Information assurance in wireless sensor networks

Stephan Olariu
Professor
Sensor Network Research Group
Old Dominion University
Talk overview

- Wireless sensor networks
- System assumptions
- Virtual infrastructure
- Information assurance in sensor networks
- Concluding remarks
Sensors
and
Wireless Sensor Networks
How it all started ...

- SmartDust program (sponsored by DARPA) defined sensor networks as:

  A sensor network is a deployment of massive numbers of small, inexpensive, self-powered devices that can sense, compute, and communicate with other devices for the purpose of gathering local information to make global decisions about a physical environment.
The National Research Council expanded the definition:

Sensor networks are massive numbers of small, inexpensive, self-powered devices pervasive throughout electrical and mechanical systems and ubiquitous throughout the environment that monitor (i.e., sense) and control (i.e., effect) most aspects of our physical world.
Sensor characteristics

- Sensors pack:
  - micro-sensor technology
  - low power signal processing
  - low power computation
  - low power short-range communications capabilities
  - modest non-renewable energy budget

- No fabrication-time identity!
Typical sensor diagram

- **Transceiver**
  - 1Kbps-1Mbps
  - 3m-300m
  - Lossy Transmission

- **Memory**
  - 128Kb-1Mb
  - Limited Storage

- **Sensor**
  - Requires Supervision
  - Multiple sensors

- **Embedded Processor**
  - 8 bit, 10 MHz
  - Slow Computation

- **Battery**
  - Limited Lifetime

- **Limitations**
  - Requires Supervision
  - Multiple sensors
  - Limited Storage
  - Slow Computation
Types of sensors

Some examples of existing sensors

- Pressure
- Temperature
- Light
- Biological
- Chemical
- Strain, fatigue
- Tilt
- Seismic
Wireless sensor network (WSN)

- **Massive number** of sensors densely deployed in the area of interest
- **Random deployment**: individual sensor positions cannot be engineered
- **Main goal**: global info from local data
- **Distributed system** with no central control
- **Only as good as the information it produces**
  - information quality
  - information assurance
Our view of a WSN system

- **deployment area**
- **user**
  - Returned results
- **Internet/satellite**
- **Low-level tasks/queries**
- **high-level interests**
  - (tasks/queries)
- **Sink** (mobile/airborne)
  - (connection to outside world)
- **sensors**
  - (in-network data repositories)
- **local sink node**

Tasks/queries

Returned results
Two application classes

- **Monitoring of static environments**
  - environmental monitoring
  - habitat monitoring
  - infrastructure surveillance

- **Monitoring of moving objects**
  - tracking animals in wildlife preserves
  - movement tracking of enemy vehicles
  - cross-border infiltration
The Virtual Infrastructure
How do we conquer scale?

Golden Rule: Divide and Conquer!

- Graft a virtual infrastructure on top of physical network
- How is this done?
  - special-purpose: protocol driven
  - general purpose: designed without regard to protocol
- General-purpose infrastructure should be leveraged by many protocols!
Components of the virtual infrastructure

- Dynamic coordinate system
  - location-based identifiers
  - coarse-grain location awareness
- Clustering scheme
  - cheap scalability
- Middleware
  - Work model
    - hierarchical specification of work and QoS
  - Task-based management model
    - low-level implementation of work model
The dynamic coordinate system
The cluster structure

- Cluster: locus of all sensors having the same coordinates
- Clustering falls out for free once coordinate system available
- Accommodates sensors with no IDs
- Clusters can be further subdivided - color graphs
What are color graphs?

- Simple way to enrich hierarchy
- Clusters are further subdivided into $p$ color sets
- What results are $p$ (global) color graphs
What’s so nice about color graphs?

- Very robust: each color graph is connected with high probability
- Thus, can serve for routing!
- They are (rich) cousins of circular arc graphs: vast body of knowledge to tap into for protocol design!
- Graceful degradation as energy budget depleted
Middleware for WSN?

- Appropriate middleware must provide standardized and portable system abstractions
- Standardize interface to WSN
- Requirements for middleware for WSN
  - negotiate QoS parameters on behalf on WSN
  - support and coordinate concurrent applications
  - translate high-level complex goals into low-level tasks
  - coordination among sensors
  - handle heterogeneity of sensors
The work model

Application layer

Interest

Negotiated QoS

Interest Result set, status
(error conditions, etc.)

