PLR ($N1 \rightarrow N0$)



(b) E2E CBR PLR ($N2 \rightarrow N0$)



(c) E2E CBR PLR ($N3 \rightarrow N0$)

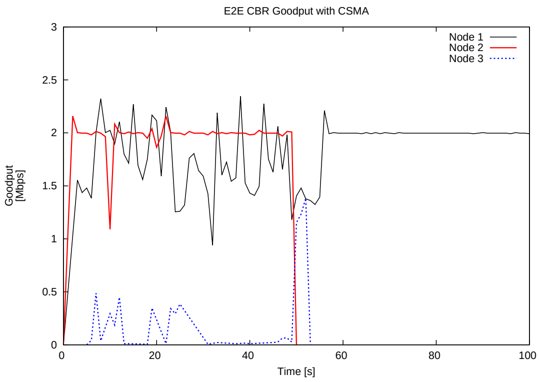


(d) E2E CBR PLR ($N4 \rightarrow N0$)

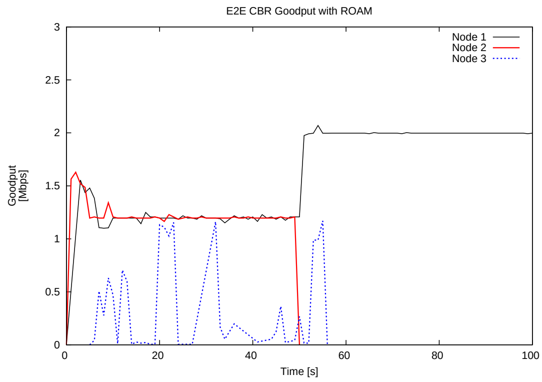
Figure 6.40: Scenario 1: Packet Loss Ratio Comparison (RTS/CTS versus ROAM)

6.3). Packet errors were more prevalent with both approaches compared to CSMA as floods of RTS/CTS packets from one source can interfere with neighbouring flows.

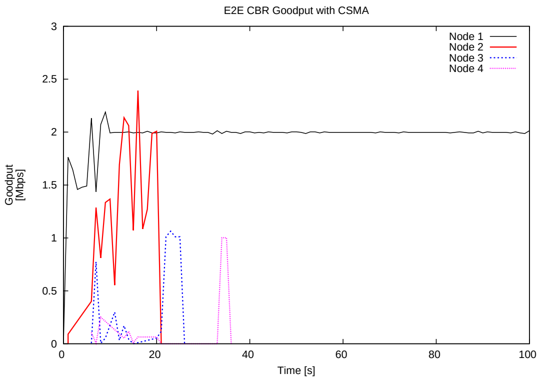
Reducing traffic rate lowers competition for resources, but ROAM also increases application packet size, which increases the probability of packet errors



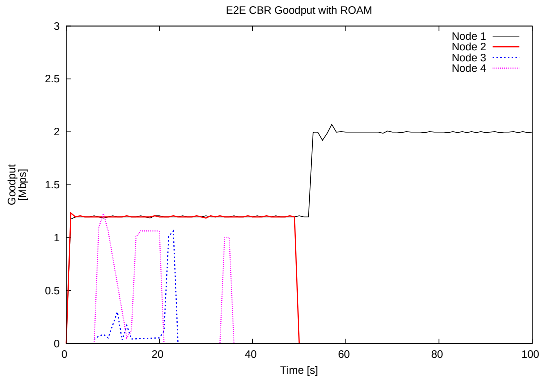
(a) CSMA (CBR Sources = 3)



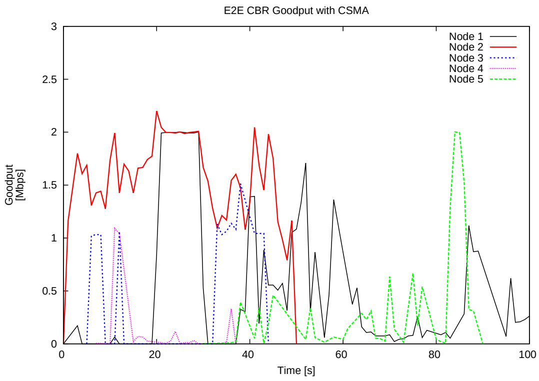
(b) ROAM (CBR Sources = 3)



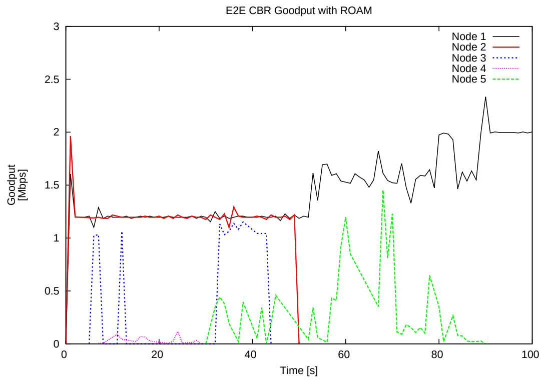
(c) CSMA (CBR Sources = 4)



(d) ROAM (CBR Sources = 4)

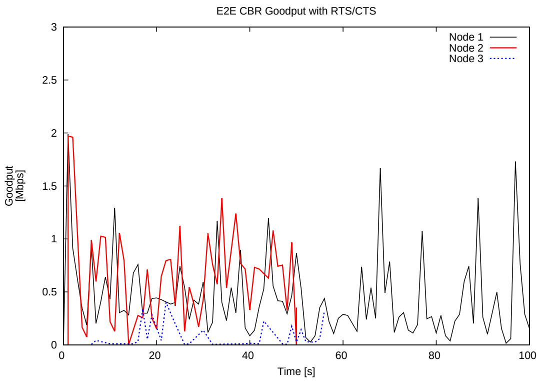


(e) CSMA (CBR Sources = 5)

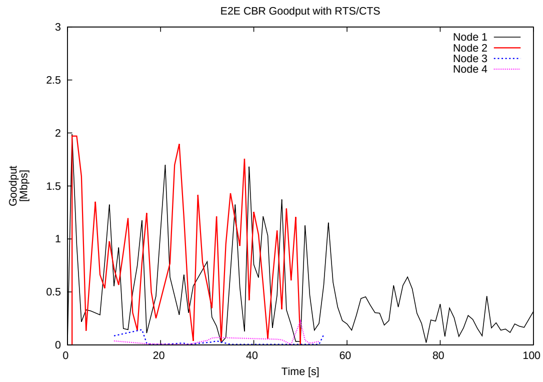


(f) ROAM (CBR Sources = 5)

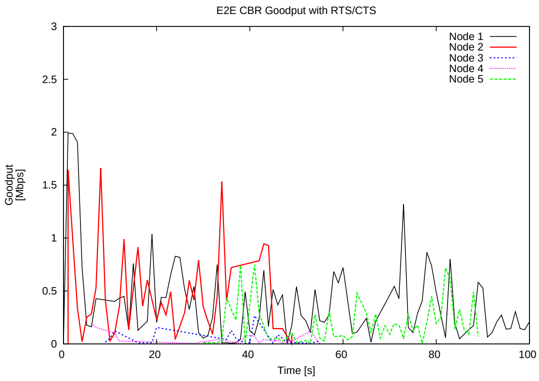
Figure 6.41: Scenario 1: Instantaneous E2E CBR Goodput (CSMA versus ROAM)



(a) RTS/CTS (CBR Sources = 3)



(b) RTS/CTS (CBR Sources = 3)

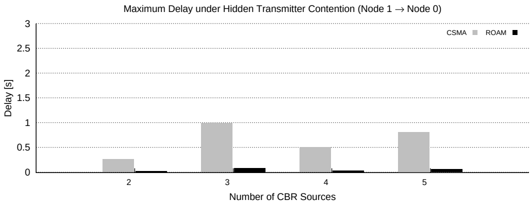


(c) RTS/CTS (CBR Sources = 3)

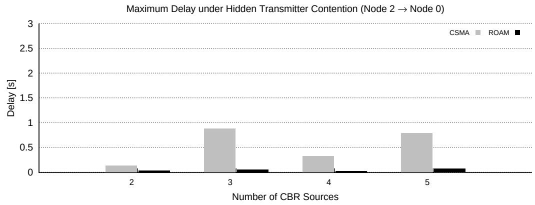
Figure 6.42: Scenario 1: Instantaneous E2E CBR Goodput (RTS/CTS)

under the same bit error rate. Total packet loss with CSMA was generally lower than when RTS/CTS was used. When a fourth and fifth 2Mbps flow were added, packet loss for all flows was extremely high. While ROAM was capable of reducing this loss for nodes 1 and 2, which were closest to the receiver, local load control at nodes 3 and 4 had little influence further along the E2E path. Figures 6.39–6.40 show that packet loss ratios were lower for nodes 1 and 2 with ROAM than the other two schemes, even with multiple CBR streams traversing the network. Packet loss reduction differed between nodes 1 and 2 due to the dynamic nature by which detection of hidden node indicators takes place, which leads to a variable interval before contention mitigation occurs.

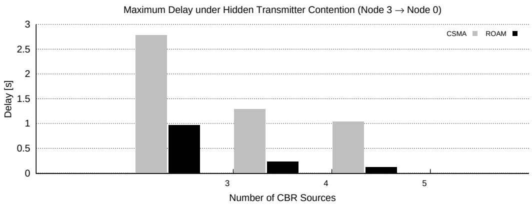
Figures 6.41–6.42 provide examples of instantaneous goodput for a single simulation run, with 3-5 CBR sources. ROAM improves the fairness of bandwidth allocation, ensuring that nodes 1 and 2 receive consistent goodput, but nodes 3-5 receive much lower levels of performance with both CSMA and ROAM. As network congestion increased, ROAM contention control also reduced E2E goodput



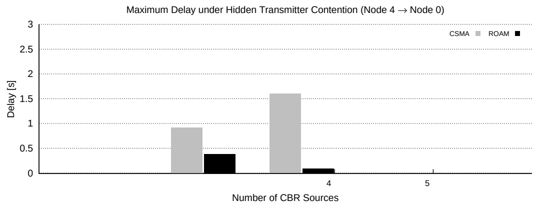
(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)



(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

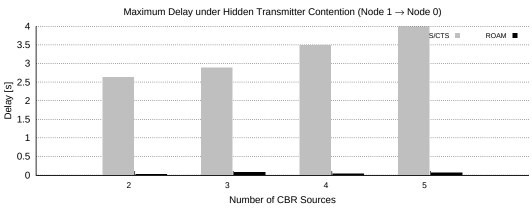


(c) Maximum E2E Delay during Contention ($N3 \rightarrow N0$)

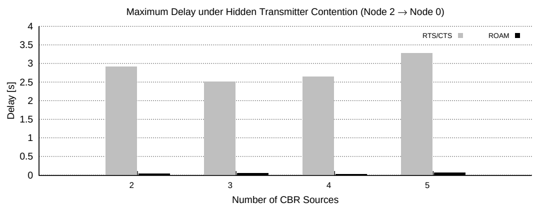


(d) Maximum E2E Delay during Contention ($N4 \rightarrow N0$)

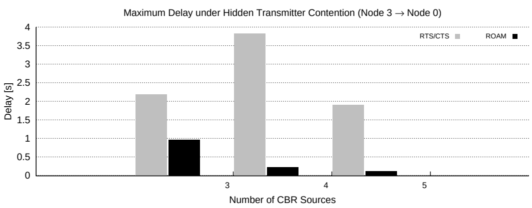
Figure 6.43: Scenario 1: Maximum Delay Comparison (CSMA versus ROAM)



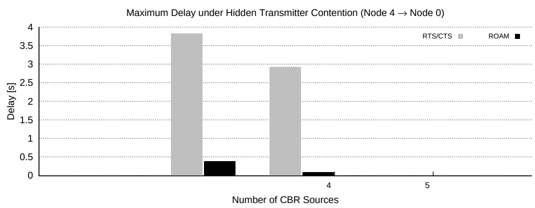
(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)



(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

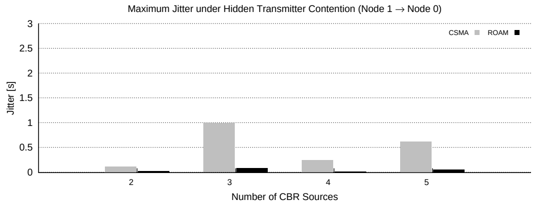


(c) Maximum E2E Delay during Contention ($N3 \rightarrow N0$)

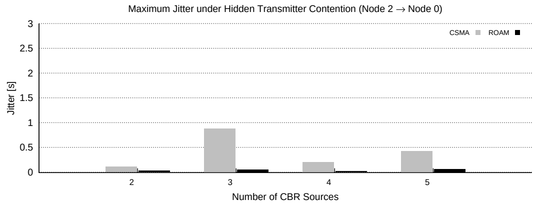


(d) Maximum E2E Delay during Contention ($N4 \rightarrow N0$)

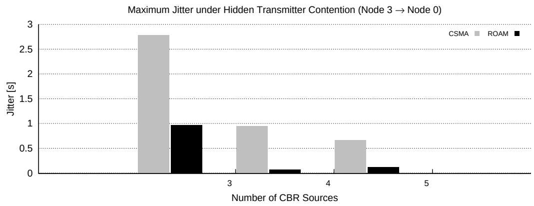
Figure 6.44: Scenario 1: Maximum Delay Comparison (RTS/CTS versus ROAM)



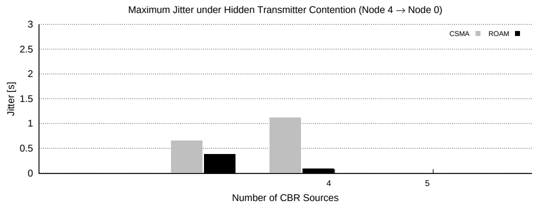
(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)



(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

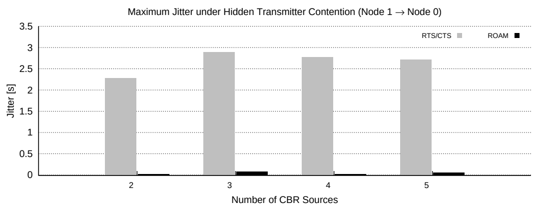


(c) Maximum E2E Jitter during Contention ($N3 \rightarrow N0$)

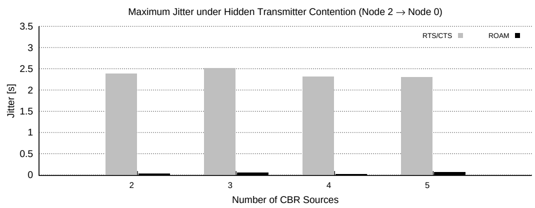


(d) Maximum E2E Jitter during Contention ($N4 \rightarrow N0$)

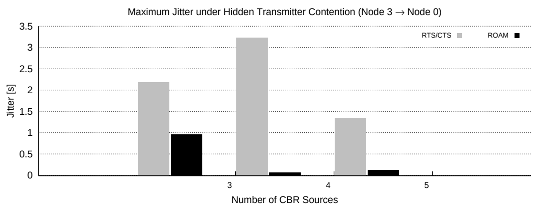
Figure 6.45: Scenario 1: Maximum Jitter Comparison (CSMA versus ROAM)



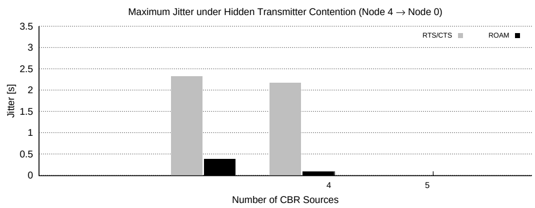
(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)



(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

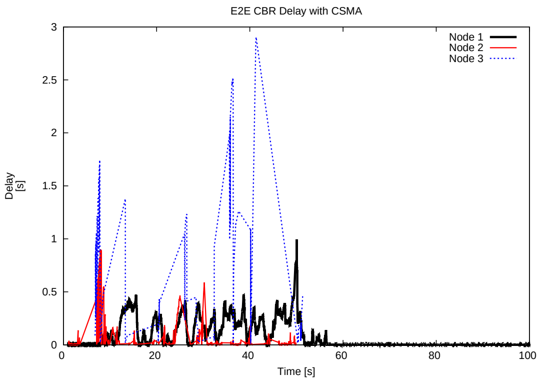


(c) Maximum E2E Jitter during Contention ($N3 \rightarrow N0$)

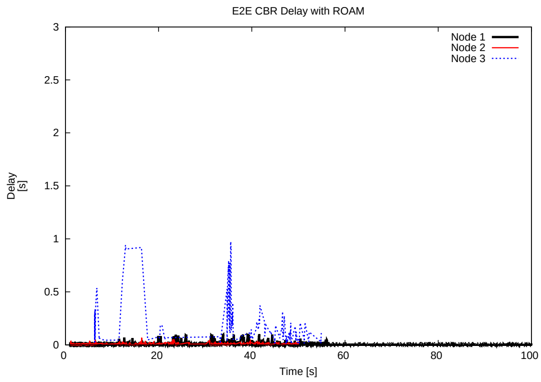


(d) Maximum E2E Jitter during Contention ($N4 \rightarrow N0$)

