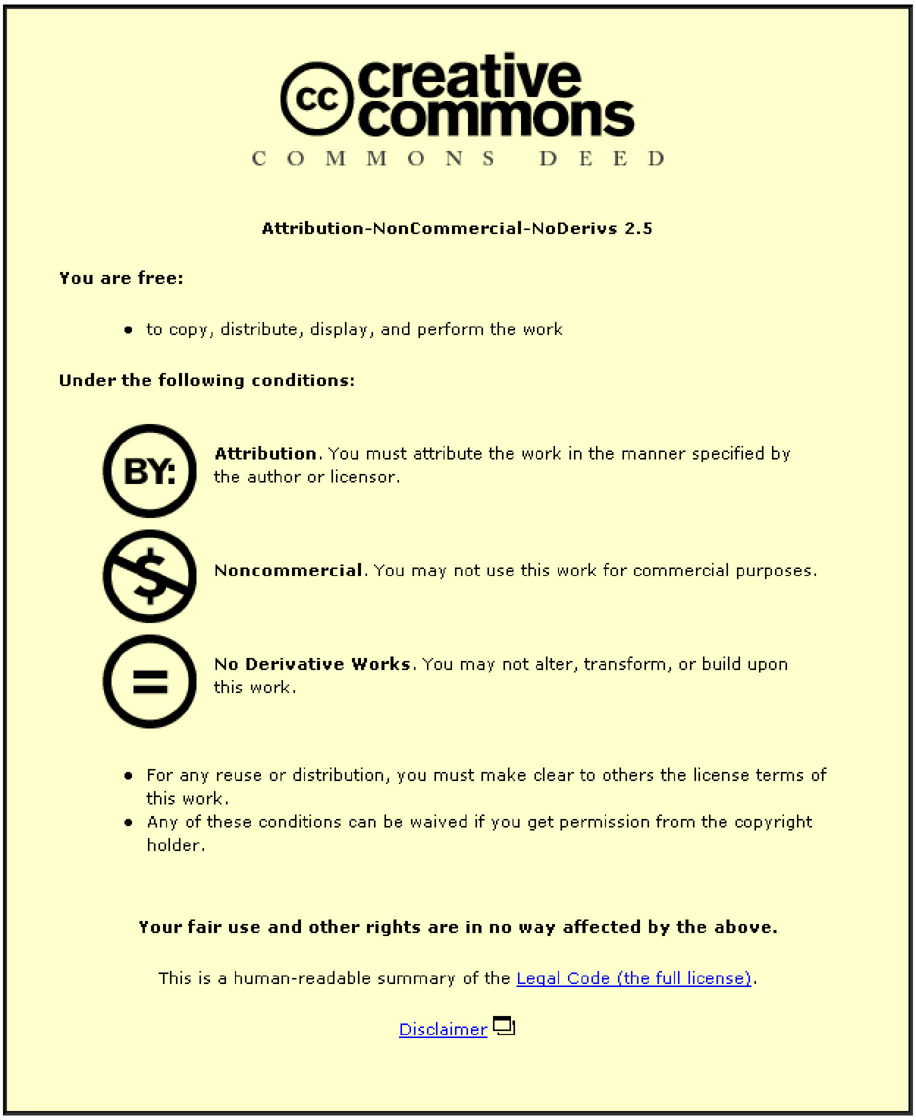
Architecture for the management and presentation of communication network performance data

PUBLISHER

© IEEE

REPOSITORY RECORD

Phillips, Iain W., David J. Parish, Mark Sandford, Omar Bashir, and Anthony Pagonis. 2019. “Architecture for the Management and Presentation of Communication Network Performance Data”. figshare. https://hdl.handle.net/2134/2578.



Architecture for the Management and Presentation of Communication Network Performance Data

Iain Phillips, David Parish, *Member, IEEE*, Mark Sandford, Omar Bashir, and Anthony Pagonis

***Abstract*—Packet-based data networks such as British Telecommunications (BT) plc's switched multimegabit data service (SMOS) have performance that varies with customer behavior, network design, and network provision. This paper describes a system researched to monitor such networks from the customer's perspective, but on behalf of the operator. The information generated allows the operator to determine whether the network is performing correctly or whether changes or customer load will require the network to be changed in some way. Such networks are increasingly important as many parts of the Internet are built on similar technologies. The paper specifically considers an architecture that encompasses all the main functionality involved in performance management of networks. Consideration is given to the management and dissemination of the data collected. A database to enhance network data mining performance is described. A method to identify the periods of significance in the measurement data based on a statistical test and a neural network is incorporated into a realization of the architecture. In addition, a novel mechanism for the real-time presentation of the network performance data in an operator-friendly manner is given.**

***Index Terms*—Internet, monitoring, networks, performance, switched multimegabit data service (SMDS).**

I. INTRODUCTION AND BACKGROUND

THE WIDESPREAD adoption of the Internet and associated services as an integral part of business, economic, and social practices has seen a huge increase in the use and deployment of related networking technologies. Monitoring the many networks that form the Internet is essential both for maintaining current services and for provisioning future services [1]. Network operators monitor many aspects of network performance. Some performance-related statistics can be collected nonintrusively by equipment that are already present in the network such as routers and switches; these would include data accessed by management protocols and systems such as simple network management protocol (SNMP) or by passive monitors capturing packets as they pass a capture point on the network. Other statistics must be actively measured, e.g., by sending test-probe traffic across a network.

