Data in a Wireless Sensor Network: A View from the Field

Gail Mitchell
BBN Technologies

Introduction

A distributed (wireless) network of sensors can be viewed as a (very) large, distributed database. Each sensor generates, stores, maintains and/or accesses some subset of the network's data. Viewed as a whole, the data in the network ranges from constantly streaming input signals to binary code files; it represents the environment outside the network (i.e., the "sensed" environment) as well as the network environment itself. The management of this data is tightly inter-twined with an active, dynamic processing system that is 1) controlling the network, while 2) monitoring, and perhaps changing, the environment within which the network resides.

In this paper, I present some requirements for storing, managing and accessing data in a sensor network. These requirements are derived from personal observations in integrating and deploying a SenseIT distributed sensor network [1]. SenseIT is a DARPA-funded program with the goal of designing and developing advanced software technology for building ad hoc, multi-tasked, distributed sensor networks for tactical and surveillance operations. Some of the current work under SenseIT includes hardware development, sensor processing, network assembly, collaborative signal processing, communications and coordination, data dissemination, and querying and tasking. A recent implementation involving a variety of different organizations, integrated by BBN Technologies, provides real-time target (vehicle) localization, tracking, and classification for military force protection. In the rest of this paper I briefly describe this system and discuss some of the characteristics of data and requirements for its management in collaborative sensor networks.

A Sensor Network Environment

The sensor networks envisioned for the (near) future consist of large numbers of "intelligent" sensor nodes that, in addition to sensing capabilities, have processing, storage, and communications capabilities. For example, each prototype sensor node currently used in SenseIT includes a 167 MHz processor, a Linux OS, two independently assignable 2.4GHz radios, 64 MB of volatile memory and 64 MB of flash storage. This is in addition to sensor channels and digital signal processing components that collect and digitize the analog sensor input.

A defining characteristic of such sensor networks is that application processing is accomplished within the network. For example, consider the collaborative sensing application illustrated in Figure 1. All functions displayed, except the base station and gateway node, are available on each node of the network. In addition, one or more nodes of the network can act as a gateway to manage communications between the network and stations outside the network (aka, base stations). In this application, each node is individually responsible for analog to digital processing of its sensor input signals and for sensor-specific local processing (e.g., noise filters) and analysis to detect signals that could indicate vehicle presence. Nodes collaborate (CSIP – collaborative signal and information processing) to determine whether multiple detections indicate the presence of a target and, if so, to determine the direction, speed and type of target. All of this processing is dependent on the timely availability and processing of data: analog signals arrive at a rate (between about 200Hz and 20KHz) that may be predetermined or set by an application and must be processed accordingly; events are shared between nodes to aid in determining track information; classifying vehicles is "triggered" by events and track identification, yet requires raw or processed sensor data for identifying vehicle signatures; nodes require information about their physical location and neighbors' locations to aid in establishing collaborating teams and calculating track information; track data and processing move through the node network as the tracked vehicles move through the sensor field; information about tasks to be performed (i.e., queries) is moved from outside the sensor network, through a gateway, to task the appropriate nodes; information about the nodes, events and tracks moves from the network, through a gateway, to respond to user queries. Efficient and effective management of the network's data resources is a requirement for success in this environment

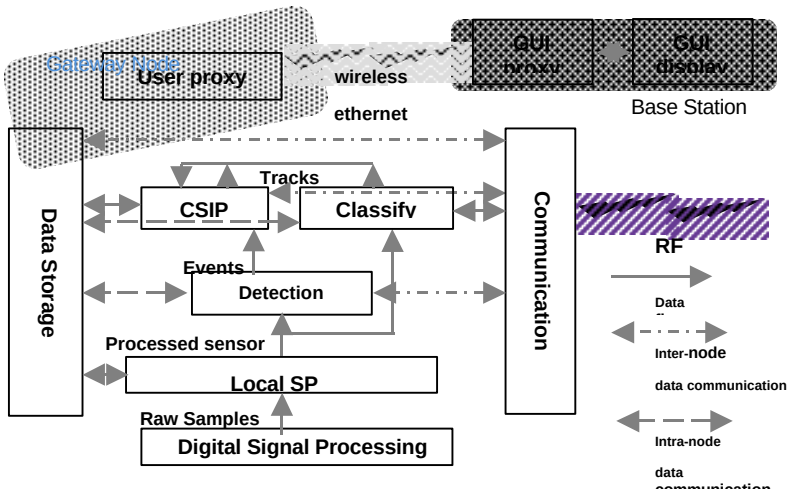

Figure 1. Collaborative Sensor Processing Application

A View from the Field

Processing in distributed, collaborative sensor networks must accommodate large amounts of volatile data (e.g., a SenseIT node can have as many as 4 sensors attached concurrently; a 20Hz input signal on one sensor produces over 40KB of raw data every second) with limited resources (e.g., storage, processing power, electrical power, communication bandwidth).