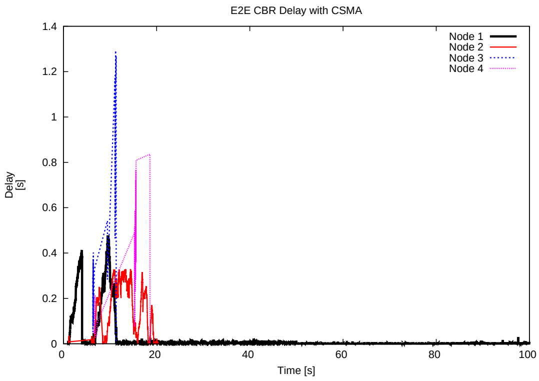
Figure 6.46: Scenario 1: Maximum Jitter Comparison (RTS/CTS versus ROAM)



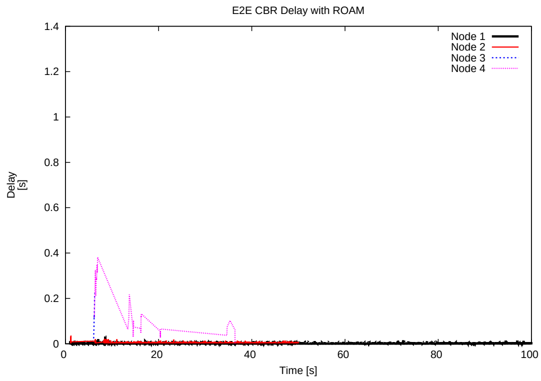
(a) CSMA (CBR Sources = 3)



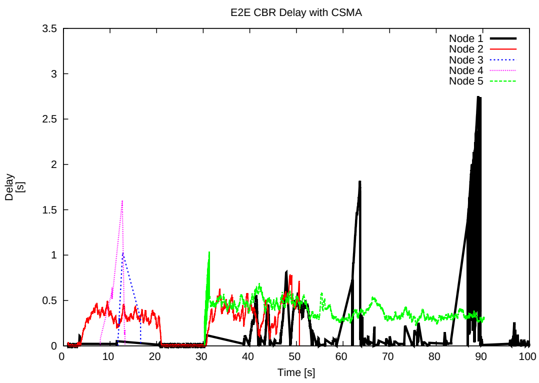
(b) ROAM (CBR Sources = 3)



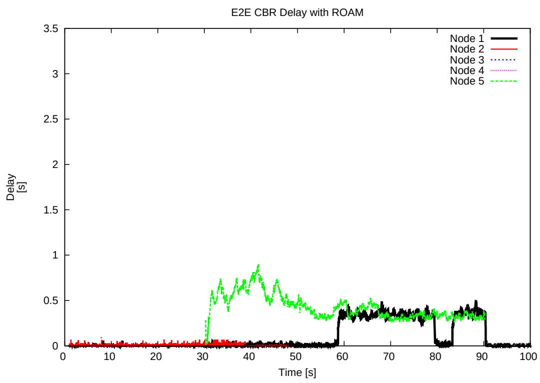
(c) CSMA (CBR Sources = 4)



(d) ROAM (CBR Sources = 4)

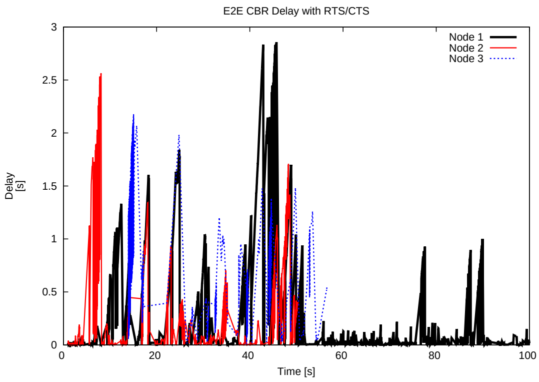


(e) CSMA (CBR Sources = 5)

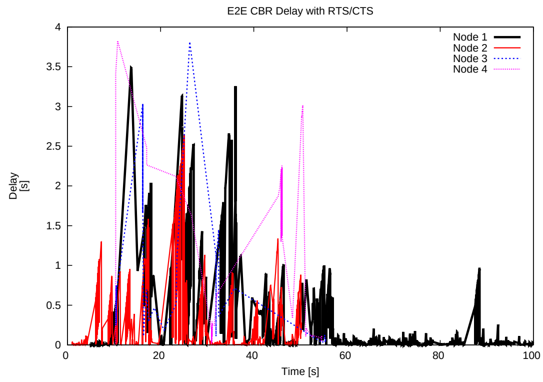


(f) ROAM (CBR Sources = 5)

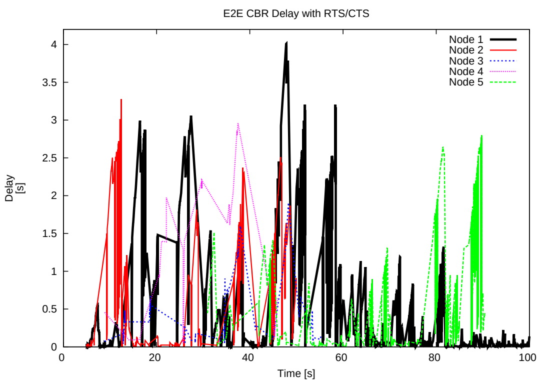
Figure 6.47: Scenario 1: Instantaneous E2E CBR Delay (CSMA versus ROAM)



(a) RTS/CTS (CBR Sources = 3)



(b) RTS/CTS (CBR Sources = 3)



(c) RTS/CTS (CBR Sources = 3)

Figure 6.48: Scenario 1: Instantaneous E2E CBR Delay (RTS/CTS)

for nodes not implementing reduction in transmitted load, causing buffers to fill and overflow along the E2E path. However, figure 6.42 shows that the addition of RTS/CTS control traffic further degraded goodput for all flows.

Increasing the number of CBR sources has the added effect of increasing the circulation of control packets through the network, improving the freshness of routing path information. These control packets also contribute to reduced channel quality and increased and varying channel busy time, or contention delay, creating fluctuation in maximum delay and jitter between the sub-scenarios.

Figures 6.43–6.44 show the maximum delay measured during the first 50s of the simulation time, when nodes 1 and 3 were hidden from nodes 2 and 4, for CSMA versus ROAM and RTS/CTS versus ROAM. They indicate that ROAM was able to reduce maximum delay for all nodes during this period. Maximum E2E delay with CSMA was more than twice the amount resulting when ROAM was implemented. However, ISRT performance was not provided for nodes 3-5 as these were subject to excessive packet loss.

Packet delivery was not improved by ROAM in node 5 as no hidden node was present. Correspondingly, this node transmitted its flows at the full initial rate. While this did not lead to increased packet loss, packets from this CBR source increased congestion and IFQ backlogs along the shared E2E path. Therefore, with this configuration the maximum delay and jitter were increased for node 5 as a result of the improved delivery received by other nodes implementing ROAM contention control.

Increasing the number of CBR sources did not create a significant pattern of increase in maximum delay and jitter between the sub-scenarios with CSMA. The reason for this is the high packet loss experienced by the additional transmitters due to the configuration of the topology, with a low shortest hop count and convergence of available E2E paths. As the number of CBR streams was increased the E2E paths became a series of bottleneck links.

Congestion created a backlog in IFQs of nodes farther from the receiver and packets became subject to extreme queueing delays and peak E2E delay of 1-4s. Eventually buffer provisioning was exceeded and the IFQs overflowed, resulting in high packet loss. There was also fluctuation in maximum jitter for each transmitter in the different simulations as hop-count to the receiver and contention along this path varied for each pair of hidden transmitters that contended for the same channel.

This scenario has shown that ROAM is capable of being implemented on a large scale in multiple hidden transmitter pairs. Performance improvements result for nodes with short E2E paths, however, these results demonstrate that distributed local load reduction cannot prevent congestion in a large multi-hop MANET. It would be necessary for all nodes in the MANET to reduce their forwarding rates in order to prevent the creation of a bottleneck at a shared receiver.

6.2.2.2 Scenario 2: Different Topologies

The previous scenarios have shown that ROAM is able to bound both maximum delay and jitter, under a range of configurations. However, Scenario 1 (Section 6.2.2.1) has indicated that this is only true when the network is not congested. Increased congestion is a feature of a bus topology, with few and convergent E2E paths. Therefore, in this scenario ROAM has been evaluated with two topologies, not implemented in previous scenarios, that have varied mean shortest hop counts (HC) in order to demonstrate the scalability of the middleware architecture. The results are also compared to those with the bus topology used in the previous scenarios.

These novel topologies are a star (HC = 2.1) and ring topology (HC = 2.3), the

Table 6.5: Scenario 2: Packets Dropped in Ring Topology

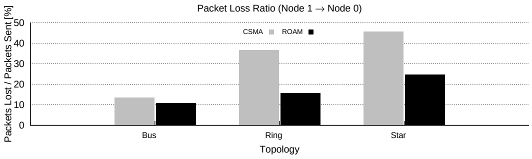
Ring	ROAM	CSMA	RTS
Collision Count	1229	1787	4
Packet Error	14216	14517	9825
IFQ Full	0	0	186
No Route	1410	1792	26717

Table 6.6: Scenario 2: Packets Dropped in Bus Topology

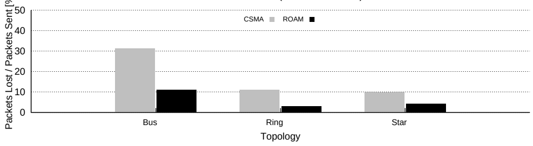
Bus	ROAM	CSMA	RTS
Collision Count	121	1712	2
Packet Error	8647	8135	4683
IFQ Full	0	0	836
No Route	247	418	8621

Table 6.7: Scenario 2: Packets Dropped in Star Topology

Star	ROAM	CSMA	RTS
Collision Count	1247	1558	9
Packet Error	12051	12570	8225
IFQ Full	0	0	2509
No Route	1114	1252	20029

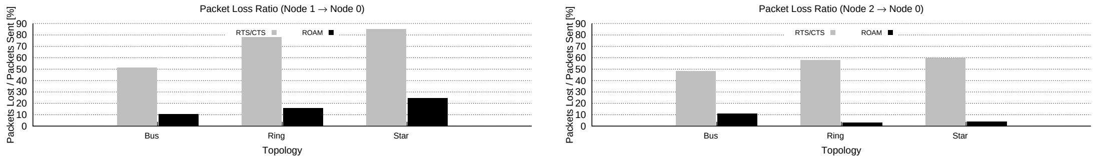


(a) E2E CBR PLR ($N1 \rightarrow N0$)



(b) E2E CBR PLR ($N2 \rightarrow N0$)

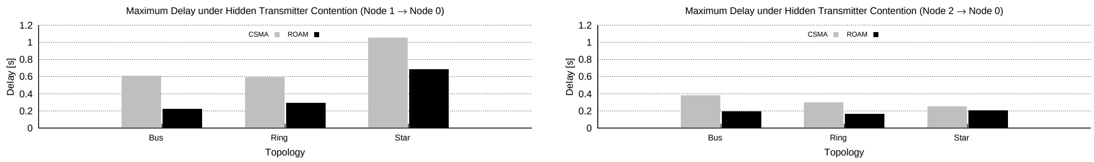
Figure 6.49: Scenario 2: Packet Loss Ratio Comparison (CSMA versus ROAM)



(a) E2E CBR PLR ($N1 \rightarrow N0$)

(b) E2E CBR PLR ($N2 \rightarrow N0$)

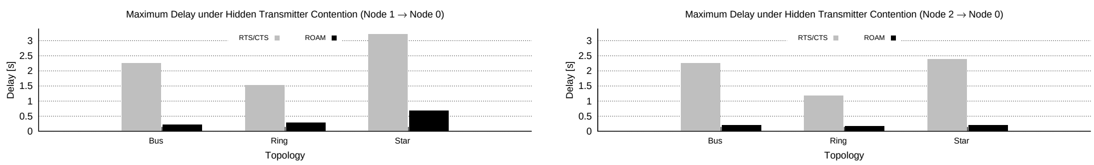
Figure 6.50: Scenario 2: Packet Loss Ratio Comparison (RTS/CTS versus ROAM)



(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)

(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

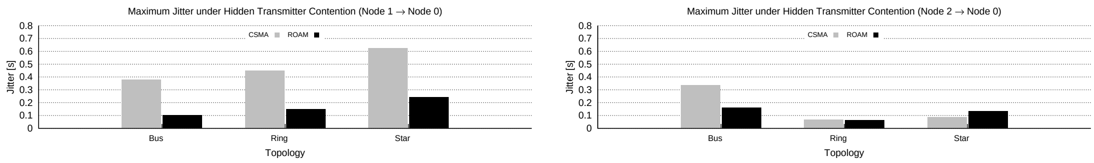
Figure 6.51: Scenario 2: Maximum Delay Comparison (CSMA versus ROAM)



(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)

(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

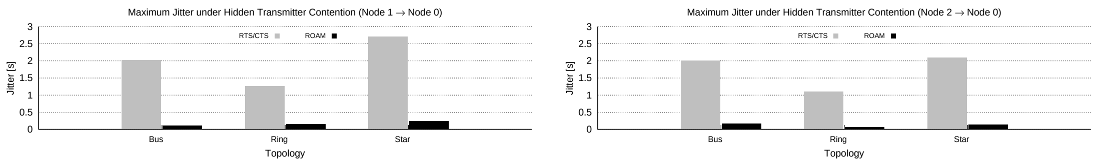
Figure 6.52: Scenario 2: Maximum Delay Comparison (RTS/CTS versus ROAM)



(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)

(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

Figure 6.53: Scenario 2: Maximum Jitter Comparison (CSMA versus ROAM)



(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)

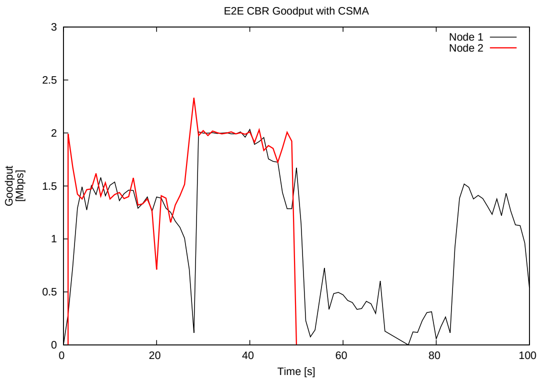
(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

Figure 6.54: Scenario 2: Maximum Jitter Comparison (RTS/CTS versus ROAM)

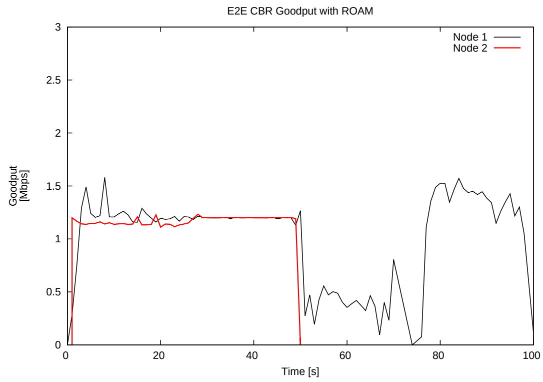
configuration details of which are given in Chapter 3. Two mobile CBR sources transmitted packets of 700B, with the same traffic rate of 2Mbps. In the star topology, node 2 was a static node and, by being located at the centre of the star, was hidden from node 1 which orbited the MANET. In the ring topology node 2 formed part of the ring association of nodes, orbited by node 1. As a result nodes 1 and 2 were exposed to each other when in transmission range. In order to increase mobility through the three topologies, node speeds were increased to 10m/s.

The ROAM contention control optimiser does not rely on particular topological arrangements of nodes. The middleware avoids network-wide signalling of global information, instead acquiring relative local information from control packets intercepted. For example, ROAM relies on control packet RSS as part of evaluation of whether nodes are in range. The optimiser uses a relative comparison between packets intended for the current node and those that are not, and is thus not dependant on inter-nodal distances. As node 1 orbits the bus and star shaped MANETs, these distances continually vary.

