Towards Survivability of Application-Level Multicast

Gal Badishi
EE Department, Technion

Idit Keidar
EE Department, Technion

Roie Melamed
CS Department, Technion

1 Introduction

This position paper focuses on challenges in providing survivable and scalable multi-point to multi-point reliable *application-level multicast systems* (ALMs) for very large groups in wide-area networks. A protocol deployed in such settings must be able to withstand frequent node failures as well as non-negligible message loss rates. A survivable system should also cope with uncooperative users. Moreover, in typical wide-scale multicast sessions, users frequently join and leave [1]. Rapid joining and leaving, also called *churn*, may effectively cause *denial of service* (DoS) if handling joins and leaves induces high overhead.

Survivability also mandates withstanding attacks. One of the most devastating security threats faced by a distributed system is a DoS attack. Coping with DoS attacks is essential when deploying services in a hostile environment such as the Internet; in 2003, approximately 42% of U.S. organizations were faced with DoS attacks [4].

As a first defense, one may protect a system against DoS attacks using network-level mechanisms [3]. However, network-level filters cannot detect DoS attacks at the application level, when the traffic seems legitimate. This is especially true when the application performs intensive computations for each message, as occurs, e.g., with secure protocols based on digital signatures.

An attack that targets every node in a large system inevitably causes performance degradation, but also requires vast resources. In order to be effective even with limited resources, attackers target vulnerable parts of the system. E.g., consider a tree-based multicast protocol; by targeting a single inner node in the tree, an attacker can effectively partition the multicast group. Hence, eliminating single points of failure is an essential step in constructing survivable protocols.

More generally, our goal is to design a protocol that would not allow an attacker to increase the damage it causes by focusing on a subset of the nodes.

There are two leading approaches to building scalable ALMs: gossip-based (epidemic) protocols, and dynamic overlay networks. In order to be survivable, overlays require redundancy, as opposed to a single multicast tree. We begin by examining gossip, since it has no single points of failure, overcomes churn, and exhibits a graceful performance degradation as the number of faulty or uncooperative nodes rises [5]. In Section 2 we summarize our work on Drum [2], a DoS-resistant gossip protocol. In [2], we also present a systematic quantitative study of the effect of DoS on gossip protocols; we are not familiar with any previous attempts to quantify the impact of DoS attacks on a distributed system.

We note that gossip may induce high overhead, and only provides probabilistic reliability. Overlay networks can reduce the overhead and increase the reliability. In Section 3, we summarize our efforts on building Araneola, a robust multicast overlay that deals with churn with low overhead, and overcomes random failures of a certain percentage of nodes/links. However, this work does not address DoS and uncooperative users. Finally, in Section 4, we discuss remaining challenges and suggest directions for future research.

2 Drum: Dos-Resistant Gossip

One may expect that a gossip-based system will not suffer from vulnerabilities to DoS attacks, since it can continue to be effective when many nodes fail. Surprisingly, we show in [2] that gossip-based protocols can be extremely vulnerable to DoS attacks targeted at a small subset of the nodes. This occurs because an attacker can effectively isolate a small set of nodes from the rest of the group by attacking this set. To quantify the effects of DoS attacks, we measure their influence on the time it takes to propagate a message to 99% of the nodes in the system, as well as on the average throughput nodes can receive, using asymptotic analysis, simulations, and measurements. Here, we include only exemplary results.

Solution. In [2], we present *Drum* (DoS-Resistant Unforgeable Multicast), a gossip-based multicast protocol, which, using a few simple ideas, eliminates common vulnerabilities to DoS attacks. Drum uses both the push and pull gossiping techniques in order to resist DoS attacks. Nodes attacked using the push channels can still receive messages using pull, and nodes whose pull channels are attacked can still send messages using push. In order to realize this, resources are separated and bounded for each operation. Thus, using the push channels does not affect the ability to utilize the pull channels, and vice versa. Finally, well-known ports are solely used for communicating control messages, while data messages are delivered on random ports in order to further decrease the attacker's probability of launching an effective DoS attack.

In [2] we present a mathematical analysis of Drum, and simulation results that validate the analysis. We have also implemented Drum in Java and tested it on a cluster of workstations. We prove analytically and show empirically that when an adversary has a large sending capacity, its most effective attack against Drum is an all-out attack that distributes the attacking power as broadly as possible. Figure 1(a) presents exemplary simulation results illustrating this property. Obviously, performance degradation due to a broad all-out DoS attack is unavoidable for any multicast protocol, and indeed all the tested protocols exhibit the same performance degradation under such a broad attack (see rightmost data point in Figure 1(a)).

