StAN: Exploiting Shared Interests without Disclosing Them in Gossip-based Publish/Subscribe[*]

Miguel Matos *Ana Nunes* *Rui Oliveira* *José Pereira*

Universidade do Minho
{miguelmatos,ananunes,rco,jop}@di.uminho.pt

Abstract

Publish/subscribe mechanisms for scalable event dissemination are a core component of many distributed systems ranging from Enterprise Application Integration middleware to news dissemination in the Internet. Hence, a lot of research has been done on overlay networks for efficient decentralized topic-based routing. Specifically, in gossip-based dissemination, bringing nodes with shared interests closer in the overlay makes dissemination more efficient. Unfortunately, this usually requires fully disclosing interests to nearby nodes and impacts reliability due to clustering.

In this paper we address this by starting with multiple overlays, one for each topic subscribed, that then separately self-organize to maximize the number of shared physical links, thereby leading to reduced message traffic and maintenance overhead. This is achieved without disclosing a node's topic subscription to any node that isn't subscribed to the same topic and without impacting the robustness of the overlay. Besides presenting the overlay management protocol, we evaluate it using simulation in order to validate our results.

1 Introduction

There are two straightforward approaches to extend gossip-based broadcast [4, 7], also known as probabilistic or epidemic broadcast, for topic-based publish/subscribe [8]. The first is to maintain several stacked overlay networks, one for each topic, and have each node independently join overlays for each of its subscriptions. Unfortunately, this increases maintenance overhead and leads to redundant retransmissions, as messages published on multiple topics are separately relayed on different overlays among the same nodes.

The second approach is to keep a single overlay but structure it such that nodes with similar interests become close to each other. Hence, shared interests are recognized and redundant message transmissions avoided. Assuming that all subscription sets are known, this can be achieved efficiently using gossip itself [11, 14]. The resulting overlay is however likely to exhibit a high clustering coefficient and thus become much more prone to partitioning when nodes or links fail [10].

Several proposals address these challenges and build efficient random overlays for topic-based publish/subscribe in large scale scenarios [6, 5, 2, 3]. Unfortunately, these proposals require nodes' subscription sets to be fully disclosed to any other peer. This is itself a source of overhead, since each node might be interested in a large number of topics and thus, sharing this list will result in substantial network traffic. Moreover, fully disclosing subscription sets to every other peer might be perceived by users as violating their privacy and thus undesirable.

In this paper we present StAN, a protocol to maintain multiple stacked aligned overlay networks. Although these overlays are managed independently and retain the desired properties for gossiping, we show that they converge to share a large number of links and thus become an efficient infrastructure for gossip-based publish-subscribe. Moreover, a node p may learn that some node q is interested in topic a only if p is itself interested in a and has previously joined the overlay for such topic.

Our proposal rests on the assumption that topic subscriptions (interests) are modeled by a power law distribution in both topic popularity and number of subscriptions per node [13, 1], and that subscriptions are strongly correlated. This would mean that there is a non-negligible probability that subscription sets overlap, which is easily observed in real scenarios.

The rest of this paper is organized as follows: in Section 2, we present the protocol and the key ideas behind it, in Section 3 we evaluate it using simulation, and then discuss related work and future directions in Section 4.

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

2 The StAN Protocol

Our approach assumes that a separate random overlay has been built for each topic with all interested nodes using an existing construction protocol [9, 16]. The key properties of these initial overlays are that average node degree grows logarithmically with the system size, making them scalable, and that clustering is low, leading to resilience in face of faults and churn [10]. Choosing peers uniformly at random is key to ensuring such properties and when trying to optimize overlays, it is fundamental to maintain this property.

We also assume that a gossip protocol making use of StAN is able to exploit links on different overlays that share the same physical destination, should that occur by chance. For instance, by using underneath a dynamic pool of shared TCP/IP connections [15].

Our goal is to align these overlays to promote physical link sharing among them and thus reduce the number of physical links that must be established by each node, alleviating resource consumption and scalability problems, while allowing a message published on multiple topics to be relayed just once.