What kinds of data reside in a sensor network? As noted in the SenseIT example, data about the environment being monitored is acquired (via sensors), generated (e.g., events and tracks), and updated (e.g., tracks) within the network, and shared among nodes that are grouped dynamically in response to the outside activity. In addition, processing within the sensor network requires data about the network, and data to aid in processing, such as

- Code – the processes executing on a node must reside on the node during execution
- Node location and time – a node will typically have GPS capability for determining its location and setting clocks to synchronize with the rest of the network. Some processes may want to know what nodes are located within a particular radius of the node on which they are processing.
- Radio and network configuration – each radio has information about its local connections and performance parameters (e.g., radio frequency, packet size); network routing capabilities may maintain information about neighbors and paths
- Sensor information – sensor type and settings (e.g., speed, gain)
- Processing parameters and metadata – e.g., vehicle classification can involve matching the input signals against stored acoustic signatures for different vehicles at different speeds

Data storage within the sensor network involves decisions to determine what data needs to be stored, where it should be stored (i.e., on what node(s)), and for how long it should be stored. Such decisions must be based on resource limitations, as well as processing requirements. For example, each node may not have space to store all required acoustic signatures, so classification could involve moving additional signatures to a node when needed for processing, moving processing to a group of nodes that contains, in the aggregate, all of the signatures in the network, or some hybrid of these that perhaps caches a subset of signatures close to nodes where tracks are projected to arrive. Identifying the general characteristics of this type of optimization, and how a data management system would use these characteristics to manage data storage, is a promising research problem. A particularly interesting part of this problem is that the data management processing will need to be distributed; i.e., optimization of data storage is determined by collaboration among local data management processes.

Storage decisions are also (obviously) required for data that is produced within the network. Sensor input is an example of streamed data; in the SenseIT example, it is processed as it arrives at a node (digital signal processing and local signal processing), stored for a short period in a buffer (for use in classification if needed), and aggregated for longer-term storage (via event detection and, later, track formation). In general, however, limited storage resources dictate that all data generated within the sensor network must be transient since the network may operate (and produce data) for very long periods of time. General techniques for specifying data lifetimes, aggregating

data, and for moving data within, and in and out of, the network could be part of a query language and processing system.

Languages and processing techniques are required for queries within the network and for queries originating from outside the network. Queries originating from outside the network (we'll call them to-network queries) could be expected to inquire about current or recent activities monitored by the network (e.g., have any vehicles been detected at the east perimeter in the past hour?) – this is a view of the network as a large database. However, to-network queries will more typically be long-running specifications of information to be extracted from the network when and if it becomes available (e.g., notify me of intruders posing a threat to Camp X). One way to conserve network resources is to limit network operations to those necessary to operate the network and respond to such queries. In this view, queries essentially define tasks that need to be performed within the network to collect the appropriate data, and query processing must decide, collaboratively within the network, what tasks need to be performed by which nodes. For example, monitoring for intruder threats might involve starting sensor and detection processing at a subset of nodes whose sensors cover an area including entry points to Camp X, and establishing long-running queries within the network that move any detection information to nodes closer to the camp.

Queries within the network could be operations to perform externally introduced queries, but in-network query languages and processing must also support the exchange of information between (collaborating) processes operating in the network. Often the data requested in such queries will be identified by the location of the node to which the data refers. For example, a collaborative signal processing application will want notification of events that are detected on the node on which it is running and then may query nearby nodes to see if they have detected any events within a short time frame of the local detection. [*nb*. One would expect that event detection information would be stored locally on a detecting node, but the data management system should make the decisions about where to store data and (as usual) provide location transparency to the processes on the nodes.]

Moving data between nodes to respond to queries generally involves "pulling" data around/from the network for use in processing. However, as we've seen, there is also a need within the network for a push-based mode of data movement; tasking originating from to-network queries is one example. As another example, consider an approach to collaborative tracking in which a node that identifies a track notifies nodes in the expected path of the vehicle to watch for it. This notification involves pushing data about the track to other nodes, and also involves initiation of tasks at the receiving node.

Both in-network and to-network queries require declarative languages, procedural operations, and optimization techniques. Research is required to address the question of how general such languages can be, and whether a single language can satisfy both in- and to-network requirements. In addition, the relationships between query processing, data storage, and data movement must be examined in the presence of resource limitations.

Other data management issues that will only be mentioned (due to space limitations) include:

- Consistency – will not be needed across the entire network, may be needed across some subsets of nodes
- Replication – forced data replication for the usual efficiency and recovery reasons; inherent replication due to lack of data uniqueness (e.g., similar signals received by a vehicle passing multiple nodes)
- Backup and recovery – what are the characteristics of data that will need to survive a node failure?
- Data communication –data may not be delivered in a timely fashion, or at all, in a wireless network

Summary

A defining characteristic of the envisioned distributed sensor networks is that sensor data is processed within the network. This includes processing at single nodes and, more importantly, collaborative processing between nodes. Collaborative sensor applications include "user applications" that monitor, report on, and possibly modify the environment within which the sensor network is housed, and network applications that monitor, report on, and manipulate the sensor network to ensure that it executes the required applications. A distributed, collaborative network data management system will be essential to the success of these systems.

References

1. DARPA SOL BAA 99-16. Sensor Information Technology. For information refer to www.darpa.mil/ito/research/sensit/index.html.