Check for updates

Weak Amnesiac Flooding of Multiple Messages

Zahra Bayramzadeh[1], Ajay D. Kshemkalyani[2], Anisur Rahaman Molla[3], and Gokarna Sharma[1](✉)

[1] Kent State University, Kent, OH 44242, USA
zbayramz@kent.edu, sharma@cs.kent.edu
[2] University of Illinois at Chicago, Chicago, IL 60607, USA
ajay@uic.edu
[3] Indian Statistical Institute, Kolkata 700108, India
molla@isical.ac.in

Abstract. Flooding is a fundamental concept in distributed computing. In flooding, typically, a node forwards a message to its neighbors for the first time when it receives a message. Later if the node receives the same message again, it simply ignores the message and does not forward it. The nodes store a "message record" to ensure that the same message is not forwarded again.

Hussak and Trehan [STACS'20] introduced *amnesiac flooding* where nodes do not require to keep the message record. They established a surprising result that the amnesic flooding of a single ($k = 1$) message starting from some source node always terminates in bipartite graphs in e rounds and in non-bipartite graphs in $e < j \leq e + D + 1$ rounds, where e is the eccentricity of the source node and D is the diameter of the graph. Recently, Hussak and Trehan [arXiv'20] introduced *dynamic* amnesiac flooding initiated in possibly multiple rounds with possibly multiple ($k > 1$) messages from possibly multiple source nodes. They showed that the *partial-send* case where a node only sends a message to neighbours from which it did not receive *any* message in the previous round and the *ranked full-send* case where a node sends some highest ranked message to all neighbors from which it did not receive *that* message in the previous round, both terminate. However, they showed that the *unranked full-send* case, where a node sends some random message (not necessarily the highest ranked message) to all the neighbors from which it did not receive *that* message in the previous round, does not terminate.

In this paper, we show that the unranked full-send case also terminates, provided that diameter D is known to graph nodes. We further show that the termination time is $D \cdot (2k - 1)$ rounds in bipartite graphs and $(2D + 1) \cdot (2k - 1)$ rounds in non-bipartite graphs.

1 Introduction

Flooding is one of the fundamental and most useful primitives in distributed computing. In flooding, the task is to disseminate message(s) from source nodes

K. Echihabi and R. Meyer (Eds.): NETYS 2021, LNCS 12754, pp. 88–94, 2021.
https://doi.org/10.1007/978-3-030-91014-3_6

to all the nodes of the network. Suppose a distinguished source node has a message θ initially. The goal is to disseminate θ to all the nodes of the network. In a synchronous, round-based distributed network, flooding is typically performed as follows: In the first round, the distinguished source node sends θ to all its neighbors. From the next round onwards, when a node receives θ for the first time, it sends a copy of θ to its neighbors (except the neighbors from which it receives θ). If it receives θ again, it doesn't do anything. This essentially requires each node in the network to maintain a "message record" of θ to indicate whether that node has seen θ in some previous round. If a node receives θ and it has a record that it has seen θ before, then it does not forward θ. This ensures that the node never floods θ twice. It is well-known that this *classic* flooding process always terminates and the number of rounds until termination is $D+1$, the diameter of the network. The message record is of size at least 1 bit for a message.

Moving from single message flooding to multiple message flooding, the flooding approach for a single message has to be applied to each of the messages separately. Therefore, each node has to have the message record of at least 1 bit per message, i.e., $\Omega(k)$ bits for $k > 1$ messages, which may be a problem for resource-constrained devices [23,24].

Hussak and Trehan [11] asked an interesting question for the single message flooding starting from a distinguished source node: What will happen if nodes do not keep the record of the message θ? Will the flooding process still terminate? Not keeping a record means that message travels on its own without depending on a message record. Not having a message record simplifies client-server application design as well as makes it scalable due to the fact that servers do not need to keep track of session information [25]. It will also provide fault tolerance even when network nodes crash.

