Event Order with Interval Timestamp in Event Correlation Service over Wireless Ad Hoc Networks

Eiko Yoneki and Jean Bacon
University of Cambridge Computer Laboratory
J J Thomson Avenue, Cambridge CB3 0FD, UK
{eiko.yoneki, jean.bacon}@cl.cam.ac.uk

Event correlation is an important service for event-based middleware which allows components in a publish/subscribe architecture to subscribe to event patterns rather than individual events. The diversity of network environments makes events flow from tiny sensor networks to Internet-scale peer-to-peer systems. One of the most difficult challenges in an event correlation is determining event order over distributed systems, which is more complex in such mixed network environments. This poster addresses event order in event correlation service over wireless ad hoc networks. In environment monitoring systems such as detecting the direction and speed of certain phenomena (e.g. a moving glacier or migrating birds), mobile computing devices with sensors and clocks can record the time an event occurs and communicate this information to other devices as they pass by. An event correlation service combines the information collected by the individual devices into higher level information or knowledge.

Temporal ordering of these events (a happened before b) originating from different devices and clocks has to be determined and real-time issues (a and b happened within a certain time interval) have to be solved. Logical time cannot be used to determine temporal ordering, because causal ordering of events in the real world must be honoured. Thus, physical time has to be used requiring clock synchronization. However, most of the synchronization algorithms rely on partitioned networks. Global infrastructures like GPS provide an accurate time base, but GPS is not suitable for resource-constrained devices.

We previously used an interval timestamp for event correlation [1]. A two-part interval timestamp represents the clock uncertainty, and intervals can include the estimated network delays. Further consideration is required to support wireless ad hoc networks. We propose an event ordering algorithm for real-time event correlation using an interval-based timestamp based on the time synchronization mechanism [2]. When a publisher node records an event in real-time it generates a timestamp using its unsynchronized local clock, which is passed to other nodes. The algorithm uses interval-based timestamps as a lower and upper bound for the exact value, and transforms them to the local time of the receiver instead of adjusting the clocks. Event delay is measured using the round trip time by exchanging another event (an acknowledgment or control event for maintaining the state may be used with no overhead). The timestamps are updated along its way by each node using its own local clock. As the result of clock shift and event propagation delay, the final timestamp is expressed as a lower and upper bound relative to the local time. This algorithm assumes that the maximum clock skew is known. Note that the interval timestamp is only necessary for the events that are monitored by event correlation services or specific subscription groups. The interval arithmetic is based on [1] with real-time consideration. For example, to determine if $[t_1^l, t_1^h]$ happened before $[t_2^l, t_2^h]$ the following formula can be used.

$$[t_1^l, t_1^h] < [t_2^l, t_2^h] = \begin{cases} YES: & t_1^h < t_2^l \\ NO: & t_2^h < t_1^l \\ MAYBE: & otherwise \end{cases}$$

To determine if $[t_1^l, t_1^h]$ and $[t_2^l, t_2^h]$ happened within certain real-time period R the following formula is used.

$$[t_1^l, t_1^h] - [t_2^l, t_2^h] < R$$

$$= \begin{cases} YES: & max(t_2^h, t_1^h) - min(t_2^l, t_1^l) < R(1-\rho) \\ NO: & max(t_2^l, t_1^l) - min(t_2^h, t_1^h) < R(1+\rho) \\ MAYBE: & otherwise \end{cases}$$

where ρ is maximum clock skew.

In general, the semantics of event order in event-based middleware is not defined clearly. Even the well established Java Message Service (JMS) only guarantees the event order within a session. The proposed approach will be integrated into our ongoing Event Brokering Project for wireless ad hoc networks and, when composite events flow towards wide area networks, interval timestamps will be federated. Event correlation helps the data aggregation task, and reduces complexity by moving functionality to the middleware and integrating mixed/hybrid network environments.

Acknowledgment. This research is funded by EPSRC (Engineering and Physical Sciences Research Council) under grant GR/557303.

REFERENCES

[1] P. Pietzuch, B. Shand, and J. Bacon. Composite event detection as generic middleware extension. *IEEE Network Magazine*, 2004.

[2] K. Roemer. Time synchronization in ad hoc networks. In *Proc. of ACM MobiHoc*, 2001.

ACM/IFIP/USENIX 5th International Middleware Conference Toronto Canada