One-way latency is a useful performance-related statistic that accurately reflects the performance of a network as it is perceived by a user. A user's requirement is that data are transferred reliably and that it is transferred as quickly as possible. Assuming reliability, delay becomes the ultimate indicator of performance. Delay may be attributed to many other monitored factors such as congestion or errors (if some retransmission scheme is used, which would be necessary if reliability is to be assumed). A user, however, is primarily concerned with the resultant performance and this is best measured in terms of delay. Several groups have developed measurement schemes to monitor one-way latency. These schemes typically involve a large number of widely distributed latency probes and often use global positioning system (GPS) to achieve time synchronization. The volume of data collected by these monitor probes can be significantly large and schemes are needed to store, analyze, and present the information [2]–[6].

Other network monitoring projects have tended to concentrate on a specific aspect of measurement, such as a particular probe tool or statistic. Examples include Skitter, which has concentrated on trace-route information and visualization of network topology, active measurement project (AMP), which has collected round-trip time (RTT) measurements between a number of points on the network, and cooperative monitoring (COMO) (formerly nProbe and gridProbe [7]), which concentrates on passive measurements. The aim of this paper has been to produce a general system that can adapt to all types of measurement.

The key points addressed by this paper are the issues of how all aspects of the performance measurement process can be integrated into a coherent system, i.e., a system that controls all aspects of network monitoring from data collection to storage of raw and processed data to the efficient presentation of important performance changes. The paper described includes the storage of data summaries, the detection of performance changes, and the real-time visual presentation of overall performance. This system has the following novel aspects.

1) Measurement data and its summaries are efficiently managed for speed of later processing to support data mining operations.
2) Raw delay data collected from the network are filtered into "interesting" events. These limit the volume of information presented to network analysis staff.
3) Both these event data and trivially processed raw data are presented to the operators using a newly researched interface that aims to improve the efficiency of real-time system-human interaction.
4) Architecture to integrate different aspects of the system with each other and with those of other systems.

Network performance can be fundamentally considered in terms of latency and loss. The paper described here aimed to

---

Manuscript received May 27, 2005; revised October 25, 2005. This work was supported by British Telecommunications.

I. Phillips is with the Department of Computer Science, Research School of Informatics, Loughborough University, Leics, LE11 3TU, U.K. (e-mail: I.W.Phillips@lboro.ac.uk).

D. Parish, M. Sandford, O. Bashir, and A. Pagonis are with the High Speed Networks Group, Department of Electronic and Electrical Engineering, Loughborough University, Leics, U.K. (e-mail: D.J.Parish@lboro.ac.uk).

Digital Object Identifier 10.1109/TIM.2006.873783

0018-9456/$20.00 © 2006 IEEE

produce a system that can provide this information. Initially, standard methods such as ping were investigated, but while these produced adequate results for most applications, more accuracy was required. The main problems with ping were the fluctuation in delays due to operating system being busy (in our experiments, the pinging machine was a SUN workstation) and the inability to measure single-way delays, as ping measures round-trip times. Further, the Internet control message protocol (ICMP) echo request message of ping needs to be processed by a device at the far end of the test, in order to produce an echo reply message, and this processing takes further time, which is not directly attributable to the network.

A large data network run by a telecommunications service provider such as British Telecommunications plc (BT) will provide services for a number of subscribers. Each of these will have negotiated (or purchased) a particular level of service, defined by a service level agreement (SLA). The aim of this research is to monitor the network from a customer's perspective, but on behalf of the network operator, in order to check how the network is performing in relation to its SLA. This set of operations involves much more than simply detecting whether or not the SLA has been met. The degree to which the SLA is adhered to is also important, as is the ability to present such information quickly and efficiently to the network operators. Further outcomes of this style of performance monitoring include noting when a network is approaching its SLA, due to additional customers, and also seeing how changes to the network affect the customer's performance.

Section II covers the measurement problem in more detail. Section III describes some of the system modules that were researched and developed to meet the aims described. Section IV then describes a reference implementation on BT's switched multimegabit data service (SMDS) service. Section V gives some conclusions.

II. SCOPE

The aim of the system is to produce real-time measurements of delay and loss across the network and derive various performance indicators from these, with minimal human intervention. Furthermore, these measurements need to be collected and disseminated in a suitable form to the network operators and if necessary the subscribers. This dissemination may be in real-time to operational support staff or in the form of regular (weekly or monthly) reports to provision engineers. These reports can consist of summaries of the measured data with particular interesting events being highlighted.

A system for such measurement needs elements to gather information, store this information, and then process it for dissemination and presentation. Secondary aims for this system include that it be integrated and automatic, with human intervention eliminated where possible. Accompanying these aims is the notion that where human interaction with the system is necessary, then efficient use of time is made.

Many integrated network architectures exist [8]. We consider only the performance management of networks and propose an architecture for this that abstracts the measurement process away from the entire management aspects of such architectures.

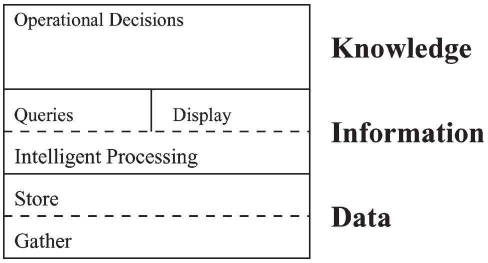


Fig. 1. Architecture of a measurement system.

Fig. 1 shows a specification architecture of a measurement system. At the lowest layer, we are concerned with gathering and storing raw data, such as measurements of a packet transmission and receipt. Above this, an information layer begins to process these data into more useful values, which can then be used by the top layer to provide a useful service to the network operator. These services might include automatic provision decisions for enhancements to the network where loading is predicted to be high, or automatic investigation of faults, by scheduling more investigative testing on links that display faults at specific times of the day or under specific circumstances. This layer could also be referred to as an action layer as the knowledge determined here could lead to changes in the network being monitored.

