CLON: Overlay Network for Clouds*

Miguel Matos
Universidade do Minho

António Sousa
Universidade do Minho

José Pereira
Universidade do Minho

Rui Oliveira
Universidade do Minho

ABSTRACT

Gossip-based protocols have been gaining an increasing interest from the research community due to the high resilience to node churn and high scalability, thus making them suitable to modern large-scale dynamic systems. Unfortunately, these properties come at the cost of redundant message transmissions to ensure bimodal delivery to all interested peers. In systems with high message throughput, those additional messages could pose a significant burden on the excess of required bandwidth. Furthermore, the overlays upon which message dissemination takes place are oblivious to the underlying network, or rely on posterior optimizations that bias the overlay to mimic the network topology. This contributes even more to the required bandwidth as 'undesirable' paths are chosen with equal probability among desired ones.

In a Cloud Computing scenario, nodes tend to be aggregated in sub-nets inside a data-center or in multiple datacenters, which are connected by costlier, long-distance links. The goal of this work is, therefore, to build an overlay that approximates the structure of the physical network, while ensuring the connectivity properties desirable to ensure reliable dissemination. By having each node judiciously choose which nodes are in its dissemination list at construction time, i.e. by giving preference to local nodes, we are able to significantly reduce the number of messages traversing the long-distance links. In a later stage, this overlay shall be presented as a service upon which data dissemination and management protocols could be run.

1. INTRODUCTION

An essential mechanism to the proper functioning of gossip-based dissemination protocols is the peer-sampling service abstraction, that is responsible to provide a random sample of peers drawn from all the nodes interested in the dissemination process [4]. The quality of this service is essential to provide dissemination guarantees as each peer intending to send a message consults it to obtain a list of communication targets. This service has been realized using gossip mechanisms to build an overlay that encompasses all the peers willing to participate in the dissemination process, thus allowing completely decentralized gossip-based solutions to emerge [3, 8, 9, 5, 6, 7]. Those overlays, despite the concrete mechanisms used to build them, must abide to some properties to enable bimodal dissemination and high resilience to node churn [4].

In this paper we focus on building overlays that approximate the network topology with the goal of reducing the number of messages exchanged between remote peers while preserving dissemination reliability. The final goal, in the context of the DC2MS project [1], is to provide a peer-sampling service to be used by distributed data dissemination and management protocols. In the following, we will briefly analyze some of the decentralized overlay construction algorithms available and how they relate to our work.

Scamp [3] is a peer-to-peer membership service with the interesting property that the size of the local view converges naturally to the adequate size without requiring global knowledge whatsoever. This is achieved by integrating nodes in the local view based on a probabilistic function that ensures that the view size converges to $(c + 1)log(n)$ where c is a protocol parameter and n is the system size. As pointed by its authors, Scamp is oblivious to the locality and does not do any effort on the evolution of the overlay.

Cyclon [9] is a highly scalable and robust peer-to-peer overlay manager. The main mechanism relies on shuffling information among peers to achieve reliable dissemination. In opposition to Scamp, Cyclon continuously tries to enhance the overlay by means of periodic shuffles. The shuffle operation is very simple: each peer selects a random subset of peers in its local view and chooses an additional peer to which it will send the previous set. The receiving node selects a subset of its neighbors of the same size of the received one and sends it to the initial node. The receiver then discards entries pointing to itself and includes the remaining peers on its own view, discarding the entries sent to the initial node if necessary. The initial node integrates the received subset using the same approach.

HyParView [5] also uses shuffling to build an overlay network. However, each node maintains two distinct views: A small stable active view is used for message exchange, reducing redundancy. A larger passive view is maintained by shuffling and used to restore the active view when a failure is detected, thus coping with very large numbers of failed

*This work is supported by HP Labs Innovation Research Award, project DC2MS (IRA/CW118736).

WDDDM '09, March 31, 2009, Nuremberg, Germany

nodes.

