Connectivity recovery in epidemic membership protocols

Proceedings of the 8th International Conference on Internet and Distributed Computing Systems (IDCS), Windsor, 2-4 Sept. 2015, Springer. (in press)

Connectivity Recovery in Epidemic Membership Protocols

Pasu Poonpakdee and Giuseppe Di Fatta

School of Systems Engineering, University of Reading, Whiteknights, Reading, Berkshire, RG6 6AY, United Kingdom {p.poonpakdee,g.difatta}@reading.ac.uk

Abstract. *Epidemic protocols are a bio-inspired communication and computation paradigm for extreme-scale network system based on randomized communication. The protocols rely on a membership service to build decentralized and random overlay topologies. In a weakly connected overlay topology, a naive mechanism of membership protocols can break the connectivity, thus impairing the accuracy of the application. This work investigates the factors in membership protocols that cause the loss of global connectivity and introduces the first topology connectivity recovery mechanism. The mechanism is integrated into the* Expander Membership Protocol*, which is then evaluated against other membership protocols. The analysis shows that the proposed connectivity recovery mechanism is effective in preserving topology connectivity and also helps to improve the application performance in terms of convergence speed.*

Keywords: topology connectivity; expander graphs; epidemic protocols; extreme-scale computing; decentralized algorithms

1 Introduction

In extreme-scale networked systems, the decentralized computation of aggregation functions is an interesting and challenging task. Due to problems such as communication bottlenecks and fault intolerance, centralized paradigms are not desirable solutions. Epidemic, or Gossip-based, protocols are fully decentralized and fault tolerant, which are particularly suitable for information dissemination and global aggregation tasks.

Applications based on epidemic protocols for large and extreme-scale networked systems are emerging in many fields, including Peer-to-Peer (P2P) overlay networks [1], distributed computing [2], mobile ad hoc networks (MANET) [3], wireless sensor networks (WSN) [4] and exascale high performance computing [5, 6].

Epidemic protocols use a randomised communication paradigm, which is the foundation for their robustness and scalability. In order to perform randomised communication, a peer sampling service is required, which is considered a fundamental network service. Obviously, maintaining global knowledge, i.e. a complete list of nodes, is not a feasible approach in very large and extreme-scale distributed systems. A Membership Protocol is typically employed to provide this service.

Scalable and fault tolerant membership protocols can be implemented with an epidemic approach. The aim of membership protocols is to provide a node selection service that returns a random node with uniform probability, similar to a random selection from the global view of the system [7]. Instead of maintaining a complete list of nodes at each node, a membership protocol builds a local partial view (cache) of the system. The local view is continuously and randomly changed: the local partial membership information at each nodes is disseminated and mixed by exchanging messages with random peers.

Several membership protocols ([8–11]) have been proposed, which have been designed for generating random overlay topologies.

In particular, the *Expander Membership Protocol* (EMP) [11] is inspired by the expansion property of a graph, which is a fundamental mathematical concept [12]. EMP is based on a push-pull scheme that introduces a bias in the random node selection in order to maximise the expansion property of the overlay topology. To this aim EMP employs a push-forwarding procedure (a random walk) to search and select an ideal communication peer (quasi-random gossiping) for the exchange of information.

Topology connectivity is a fundamental property of the overlay graphs that is required to guarantee the accuracy of epidemic protocols and their applications. To the best of our knowledge, many approaches have focused on mechanisms aimed at preserving the topology connectivity in strongly connected graphs, while none has been dedicated at recovering the connectivity when lost. In fact, this is an important problem for applications deployed in real-world distributed systems, where overlay graphs may be weakly connected and the global connectivity be lost in spite of the best effort in trying to preserve it.

In this work, the first mechanism that addresses the connectivity problem in weakly connected graphs is introduced. The novel mechanism, the *Interleaving Management Procedure* (IMP), is integrated into EMP in order to recover from the degradation of the overlay topology from a single connected component to multiple connected components. The enhanced version of EMP will be referred to as EMP+.