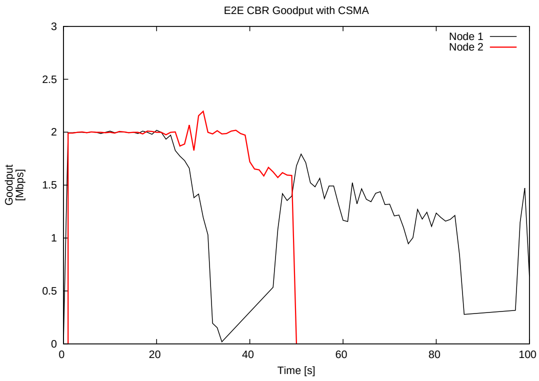
Due to the size and structure of the ring topology, collisions, packet and routing errors were higher with CSMA than in the star and bus topologies, even when a



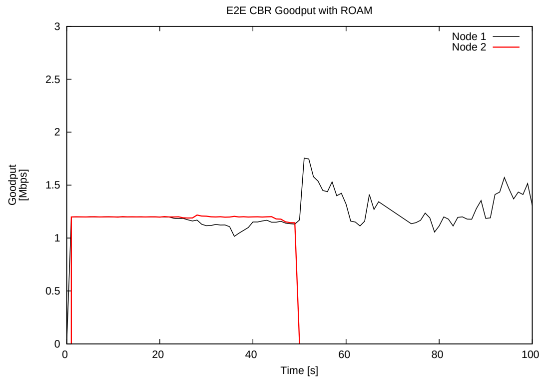
(a) CSMA (Star Topology)



(b) ROAM (Star Topology)



(c) CSMA (Ring Topology)



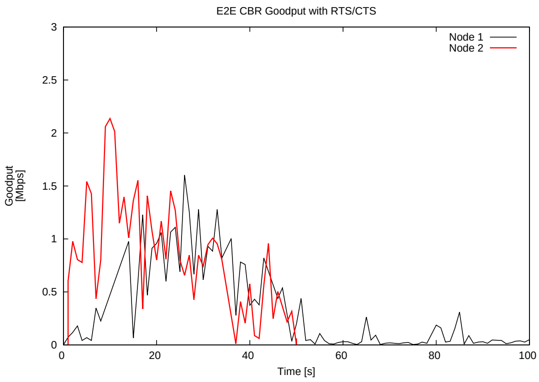
(d) ROAM (Ring Topology)

Figure 6.55: Scenario 2: Instantaneous E2E CBR Goodput (CSMA versus ROAM)

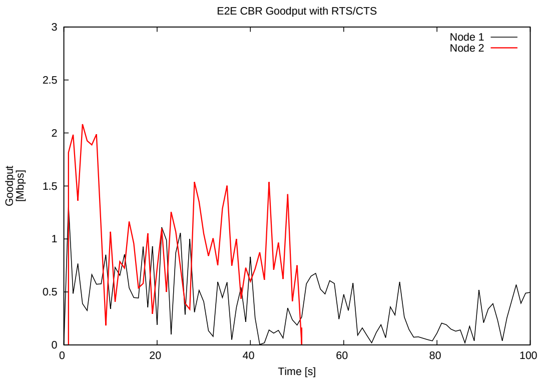
consistent traffic rate of 2Mbps was maintained by all CBR sources. Interference errors are the greatest issue in the star and ring topologies that place more nodes within interference range of each other. However, spatial diversity of E2E paths is limited in the latter and these rapidly converge. Collisions in the bus topology were also more prevalent than in the star due to the path limitations. Within a star topology there are many more available E2E paths, therefore, ad hoc routing RREPs are more often received on less congested links that are then selected.

The ring and bus topology have a similar drawback: that increased congestion and bottleneck links occur due to the convergence of E2E paths. Tables 6.5–6.7 show that collision counts with ROAM and RTS/CTS were reduced when compared to CSMA, but that ROAM provided the best improvement in terms of routing error and buffer overflow reduction. With more paths available to a routing protocol, AODV-UU provides improved performance as many packets can travel fewer hops to the destination.

Bandwidth availability increases as multiple flows utilise differing paths rather



(a) RTS/CTS (Star Topology)



(b) RTS/CTS (Ring Topology)

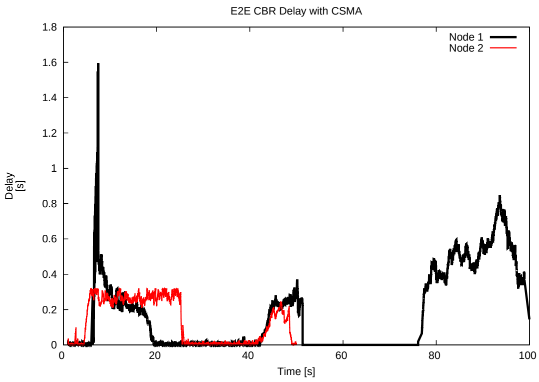
Figure 6.56: Scenario 2: Instantaneous E2E CBR Goodput (RTS/CTS)

than competing for and increasing congestion on a single E2E path. Figure 6.49 shows that ROAM is able to reduce packet loss in comparison to CSMA, to the greatest levels in the ring and then star topologies. ROAM shows the lowest levels of performance optimisation under congestion. When the number of hops to the receiver increases, such as in a bus topology with few varying E2E paths, more nodes within transmission range compete for available bandwidth and channel access.

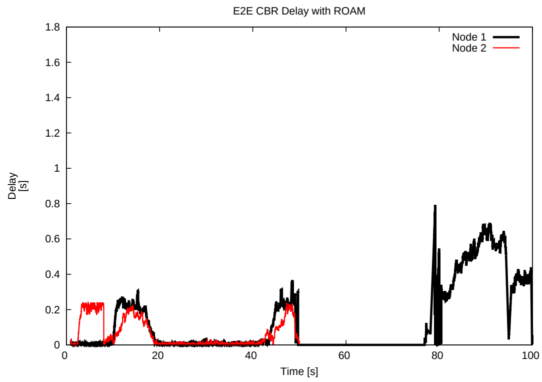
ROAM does not require signalling of global network information, but optimiser performance is influenced by topology. With more INs present, heightened control packet circulation increases the amount of local information available to the optimiser. However, the increased interference capacity of nodes further degrades performance with RTS/CTS, resulting in very high packet loss.

Hidden node contention for a single shared channel generates a similar pattern of performance degradation in a localised area of a network. However, with longer and varied E2E paths in the star topology, delays due to routing and channel quality contribute to maximum delay and jitter. Reducing traffic load when a hidden node is present decreases congestion in a localised area and when multiple paths are available this can result in reduced performance for packets that are then forwarded on links subject to high levels of interference. While ROAM provides improved packet delivery in the ring and star topologies, peak E2E delay is actually not significantly altered for node 2 in the star topology.

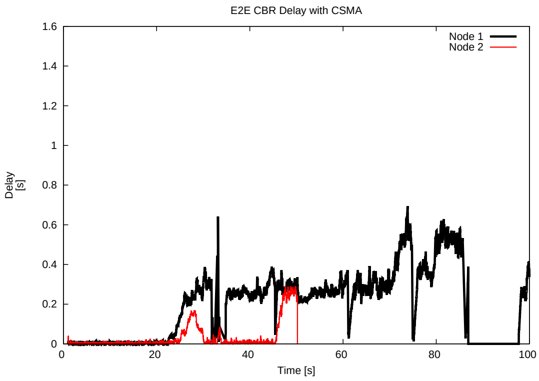
With multiple alternative E2E paths, interference induced retransmissions in the star topology mean that MAC delay can be elevated due to contention further along a path. While ROAM reduces contention on a single shared channel, this is not implemented along the E2E path. As RTS/CTS is implemented across the network, rather than solely in high collision areas, the protocol resulted higher



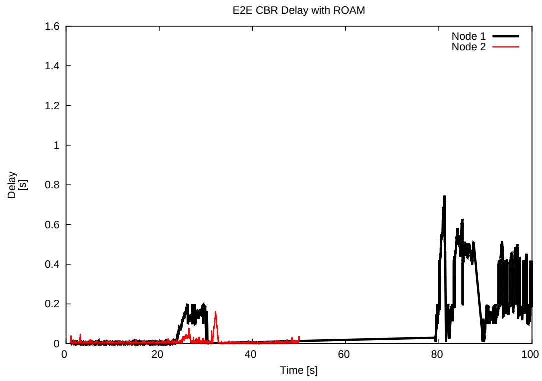
(a) CSMA (Star Topology)



(b) ROAM (Star Topology)



(c) CSMA (Ring Topology)



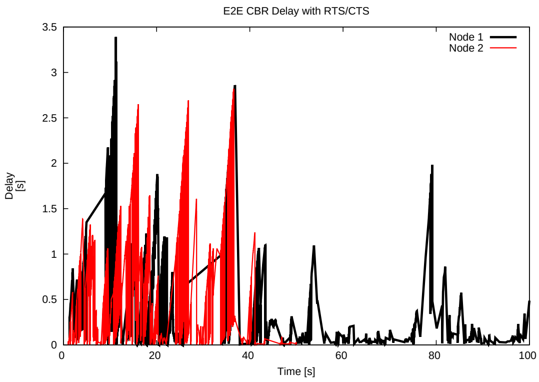
(d) ROAM (Ring Topology)

Figure 6.57: Scenario 2: Instantaneous E2E CBR Delay (CSMA versus ROAM)

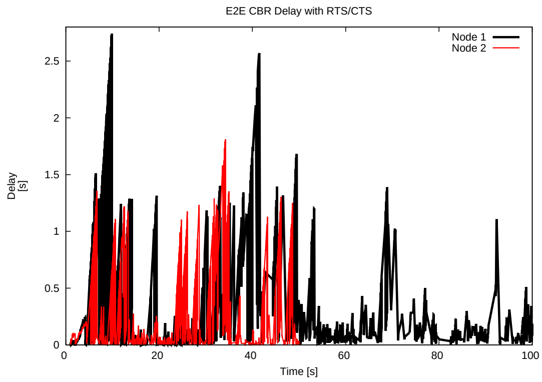
E2E delays and more frequent IFQ overflow in the ring and star topologies than the previously investigated bus topology.

The queueing required for RTS/CTS handshaking results in increased delay and jitter. As a result, ROAM provides improved performance by constraining E2E delay in all topologies. As multiple nodes are in interference range and will contend for channel access with each other in a star or ring topology, MAC layer delay and consequently E2E delay is higher than in a bus topology. Reducing this contention from hidden transmitters enables bounded E2E delay.

Figure 6.55 shows the goodput during one run of the simulation in each topology. With CSMA, the auto-fallback mechanism at the MAC layer steps up the frame rate when noise decreases, and a good quality channel is available. Without detection of other flows, the maximum available bandwidth is used at the expense of neighbouring transmissions and loss of packets. Therefore, use of RTS/CTS and ROAM, that respond to hidden transmitters, results in lower goodput (figure 6.56). Additionally, goodput is lowered while RTS/CTS causes nodes to



(a) RTS/CTS (Star Topology)



(b) RTS/CTS (Ring Topology)

Figure 6.58: Scenario 2: Instantaneous E2E CBR Delay (RTS/CTS)

backoff during the handshaking process. The use of ROAM enables more consistent goodput for both transmitters than with CSMA, and higher and less varying goodput than RTS/CTS. Both CSMA and RTS/CTS demonstrate an unfair distribution of bandwidth as the peak goodput of one CBR source coincides with the lowest goodput of the second, when competing for the same forwarding node. In comparison, by reducing transmitted load ROAM enables fairer bandwidth usage by the MAC layer.

Figure 6.57 shows instantaneous delay was also reduced by ROAM in comparison to CSMA. However, elevated delay at handoff is not prevented by the optimiser. The peak delay and jitter for node 1, with ROAM, occurred when transmission rate and packet size were returned to their original settings after node 2 leaves the network. During the period when the two nodes are transmitting, although ROAM has reduced application transmitted load, packets are delayed due to a buffer backlog. As the transmitters orbit the star topology, handoff occurs between one forwarding node and the next, during which packets are buffered as new E2E paths are set up. Packets are continually enqueued as the IFQ begins to empty, resulting in a gradual lowering of queueing delay.

After this initial period ROAM is able to increase transmission rates again as the hidden transmitter goes out of range, implementing load control if it is detected again. With RTS/CTS, handshaking takes place whenever a packet is to be transmitted, enabling the continual detection of contention but also leading to much higher instantaneous E2E delays (figures 6.56).

6.2.2.3 Scenario 3: Different Mobile Node Speed

This scenario is to demonstrate that ROAM is able to constrain network delay and jitter under the rapidly dynamic conditions created by variation in node velocity. The bus topology and simulation configuration used in previous scenarios has been implemented with two mobile CBR sources transmitting traffic at a rate of 2Mbps using a packet size of 700B. Mobile node speed was elevated between 10-50m/s, at intervals of 10m/s and compared to results for the speed used in previous scenarios (1m/s).

Table 6.8: Scenario 3: Packets Dropped with CSMA

CSMA Speed (m/s)	No Route	Pkt Error	IFQ Full	Collision Count
1	277	7786	0	2081
10	505	8011	0	1828
20	839	10071	0	1740
30	1976	10029	0	1738
40	2229	9644	209	1327
50	2634	9808	41	1375

Table 6.9: Scenario 3: Packets Dropped with ROAM

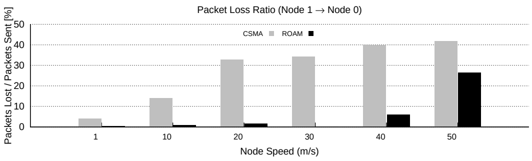
ROAM Speed (m/s)	No Route	Pkt Error	IFQ Full	Collision Count
1	166	7838	0	1718
10	247	4886	0	199
20	602	4541	0	124
30	987	4313	8	174
40	1930	5350	0	98
50	1610	5157	0	89

Tables 6.8–6.10 show the overall causes of packet dropping for the three sub-cases: CSMA, RTS/CTS and ROAM. RTS/CTS and ROAM produced lower collision counts than CSMA, which reduced with increasing node speeds. However, in comparison to previous scenarios ROAM and RTS/CTS provided similar levels of performance in terms of packet errors at speeds of 20-50m/s. When nodes move at higher velocities, E2E routes are setup and torn down on a more regular basis. As a result, nodes move more frequently between busy and available channels.

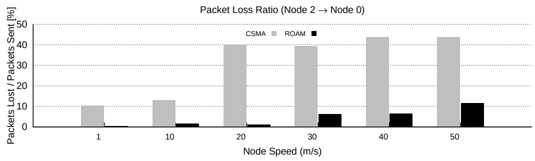
CSMA will continue to transmit onto busy channels, resulting in increased collisions. In contrast, RTS/CTS and ROAM have the result of reducing MAC layer

Table 6.10: Scenario 3: Packets Dropped with RTS/CTS

RTS/CTS Speed (m/s)	No Route	Pkt Error	IFQ Full	Collision Count
1	12260	7222	551	4
10	7483	4392	730	1
20	8422	4779	731	2
30	9316	4800	590	1
40	9760	5504	691	3
50	9739	5292	740	3

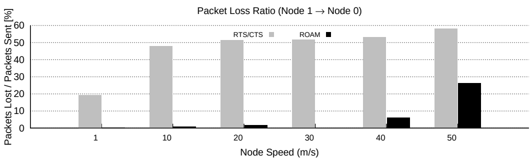


(a) E2E CBR PLR ($N1 \rightarrow N0$)

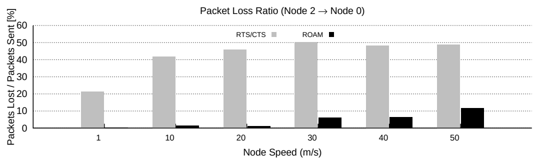


(b) E2E CBR PLR ($N2 \rightarrow N0$)

Figure 6.59: Scenario 3: Packet Loss Ratio Comparison (CSMA versus ROAM)

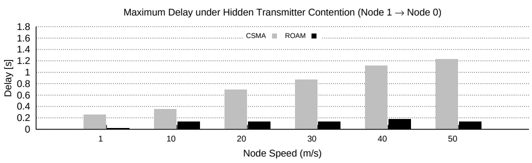


(a) E2E CBR PLR ($N1 \rightarrow N0$)

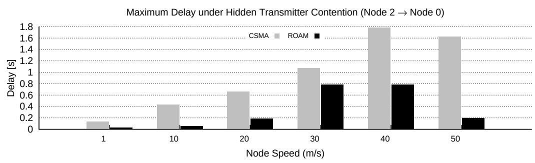


(b) E2E CBR PLR ($N2 \rightarrow N0$)

Figure 6.60: Scenario 3: Packet Loss Ratio Comparison (RTS/CTS versus ROAM)

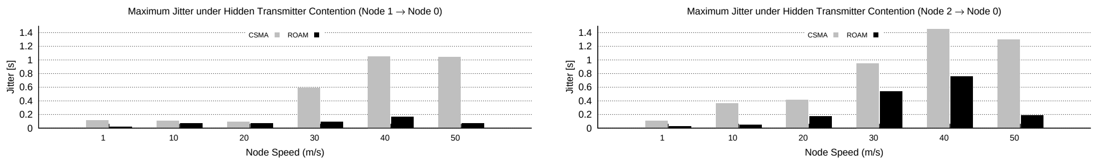


(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)



(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

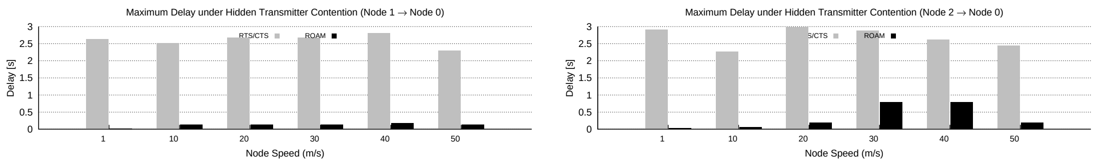
Figure 6.61: Scenario 3: Maximum Delay Comparison (CSMA versus ROAM)



(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)

(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

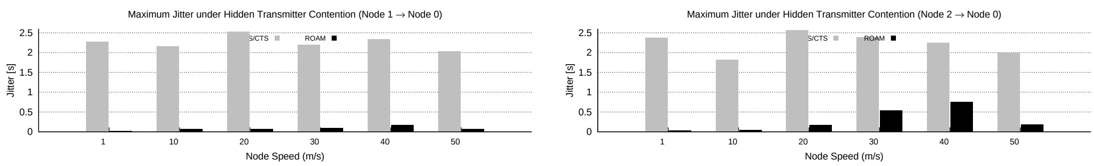
Figure 6.62: Scenario 3: Maximum Jitter Comparison (CSMA versus ROAM)



(a) Maximum E2E Delay during Contention ($N1 \rightarrow N0$)

(b) Maximum E2E Delay during Contention ($N2 \rightarrow N0$)

