Time and Location in Sensor Networks

Kay Römer
Institute for Pervasive Computing
ETH Zurich
`roemer@inf.ethz.ch`

Abstract

Due to the close coupling of sensor networks to the real world, physical time and location play an important role in many sensor network applications. We explain why this is true and discuss several issues with time synchronization and sensor node localization for sensor networks.

1 Introduction

Recent advances in wireless communication and micro system technology allow the construction of so-called sensor nodes. Such sensor nodes combine means for sensing environmental parameters, processors, wireless communication capabilities, and autonomous power supply in a single tiny device. Large and dense networks of these untethered devices can then be deployed unobtrusively in the physical environment in order to monitor a wide variety of real-world phenomena with unprecedented quality and scale while only marginally disturbing the observed physical processes. Examples include monitoring the behavior of animals in their natural habitats, monitoring the spreading of environmental pollutions in air and water, and monitoring seismic activity and its influence on the structural integrity of buildings.

These exemplary applications demonstrate one important property of sensor networks: their inherent and close integration with the real world, with data about the physical environment being captured and processed automatically, online, and in real time. This has a number of conceptual and technical implications. One such implication is that physical time and location play a crucial role in sensor networks. To understand why this is true, let us consider the basic operation of a sensor network. The functionality of individual sensor nodes is rather simple, they typically measure environmental parameters (e.g., temperature, light intensity) at regular sampling intervals and apply certain filters to the obtained time series of sensor readings to detect "interesting" environmental conditions, emitting a so-called sensor event which describes the detected situation (e.g., the proximity of an observed object). In order to accomplish more complex tasks (e.g., estimating the velocity of a moving object), sensor events obtained from various nodes throughout the network have to be merged in a process called data fusion.

Physical time and location are of importance here for various reasons. Firstly, applications are often interested in time and location of occurrence of a sensor event. Secondly, filters for selecting "interesting" events often contain spatial and temporal constraints. Thirdly, time and location are often a crucial foundation for performing data fusion. Consider for example the estimation of the velocity of a moving object, which can be accomplished by considering the distance in space and time of two "object detection" sensor events obtained from distinct nodes of the sensor network.

For these purposes, nodes of the sensor network have to share a common reference system in time and space, requiring adequate means for time synchronization and node localization. While both of the latter have been examined before in various research contexts, the characteristics and requirements of sensor networks necessitate new solutions. In the following, we will briefly review these characteristics and requirements, before discussing more specific issues with time synchronization and node localization.

2 Sensor Network Characteristics

Individual sensor nodes should by very small (few cubic millimeters), long-living (months to years), cheap, and robust to environmental influences. The small size severely limits the onboard resources of a sensor node (energy, communication bandwidth, computing power, memory). To guarantee longevity despite the limited available energy, all hardware and software components must be consequently optimized for *energy efficiency*. Sensor networks are highly *dynamic*: sensor nodes die due to depleted batteries or harmful environmental influences, new nodes are added to replace failed ones, some nodes are mobile due to environmental factors like wind and water. Despite these dynamics, sensor networks should perform their task in a *robust* way. Due to their potential deployment in remote, inaccessible, or unexploited regions, sensor nodes must operate without excessive external infrastructure – forming so-called ad hoc networks. Many thousands of sensors may have to be deployed for a given task, an individual sensor's small effective range relative to a large area of interest makes this a requirement. Therefore, *scalability* is another critical factor in the design of the system. The expected scale makes manual configuration and management of individual sensor nodes impossible, thus sensor networks should be *self-configuring* and *self-managing*.

3 Time Synchronization

Energy, size, and cost constraints typically preclude equipping sensor nodes with receivers for time infrastructure like GPS [4] or DCF77 [15]. Also, logical time [5] is not sufficient, since it only captures causal relationships between "in system" events, defined by message exchanges between event-generating processes. In contrast, phenomena sensed by sensor nodes are triggered by external physical events which are not defined by in-system message exchanges; physical time must be used to relate events in the physical world.

Time synchronization services for traditional distributed systems like NTP [6] are typically based upon a manually configured hierarchy of network nodes. At the top of the hierarchy are one or more so-called master nodes – canonical sources of time which are synchronized to each other via some out-of-band mechanism such as GPS. Nodes further down in the hierarchy are synchronized to this global time scale by evaluating "time beacons" received from their immediate parent(s). Such beacon messages are frequently sent by a network node to its child nodes, containing the current clock-time of the parent at the time of message generation.