III. SYSTEM MODULES

A. Gathering Delays

The lowest layer of the network management system architecture includes the actual gathering of the basic data (in this case packet delay and loss). Such measurements are obtained intrusively by injecting additional traffic into the network in a controlled manner. Commercially available equipment can be used here, but to ensure full integration with the rest of the system a proprietary gathering device was designed. This is able to send and receive user datagram protocol (UDP) packets over a network. UDP is used to reduce delay in the network stack and to ensure that loss is not automatically corrected. This tool combined a disk operating system (DOS)-based PC with a custom timing card, conventional network adapter (for the access network used), and measurement software, and is known as a monitor station. DOS was chosen as the operating system with the lowest overhead. The timing card comprised a GPS [9] receiver that produces a 1-Hz signal, the rising edge of which is synchronized world wide. A 10-MHz counter, which is reset on the rising GPS pulse, provides subsecond measurement. Recording the count immediately prior to reset provided for variations in oscillator frequency due to temperature and manufacture. Each time the timing card is latched, three values can be read: the current time and date, the counter value at the time of latching, and the counter value immediately prior to the last reset. These are, respectively, known as the date-time, the partial count and the full count. Each quantity is a 32-bit value. The subsecond quantity of the timing measurement is obtained by simply dividing the partial count by the full count. This provides additional accuracy as variations in the oscillator

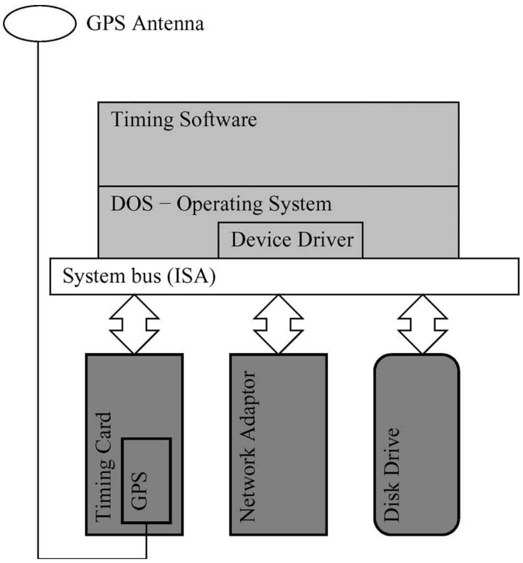


Fig. 2. Functional diagram of the monitor station.

performance due to either manufacture or ambient conditions are now factored out of the measurement process [10].

Software to start tests, where a test is a stream of packets regularly spaced in time, stop tests, retrieve logs, and update the workstation was written. Typical tests vary from one packet per second to one per minute, although considerable variation beyond these limits is possible. It is also possible to generate packets according to a statistical distribution, e.g., Gaussian, Poisson, etc. Packet sizes are limited to those supported by the access network. Each monitor station can receive measurement packets from any other monitor station and support a number of concurrently running tests. Each transmitted and received packet is logged and these logs are retrieved to a control station. Here, higher level processing is performed and reports generated. A functional diagram of the monitor station is shown in Fig. 2.

B. Storing Data

With any measurement of a large-scale system, a large amount of data are generated. For the later functions of processing and dissemination of information to be efficient the design of the data store or database is significant. This is not purely a case of efficient use of storage space as processing time is also a significant factor. It is possible that successive high-level queries require the repeated processing of the same lower level requests, as an operator searches for reasons behind a simple network event. The concept that a particular type of query may be regularly repeated has been mentioned in the literature [11]. A typical scenario is that when presented with a month's worth of performance data, an operator may notice an event and choose to investigate a particular day in more detail. Such an activity will require the repeated processing of that day's data. Caching the results of such queries gives a simple and obvious performance gain at the expense of additional storage. Further investigation showed that typical queries are often related in other ways, particularly temporally. For example, a query for median delay across each day for a month might be followed by a query for median data across a single day within that month. An intermediate data structure can be used to hold part-processed data, which can be used by multiple requests and therefore saves repeated operation on the individual measurements [12]. The structure of the so-called intermediate information (II) can have a number of forms, and be organized to enhance the efficiency of the system [13]. These data structures will cover all delays in a time interval for a particular test route. An important feature of the information is that neighboring (in time) structures can be combined into larger structures. For example, an intermediate object that covers a particular test route for an hour can easily be combined with the intermediate object from the following hour for the same route to give a 2-h intermediate object. This paper considered two such intermediate data structures:

1) object consisting of the delay distribution for time interval;
2) object consisting of the sorted delay values for a particular time interval.

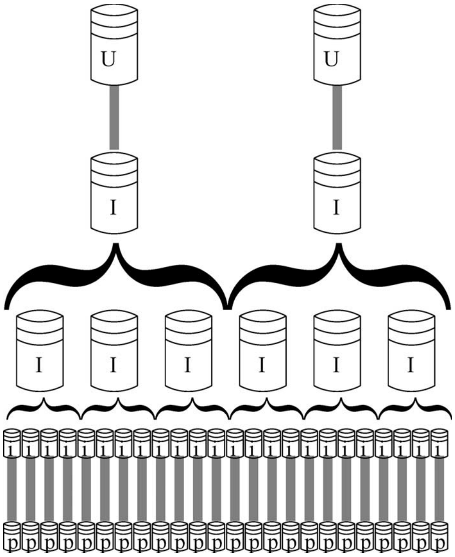


Fig. 3. Intermediate information data structure.