The Directional Gossip Protocol [6], aims at providing dissemination guarantees in a WAN. The authors adopt a two-level gossip hierarchy to accomplish this: one level performs a traditional gossip approach within the LAN and the other level is responsible for gossiping among the WANs. The latter is achieved by using the notion of gossip servers. For each LAN there is a gossip server that, from an internal point of view, is yet another process. When a gossip server receives a message from its LAN, it sends it to the known gossip servers of the other LANs. When a gossip server receives an external message it disseminates the message internally using traditional gossip protocols. While this protocol achieves good results in the amount of messages that cross the WAN links, it relies on the undesirable feature that some nodes are special (the gossip servers). This has many consequences, namely how to 'elect' those nodes, how to make them known to the other gossip servers, and how to properly handle their failures.

The Localizer algorithm [8], builds on the previous work done in Scamp. It tries to constantly optimize the underlying overlay according to some proximity criterion. Periodically, each node chooses randomly two nodes from its neighborhood, computes the link cost to them and sends those values to both. Both nodes reply with their respective degrees and additionally one of them sends the estimate cost of establishing a link with the other. The initiator locally computes the gain of exchanging one of its links with one between the other nodes and, if desirable, performs the transition with a probability p given by a temperature function which specifies the trade-off between the closeness to an optimal configuration and the speed of convergence. Localizer has not been deeply studied in presence of high churn rates and requires the interaction among three nodes to work properly. Furthermore, if the transition is successful the initiator nodes behaves in a self-sacrificing manner as it loses one of its links.

In [7] the authors also try to solve the network mismatch problem with the goal of building an overlay with both low link cost and short paths. To achieve this the algorithm elects special nodes with a given probability and uses them to set up long distance links that reduce the global hop-count. A node with at least one neighbor, and if selected as a special node, establishes a long distance link thus eliminating the self-sacrificing behavior of Localizer and reducing the number of nodes involved on a link exchange from two to one. Apart from relying on special nodes, the authors have not considered the impact of node failures or leaves on the system, thus impairing its applicability in the highly dynamic scenarios we are aiming at.

Summarizing, the previous approaches rely on optimizing a given overlay after its construction and, on some of them, the locality-awareness is enforced by relying on specialized nodes. This poses severe impairments due to the selection process of those nodes, how to handle failures and, in certain cases, how to make them known to each other.

In this work we follow a different approach, based on some guiding principles: locality-awareness and no reliance on special nodes. The first should be a natural part of the algorithm, as it is inherent to the target topology. Secondly, nodes shouldn't assume different roles based in some criterion, if all nodes could contribute to some extent to the locality-awareness, then it is achieved without compromising other properties such as reliability. Lastly, failures should be considered as a natural event, as we aim to highly dynamic environments with considerable churn rate.

```
1   proc handleJoin(nodeId) do
      keep = randomFloat(0,1)
      if isLocal(myID,nodeId) do
        keep = Math.floor((viewSize - viewSize * localBias) * keep)
5     else
        keep = Math.floor((viewSize - viewSize * (1 - localBias)) * keep)
      endif

      if (keep == 0) and nodeId notIn view do
10      view.Add(nodeId)
      else
        forwardJoin(nodeId)
      endif
```

Listing 1: Algorithm Listing

In the next Section 2, we describe the algorithm and how our underlying principles are reflected on it. In Section 3, we describe the experimental environment and attest the relevance of the results obtained. Finally, in Section 4 we conclude the paper presenting the contributions and point to the future planned work.