However, for each overlay, links are established in a random manner. Thus, the probability that any two given nodes are logically linked in more than one overlay, precisely what we want to promote, is dismayingly small. Furthermore, even with global knowledge about the system and subscription sets, finding a minimal solution (with the fewest links) was found to be NP-complete [5].

The key insight behind our approach is to make each node select neighbors using a pseudo-random criterion. The significant overlap in subscription sets means that, by using the same criterion in all overlays, a node will independently draw approximately the same neighbors for different subscriptions. As soon as the same neighbor has been independently discovered in distinct overlays, the commonality of interests between the two nodes is recognized and can be used. As a consequence, overlays for each topic get progressively aligned on the same physical links.

This approach has two interesting consequences. First, since the actions of each node on an overlay for a given subscription are independent of whatever other subscriptions the node might have, they do not disclose information about such other interest. In fact, a node will perform the exact same actions on a specific topic overlay (i.e. optimizing according to the pseudo-random function) regardless of what other topics it is interested on. Second, as long as the pseudo-random function provides an uniform sample of the node space, the desirable properties of the overlay for gossip-based dissemination are ensured.

To achieve this goal, each node assigns pseudo-random weights to known nodes by computing a hash value of the concatenation of its own unique name with the target node's unique name. This provides a different ordering of the identifier space at each node, which then tries to minimize the weights in its neighborhood for each topic overlay by replacing existing links. Since each node uses a different set of weights, this does not result in clustering. This makes this method substantially different from node distance as used in consistent hashing [12], which is aimed precisely at clustering nodes according to their identifiers.

```
1   periodically foreach topicId ∈ subscribedTopics
2      targetId = randomNode(myView[topicId])
3      send(targetId ,COLLECTNODES(myId ,∅ ,TTL,topicId))
4
5   proc handleCOLLECTNODES(sourceId ,idsSet ,TTL,topicId)
6     idsSet = idsSet ∪ myId
7     for nodeId ∈ myView[topicId]
8       idsSet = idsSet ∪ myID
9     if TTL > 0
10      target = randomNode(myView[topicId])
11      send(target ,COLLECTNODES(sourceId ,idsSet ,TTL−1,
12      topicId))
13    else
14      send(sourceId ,COLLECTNODESREPLY(idsSet ,topicId))
15
16  proc handleCOLLECTNODESREPLY(idsSet ,topicId)
17    viewSize = #myView[topicId]
18    idsSet = idsSet ∪ myView[topicId]
19    weigthSet = map (weigthFunction ,idsSet)
20    idsList = sort(weigthSet)
21    newView = pick first viewSize ids from idsList
22    myView[topicId] = newView
23
24  proc weigthFunction(id)
25    return (id ,Hash(string(myself)+string(id)))
```

Listing 1: StAN Protocol

The remaining challenge is to design a protocol that allows discovery of neighbors with minimum weight for each node and then is able to replace links to achieve the desired configuration. This is hard precisely due to the absence of clustering, that would enable the direct usage of known methods [11].

2.1 Protocol Description

The pseudo-code for the overlay management protocol is presented in Listing 1. We model the overlay as a directed graph and assume that the overlay management protocol ensures that the resulting graph is strongly connected.

Periodically, each node initiates a random walk with a given TTL in each overlay it belongs to, by sending its id to a random node in that topics neighborhood, as shown in lines 1 to 3. As the random walk traverses the overlay in lines 5 to 14, each node adds its id and the ids of its neighbors to the $idsSet$. When the TTL expires, the random walk ends and the originator receives the set of ids collected.

Upon reception of the set of ids collected by the random walk, in lines 16 to 22, the node computes its view size, merges the information received with what is already known about its neighbors, calculates the weight for each element of the set and selects the best $viewSize$ neighbors, replacing neighbors as appropriate to maintain the original $viewSize$. In this way, the protocol preserves the number of logical links established on each overlay, which is essential to preserve resilience.

The weight assigned to each node is given by the $weightFunction$, in lines 24 to 25. As stated above, we rely on the properties of hash functions to obtain uniformity and determinism. In order to satisfy the requirement of a unique ordering of the nodes for each node, the id of the local node is concatenated with the id of the remote one so that the input of the hash function is unique for each pair of nodes. Thus the resulting weight will be unique for each pair of nodes.