The merging of two objects of this type is a simple $O(N)$ process. This leads to a number of performance enhancements in terms of the processing effort required to generate the requested result. In all cases there is, however, an increased storage requirement for the II objects.

Fig. 3 illustrates II in operation. In this figure, there are five data layers, at the bottom there is basic primitive information as collected by the monitoring system (shown with a P). Above this are three layers of II (shown with an I). The lowest of these alters, the form of the primitive data for further processing, an example would be the ordering of the delay values. The middle II layer groups together elements of the first II layer into fewer larger objects (for example these may represent a full day's worth of sorted delay measurements). The final II layer collects the data into a larger object, perhaps a week's worth of measurements. Finally, the top layer of the diagram represents Useful Information (shown as U), which has been generated in response to a user query. An example would be the 95th percentile of delay for each week of measurements. Note that the

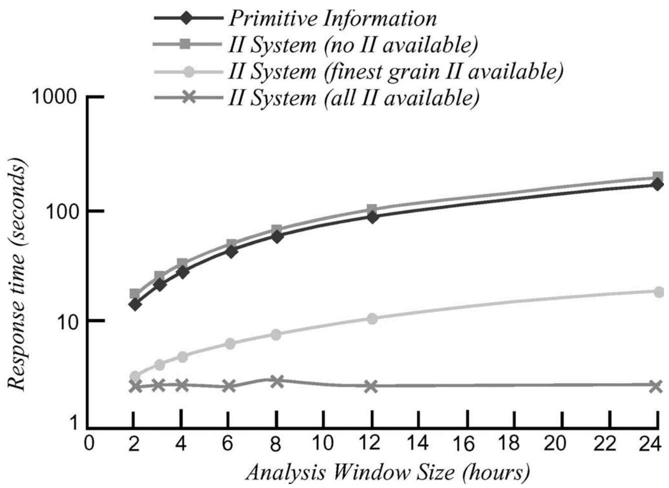


Fig. 4. Intermediate information performance.

generation of the 95th percentile of delay on a daily basis could now be performed very simply by using the middle II layer objects.

An example of the speed up obtained for query processing is shown in Fig. 4. In this example, the user has queried the mean delay for a series of different window sizes. The system response time is considered for four different situations: a simple database; an II database where no II is available; an II database where some II is available; the II database when ideal II is already available. The significant speed up in response time is clearly seen in [14] and [15].

C. Dissemination of Information

In addition to servicing specific user requests, the system needs to provide regular performance updates on an ongoing basis. In our system, we generate both simple reports and more specific indications of unexpected performance. We term these data exceptions (DEs). In addition, a real-time display of data is also presented to network operations staff in an efficient manner to reduce information overload. This incorporates a technique we term a figural deformity visualization (FDV).

1) Reports: The simplest form of presentation of network delay measurements is with a graph of percentiles of delay against time. A normal arithmetic mean will hide certain interesting features, and therefore, three percentile values are used. The 50th percentile provides a measure of average performance; the 95th percentile provides performance relative to the network SLA; and the 99th percentile is indicative of the worst-case performance. Loss can be represented in graphical form, but where loss rates are low, a text-based reporting mechanism can be used.

2) DEs: When viewing these graphs, the network operator will sometimes observe interesting events, when the delay experiences an unexpected change. [16], [17] Possible changes are as follows.

1) *Spikes.* These are a short period of high delay, commonly caused by network fault conditions.
2) *Steps.* These, as the name suggests, are fixed increases or decreases to a steady-delay measure, which remains at the new level for some time. These are commonly caused by routing changes in the network, where traffic will usually take a new route, either longer or shorter, after the change.
3) *Changes in time-of-day-delay variation.* Time-of-day-delay variation is the pattern of delay over a day. This usually increases during working hours. This pattern can change due to increased traffic, sometimes after rerouting, and sometimes after increased customer traffic load.

Examples of these are shown in Fig. 5. All of these graphs cover a two-week period. The weekends are shown by the reduced maximum values, i.e., reduced time-of-day-delay variation. On the final graph, the time-of-day-delay variation increases on the first Thursday.

We have denoted these events DEs, as they indicate periods where the delay data differs from some expectation. DEs may be related, in that a step change may be accompanied by a change in time-of-day-delay variation. Note that a change in delay does not always yield a DE. For example, time-of-day delay variation reduces at the weekend due to reduced user load. This is to be expected and is not classified as a DE.

DEs are automatically detected using a two-stage process. First, a statistical test is performed (Kolmogorov–Smirnov [18]); this gives an indication that an exception has occurred in the data. Subsequently the delay data immediately surrounding the exception is passed to a neural network for classification [19]. The neural network is initially trained with a set of DEs of various types taken from the communication network. In our evaluation of this approach, a classification accuracy of 99% was eventually achieved on this training set. When used on unseen exceptions, the trained neural network achieved a classification accuracy of 79%, averaged over all exception types. The KS test, which proceeded the neural network, correctly identified 99.5% of the exceptions presented to it.

