

This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

On termination of a flooding process

PLEASE CITE THE PUBLISHED VERSION

PUBLISHER

ACM

VERSION

AM (Accepted Manuscript)

PUBLISHER STATEMENT

© Owner/Author 2019. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in the Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, <http://dx.doi.org/10.1145/3293611.3331586>.

LICENCE

All Rights Reserved

REPOSITORY RECORD

Hussak, Walter, and Amitabh Trehan. 2019. "On Termination of a Flooding Process". figshare.
<https://hdl.handle.net/2134/9756566.v1>.

Brief Announcement: On Termination of a Flooding Process

Walter Hussak

Amitabh Trehan

W.Hussak@lboro.ac.uk

A.Trehan@lboro.ac.uk

Computer Science, Loughborough University
Loughborough, Leicestershire, U.K.

ABSTRACT

Flooding is a fundamental distributed algorithms technique. Consider the following flooding process, for simplicity, in a synchronous message passing network: A distinguished node begins the flooding process by sending the (same) message to all its neighbours in the first round. In subsequent rounds, every node receiving the message relays a copy of the message further to all those, and only those, nodes it did not receive the message from in the previous round. However, the nodes do not remember if they've taken part in the flooding before and therefore will repeat the process every time they get a message. In other words, they execute an *amnesiac* flooding process with memory only of the present round. The flooding process terminates in a particular round when no edge in the network carries the message in that, and, hence, subsequent, rounds. We call this process *Amnesiac Flooding (AF)*.

In this work, the main question we address is whether *AF* will terminate on an arbitrary network (graph) and in what time? We show that, indeed, *AF* terminates on any arbitrary graph. Further, *AF* terminates in at most D rounds in bipartite graphs and at most $2D + 1$ rounds in non-bipartite graphs - in this brief announcement, we show this for the bipartite case only.

We also show that in a natural asynchronous variant of *AF*, an adversary can always ensure non-termination.

CCS CONCEPTS

• **Mathematics of computing** → **Graph algorithms**; Combinatorial algorithms; • **Networks** → **Network algorithms**; *Network protocol design*; • **Theory of computation** → **Graph algorithms analysis**; *Concurrent algorithms*; • **Computing methodologies** → **Distributed algorithms**; • **Information systems** → Social networks;

KEYWORDS

Flooding, Memory, Termination, Bipartiteness, Synchrony

ACM Reference Format:

Walter Hussak and Amitabh Trehan. . Brief Announcement: On Termination of a Flooding Process. In . ACM, New York, NY, USA, 3 pages.

1 INTRODUCTION

Flooding is among the most basic of distributed algorithms. To quote Apnes [1]: *Flooding is about the simplest of all distributed algorithms. It's dumb and expensive, but easy to implement, and*

gives you both a broadcast mechanism and a way to build rooted spanning trees. Often flooding is implemented with a flag that is set when the message is seen for the first time to ensure termination and with other mechanisms to detect termination of the process (see e.g. [2]). We are interested in a variant of flooding which does not explicitly use such a flag or even keep a memory of having seen the message before. A node only sends the message to the complement of its neighbours from whom it has just received the message and subsequently forgets about that activity. In a way, the node behaves like an aggressive social media (say, *WhatsApp*) user that has a compulsion to forward every message but does not want to annoy those who have just sent it the message it's forwarding. This idea of avoiding the most recently chosen node(s) has been used before in distributed protocols e.g. in social networks [3] and broadcasting [4] but we are not aware of this fundamental variant of flooding having been studied before.

1.1 Model and Problem Definitions:

Let $G(V, E)$ be the graph representing a network where the vertices represent the nodes of the network and edges represent the connections between the nodes. Consider the following process in a synchronous network where the computation proceeds in synchronous rounds where each round consists of every node receiving messages from all its neighbours, doing local computation and sending messages to all (or some of) its neighbours. No messages are lost in transit.