The rest of the paper is organized as follows. In section 2, details of the connectivity problem in epidemic membership protocols are given. Section 3 presents a brief description of EMP and the message interleaving event. Section 4 introduces a novel procedure that addresses the message interleaving problem to avoid and to recover from the loss of topology connectivity. Section 5 presents the integration of this procedure into EMP. Simulations and experimental results are presented in section 6. Finally, section 7 draws some conclusions and provides the direction of future work.

2 Connectivity Problems

Graph connectivity is a fundamental concept of graph theory which is also applied to overlay topology in Epidemic Protocols. For example, epidemic aggregation protocols are employed to compute local estimations of a global aggregation

function: such estimations can converge to the true target value if and only if the global topology connectivity is preserved.

There are several reasons that may cause the degradation of overlay topologies, e.g. external causes may be node churn and node failures [11]. Surprisingly, in a weakly connected overlay topology, the internal mechanisms of a membership protocol can also turn the overlay topology from a single connected component into multiple connected components. This section briefly reviews some membership protocols and identifies their components that may introduce such connectivity problems.

The Node Cache Protocol [8] is a simple membership protocol that adopts a symmetric push-pull mechanism to exchange and shuffle local membership information (node cache). At each node, the protocol contains a local cache Q of node identifiers, where $|Q| = q_{max}$ is the maximum local cache size of each node (this parameter is applied to all membership protocols used in this work). At each cycle, the local cache is sent with a push message to a node randomly selected from the local cache. When a push message is received, the local cache is sent in a reply (pull message) to the remote node originating the push message. The local cache is merged with the remote cache and the remote node ID (refreshing mechanism). The local cache is finally trimmed to q_{max} entries by randomly eliminating the number of entries exceeding q_{max}. In the Node Cache Protocol, the trimming operation is the component that may cause connectivity problems, because the removed entries could be the single link between two connected components in an overlay topology with weak connectivity.

Cyclon [9] is a membership protocol that is an enhanced version of a basic node cache shuffling. The mechanism of Cyclon is similar to the Node Cache Protocol, which also adopts a push-pull mechanism. In Cyclon, cache entries are assigned an attribute age to track their lifetime. At each cycle, a number of entries randomly selected from the local cache are sent (push message) to the node corresponding to the oldest entry in the local cache. When the push message is received, the node replies with a pull message containing entries a number of randomly selected entries from its local cache. The received entries are used to replace the donated entries at both ends. Connectivity problems in Cyclon may arise when there is message interleaving between independent pairs of push-pull exchanges involving the same node. Message interleaving has been identified as a potential threat to the accuracy of those epidemic aggregation protocols [8] that would require atomic push-pull operations. Similarly, in Cyclon message interleaving introduces the risk of removing critical cache entries, as the atomicity of the push-pull operation for cache exchange is not guaranteed.

Eddy [10] is arguably the most complex membership protocol. In order to provide a better random distribution of node samples in the system, Eddy tries to minimize temporal and spatial dependencies between local caches. The mechanism in Eddy can be separated into two independent processes: gossiping and refreshing. Gossiping is based on a symmetric push-pull operation: when the entries in the local cache are chosen for an exchange, they are also removed from the local cache. Refreshing adopts entry lifetime and push-forwarding mechanism. A

limited lifetime is assigned to each entry which is removed when expires. Expired entries are replaced with fresh entries by forwarding the entry to a random node within two hops. The refreshing process of Eddy is effective, however it introduces a significant communication overhead and an entry removal mechanism that can cause connectivity problems.

3 The Expander Membership Protocol

The Expander Membership Protocol (EMP) ([11]) directly employs the concept of expansion in graphs. Expander graphs are sparse graphs with strong connectivity properties. In general, a graph is an expander if any vertex subset (not too large) has a relatively large set of one-hop distant neighbours.