3) Displaying Real-Time Delay Data: DEs and graphs of delay are suitable for long-term management of the network. However, in some operational environments, there is a need for a more immediate response. Generally, the need here is for an easily assimilated view of the network performance to be given to aid routine operation or management of the network. The aim of our paper here is to combine value information from a number of different tests onto a single computer display so that an operator can see at a glance the information required and also observes trends. The research used an interface developed specifically for the purpose and referred to as the FDV [20]. This takes a common object with a recognizable shape and deforms certain aspects of this object depending on the values of the data being displayed. Fig. 6 shows an example FDV object based on the letter E and some of its possible deformations. As can be seen there are four possible degrees of freedom in the object and examples of some of these being deformed are shown. The principles behind the FDV include human perception [21] and short-term memory operation [22]. The concept visually presents the relative rather than the absolute value of the delay information. Relative trends in the delay data can now be simply visualized allowing the network operators to easily discern how well or badly network routes are performing. The information presented, however, is much more than a simple yes or no

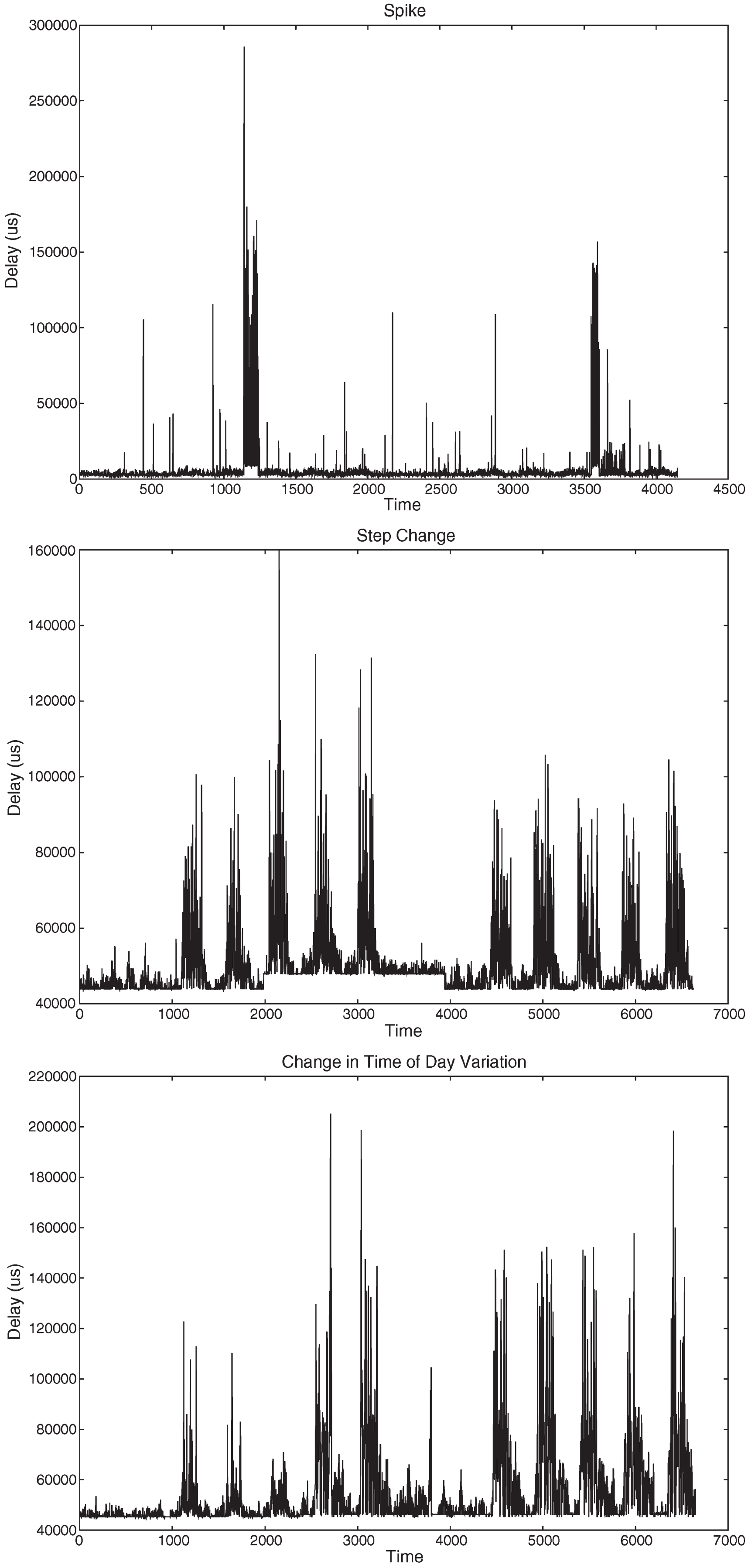


Fig. 5. Delay events.

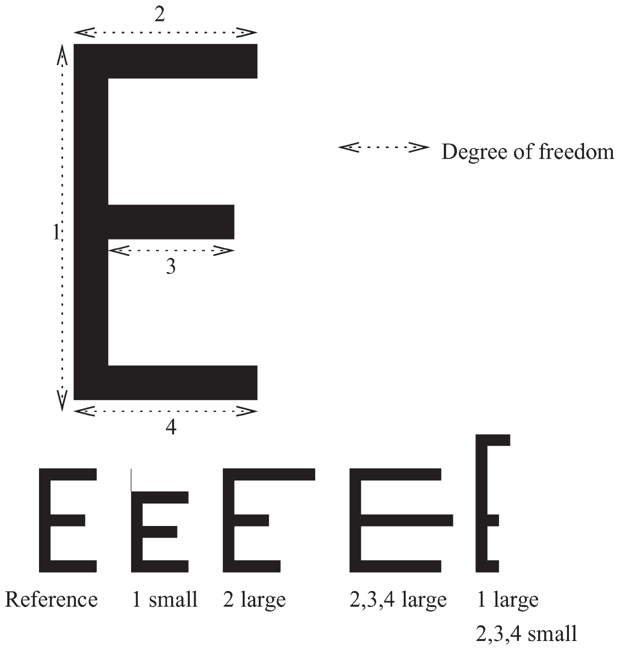


Fig. 6. FDV object based on an E.

