Scalable Group Composition with End-to-end Delivery Semantics

Scott Johnson, Farnam Jahanian and Jigney Shah
Dept. of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2122
{scottdj,farnam,jiggey}@eecs.umich.edu

Group communication is a widely studied paradigm for building fault-tolerant distributed systems. A number of group multicast protocols have been developed which allow a set of processors to communicate using various semantics such as FIFO, causal, or total ordering, while providing atomicity and timeliness guarantees. In order to improve scalability and flexibility, it is often desirable to be able to compose a system from multiple cooperating process groups, each of which may provide specific service guarantees appropriate for a subset of the processes in the system. Our research focuses on the problem of building scalable, fault-tolerant distributed systems from collections of communicating process groups while maintaining well-defined end-to-end delivery semantics.

Our approach to inter-group communication takes advantage of modular group composition. We have developed a generic architecture which distinguishes between intra-group and inter-group communication. Our model allows for independent composition of communication protocols and improves scalability by reducing the need for shared state between groups. Using this model, we have formally analyzed the end-to-end semantic behavior of messages in a number of composed systems, and determined exactly which delivery semantics a set of receivers will observe for a given set of messages in various topologies. Using our analysis, a system designer can compose groups into a larger distributed system such that desired end-to-end delivery semantics on messages sent between groups are enforced. More information on our analysis is available in [3].

Our model relies on an abstraction which we call an *intergroup router*. Intergroup routers provide a mechanism for the exchange of messages between groups. They allow process groups to communicate in a fault-tolerant manner with minimal exchange of group or message state information and without modification of the underlying communication protocols. A sample group composition using intergroup routers is shown in Figure 1. The architecture of the inter-group routers is shown in Figure 2a. Each router consists of a routing protocol that runs on top of the attached

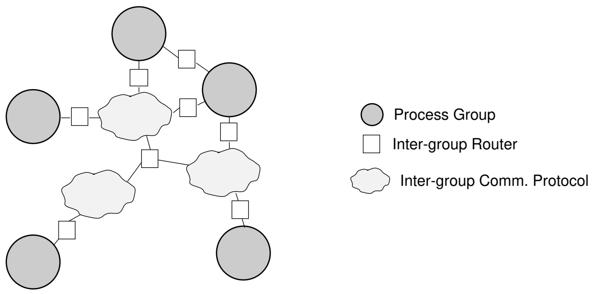

Figure 1. Example of a group composition using intergroup routers. Process groups are connected via intergroup routers and (optionally) one or more intergroup communication protocols.

communication protocols. This routing protocol is seen as a normal application by each communication protocol; it is a member of each process group and a regular sender/receiver in other types of communication protocols. It forwards messages between protocol stacks in FIFO order, ensuring that any message orderings enforced by a sending protocol are maintained as messages are forwarded across the distributed system.

To ensure that messages destined for other groups are forwarded to the intergroup router, a simple routing filter is inserted between the application and the group communication protocol on each node (Figure 2b). This filter uses the local group communication protocol to send intergroup messages to the intergroup router for forwarding, while allowing messaged destined for the local group to be sent and delivered normally. This enables the application to address messages to other groups without the local group communication protocol having to be aware of the rest of the composed system.

There are several benefits to this approach:

- Each group can reside on a separate physical network,

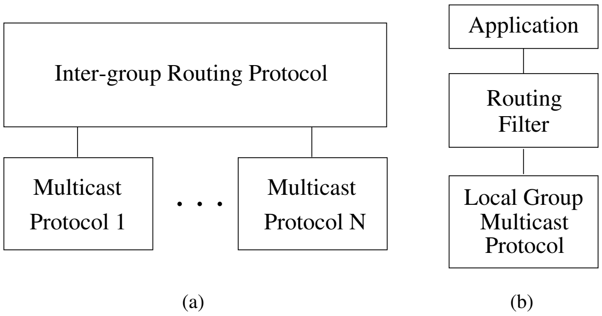

Figure 2. Intergroup router architecture

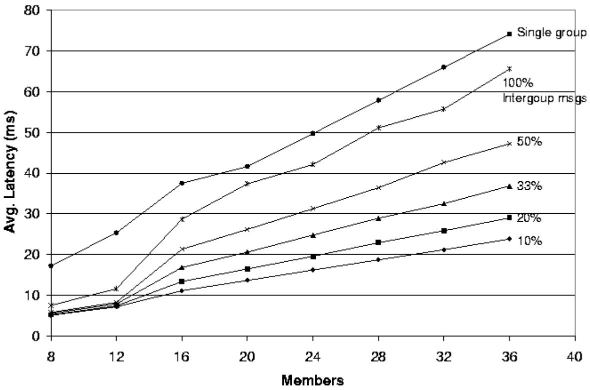

Figure 3. Simulation of one process group vs. four groups with varying percentages of intergroup messages.

or multiple groups can reside on the same physical network.

- Group communication protocols do not need to be modified for applications to be able to use the intergroup routers.
- Messages can be sent to/from closed process groups by non-members.

To test the scalability of our architecture, we conducted a number of simulations to compare the performance of a distributed system composed of multiple process groups using intergroup routers to that of a single process group containing the same number of processors as the composed system. The simulation was performed using OpNet, and was built using the actual implementation code from RTCAST, an atomic, totally ordered real-time group multicast protocol developed at the University of Michigan. Using data from our published performance tests of the RTCAST protocol [1], we tuned the simulation so that it yielded the same performance characteristics for a single group as the actual implementation of RTCAST. Since the performance of RTCAST compares very favorably with the published performance data of other group multicast protocols (most notably Totem [2] and Horus [4]), we expect that our simulation results will be applicable to systems built using other protocols besides RTCAST.

Figure 3 shows the results of one of our tests, comparing the average message latency of a single group to a system in which four RTCAST groups are composed around a single intergroup protocol with four intergroup routers. To mimic the behavior of a real distributed system, we varied the proportion of messages that were sent to all processors in the multiple group case from 10% to 100%. Messages that were not sent to all groups were delivered only in the sender's group. Note that the latency is highest for the single group case. Even when every message is sent to all processors, the multigroup system yields lower message latency than the single group case. This is likely because message latency is proportional to the group size, and once the group reaches a certain critical size it is faster to send the message through several smaller groups than one large one. Other simulations showed similar performance benefits from using multiple process groups.

In conclusion, we have shown how our abstraction of inter-group routers can support group composition while maintaining scalable performance and well-defined message delivery semantics. We are now developing a test application to utilize the intergroup routers and examine their behavior in a real distributed system. We are in the process of adding priority-based communication to our architecture, to provide better application control of end-to-end message latency.

References

[1] T. Abdelzaher, A. Shaikh, S. Johnson, F. Jahanian, and K. Shin. RTCAST: Lightweight multicast for real-time process groups. Technical report, Dept. of Electrical Engineering and Computer Science, University of Michigan, 1997. To appear in IEEE Transactions on Software Engineering.

[2] Y. Amir, L. Moser, P. Melliar-Smith, D. Agarwal, and P. Ciarfella. The Totem single-ring ordering and membership protocol. *ACM Transactions on Computer Systems*, 13(4):311–342, Nov. 1995.

[3] S. Johnson, F. Jahnaian, and J. Shah. Scalable group composition with end-to-end delivery semantics. Technical report, Dept. of Electrical Engineering and Computer Science, University of Michigan, 1998. Available at http://www.eecs.umich.edu/~scottdj/papers/group_composition.ps.

[4] R. van Renesse, K. Birman, and S. Maffeis. Horus: A flexible group communication system. *Communications of the ACM*, 39(4):76–83, Apr. 1996.