Micro-task
Results, status

Communication

Sensor Network Layer

Cluster level

Sensor 1
Sensor 2
Sensor n

Middleware

Sink

Capability
(P-tasks/QoS)

Micro-task
Results, status

Communication

CPL

CPL

CPL

ODU Sensor Network Research Group
A P-task is a tuple $T(A,c,S,D,\pi,q)$ where:

- $A$ – action to be performed
- $c$ – color set to be used
- $S$ – source cluster
- $D$ – destination cluster
- $\pi$ – routing path from $S$ to $D$
- $q$ – desired QoS level
In-network storage
(WSN as databases)
Interacting with WSN

- Middleware pushes queries into WSN
- Query types:
  - one-shot: run once on the current data set; provides snapshot view of data/network
  - persistent: issued once and then logically run recurrently on the database; useful for analysis of data collected over time (especially for in-network storage)
- Responding to a query:
  - push/pull -- application-specific
  - data aggregation capability desirable
- Strategy: in-network storage desirable!
In-network storage and processing reduces energy usage
Trades off communication with local computation
Makes sense: communication consumes more energy than computation
Extends lifetime of WSN
Persistent queries (PQ)

- PQ=(Q, trigger, termination)
- Execution of PQ
  - Executed when the query is issued
  - Subsequently executed when trigger condition holds
  - Stops execution when termination condition satisfied
Trigger conditions

- **Time-based**
  - immediate
  - at a specific time point
  - at regular time intervals

- **Content-based**
  - a simple condition
  - an aggregate condition (based on the combined value of data in a locale)
  - a relationship between previous and current data values
Challenges in PQ

Example: Intrusion detection/target tracking

- Adaptivity to dynamically changing environments
- Scalability
- Graceful degradation for extreme conditions
  - Fluctuations such as increased workloads, bursty data
  - How can the system keep up?
  - Maybe drop some data or work with filtered data
Information assurance in WSN
What is information assurance?

“Information operations that protect and defend information and information systems ensuring their availability, integrity, authentication, confidentiality, and non repudiation. This includes providing for restoration of information systems by incorporating protection, detection, and reaction capabilities"
Key components

- **Network survivability**: ability of the network to function in the wake of failures by minimizing their impact.

- **Information availability (information survivability)**: need for a user to have uninterrupted and secure access to information on the network.

- **Network security**: attempts to provide basic security services.

- **Information security**: an ongoing process that utilizes software and hardware to help secure information flow.
Thus...

- Information assurance is more inclusive than information security
- Assurance involves not only protection and detection but also reaction (mainly survivability and dependability of the system that has been subject to successful attack)
- It also includes proactive (offensive) information operations, termed information warfare, against attackers
Extending WSN longevity

- Enforce *(quasi-*)optimal number of sensors per task
- Power control to maintain network connectivity in spite of sensor failure/energy depletion
- Topology control to enhance effective functional lifetime of the network
Problems with sleeping

- **Basic scheme**
  - sleep, wakeup periodically
  - check for “calls for participation”
  - if eligible to participate stay awake

- **Sleeping affects**
  - density of deployment
  - readiness of the WSN
  - response time

- Adjust sleep time dynamically
WSN health monitoring

- Query resource availability
- Energy map: spatial and temporal energy gradient of the WSN
- Usage pattern: identify
  - periods of activity for sensors
  - hot spots
- Selectively place additional sensors at hot spots to improve performance (not always an option)
- Self-healing a must!
Major insecurities in WSN

- Problems arising from lack of individual IDs
  - authentication is hard
  - trust relationships hard to establish
  - non-repudiation is hard to enforce
  - node impersonation is easy

- Eavesdropping: may give an adversary access to secret information violating confidentiality

- Sensors run the risk of being compromised
  - by infiltration
  - by tampering
Security goals

- **Availability**: ensures the survivability of network services despite denial-of-service attacks
- **Confidentiality**: ensures that certain information is not disclosed to unauthorized entities
- **Integrity**: guarantees that a message being transferred is never corrupted
- **Authentication**: enables a node to ensure the identity of the peer node with which it communicates
- **Non-repudiation**: ensures that the origin of a message cannot deny having sent the message
- **Anonymity**: hide the identity of sources, destinations and routes
A succinct list of attacks

- **Eavesdropping**: an attacker that monitors traffic can read the data transmitted and gather information by examining the source of a packet, its destination, size, number, and time of transmission.

