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 EAI 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, approximating nodes with shared interests 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 share a large number of physical connections, thereby leading to reduced message traffic and maintenance overhead. This is achieved without a node ever disclosing an interest to another node that doesn't share it and without impacting the robustness of the overlay. Besides presenting the overlay maintenance 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 all 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 interests are known, this can be achieved efficiently using gossip itself [11, 13]. The resulting overlay is however likely to exhibit a high clustering coefficient and thus become much easier to partition 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 that a node's interests are 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, having to share this list will incur in substantial network traffic. Moreover, fully disclosing the list of subscribed topics 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 to 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 the subscriptions to topics (interests), is modeled by a power law distribution in both topic popularity and number of subscriptions per node [12, 1], and that subscriptions are correlated with significant degrees of confidence. This in practice means that there is a non-negligible probability that the interests of nodes 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 in Section 4 we compare our protocol with previous proposals and discuss future directions.

2 The StAN Protocol

2.1 Rationale

Our approach starts with a random overlay being built for each topic, containing all interested nodes [9, 15]. The key properties of these overlays are that 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 [14].

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, the neighbouring relationships among nodes are established in a random manner. The probability that any two given nodes have the same logical link in more than one overlay, precisely what we want to promote, is dismayingly small. Furthermore, even with global knowledge about the system and node subscriptions, finding a minimal solution (with the fewest links) was found to be NP-complete [5].

The key insight behind our approach is that if nodes are able to choose the same set of neighbours in all overlays they participate in, but nonetheless preserve the randomness of the choice, then the protocol will be promoting link sharing due to the correlation of the interests.

This is achieved by attributing pseudo-random weights to neighbours and giving preference to the establishment of links with the neighbours of least weight. Thus, by attributing the same weight in all the overlays to the potential neighbours the protocol is able to deterministically establish links and thus promote link sharing. Naturally, this applies if and only if the neighbours in a given overlay also belong to other overlays to which the node belongs, but due to the interest correlation this is likely. Additionally, the weight of neighbours should be attributed in an uniform fashion, otherwise it will induce clustering among nodes, leading to the degradation of the overlay quality and possibly to partitions. These two properties, uniformity and determinism can be found in hash functions and thus we use a hash function to obtain the weight that should be given to the neighbours by simply concatenating both ids. This is important as it enables each node to attribute different weight to the same set of neighbours and thus avoid clustering.

