Probabilistic Semantically Reliable Multicast
(Extended Abstract)

José PEREIRA* *Universidade do Minho* *jop@di.uminho.pt*
Luís RODRIGUES* *Universidade de Lisboa* *ler@di.fc.ul.pt*
Rui OLIVEIRA* *Universidade do Minho* *rco@di.uminho.pt*
Anne-Marie KERMARREC *Microsoft Research, Cambridge, UK* *annemk@microsoft.com*

1. Introduction

Traditional reliable broadcast protocols fail to scale to large settings [13, 1]. This paper proposes a reliable multicast protocol that integrates two recent approaches to deal with the large-scale dimension in group communication protocols: gossip-based probabilistic broadcast [2] and semantic reliability [10]. The aim of the resulting protocol is to improve the resiliency of the probabilistic protocol to network congestion by allocating scarce resources to semantically relevant messages.

Although intuitively it seems that a straightforward combination of probabilistic and semantic reliable protocols is possible, we show that it offers disappointing results. Instead, we propose an architecture based on a specialized probabilistic semantically reliable layer and show that it produces the desired results. The combined primitive is thus scalable to large number of participants, highly resilient to network and process failures, and delivers a high quality data flow even when the load exceeds the available bandwidth.

We present a summary of simulation results that compare different protocol configurations. The details of the simulation model are given in the full version of the paper [12] which presents the evaluation of different implementation policies and the study of relevant system parameters in the performance of the protocol.

2. Background

2.1. Probabilistic broadcast

In probabilistic protocols [5, 8] messages are not disseminated in a deterministic manner. Instead, messages are disseminated by gossiping: each participant relays each message received to a random subset of processes, and each message is relayed at most a bounded number of times. It can be shown that these protocols deliver messages to none or all processes with a high probability. The probability of a message being delivered to some but not all processes can be made as small as required by adjusting the protocol's parameters. The decentralized nature of epidemic dissemination results in a protocol that is scalable to a large number of participants without overloading any particular member of the group.

In addition to their performance in the large scale, gossip-based protocols are highly resilient both to process failures and to independent packet losses in the network. However, these protocols do not tolerate correlated message losses, such as resulting from network congestion. Unless there is a fixed set of bounded rate senders, the maximum input rate has to be set conservatively thus preventing full network usage. This is a potential weakness of the approach since with probabilistic protocols the network usage is notably high in comparison with deterministic reliable broadcast protocols.

2.2. Semantic reliability

Deterministic protocols have to retransmit the messages until all recipients acknowledge their reception or are declared failed. When the network is congested or a receiver is perturbed, messages accumulate in buffers until fully acknowledged. Since buffers are bounded, when buffers fill up the sender is not allowed to send further messages. This means that a single slow receiver or network link may slow down the entire group.

Semantic reliability [10, 11] stems from the observation that in distributed applications many messages are made obsolete by messages sent shortly after. As such, when messages accumulate in buffers it is likely that already obsolete messages are in the same buffer as the messages that make them obsolete. If obsolete messages are purged, the resulting buffer space can be immediately reused ensuring that the sender is never blocked and thus fast receivers are not affected. This allows receivers with different throughputs to be accommodated within the same group: Fast receivers get all the messages with low latency and slower receivers get less messages with higher latency.

The performance of semantic reliability depends both on the obsolescence profile of the traffic and on the size of

*Partially funded by FCT SHIFT project (POSI/32869/CHS/2000) and by a Microsoft grant.

buffers available for purging. A simple analytical model that enables reasoning about the efficiency of the protocol and the configuration of system parameters according to the obsolescence function of the target application has been proposed in [10]. This model was validated through simulation. It was shown that when applied to a traffic profile of an on-line trading system, the protocol can be configured to tolerate a receiver exhibiting processing delays 40% higher than those required to process all messages in due time. Using a formal definition of semantic reliability, a similar model was also proposed to reason about the advantages of the approach to support strongly consistent replication [11].