2. ALGORITHM

This work was based on the initial Scamp algorithm mainly due to its simplicity and the natural convergence to the right node degree. CLON improves over Scamp by taking into account the locality of a joining node, when integrating it into its own view. The integration process consists in deciding whether the joining node shall be integrated into the view and depends on two variables, $localBias$ and $viewSize$. $viewSize$ is the size of the view of the node handling the joining node and has the same impact as in the original Scamp, it is inversely proportional to the probability of integrating the node into the view. $localBias$ is a float parameter ranging between 0 and 1, and represents the preference that should be given to local nodes. As an example with $localBias = 2/3$ the view of a given node will tend to have two thirds of local nodes and one third of remote nodes. The remaining piece is how to determine that the joining node is local to a given node, which is abstracted by the $isLocal$ function. This function should return a boolean value indicating the locality of the joining node. There are several mechanisms to perform this calculation based only on local information that are however out of the scope of this work. In Listing 1 we present the algorithm of the joining process as pseudo-code.

The mathematics behind the basic algorithm are well detailed in [3] and therefore we will skip them. The improvement is that the *keep* variable which determines whether or not a given node will be integrated into the view is weighted against the locality of the joining node, thus ensuring a locally biased overlay without impacting the correctness of the original algorithm. The $forwardJoin$ just forwards the join request to another node, to guarantee that the joining node will be eventually integrated.

3. EXPERIMENTAL EVALUATION

In this section we test the proposed algorithm and compare it with the original Scamp.

The simulations presented have been run on a custom simulator in Python. To this end, we used the NETWORKX package which provides graph manipulation facilities to rep-

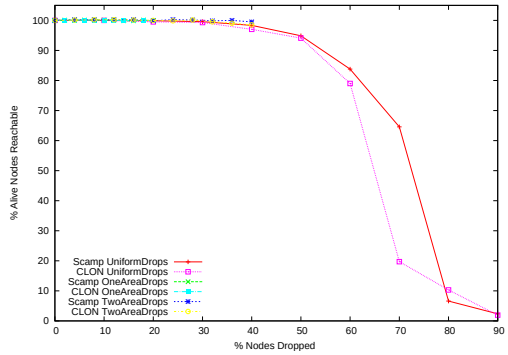

Figure 1: Overlay Connectivity.

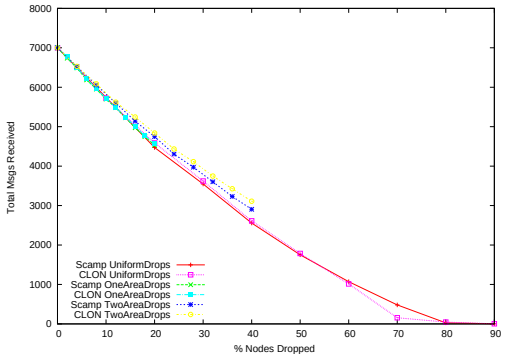

Figure 2: Total Messages Received (Average).

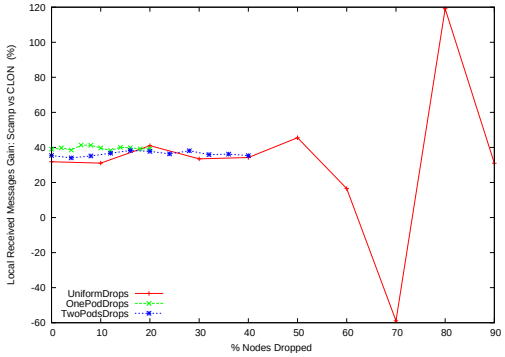

Figure 3: Local Messages Received (Average).

resent the overlay and the gossiping process on top of it. To manipulate the simulation results we recurred to the RPY package which provides a wrapper to the R programming language. The simulation kernel uses discrete time and relies on global state, handling events in a FIFO fashion.

The experiments proceeded in the following manner: first we generated a overlay encompassing all the nodes, using both algorithms. Next, we employed different drop strategies on top of the overlays without allowing them to heal. It is important to note that each drop strategy and their respective drop rates are always done on the initial overlay to allow us to compare them. The overlay is composed, initially, by 1000 nodes and has five local areas, each one with 200 nodes, along with the appropriate *isLocal* function. The $localBias$ parameter is set to 2/3.