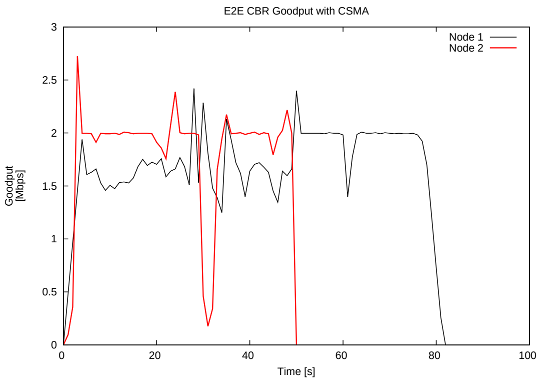
Figure 6.63: Scenario 3: Maximum Delay Comparison (RTS/CTS versus ROAM)



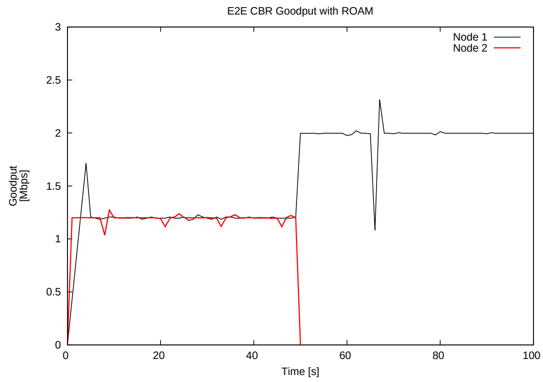
(a) Maximum E2E Jitter during Contention ($N1 \rightarrow N0$)

(b) Maximum E2E Jitter during Contention ($N2 \rightarrow N0$)

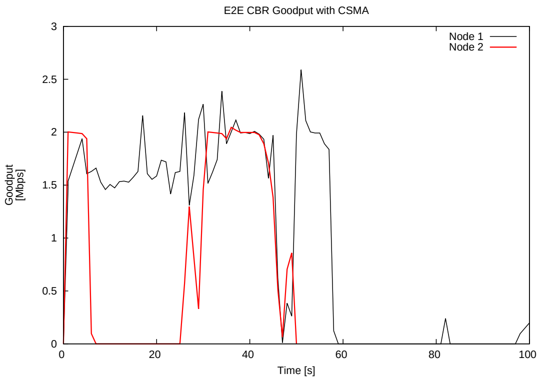
Figure 6.64: Scenario 3: Maximum Jitter Comparison (RTS/CTS versus ROAM)



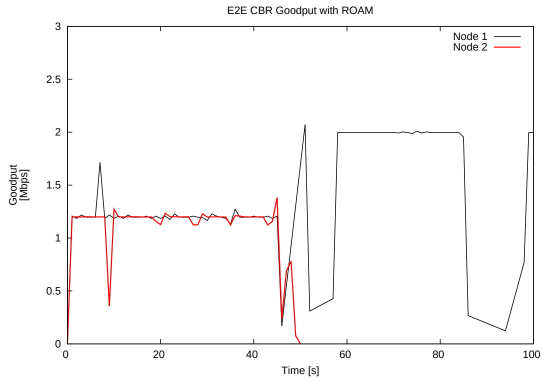
(a) CSMA (MN Speed = 20m/s)



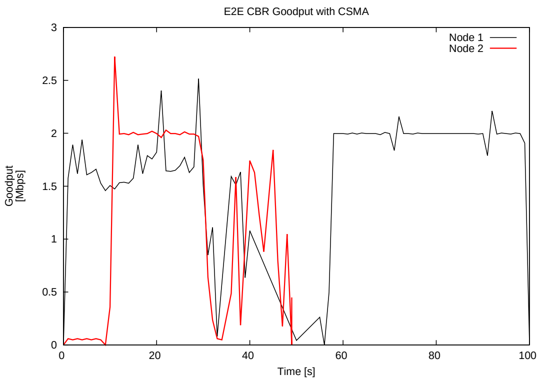
(b) ROAM (MN Speed = 20m/s)



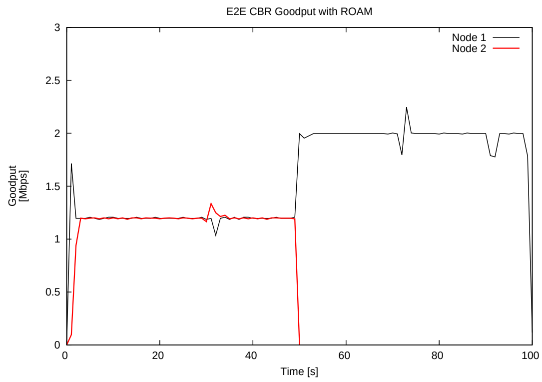
(c) CSMA (MN Speed = 30m/s)



(d) ROAM (MN Speed = 30m/s)



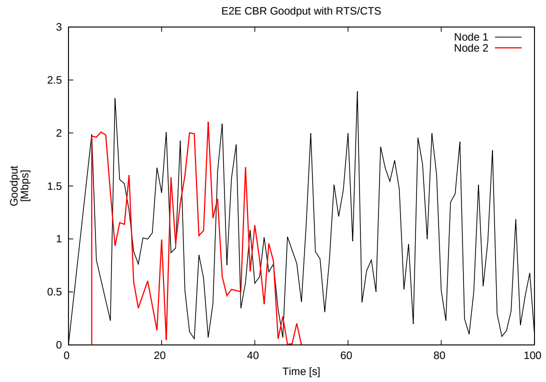
(e) CSMA (MN Speed = 50m/s)



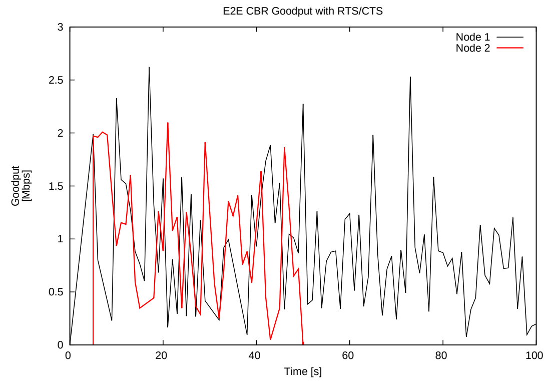
(f) ROAM (MN Speed = 50m/s)

Figure 6.65: Scenario 3: Instantaneous E2E CBR Goodput (CSMA versus ROAM)

traffic: RTS/CTS in response to a busy channel and ROAM, following indication of the presence of a hidden node. The flooding of RTS and CTS packets interferes not only with data transmissions but also routing control packets, slowing handoff between paths. Extension of the handoff period results in an increased backlog in

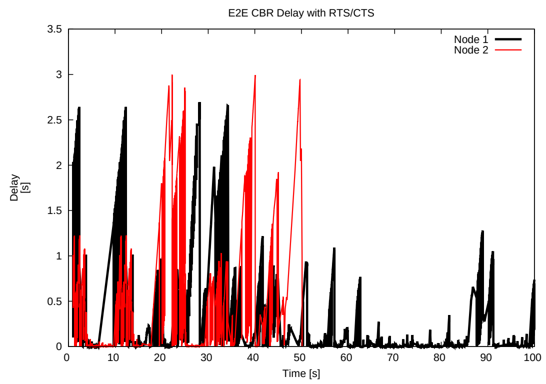


(a) RTS/CTS (MN Speed = 30m/s)

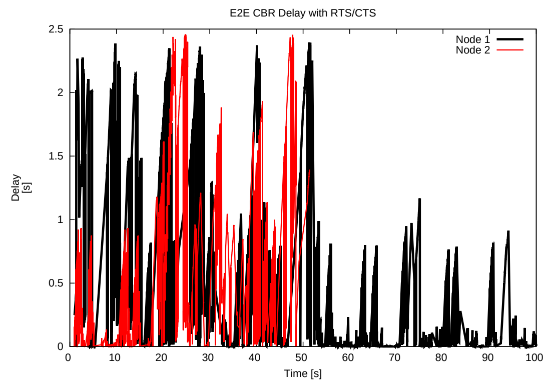


(b) RTS/CTS (MN Speed = 50m/s)

Figure 6.66: Scenario 3: Instantaneous E2E CBR Goodput (RTS/CTS)



(a) RTS/CTS (MN Speed = 30m/s)

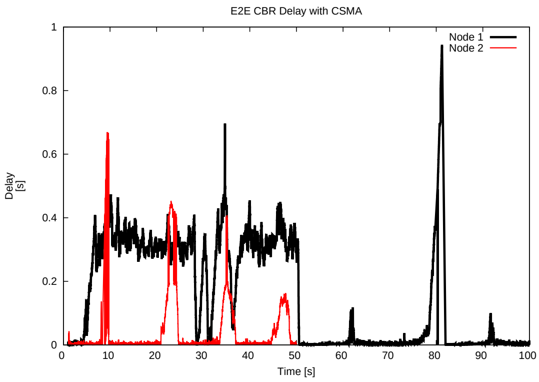


(b) RTS/CTS (MN Speed = 50m/s)

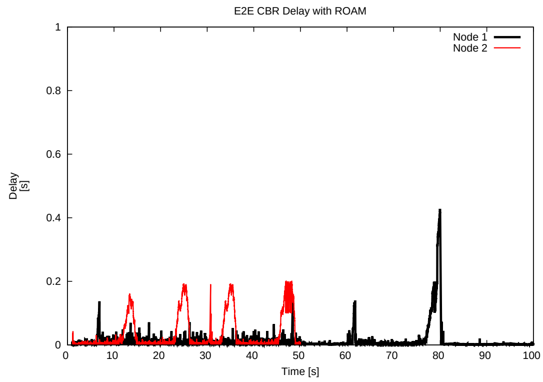
Figure 6.67: Scenario 3: Instantaneous E2E CBR Delay (RTS/CTS)

the buffer and eventual overflow. Without any exchange of control packets, ROAM performs well at high speeds by implementing a short-term reduction in load that reduces collisions, errors and buffering requirements. Figures 6.59–6.60 demonstrate that packet loss increases with node speed, when CSMA and RTS/CTS are implemented. ROAM, by ensuring a reduced incidence of routing errors enables timely maintenance of rapidly changing E2E paths, reducing overall loss of E2E connectivity and packet loss.

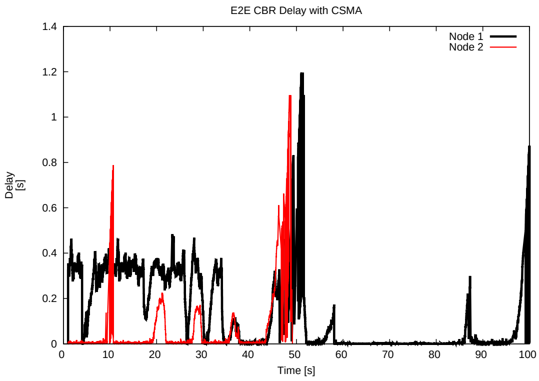
Figures 6.61–6.64 show that the overall reduction in queueing requirements and error recovery resulted in reduced maximum delay and jitter with ROAM. ROAM provides more successful packet delivery than CSMA or RTS/CTS for both transmitters, with similar overall reductions in all sub-scenarios. Figure 6.65 demonstrates that with CSMA, even when nodes moved at a speed of 20m/s following the period of contention, E2E connectivity was eventually lost for node 1.



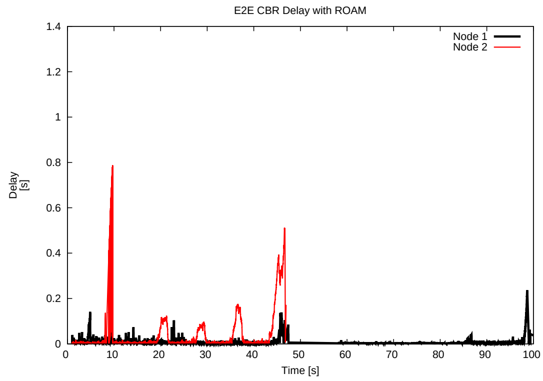
(a) CSMA (MN Speed = 20m/s)



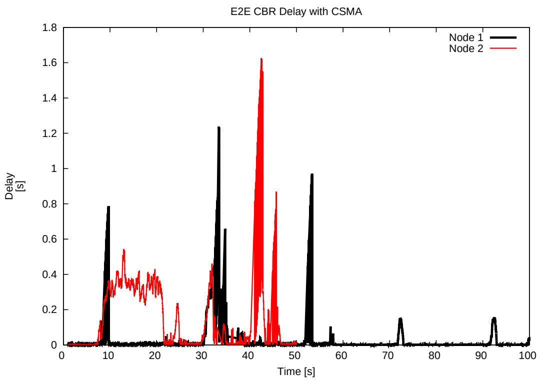
(b) ROAM (MN Speed = 20m/s)



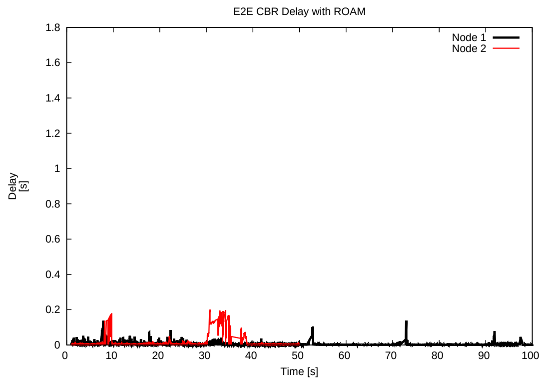
(c) CSMA (MN Speed = 30m/s)



(d) ROAM (MN Speed = 30m/s)



(e) CSMA (MN Speed = 50m/s)



(f) ROAM (MN Speed = 50m/s)

Figure 6.68: Scenario 3: Instantaneous E2E CBR Delay (CSMA versus ROAM)

Nodes 1 and 2 were both in transmission range of the receiver and hidden terminals to each other from 0-50s into the simulation. When links are continually overloaded, under CSMA and hidden node contention, the recurrent loss of routing packets, combined with rapid path change means that neighbouring nodes do not receive up to date routing information. As a result, when a handoff requires node 1 to use this neighbour as a forwarding node, the E2E path is lost and, under further rapid handoffs, is not regained. Persistent failure to attain an E2E path did not occur at a speed of 50m/s because node 1 rapidly moved back into range of previously utilised forwarding nodes.

With RTS/CTS, goodput of both CBR sources 1 and 2 periodically dropped to a negligible value, as a result of increased packet buffering and with eventual IFQ overflow. Correspondingly, goodput with CSMA and with ROAM did not drop as low as with RTS/CTS (figure 6.66). As node speed increased, the presence of a hidden node had a greater impact on peak delay. While ROAM reduces the rate at which packets are enqueued by the application, MAC layer error recovery due to previous collisions results in high delay. Additionally, each subsequent handoff compounds the time taken for the IFQ to drain, even after the hidden node has left the network. However, both peak and instantaneous delay were still lower with ROAM than CSMA or RTS/CTS (figure 6.67 and figure 6.68).

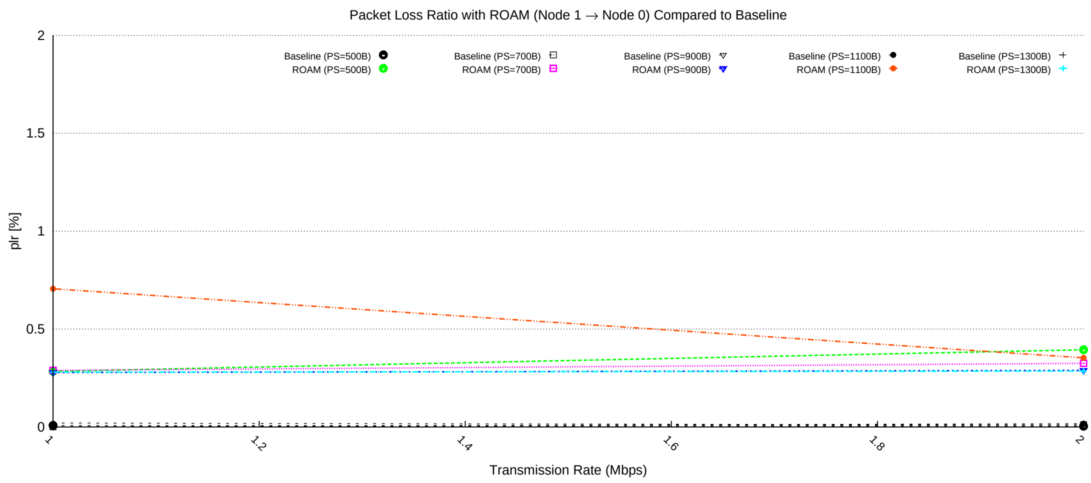
6.3 Validation of Results against Baseline Performance

This Section qualifies the scope of ROAM to constrain maximum delay and jitter. The best performance with the optimiser is, therefore, compared to results in a baseline scenario. This is to evaluate the ability of ROAM to reduce the disparity between performance in MANETs with complex network dynamics, and those with low levels of available resource variation. The previous sections have validated ROAM through comparison with worst-case network performance, when CSMA and RTS/CTS protocols operate without access to cross-layer information.

