# Brief Announcement: Amnesiac Flooding: Easy to Break, Hard to Escape

Henry Austin
Department of Computer Science, Durham University
Durham, UK
henry.b.austin@durham.ac.uk

Maximilien Gadouleau
Department of Computer Science, Durham University
Durham, UK
m.r.gadouleau@durham.ac.uk

George B. Mertzios*
Department of Computer Science, Durham University
Durham, UK
george.mertzios@durham.ac.uk

Amitabh Trehan†
Department of Computer Science, Durham University
Durham, UK
amitabh.trehan@durham.ac.uk

## Abstract

Broadcast is a central problem in distributed computing. Recently, Hussak and Trehan [PODC'19/DC'23] proposed a stateless broadcasting protocol (Amnesiac Flooding), which was surprisingly proven to terminate in asymptotically optimal time (linear in the diameter of the network). However, it remains unclear: (i) Are there other stateless terminating broadcast algorithms with the desirable properties of Amnesiac Flooding, (ii) How robust is Amnesiac Flooding with respect to *faults*?

In this paper we make progress on both of these fronts. Under a reasonable restriction (obliviousness to message content) additional to the fault-free synchronous model, we prove that Amnesiac Flooding is the *only* strictly stateless deterministic protocol that can achieve terminating broadcast. We identify four natural properties of a terminating broadcast protocol that Amnesiac Flooding uniquely satisfies. In contrast, we prove that even minor relaxations of *any* of these four criteria allow the construction of other terminating broadcast protocols.

On the other hand, we prove that Amnesiac Flooding can become non-terminating or non-broadcasting, even if we allow just one node to drop a single message on a single edge in a single round. As a tool for proving this, we focus on the set of all *configurations* of transmissions between nodes in the network, and obtain a *dichotomy* characterizing the configurations, starting from which, Amnesiac Flooding terminates. Additionally, we characterise the structure of sets of Byzantine agents capable of forcing non-termination or non-broadcast of the protocol on arbitrary networks.

*Supported by the EPSRC grant EP/P020372/1.
†Supported by the EPSRC grant EP/P021247/1

A full version of this paper appears at [2].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*PODC '25, Huatulco, Mexico*
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1885-4/25/06
https://doi.org/10.1145/3732772.3733523

## CCS Concepts

• **Mathematics of computing** → *Discrete mathematics*; **Graph algorithms**; • **Theory of computation** → **Distributed algorithms**; **Graph algorithms analysis**.

## Keywords

Amnesiac flooding, Terminating protocol, Algorithm state, Stateless protocol, Flooding algorithm, Network algorithms, Graph theory, Termination, Communication, Broadcast.

**ACM Reference Format:**
Henry Austin, Maximilien Gadouleau, George B. Mertzios, and Amitabh Trehan. 2025. Brief Announcement: Amnesiac Flooding: Easy to Break, Hard to Escape. In *ACM Symposium on Principles of Distributed Computing (PODC '25), June 16–20, 2025, Huatulco, Mexico.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3732772.3733523

## 1 Introduction

The dissemination of information to disparate participants is a fundamental problem in both the construction and theory of distributed systems. A common strategy for solving this problem is to "broadcast", i.e. to transmit a piece of information initially held by one agent to all other agents in the system[1, 13, 15–17]. In fact, broadcast is not merely a fundamental communication primitive in many models, but also underlies solutions to other fundamental problems such as leader election and wake-up. Given this essential role in the operation of distributed computer systems and the volume of broadcasts, an important consideration is simplifying the algorithms and minimizing the overhead required for each broadcast [8].