The drop strategies used are: **uniform:** where nodes are dropped uniformly from the universe of nodes; **one local area:** where a given local area is chosen and their nodes are dropped; **two local areas:** which is equal to the previous but with two local areas selected. The dropped nodes are chosen randomly from the appropriate set. Each strategy is applied to the overlays generated by Scamp and CLON, with an increasing drop rate from 0 to 100%, in steps of 10%. The global rate of dropped nodes could be read in the horizontal axis in all figures. When analysing the graphics it is important to keep in mind that the drop rate of the different strategies is applied to different sets. For example, in the **one area** strategy a drop rate of 100% corresponds to 20% of the global nodes dropped, as reflected on the presented graphs.

Analysing Figure 1, we observe that the two algorithms are nearly identical in the number of nodes reachable which attests our claim that our biasing strategy does not impair the reliability of the generated overlay. The reachability of the overlay only gets compromised around 70% nodes dropped globally which reflects the degenerated values obtained in the subsequent graphics, pertaining to the gossiping measurements. After that value the overlays disrupt completely, making the results obtained meaningless.

After analysing the overlay, we focused on the impact on the number of messages traversing long-distance links. To better assess the impact of the overlay on the dissemination process we used a flooding protocol that delivers a given message to all the nodes on its local view and then stops gossiping. The rationale beyond this choice of protocol is that if we can achieve significant improvements in a non-optimized message hungry protocol, the results on a more carefully designed dissemination protocol should be even more attractive. The simulation of this step is done in independent cycles. In each cycle a node gossips a single message, and we run as many cycles as the number of alive nodes. As each node will be responsible for the initiation of a gossip round we can infer the reliability of the dissemination process by building the dissemination tree and checking whether it reaches almost all or almost none of the alive nodes. For each cycle we keep the number of total, remote and local messages and the dissemination tree.

In Figure 2, we observe the total number of messages received is similar in both overlays. The received number of messages approximates $\ln(N) * N$, where N is the number of alive nodes, and reflects the average degree of the nodes, due to the flooding protocol used. As we do not allow the overlays to heal, this relation decreases with the drop rate, as the average degree decreases. For example, with a drop rate of zero, the number of messages received is around 7000 which reflects the fact that $\ln(1000) \approx 7$.

Figures 3 and 4 compare the number of messages received by each protocol, regarding the location of the message sender. Figure 3 presents the increase on the number of received messages whose sender is on the same local area as the recipient. In this case our algorithm increases on 30 to 40% the ratio of local messages. On remote messages, i.e. messages traversing the long-distance links, our algorithm, as depicted in Figure 4, reduces the number of such messages on 15 to 20% as showed by the negative values.

The fact of having different ratios on the increase of local

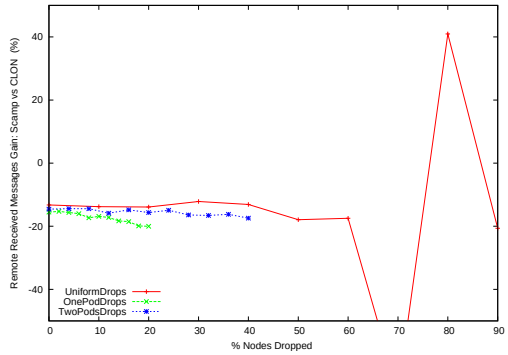

Figure 4: Remote Messages Received (Average).

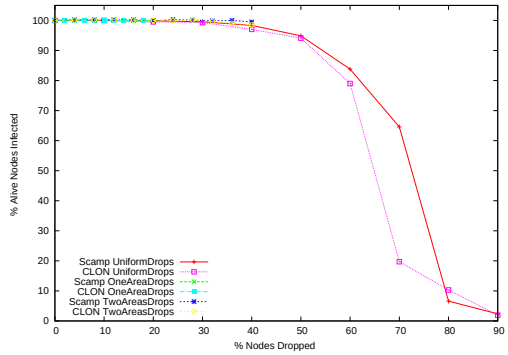

Figure 5: Message Dissemination Reliability

messages and on the reduction of remote messages results directly from the algorithm used on the construction of the overlay. In Scamp, since we have 5 local areas the number of remote nodes in the dissemination list should be of about 4/5, while in CLON the number of remote nodes is reduced to 1/3. This significantly increases the percentage of local messages, while maintaing the overall number of messages comparable.