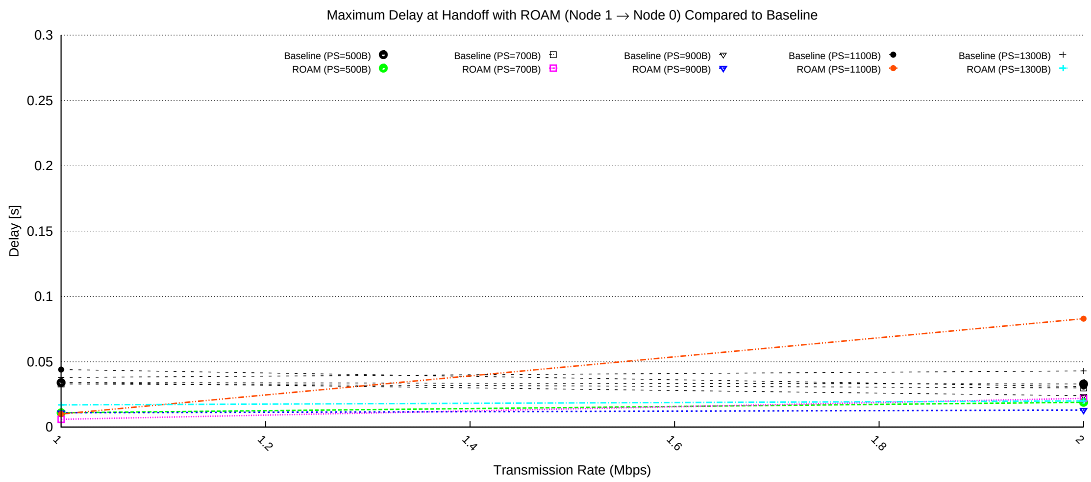
In Chapter 3, static transmitter-receiver ad hoc network simulations were investigated and results collated to form this baseline. A transmission distance of 120m was used in all simulations in this chapter, therefore, those showing the greatest ISRT performance improvements in comparison to the baseline results with an inter-nodal distance of 120m are considered here. This is in terms of lowest maximum delay, maximum jitter and packet loss ratio for each transmitter with ROAM.

In Section 6.2.1.1, the contention control optimiser was implemented in two mobile transmitters, on a distributed basis, with a range of CBR traffic configurations in a bus topology. These results are now, therefore, compared to results under the same configurations in the baseline scenario.

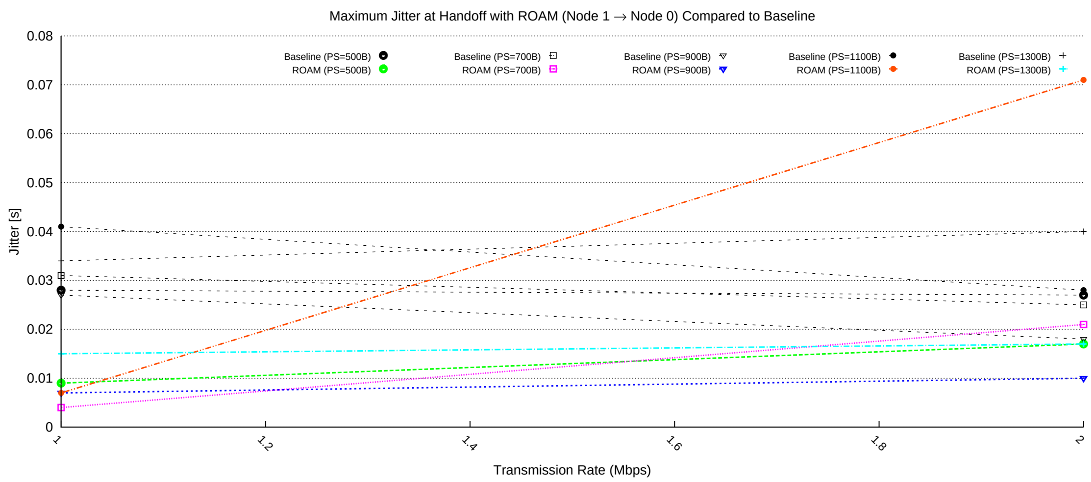
Figure 6.69(a) shows that ROAM was not capable of ensuring best-case packet



(a) Packet Loss Ratio

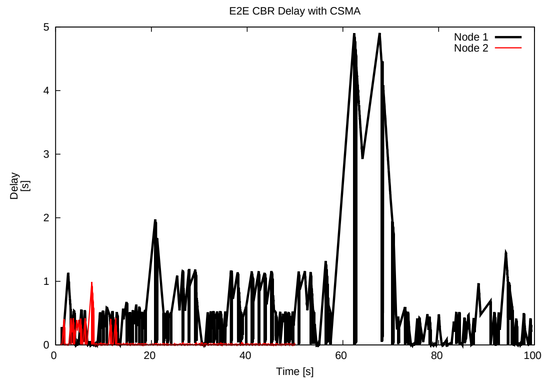


(b) Maximum Delay during Contention

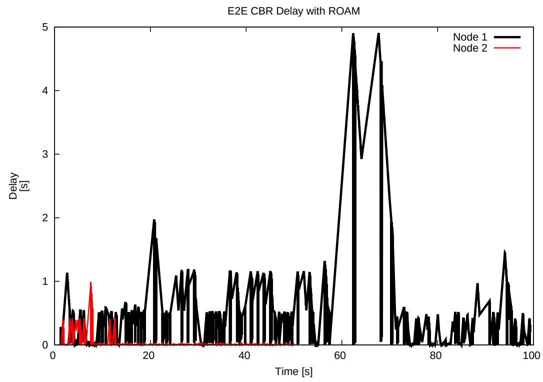


(c) Maximum Jitter during Contention

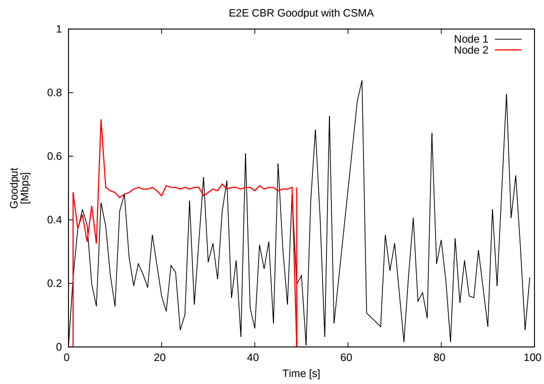
Figure 6.69: Performance of ROAM compared to Best-Case Ad Hoc Network Performance



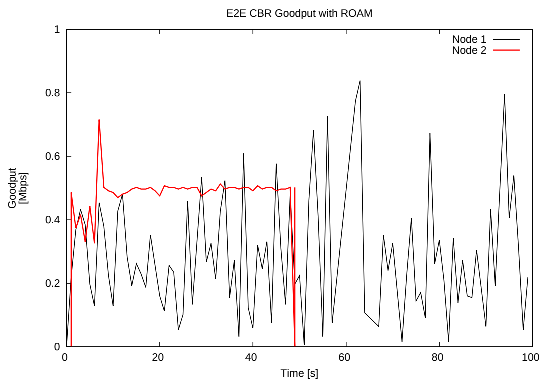
(a) CSMA E2E Delay (TR = 0.5Mbps, PS = 700B)



(b) ROAM E2E Delay (TR = 0.5Mbps, PS = 700B)



(c) CSMA Goodput (TR = 0.5Mbps, PS = 700B)



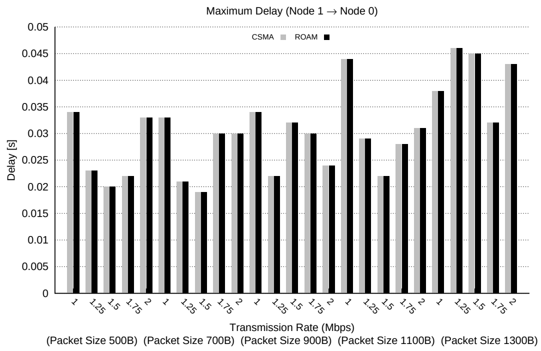
(d) ROAM Goodput (TR = 0.5Mbps, PS = 700B)

Figure 6.70: Exposed Transmitter Simulation: Instantaneous E2E Performance (CSMA versus ROAM)

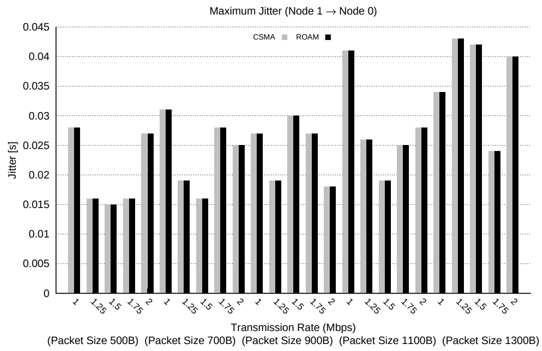
loss, as in the baseline scenario, and also provided a less consistent level of performance. The performance improvement with the middleware varies between nodes and under different configurations due to the time taken to identify the hidden transmitter and the multi-hop effects of lowering load under contention.

The lowest capabilities therefore resulted with a packet size of 700B transmitted at the lowest CBR rate. However, in previous scenarios, ROAM has also shown lowered capability with small packets transmitted at rates of 4-6Mbps. All ROAM packet loss ratio results were thus within the limits of best and worst performance. These results demonstrate that ROAM is capable of reducing the difference between performance in MANETs with complex network dynamics, and those with low levels of available resource variation.

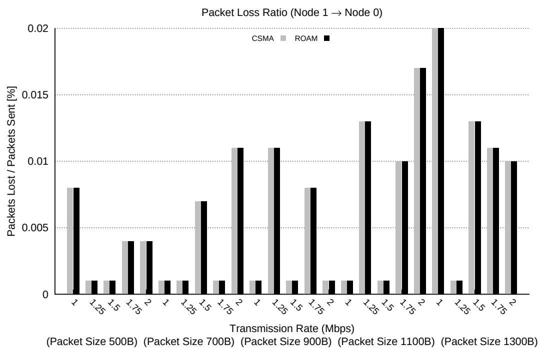
ROAM reduced maximum delay under hidden node contention to close to best-case performance, as shown in figure 6.69(b) when the packet size was smaller than



(a) Maximum Delay during Contention



(b) Maximum Jitter during Contention



(c) Packet Loss Ratio

Figure 6.71: Performance in Baseline Simulation Scenario (CSMA versus ROAM)

1100B. These results excluded the period following hidden node contention, which could become elevated under congestion in the multi-hop scenario that the optimiser was implemented in. Previous results have shown that ROAM performance deteriorates under congestion.

Notably, in many of these sub-scenarios (figure 6.69) ROAM is capable of bringing all three metrics to close to baseline limits, but not with consistency over all traffic configurations. Maximum delay and jitter were, under certain traffic settings, reduced to within 0.01s of the baseline, but were generally much higher than the baseline. The approach utilised by the optimiser can introduce jitter into a flow through the rapid change in application load. However, the results in Section 6.2.1.1 demonstrated that maximum jitter with ROAM was lower than with CSMA alone.

At the same time, only one CBR source was used in the baseline simulations and, as demonstrated in Subsection 5.2.2.1, increasing the number of CBR sources resulted in reduced performance for all CBR flows. The results are, therefore, promising in terms of ISRT performance and show the capability of cross-layer middleware in bounding both E2E delay and jitter and packet loss in comparison to widely implemented IEEE 802.11 mechanisms: CSMA and RTS/CTS.

It has been previously demonstrated that ROAM is capable of providing bounded delay and jitter in a range of transmission setting scenarios and MANET configurations. These results have demonstrated the best-cases of performance, where maximum delay, jitter and packet loss ratio were most significantly constrained compared to the baseline simulations. ROAM is able to approach baseline maximum delay and jitter through contention reduction. The results are promising in terms of ISRT performance and show the capability of ROAM in bounding both E2E delay and jitter and packet loss in comparison to both CSMA and RTS/CTS.

The optimiser evaluated in this chapter has been developed to respond to the presence of a hidden transmitter, reducing overall load in order to alleviate pressure on the shared forwarding node, avoiding unnecessary repeated collisions and retransmissions in order to reduce both contention and queueing delays. The optimiser has been developed to identify the presence of an exposed node, avoiding a load control response if one is detected. Figure 6.70 demonstrates that while ROAM monitors protocol layers for information on neighbouring transmitters, when a load control response is not implemented, the middleware and optimiser do not significantly influence overall network performance.

Correspondingly, without the presence of a hidden transmitter, ROAM provides no contention control response. This postulation has been demonstrated by implementing the middleware optimiser within the baseline scenario previously discussed. With only one transmitter and receiver present, the results with ROAM are not significantly different from those with CSMA. The ROAM contention control optimiser is a standalone cross-layer scheme that relies on intercepted layer parameters to detect the specific conditions of hidden transmitter contention and correspondingly limited resource availability. With only one transmitter on the network, ROAM in this node did not identify a hidden transmitter and there was a 0% or negligible change in RT performance, in terms of maximum delay, jitter or packet loss ratios (figure 6.71).

6.4 Summary and Discussion

This chapter has validated a contention control optimiser, implemented within a cross-layer middleware architecture. The optimiser can constrain E2E delay, jitter and packet loss associated with the presence of a hidden node through rapid detection of the presence of this node and a temporary tuned reduction of load. A hidden transmitter is detected based on current nodal performance and channel quality information. Additionally, the approach exploits a characteristic of wireless networks, wherein the MAC layer commonly intercepts and drops low power packets received from neighbouring nodes. On the basis of packets intercepted from the current forwarding node in use, the ROAM node is able to identify that this node is repeatedly busy. The aim of the ROAM contention control optimiser is to constrain E2E delay for ISRT traffic.

MANET MAC layer protocols utilise CSMA and RTS/CTS for media access and hidden node detection, respectively. CSMA does not respond to neighbouring transmissions, this results in increasing collisions and retransmissions in the presence of hidden terminals. RTS/CTS utilises repetitive control packet handshaking to assure a free channel prior to media access at the expense of inserting artificial delay into streams as packets are enqueued awaiting completion of the RTS-CTS exchange. This can result in applications exceeding buffer provisioning and the novel control packet flood interferes with ad hoc routing protocol functioning which also relies on control packet flooding for path maintenance. Both of these approaches incite QoS violations for ISRT flows (Chapter 3). The simulation scenarios used to test ROAM, CSMA and RTS/CTS performance in this chapter implemented widely varied traffic, mobility and topology configurations, creating dynamic layer-1 and 2 conditions. With CSMA and RTS/CTS, packet loss ratios and delay were elevated in comparison to the baseline, best-case performance.

Optimal available bandwidth use is essential in ad hoc networks with multiple nodes in transmission range sharing channels and interference between neighbouring transmissions. While MAC layer retransmissions are useful in ensuring E2E packet delivery, the associated random and deterministic backoff components are detrimental to delay-sensitive traffic. Additionally, ad hoc routing is dependant on timely control packet receipt, particularly when nodes are highly mobile and paths change rapidly. ROAM contention control relies solely on locally intercepted information gathered from the internal protocol stack, but is implemented on a distributed basis in all transmitters. As such, the overheads associated with global signalling and packet exchange, such as with RTS/CTS is avoided.

The ns2-MIRACLE simulation results show that ROAM is able to adapt to dynamic changes in resource availability, to bound E2E delay, jitter and loss of wireless nodes under hidden node contention. The peak delay and jitter were then shown to occurr when transmission rate and packet size were returned to their original settings by ROAM.

In a MANET these dynamics are primarily caused both by application transmission settings and MANET configurations. Under all of these circumstances the optimiser is able to bound maximum E2E delay, packet loss and maximum jitter to all of the transmitters on the network in comparison to RTS/CTS. Overall packet loss ratios were bounded when compared in all implementations to CSMA, however in certain configurations CSMA is able to provide lower maximum delays. Only when the network becomes extremely congested is the performance of the optimiser impeded and overall network performance was then comparable to RTS/CTS.

In all other scenarios, maximum E2E delay and packet loss ratio are also constrained and network guarantees, in the presence of a hidden node, can be provided. The low performance under congestion is justifiable when compared to the critical impact of congestion on the instantaneous performance of CSMA and RTS/CTS. The aim of this thesis has been to bound network delay, jitter and packet loss, therefore comparison between average metrics has not been investigated. However, the results in this chapter have demonstrated that ROAM is capable of reducing both maximum and instantaneous delay and avoiding extended periods of E2E path loss. The results have shown that ROAM reduces the impact of complex network dynamics by ensuring a level of E2E performance closer to that in a simple two-node network. However, ROAM can be improved through the addition of admission control, queue management or congestion notification to ensure a performance improvement under congestion and provide congestion avoidance capabilities. These areas, to be considered in future work for the contention control optimiser, are discussed in the following chapter.

Chapter 7

Conclusions and Future Work

This thesis researches the use of cross-layer middleware to provided bounded guarantees of delay and packet loss to applications in Mobile Ad hoc Networks (MANET), without the need to modify the functionality of safety-tested protocols. This chapter presents conclusions on the completed work (Section 7.1) and discussion of the potential for future extensibility (Section 7.2).

7.1 Evaluation

In military and disaster response situations, nodes that are passing through a region or scouting ahead may require the capacity to communicate with or transfer media to nodes that will stay in a region for a longer term. Additionally, communication will be required between devices that are not in line-of sight contact. MANETs therefore provide a suitable multi-hop solution. However, timeliness of packet delivery and packet loss can influence both the usefulness of data and safety of a system. When protocols, firmware or hardware are safety-certified, a requirement emerges that performance improvement should not be at the expense of modifications to their extensively tested functionality. Investigation of the research scenarios also considered the vision of the UK Ministry of Defence Network Enabled Capability (NEC) Project [114] for collaborative architectures, to provide timely intelligence sharing between networked mobile entities. The integration of these military systems requires the use of generic functionality that is extensible to all networks and the specified interactions of systems designed under the Integrated Modular Systems (IMS) software architecture [42]. Military onboard networks have traditionally streamlined Quality of Service (QoS) requirements by treating all transmitted data as having hard real-time (HRT) deadlines, due to the operational safety impact of data losses. However, in the wireless domain all of these applications will be supported as inelastic soft real-time (ISRT) and the provision of bounded delay enables specification of a Best-case Execution Time (BCET) and Worst Case Execution Time (WCET).