Within a synchronous setting, *Amnesiac Flooding* as introduced by Hussak and Trehan in 2019 [9–11] eliminates the need to store historical messages. The algorithm terminates in asymptotically optimal $O(D)$ time (for $D$ the diameter of the network) and is stateless as agents are not required to hold any information between communication rounds. The algorithm in the fault-free synchronous message passing model is defined as follows:

Definition 1.1. ***Amnesiac flooding algorithm.*** *(adapted from [11]) Let $G = (V, E)$ be an undirected graph, with vertices $V$ and edges $E$ (representing a network where the vertices represent the nodes of the network and edges represent the connections between the nodes). Computation proceeds in synchronous 'rounds' where each* *round consists of nodes receiving messages sent from their neighbours. A receiving node then sends messages to some neighbours in the next round. No messages are lost in transit. The algorithm is defined by the following rules:*

(i) *All nodes from a subset of sources or initial nodes $I \subseteq V$ send a message $M$ to all of their neighbours in round 1.*
(ii) *In subsequent rounds, every node that received $M$ from a neighbour in the previous round, sends $M$ to all, and only, those nodes from which it did not receive $M$. Flooding* terminates *when $M$ is no longer sent to any node in the network.*

Extending Amnesiac Flooding and other stateless flooding algorithms (such as those proposed in [3, 18, 20]) beyond synchronous fault-free scenarios is challenging. This is due to the fragility of these algorithms and their inability to build in complex fault-tolerance due to the absence of state and longer term memory. It has been shown that no stateless flooding protocol terminates under moderate asynchrony, unless allowed to perpetually modify a super-constant number (i.e. $\omega(1)$) of bits in each message [18]. Yet, given the fundamental role of broadcast in distributed computing, the resilience of these protocols is extremely important even on synchronous networks.

Outside of a partial robustness to crash failures, the fault sensitivity of Amnesiac Flooding under synchrony has not been explored in the literature. This omission is further compounded by the use of Amnesiac Flooding as an underlying subroutine for the construction of other broadcast protocols. Multiple attempts have been made to extend Amnesiac Flooding to new settings (for example routing multiple concurrent broadcasts [3] or flooding networks without guaranteed edge availability [20]), while maintaining its desirable properties. However, none have been entirely successful, typically requiring some statefulness. It has not in fact been established that *any other* protocol can retain all of Amnesiac Flooding's remarkable properties even in its original setting. These gaps stem fundamentally from the currently limited knowledge of the dynamics of Amnesiac Flooding beyond the fact of its termination and its speed to do so. In particular, both of the existing techniques (parity arguments such as in [11] or auxiliary graph constructions such as in [19]) used to obtain termination results for Amnesiac Flooding are unable to consider faulty executions of the protocol and fail to capture the underlying structures driving terminating behaviour.

We address these gaps through the application of novel analysis and by considering the structural properties of Amnesiac Flooding directly. By investigating the sequence of message configurations, we are able to identify the structures underlying Amnesiac Flooding's termination and use these to reason about the algorithm in arbitrary configurations. The resulting dichotomy gives a comprehensive and structured understanding of termination in Amnesiac Flooding. For example, we apply this to investigate the sensitivity of Amnesiac Flooding with respect to several forms of fault and find it to be quite fragile. Furthermore, we show that under reasonable assumptions on the properties of a synchronous network, any strictly stateless deterministic terminating broadcast algorithm oblivious to the content of messages, must produce the exact same sequence of message configurations as Amnesiac Flooding on any network from any initiator. We therefore argue that Amnesiac Flooding is unique. However, we show that if any of these restrictions are relaxed, even slightly, distinct terminating broadcast algorithms can be obtained. As a result of this uniqueness and simplicity, we argue that Amnesiac Flooding represents a prototypical broadcast algorithm. This leaves open the natural question: do there exist fundamental stateless algorithms underlying solutions to other canonical distributed network problems? Though memory can be essential or naturally useful in certain scenarios [4–7, 12, 14], understanding what we can do with statelessness can help us push fundamental boundaries. A full version of this paper appears at [2], which should be referred to for the full formal statement of all definitions and results.