Intuitively, if the nodes do not keep any record, they may forward the message again and again when received in subsequent rounds. Thus, the absence of a message record raises the possibility that θ may be circulated infinitely. Hussak and Trehan [11] formally studied flooding without the message record, calling it *amnesiac flooding*, and showed that the single message ($k = 1$) flooding that starts from a distinguished source node terminates in bipartite graphs in e rounds and in non-bipartite graphs in $e < j \leq e + D + 1$ rounds, where e is eccentricity of the source node. Using two rounds to initiate flooding with the second round dependant on the first, termination time was improved to $e + 1$ rounds in any (non-bipartite) network by Turau [24], reducing the $e+D+1$ rounds of [11] by D rounds. However, the dependency on the first two rounds makes the result from Turau [24] not truly amnesiac compared to Hussak and Trehan [11]. Interestingly, the result of Turau [24] matches the termination time of classic flooding, since $e \leq D$, and the termination time of classic flooding is $D+1$ rounds. In the recent followup work, Hussak and Trehan [12] showed that the same termination time of e rounds in bipartite graphs and $e \leq j \leq e + D + 1$ rounds in non-bipartite graphs can be achieved for a single message θ starting from multiple source nodes concurrently. Essentially, Hussak and Trehan [12] showed that the proofs of [11] for the single source case carry over with simple modifications mainly to the definitions for the multiple source case. Turau [25] gave an alternative detailed proof.

Recently in [12], Hussak and Trehan considered *dynamic* amnesiac flooding of multiple $k > 1$ messages, where the messages may be initiated in possibly different rounds (i.e., not necessarily in the same first round) by different source nodes in the graph. Dynamic flooding arises in different real-world applications. One prominent example is disaster monitoring [25] where a distributed system of sensors is deployed to monitor a disaster event. As soon as sensors detect an event which may happen at different times for different sensors, they start flooding this information in the network. Furthermore, one source node may initiate multiple (different) messages (the source nodes may not be all different, i.e., $1 \leq k' < k$ source nodes for k messages). They considered the following three cases (problems) of dynamic amnesiac flooding in the synchronous message passing setting where each node receives messages from neighbors, performs internal computation, and sends messages to neighbors in synchronized rounds:

- *partial-send*: a node only sends a message to its neighbors from which it did not receive *any* message in the previous round.
- *ranked full-send*: a node sends some highest ranked message to all neighbors from which it did not receive *that* message in the previous round.
- *unranked full-send*: a node sends some random message (not necessarily the higest ranked message) to all neighbors from which it did not receive *that* message in the previous round.

Hussak and Trehan [12] showed that both the partial-send and ranked full-send problems terminate, but the unranked full-send problem does not terminate.

In this paper, we establish that the unranked full-send problem also terminates, provided that diameter D is known to network nodes. We further prove the termination time for the unranked full-send problem in both bipartite and non-bipartite graphs.

Overview of the Model and Results. Let the communication network be modeled as an undirected and unweighted but connected graph $G = (V, E)$, where V is the network nodes and $E \subseteq V \times V$ is the edges of G. Every node is assumed to have a unique identifier (e.g., its IP address). The nodes are allowed to communicate through the edges of the graph G. We consider a *synchronous message passing*[1] network, where computation proceeds in synchronous rounds with a node performing the following three tasks in each round: (i) receive messages from its neighbors, (ii) perform local computation, and (iii) send messages to its neighbors. No message is lost in transit. The messages are assumed to have unique IDs (which may not necessarily be consecutive and the smallest message ID may not be 1). A message θ is called globally i-th ranked if and only if the ID of θ is i-th largest among the IDs of all the messages in the set. The (global)

[1] In the asynchronous message passing framework, it was shown by Hussak and Trehan [11] that amnesiac flooding does not terminate.

rank of the messages is not known to graph nodes (i.e., the unranked problem), otherwise it becomes the ranked problem which terminates.

We prove the following theorem for the unranked full-send problem.

Theorem 1. (unranked full-send). *Given a set* $\{\theta^1, \ldots, \theta^k\}$ *of* $k > 1$ *messages positioned on* $1 \leq k' \leq k$ *nodes of a network* G *initiated at possibly different rounds, the unranked full-send problem terminates in bipartite graphs in* $D \cdot (2k - 1)$ *rounds and in non-bipartite graphs in* $(2D + 1) \cdot (2k - 1)$ *rounds*[2] *with each node storing* $O(\log(\max\{k, D\}))$ *bits, provided that the diameter* D *is known to the graph nodes.*

Theorem 1 is interesting and important since it was shown in Hussak and Trehan [12] that the unranked full-send problem does not terminate.