Definition 1.1. Amnesiac Flooding (AF): A distinguished node, say ℓ , sends a message (say, M) to all its neighbours in round 1. In subsequent rounds, every node receiving M forwards a copy of M to every, and only those, nodes it did not receive the message from in that round.

Note that this is an 'amnesiac' process i.e. nodes do not retain memory of having received or sent the message in the previous rounds. We say that *AF terminates* when no message (i.e. a copy of M) is being sent over any edge in the network. The question is whether, *for any finite graph G over n nodes, beginning from any arbitrary node, will AF always terminate?* In this work, we prove that this is indeed true i.e. this flooding process will terminate for every G . We prove that *AF* terminates in at most D rounds on a bipartite graph, where D is the diameter of the graph. We also prove (in the full paper) that the termination time on any arbitrary graph is, in fact, $O(D)$ too (but strictly larger than D in non-bipartite graphs).

We believe that we pose an interesting theoretical question. We believe our results may have applications in designing amnesiac/low-memory algorithms, in topology detection (e.g. to detect/test non-bipartiteness of graphs), analysing social media and natural flooding processes, among others. We also show that in a natural asynchronous variant of *AF*, an adversary can always ensure non-termination and pose a number of interesting questions.

1.2 Some illustrative examples:

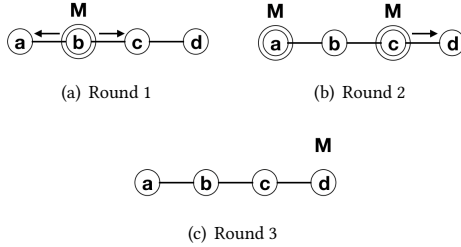


Figure 1: Amnesiac Flooding over a line network beginning with node b in 2 ($< \text{diameter} = 3$) rounds. Circled nodes are sending M in that round.

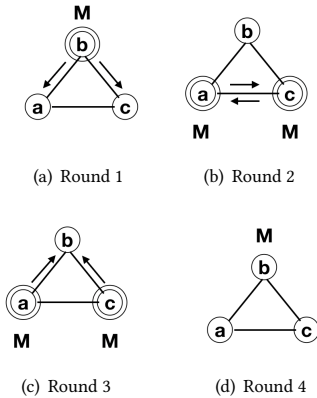


Figure 2: *AF* over a Triangle (Odd Cycle/Clique) network beginning with node b . Both node a and c send M to each other in round 2 and to b in round 3. Also, this is an odd (# nodes) cycle and termination takes $2D + 1$ time ($D = \text{diameter} = 1$).

Figure 1 shows *AF* over a line graph. The process begins with the node b and terminates at the ends of the graph and takes only 2 rounds, which is less than the diameter of the graph. Note that a line is an example of a bipartite graph. The triangle graph is another interesting illustrative example (Figure 5) – here, termination takes 3 rounds, whereas, the diameter is only 1. Note that the triangle is also the smallest clique and the smallest non-trivial cycle with odd number of nodes (an important topology for us). The even cycle is another interesting topology but here termination will happen in D rounds (as expected according to our bipartite graphs result).

Of course, a graph can have far more complicated topology with cyclic and acyclic subgraphs. For lack of space, we do not discuss more individual topologies in detail.

2 TERMINATION IN BIPARTITE GRAPHS

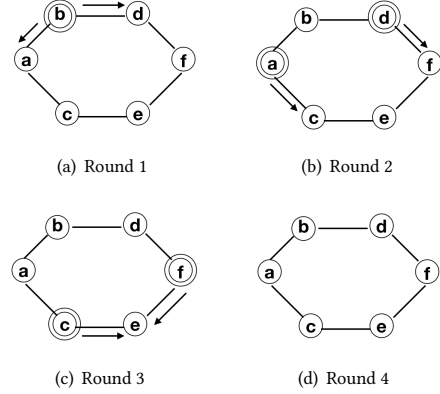


Figure 3: Termination in a bipartite graph (an even cycle) in diameter $D = 3$ time.