3. Challenges in combining both approaches

The straightforward manner of obtaining a probabilistic semantic protocol would be to apply semantic purging to the buffers maintained internally by the probabilistic protocol. However, some probabilistic protocols, such as pbcast [5], do not rely on buffering and that prevents semantic purging from being applied. Thereby the use of purging would be restricted to protocols that use buffering such as lpbcast [4] in which it is hard to balance the buffering requirements of semantic purging with the configuration of the probabilistic protocol for optimum performance.

The reason for the poor performance of such a naive combination of both protocols is the following: Consider that the gossip protocol is statically configured to use small buffers. In such case it is hard to apply obsolescence relations since it unlikely that related messages can be found simultaneously in the buffer. On the other hand if the protocol is statically configured to use large buffers to increase the chances of applying semantic purging, then the latency of message dissemination increases since messages are buffered for longer periods at each hop. The likelihood of overloading the network when the buffers are flushed is higher, which may cause correlated losses at the network level. Furthermore, even if a suitable buffer configuration that allows for purging when the system is overloaded can be found, excessive purging will occur also when there is enough network bandwidth to disseminate all the messages.

Simply adding semantic purging to gossip-based protocols does not enable the system to react efficiently to changes in the network load. Therefore, it is necessary to design a solution that allows purging to be applied as a function of the network usage.

4. Probabilistic semantically reliable multicast

The architecture proposed in this section stems from the observation that probabilistic broadcast protocols do not tolerate network congestion as it induces correlated loss. This can be avoided by a congestion control mechanism and buffering, allowing the system to cope with short load peaks. If the system is subject to a sustained high load, it is likely that messages have to be buffered long enough so that obsolete messages can be recognized and subsequently purged. Resources saved by purging obsolete messages can then be applied to ensure that the delivery of non-obsolete messages is done. Therefore, even if, like in any other probabilistic broadcast protocol, all messages cannot be delivered, the protocol selectively tries to deliver those messages that are required for consistent operation. On the other hand, when the network is not overloaded messages do not need to be buffered and the dissemination latency is not affected by the introduction of semantic purging.

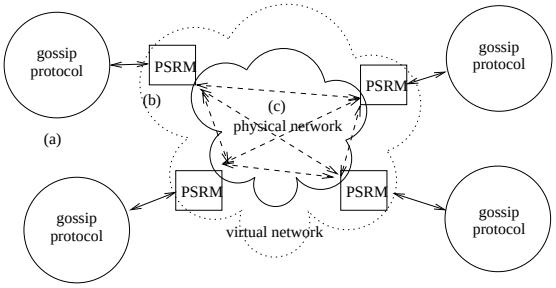

Figure 1. Three-tiered architecture: (a) Gossip protocol; (b) PSRM; (c) network.

The purpose of our architecture is to separate the mechanism exploiting message semantics both from the probabilistic protocol and from the physical network. It consists of inserting a specialized probabilistic semantically reliable multicast layer in between the gossip protocol and the network. The layer takes into account the available bandwidth at each link and performs buffering only to avoid overloading the network. Figure 1 presents a graphical overview of the proposed architecture: The lower tier (c) is the physical network. The middle tier (b) is a buffering layer, where any purging mechanism may be applied. The top tier (a) is an unmodified gossip protocol.

The lower tier is the physical network which, by controlling admission of messages, provides bounded independently distributed message loss among each pair of participating processes. This ensures that, when the system is overloaded, message losses do not happen inside the network but at the edges under control of a purging policy that minimizes their consequences. The concrete congestion control mechanism used depends on the characteristics of the physical network and is out of the scope of this paper. Possible options are a window-based congestion control mechanism compatible with TCP/IP [6] and rate-based congestion control in ATM networks [7].

The middle tier is our specialized probabilistic semantically reliable multicast layer: it buffers messages when the capacity of the network is exceeded and, in this case, it selectively drops obsolete messages to accommodate each

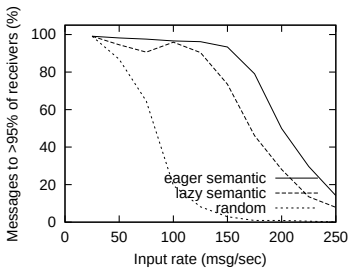

Figure 2. Atomicity of non-obsolete messages.

newly arrived message. This optimizes system usage by forcing correlated loss of obsolete messages and avoiding loss of non-obsolete messages. Therefore, this layer offers a virtual network which when overloaded provides independent loss preferably for non-obsolete messages.

