Latency-Efficient Atomic Broadcast for WANs

Dan Dobre, Matthias Majuntke, Neeraj Suri
{dan, matze, suri}@deeds.informatik.tu-darmstadt.de
Technische Universität Darmstadt, Germany

1 Context and Motivation

Fault-tolerant state-machine replication has garnered strong interest as a means of enhancing the availability and reliability of distributed systems. A fault-tolerant state machine is implemented by a collection of deterministic servers, which starting from the same initial state and given the same sequence of client requests, go through the same sequence of state transitions. The difficulty of implementing a replicated state machine lies in ensuring that (1) all correct replicas agree on the *sequence* of client requests they deliver (consistency) and that (2) every client request is actually delivered (progress), known as the *atomic broadcast* problem. The problem of agreeing on a sequence of requests further reduces to executing a sequence of agreement instances on each *single* request, known as the *consensus* problem, a fundamental building block for constructing distributed systems.

In the consensus problem, a set of processes need to agree on a single output value which is the input value of one of the processes. Fault-tolerant consensus has been shown to have no solution in the asynchronous system model [FLP85]. Even when using the most benign fault model, the crash/stop model, there is no algorithm satisfying all its properties. However, as real systems rarely behave asynchronously, failure detectors and leader election services [MOZ05] have been built, to solve consensus in practice and hence to implement atomic broadcast.

While in the context of benign faults, it is reasonable to assume that the necessary synchrony conditions eventually hold "long enough" for consensus to terminate, in the byzantine fault model one could argue that the adversary can emulate perpetual asynchrony, thus impeding the system from making progress. Therefore, fully asynchronous atomic broadcast protocols that do not rely on any synchrony assumptions for correctness and terminate with probability 1 have been the focus of recent research [KS05, HVR05].

In order to increase the responsiveness of fault-tolerant replicated services, the efficiency of atomic broadcast has recently garnered strong interest in the dependability community [CL02, JPM05]. Additionally, in order to overcome catastrophic failures of a single site such as power outages, intrusions and Denial of Service, maintaining replicas at geographically dispersed sites becomes a stringent requirement. Therefore, designing efficient atomic broadcast protocols for Wide Area Networks (WANs), an enviroment where *both* the message complexity and the link latency have a significant impact on the overall system performance [BK02], becomes a challenging problem.

2 Problems and Issues

In the context of state-machine replication in the client/server model, two metrics are of particular interest: (i) *time complexity* given by the number of communication steps needed to produce a valid response and (ii) *message complexity*, the total number of point-to-point messages generated (by the correct processes) for that response.

In byzantine fault-tolerant replication, there is a well known tradeoff between time and message complexity. Decreasing the value of one parameter consequently increases the value of the other. This problem is characteristic to the plethora of atomic broadcast protocols which can be roughly classified as either optimized for latency or message complexity. While the former has been studied quite well, e.g. [CL02, JPM05, KS05], the latter has called attention only recently [HVR05]. On a WAN, it is not well understood whether time or message complexity is the main factor determining the overall latency of a protocol. However, as shown in [BK02], on a WAN, relayer-based centralized communication schemes with only $\mathcal{O}(n)$ messages often outperform all-to-all message exchange patterns with message complexity $\mathcal{O}(n^2)$. Motivated by this result and following the lines of [HVR05], our goal is to develop a family of latency-efficient atomic broadcast protocols tailored for the use in WANs.

A second problem that has been overlooked for quite a long time is that the overall latency of a protocol might be affected by its resilience. By resilience we mean the maximum number of faults despite which correctness can be ensured. As demonstrated in [JPM05], for a fixed number of nodes n, the time complexity of a protocol can be optimized only at the cost of lower resilience. Thus, in order keep the resilience constant, more nodes are needed, and thus the quorums become larger.

As the number of critical links increases with the quorum size, the likelihood of having a really slow critical link also increases. Therefore, the larger the quorums, the larger become the individual communication rounds. Hence, for protocols designed for WANs, it is questionable whether the traditional efficiency metrics are representative enough. The natural question that arises is how to determine the conditions for which latency-optimized protocols actually outperform resiliency-optimized ones and to use this information to find better metrics for analyzing their efficiency.

3 Some Possible Solutions

Having as overall objective the improvement in responsiveness of services replicated over WANs via atomic broadcast and having message frugality as the primary constraint, our target is threefold. The protocols we develop must (a) be tight with respect to the number of communication steps and fault resilience, (b) must not employ expensive public key cryptography and (c) must be as latency-efficient as their optimized $\mathcal{O}(n^2)$ counterparts given the ability to recover the state prior to the atomic delivery of a payload. All three properties make atomic broadcast protocols with message complexity $\mathcal{O}(n)$ attractive as a candidate for the use in WANs.

Given the vast amount of byzantine fault-tolerant atomic broadcast protocols in the literature [HVR05, CL02, KS05, JPM05], to name only a few, raises the need for a single unifying framework with the ability to accomodate several protocols. Our objective is to design such a framework by exploring the weakest conditions under which the desired properties are still achievable. By using fine-grained communication abstractions that capture exactly these conditions we can encapsulate most of the differences at the communication level. Consequently, different implementations of the communication abstraction lead to different properties of the overall protocols, thus synthethising known protocols from the literature. Futhermore, we believe that a unified framework for atomic broadcast enables understanding the subtle similarities/differences among the large number of existing protocols and to design new protocols.

References

[BK02] Omar Bakr and Idit Keidar. Evaluating the running time of a communication round over the internet. In *Proc. of PODC*, 2002.

[CL02] M. Castro and B. Liskov. Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM (TOCS)*, 20(4), 2002.

[FLP85] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *JACM*, 32, 1985.

[HVR05] C. Cachin H. V. Ramasamy. Parsimonious Asynchronous Byzantine-Fault-Tolerant Atomic Broadcast. In *Proc. of OPODIS*, 2005.

[JPM05] Lorenzo Alvisi Jean-Philippe Martin. Fast byzantine consensus. In *Proc. of DSN*, 2005.

[KS05] Klaus Kursawe and Victor Shoup. Optimistic Asynchronous Atomic Broadcast. In *Proc. of ICALP*, 2005.

[MOZ05] Dahlia Malkhi, Florian Oprea, and Lidong Zhou. Omega meets paxos: Leader election and stability without eventual timely links. In *Proc. of DISC*, 2005.