There are various problems with such an approach in the context of sensor networks. As noted above, equipping master nodes with infrastructure such as GPS receivers is typically not an option. In the case of one master (where no external infrastructure for out-of-band synchronization is required), synchronization paths tend to be very long due to the expected scale of sensor networks. This may lead to poor synchronization of nodes far away from the master node. Even worse, nodes which are close to each other, but are far away from the synchronization master, may experience a large synchronization error with respect to each other due to using different synchronization paths to the master with different synchronization quality. This can be a major problem, since co-located nodes tend to require accurate synchronization in order to correlate local sensor events.

Moreover, synchronization schemes like NTP are not optimized for energy efficiency. For example, the CPU is used continuously to perform frequency disciplining of the oscillator by adding small increments to the system clock. In addition, synchronization beacons are frequently exchanged, which also requires constantly "listening" to the network for such beacons. However, with low-power radios used in sensor networks, listening to, sending to, and receiving from the network all require significant amounts of energy. Also, the CPU may not be available if the processor is powered down to save energy.

The manually and statically configured synchronization topology used by NTP is not compatible with the network dynamics in sensor networks. The frequently changing network topology precludes static configuration, the unattended operation of sensor networks precludes manual configuration of individual nodes. Moreover, sensor networks are likely to be temporary partitioned due to node failures or environmental obstructions. Clocks in different partitions are poorly synchronized, which may lead to difficulties when trying to temporally correlate sensor events originating from different partitions after a rejoin of the partitions.

Many of the above problems can be solved by rethinking various aspects of a time synchronization service [3]. Energy efficiency, for example, can be significantly improved by exploiting certain characteristics of sensor network applications. Since sensor networks are typically triggered by physical events, sensor network activity is rather bursty than continuous and rather local than global. This leads to a situation, where synchronized clocks are only required occasionally and only for certain subsets of nodes. Also, the required synchronization accuracy heavily depends on the application, ranging from microseconds (e.g., for acoustic ranging with cm accuracy) to milliseconds or even seconds (e.g., for ordering infrequent events by time of occurence). One possible way to exploit these characteristics is called post-facto synchronization. There, unsynchronized clocks are used to timestamp sensor events. Only when two timestamps have to be compared by the application, they are reconciled to a common time scale.

The problems related to long synchronization paths with varying quality can be avoided by no longer trying to force all clocks of the system to adhere to a global time scale. Instead, local time scales with limited scopes should be established, with timestamps being transformed between scales when crossing a time scale boundary.

In [7] we present a synchronization scheme which adheres to the above principles. There, the unsynchronized clock of each node defines its own local time scale. Timestamps are generated according to this scale. Whenever a timestamp is sent to another node inside a message, a simple computation is used to transform the time stamp to the receiver's time scale. Synchronization can be piggybacked to existing message exchanges, thus keeping the energy overhead for synchronization to a minimum. Also synchronization works across (temporary) network paritions.

4 Node Localization

As with time synchronization, energy, size, and cost constraints typically preclude equipping sensor nodes with receivers for localization infrastructures like GPS. In extreme cases such as Smart Dust [13], it might not even be possible to equip sensor nodes with transceivers for radio waves or ultra sound due to the tiny size and energy budget of Smart Dust nodes. Hence, traditional ranging approaches such as ones based on time of flight of ultrasound signals or received radio signal strength might render unusable in the context of sensor networks.

Many localization systems such as [1, 12] depend on an extensive hardware infrastructure. Localization systems based on trilateration, for example, require many spatially distributed and well-placed infrastructure components in order to achieve high accuracy. For various reasons, this is not an adequate solution for sensor networks. Firstly, this contradicts the ad hoc nature of sensor networks, where nodes may have to be deployed in remote, inaccessible, or unexploited regions. Secondly, sensor nodes often need to know their own location, for example to filter sensor data based on spatial constraints as mentioned earlier in the paper. This may lead to a situation, where many nodes of the network regularly poll the infrastructure for their respective current location, leading to bad scalability of the system. For similar reasons, localization approaches requiring centralized computation such as [2, 11] do not scale well to large networks.