By relying on the uniformity of the hash function, the other properties of the overlay, such as the degree distribution are preserved. Because the overlays are modelled as directed graphs, decisions are made strictly locally and unilaterally, which contributes to the scalability and robustness of the proposal.

3 Evaluation

In this section we assess, using discrete-event simulation, to what extent StAN produces an efficient overlay for publish/subscribe information dissemination. To fulfill the assumptions on subscription distribution presented in Section 1, we built a two dimensional grid and randomly placed nodes and topic ids on it. Then, each node was assigned a given interest radius, and subscribed to the topics in that radius. Both the number of topic ids placed on the grid for each topic and the interest area follow a power law distribution, thus complying with the aforementioned model. Nodes close to each other on the grid are likely to subscribe to the same topics, thus modeling the correlation in subscriptions.

Figure 1 shows a typical distribution of subscriptions for 1000 nodes and 100 topics. On the left, we have the number of subscribers for each topic which shows that there are few topics that are highly popular and many topics that have a smaller number of subscriptions. On the right, we have the number of subscriptions per node, which shows that in this scenario, the vast majority of nodes is subscribed to few topics and there are few nodes with a considerable number of subscriptions. To assess the correlation among subscriptions we built a correlation matrix with the subscription sets and calculated the correlation among each pair of subscriptions. The result is depicted in Table 1.

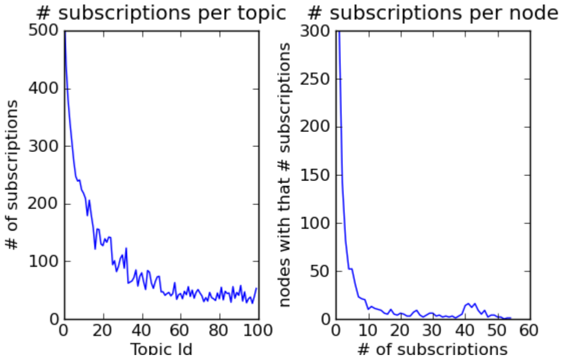

Figure 1: Subscriptions for 1000 nodes/100 topics.

Confidence Level	% Nodes
90%	69%
95%	61%
98%	51%
99%	45%

Table 1: Interest correlation.

After assigning the topics to the nodes, we generated an Erdos-Renyi random graph for each topic ensuring that it is strongly connected and analyzed: 1) logical link sharing and 2) the impact on the properties of the overlays. The experiments were run with combinations of 1000, 2000 and 3000 nodes with 100, 200 and 300 topics with $TTL = 5$.

3.1 Results

First, we analysed the impact of the protocol in promoting link sharing. To this end, we defined two measurements: Total View Size and Unique View Size, which are, respectively, the number of total and unique known neighbors. The Total View Size is the sum of the view sizes of all the overlays in which the node participates and measures the total number of logical links. The Unique View Size measures which node identifiers are unique and captures the number of physical links. Therefore, the Total View Size remains constant across the whole experiment, since the protocol preserves the number of logical links and the Unique View Size tends to decrease as the protocol promotes link sharing. The results obtained are depicted in Table 2.

For each configuration, we present the Total View Size and the initial and final Unique View Size. As it is possible to observe, the final Unique View Size is considerably smaller in all the analyzed configurations, which shows that our protocol is effective in aligning the overlays with the same physical links.

In Figure 2, we show the evolution of the Total and

# Nodes	View Sizes	# Topics		
		100	200	300
1000	Total	89	183	277
	Unique Initial	65	109	145
	Unique Final	29	44	55
2000	Total	104	209	315
	Unique Initial	84	146	197
	Unique Final	40	58	72
3000	Total	110	224	335
	Unique Initial	95	169	231
	Unique Final	45	67	83

Table 2: View sizes for several configurations.

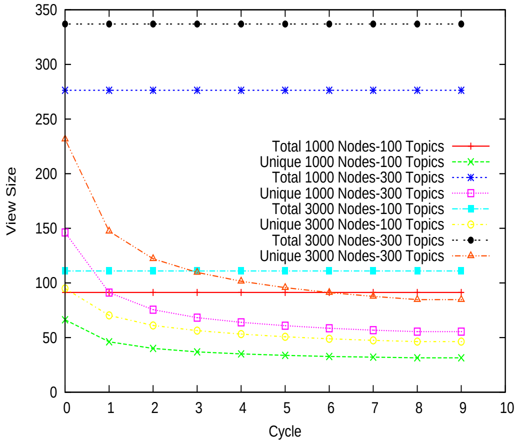

Figure 2: View size evolution.