The major recent developments in this field include two theoretical MANET architectures: ECLAIR [124] and Performance-Oriented Model (POEM) [58] and a single kernel-implementation, Cross-layer Interface for wireless Ad hoc Networks (XIAN) [6] that have been proposed to introduce interface access to protocol parameters, without protocol modification. All of these proposals have focused on architecture design, not implementing or performance testing particular optimisers. Many existing optimisation approaches continually tune TCP congestion control responses to resource conditions or offer a joined-up solution, modifying multiple protocols in a concerted approach to QoS provisioning. The latter reduces the ability of the design to evolve alongside concurrent protocol improvements. Recent proposals for channel assignment and routing have benefited from global signalling and resource reservation. Throughput was increased by adjusting source rate to link capacity when combined with resource reservation [140]. While packet loss was reduced when dummy packets probed instantaneous signal strength on multiple links to inform induced handoff [151]. The hidden terminal problem has been variously prevented through widening carrier sense capabilities and extensive control packet handshaking such as the IEEE 802.11 implementation of virtual handshaking with RTS/CTS [48]

These alternatives relied on global MANET signalling, using control packets to gauge network conditions or to maintain bandwidth reservation, contributes to elevated queueing of data packets, collisions and interference. Flows are extensively queued, awaiting completion of handshaking or feedback processes. Rapidly changing topologies with link breakages and the underlying processes of the MAC layer such as repeated backoff can all lead to such adaptation taking an inaccurate view of network conditions. This is because dynamic variation in link quality reduces the validity of E2E feedback based approaches. A continual or periodic tuning of application or transport layer settings, or the requirement to wait for packet probing and handshaking introduce unnecessary artificial delays into flows. Bounded E2E delay and loss is not provided by these approaches.

In comparison to previous proposals for channel assignment and routing, discussed in Chapter 2, this thesis proposes utilisation of available control packet information from unmodified protocols. By responding to detection of an imminent, but temporary change in conditions, such as link quality deterioration or shared forwarding node contention, it becomes possible to induce a similarly short term optimisation, limiting the period of intervention with the protocol stack.

The scoping simulations in Chapter 3 demonstrated that significant increases in E2E delay and loss resulted from suboptimal link selection by an ad hoc routing protocol and when two transmitters competed for the same intermediate node (IN). Oblivious to the cause of packet loss, the wireless MAC layer attempts to elevate delivery by retransmitting data at randomly determined intervals. On a fading link this increases interference, as well as queueing and contention delays. In the hidden terminal scenario this has been shown to increase errors and collisions. Reducing the frequency of requirement for this MAC layer approach, which can increase contention and queueing delays, enables the provision of maximum delay, loss and jitter guarantees. This is by providing an early, informed response to handoff and hidden node contention. The architecture was therefore designed in light of the performance outcomes and network analysis of the scoping simulations.

On the basis of an extensive review of existing approaches, this thesis therefore proposes a novel cross-layer middleware architecture (ROAM), suitable to support safety-critical protocols and two middleware-implemented optimisers to constrain E2E delay and loss associated with handoff and hidden node contention. ROAM is an intralayer entity that uses generic API, associated with participating protocol layers, to monitor parameters held in protocol data structures relating to the exchange of data and control packets and return tuned parameters without changing the functionality of these structures. Independence from the protocol stack enables differing protocols to be plugged into the middleware, which may be popular and established wireless protocols or specific disaster response or military protocols. This also allows for evolution in protocol design. Current middleware approaches have proposed the use of protocol specific API, as well as protocol generic API, when the impact of increased processing of cross-layer signals on E2E delay was not under consideration. ROAM is specifically developed for the support of real-time (RT) applications in MANETs. As such, the middleware cannot be implemented alongside protocols that directly induce RT applications to violate their QoS requirements; such as TCP at the transport layer, a proactive routing protocol or one that utilises bandwidth reservation.

Two original cross-layer optimisers are proposed in this thesis, to manage horizontal handoff and hidden terminal contention control. Horizontal handoff without reference to channel conditions can result in localised increases in packet loss, jitter and delay as packets are repeatedly transmitted over fading links. The ROAM horizontal handoff optimiser collates information intercepted on neighbouring nodes to provide an early identification of link fading and institute rapid, controlled handoff. The need for complex parameter computation or exchanges is eliminated as the optimiser uses a relative comparison between optimal and suboptimal paths. Executing concurrently with the stack, the optimiser does not pre-empt routing, selecting the optimal next hop only when this node appears within the routing table. The optimiser does not modify the MAC layer, but transparently monitors all packets passing this layer to ensure that optimal links are successfully selected. This entails preventing ad hoc routing protocols from switching repeatedly between links and utilising non-robust links to highly mobile nodes.

When hidden terminals contend for the same channel, each node will overhear the ACKs sent by the mutual forwarding node. A common ACK rate, combined with deteriorating performance at the MAC layer (increasing retransmissions, queue length and path delay; in spite of high link received signal strength) can be exploited by the middleware that also accesses information at the routing and application layers. Rather than providing the continually changing response of approaches such as TCP, the ROAM optimiser incites a short term optimisation of application settings in order to reduce pressure on the queue and link, and accordingly bound queueing and contention delays. The optimisation relies on minimal control packet exchange, does not require interaction of any of the protocols and requires no MAC or network layer cooperation.

The two ROAM optimisers have been simulated with ns2-MIRACLE, an addon to the popular ns-2 simulator. ns2-MIRACLE provides more realistic propagation and interference computation and MAC and physical layer implementations than ns-2. The ns2-MIRACLE simulator computes these conditions based on interference and packet error models, whereas the widely used ns-2 simulator computes these values based solely on inter-nodal distance. Additionally, cross-layer middleware can be fully implemented in the simulator, where in-band piggybacking would have to be substituted in ns-2.

These optimisers have been validated to demonstrate their usefulness in MANETs and ability to provide improved network performance to ISRT applications that require bounded delay and loss. Chapter 3 showed that factors such as traffic configuration (homogeneous and heterogeneous traffic rate and packet size) and network configuration (node speed, topology, inter-nodal distance, neighbour contention and interference) influence network dynamics. In a MANET these factors create network-wide fluctuations in performance. Therefore, the ROAM optimisers have been rigorously tested under a range of CBR and VBR traffic settings, as well as mobility and topology settings. In differing topologies link length also changes, however ROAM relies on comparison of suboptimal and optimal parameter values, enabling a consistent performance improvement. Performance analysis has demonstrated that E2E delay, jitter and packet loss can be bounded with ROAM, also improving the performance of some non-ROAM nodes sharing the network. Under middleware optimisation, the lowest system performance levels are then solely associated with initial path setup costs. This enables the provision of guarantees to timing-sensitive applications that maximum delay occurs only during initial path setup.

The performance with ROAM has been compared to that of widely implemented and successfully established wireless ad hoc protocols. In comparison to horizontal handoff controlled by an ad hoc routing protocol, Ad hoc On Demand distance Vector (AODV) and hidden node contention response of CSMA and RTS/CTS, ROAM provides improved E2E performance with minimal overheads.

The middleware architecture has been developed to manage optimisation of many MANET protocols, however, assumptions made in the design of the optimisers place certain limits on their operation. Wireless networks predominantly rely on IP addressing, while the ability to maintain E2E connections is dependant on control packet exchange at the network and MAC layers. Correspondingly, ROAM is designed for use solely in MANETs and assumes that the aforementioned protocol functions and associated data structures are present. The performance improvements have been shown to be independent of network topology and size, as well as number of competing flows. However, while the contention control optimiser is able to reduce localised congestion, a performance benefit can only be assured to nodes closest to the receiver under the elevated congestion resulting from multiple hidden node pairs.

MANET performance improvement is a growing research field as, currently, widespread reliance on cellular infrastructure is resulting in oversubscription of a limited spectral range. MANETs offer the capability to support medium range communications between vehicles without reliance on fixed base stations or static predefined routing. Although many optimisation approaches have been proposed, few aim to be implemented in middleware or without protocol modifications, or have been experimentally validated. A limited number of cross-layer middleware architectures have also been proposed. Cross-layer middleware supports the evolution of new optimisers alongside the maturation of established protocols, allowing concurrent and mutually beneficial development in both fields. ROAM has been designed in view of the stringent modification requirements in military networks, which enables the middleware to plug into a network stack using any MANET protocol, as long as a predominantly reactive, control packet based approach to routing is implemented. However, ROAM has been designed to function alongside the IEEE 802.11 suite of MAC protocols and there is potential for novel MAC layer approaches to be developed and become established in the military and commercial domains.

The limitations of validation of ROAM are dependant on the protocol modules currently provided by ns2-MIRACLE and further validation would entail full development from scratch of ad hoc routing or VoIP modules for the ns2-MIRACLE simulator. ROAM has been tested alongside AODV, based on the assumption that other ad hoc routing protocols that are suitable to use with ISRT applications will rely on reactive routing, due to the high overheads of proactive network-wide maintenance of routing tables. The contention control optimiser has been tested with CBR but not VoIP, or another VBR application layer, as the VoIP implementation in ns2-MIRACLE is not an independent module that can be accessed by middleware. CBR QoS requirements are the most stringent amongst ISRT applications and provide a representative benchmark for other ISRT scenarios. It is therefore assumed that validation with CBR will entail a functional degree of support with VBR traffic patterns, as has been shown by comparison with the horizontal handoff optimiser.

7.2 Future Work

The following section presents some recommendations for future evolution of ROAM. This considers the potential for ROAM to provide an extended contribution to performance improvement for time-critical applications through interaction with other QoS control approaches, previously discussed in Chapter 2. The ROAM optimisers do not provide perpetual delay control or traffic conditioning. Therefore, if a circumstantial requirement to provide optimal horizontal handoff or contention control does not appear, ROAM is incapable of managing the provision of bounded delay, jitter or packet loss ratio. For example, in the former, if a more robust channel is not available or, in the latter, if an exposed node contends for the channel.

The handoff optimiser monitors link quality to manage handoff when a link in use begins to fade and a more preferable link becomes available. While the contention control optimiser monitors for reduced performance, in the absence of a fading link, and the presence of a hidden terminal: corresponding to a high ACK rate at the shared forwarding node. The ROAM optimisers alone are not capable of guaranteeing deterministic delay or preventing network congestion. However, ROAM is capable of providing information to the following QoS control measures to improve responsiveness to these ends:

- ROAM handoff and contention control optimisers could be applied alongside Active Queue Management (AQM) approaches, including traffic policing and admission control for service differentiation in heterogeneous traffic and buffer management with homogeneous traffic. The performance improvements of such an approach for ISRT and non-real-time (NRT) traffic would then require analysis. AQM is predominantly used for NRT traffic, due to the reliance on TCP rate control. However, the non-deterministic nature of wireless links, due to interference and multihop path selection, reduces the efficiency of these schemes. The ROAM contention control optimiser uses MAC queue length increases to determine a low level of performance as part of the hidden node detection. However, it is also possible to utilise both optimisers in order to prevent congestion.

- If multiple flows are to be transmitted, with differing performance requirements, such as mixed ISRT and NRT traffic, these may be attributed different priorities. Therefore, when a fading link is in use, prior to horizontal handoff, low-priority packet dropping can be combined with admission control to control buffer occupancy and prevent unnecessary traffic on the suboptimal link, on the basis that these packets will be successfully retransmitted when an optimal link is selected. The same approach could be implemented when a hidden terminal competes for the forwarding node, however, as the time for which a hidden terminal is present is outside of network control, such traffic conditioning over a long period would significantly increase packet loss for NRT flows.

- In the case of military traffic (where all flows are treated as ISRT and safety-critical), when imminent handoff is detected, packets with low retransmission counts could be dropped from the queue prior to fast handoff. This is under the assumption that they are more likely to be successfully transmitted on the optimal channel. Admission control can then prevent entry of new flows until handoff is completed.

- A network-wide approach to cross-layer signalling entails the introduction of novel signalling packets and mechanisms or the modification of existing protocols in order to allow global network information to filter with ordinary control packet flow. The results in Chapter 6 demonstrated that ROAM contention control would benefit from a response to congestion in the presence of multiple hidden node clusters. Future work could explore the feasibility of implementing a method of notification: passing a cumulative count of hidden transmitters, when these are detected by a ROAM node, to neighbouring ROAM nodes. If the number of hops to the receiver could also be transmitted, the load reduction managed by ROAM could be tuned according to distance of the hidden terminals from the receiver, to ensure fairness and a performance benefit to multiple transmitters across the network. A proliferation of control packet exchanges can increase the severity of, or create network congestion, therefore such a study would need to evaluate the benefit to ISRT applications and the impact on performance in an uncongested network. Modification of an existing packet type, for signalling, such as use of the IP header option, would support the extensibility of this solution. However, this would prevent implementation alongside contemporary protocols that already make use of this option, for example NRT nodes utilising ECN.

The aforementioned approaches demonstrate the extensibility of ROAM, but would entail a detour from the strategies previously implemented in this thesis, allowing ROAM to implement multiple tuned behaviours or using a global signalling rather than distributed solution in response to degrading network conditions. Evaluation of performance of these four potential applications for ROAM would require the development of multiple protocol modules for the ns2-MIRACLE simulator, in order to implement admission control, traffic conditioning at the transport layer or network-wide signalling.

ns2-MIRACLE has been developed to simulate varied and complex environmental interference, fading and attenuation conditions. However, potential experiments include validating the ROAM architecture in a testbed to demonstrate comparable performance under non-deterministic environmental interference conditions. This would require deployment of a wide-scale MANET, to represent the conditions of a military or disaster response network. The deployment of large mobile networks with non-linear vehicular motion would provide a better understanding of the dynamics of MANETs, particularly where the distance between nodes resembles avionic formations.

A full real-world implementation of ROAM would entail modification of open-source kernel components, to create the middleware and API and associate the latter with MANET protocol implementations. The developers of ECLAIR have partially implemented a version of the middleware (but not the proposed API) in the Linux kernel, as a proof of concept demonstration that middleware can be developed and used to adapt TCP. Development and evaluation of a full kernel implementation of cross-layer middleware in a MANET of realistic scale is beyond the scope of this thesis.

7.3 Final Conclusions

The major contributions of this thesis are to demonstrate that cross-layer middleware can be used to bound E2E delay, jitter and packet loss, without the need to alter the existing functionality of popular and established or extensively safety-tested protocols and systems.

The results indicate the value of implementing a cross-layer optimiser that transparently monitors protocol performance and the transmission and receipt of control packets in order to gauge local network performance. The ROAM optimisers assess reduced MAC and network layer performance and utilise information gathered from control packet receipt to then establish how protocol parameters should be adapted to ensure optimal handoff or reduce contention for a shared channel. Both of these approaches demonstrate that performance optimisation can be provided without recourse to protocol modification, complex protocol interactions or the use of congestion-inducing signalling. QoS awareness can therefore be added to devices as desired, without the need to encourage widespread uptake of a novel protocol or architecture.

References

[1] B. Abbasov. AHRED: A Robust AQM Algorithm for Wireless Ad hoc Networks. In *International Conference on Application of Information and Communication Technologies, AICT 2009*, October 2009.

[2] Actel Corporation. Actel: Developing AFDX Solutions. Technical report, 2005.

[3] S. Adlakha, X. Zhu, B. Girod, and A. J. Goldsmith. Joint Capacity, Flow and Rate Allocation for Multiuser Video Streaming Over Wireless Ad-Hoc Networks. In *IEEE International Conference on Communications (ICC)*, pages 1747–1753, 2007.

[4] A. E. A. Ahmed, T. Saadawi, and M. Lee. A Cross-Layer Optimized Error Recovery Mechanism for Real-Time Video in Ad-Hoc Networks. In *ICPADS '06: Proceedings of the 12th International Conference on Parallel and Distributed Systems*, pages 77–82, Washington, DC, USA, 2006. IEEE Computer Society.

[5] G.-S. Ahn, A. T. Campbell, A. Veres, and L.-H. Sun. SWAN: Service Differentiation in Stateless Wireless Ad hoc Networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 457–466 o.2, 2002.

[6] H. Aiache, V. Conan, L. Lebrun, J. Leguay, S. Rousseau, and D. Thoumin. XIAN Automated Management and Nano-Protocol to Design Cross-Layer Metrics for Ad Hoc Networking. In Amitabha Das, Hung Keng Pung, Francis Bu Sung Lee, and Lawrence Wai Choong Wong, editors, *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, volume 4982 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2008.

[7] H. Aiache, V. Conan, J. Leguay, and M. Levy. Xian: Cross-layer Interface for Wireless Ad hoc Networks. In *Mediterranean Ad Hoc Networking Workshop, 2006. Med-Hoc-Net 2006. Proceedings of the*, June 2006.

[8] R. L. Alena, J. P. Ossenfort, K. I. Laws, A. Goforth, and F. Figueroa. Communications for Integrated Modular Avionics. In *Aerospace Conference, 2007 IEEE*, pages 1–18, March 2007.

[9] O. A. Alim and K. F. Mohamed. Using Traffic Regulation to Meet End-to-end Deadline in Hard Real time ATM Networks, 2003.

[10] L. Alonso and A. Ramon. Optimization of Wireless Communication Systems using Cross-layer Information. *Signal Process.*, 86(8):1755–1772, 2006.