A key issue in the configuration of this layer is the purging policy. Two options are considered:

- Eager semantic purging. In this approach, the reception of every message triggers semantic purging and every obsolete message, if any, is immediately removed from the buffers.
- Lazy semantic purging. In this approach, purging occurs only when the buffer is full and a new message arrives.

The top tier is a standard gossip-based broadcast protocol. Each message broadcast or locally received is forwarded to a subset of processes randomly chosen. Important parameters are the cardinality of this subset (also called the fan-out) and the maximum number of times each message is relayed (called the number of rounds). The configuration of such parameters is out of the scope of this paper and has been described in [5, 8].

5. Evaluation

As noted before, the combination of semantic reliability with probabilistic protocols does not aim at reducing the number of message losses when the network is congested. If the load exceeds the network capacity either the source is controlled or a number of message drops will inevitably occur. The purpose of our protocol is to create the conditions to drop more obsolete information than up-to-date information, thus improving the quality of the information provided to the users according to the semantics of the application.

A good metric of the quality of the information delivered to the users in a system where a degree of message obsolescence exists is to count the loss of messages that never become obsolete. This is an interesting metric in the sense that messages that don't ever become obsolete are the absolute minimum messages that have to be reliably delivered in order to ensure consistency.

Therefore, the primary evaluation criterion is that non-obsolete messages are atomically delivered according to the probabilistic protocols metric (i.e. either to almost all or to almost none recipients) and that those messages are given higher priority than obsolete messages (i.e. if the total amount of non obsolete messages is less than system capacity then all non-obsolete messages are delivered). Results for a particular configuration are shown in Figure 2, compared to a random selection of message to be dropped as used in other gossip-based protocols. Notice that the combined protocol tolerates a higher input flow without loss of atomicity of non-obsolete messages. A detailed description of the experimental conditions as well as other results are presented in the full paper [12].

We have also compared different purging strategies, namely eager and lazy purging with a pure gossip protocol without semantic purging, and concluded that eager purging represents a significant advantage in all performance metrics considered. Contrary to intuition, the eager strategy does not result in unnecessary purging when the system is not congested as in that situation low buffer occupancy automatically disables purging.

Secondary criteria are reducing latency when the system is not overloaded (i.e. messages are not arbitrarily delayed in order to allow purging) and that system configuration is robust (i.e. performance is stable despite variation of traffic and system parameters).

In contrast to the naive combination described in Section 3, our three tier architecture only increases latency when the input load exceeds the available network capacity, as otherwise there is no noticeable occupancy of message buffers.

We have also shown that this configuration of the PSRM layer is robust in face of changing traffic diversity, which makes deployment easier due to less strict configuration requirements.

6. Targeted applications

In publish-subscribe systems, subscribers register their interest in an event or a pattern of events in order to be subsequently notified of any event published which matches their interest regardless of how and by whom that information is produced. Many variants exist and differ mainly by their ability to filter events [3].

Message dissemination may take the subscription pattern into account. For instance, in content-based publish-subscribe systems the structure of the message can be inspected by the protocol in order to route messages from publishers to subscribers in situations where interest is sparse, thus avoiding flooding the network. However, in situations where interest in events is dense, flooding is the best option [9]. In this situation knowledge of message contents by the protocol can be used to improve flooding itself by using a semantically reliable protocol. This makes the

protocol described in this paper ideal for publish subscribe applications for very large numbers of publishers and subscribers with uniformly distributed interest.

In addition, applications supported by such publish-subscribe systems are likely to exhibit enough message obsolescence to obtain meaningful purging rates. For example, a information dissemination application for the stock-market would benefit from message obsolescence as strict reliability is expendable for stable high throughput [13, 1]. The semantic obsolescence may be carried by the absolute time a message had been sent. Any message carrying a more recent value (either absolute, either from the same publisher) would obsolete any older message.

