Temporal Ordering of Wireless Sensor Events

Eiko Yoneki
University of Cambridge Computer Laboratory
Cambridge CB3 0FD, United Kingdom
eiko.yoneki@cl.cam.ac.uk

ABSTRACT

To manage high volume and faulty sensor data requires complex event filtering, aggregation and correlation over time and space in heterogeneous network environments. Event management is a multi-step operation from sources to final subscribers, combining information collected by wireless devices into higher-level information or knowledge. Temporal ordering in real-time is a critical aspect in such environments (e.g., object tracking). In this paper, we discuss issues on temporal ordering of sensor data.

EMERGENCE OF WSN

A rapid increase of event monitoring capability by wireless devices brought a further evolution in ubiquitous computing. This new paradigm is about networked microprocessors embedded in everyday objects surrounding us, talking to each other over wireless links. Wireless Sensor Networks (WSNs) revolutionise information gathering and processing. An important issue is that the sensed data need to be filtered, correlated, and managed at the right time and place when they flow over heterogeneous network environments (e.g., from WSNs to Peer-to-Peer systems in the Internet). Event management will be a multi-step operation from event sources to final subscribers, combining information collected by wireless devices into higher-level information or knowledge. Thus, middleware takes an important task to manage sensor data over global distributed systems.

Most extant approaches to define event correlation lack a formal mechanism for complex temporal relationships among correlated events. In [8], we introduced an unified semantics definition for event correlation, which combines traditional event composition and data aggregation in WSNs. Its focus is on supporting time issues such as temporal ordering, duplication handling, and interval-based semantics, especially for wireless network environments. In this paper, we discuss issues on temporal ordering of sensor data and introduce our interval-based timestamp approach.

TEMPORAL ORDERING

Temporal ordering in real-time is a critical aspect for event correlation in wireless networks. The context of real-time in this paper is the time of a real event occurrence. For object tracking, the direction of movement of a real world phenomenon has to be detected, thus, temporal ordering of events originating from different devices has to be determined. The event can also be triggered by real world phenomena. Neither logical time nor classical physical clock synchronization algorithms may be useful.

Multimedia synchronization for real-time tele-conferencing, and distributed network gaming require precise timing in distributed environments. When applications, such as virtual reality, require high precision tracking, the existing systems provide such precision through expensive infrastructure and hardware. For example, the Active BAT system [2] at Cambridge University requires an high accuracy and real-time tracking system. The timestamp is derived from a Global Clock Generator (GCG), which is a hardware clock that sends ticks to every component of the system over a serial link. When a location is computed, the location message is timestamped using GCG. The GCG delay is in the order of microseconds (speed of light), and the slowest part of the system is waiting for ultrasound to propagate (speed of sound, 100 million times slower) after a position is requested but before a position can be calculated. This delay is used to measure the distance between the BAT and the receiver at the ceiling. Once the location is calculated, the message then has to travel up to the tracking system (order of milliseconds).

In general ordering events in distributed environments is difficult without a global clock. Moreover when supporting the communication model of store-and-forward paradigm, message propagation delay is unavoidable. Traditional message ordering based on transport layer protocol is not applicable. Thus, timestamps embedded in events should be used for correlation. In existing systems, the semantics of event order often depends on the application logic. Even the established Java Message Service (JMS) only guarantees the event order within a session, a single threaded context that handles message passing.

Temporal ordering of events is highly influenced by the event detection method, timestamping methods and underlying time systems. For real-time support, a common solution in wired networks provides a virtual global clock that bounds the value of the sum of precision and granularity within a few milliseconds. Network Time Protocol (NTP) is an Internet standard for real-time mechanism. The NTP servers are synchronised by the external time sources typically using GPS. The NTP allows for assignment of real-time timestamps with given maximal errors. However, in open distributed environments, not all servers are interconnected and event ordering based on NTP may lead to false event detection. Interval-based time systems define event order based on intervals. In [4], timestamps of events can be related to the global reference time with bounded accuracy and event timestamps are modelled using accuracy intervals. They use NTP that provides reference time injected by GPS time server and in addition return reliable error bounds.

In WSNs, transmission delay introduces several hundreds of milliseconds by MAC layer at each hop, thus NTP is only suitable for applications with low precision demands. In [5] GPS based virtual global clock that is used for timestamping events is presented deploying a similar concept of 2g-precedence (see [7]). Without the existence of GPS, there is no mean to synchronise the clocks of all the nodes in a deterministic fashion. Post-facto synchronization [1] is based on unsynchronised local clocks but limits synchronisation to the transmit range of the mobile nodes. This model requires continuous message exchange even if no significant event may be reported. In WSNs, energy consumption has to be considered, and on-demand based approaches may fit better. For example, in [6], they take a similar approach of unsynchronised clocks. The idea of the algorithm is not to synchronise the local computer clocks of the devices but instead to generate timestamps from a local clock. When such locally generated timestamps are passed between devices, they are transformed to the local time of the receiving device.