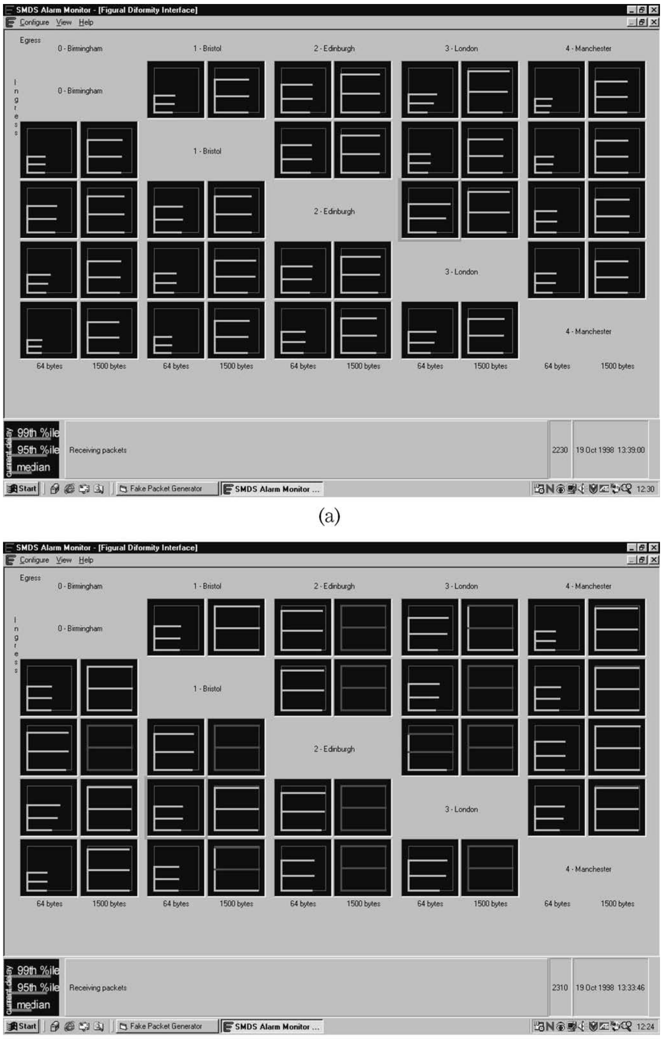


Fig. 7. FDV interface in use.

alarm, as the closeness to the SLA threshold is visualized very easily. Fig. 7 shows a complete FDV display for 40 different tests. Each has its own E figure, which are grouped in a matrix format according to ingress and egress point. A box is shown around each figure, this represents the SLA value. It is now easy to see which tests are close to or exceed their SLA. Further, the grouping of figures allows problematic ingress or egress points to be spotted easily. Fig. 7(a) shows the FDV display generated when the network is operating well within its SLA defined performance envelope as all elements of the E figures are within their defined boxes. Fig. 7(b), however, shows the display when SLA performance has been exceeded for a number of tests. The E objects related to such tests have grown to meet the SLA box edges, and if they exceed this bound, the object changes color.

With experience, the visualization of the delay percentile derivations yields useful information regarding the variation of traffic on the test routes. An assessment of the effectiveness of the FDV display for the network performance has been evaluated and has been shown to increase both operator response time and accuracy [23].

D. Implementation Architecture

The different modules shown earlier are designed to fit into an architecture governed by well-defined interfaces. The overall aim of the research has been to produce a totally automated system that could support a number of advanced features. These include the following:

1) detection of events;
2) intelligent future scheduling of tests, adapting the tests to the observed data;
3) intelligent storage and archiving of measurements;
4) report generation and distribution for regular reports;
5) identification of possible future problems based on past trends;
6) determination of relevant monitored events;
7) communication of monitored information to the operators.

An architectural overview of the system shown in Fig. 8. The director object, as its name suggests, directs operations by receiving and scheduling jobs from the system controllers. These can either be human users or machine agents. The jobs will usually cause actions on the stores and gatherers and the results from these jobs are returned back to the controllers. In some cases, the results from a job can automatically cause further jobs to be executed and this functionality is loaded onto the director.

IV. WORKING REALIZATION

The system has been realized in a number of ways but the most significant is an implementation on BT's SMDS network. This system, which is shown in Fig. 9, consists of a number of monitor stations (M), a focused test controller (F), a control station (C), and a display station (A). These elements work in conjunction with an access network and the SMDS network, as shown in Fig. 9. Each of these is now considered in turn.

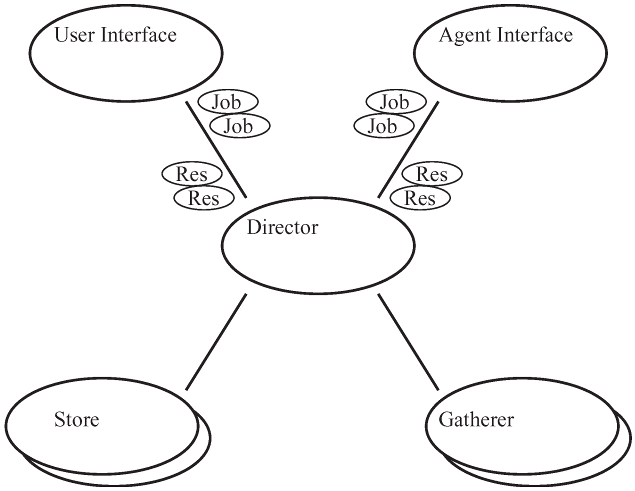


Fig. 8. Implemented system architecture.