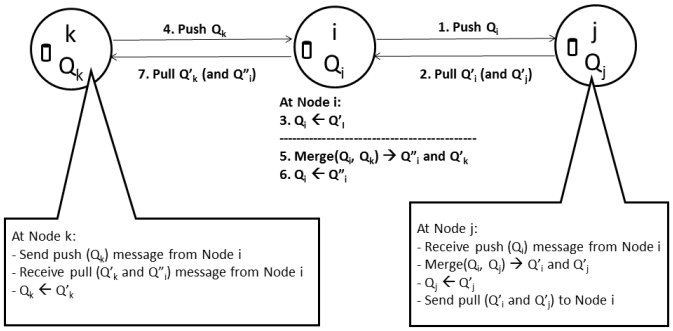
The typical cache shuffling mechanism in membership protocols is used to generate a continuous series of random overlay topologies that are sparse and have a strong connectivity. This task is particularly easy when the initial graph is already an expander, e.g. a random regular graph. The speed of transformation from an initial topology to an expander can be used as an indicator to evaluate the performance of membership protocols, which also affects the convergence speed of the application. EMP is inspired by these considerations and built on the concept of vertex expansion. The protocol adopts a symmetric push-pull mechanism and a push-forwarding mechanism. EMP adopts a random walk in order to search for a better communication partner (quasi-random gossiping).

4 Message Interleaving

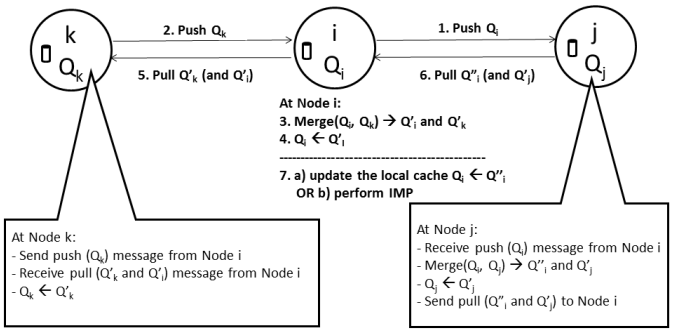
Applications in real-world distributed systems have to cope with asynchronous communication and network latency. As a result of that, in epidemic protocols there is a possibility that some node is receiving a pull message while it is waiting for a pull message. In weakly connected overlay topologies, this message interleaving can harm the global connectivity of the system.

To describe the effect of message interleaving, it is useful to compare two scenarios with and without message interleaving. The first scenario (figure 1(a)) considers three nodes (i, j and k), which are exchanging their membership information without interleaving. First node i sends a push message to node j; then node i receives a pull message from node j. Eventually node k sends a push message to node i and, finally, node i sends a pull message to node k. In this scenario, the two independent push-pull operations happen in the expected sequence without message interleaving. Let Q_i be the local cache at node i, the sequence of events at node i are as follows:

1. Node i sends a push message (Q_i) to node j.
2. Node i receives a pull message (Q'_i and Q'_j) from node j.
3. Node i updates the local cache $Q_i \leftarrow Q'_i$.
4. Node i receives a push message (Q_k) from node k.
5. Node i merges Q'_i and Q_k and generates two partitions Q''_i and Q'_k.



(a) without message interleaving



(b) with message interleaving

Fig. 1. The scenarios of message transmission with and without message interleaving

6. Node i updates the local cache $Q'_i \leftarrow Q''_i$.
7. Node i sends a pull message (Q'_k and Q''_i) to node k.

The second scenario (figure 1(b)) is similar to the first scenario, except that this time the two push-pull operations are overlapped and message interleaving happens. Node i sends a push message to node j. Before node i receives a pull message from node j, it receives a push message from node k. The sequence of events at node i are as follows:

1. Node i sends a push message (Q_i) to node j.
2. Node i receives a push message (Q_k) from node k.
3. Node i merges Q_i and Q_k and generates two partitions Q'_i and Q'_k.
4. Node i updates the local cache $Q_i \leftarrow Q'_i$.
5. Node i sends a pull message (Q'_i and Q'_k) to node k.
6. Node i receives a pull message (Q''_i and Q'_j) from node j.
7.
   (a) Node i updates the local cache $Q_i \leftarrow Q''_i$.
   (b) **OR** Node i detects message interleaving and performs IMP.