- **Traffic analysis**: allows an attacker to determine that there is activity in the network, the location of base stations, and the type of protocol being used in the transmission.

- **Man-in-the-middle**: attack establishes a rogue intermediary pretending to be a valid sensor.

- **Tampering**: involves compromising data stored inside sensor usually by node capturing.

- **DoS attacks**: can be grouped into three categories:
  - disabling of service (e.g., sinkhole, HELLO flood attack),
  - exhaustion, and
  - service degradation (e.g., selective forwarding attack).

*Can we guard against them?*
Philosophy of our solution

“An ounce of prevention is worth a pound of cure”
What do we do?

- Physical-layer encoding: virtually stamps out infiltration by the adversary
- Also, leverage the virtual infrastructure!
- Problems discussed
  - tamper resistance
  - authentication
  - traffic anonymity
Prior to deployment sensors are injected with the following genetic material:

- a public-domain pseudo-random number generator
- an initial time -- at this point all the sensors are synchronous to the sink

Each sensor can generate pointers into:

- a random sequence $t_1, t_2, \ldots, t_i, \ldots$, of time epochs
- a random sequence $n_1, n_2, \ldots, n_i, \ldots$, of frequency channels
- for every $n_i$ a random hopping sequence $f_{i1}, f_{i2}, \ldots, f_{ip}, \ldots$, 
Illustrating time epochs, etc
Synchronization - generalities

- Synchronization does not scale!
- Thus, synchronization must be
  - short-lived
  - task-based
- Just prior to deployment, the sensors are synchronized
- Due to clock drift re-synchronization is necessary
- Sensors synchronize by following the master clock running at the sink
- Idea: determine the epoch and the position of the sink in the hopping sequence corresponding to the epoch
Synchronization - the details

- The sink dwells $t$ micro-seconds on each frequency in hopping sequence.
- Assume that when a sensor wakes up during its local time epoch $t_i$, the master clock is in one of the time epochs $t_{i-1}$, $t_i$, or $t_{i+1}$.
- Each sensor knows the last frequencies $\lambda_{i-1}$, $\lambda_i$, and $\lambda_{i+1}$ on which the sink will dwell in the time epochs $t_{i-1}$, $t_i$, and $t_{i+1}$.
- **The strategy:** tune in, cyclically, to $\lambda_{i-1}$, $\lambda_i$, and $\lambda_{i+1}$ spending time $t/3$ units on each of them.
Synchronization - the details (cont'd)

- Assume the sensor meets the sink on frequency $f_i$ in some unknown slot $s$ of $t_{i-1}$, $t_i$, or $t_{i+1}$.
- To verify the synchronization, the sensor attempts to meet the sink in slots $s+1$, $s+2$ and $s+3$ according to its own frequency hopping for epoch $t_{i+1}$.
- If a match is found, the sensor declares itself synchronized.
- Otherwise, it will return to scanning frequencies.
Making sensors tamper-resistant

- Philosophy: no additional hardware!
- Tampering threat model for sensors
  - forcing open in-situ
  - removal from the deployment area
- Play it safe: if in doubt blank out memory
Using neighborhood signatures

- Immediately after deployment each sensor transmits on a specified sets of frequencies, using a special frequency hopping sequence.
- Each sensor collects an array of signal strengths from the sensors in its locale.
- NSA - the Neighborhood Signature Array.
- Removal from deployment area → changes in the NSA!
NSA-based authentication

- Idea: neighbors exchange NSA information, creating a matrix of signatures
- A sensor that wishes to communicate with a neighbor identifies itself with its own NSA
- Upon receiving the NSA the sensor checks its validity
- Additional twist: store several instances of the matrix of NSAs
- Authentication dialogue: “what is your second to the last NSA?”
Handling DoS attacks

- Our physical-layer encoding
- + Tamper resistance
- + Infrastructure anonymity
- Make DoS attacks next-to-impossible
Concluding remarks

- Wireless sensor network research extremely hot!
- Lots of attention from funding agencies!
- WSN - far more vulnerable than wireless networks
- Securing sensor networks - a subject of active work
- Major challenge: comprehensive information assurance in hybrid wired+wireless networks