## 2 Model and Notation

Throughout this work we consider only finite, connected graphs on at least two nodes. We denote the set $\{1, ..., x\}$ by $[x]$ and $R(r, s)$ the Ramsey number such that any graph of size $R(r, s)$ contains either a clique on $r$ vertices or an independent set on $s$ vertices. In this work, we make use of a generic synchronous message passing model with several additional assumptions based on the truly stateless model of [18]. In particular, nodes cannot maintain any additional information between rounds (such as routing information, previous participation in the flood or even a clock value), cannot hold onto messages and can only forward, not modify the messages. For a graph $G = (V, E)$ and an *initiator* set $I \subseteq V$ we say that a node is informed if it has ever received a message from a previously informed node (where initiators are assumed to begin informed). An algorithm *correctly* solves broadcast (resp. multicast) on $G$ if for all singleton (resp. non-empty) initiator sets there exists a finite number of rounds after which all nodes will be informed. Unless specified otherwise, we assume that initiator nodes remain aware of their membership for only a single round. We say that an algorithm *terminates* on $G = (V, E)$ if, for all valid initiator sets, there exists a finite round after which no further messages are sent. We refer to a *configuration* of messages $S \subseteq \{(u, v) | uv \in E\}$ where $(u, v) \in S$ implies that in the current round $u$ sent a message to $v$.

## 3 Uniqueness

In this work, we investigate the existence of other protocols possessing the following four desirable properties of Amnesiac Flooding:

(1) *Strict Statelessness*: Nodes maintain no information other than their port labellings between rounds. This includes whether or not they were in the initiator set.
(2) *Obliviousness*: Routing decisions may not depend on the contents of received messages.
(3) *Determinism*: All decisions made by a node must be deterministic.
(4) *Unit Bandwidth*: Each node may send at most one message per edge per round.

Our main technical result regarding the existence of alternative protocols to Amnesiac Flooding is the following:

Theorem 3.1 (Uniqueness of Amnesiac Flooding). *Any terminating broadcast algorithm possessing all of* Strict Statelessness, Obliviousness, Determinism *and* Unit Bandwidth *behaves identically to Amnesiac Flooding on all graphs under all valid labellings.*

Note that this theorem allows, but does not require, that nodes have access to unique identifiers labelling themselves and their ports. However, we enforce the condition that these identifiers, should they exist, may be drawn adversarially from some super set of $[n+\kappa]$ where $n$ is the number of nodes on the network and $\kappa = R(9,8)$. Here $R(9,8)$ is the Ramsey number describing the smallest number of vertices such that a graph must have either a clique on at least 9 vertices or an independent set on 8 vertices.

Intuitively, the *Strict Statelessness* condition forces any broadcast protocol to make its forwarding decisions based only on the messages it receives in a given round. The combination of *Obliviousness* and *Unit Bandwidth* forces any protocol meeting the conditions to view messages as atomic. Finally, *Determinism* forces the protocol to make identical decisions every time it receives the same set of messages. Therefore, we can model the routing decisions of the protocol as a function from the set of ports it received a message over to the set of ports it will then send messages over.

It is important to stress here that this result holds even if the space of unique identifiers is only greater than $n$ by an additive constant. The proof is deferred to the full version, but has three key steps. The first is combinatorial. For any given protocol we derive a directed graph describing its behaviour and demonstrate via a forbidden subgraph argument that any set of IDs of size $R(x,8)$ must contain a subset of size at least $x$ that do not respond to each other as leaf nodes. The second leverages a set of 9 IDs with this property to establish that they behave identically to Amnesiac Flooding on all sub-cubic graphs labelled from only that set. The third and final step is a sequence of inductive arguments constructing topologies that use these 9 IDs to force all IDs to behave identically to Amnesiac Flooding for any set of neighbours.

The previous result is sharp, as we are able to obtain the following:

Theorem 3.2 (Existence of relaxed Algorithms). *There exist terminating broadcast algorithms which behave distinctly from Amnesiac Flooding on infinitely many networks possessing any three of:* Strict Statelessness*,* Obliviousness*,* Determinism *and* Unit Bandwidth.