After step 6, as a result of message interleaving, the three local caches have been updated (Q'_i, Q'_k, Q'_j) and an additional buffer (Q''_i) has been received at node i. In step 7, if node i would perform the simple update operation of 7(a), as in the case without message interleaving, there would be a risk of removing critical links, thus exposing the system to potential connectivity problems. Alternatively, node i can actually detect the message interleaving event and a more complex operation, the Interleaving Management Procedure (IMP) of 7(b), has to be performed to process the incoming pull message.

Figure 2 is a snapshot of the cache configuration after step 6, when node i detects the message interleaving event. The problem in this configuration is that some duplicated cache entries have been generated by the two merging operations. The total number of cache entries in the system is bounded by $N * q_{max}$, where N is the number of nodes. When a cache entry duplicate is generated, another cache entry needs to be discarded in order to accommodate the duplicate, introducing the risk that a critical link is removed from the system. Duplicated entries also negatively affect the node outdegree distribution in the system. Duplicated entries must be detected and eliminated when possible.

The next section describes the procedure used to detect and discard duplicated cache entries.

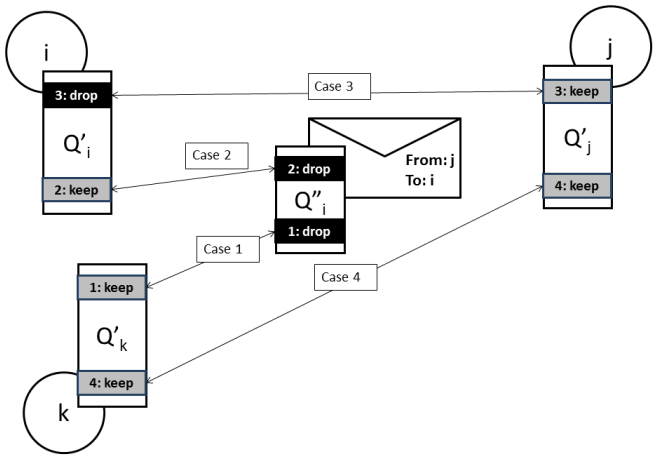


Fig. 2. The snapshot of cache configuration when node i detects a message interleaving event

4.1 Detection of Cache Entry Duplicates

The aim of this procedure is to identify and eliminate the duplicated entries generated by the message interleaving event. Let us assume that the initial local

caches (Q_i, Q_j and Q_k) of the three nodes in the last scenario do not share any entry, i.e. $Q_i \cap Q_j = \varnothing$, $Q_i \cap Q_k = \varnothing$ and $Q_j \cap Q_k = \varnothing$.

The duplicates have been generated because the push-pull operation between i and j has not been performed atomically: the merge and partition operation on Q_i and Q_j has generated Q'_j and Q''_i, while node i has changed its local cache.

Figure 2 shows the duplicated portions of the four cache buffers in the scenario when node i detects the interleaving event (after step 6).

In order to identify and remove duplicated entries, node i requires the information of the four partitions generated by the two merging operations: Q'_i, Q'_k, Q''_i and Q'_j. While the first three are locally available to node i, the buffer Q'_j needs to be included in the pull message from node j.

The duplicated entries can be detected by comparing these four buffers, for a total of $\binom{4}{2} = 6$ possible combinations. Two combinations cannot generate duplicates, as $Q'_i \cap Q'_k = \varnothing$, $Q'_j \cap Q''_i = \varnothing$. Four groups of potential duplicates can be identified and are shown in figure 2: 1) $Q''_i \cap Q'_k$, 2) $Q'_i \cap Q''_i$, 3) $Q'_i \cap Q'_j$, 4) $Q'_k \cap Q'_j$.

In the figure some buffer subsets have been indicated as 'drop' and others as 'keep': node i can identify and drop the duplicates of three cases. However, in case 4 node i cannot take any action to identify and remove the duplicates, nor can node j and node k.

This analysis has inspired a mechanism to detect message interleaving and perform a procedure to remove most, but not all, duplicates. This procedure will reduce the likelihood of connectivity problems, but will not eliminate the risk completely (case 4). For this reason a mechanism for connectivity recovery is still required.

The procedure to remove duplicated cache entries, the one for connectivity recovery and their integration in EMP are described in the next section.