Likewise, a large-scale auction system hosted by a publish-subscribe system could implement a very natural obsolescence semantic. Message carrying a bid higher than that of any other message would obsolete the latter one. In this case, the semantic information is not computed locally but is carried by the message itself, the bid absolute value as the parameter to define obsolescence relations. Some other causal criteria might be useful in this context: if p publishes m, m may obsolete all messages received by p before firing m.

7. Conclusions

Recently, gossip-based and semantically reliable protocols have independently emerged as appropriate solutions to cope with large number of participants in group communications. For scalability, gossip-based protocols rely on a peer-to-peer interaction model and they provide a probabilistic guarantee of delivery. In semantic reliability, semantics of the application is used to accommodate receivers and network links with heterogeneous performance within the same group.

In this paper we have proposed to combine them. Using semantic knowledge in gossip-based broadcast protocols enables to increase the quality of the information delivered to the users in the cases the load exceeds the available bandwidth for applications exhibiting obsolescence patterns. The paper presented and evaluated different strategies to combine the two protocols. We have shown that a three tier architecture, that uses a specialized PSRM layer that intercepts the interactions between the gossip protocol and the network can provide good results if configured appropriately. The best configuration uses a congestion control mechanism to prevent correlated loss inside the network and eager semantic purging on each of the resulting buffers. We have shown that the quality of the delivery, measured as the percentage on non-obsolete information that is delivered to almost all of the intended recipients, is better for such configuration than for other approaches, and much better than relying on random drop upon network congestion.

Obviously, the efficiency of the protocol depends of the obsolescence patterns of applications. However, when appropriate, the combined protocol performs better that a standard gossip-based protocol in congested networks without deteriorating it under normal circumstances. Especially, it enables to extend the applicability of gossip-based protocols to congested networks whereas such configurations are hardly addressed in the design of gossip-based protocols.

References

[1] K. Birman. A review of experiences with reliable multicast. *Software Practice and Experience*, 29(9):741–774, July 1999.

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

[3] P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. Technical Report DSC ID:2000104, EPF Lausanne, 2001.

[4] P. Eugster, R. Guerraoui, S. Handrukande, A.-M. Kermarrec, and P. Kouznetsov. Lighweight probabilistic broadcast. In *Intl. Conf. on Dependable Systems and Networks (DSN'2001)*, 2001.

[5] M. Hayden and K. Birman. Probabilistic broadcast. Technical Report TR96-1606, Cornell University, Computer Science, 1996.

[6] V. Jacobson. Congestion avoidance and control. *ACM Computer Communication Review; Proceedings of the Sigcomm '88 Symposium in Stanford, CA, August, 1988*, 18(4):314–329, 1988.

[7] R. Jain. Congestion control and traffi c management in ATM networks: Recent advances and A survey. *Computer Networks and ISDN Systems*, 28(13):1723–1738, Oct. 1996.

[8] A.-M. Kermarrec, L. Masssouli´e, and A. Ganesh. Reliable probabilistic communication in large-scale information dissemination systems. Technical Report 2000-105, Microsoft Research, 2000.

[9] L. Opyrchal, M. Astley, J. Auerbach, G. Banavar, R. Strom, and D. Sturman. Exploiting IP multicast in content-based publish-subscribe systems. In J. Sventek and G. Coulson, editors, *Middleware 2000*, volume 1795 of *LNCS*, pages 185–207. Springer-Verlag, 2000.

[10] J. Pereira, L. Rodrigues, and R. Oliveira. Semantically reliable multicast protocols. In *Proceedings of the Nineteenth IEEE Symposium on Reliable Distributed Systems*, pages 60–69, Oct. 2000.

[11] J. Pereira, L. Rodrigues, and R. Oliveira. Enforcing strong consistency with semantically reliable multicast. Technical Report DI/FCUL TR-2001-02, Department of Computer Science, University of Lisbon., June 2001.

[12] J. Pereira, L. Rodrigues, R. Oliveira, and A.-M. Kermarrec. Probabilistic semantically reliable multicast. Technical report, Department of Computer Science, University of Lisbon., Aug. 2001.

[13] R. Piantoni and C. Stancescu. Implementing the Swiss Exchange Trading System. In *Proceedings of The Twenty-Seventh Annual International Symposium on Fault-Tolerant Computing (FTCS'97)*, pages 309–313. IEEE, June 1997.