[11] K. J. An and H. Song. An Effective Cross-layer Packet Scheduling and Routing Algorithm for Delay-sensitive Media Transmission over MANET. In *Proceedings of the 2009 IEEE international conference on Communications (ICC)*, pages 124–127, 2009.

[12] M. Anand, S. Vestal, S. Dajani-Brown, and I. Lee. Formal Modeling and Analysis of the AFDX Frame Management Design. In *Object and Component-Oriented Real-Time Distributed Computing, 2006. ISORC 2006. Ninth IEEE International Symposium on*, page 7 pp., April 2006.

[13] A. Bag and M. A. Bassiouni. A Tunable Cross-Layer Congestion Reducing Medium Access Control (CRMAC) Protocol for Wireless Networks. In *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–7, Orlando, FL, USA, 2007.

[14] N. Baldo, F. Maguolo, and M. Miozzo. A New Approach to Simulating PHY, MAC and Routing. In *ValueTools '08: Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, pages 1–9, Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering; ICST, Brussels, Belgium, Belgium, 2008. ICST.

[15] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, and M. Zorzi. ns2-MIRACLE: a Modular Framework for Multi-technology and Cross-layer Support in Network Simulator 2. In *ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, pages 1–8, ICST, Brussels, Belgium, Belgium; Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2007. ICST.

[16] U. Baroudi and B. A. Qadous. An efficient adaptive Cross-Layer interaction mechanism for TCP traffic over heterogeneous networks, 2008.

[17] C. Barrett, A. Marathe, M. Drozda, and M. Marathe. Characterizing the Interaction between Routing and MAC Protocols in Ad-hoc Networks. In

MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing, pages 92–103, New York, NY, USA, 2002. ACM.

[18] P. Becker. QoS Routing Protocols for Mobile Ad-hoc Networks - a Survey. Technical report, University of Kaiserslautern, August 2007.

[19] P. Bellavista, A. Corradi, and C. Giannelli. A Layered Infrastructure for Mobility-aware Best Connectivity in the Heterogeneous Wireless Internet. In *MOBILWARE '08: Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, pages 1–8. ICST, 2007.

[20] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a Media Access Protocol for Wireless LAN's. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 212–225, New York, NY, USA, 1994. ACM.

[21] N Bisnik. Protocol Design for Wireless Ad hoc Networks: The Cross-Layer Paradigm. Technical report, ECSE Department, 2005.

[22] A. Boukerche, B. Turgut, N. Aydin, M.Z. Ahmad, L. Bölöni, and D. Turgut. Routing Protocols in Ad hoc Networks: a Survey. *Computer Networks*, 55(13):3032–3080, September 2011.

[23] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. IETF RFC 2205: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification, 1997.

[24] C. T. Calafate. Supporting Soft Real-Time Services in MANETs Using Distributed Admission Control and IEEE 802.11e Technology. In *ISCC '05: Proceedings of the 10th IEEE Symposium on Computers and Communications*, pages 217–222, Washington, DC, USA, 2005. IEEE Computer Society.

[25] G. Carneiro, J. Ruela, and M. Ricardo. Cross-layer Design in 4G Wireless Terminals. *Wireless Communications, IEEE*, 11; 11(2):7–13, 2004.

[26] I. D. Chakeres and E. M. Belding-Royer. AODV Routing Protocol Implementation Design, 2004.

[27] I. D. Chakeres, E. M. Belding-Royer, and J. P. Macker. Perceptive Admission Control for Wireless Network Quality of Service. *Ad Hoc Netw.*, 5(7):1129–1148, 2007.

[28] S. Chakrabarti and A. Mishra. QoS Issues in Ad hoc Wireless Networks. *Communications Magazine, IEEE*, 39(2):142–148, 2001.

[29] Y. S. Chan and J. W. Modestino. A Joint Source Coding-power Control Approach Combined with Adaptive Channel Coding for Video Transmission over CDMA Cellular Networks. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 5, pages 3415–3419 o.5, October 2003.

[30] Z. Chang, G. Gaydadjiev, and S. Vassiliadis. Infrastructure for Cross-Layer Designs Interaction. In *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pages 19–25, August 2007.

[31] H. Charara, J.-L. Scharbarg, J. Ermont, and C. Fraboul. Methods for Bounding End-to-end Delays on an AFDX Network. In *Real-Time Systems, 2006. 18th Euromicro Conference on*, pages 10 pp.–202, 2006.

[32] G. Chen, M. Song, Y. Zhang, and J. Song. Cross-Layer Adaptation with Coordinated Scheduling for Heterogeneous Wireless Networks. In *VTC Fall*, pages 1–5, 2010.

[33] K. Chen, S. H. Shah, and K. Nahrstedt. Cross-Layer Design for Data Accessibility in Mobile Ad Hoc Networks. *Wireless Personal Communications*, 21(2):49–76, 2002.

[34] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle. Cross-Layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–13, 2006.

[35] S. Chen and K. Nahrstedt. Distributed Quality-of-Service Routing in Ad hoc Networks. *Selected Areas in Communications, IEEE Journal on*, 17(8):1488–1505, August 1999.

[36] S. Chen, K. Nahrstedt, and Y. Shavitt. A QoS-aware Multicast Routing Protocol. *Selected Areas in Communications, IEEE Journal on*, 18(12):2580–2592, December 2000.

[37] P. Cheng, Z. Zhang, H.-H. Chen, and P. Qiu. A Framework of Cross-layer Design for Multiple Video Streams in Wireless Mesh Networks. *Computer Communications*, 31(8):1529–1539, 2008.

[38] H.-Y. Choi and S.-G. Min. RED-RT for Congestion Control and Real Time Traffic Protection. In *CIT '06: Proceedings of the Sixth IEEE International Conference on Computer and Information Technology*, page 115, Washington, DC, USA, 2006. IEEE Computer Society.

[39] J. R. Cleveland. Simulation of Digital Message Transfer with MIL-STD Protocols across HF Radio Networks. pages 885–889 o.3, October 1994.

[40] M. Conti, G. Maselli, and G. Turi. Design of a Flexible Cross-Layer Interface for Ad Hoc Networks. In *Challenges in Ad Hoc Networking*, volume 197 of *IFIP International Federation for Information Processing*, pages 189–198. Springer Boston, 2006.

[41] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick. IETF RFC 2386: A Framework for QoS-based Routing in the Internet. RFC 2386 (Informational), August 1998.

[42] Publication Date. Ministry of Defence Defence Standard 00-74 ASAAC Standards Part 1 : Standards for Software. *Library*, (2), 2008.

[43] K. M. E. Defrawy, M. S. E. Zarki, and M. M. Khairy. Proposal for a Cross-layer Coordination Framework for Next Generation Wireless Systems. In *IWCMC '06: Proceedings of the 2006 international conference on Wireless communications and mobile computing*, pages 141–146, New York, NY, USA, 2006. ACM.

[44] P. Dely. ns2voip port for nsmiracle. http://www.cs.kau.se/ pdely/index.php?n=Main.Ns2voipPortForNsmiracle.

[45] B. P. Deosarkar, N. S. Yadav, and R. P. Yadav. Clusterhead Selection in Clustering Algorithms for Wireless Sensor Networks: A Survey, 2008.

[46] A. Dhatrak and A. Deshmukh. Modified AODV Protocols: A Survey. *International Journal of Computer Applications*, pages 25–28, 2011.

[47] H. Dhillon and H. Q. Ngo. CQMP: a Mesh-based Multicast Routing Protocol with Consolidated Query Packets. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 4, pages 2168–2174 o. 4, March 2005.

[48] M. A. Ergin, M. Gruteser, L. Luo, D. Raychaudhuri, and H. Liu. Available Bandwidth Estimation and Admission Control for QoS Routing in Wireless Mesh Networks. *Computer Communications*, 31(7):1301, 2008.

[49] Y. Faizulkhakov. Time Synchronization Methods for Wireless Sensor Networks: A Survey. *Programming and Computer Software*, 33(4):214–226, July 2007.

[50] H. Fattah and C. Leung. An Overview of Scheduling Algorithms in Wireless Multimedia Networks. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 9(5):76–83, 2002.

[51] M. Ferreira and A. S.J. Helberg. Testbed Evaluation of a Delay Aware Routing Metric for Wireless Ad Hoc Networks. In *Southern Africa Telecommunication Networks and Applications Conference, SATNAC 2010*, 2010.

[52] F. Foukalas, V. Gazis, and N. Alonistioti. Cross-layer Design Proposals for Wireless Mobile Networks: a Survey and Taxonomy. *Communications Surveys and Tutorials, IEEE*, 10(1):70–85, 2008.

[53] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla. The Impact of Multihop Wireless Channel on TCP Performance. *IEEE Transactions on Mobile Computing*, 4(2):209–221, 2005.

[54] G. Gaertner and V. Cahill. Understanding Link Quality in 802.11 Mobile Ad Hoc Networks. *IEEE Internet Computing*, 8(1):55–60, January 2004.

[55] M. Gerla and J. T. Tsai. Multicluster, Mobile, Multimedia Radio Network. *Journal of Wireless Networks*, 1:255–265, 1995.

[56] A. J. Goldsmith and S. B. Wicker. Design Challenges for Energy-constrained Ad hoc Wireless Networks. *Wireless Communications, IEEE*, 9(4):8–27, 2002.

[57] S. Gowrishankar, S. K. Sarkar, and T. G. Basavaraju. Performance Analysis of AODV, AODVUU, AOMDV and RAODV over IEEE 802.15.4 in Wireless Sensor Networks, 2009.

[58] X. Gu, X. Fu, H. Tschofenig, and L. Wolf. Towards Self-optimizing Protocol Stack for Autonomic Communication: Initial Experience. In *Autonomic Communication*, volume 3854 of *Lecture Notes in Computer Science*, pages 186–201. Springer Berlin / Heidelberg, 2006.

[59] R. Guimaraes, L. Cerda, J. M. Barcelo, J. Garcia, M. Voorhaen, and C. Blondia. Quality of Service Through Bandwidth Reservation on Multirate Ad hoc Wireless Networks. *Ad Hoc Networks*, 7(2):388–400, 2009.

[60] R. Hamzaoui, V. Stankovic, and Z. Xiong. *Forward Error Control for Packet Loss and Corruption*, pages 271–291. Multimedia over IP and Wireless Networks. Academic Press, Burlington, 2007.

[61] L. Hogie, P. Bouvry, and F. Guinand. An Overview of MANETs Simulation. *Electronic Notes in Theoretical Computer Science*, 150(1):81–101, 2006.

[62] K. Hong, S. Sengupta, and R. Chandramouli. Cross-Layer MAC Enabling Virtual Link for Multi-Hop Routing in Wireless Ad Hoc Networks. *Communications Society*, 2010.

[63] Y.-C. Hu and A. Perrig. A Survey of Secure Wireless Ad hoc Routing. *Security and Privacy, IEEE*, 2(3):28–39, 2004.

[64] S. Hua and G. Qu. A New Quality of Service Metric for Hard/soft Real-time Applications, 2003.

[65] C. Huang, M. Loiacono, J. Rosca, and J.-N. Hwang. Distributed Cross Layer Congestion Control for Real-Time Video over WLAN. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 2270–2276, 2008.

[66] T. Itu. *The Emodel: A Computational Model for Use in Transmission Planning*. ITU-T Rec. G.107. ITU-T, July 1998.

[67] M. T. Ivrla, L. U. Choi, E. Steinbach, and J. A. Nossek. Models and Analysis of Streaming Video Transmission over Wireless Fading Channels. *Signal Processing: Image Communication*, 24(8):651–665, 2009.

[68] A. K. Jagadev, B. K. Pattanayak, M. K. Mishra, and M. Nayak. Power and Delay Aware On-Demand Routing For Ad Hoc Networks. *Indian Journal of Computer Science and Engineering (IJCSE)*, cs.NI(04):7, 2010.

[69] A. K. Jagadev, B. K. Pattanayak, M. K. Mishra, and M. R. Nayak. A QoS Solution to Hard Real Time Delay Satisfaction in Ad Hoc Networks. *IJCSNS International Journal of Computer Science and Network Security*, 9(12):238–248, 2009.

[70] E. D. Jensen. Eliminating the 'Hard'/'soft' Real-time Dichotomy. *Computing and Control Engineering Journal*, 8(1):15–19, 1997.

[71] A. Jeyaraj, L. Cheng, and M. E. Zarki. Proposal for a Cross layer Scheme for Real-time Wireless Video. *Journal of Zhejiang University - Science A*, 7(10):1690–1694, 2006.

[72] W. Jia, H. Wang, M. Tang, and W. Zhao. Effective Delay Control for High Rate Heterogeneous Real-time Flows. In *ICDCS '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*, page 367, Washington, DC, USA, 2003. IEEE Computer Society.

[73] H. Jiang, W. Zhuang, and X. Shen. Cross-layer design for Resource Allocation in 3G Wireless Networks and Beyond. *Communications Magazine, IEEE*, 43; 43(12):120–126, 2005.

[74] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, February 2007.

[75] W. Kasch, J. Ward, and J. Andrusenko. Wireless Network Modeling and Simulation Tools for Designers and Developers, 2009.

[76] V. Kawadia and P. R. Kumar. A Cautionary Perspective on Cross-layer Design. *Wireless Communications, IEEE*, 12(1):3–11, 2005.

[77] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer. Application-driven Cross-layer Optimization for Video Streaming over Wireless Networks. *Communications Magazine, IEEE*, 44(1):122–130, January 2006.

[78] S. A. Khayam, S. S. Karande, M. Krappel, and H. Radha. Cross-layer Protocol Design for Real-time Multimedia Applications over 802.11b Networks. In *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo*, pages 425–428, Washington, DC, USA, 2003. IEEE Computer Society.

[79] S. Khurana, A. Kahol, and A. P. Jayasumana. Effect of Hidden Terminals on the Performance of IEEE 802.11 MAC Protocol, 1998.

[80] B.-J. Kim. A Network Service Providing Wireless Channel Information for Adaptive Mobile Applications. I. Proposal, 2001.

[81] J. N. Kim, K. J. Kim, and K. J. Han. Supporting Differentiated Service in Mobile Ad hoc Networks through Congestion Control. In *Lecture Notes in Computer Science*, volume 3391, pages 643–651, 2005.

[82] D. Kliazovich, M. Devetsikiotis, and F. Granelli. Formal Methods in Cross Layer Modeling and Optimization of Wireless Networks: State of the Art and Future. In *Heterogeneous Next Generation Networking Innovations and Platforms*, chapter Chapter 1, pages 410–412. Information Science Reference, 2008.

[83] D. Kliazovich and F. Granelli. Introduction: Why Cross-Layer? Its Advantages and Disadvantages. In N Zorba, C Skianis, and C Verikoukis, editors, *Cross Layer Designs in Wlan Systems*, number v. 1 in Emerging Communication and Service Technologies Series. Troubadour Publishing Limited, 2011.

[84] Y.-B. Ko and N. H. Vaidya. Location-aided Routing (LAR) in Mobile Ad hoc Networks. *Proceedings of the 4th annual ACMIEEE international conference on Mobile computing and networking MobiCom 98*, 6(4):66–75, 1998.

[85] E. Kohler, M. Handley, and F. Floyd. IETF RFC 4340: Datagram Congestion Control Protocol (DCCP). (4340):1–9, 2006.

[86] J. A. Kroon, E. N. Koski, and G. R. Robertson. BOWMAN HF IP network solution. In *HF Radio Systems and Techniques, 2003. Ninth International Conference*, pages 25–30, June 2003.

[87] M. Kumar and R. Mishra. An Overview of MANET: History, Challenges and Applications. *Indian Journal of Computer Science and Engineering (IJCSE)*, 3(1), March 2012.

[88] S. Kumar, V. S. Raghavan, and J. Deng. Medium Access Control Protocols for Ad hoc Wireless Networks: A Survey. *Ad Hoc Networks*, 4(3):326–358, May 2006.

[89] J. F. Kurose and K. W. Ross. *Computer Networking - a Top-down Approach Featuring the Internet*. Addison-Wesley-Longman, 2001.

[90] T. Kwon, H. Lee, S. Choi, J. Kim, D.-H. Cho, S. Cho, S. Yun, W.-H. Park, and K. Kim. Design and Implementation of a Simulator Based on a Cross-layer Protocol between MAC and PHY Layers in a WiBro Compatible IEEE 802.16e OFDMA system. *Communications Magazine, IEEE*, 43(12):136–146, 2005.

[91] R. A. Lackman, J. D. Spragins, and D. Tipper. Scheduling Real-time and Non-real-time Traffic under Nonstationary Conditions. *Annals of Operations Research*, 36(1):193–224, December 1992.

[92] Y.-C. Lai. DCCP: Transport Protocol with Congestion Control and Unreliability. *Internet Computing, IEEE*, 12(5):78–83, September 2008.

[93] H.-J. Lee, H.-J. Byun, and J.-T. Lim. TCP-friendly Congestion Control for Streaming Real-time Applications over Wireless Networks. *Communications, IET*, 2(1):159–163, January 2008.

[94] M. Lee, M. Kang, M. Kim, and J. Mo. A Cross-layer Approach for TCP Optimization over Wireless and Mobile Networks. *Computer Communications*, 31(11):2669–2675, July 2008.

[95] S.-B. Lee, G.-S. Ahn, and A. T. Campbell. Improving UDP and TCP performance in Mobile Ad hoc networks with INSIGNIA. *Communications Magazine, IEEE*, 39(6):156–165, June 2001.