5 The Enhanced Expander Membership Protocol

In the previous section it was shown how the number of potential duplicated cache entries can be reduced, thus minimising the negative effect in the degree distribution of the overlay topology and, more importantly reducing the risk of connectivity problems. However, the latter issue cannot be eliminated completely. This problem may not be likely in graphs with strong connectivity. However, in weakly connected overlay topologies, such as ring of communities [11], discarding even a few critical cache entries may result in the loss of global connectivity.

For this reason, an enhancement of EMP, EMP+, is introduced in this section, which is the integration of IMP into EMP. The goal of EMP+ is maintaining a global connectivity, while still supporting a good convergence speed of the applications.

The proposed method for solving the connectivity problem adopts a reserve cache, which is used to store the entries that are removed during the merging operation. To avoid that the size of the reserve cache may grow indefinitely, a

Table 1. Notation adopted in the EMP+ pseudocode

i	a node in the network, $i \in V$, where V is the set of nodes
Q_i	main cache at node i, $\|Q_i\| \leq q_{max}$
R_i	reserve cache at node i
H_i	history cache at node i
h_{max}	maximum number of hops in the random walk
r_{max}	maximum reserve cache size
$m_{\rightarrow}$	push message with payload:
	- s, node originating the push
	- Q_s, main cache at s
	- d, current best destination node
	- v_d , current minimum overlap
	- h, hop count
$m_{\leftarrow}$	pull message with payload:
	- d, node originating the pull
	- Q, set of donated cache entries
	- Q_d, main cache at node d

maximum reserve cache size (r_{max}) is enforced. When the size of the reserve cache has reached the maximum, some entries must be discarded.

It is also necessary to store the entries donated with a pull message and a history cache is introduced for this purpose (line 16). In this cache, entries are associated with a maximum lifetime and when it expires the entry is removed to avoid that the cache may grow indefinitely.

The notation adopted in the following pseudocode is summarised in table 1.

Algorithm 1 shows the pseudocode of EMP+, which is based on the pseudocode of EMP [11]. Here, the novel components of the protocol EMP+ are highlighted and discussed.

The main difference between EMP and EMP+ is the utilization of the reserve and history caches. Lines 10 and 11 describe how entries from the reserve cache are used during the merge operation to generate the two disjoint partitions Q_1 and Q_2. Lines from 28 to 34 describe the procedure to process incoming pull messages with and without message interleaving. If there is no message interleaving, the local cache can be immediately updated with the content of the message. When message interleaving occurs, the procedure IMP is performed.

Algorithm 2 shows the pseudocode of the Interleaving Management Procedure, which performs the removal of the duplicated entries. Lines from 2 to 7 show how the duplicates from the cases 1, 2 and 3 are detected and discarded. IMP is the only procedure in EMP+ that inserts entries into the reserve cache as shows in line 10. The reserve cache must contain unique entries for the same node ID at any time.

Algorithm 1 EMP+

```
1: procedure SendPushMessage
2:     remove expired entries from H_i
3:     j ← get the oldest node from Q_i
4:     send a push message to j :m→(s = i, Q_s = Q_i, d = null, v_d = ∞, h = 0)
5:
6: procedure ReceivePushMessage( message m→)
7:     compute total cache size v = |Q_i ∪ m→.Q_s ∪ R_i|
8:     if (v + 1 ≥ 2 * q_max) or (m→.h > h_max) then
9:         Q_m ← Q_i ∪ m→.Q_s
10:        while (|Q_m| + 1 < 2 * q_max) and (|R_i| > 0) do
11:            insert entry from R_i into Q_m
12:        while (|Q_m| + 1 < 2 * q_max) do
13:            insert random entry into Q_m
14:        randomly partition Q_m into Q_1 and Q_2(|Q_1 ∪ {m→.s}| = |Q_2| = q_max)
15:        update local main cache: Q_i ← Q_1 ∪ {m→.s}
16:        add the donated cache entries to the history cache: H_i ← H_i ∪ Q_2
17:        send a pull message to m→.s :m←(d = i, Q = Q_2, Q_d = Q_i)
18:    else if (m→.h < h_max) then
19:        if v < m→.v_d then
20:            set m→.d = i and m→.v_d = v
21:        select random node j from Q_i
22:        m→.h + +
23:        send m→ to j
24:    else if (m→.h == h_max) then
25:        m→.h + +
26:        send m→ to m→.d
27:
28: procedure ReceivePullMessage( message m←)
29:    if message interleaving == false then
30:        update local main cache: Q_i ← m←.Q
31:    else
32:        perform IMP(m←)
33:    while |R_i| > r_max do
34:        remove the oldest entry in R_i
```