First, we show that *AF* terminates on a bipartite graph and, in fact, terminates, in the least time possible visiting each node exactly once and terminating in time equal to the eccentricity of the originating node. Figure 1 shows termination in a simple bipartite graph i.e. in a line. It is easy to see that *AF* terminates in 2 rounds which is the maximum distance from node b to any other node in the graph. Consider a more sophisticated example in Figure 3 - that of a cycle with six nodes. Here, *AF* from any originating node will terminate in diameter ($=3$) rounds. Since the eccentricity of nodes is upper bounded by the graph diameter, *AF* terminates in at most diameter rounds. Consider a connected bipartite graph. The following holds.

LEMMA 2.1. *In a connected bipartite graph B , Amnesiac flooding terminates in rounds $= e(a)$, where $e(a)$ is the eccentricity of the vertex a in graph B , where a is the node originating the process.*

Proof It is straightforward to see that amnesiac flooding executes a parallel BFS traversal from node a , which is the originating process. Nodes at a distance i from a receive the message at the same time in round i . All and only the nodes at distance $i + 1$ will now receive the message in round $i + 1$ since all the edges from distance- $(i - 1)$ nodes to distance- i nodes have been used and nodes at distance i do not share any edge since this is a bipartite graph.

The process terminates at the leaves of the BFS tree which have a maximum depth of $e(a)$. ■

Since the maximum eccentricity in a graph is its diameter, this corollary follows:

COROLLARY 2.2. *In a connected bipartite graph, the process terminates by round D , where D is the diameter of the graph.*

3 TERMINATION IN SYNCHRONOUS NETWORKS

The following theorem gives a proof of termination in general for synchronous networks. Note that Figure 3 illustrates AF in a triangle which is the smallest non-bipartite graph, showing termination, and in $2D + 1$ synchronous rounds where $D = 1$ is the diameter.

THEOREM 3.1. *Given a finite graph G , Amnesiac Flooding (AF) from a single source will terminate in a finite number of rounds.*

Proof

Let G be the network graph. We define *round-sets* R_0, R_1, \dots as:

- R_0 is the singleton containing the initial origin node,
- R_i is the set of nodes which receive a message at round i ($i \geq 1$).

Define \mathcal{R} to be the set of finite sequences of consecutive round-sets of the form:

$$\underline{R} = R_s, \dots, R_{s+d} \quad \text{where } s \geq 0, d > 0, \text{ and } R_s \cap R_{s+d} \neq \emptyset. \quad (1)$$

In (1), s is the *start-point* $s(\underline{R})$ and d is the *duration* $d(\underline{R})$ of \underline{R} . Note that, a node $x \in G$ belonging to R_s and R_{s+d} may also belong to other R_i in (1). Consider the subset \mathcal{R}^e of sequences in \mathcal{R} where d is even.

LEMMA 3.2. *AF is non-terminating only if \mathcal{R}^e is non-empty.*

Proof Since G is finite, some node $x \in G$ must occur in infinitely many round-sets R_i . If R_{i_1}, R_{i_2} and R_{i_3} are the first three round-sets that x occurs in, then the duration between R_{i_1} and R_{i_2} , R_{i_2} and R_{i_3} , or R_{i_1} and R_{i_3} will be even. ■

To prove termination, it suffices to prove that \mathcal{R}^e is empty as follows from Lemma 3.2. We assume that \mathcal{R}^e is non-empty and derive a contradiction. Let \mathcal{R}_{md}^e be the subset of \mathcal{R}^e comprising sequences of minimum (even) duration md , i.e.

$$\mathcal{R}_{md}^e = \{\underline{R} \in \mathcal{R}^e \mid \forall \underline{R}' \in \mathcal{R}^e. d(\underline{R}') \geq d(\underline{R}) = md\} \quad (2)$$

Clearly, if \mathcal{R}^e is non-empty then so is \mathcal{R}_{md}^e . Let $\underline{R}^* \in \mathcal{R}_{md}^e$ be the sequence with earliest start-point ms , i.e.