Additionally, our results show that Drum can withstand severe DoS attacks, where naïve protocols that do not take any measures against DoS attacks completely collapse. E.g., we prove that under an attack that focuses on a strict subset of the nodes, Drum's latency and throughput remain *constant* as the attack strength increases, whereas in traditional protocols, the latency grows *linearly* with the attack strength, and the throughput continuously degrades; the simulation results in Figure 1(b) illustrate this for the scenario that 10% of the nodes are attacked.

Pros. Drum, as all gossip-based protocols, operates well in adversarial scenarios where arbitrary subsets of nodes fail or are uncooperative. Additionally, Drum eliminates vulnerabilities to DoS attacks.

Cons. Drum is not bandwidth optimized; it gossips about each message identifier many times. Moreover, the rapid change in communication partners makes a reliable analysis of the correct operation of neighboring nodes difficult. Thus, incentivizing nodes to cooperate is problematic.

3 Araneola

In [6], we present Araneola[1], a scalable reliable ALM for highly dynamic wide-area environments. Araneola disseminates messages on an unstructured overlay in which each node has a small number of neighbors: for a tunable parameter $k \geq 3$, each node's degree is either k or $k+1$, and roughly 90% of the nodes have degree k. This parameter can be tuned according to the desired resilience. Thanks to the use of low degree overlay, Araneola sends less packets per application message than gossip-based protocols or high degree overlays.

We have implemented Araneola in Java and evaluated it extensively on up to $10,000$ nodes. Our evaluation shows that Araneola's overlay achieves three important mathematical properties of k-regular random graphs (random graphs in which each node has exactly k neighbors) with N nodes: (i) its diameter grows logarithmically with N; (ii) it is generally k-connected; and (iii) it remains highly connected following random removal of linear-size subsets of edges or nodes. Figure 2(a) shows the resistance of Araneola's overlay to random edge removals: for each percentage x of removed edges, the graphs shows the average size of the the largest connected component remaining in the overlay after a random $x\%$ of the edges are removed, for overlays constructed with $k = 4$ and with $k = 5$. It shows that for a fixed $k = 5$, the number of nodes does not affect the overlay's resilience. Thus, Araneola can achieve high robustness while inducing a much smaller overhead than a gossip-based protocol.

Another important property of Araneola is its ability to deal with churn with a low overhead: each join, leave, or failure is handled locally, and entails the sending of only about $3k$ messages in total. Remarkably, as illustrated in Figure 2(b), the cost of handling a single join or leave operation *decreases* as the join and leave rate increases. This is in contrast to virtually all existing structured peer-to-peer overlays, with which the overhead for handling joins grows logarithmically with N.

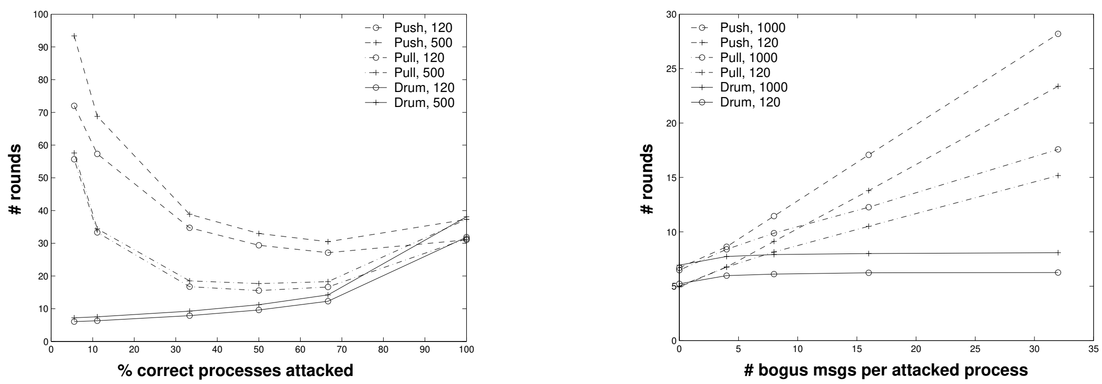
Pros. Lower overhead and latency than in gossip-based protocols. Constant overhead for join/leave operations. Graceful degradation of performance as the failure rate increases. **Cons.** Araneola does not currently deal with DoS attacks, uncooperative behavior, and non-random failures.