6 Experimental Analysis

The goal of this experimental analysis is to evaluate the proposed protocol EMP+ and to compare it against other membership protocols. The analysis is based on simulations, which are used to verify the global connectivity of the overlay topology and to measure the performance of an application when different membership protocols are used.

When membership protocols are executed over a random overlay topology, all of them seem to provide an optimal peer sampling service with respect to the convergence speed of a global aggregation. However, when the overlay topology is not random, membership protocols may induce different results on the application performance. This may happen, for example, when the overlay topology is initialised or when high node churn is present. Rather than studying optimal initialization procedure for the overlay topology, in this work we evaluate the performance of different membership protocols when the overlay topology has weak connectivity.

In past related work, the initial overlay topology is often chosen as a random regular graph. This is an arbitrary and unrealistic choice, which makes the overlay topology very robust to the loss of connectivity. On the contrary, the

Algorithm 2 IMP

```
1:  procedure IMP( message m_←)
2:      D_1 ← m_←.Q ∩ H_i, detect and remove the duplication from case 1
3:      m_←.Q ← m_←.Q \ D_1
4:      D_2 ← m_←.Q ∩ Q_i, detect and remove the duplication from case 2
5:      m_←.Q ← m_←.Q \ D_2
6:      D_3 ← Q_i ∩ m_←.Q_d, detect and remove the duplication from case 3
7:      Q_i ← Q_i \ D_3
8:      T ← Q_i ∪ m_←.Q, create a temporary cache
9:      while |T| > q_max do
10:         remove the oldest entry in T and add it to R_i
11:     while |T| < q_max and |R_i| > 0 do
12:         remove the oldest entry in R_i and add it to T
13:     while |T| < q_max do
14:         select a random entry from the duplications and add it to T
15:     update local main cache: Q_i ← T
```

simulations carried out for this work, have used initial overlay topologies with weak connectivity in order to show the effect of membership protocols in the degradation of the topology. A ring of communities [11] is an artificial topology with poor expansion property and a good load balance: it has been used as initial overlay topology in the simulations.

The experimental tests have been carried out in PeerSim [13], a Java-based network simulation based on discrete events. The simulations have adopted an asynchronous network model with a uniform distribution of network latency. The simulations have been run with the following membership protocols, where their settings have been chosen for best performance according to the literature and to a preliminary analysis.

- The Node Cache Protocol [8] ($q_{max} = 30$).
- Cyclon [9] ($q_{max} = 30$ and shuffle length = 15).
- Eddy [10] (shuffle length = 15 and refresh rate = 10 cycles).
- EMP [11] ($q_{max} = 30$ and $h_{max} = 5$).
- EMP+ ($q_{max} = 30$, $h_{max} = 5$, $r_{max} = 100$ and history lifetime = 2 cycles).

The initial local cache in all membership protocols was populated with 30 entries according to the same initial overlay topology. The initial ring of communities topology is generated with 10 random connected communities of 1000 nodes: there are only two links between each pair of communities which make them to have weak inter-community connectivity.

6.1 Global Connectivity

In the first set of simulations, each protocols was run for 100 cycles starting from the initial ring of communities topology. The aim of the simulations is to collect information about the number of connected components in the topology to detect any loss of global connectivity. Each simulation was repeated 20 times with a different seed of the random number generator.