Comparison to Amnesiac and Classic Flooding. We first compare our result to amnesiac flooding and then to classic (non-amnesiac) flooding. Nodes do not need to store any information in the amnesiac flooding definition of Hussak and Trehan [11]. However, the assumption of graph nodes knowing D in our algorithm is a stronger condition than the amnesiac flooding definition of [11]. This is because knowing D requires each graph node to keep $\lceil \log D \rceil$ bits record in memory. Therefore, the storage requirement for any algorithm knowing D is at least $\Omega(\log D)$ bits. The total storage $O(\log(\max\{k, D\}))$ bits at each node in our algorithm is due to the fact that it also uses a *wait* variable which needs $O(\log k)$ bits. Therefore, our algorithm provides a trade-off between two parameters k and D regarding memory; $O(\log D)$ bits when $k = O(D)$ and $O(\log k)$ bits otherwise. Nodes need to store record of each message in classic (non-amnesiac) flooding, i.e., at least $\Omega(k)$ bits memory to flood k different messages. Therefore, the memory requirement in our algorithm is a significant reduction on the memory requirement at graph nodes compared to classic flooding when $k > \Omega(\log D)$.

The above comparison to amnesiac and classic flooding shows that our algorithm provides a '***weak***' variant of amnesiac flooding, that is, it reduces storage requirement of classic flooding but does not completely remove it as in amnesiac flooding [11,24]. An interesting direction for future research is whether a weaker assumption than D is enough to make the unranked full-send problem terminate. Finally, we prove the termination time of our algorithm using the single message termination time of [11]. One interesting property of our algorithm is that if a better termination time is available for the single message flooding, then the termination time improves proportionally.

Techniques. Suppose all messages are initiated in the beginning of round 1. Knowing D, the proposed algorithm asks messages to start their flooding process in the interval of $(2D + 1)$ rounds, i.e., at rounds $1, (2D + 1) + 1, 2 \cdot (2D + 1) + 1, \ldots, (k-1) \cdot (2D+1) + 1$. Suppose the source nodes of $k > 1$ messages $\theta^1, \ldots, \theta^k$

[2] If eccentricity $e_1, e_2, \ldots, e_{k'}$ of the k' source nodes is known instead of D, then the bounds translate to $e_{\max} \cdot (2k - 1)$ in bipartite graphs and $(2e_{\max} + 1) \cdot (2k - 1)$ in non-bipartite graphs with memory $O(\log(\max\{k, e_{\max}\}))$ bits, where $e_{\max} := \max_{1 \leq l \leq k'} e_l$.

know the rank (ID) of all the messages, say, $1, \ldots, k$, with message θ_i having rank i. Let us call this rank order as *global rank*. Knowing the global rank, θ^i can immediately decide how long to wait before starting the flooding process. Since it is known that a single message θ^i finishes flooding in $(2D + 1)$ rounds [11] ($e \leq D$), all k messages finish flooding by $k \cdot (2D + 1)$ rounds. That is, the ranked full-send problem terminates in $k \cdot (2D + 1)$ rounds.

The challenge to overcome is when the source nodes do not know the global rank of the messages (the unranked problem). We devise an algorithm that takes into account local ranks of the messages (i.e., the positions in the ranks of the messages at a node) in deciding the wait time for the messages. Except the globally lowest ranked message, the wait time assigned at round 1 may not be equal to its wait time knowing its global rank. The algorithm asks locally lowest ranked messages to start amnesiac flooding at round $(\kappa - 1) \cdot (2D + 1) + 1$, $\kappa \geq 1$ following the single message algorithm of Hussak and Trehan [11]. If the message that starts flooding at round $(\kappa - 1) \cdot (2D + 1) + 1$, $\kappa \geq 1$, is globally κ ranked, we show that it terminates by round $\kappa \cdot (2D + 1)$; otherwise during the round between $(\kappa - 1) \cdot (2D + 1) + 2$ and $\kappa \cdot (2D + 1) + 1$ (inclusive), it finds that its global rank is higher than κ and starts waiting increasing its wait time proportional to its local rank at that time. We will also show that the wait time update stops at round $(\kappa' - 1) \cdot (2D) + 1$ for the globally κ' ranked message. This altogether guarantees that the algorithm terminates in $k \cdot (2D + 1)$ rounds for $k > 1$ messages.