Lastly, we checked the impact of the chosen overlay on the reliability of the dissemination process, as can be analysed in Figure 5. As it is possible to observe both protocols have nearly the same rate of infection of nodes only breaking delivery guarantees at around 60% nodes dropped in the uniform strategy. As we used a flooding protocol that disseminates to all its neighbors this graphic is very similar, as expected, to the reachability of the overlay in Figure 1.

4. FINAL REMARKS AND FUTURE WORK

In this work we argued that the total number of messages received from external nodes could be significantly reduced by building an adequate overlay. We used a novel approach, to the best of our knowledge, that imposes the network structure while constructing the overlay, instead of devising strategies to optimize it afterwards. The results look promising with an improvement of about 20% over the original Scamp in terms of messages received via long-distance links, even when using a greedy gossiping protocol and oblivious to locality. We still need to assess the result with more nodes and with different dropping strategies, that should mimic failure patterns in real scenarios. Furthermore, the adequate values for the *locaBias* parameter need to be studied carefully in order to assess to how much we are able to bias the overlay without compromising its reliability,

As future work, we intend to improve existing gossip protocols [2], by embodying the same philosophy we employed here. We do believe that with locality aware gossip protocols the results presented here could be significantly improved. Furthermore, and on the overlay management level, we still have to study the adequacy of using mechanisms to constantly evolve the overlay and the impact they have on the healing capabilities of the overlay.

5. REFERENCES

[1] DC2MS: Dependable Cloud Computing Management Services . http://gsd.di.uminho.pt/projects/DC2MS, 2008.

[2] N. Carvalho, J. Pereira, R. Oliveira, and L. Rodrigues. Emergent structure in unstructured epidemic multicast. In *DSN '07: Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 481–490, Washington, DC, USA, 2007. IEEE Computer Society.

[3] A. J. Ganesh, A.-M. Kermarrec, and L. Massoulie. SCAMP: Peer-to-peer lightweight membership service for large-scale group communication. In *Networked Group Communication*, pages 44–55, 2001.

[4] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. The peer sampling service: experimental evaluation of unstructured gossip-based implementations. In *Proc. of the 5th ACM/IFIP/USENIX Intl. Conf. on Middleware*, pages 79–98, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[5] J. Leitão, J. Pereira, and L. Rodrigues. HyParView: A membership protocol for reliable gossip-based broadcast. In *IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 419–428. IEEE Computer Society, 2007.

[6] M. J. Lin and K. Marzullo. Directional gossip: Gossip in a wide area network. In J. Hlavicka, E. Maehle, and A. Pataricza, editors, *Dependable Computing - EDCC-3, Third European Dependable Computing Conference, Prague, Czech Republic, September 15-17, 1999, Proceedings*, volume 1667 of *Lecture Notes in Computer Science*, pages 364–379. Springer, 1999.

[7] F. Makikawa, T. Matsuo, T. Tsuchiya, and T. Kikuno. Constructing overlay networks with low link costs and short paths. *Network Computing and Applications, 2007. NCA 2007. Sixth IEEE International Symposium on*, pages 299–304, July 2007.

[8] L. Massouli, A. marie Kermarrec, and A. J. Ganesh. Network awareness and failure resilience in self-organising overlay networks. In *In Proceedings of the 22nd Symposium on Reliable Distributed Systems (SRDS 2003*, pages 47–55, 2003.

[9] S. Voulgaris, D. Gavidia, and M. Steen. Cyclon: Inexpensive membership management for unstructured p2p overlays. *Journal of Network and Systems Management*, 13(2):197–217, June 2005.