Figure 3 shows the maximum number of connected components over the 20 trials. It shows that all the membership protocols have lost global connectivity

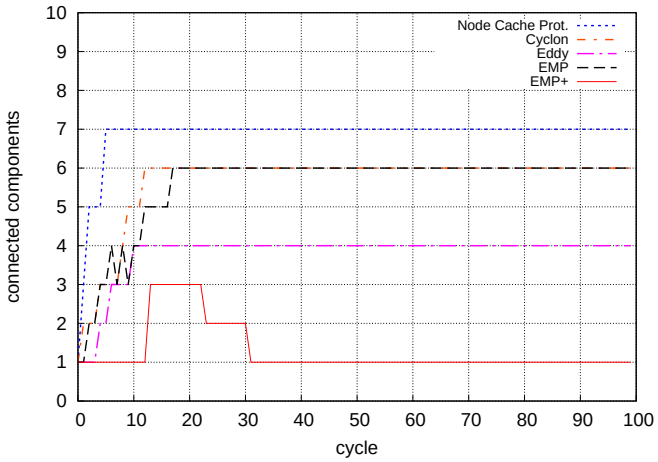


Fig. 3. The number of connected components in the overlay topology: maximum over 20 trials. (network size: 10000 nodes)

in at least one trial and at some point in time. For all protocols but EMP+ a connectivity problem is irreversible. The Node Cache Protocol produced the highest number of connected components. Between cycle 5 and cycle 10, the number of connected components for EMP is unstable because of the effect of message interleaving. Like EMP, EMP+ also has a loss of connectivity, though only temporarily. EMP+ is the only membership protocol that adopts IMP to be able to recover the global connectivity when lost.

The results of our analysis have shown that IMP is often effective in recovering the global connectivity of the overlay topology. However, it does not provide guarantees because of the practical limitations imposed to the size of the history and reserve caches.

6.2 Application Accuracy

The second set of simulations is related to the accuracy of an application that makes use of the service provided by the membership protocol. The specific application used is an epidemic aggregation protocol, which computes the estimate of the global average of a set of distributed values. The epidemic aggregation protocol SPSP [8] was adopted to perform the global aggregation. All other settings are the same as in the previous set of simulations.

The local values of the aggregation protocol are initialized with a peak distribution: all nodes have initial value of 0, but one node that has a peak value equal to network size. After some cycles, the local estimates are expected to converge to the target value of the global average of 1.

The application performance is measured by the symmetrical mean absolute percentage error (sMAPE) [14], which is a statistical measure of accuracy based on the percentage errors. The performance index sMAPE is defined as:

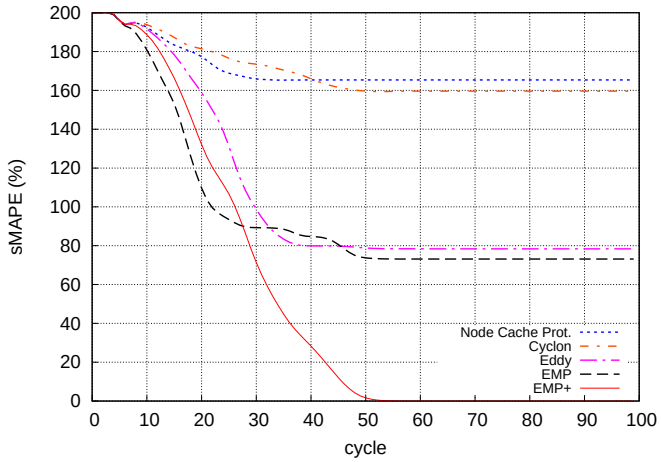


Fig. 4. The sMAPE in the overlay topology: average over 20 trials (network size: 10000 nodes)

$$sMAPE = \frac{200}{n} \times \sum_{t=1}^{n} \frac{|F_t - X|}{F_t + X},$$

where X is the target value, F_t is the forecast value (estimate) at each node and n is number of nodes. The index sMAPE is limited between 200% and -200%. Values closer to zero indicate better accuracy. This index is useful to avoid the problem of large errors when the real values are close to zero and there is a large different between the real value and the forecast.

Figure 4 shows the application performance when different membership protocols are used. At the beginning every protocol has an index of 200%, which means that none of the nodes has reached the target values of the aggregation. A loss of connectivity hinders the application to reach a sufficient accuracy and to converge to the target value. The results show that EMP+ is the only membership protocol that helps the application to achieve the ideal target value (sMAPE=0%) after about 50 cycles. EMP+ can maintain the global connectivity of the overlay topology and can also provide a good speed of convergence.

