Membership Protocols for Conference Control

Bala Rajagopalan
AT&T Bell Laboratories
(braja@qsun.att.com)

3/30/1994

Conference Control Functions

- Floor Control
- Admission Policies
- Security
- Configuration & Negotiation
- Application Control
- Network Functions
- Synchronization
- Meta-Conference Management
- Others....

Membership management is the basis for at least the first four functions.

Some functions may require ordering of message from the source, but the need for more sophisticated ordering (eg.: total ordering of all messages) is not apparent.

Some Issues in Designing Membership Management Protocols

- Membership in the presence of failures

  Communication should persist among subsets of members, perhaps independently of other such subsets.

- Scalability

  As group size increases, unicast burden increase.

- Latency

  Communication latency may increase with group size.

Trade-off Between Scalability and Control: A Possible Conf. Control Framework

- Decide “type” of conference at initiation (appropriate control policies are selected)

- Gracefully loosen policies with increasing group size or terminate session

“type” could itself specify a control policy how the control policy changes with size (this may be difficult to achieve in practice).

How Difficult is the Membership Problem?

- Consistency of information at each member at every instant is impossible.

- Eventual consistency is possible under certain definitions of membership.

- Even eventual consistency may not be simple to achieve when certain anomolous network conditions occur (eg.: communication is not transitive).

- Membership protocols can work on top of unreliable multicast, but may also depend on unicast heavily.

- Correctness of new proposals may be hard to prove and understand.

We look at some sample protocols next.

A Membership Protocol

- Membership defined as a set of all members in the same network partition
- Distributed protocol maintains membership as a sequence of “views”
- View changes according to network and group dynamics
- Views at all members in the same partition converge eventually.

What this means to conference control

- Policies for handling sudden changes in view required
- View might change while implementing these policies
- Multiple, concurrent offshoots of the same session must be handled

Protocol Model and Assumptions

- Protocols: Join, Leave, Delete, Reformation and Merge
- Protocols are executed under the control of a coordinator. Any member can be coordinator
- Protocols can be executed with small latency
- Assumptions: Keepalive, Timestamps, Transitivity of multicast communication, Bi-directional connectivity.

Overheads

- Join: A multicast request + a multicast *Join* message + n unicast acks (n is the size of the group)
- Leave, Delete: Unicast request + one multicast *Leave/Delete* message + n acks
- Reformation: Unicast request + two multicast *Reformation* messages + $2n$ acks
- Merge: Unicast request + unicast membership exchange + multicast merge message + n acks
- Max Latency (best case): 4 x *end-to-end delay*

Token-Based Protocols

- Membership maintained as an ordered token list
- Token is passed periodically among members (sequentially)
- Failure to pass or receive token indicates member failure
- Join, Leave, Delete, Reformation and Merge can be implemented via token passing
- Examples: TPM [2], Transis [3], TRP [4] and RMP [5]
- Problem: Latency proportional to group size.

Future Work

Restricted membership models for conference control? These models should specifically accommodate real-world network anamolies

References

[1] B. Rajagopalan, “Membership Protocols for Wide-Area Group Communication,” Draft, 1994. (ftp: isi.edu, confctrl/bib/consensus.ps)

[2] B. Rajagopalan and P. K. Mckinley, “A Token-Based Protocol for Reliable, Ordered Multicast Communication,” Proc. IEEE Reliable Distributed Systems Symposiusm, 1989

[3] Y. Amir, et al., “Fast Message Ordering and Membership Using a Logical Token-Passing Ring,” Proc. IEEE 13th International Conf. on Distributed Computing Systems, 1993

[4] J. Chang and N. Maxemchuk, “Reliable Broadcast Protocols,” ACM Transactions on Computer Systems, 1984

[5] B. Whetten and S. Kaplan, “A High-Performance Totally Ordered Multicast Protocol,” Draft, 1994 (whetten@tenet.berkeley.edu)