

Using Randomized Techniques to Build Scalable Intrusion-Tolerant Overlay Networks

Robbert van Renesse

Department of Computer Science
Cornell University, Ithaca, NY 14853
rvr@cs.cornell.edu

Abstract

Overlay networks provide important routing functionality not easily supported directly by the Internet. Distributed Hash Tables (DHTs) have been proposed to support such overlay networks. While it is often straightforward to support overlay networks on DHTs, this choice can be questioned. DHTs dictate routes that are not optimal, and DHTs are hard to secure. As overlay networks are beginning to be deployed for critical applications, efficiency and security are becoming important attributes.

We present an alternative support structure called *Fireflies*. *Fireflies* provides each of its members with a complete view of its live peers. A small subset of these peers are marked as *neighbors*. With high probability, the mesh formed by the members and their neighbor links has a diameter logarithmic in the number of live members, and connects all the reachable members that are not Byzantine.

Fireflies uses several randomized algorithms, which are briefly described. We also discuss how *Fireflies* may be used to build intrusion-tolerant overlay networks.

⁰We are supported by DARPA's SRS program, a MURI grant, the AFRL/Cornell Information Assurance Institute, and the NSF CyberTrust program.

1 Introduction

We describe *Fireflies*, a scalable protocol for supporting intrusion-tolerant overlay networks. While such a protocol cannot distinguish Byzantine nodes from correct nodes in general, it provides correct nodes with a reasonably current view of which nodes are live, and a pseudo-random mesh for communication. The amount of data sent by correct nodes grows linearly with the aggregate rate of failures and recoveries, even if provoked by Byzantine nodes.

There are limitations to group membership guarantees if intrusions are allowed. A Byzantine member can disguise itself as a flaky correct member. Nonetheless, intrusion-tolerant overlay network routing protocols may be built using intrusion-tolerant group membership as a building block.

Below we give an overview of the four randomized techniques on which *Fireflies* is based (Section 2, and how *Fireflies* may be used in applications (Section 3). Section 4 briefly describes related work.

2 Randomized Techniques

Fireflies uses a variety of randomized techniques, in particular:

- a broadcast channel based on gossip;
- partial membership views for gossip, determined by a collision-resistant hash function;
- a membership protocol that limits who can accuse whom using a collision-resistant hash function;
- an adaptive failure detection protocol based on negative binomial distributions and exponential smoothing.

We will now describe these in more detail. Complete details, an evaluation, and a proof of correctness, will appear in a later paper.

A gossip protocol is a simple group communication protocol whereby each member periodically picks a random member from its view and exchanges state information. Such protocols are known to be highly robust. In our particular situation, we have to concern ourselves with Byzantine members.

Say we have two members m_1 and m_2 exchanging updates. All updates are signed by their initiators, and because we assume that Byzantine members cannot break the cryptographic building blocks, we do not have to worry about impersonation attacks. We also assume that trivial Denial-of-Service attacks can be detected and suppressed.

But Byzantine members can still attack the gossip protocol in the following two ways. In order to slow down dissemination, they can neglect to forward recent updates. Byzantine members can also pretend that they have no information, causing correct members to transmit their entire state to them and thus causing unnecessary load on the correct members. In order to reduce the opportunities of Byzantine members to launch this attack, we will consider gossip protocols in which each member can only gossip with a small subset of the membership.

In *Fireflies*, each member determines a small set of gossip neighbors based on their view using a collision-resistant hash function. The size of this set is determined by the desired probability that the correct members form a connected graph. The resulting graph has a logarithmic diameter with high probability, and thus dissemination takes logarithmic time with high probability.

The basic idea of the membership protocol is that members monitor one another, and use the gossip channel to disseminate *accusations* (failure notices). When a member m_1 receives an accusation for a member m_2 , m_1 waits a time period determined by the gossip latency before removing m_2 from its view. Should m_2 receive, through gossip, an accusation about itself, then m_2 has the opportunity to gossip a *rebuttal*. There is an overhead associated with gossip, so we have to prevent Byzantine members from submitting frequent accusations about correct members. In order to do so, members are restricted using a collision-resistant hash function whom they are allowed to accuse, and recently rebutted members are prevented from making repeated accusations about the same member.

Members use pinging to detect failures. Essentially, a member m_1 monitors a member m_2 by sending “ping” messages to m_2 at regular intervals. m_2 returns a “pong” message for each ping that it receives. If m_1 does not receive pong messages from m_2 for over some time period, m_1 considers m_2 stopped and issues an accusation.

A tricky detail is determining how long to wait before issuing an accusation. Using a static global timeout is not a good choice, as this will not scale well and can cause correct members to accuse other correct members more often than necessary, complicating the defense against Byzantine attacks. The timeout period has to adapt to the message loss characteristics between monitor and monitoree.

In *Fireflies*, the members estimate the probability of message loss. For this, we model ping-pong as a negative binomial experiment, and use exponential smoothing. The resulting protocol can be shown to be effective.