7 Conclusions

This work has investigated the effect of message interleaving on the global connectivity of the overlay topology generated by epidemic membership protocols. The internal mechanisms in membership protocols continuously transform the overlay topology by randomly rewiring the edges. This transformation is quite robust when applied to a random graph with good expansion properties. However, if the transformation is applied to an overlay topology with a weak connectivity, some edge rewiring can cause an irreversible loss of global connectivity.

The main contribution of this work is to introduce the first connectivity recovery mechanism. This mechanism has been embedded in EMP+, an enhanced version of Expander Membership Protocol (EMP). The key to achieve this goal is the Interleaving Management Procedure (IMP). The experimental analysis based on simulations has shown that EMP+ is effective in preserving and recovering the global connectivity of the overlay topology and can also provide a good speed of convergence at the application layer.

Future work will focus on the effect of node churn to the performance of epidemic membership protocols in very large and dynamic networks.

References

1. N. Bansod, A. Malgi, B. K. Choi, and J. Mayo, "Muon: Epidemic based mutual anonymity in unstructured P2P networks," *Computer Networks*, vol. 52, no. 5, pp. 915-934, 2008.
2. D. H. H. Sheng Di, Cho-Li Wang, "Gossip-based dynamic load balancing in an autonomous desktop grid," in *Proc. of the 10th International Conference on High-Performance Computing in Asia-Pacific Region*, 2009, pp. 85-92.
3. Y. Ma, and A. Jamalipour, "An epidemic P2P content search mechanism for intermittently connected mobile ad hoc networks," *IEEE GLOBECOM*, 2009, pp.1-6.
4. S. Galzarano, C. Savaglio, A. Liotta, and G. Fortino, "Gossiping-based aodv for wireless sensor networks," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, Oct 2013, pp. 26-31.
5. H. Strakov, G. Niederbrucker, and W. N. Gansterer, "Fault tolerance properties of gossip-based distributed orthogonal iteration methods," *Proc. Int. Conf. on Computational Science*, V.18, pp. 189-198, 2013.
6. P. Soltero, P. Bridges, D. Arnold, and M. Lang, "A gossip-based approach to exascale system services," in *Proceedings of the 3rd International Workshop on Runtime and Operating Systems for Supercomputers*, ser. ROSS 13. ACM, 2013.
7. M. Jelasity, S. Voulgaris, R. Guerraoui, A. M. Kermarrec, and M. van Steen, "Gossip-based peer sampling," *ACM Trans. Comput. Syst., 25(3)*, Aug. 2007.
8. F. Blasa, S. Cafiero, G. Fortino, and G. Di Fatta, "Symmetric push-sum protocol for decentralised aggregation," in *Proc. of the Int.l Conf. on Advances in P2P Systems*, 2011, pp. 27–32.
9. S. Voulgaris, D. Gavidia, and M. van Steen, "Cyclon: Inexpensive membership management for unstructured p2p overlays," *Journal of Network and Systems Management*, vol. 13, no. 2, pp. 197–217, 2005.
10. E. Ogston, and S. A. Jarvis, "Peer-to-peer aggregation techniques dissected," *Int. J. Parallel Emerg. Distrib. Syst.*, vol. 25, no. 1, pp. 51–71, Feb. 2010.
11. P. Poonpakdee, and G. Di Fatta, "Expander Graph Quality Optimisation in Randomised Communication," in *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*, pp. 597–604, Dec. 2014.
12. S. Hoory, N. Linial, and A. Wigderson,"Expander graphs and their applications," *Bullettin of the American Mathematical Society*, V. 43, N. 4, pp. 439-561, Oct. 2006.
13. A. Montresor and M. Jelasity, "PeerSim: A scalable P2P simulator," in *Proc. of the 9th Int. Conference on Peer-to-Peer (P2P'09)*, sept 2009, pp. 99–100.
14. S. Makridakis, and M. Hibon, "The M3-Competition: results, conclusions and implications," *International Journal of Forecasting 16*, pp. 451–476, 2000.