To overcome the limitations of infrastructure-based approaches, various schemes for ad hoc localization have been devised (e.g., [9, 10]). They are typically based on the assumption that few nodes of the network – so-called anchor nodes – know their exact location via some out-of-band mechanism. Other nodes derive their location by, for example, multilateration based on the distances to three or more neighbors with known locations. By iterating this process, all nodes of the network should eventually end up with three or more neighbors with known locations in order to be able to estimate their own location. To avoid accumulating errors inherent to such iterative approaches, many schemes calculate initial location estimates in a first round and iteratively improve these estimates in a second round. However, there are several problems with these approaches. Firstly, good location estimates are only obtained if each node has many neighbors, i.e., if the network is dense. But even then, nodes at the edges of the network tend to end up with poor estimates since they have fewer neighbors. Secondly, the iterative nature of many of the algorithms typically implies a high message overhead, leading to poor energy efficiency.

An important overhead involved in setting up a localization system is node calibration in order to enforce a correct mapping of sensor readings to location estimates

[14]. In systems based on radio signal strength, for example, the received signal strength is mapped to a range estimate. Variations in transmit power and frequency among the nodes can cause significant inaccuracies in the range estimates when used without calibration. Since the cheap low-power hardware used in sensor nodes typically introduces a high variability between nodes, sensor nodes have to be individually calibrated. This, however, may not be feasible in large sensor networks.

Some simple design principles can help solving the above problems. Localized location computation, where nodes autonomously estimate their location without consulting an infrastructure or relying on centralized computations, can help achieve better scalability. As with time synchronization, exploiting certain application characteristics can help improve energy efficiency. The required localization accuracy, for example, heavily depends on the application. In order to track the location of a moving object, for example, a localization accuracy in the order of the size of the tracked object is often sufficient. The calibration problem can be reduced by using differential measurements, where constant offsets cancel out due to using the difference between two measurements that use the same signal path.

In [8] we present a localization system suitable for large networks of tiny sensor nodes. This system consists of a single infrastructure device, which emits certain laser light patterns. By observing these patterns, sensor nodes can autonomously estimate their location with high accuracy. Since sensor nodes only passively observe light flashes, this system is very energy efficient on the side of the sensor nodes. Moreover, optical receivers consume only little power and can be made small enough to fit in a volume of few cubic millimeters. Since sensor nodes do not need to interact with other nodes in order to estimate their location, the system scales to very large networks. Also, sensor node calibration is not necessary due to using differential measurements.

References

[1] N. Bulusu, J. Heideman, and D. Estrin. GPS-less Low Cost Outdoor Localization for Very Small Devices. *IEEE Personal Communications*, 7(5):28–34, October 2000.

[2] L. Doherty, K. S. J. Pister, and L. E. Ghaoui. Convex Position Estimation in Wireless Sensor Networks. In *Infocom 2001*, Anchorage, Alaska, April 2001.

[3] J. Elson and K. Römer. Wireless Sensor Networks: A New Regime for Time Synchronization. *ACM SIGCOMM Computer Communication Review (CCR)*, 33(1):149–154, January 2003.

[4] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice, 4th Edition*. Springer-Verlag, 1997.

[5] L. Lamport. Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM*, 21(4):558–565, July 1978.

[6] D. L. Mills. Improved algorithms for synchronizing computer network clocks. In *Conference on Communication Architectures (ACM SIGCOMM'94)*, London, UK, August 1994. ACM.

[7] K. Römer. Time Synchronization in Ad Hoc Networks. In *MobiHoc 2001*, Long Beach, USA, October 2001.

[8] K. Römer. The Lighthouse Location System for Smart Dust. In *MobiSys 2003*, San Franscisco, USA, May 2003.

[9] C. Savarese, J. M. Rabaey, and K. Langendoen. Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks. In *USENIX Annual Technical Conference*, Monterey, USA, June 2002.

[10] A. Savvides, C. C. Han, and M. Srivastava. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In *Mobicom 2001*, Rome, Italy, July 2001.

[11] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz. Localization from Mere Connectivity. In *ACM MobiHoc 2003*, Annapolis, USA, June 2003.

[12] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

[13] B. Warneke, M. Last, B. Leibowitz, and K. S. J. Pister. Smart Dust: Communicating with a Cubic-Millimeter Computer. *IEEE Computer Magazine*, 34(1):44–51, January 2001.

[14] K. Whitehouse and D. Culler. Calibration as Parameter Estimation in Sensor Networks. In *Workshop on Wireless Sensor Networks and Applications (WSNA) 02*, Atlanta, USA, September 2002.

[15] DCF77 Radio Time Signal. www.dcf77.de.