Unique View Sizes for the configurations of 1000 and 3000 nodes with 100 and 300 topics, which attests that the protocol converges quickly, in few iterations.

It is important to notice that the view size increases much quicker with the number of topics than with the number of nodes, as can be observed in Table 2 and Figure 2. This is because the view sizes for each overlay typically grow logarithmically with the number of nodes, but a subscription to more topics implies a linear growth, as $fanout$ links need to be established. This (expected) behaviour further stresses the importance of sharing links across the overlays in order to promote scalability. In fact, the final view sizes obtained hint that StAN is able to scale properly in both the number of nodes and topics.

Finally, we analyse StAN's impact on the structural properties of the overlays, namely the degree distribution, diameter and clustering of the graph induced by all topics, i.e. the graph resulting from the physical links. In Figure 3 we present the Cumulative Distribution Function for the in and out node degree before and after running the protocol. The difference between the before and after curves shows that StAN eliminates, as expected, the highest degree occurrences, which is important for scalability, but does not affect the distribution of lower degree occurrences, which is important for robustness. The similarity of the in and out degree curves for each instant shows that, on average, each node is known by as many nodes as it knows, as expected from a random graph and thus indicates that our approach does not have a meaningful impact on randomness.

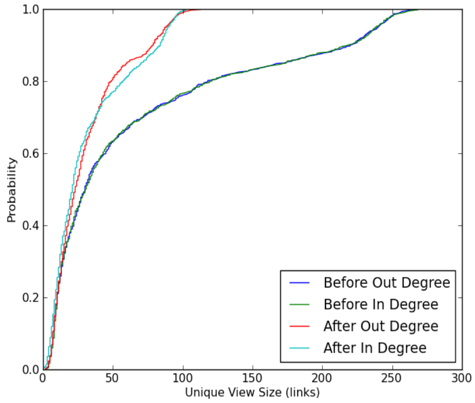

Figure 3: Unique view size for 1000 nodes/100 topics.

Measure	Before	After
Clustering Coefficient	0.50	0.13
Diameter	3	4

Table 3: Overlay properties for 1000 nodes/100 topics.

Table 3 shows StAN's impact on the clustering coefficient and on the diameter. The clustering coefficient is affected by two contradictory factors. First, it tends to decrease as the number of links decreases, and second it tends to increase as nodes establish links with the same neighbors on multiple overlays. However, because neighbors are chosen uniformly, this impact is reduced, leading to the overall reduction of the clustering coefficient. Finally, the small increase in the diameter, which has an impact on latency, is justified by the reduction in the number of physical links.

4 Discussion

Most approaches to topic-based publish/subscribe focus on constructing overlays from scratch. To the best of our knowledge, this work is the first attempt to optimize, in terms of the number of physical links needed, preexistent overlays, by exploiting shared interests between participants without actually disclosing them. Additionally, this is accomplished while preserving the robustness of the pre-existent overlays.

Sub-2-Sub [17] is a content-based protocol that clusters nodes according to their subscriptions to construct a ring for each attribute, but it requires subscription sets to be exchanged between neighbors. Also, the rings evaporate randomness and consequently, the resilience it affords. In TERA [3], participants are clustered into separate overlays according to topic subscriptions. However, subscription correlation is not taken into account while still requiring subscriptions to be disclosed. SpiderCast [6] builds an overlay topology that scales well with the number of topics and nodes. Unfortunately, the approach requires knowledge of significant parts of the system to obtain good results, subscription disclosure and moreover nodes need to agree on link establishment and removal.

In [5], the authors define the relevant Min-TCO problem, and a greedy, centralized algorithm that assumes global knowledge is presented to solve it. However, the resulting minimal graph is brittle, and therefore not suited for dynamic large scale scenarios.