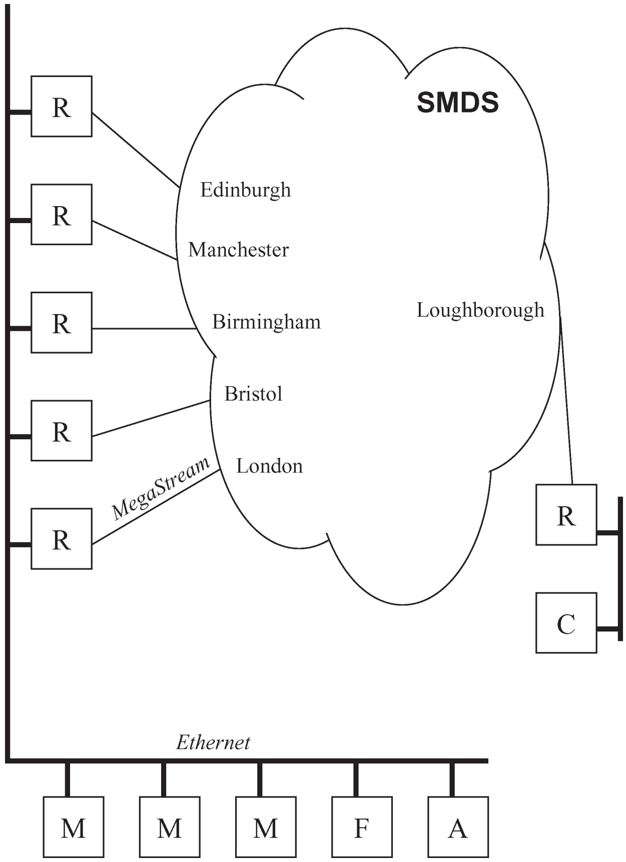


Fig. 9. System implementation on BT SMDS.

A. Monitor Station

This is the station described earlier. Packets are generally sent from and received at the same station in this realization. This obviates the need for separate GPS synchronization. Copies of the received test packets are in some cases forward to the display station (see below) for further processing and real time display. Periodically, data are collected from the monitor station by the control station for off-line processing and archiving.

B. Real-Time Display

This station receives relevant test packets and displays a figural deformity-based interpretation of them on its screen. This can be observed by the network operators and any unusual circumstances can be seen at a glance. While a simple alarm could be generated automatically, and then attract the operator's attention, the amount of human effort required to glance at the screen is minimal and psychologically preferred as significantly more information is conveyed. Specifically, the operator can now easily appreciate which tests are approaching their SLA defined performance thresholds.

C. Control Station

This station, which is in fact based at Loughborough, is where the tests are started and stopped, where the data are retrieved to and where regular status checking is made. The operator at Loughborough then processes the data and e-mails reports to network operators at weekly intervals for off-line analysis. It is possible for additional tests to be started on one of the spare monitor stations, these are either initiated from this control station or the focused test controller. The database of DEs is also held on this machine.

D. Focused Test Controller

This is located at the operational site. In addition to being responsible for the starting of addition tests, it also holds a database of DEs for network operational personnel to view at will.

E. Operational Experience

The system as described has been used by professional network operations staff and provides the only source of delay data for the network. They have commented favorably regarding the data produced and the usability of the system. This is confirmed by the three qualitative indicators listed as follows.

1) The system has been an instrumental in identifying "soft" faults, which do not trigger conventional alarms. One example of this is the identification of an interface card with a progressively degenerating optical interface.
2) Providing the ability to understand the impact of planned network changes on performance elsewhere in the network.
3) Operators requested additional tools based on the FDV principle to allow them to visualize other information not currently collected as part of the system described in this paper.

V. CONCLUSION

This paper has addressed the issue of integration of different elements of a network performance measurement system into a complete system. The special facilities it provides include the following:

1) proprietary system to intrusively measure networks, using GPS to ensure accuracy in one-way measurements, if appropriate;

2) data structure for a stored measurement data that provides an increased performance for multiple similar queries;
3) real-time display of measured information with suitable data-reduction techniques to allow an efficient use of an operator time;
4) means of identifying "interesting" data from those collected;
5) architecture to link the aspects of gathering, storing, and analyzing delay data.

ACKNOWLEDGMENT

The authors would like to acknowledge the contribution of BT to this research via the University Research Initiative entitled Managing Multiservice Networks. This study was performed when Dr. Phillips was a member of the High-Speed Networks Group in Electronic and Electrical Engineering at Loughborough University, Leics, U.K..