[96] T. Li, D. Leith, and D. Malone. Buffer Sizing for 802.11 Based Networks. *Topology*, cs.NI:1–14, 2010.

[97] C. R. Lin and M. Gerla. Asynchronous Multimedia Multihop Wireless Networks. In *INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, page 118, Washington, DC, USA, 1997. IEEE Computer Society.

[98] C. R. Lin and M. Gerla. Real-time Support in Multihop Wireless Networks. *Science*, 5:125–135, 1999.

[99] X. Lin, N. B. Shroff, and R. Srikant. A Tutorial on Cross-layer Optimization in Wireless Networks. *Selected Areas in Communications, IEEE Journal on*, 24; 24(8):1452–1463, 2006.

[100] A. Lindgren and E. M. Belding-Royer. Multi-path Admission Control for Mobile Ad hoc Networks. *SIGMOBILE Mob.Comput.Commun.Rev.*, 8(4):68–71, 2004.

[101] J. W. S. Liu. *Real-Time Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[102] M. Liu, R. R. Talpade, A. Mcauley, and E. Bommaiah. IETF Draft: AM-Route: Adhoc Multicast Routing Protocol, 1999.

[103] Q. Liu, X. Wang, and G. B. Giannakis. A Cross-layer Scheduling Algorithm with QoS Support in Wireless Networks. *Vehicular Technology, IEEE Transactions on*, 55(3):839–847, 2006.

[104] C. Lochert, B. Scheuermann, and M. Mauve. A Survey on Congestion Control for Mobile Ad hoc networks: Research Articles. *Wirel.Commun.Mob.Comput.*, 7(5):655–676, 2007.

[105] M. Loiacono, J. Johnson, J. Rosca, and W. Trappe. Cross-Layer Link Adaptation for Wireless Video. In *IEEE International Conference on Communications (ICC)*, pages 1–6, Piscataway, NJ, USA, May 2010. IEEE Communications Society.

[106] J. Luo, D. Ye, X. Liu, and F. Mingyu. A Survey of Multicast Routing Protocols for Mobile Ad-Hoc Networks. *IEEE Communications Surveys and Tutorials*, 11(1), 2009.

[107] P. Macharla, R. Kumar, A. K. Sarje, and M. Misra. A QoS Routing Protocol for Delay-sensitive Applications in Mobile Ad hoc Networks, 2006.

[108] B. S. Manoj and C. S. R. Murthy. Real-time Traffic Support for Ad hoc wireless Networks. In *Networks, 2002. ICON 2002. 10th IEEE International Conference on*, pages 335–340, 2002.

[109] M. Masoudifar. A review and performance comparison of QoS multicast routing protocols for MANETs. *Ad Hoc Networks*, 7(6):1150–1155, 2009.

[110] S. Merigeault and C. Lamy. Concepts for Exchanging Extra Information between Protocol Layers Transparently for the Standard Protocol Stack, 2003.

[111] S. Murthy, P. Hegde, and A. Sen. Design of a Delay-Based Routing Protocol for Multi-Rate Multi-Hop Mobile Ad Hoc Networks, 2009.

[112] P. Navaratnam, H. Cruickshank, and R. Tafazolli. A Link Adaptive Transport Protocol for Multimedia Streaming Applications in Multi Hop Wireless Networks. In *MobiMedia '07: Proceedings of the 3rd international conference on Mobile multimedia communications*, pages 1–6, Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering; ICST, Brussels, Belgium, Belgium, 2007. ICST.

[113] T. Nguyen, K. Nguyen, and L. He. Collaborative Distributed Admission Control (CDAC) for Wireless Network. In *ICDCSW '07: Proceedings of the 27th International Conference on Distributed Computing Systems Workshops*, page 75, Washington, DC, USA, 2007. IEEE Computer Society.

[114] Ministry of Defence. Network enabled capability, 2005.

[115] G. Pei, P. A. Spagnolo, S. Bae, T. R. Henderson, and J. H. Kim. Performance Improvements of OSPF MANET Extensions: A Cross Layer Approach. In *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–7, 2007.

[116] F. Peng, S. Cheng, and J. Ma. Proposal to Apply ECN to Wireless and Mobile Networks. In *The 10th Annual Internet Society Conference*, July 2000.

[117] C. Perkins, E. Belding-Royer, and S. Das. IETF RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing, 2003.

[118] D. D. Perkins and H. D. Hughes. A Survey on Quality-of-Service Support for Mobile Ad hoc Networks. *Wireless Communications and Mobile Computing*, 2(5):503–513, 2002.

[119] L. L. Peterson and B. S. Davie. *Computer Networks: A Systems Approach, Fourth Edition (The Morgan Kaufmann Series in Networking)*. Morgan Kaufmann, March 2007.

[120] R. S. Prasad, M. Murray, C. Dovrolis, K. Claffy, R. Prasad, and C. D. Georgia. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. *IEEE Network*, 17:27–35, 2003.

[121] L. Qin and T. Kunz. Survey on Mobile Ad Hoc Network Routing Protocols and Cross-Layer Design. Technical report, Carleton University, August 2004.

[122] V. Raisinghani and S. Iyer. Architecting Protocol Stack Optimizations on Mobile Devices. In *First International Conference in Communication System Software and Middleware 2006*, 2006.

[123] V. T. Raisinghani and S. Iyer. Cross-layer Design Optimizations in Wireless Protocol Stacks. *Computer Communications*, 27(8):720–724, May 2004.

[124] V. T. Raisinghani and S. Iyer. ECLAIR: An Efficient Cross layer Architecture for Wireless Protocol Stacks. In *In World Wireless Congress, SF*. Delson Group Inc, 2004.

[125] V. T. Raisinghani and S. Iyer. Cross-layer Feedback Architecture for Mobile Device Protocol Stacks. *Communications Magazine, IEEE*, 44(1):85–92, 2006.

[126] K. Ramakrishnan, S. Floyd, and D. Black. IETF RFC 3168: The Addition of Explicit Congestion Notification (ECN) to IP, 2001.

[127] S. Ray, J. B. Carruthers, and D. Starobinski. RTS/CTS-induced Congestion in Ad hoc Wireless LANs. *IEEE WCNC*, pages 1516–1521, 2003.

[128] M. A. Razzaque, S. Dobson, and P. Nixon. Cross-layer Optimisations for Autonomic Networks. In *Advanced Autonomic Networking and Communication*, Whitestein Series in Software Agent Technologies and Autonomic Computing, pages 127–148. Birkh{ä}user Basel, 2008.

[129] T. B. Reddy, I. Karthigeyan, B. S. Manoj, and C. S. R. Murthy. Quality of sService Provisioning in Ad hoc Wireless Networks: a Survey of Issues and Solutions. *Ad Hoc Netw.*, 4(1):83–124, January 2006.

[130] X. Ren and J. Zhang. A Novel Cross-layer Architecture for Wireless Protocol Stacks. In *International Conference on Multimedia Technology (ICMT)*, Ningbo, China, 2010.

[131] F. Ridouard, J.-L. Scharbarg, and C. Fraboul. Stochastic Network Calculus for End-to-end Delays Distribution Evaluation on an Avionics Switched Ethernet. In *Industrial Informatics, 2007 5th IEEE International Conference on*, volume 1, pages 559–564, June 2007.

[132] B. Roman, I. Chatzigeorgiou, I. J. Wassell, and F. Stajano. Evaluation of Multi-carrier Burst Contention and IEEE 802.11 with Fading During Channel Sensing. *Personal Indoor and Mobile Radio Communications 2009 IEEE 20th International Symposium on*, pages 57–61, 2009.

[133] E. Royer and C. Perkins. IETF Draft: Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing, 2000.

[134] E. M. Royer and C.-K. Toh. A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks. *Personal Communications, IEEE*, 6(2):46–55, April 1999.

[135] P. Sarolahti and S. Floyd. Cross-layer Indications for Transport Protocols, October 2007.

[136] J.-L. Scharbarg, F. Ridouard, and C. Fraboul. A Probabilistic Analysis of End-To-End Delays on an AFDX Avionic Network. *Industrial Informatics, IEEE Transactions on*, 5; 5(1):38–49, 2009.

[137] B. Scheuermann, C. Lochert, and M. Mauve. Implicit hop-by-hop congestion control in wireless multihop networks. *Ad Hoc Networks*, 6(2):260–286, 2008.

[138] S. Sesay, Z. Yang, and J. He. A Survey on Mobile Ad Hoc Wireless Network. *Information Technology Journal*, 3(2):168–175, 2004.

[139] E. Setton. Congestion-aware Video Streaming over Peer-to-peer Networks, 2008.

[140] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod. Cross-layer Design of Ad hoc Networks for Real-time Video Streaming. *Wireless Communications, IEEE*, 12(4):59–65, 2005.

[141] E. Setton, X. Zhu, and B. Girod. Congestion-optimized Scheduling of Video over Wireless Ad hoc Networks, 2005.

[142] M. Seyedebrahimi and X.-H. Peng. Investigation of PHY, MAC and APP Layers for Adaptive and Cross-Layer Optimization in IEEE802.11 WLAN. In *Proceedings of the 10th IEEE International Conference on Computer and Information Technology*, CIT '10, pages 2581–2588, Washington, DC, USA, 2010. IEEE Computer Society.

[143] A. Sharma, B. Panigrahi, and S. De. Impact of Interference on Nodal Communication Range in Wireless Ad hoc nNetworks, 2011.

[144] S. Sharma, R. Mishra, and K. Singh. Current Trends and Future Aspects in Cross-Layer Design for the Wireless Networks. In *Advances in Computer Science and Information Technology. Computer Science and Engineering: Second International Conference*, CCSIT '12, pages 283–296, 2012.

[145] S.-T. Sheu and J. Chen. A Novel Delay-oriented Shortest Path Routing Protocol for Mobile Ad hoc Networks, 2001.

[146] J. L. Sobrinho, R. D. Haan, and J. M. Brazio. Why RTS-CTS is not your Ideal Wireless LAN Multiple Access Protocol. In *IEEE Wireless Communications and Networking Conference 2005*, volume 1, pages 81–87, 2005.

[147] V. Srivastava and M. Motani. Cross-layer Design: a Survey and the Road Ahead. *Communications Magazine, IEEE*, 43(12):112–119, 2005.

[148] P. Sudame and B. R. Badrinath. On Providing Support for Protocol Adaptation in Mobile wWireless Networks. *Mob.Netw.Appl.*, 6(1):43–55, 2001.

[149] BAE Systems. Information and Networks, 2009.

[150] Embedded Systems. AFDX / ARINC 664 Protocol. *Network*, page 24, 2010.

[151] T. Taleb, K. Kashibuchi, A. Leonardi, S. Palazzo, K. Hashimoto, N. Kato, and Y. Nemoto. A Cross-Layer Approach for an Efficient Delivery of TCP/RTP-Based Multimedia Applications in Heterogeneous Wireless Networks. *Vehicular Technology, IEEE Transactions on*, 57(6):3801–3814, 2008.

[152] A. Tanenbaum. *Computer Networks*. Prentice Hall Professional Technical Reference, 4th edition, 2002.

[153] H. Tian, S. K. Bose, C. L. Law, and W. Xiao. CLA-QOS: A Cross-Layer QoS Provisioning Approach for Mobile Ad-hoc Networks, 2005.

[154] C. Tschudin and R. Gold. LUNAR: a Lightweight Underlay Network Ad-hoc Routing Protocol and Implementation. In *In Proceedings of the 4th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN*, 2004.

[155] N. H. Vaidya, A. Dugar, S. Gupta, and P. Bahl. Distributed Fair Scheduling in a Wireless LAN. *IEEE Trans. Mob. Comput.*, 4(6):616–629, 2005.

[156] R. Vaishampayan and J. J. Garcia-Luna-Aceves. Robust Rree-based Multicasting in Ad hoc Networks. In *Performance, Computing, and Communications, 2004 IEEE International Conference on*, pages 647–652, 2004.

[157] T. Vanhatupa, M. Hannikainen, and T. D. Hamalainen. Evaluation of Throughput Estimation Models and Algorithms for WLAN Frequency Planning. *Computer Networks*, 51(11):3110, 2007.

[158] S. Varatharajan, A. Jabbar, Mohammad A., R. Naidu, J. P. Rohrer, P. Upadhyay, and J. P. G. Sterbenz. Cross-Layer Framework for the ns-2 Simulator, June 2008.

[159] V. Vishnumurthy, T. Sandeep, B. S. Manoj, and C. S. R. Murthy. A Novel Out-of-band Signaling Mechanism for Enhanced Real-time Support in Tactical Ad hoc Wireless Networks. In *Real-Time and Embedded Technology and Applications Symposium, 2004. Proceedings. RTAS 2004. 10th IEEE*, pages 56–63, May 2004.

[160] C. Wang, K. Sohraby, B. Li, M. Daneshmand, and Y. Hu. A Survey of Transport Protocols for Wireless Sensor Networks. *Network, IEEE*, 20(3):34–40, 2006.

[161] K.-C. Wang and P. Ramanathan. QoS Assurances through Class Selection and Proportional Differentiation in Wireless Networks, 2005.

[162] M. Wang and K. Nahrstedt. Social structure based Routing of Intermittently Connected Network using Contact Information. In *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pages 1–7, 2008.

[163] Q. Wang and M. A. Abu-Rgheff. Cross-layer Signalling for Next-generation Wireless Systems. In *Wireless Communications and Networking, 2003, IEEE*, volume 2, pages 1084–1089 o.2, March 2003.

[164] C. Ware, J. Judge, J. Chicharo, and E. Dutkiewicz. Unfairness and Capture Behaviour in 802.11 Adhoc Networks, 2000.

[165] R. Winter, J. H. Schiller, N. Nikaein, and C. Bonnet. CrossTalk: Cross-layer Decision Support based on Global Knowledge. *Communications Magazine, IEEE*, 44(1):93–99, January 2006.

[166] W. Wolf. *High-Performance Embedded Computing: Architectures, Applications, and Methodologies*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.

[167] C W Wu and Y C Tay. AMRIS: a Multicast Protocol for Ad hoc Wireless Networks, 1999.

[168] G. Wu, Y. Bai, J. Lai, and A. Ogielski. Interactions between TCP and RLP in wireless Internet, 1999.

[169] Z. Wu and J. Wu. Cross-layer Design for Packetized Video Communications over Wireless Ad hoc Networks. In *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010.

[170] Z. Wu and J. Wu. Cross-Layer Routing Optimization for Video Transmission over Wireless Ad Hoc Networks. In *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010.

[171] H. Xiao, W. K. G. Seah, A. Lo, and K. C. Chua. A Flexible Quality of Service Model for Mobile Ad-hoc Networks. In *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, volume 1, pages 445–449 o.1, 2000.

[172] K. Xu, M. Gerla, and S. Bae. Effectiveness of RTS/CTS Handshake in IEEE 802.11 Based Ad hoc Networks. *Ad Hoc Networks*, 1(1):107–123, 2003.

[173] K. Xu, M. Gerla, L. Qi, and Y. Shu. TCP Unfairness in Ad hoc Wireless Networks and a Neighborhood RED Solution. *Wirel.Netw.*, 11(4):383–399, 2005.

[174] Q. Xue and A. Ganz. Ad hoc QoS On-demand Routing (AQOR) in Mobile Ad hoc Networks. *Journal of Parallel and Distributed Computing*, 63(2):154, 2003.

[175] G. Xylomenos, G. Polyzos, P. Mahonen, and M. Saaranen. TCP Performance Issues over Wireless Links. *IEEE Communications Magazine*, 39(4):52–58, 2001.

[176] G. Xylomenos and G. C. Polyzos. Internet Protocol Performance over Networks with Wireless Links. *Network, IEEE*, 13(4):55–63, 1999.

[177] Y. Yamao, Y. Kida, and Y. Kadowaki. Cross-Layer Multi-Hopping Scheme for Efficient and Reliable Transmission in Fading Environment. In *VTC Fall*, pages 1–5, 2010.

[178] S. Yang, N. Sang, and G. Xiong. Integrated safety critical systems on reliable real time network, 2003.

[179] Y. Yang and R. Kravets. Distributed qos guarantees for realtime traffic in ad hoc networks, 2004.

[180] Y. Yang and R. Kravets. Contention-Aware Admission Control for Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 4(4):363–377, 2005.

[181] M. Yao, Z. Qiu, and K. Kwak. Leaky Bucket Algorithms in AFDX. *Electronics Letters*, 45; 45(11):543–545, 2009.

[182] S. Yuhe and X. Jie. A New Cross-Layer Interface for MANET. *Circuits, Communications and Systems, Pacific-Asia Conference on*, pages 175–178, 2009.

[183] D. Zhang, J. Jiang, A. Anani, and H. Li. QoS-guaranteed Packet Scheduling in Wireless Networks. *The Journal of China Universities of Posts and Telecommunications*, 16(2):63, 2009.

[184] L. Zhou, B. Zheng, B. Geller, A. Wei, and J. Cui. Joint QoS control for Video Streaming over Wireless Multihop Networks: A Cross-layer Approach. *AEU - International Journal of Electronics and Communications*, 63(8):638, 2009.

[185] X. Zhu and B. Girod. Distributed Media-Aware Rate Allocation for Wireless Video Streaming. *Information Systems Journal*, 20(11):1462–1474, 2009.

[186] X. Zhu and B. Girod. A Unified Framework for Distributed Video Rate Allocation over Wireless Networks. *Ad Hoc Networks*, 9(4):608–622, 2011.