Finally, we show that this approach extends to the case of messages initiated at different rounds with termination time at most $(2k-1) \cdot (2D+1)$. For bipartite graphs, the only change is replacing $(2D+1)$ with D so that the bound becomes $(2k - 1) \cdot D$.

Related Work. Hussak and Trehan [11] were the first to consider amnesiac flooding. They showed that amnesiac flooding of a single message θ starting from a distinguished source node in the beginning of round 1 terminates in e rounds in bipartite graphs and in $e + D + 1$ rounds in non-bipartite graphs, $e \leq D$. They showed in [12] that this result also holds even when a single message θ starts flooding in the beginning of round 1 from multiple source nodes. In the asynchronous setting, they showed that amnesiac flooding does not terminate even for a single message starting from a source node. Recently, Hussak and Trehan [12] introduced dynamic amnesiac flooding initiated in multiple rounds by possibly multiple source nodes with possibly multiple messages. They showed that the partial-send and ranked full-send problems terminate but the unranked full-send problem does not terminate. In this paper, we show that the unranked full-send problem also terminates, provided that D is known.

Turau [24] improved the result of Hussak and Trehan [11] such that the amnesiac flooding terminates in $e+1$ rounds, even in non-bipartite graphs. This result is interesting since this termination time matches the classic flooding termination time of $D + 1$, since $e \leq D$. This result also applies to the single message starting flooding from multiple source nodes in the beginning of round 1. However, the assumption behind this result – the second round depending on the first – makes this result not truly amnesiac. Turau [24] also proved that the problem of selecting κ source nodes with minimal termination time is NP-hard. Particularly, Turau showed that unless NP = P there is no approximation algorithm for amnesiac flooding with approximation ratio $3/2 - \epsilon$. For asynchronous systems, Turau proved that deterministic amnesiac flooding is only possible if a large enough part of the message can be updated by each node. Very recently, Turau [25] provided an alternative detailed proof for the single message flooding starting from multiple source nodes in the beginning of round 1. Specifically, Turau showed that, for every non-bipartite graph G and every set V' of source nodes that start flooding simultaneously, there exists a bipartite graph $G(V')$ such that the execution of amnesiac flooding on both graphs G and $G(V')$ is strongly correlated and termination times coincide. This led to bounds that are independent of the diameter as well as it allowed to determine source nodes for which amnesiac flooding terminates in minimal time. Turau also gave tight lower and upper bounds for the time complexity in special cases of $|V'| = 1$ and $|V'| > 1$. In fact, the case of $|V'| > 1$ was reduced to the case of $|V'| = 1$.

Flooding is a fundamental concept used in solving a diverse set of fundamental problems in distributed computing, e.g., leader election [14,15], spanning tree construction [2,13,16,17,21], shortest paths computation [9,10,20], aggregation [5], routing [18], etc. Flooding of multiple messages is a must in many distributed applications, e.g., k-information dissemination or gossiping [1,3–5,7,16,17,19,22].

Amnesiac flooding uses the most recent edges from which the message is received to a node to decide which neighboring edges of that node are used to flood the message from that node. This concept finds applications and uses in social networks [6], broadcasting [8], and client-server application design [25]. More details in [11,12,23–25].