The remaining challenge is to design a protocol that optimizes each overlay in the stack according to this criterion.

```
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

2.2 Description

The pseudo-code for the overlay management protocol is presented in Listing 1 and works as follows. 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 walks with a given TTL in each overlay it belongs to, by sending its id to a random node in that topic's view, as shown in lines 1 to 3. $send$ is a network level level primitive that receives a destination node and a function to be invoked at the receiver side, by means of the $handle*$ primitive. How often a random walk is started offers a trade-off between the speed of convergence and the load imposed on the network.

Upon reception of the random walk, the receiver adds its id to the set of ids already in the random walk, $idsSet$, as well as the ids of its neighbours on that overlay, which allows the random walk to collect a broader set of ids. Then, the receiver checks the TTL and either forwards the random walk to a random neighbour, or sends it back to the source if it was already expired.

Upon reception of the reply, in lines 18 to 24, the node computes its view size, merges the information it knows about its neighbours with the one received from the random walk, calculates the weight for each element of the

set and selects the $viewSize$ neighbours, replacing them as appropriate.

In this way, the protocol preserves the amount of logical links established on each overlay, which is essential to preserve resilience. Additionally, and as stated above, 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 dependability of the proposal.

3 Evaluation

In this section we describe the evaluation process conducted in order to assess the proposed protocol.

All experiments have been conducted on a Python-based, custom made, discrete event simulator.

To cope with the model requirements presented in Section 1 we proceeded as follows. Initially, we built a two dimensional grid and randomly placed nodes and topic ids on it. Then, each node was attributed a given interest radius, and subscribes to the topics that fall into that radius. Both the amount of topic ids placed on the grid for each topic, and the interest radius follow a power law distribution, thus complying with the aforementioned model. Finally, nodes close to each other on the grid will likely subscribe to the same topics, thus modelling the interest correlation.

To assess the validity of the scenario, we present in Figure 1 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 amount of subscriptions. To assess the correlation among subscriptions we built a correlation matrix with the subscriptions of each node and calculated the correlation among each pair of nodes. As depicted in Table 1, the result is that almost all node subscriptions are reasonably correlated. In fact, we can state with a 90% confidence level that almost 70% of the nodes' subscriptions are correlated.

After attributing the topics to the nodes, we generate a random graph for each topic ensuring that it is strongly connected and analyze: 1) the sharing of logical links and 2) the impact on the properties of the overlays.

The experiments use the following configurations: combinations of 1000, 2000 and 3000 nodes with 100, 200 and 300 topics with $TTL = 5$.

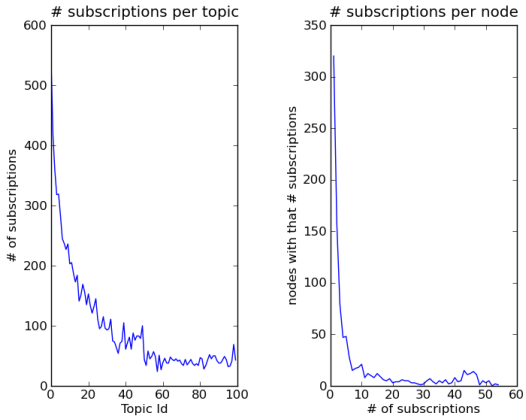

Figure 1: Subscription distribution for 1000 nodes and 100 topics.

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

Table 1: Interest Correlation Confidence Levels

3.1 Results

First, we analyse the impact of the protocol in promoting link sharing among nodes with similar interests. To this end, we defined two measurements: Total View Size and Unique View Size. 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, we expect the Total View Size to remain constant across the whole experiment, since the protocol preserves the number of logical links and the Unique View Size to decrease as the protocol gives preference to neighbours with less weight. 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 promoting link sharing.

In Figure 2, we show the evolution of the Total and Unique View Sizes as the protocol proceeds for the configurations of 1000 and 3000 nodes with 100 and 300 topics, which attests that the protocol is able to stabilize

# 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: Global and Unique View Sizes for different configurations.

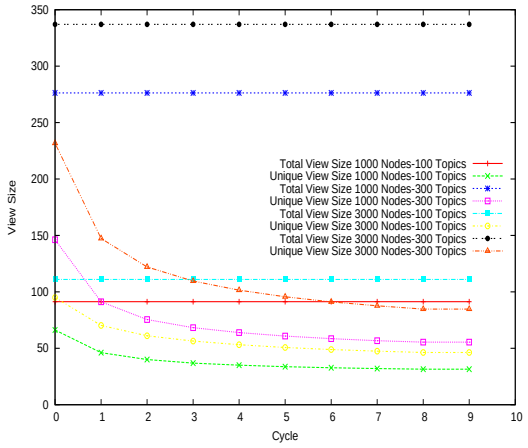

Figure 2: View Sizes Evolution.

in few iterations.

Another important remark, which can be observed both in Table 2 and Figure 2, is that the view size increases much quicker with the number of topics than with the number of nodes, as 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 overlay 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 the impact of our protocol in the structural properties of the overlays namely the degree distribution, diameter and clustering of the graph induced by all the topics, i.e. the graph that represents 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

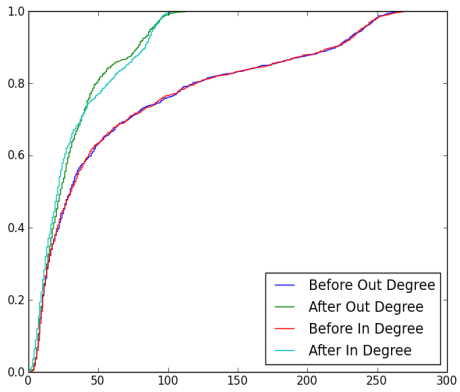

Figure 3: Degree Distributions for 1000 Nodes and 100 Topics.

Measure	Before	After
Clustering Coefficient	0.50	0.13
Diameter	3	4

Table 3: Physical Overlay Properties for 1000 Nodes and 100 Topics.

the before and after curves shows that our protocol 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 resilience. 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.

Table 3 shows the impact of the protocol in the clustering coefficient and 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, as 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 is directly related to the latency, of the overlay is justified by the reduction on the number of links.

4 Discussion

Most approaches to topic-based publish/subscribe focus on constructing overlays from scratch and, to the best of our knowledge, this work is the first attempt to opti-

mize, in terms of number of physical channels needed, pre-existent overlays by exploiting shared interests between participants without actually disclosing them.

Sub-2-Sub [16] is a content-based protocol that clusters nodes according to their subscriptions to construct a ring for each attribute, but subscription sets are exchanged between neighbours, and the ring structure formed evaporates randomness and consequently, the resilience it affords.

TERA [3] is a protocol in which participants are clustered into separate overlays according to interests, but does not take into account interest correlation and requires subscriptions to be disclosed.

SpiderCast [6] explores correlation in subscriptions by using *k*-regular random graphs. However, the approach requires subscription disclosure and nodes to agree on link establishment and removal. Furthermore, it does not improve over time and good results require knowing a large portion of the system.

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

In this paper we presented a protocol that takes advantage of the correlation of interests among nodes in a topic-based environment to reduce resource usage by sharing physical channels. By allowing each node to selectively choose its neighbours but nonetheless preserving the randomness of the process, the protocol is able to considerably reduce the number of physical channels used while preserving the base graph properties of the overlay, as the experimental 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 with real traffic, 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 used as a first step to improve other, more effective but much more complex protocols for gossip-based publish/subscribe information dissemination.

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 Comput. Surv. 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] 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).

[13] 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.

[14] 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.

[15] 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.

[16] 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