We derive three relaxed algorithms which all build upon Amnesiac Flooding and illuminate the role of each of the four conditions in the uniqueness result.

- *Strict Statelessness*: Neighbourhood-2 Flooding. Nodes know the ID of their neighbours' neighbours. The protocol behaves distinctly on star graphs, as the hub can determine the entire graph topology.
- *Obliviousness* and *Unit Bandwidth*: 1-bit flooding. Nodes are allowed to send a single bit of read-only control information (in the message header or encoded in the number of messages sent) communicating whether the initiator is a leaf vertex. If it is, nodes implement Amnesiac Flooding, otherwise they use a different mechanism called Parrot Flooding (leaves bounce the message back) which always terminates when begun from a non-leaf vertex.
- *Determinism*: Random-Flooding. Nodes have access to one bit of randomness per round. Each round every node randomly chooses to implement Amnesiac Flooding or to forward to all neighbours. Random-Flooding achieves broadcast with certainty and terminates almost surely in finite time.

## 4 Fault Sensitivity

To complement our results on uniqueness, we also perform a comprehensive investigation of the fault sensitivity of Amnesiac Flooding in a synchronous setting. In order to achieve this (as well as to support the proofs of the previous section), we need to be able to determine its behaviour outside of correct broadcasts. Unfortunately, neither of the existing termination proofs naturally extend to the case of arbitrary message configurations. We make use of a method of invariants, to obtain much stronger characterizations of termination than were previously known, for both Amnesiac Flooding, and the subsequently proposed Stateless Flooding protocol [18]. In fact we obtain the following dichotomy:

Theorem 4.1. *For a graph $G=(V,E)$ and a configuration of messages $S \subseteq \{(u,v)|uv \in E\}$, the following are all equivalent:*

1. *Amnesiac Flooding terminates on $G$ when begun from $S$.*
2. *Amnesiac Flooding terminates on $G$ within $2|E|-3$ rounds when begun from $S$.*
3. *$S$ is obtainable via Amnesiac Flooding from some sequence of multi-casts on $G$.*
4. *$S$ is* balanced.

In Theorem 4.1, *balance* is a combinatorial property describing the distribution of messages around certain structures that can retain messages indefinitely (namely cycles and systems of connected odd cycles). The definition of balance and the proof of this result are quite involved and we defer them to the full version. Both centre on determining exactly which structures drive both termination and non-termination in Amnesiac Flooding dynamics and the role parity plays. The proof is independent of all previous work on Amnesiac Flooding and captures precisely the minimal set of structural elements that determine termination in Amnesiac Flooding, which is surprisingly determined entirely by systems of at most two cycles. The balance invariant may be of independent interest, beyond fault sensitivity, as they provide strong intuition for how asynchrony interferes with the termination of both Amnesiac Flooding and the Stateless Flooding proposed in [18]. The techniques are also generalisable to related processes, such as the Parrot-Flooding process mentioned in the previous section.

Theorem 4.1 allows us to provide precise characterizations of the behaviour of Amnesiac Flooding under the loss of single messages, uni-directional link failure, and time bounded Byzantine failures. Intuitively, these correspond to a set of messages failing to send in a specific round, a link failing in one direction creating a directed edge and a set of nodes becoming transiently controlled by an adversary. While we defer the statement of these results to the full version, we highlight the following specific consequences:

1. On any graph with any initiator vertex, there exists a single message the dropping of which will force either a failure to terminate or a failure to broadcast.
2. On any bipartite graph, the dropping of any message will force either a failure to broadcast or terminate.
3. Any set of time bounded Byzantine agents containing a non-leaf node can force either a failure to broadcast or a failure to terminate.

# References


[1] Hagit Attiya and Jennifer Welch. 2004. *Distributed Computing: Fundamentals, Simulations and Advanced Topics.* John Wiley & Sons.

