

Abstract

Since, the send and receive of messages are the unique way enabling processes to cooperate and exchange data, then tracking causality between messages in the aim of efficient passing of messages with appropriate semantics is of great importance. To this end, the research community proposed the idea of ”*causal message ordering*”. The causal ordering concept has a considerable interest in the design of distributed systems and finds its application in several domains, such as management of replicated database updates, determining global snapshot, resource allocation, shared distributed memory, distributed virtual environments, teleconferencing, stock trading, collaborative applications, delivery in multimedia systems, etc.

As far as the mobile computing environment is considered, the design of causal ordering protocols must deal with the new characteristics of this environment, such as mobility, resource constraints on mobile hosts and the limited bandwidth of wireless links.

In distributed computing systems (either conventional or mobile), processes are often organized into groups for supporting various applications, such as computer-supported-cooperative work (CSCW), replicated services, news groups, etc. Group-based communication has proven an important paradigm for developing such distributed systems. In Group Communication Systems (GCS), causal ordering protocols are an essential tool to exchange information. However, achieving ordered delivery of messages for group communication in mobile environment is complicated by the fact that a group composition may change by the join of new members, the leave of existing members and the migration of members between cells.

In this work, we focused on *causal ordering* in mobile environments considering the different kinds of communication (unicast, broadcast and multicast) and dynamic groups.

The emerging trend towards multicast applications in which there is a need to communicate with several other hosts simultaneously (e.g. dissemination applications, collaborative applications, fault tolerant applications, conversational applications, etc.), drove us to make our first contribution. In this proposal, we tried to get benefit from the important characteristics of our unicast protocol (*Mobi-Causal*), such as elimination of unnecessary inhibition delay, low message overhead and scalability, in order to extend it