In many real world scenarios, wireless networks may be deployed with relay nodes to Internet backbones; it is possible that relay nodes can connect to GPS. That makes us reconsider the use of GPS in distributed systems including wireless network environments in spite of GPS not being suitable for use in a large class of smart devices due to its high power consumption and the required line of sight to satellites. GPS may be the key for providing accurate time adjustment at certain nodes where they are less resource constrained within wireless ad hoc networks. Thus, we propose a coordinated approach with and w/o GPS environments. For the w/o GPS scenario, we propose a lightweight local clock propagation model described below. For the GPS accessible networks, where NTP and GPS can be deployed, we use interval-based time system described in [4]. Both approaches use interval-based timestamp representing inaccuracy of a single point time. The sensor events could be aggregated at gateway nodes with transformed timestamps and passed towards a subscriber node in the Internet environments, where GPS based time synchronisation is deployed.

Lightweight Local Clock Propagation: is on-demand based timestamp synchronisation. Fig.1 shows the operation from source nodes X and Y to the sink node S. The basic idea is that each node calculates process time using a local clock, and at the sink node, the sum of process time is subtracted from the event arrival time. The consistency of timestamps is kept at sink nodes instead network wide. The detail of algorithms and experiments are out of scope of this paper. This requires the following two assumptions:

- Network delay is negligible, where the node is close to the radio or with very dense network deployment. Thus, N nanoseconds over 100s meters path to sink.
- Clock drift is negligible, where the node carries an oscilloscope, that guarantees less than 10 ppm drift. One part per million (ppm) corresponds to 1 second in 11.5 days.

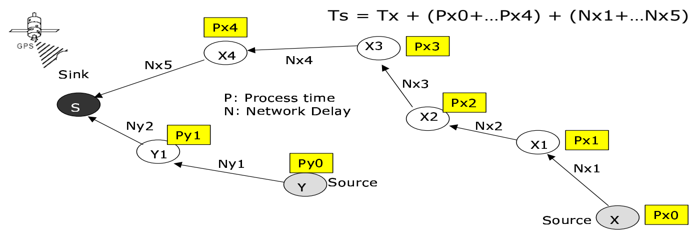

Fig.1: Lightweight Local Clock Propagation

The process time includes *Send Time*, *Access Time*, and *Receive Time* described in [3], and essentially it is an overhead between application layer and MAC layer. Thus, the proposed mechanism can be coordinated with MAC layer to eliminate a calculation overhead. This approach does not require an additional message for time synchronisation. An ultimate precision depends on the hop count of the route and on the total routing time.

CONCLUSION AND FUTURE WORKS

Our goal is providing a generic middleware architecture that supports an open platform for wireless sensor data management over heterogeneous network environments. Integration of event correlation will bring the system onto the next level, performing data aggregation and distributing filtered data based on contents. Our event correlation semantics supports a new paradigm, where more sophisticated event correlation over time and space is required on monitored events by wireless sensor devices. Temporal ordering of sensor data is a critical piece of information for applications. We are working on the complete implementation including various timestamping environments.

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

Bio. Eiko Yoneki is a PhD candidate in the Computer Laboratory at the University of Cambridge, UK. Her research interests are distributed computing systems over mobile/wireless networks, including event-based middleware and event correlation. Previously, she has spent several years with IBM (USA, Japan, Italy and UK) working on various networking products. She is a member of the ACM and IEEE Computer Societies.

REFERENCES

1. Elson, J. et al. Wireless Sensor Networks: A New Regime for Time Synchronization. *Workshop on Hotnets*, 2002.
2. Harter, A. et al. The Anatomy of a Context-Aware Application. *Proc. MobiCom*, 1999.
3. Kopetz, H. et al. Global time in distributed real-time systems. *Technical Report 15-89, Technische Universitat Wien*, 1989.
4. Liebig, C. et al. Event Composition in Time-dependent Distributed Systems. *Proc. 4th CoopIS*, 1999.
5. Prakash, R. et al. Causality and the Spatial-Temporal Ordering in Mobile Systems. *Mobile Networks and Applications*, 9(5):507-516, 2004.
6. Roemer, K. Time Synchronization in Ad Hoc Networks. *Proc. MobiHoc*, 2001.
7. Schwiderski, S. Monitoring the Behavior of Distributed Systems. *PhD thesis, University of Cambridge*, 1996.
8. Yoneki, E. and Bacon, J. Determination of Time and Order for Event-Based Middleware. *Proc. PerCom-MP2P*, 2005.