4 Challenges and Future Directions

We have seen examples of two ALMs. The first, Drum, uses simple techniques to mitigate the effects of DoS attacks on gossip-based protocols. These techniques can be used in other systems as well, and are generalized as follows: (i) allow other nodes to choose you as a neighbor, but also choose some neighbors by yourself; and (ii) minimize the use of well-known ports for communication. Moreover, the methodology used in [2] can be used to analyze the impact of DoS on various systems, as well as to evaluate the effectiveness of mechanisms for mitigating this impact. Currently, Drum uses well-known ports to communicate control messages to new neighbors each round. Future research will analyze the use of pseudo-random ports as an alternative to well-known ports.

The second ALM, Araneola, builds an overlay network that strives to minimize the overhead incurred by operating in a dynamic environment, while maintaining good fault-tolerance properties. By building a low degree overlay, Araneola allows for better performance (lower overhead) than gossip protocols and high degree overlays. Naturally, using an overlay, a node's neighbors do not

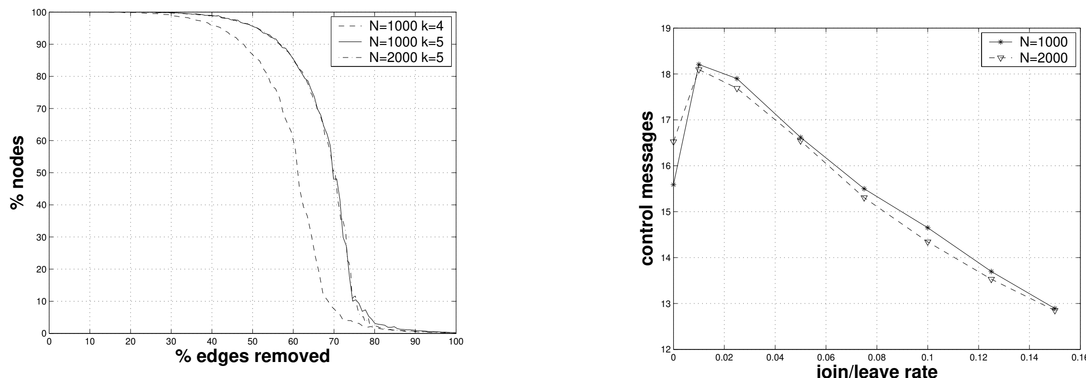
[1] Araneola means "little spider" in Latin.



(a) Fixed strength, increasing attacked percentage.

(b) Fixed attacked percentage, increasing strength.

Figure 1: Drum versus pull-based and push-based protocols: Average propagation time to 99% of the correct nodes for different scenarios and two group sizes (simulations).



(a) Percentage of nodes remaining in largest connected overlay component following random edge removals.

(b) Overhead for dealing with join/leave operations.

Figure 2: Araneola measurements: Overhead for handling churn and fault-tolerance of the overlay.

change as rapidly as in gossip-based protocols. Future enhancements may exploit this attribute for: (i) efficient message authentication schemes using symmetric cryptography; and (ii) monitoring neighbors for uncooperative behavior.

It is currently a challenge to design an ALM that is both efficient in terms of bandwidth and latency even in dynamic environments (cf. Araneola), and operates well in adversarial scenarios, where an arbitrary set of nodes can be attacked (cf. Drum). In the future, Araneola may be extended to facilitate choosing neighbors according to the principles employed in Drum: choose some of the neighbors by yourself, and let some other nodes choose you as a neighbor.

References

[1] K. C. Almeroth and M. H. Ammar. Collecting and modeling the join/leave behavior of multicast group members in the mbone. In *High Performance Distributed Computing (HPDC)*, August 1996.

[2] G. Badishi, I. Keidar, and A. Sasson. Exposing and eliminating vulnerabilities to denial of service attacks in secure gossip-based multicast. In *The International Conference on Dependable Systems and Networks (DSN)*, 2004. To appear.

[3] Cisco Systems. Defining strategies to protect against TCP SYN denial of service attacks. http://www.cisco.com/warp/public/707/4.html.

[4] CSI/FBI. Computer crime and security survey, 2003. http://www.gocsi.com/forms/fbi/pdf.jhtml.

[5] M. J. Lin, K. Marzullo, and S. Masini. Gossip versus deterministically constrained flooding on small networks. In *14th International Symposium on DIStributed Computing (DISC)*, pages 253–267, 2000.

[6] R. Melamed and I. Keidar. Araneola: A scalable reliable multicast system for dynamic environments. TR CCIT 474, Electrical Engineering Department, Technion, March 2004. http://www.ee.technion.ac.il/ idish/Xchange/araneola.ps.