[2] Henry Austin, Maximilien Gadouleau, George B. Mertzios, and Amitabh Trehan. 2025. Amnesiac Flooding: Easy to break, hard to escape. arXiv:2502.06001 [cs.DC] https://arxiv.org/abs/2502.06001

[3] Zahra Bayramzadeh, Ajay D Kshemkalyani, Anisur Rahaman Molla, and Gokarna Sharma. 2021. Weak amnesiac flooding of multiple messages. In *Networked Systems: 9th International Conference, NETYS 2021, Virtual Event, May 19–21, 2021, Proceedings.* Springer, 88–94.

[4] Petra Berenbrink, Robert Elsässer, and Thomas Sauerwald. 2014. Randomised broadcasting: Memory vs. randomness. *Theor. Comput. Sci.* 520 (2014), 27–42. https://doi.org/10.1016/J.TCS.2013.08.011

[5] Niccolò D'Archivio and Robin Vacus. 2024. On the Limits of Information Spread by Memory-Less Agents. In *38th International Symposium on Distributed Computing (DISC 2024) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 319)*, Dan Alistarh (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 18:1–18:21. https://doi.org/10.4230/LIPIcs.DISC.2024.18

[6] Benjamin Doerr, Mahmoud Fouz, and Tobias Friedrich. 2011. Social networks spread rumors in sublogarithmic time. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing* (San Jose, California, USA) *(STOC '11)*. Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/1993636.1993640

[7] 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.* Citeseer, 218–227.

[8] Ajei Gopal, Inder Gopal, and Shay Kutten. 1999. Fast broadcast in high-speed networks. *IEEE/ACM Transactions on Networking* 7, 2 (1999), 262–275.

[9] Walter Hussak and Amitabh Trehan. 2019. On Termination of a Flooding Process. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, Peter Robinson and Faith Ellen (Eds.). ACM, 153–155. https://doi.org/10.1145/3293611.3331586

[10] Walter Hussak and Amitabh Trehan. 2020. On the Termination of Flooding. In *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France (LIPIcs, Vol. 154)*, Christophe Paul and Markus Bläser (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 17:1–17:13. https://doi.org/10.4230/LIPICS.STACS.2020.17

[11] Walter Hussak and Amitabh Trehan. 2023. Termination of amnesiac flooding. *Distributed Computing* 36, 2 (2023), 193–207.

[12] Amos Korman and Robin Vacus. 2022. Early adapting to trends: Self-stabilizing information spread using passive communication. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing.* 235–245.

[13] N. Lynch. 1996. *Distributed Algorithms.* Morgan Kaufmann Publishers, San Mateo, CA.

[14] Garrett Parzych and Joshua J. Daymude. 2024. Memory Lower Bounds and Impossibility Results for Anonymous Dynamic Broadcast. In *38th International Symposium on Distributed Computing (DISC 2024) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 319)*, Dan Alistarh (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 35:1–35:18. https://doi.org/10.4230/LIPIcs.DISC.2024.35

[15] David Peleg. 2000. *Distributed Computing: A Locality Sensitive Approach.* SIAM.

[16] Andrew Tanenbaum. 2011. *Computer networks.* Pearson Prentice Hall, Boston.

[17] Gerard Tel. 1994. *Introduction to distributed algorithms.* Cambridge University Press, New York, NY, USA.

[18] Volker Turau. 2020. Stateless information dissemination algorithms. In *International Colloquium on Structural Information and Communication Complexity.* Springer, 183–199.

[19] Volker Turau. 2021. Amnesiac flooding: synchronous stateless information dissemination. In *SOFSEM 2021: Theory and Practice of Computer Science: 47th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM 2021, Bolzano-Bozen, Italy, January 25–29, 2021, Proceedings 47.* Springer, 59–73.

[20] Volker Turau. 2021. Synchronous concurrent broadcasts for intermittent channels with bounded capacities. In *International Colloquium on Structural Information and Communication Complexity.* Springer, 296–312.