$$\underline{R}^* = R_{ms}, \dots, R_{ms+md} \quad (3)$$

where

$$\forall \underline{R}' \in \mathcal{R}_{md}^e. s(\underline{R}') \geq s(\underline{R}^*) = ms \quad (4)$$

By (1), there exists $x \in R_{ms} \cap R_{ms+md}$. Choose a node y which sends a message to x in round $ms + md$ (y must exist since $md > 0$). As y is a neighbour of x , either y sends a message to x in round ms or x sends a message to y in round $ms + 1$. Each of these cases leads to a contradiction:

Case (i): x receives a message from node y in round ms (Figure 4(a)) or *Case (ii): x sends a message to y in round $ms + 1$* (Figure 4(b))

For lack of space, we give an informal sketch for the rest of the proof. Refer to Figure 4(a) for case (i). Notice that node y has M in both rounds R_{ms-1} and $R_{ms+md-1}$. However, this is a contradiction since it shows a \underline{R}^* with an earlier starting point than ms . Now, consider case (ii) referring to Figure 4(b). Notice from the figure that node y has M in both R_{ms+1} and $R_{ms+md-1}$ - this shows a \underline{R}^* with a shorter duration $md - 2$. Again, this is a contradiction by our assumptions on \underline{R}^* . ■

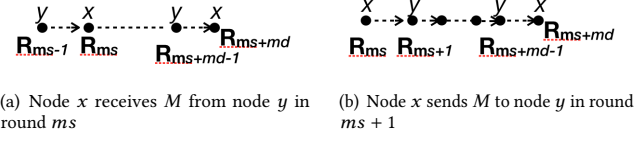


Figure 4: Termination in an arbitrary graph: two cases for a shortest even length sequence

We also show the following in the full paper:

THEOREM 3.3. *In a connected non-bipartite graph, the process terminates by round $2D + 1$, where D is the diameter of the graph.*

4 ASYNCHRONOUS AMNESIAC FLOODING

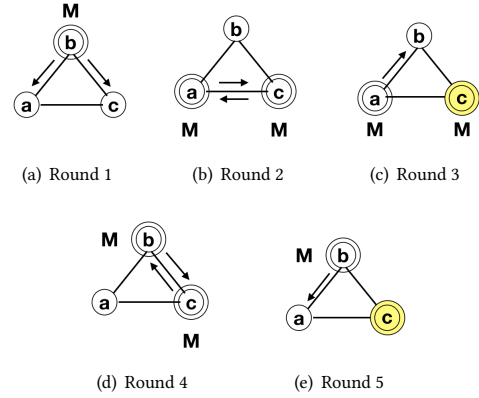


Figure 5: Asynchronous AF over a Triangle. Both node a and c send M to each other in round 2. In round 3, a sends M to b but the adversary makes c holds the message for one round (shaded node). In the next round, we have a round analogous to round 2 and so on.

Non-termination in an adversarial asynchronous setting: Informally, consider an asynchronous setting where a scheduling adversary can adaptively choose the delay on every message edge i.e. which round to forward a message on. Consider round 3 in the triangle in Figure 5. The adversary delays M at node c but a continues and sends to b . In round 4, node b and c both send M so that the beginning of the next round is now identical to round 2. This process can now continue *ad infinitum*.

REFERENCES

- [1] James Aspnes. 2019. Flooding. <http://www.cs.yale.edu/homes/aspnes/pinewiki/Flooding.html>
- [2] Hagit Attiya and Jennifer Welch. 2004. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons.
- [3] Benjamin Doerr, Mahmoud Fouz, and Tobias Friedrich. 2011. Social Networks Spread Rumors in Sublogarithmic Time. *Electronic Notes in Discrete Mathematics* 38 (2011), 303 – 308. The Sixth European Conference on Combinatorics, Graph Theory and Applications, EuroComb 2011.
- [4] Robert Elsässer and Thomas Sauerwald. 2008. The Power of Memory in Randomized Broadcasting. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '08)*. SIAM, Philadelphia, PA, USA, 218–227.