3 Applications

Because correct and Byzantine members cannot be distinguished (unless the Byzantine members give themselves away in a trivial manner), there are clear limitations to what *Fireflies* can offer. Also, views trail membership changes, and may be stale at any time. The question then is whether *Fireflies* can be put to use.

As a first example, an intrusion-tolerant Distributed Hash Table can be trivially implemented on *Fireflies*, simply by routing messages for an object identifier to the member in the view with the closest member identifier. Other DHTs provide its members with only a partial view of the membership in order to increase scalability, and messages sent between members often follow multiple hops through the membership. In *Fireflies*, messages are less likely to get lost (or deliberately dropped) along the way and encounter lower latency.

A more interesting use of *Fireflies* is to build an intrusion-tolerant multicast protocol. Our protocol is heavily based on Chainsaw [5], which floods each message on the neighbor mesh. Flooding is done efficiently: when a member receives a large message, it notifies its neighbors only of the message identifier. Each member collects such notifications from its neighbors and requests the message from one. If no response is received within a short period of time, another neighbor is selected. Measurements on Chainsaw have shown that this protocol is as efficient as the best multicast protocols based on DHTs [5].

Because the neighbor mesh connects all correct members, a message from a correct member is guaranteed to be delivered to all correct members. In order to prevent forging, members sign messages and check signatures before accepting received messages. Also, in order to discourage free-loading, correct members prefer uploading messages to neighbors from which they recently received a message.

4 Related Work

The first paper that describes concrete defenses against Byzantine behavior in peer-to-peer (P2P) network overlays is [1]. While this paper addresses the problem of impersonation attacks, many of the problems discussed have to do exclusively with routing table maintenance and message forwarding. In our protocol, the members do not need to route messages, while in [1], the problem of membership is not considered.

The problem of intrusion-tolerant membership in P2P protocols is considered in [6]. The *Eclipse attack* is an attack where malicious members isolate correct members, by filling the neighbor table of a correct member with addresses of malicious members. The paper suggests thwarting this attack by enforcing bounds on the in- and out-degrees of P2P members.

Most closely related to our work is the SWIM protocol [2]. Unlike *Fireflies*, SWIM's failure detection protocol does not adapt to varying message loss, and SWIM is not tolerant of Byzantine behavior.

The SCAMP protocol [3] is an epidemic-style membership algorithm that uses a small number of gossip partners per member in order to increase scalability. SCAMP is not intrusion-tolerant, and does not have a failure detection component. Members have to leave the group explicitly by gossiping a message.

There has been a variety of work on intrusion-tolerant epidemic protocols, apparently starting with [4]. These protocols consider the problem of correct members not accepting any updates in the absence of unforgeable signatures, and use a form of voting.

5 Conclusion

We have presented *Fireflies*, a weakly-consistent, scalable protocol that supports overlay networks and tolerates Byzantine members with high probability. *Fireflies* may be used as an intrusion-tolerant Distributed Hash Table supporting data sharing, or for building intrusion-tolerant overlay routing networks, or simply to organize computer resources in, say, a wide-area computational or storage grid.

Acknowledgements

Fireflies has been built and deployed on PlanetLab by Håvard Johansen of the University of Tromsø and has been running since February 2005. Håvard is working on further evaluations. André Allavena has proven various properties of the *Fireflies* protocol correct, and is working on a complete correctness proof. The multicast protocol has been implemented and is being evaluated and further improved by Maya Haridasan.

References

- [1] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D.S. Wallach. Secure routing for structured peer-to-peer overlay networks. In *Proc. of the 5th Usenix Symposium on Operation System Design and Implementation (OSDI)*, Boston, MA, December 2002.
- [2] A. Das, I. Gupta, and A. Motival. SWIM: Scalable Weakly-consistent Infection-style process group Membership. In *Proc. of the Int. Conf. on Dependable Systems and Networks DSN O2*, pages 303–312, Washington, DC, June 2002.
- [3] A.J. Ganesh, A.-M. Kermarrec, and L. Mas-soulié. SCAMP: Peer-to-peer lightweight membership service for large-scale group communication. In *Proc. of the 3rd International Workshop on Networked Group Communication*, London, UK, November 2001.
- [4] D. Malkhi, Y. Mansour, and M.K. Reiter. On diffusing updates in a Byzantine environment. In *Symposium on Reliable Distributed Systems*, pages 134–143, Lausanne, Switzerland, October 1999.
- [5] V. Pai, K. Kumar, K. Tamilmany, V. Sambamurthy, and A.E. Mohr. Chainsaw: Eliminating trees from overlay multicast. In *Proc. of the 4th Int. Workshop on Peer-To-Peer Systems*, Ithaca, NY, February 2005.
- [6] A. Singh, M. Castro, P. Druschel, and A. Rowstron. Defending against Eclipse attacks on overlay networks. In *Proc. of the 11th European SIGOPS Workshop*, Leuven, Belgium, September 2004. ACM.