In this paper we presented StAN, a protocol that takes advantage of the correlation in subscriptions in a topic-based environment to reduce resource usage by sharing physical links. By allowing each node to selectively choose its neighbors but nonetheless preserving the randomness of the process, StAN is able to considerably reduce the number of physical links used, while preserving the base graph properties of the overlay, as the evaluation attests. In short, a very simple protocol that does not require disclosing interests is surprisingly effective in achieving its goal.

Currently, we are applying StAN within an actual protocol implementation[1] to experimentally evaluate it in a real setting, specifically, to assess its scalability regarding the number of topics in the system and subscribed by each node. Moreover, it is interesting to speculate whether the same approach can be combined with other approaches for gossip-based publish/subscribe.

References

[1] Adamic, L., and Huberman, B. Zipf's law and the Internet. *Glottometrics 3*, 1 (2002), 143–150.

[2] Baehni, S., Eugster, P., and Guerraoui, R. Data-aware multicast. In *Proceedings of the 5th IEEE International Conference on Dependable Systems and Networks (DSN 2004)* (2004), Citeseer, pp. 233–242.

[3] Baldoni, R., Beraldi, R., Quema, V., Querzoni, L., and Tucci-Piergiovanni, S. TERA: topic-based event routing for peer-to-peer architectures. In *Proceedings of the 2007 inaugural international conference on Distributed event-based systems* (2007), ACM, p. 13.

[4] Birman, K., Hayden, M., Ozkasap, O., Xiao, Z., Budiu, M., Mihai, M., and Minsky, Y. Bimodal multicast. *ACM Transactions on Computer Systems. 17*, 2 (1999), 41–88.

[5] Chockler, G., Melamed, R., Tock, Y., and Vitenberg, R. Constructing scalable overlays for pub-sub with many topics. In *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing* (2007), ACM, p. 118.

[6] Chockler, G., Melamed, R., Tock, Y., and Vitenberg, R. Spidercast: a scalable interest-aware overlay for topic-based pub/sub communication. In *DEBS '07: Proceedings of the 2007 inaugural international conference on Distributed event-based systems* (New York, NY, USA, 2007), ACM, pp. 14–25.

[7] Eugster, P., Guerraoui, R., Handurukande, S., Kouznetsov, P., and Kermarrec., A.-M. Lightweight probabilistic broadcast. *ACM Transactions on Computer Systems 21*, 4 (2003), 341–374.

[8] Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A.-M. The many faces of publish/subscribe. *ACM Computer Survey 35*, 2 (2003), 114–131.

[9] Ganesh, A., Kermarrec, A., and Massoulié, L. SCAMP: Peer-to-peer lightweight membership service for large-scale group communication. *Lecture notes in computer science* (2001), 44–55.

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

[11] Jelasity, M., Montresor, A., and Babaoglu, O. T-man: Gossip-based fast overlay topology construction. *Computer Networks 53*, 13 (August 2009), 2321–2339.

[12] Karger, D., Lehman, E., Leighton, T., Panigrahy, R., Levine, M., and Lewin, D. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web. In *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (New York, NY, USA, 1997), ACM, pp. 654–663.

[13] Liu, H., Ramasubramanian, V., and Sirer, E. Client behavior and feed characteristics of rss, a publish-subscribe system for web micronews. In *Proc. of ACM Internet Measurement Conference* (2005).

[14] Massoulié, L., Kermarrec, A.-M., and Ganesh, A. Network awareness and failure resilience in self-organising overlay networks. In *Proceedings of the 22nd Symposium on Reliable Distributed Systems* (2003), pp. 47–55.

[15] Pereira, J., Rodrigues, L., Oliveira, R., and Kermarrec, A.-M. Neem: Network-friendly epidemic multicast. In *Proceedings of the 22nd Symposium on Reliable Distributed Systems* (2003), IEEE, pp. 15–24.

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

[17] Voulgaris, S., Riviere, E., Kermarrec, A., and Van Steen, M. Sub-2-Sub: Self-organizing content-based publish and subscribe for dynamic and large scale collaborative networks. In *IPTPS'06: the fifth International Workshop on Peer-to-Peer Systems* (2006), Citeseer.

[1] http://neem.sf.net