REFERENCES

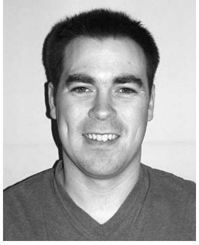
[1] K. Claffy and T. Mork, "What's next for data analysis? Status and challenges facing the community," *Proc. IEEE*, vol. 85, no. 10, pp. 1563–1571, Oct. 1997.
[2] V. Paxson, J. Mahdav, A. Adams, and M. Mathus, "An architecture for large-scale internet measurement," *IEEE Commun. Mag.*, vol. 36, no. 8, pp. 48–54, Aug. 1998.
[3] T. McGregor, H.-W. Braun, and J. Brown, "The NLANR network analysis infrastructure," *IEEE Commun. Mag.*, vol. 38, no. 5, pp. 122–128, May 2000.
[4] H. Uijterwaal, "RIPE test traffic project, Internet delay measurements using test traffic, first results," in *Proc. 1st Int. SANE Conf.*, Maastricht, The Netherlands, 1998.
[5] W. Matthews and L. Cotterell, "The PingER project: Active internet performance monitoring for the HENP community," *IEEE Commun. Mag.*, vol. 38, no. 5, pp. 130–136, May 2000.
[6] S. Kalidindi and M. J. Zekauskas, "Surveyor: An infrastructure for Internet performance measurements," in *Proc. Internet Netw.*, San Jose, CA, Jun. 1999.
[7] A. Moore, J. Hall, E. Harris, C. Kreibech, and I. Pratt, "Architecture of a network monitor," in *Proc. PAM*, Apr. 2003, pp. 77–86.
[8] G. Chen and Q. Kong, "Integrated management solution architecture," *Proc. IEEE/IFIP Netw. Operations Manage. Symp. "The Networked Planet: Management Beyond 2000,"* pp. 217–230, 2000.
[9] P. H. Dana, "Global positioning system (GPS) time dissemination for realtime applications," *Real-Time Syst.*, vol. 12, no. 1, pp. 9–40, Jan. 1997.
[10] D. Parish, I. Phillips, C. Rodgers, and P. Aliakia, "Providing an accurate global time reference to indoor computer based equipment," *Microprocess. Microsyst.*, vol. 24, no. 7, pp. 379–381, 2000.
[11] D. Omiecinski, "Concurrent file reorganization, clustering conversion and maintenance," *Bull. Tech. Comm. Data Eng.*, vol. 19, no. 2, pp. 25–32, Jun. 1996.
[12] J. Hammer, H. Garcia-Molina, J. Windon, W. Labio, and Y. Zhuge, "The Stanford data warehousing project," *IEEE Data Eng. Bull.*, vol. 18, no. 2, pp. 41–48, Jun. 1995.
[13] R. Rivest, "On self organizing sequential search heuristics," *Commun. ACM*, vol. 19, no. 2, pp. 63–67, Feb. 1976.
[14] O. Bashir, I. W. Phillips, D. J. Parish, J. L. Adams, and T. Spencer, "The management and processing of network performance information," *BT Technol. J.*, vol. 16, no. 4, pp. 203–212, Oct. 1998.
[15] O. Bashir, "Management and processing of network performance information," Ph.D. dissertation, Dept. Electron. Elect. Eng., Loughborough Univ., Leics, U.K., 1998.
[16] T. McGregor and H.-W. Braun, "Automated event detection for active measurement systems," in *Proc. Passive Active Meas.*, Amsterdam, The Netherlands, Apr. 2001.
[17] I. W. Phillips, J. M. Sandford, and D. J. Parish, "Processing network delay measurements into network events," *Proc. IEEE/IFIP Netw. Operations Manage. Symp. "The Networked Planet: Management Beyond 2000,"* pp. 955–956, 2000.
[18] H. R. Neave and P. L. Worthington, *Distribution Free Tests*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
[19] M. Sandford, "Detecting changes in network performance from low level measurements," Ph.D. dissertation, Dept. Electron. Elect. Eng., Loughborough Univ., Leics, U.K., 2001.
[20] A. Pagonis and D. Parish, "On the creation of a highly visualized interface for network performance data pre-analysis," in *Proc. 4th Commun. Netw. Symp.*, Jul. 1997, pp. 117–120.
[21] W. Jungmeister and D. Turo, "Adapting tree maps to stock portfolio visualization," Univ. Maryland, College Park, Tech. Rep. CS-TR-2996, 1992.
[22] J. Preece, *Human-Computer Interaction*. Reading, MA: Addison-Wesley, 1994.
[23] A. Pagonis, "An efficient visualization mechanism for communication network monitoring information," Ph.D. dissertation, Dept. Electron. Elect. Eng., Loughborough Univ., Leics, U.K., 1999.

Iain Phillips received the B.Sc. and Ph.D. degrees in computer science from Manchester University, Manchester, U.K.

He was a Research Assistant and Fellow with the High-Speed Networks group at Loughborough University, Leicestershire, U.K., and is currently a Lecturer in computer science. His research interests include network performance management, *ad hoc* networking, and application of the Internet.

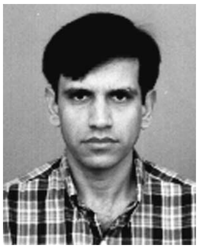
David Parish (M'02) received the B.Sc. (Honors) degree in physics with electronics and the Ph.D. degree in electronic engineering from the University of Liverpool, Liverpool, U.K., in 1979 and 1983, respectively.

From 1979 to 1980, he was with the Culham Laboratory, UKAEA. His research interests include the management, architecture, and application of communication networks.



Mark Sandford received the B.Sc. degree in mathematics and computation and the Ph.D. degree in computer networks from Loughborough University, Leicestershire, U.K.

He was a Researcher with the High-Speed Networks group at Loughborough University, Leicestershire, U.K., and is currently an Application Networking Specialist with Reuters, U.K.



Omar Bashir received the B.Eng. degree from NED University of Engineering and Technology, Karachi, Pakistan, in 1989, and the M.Sc. and Ph.D. degrees from Loughborough University, Leicestershire, U.K., in 1995 and 1998, respectively.

He is currently a Systems Engineer with the Warehousing and Distributions Systems Division, FKI Logistex, Loughborough, Leicestershire, U.K. His academic career includes lecturing and applied research. He also offers consultation sessions for a number of IT and telecommunication startup organizations.

Anthony Pagonis was a research student in the High Speed Networks Group, Department of Electronic and Electrical Engineering, Loughborough University, Leics, U.K. at the time of this research.