References

1. Ahmadi, M., Kuhn, F., Kutten, S., Molla, A.R., Pandurangan, G.: The communication cost of information spreading in dynamic networks. In: Proceedings of 39th IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 368–378 (2019)
2. Attiya, H., Welch, J.L.: Distributed Computing - Fundamentals, Simulations, and Advanced Topics. Wiley Series on Parallel and Distributed Computing, 2nd edn. Wiley, Hoboken (2004)
3. Borokhovich, M., Avin, C., Lotker, Z.: Tight bounds for algebraic gossip on graphs. In: IEEE International Symposium on Information Theory, ISIT 2010, Austin, Texas, USA, 13–18 June 2010, Proceedings, pp. 1758–1762 (2010)
4. Boyd, S.P., Ghosh, A., Prabhakar, B., Shah, D.: Randomized gossip algorithms. IEEE Trans. Inf. Theory **52**(6), 2508–2530 (2006)
5. Chen, J., Pandurangan, G.: Almost-optimal gossip-based aggregate computation. SIAM J. Comput. **41**(3), 455–483 (2012)

6. Doerr, B., Fouz, M., Friedrich, T.: Social networks spread rumors in sublogarithmic time. In: Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing, STOC '11, pp. 21–30. Association for Computing Machinery, New York (2011). https://doi.org/10.1145/1993636.1993640
7. Dutta, C., Pandurangan, G., Rajaraman, R., Sun, Z., Viola, E.: On the complexity of information spreading in dynamic networks. In: SODA, pp. 717–736 (2013)
8. Elsässer, R., Sauerwald, T.: The power of memory in randomized broadcasting. In: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 218–227 (2008)
9. Ghaffari, M., Li, J.: Improved distributed algorithms for exact shortest paths. In: STOC, pp. 431–444 (2018)
10. Holzer, S., Wattenhofer, R.: Optimal distributed all pairs shortest paths and applications. In: PODC, pp. 355–364 (2012)
11. Hussak, W., Trehan, A.: On the termination of flooding. In: STACS, pp. 17:1–17:13 (2020)
12. Hussak, W., Trehan, A.: Terminating cases of flooding. CoRR abs/2009.05776 (2020). https://arxiv.org/abs/2009.05776
13. Kshemkalyani, A.D., Singhal, M.: Distributed Computing: Principles, Algorithms, and Systems. Cambridge University Press, Cambridge (2011)
14. Kutten, S., Pandurangan, G., Peleg, D., Robinson, P., Trehan, A.: On the complexity of universal leader election. J. ACM **62**(1), 7:1–7:27 (2015). https://doi.org/10.1145/2699440
15. Kutten, S., Pandurangan, G., Peleg, D., Robinson, P., Trehan, A.: Sublinear bounds for randomized leader election. Theor. Comput. Sci. **561**, 134–143 (2015). https://doi.org/10.1016/j.tcs.2014.02.009
16. Lynch, N.A.: Distributed Algorithms. Morgan Kaufmann Publishers Inc., San Francisco (1996)
17. Peleg, D.: Distributed Computing: A Locality-Sensitive Approach. Society for Industrial and Applied Mathematics, USA (2000)
18. Rahman, A., Olesinski, W., Gburzynski, P.: Controlled flooding in wireless ad-hoc networks. In: International Workshop on Wireless Ad-Hoc Networks, pp. 73–78 (2004)
19. Sarma, A.D., Molla, A.R., Pandurangan, G.: Distributed computation in dynamic networks via random walks. Theor. Comput. Sci. **581**, 45–66 (2015)
20. Tanenbaum, A.S., Wetherall, D.J.: Computer Networks, 5th edn. Prentice Hall Press, Hoboken (2010)
21. Tel, G.: Introduction to Distributed Algorithms, 2nd edn. Cambridge University Press, Cambridge (2001)
22. Topkis, D.M.: Concurrent broadcast for information dissemination. IEEE Trans. Softw. Eng. **11**(10), 1107–1112 (1985)
23. Turau, V.: Analysis of amnesiac flooding. CoRR abs/2002.10752 (2020). https://arxiv.org/abs/2002.10752
24. Turau, V.: Stateless information dissemination algorithms. In: Richa, A.W., Scheideler, C. (eds.) SIROCCO 2020. LNCS, vol. 12156, pp. 183–199. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-54921-3_11
25. Turau, V., et al.: Amnesiac flooding: synchronous stateless information dissemination. In: Bureš, T. (ed.) SOFSEM 2021. LNCS, vol. 12607, pp. 59–73. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67731-2_5