Sensor and Actuator Networks

Lecturer: Dr Niki Trigoni
Niki.Trigoni@cs.ox.ac.uk
Objectives of this course

• To identify applications and challenges of emerging networked environments

• To become familiar with
  – communication
  – data management and
  – positioning protocols

• To experiment with different sensor network platforms
Textbook

A. Mobile Ad hoc NETworks (MANETs)

 evolution, characteristics, applications and challenges
Evolution

• 500 B.C.:
  – Darius I, the king of Persia, used a line of shouting men to send messages from his capital to the remote provinces of his empire

• 1960s:
  – ARPANet: Packet switching technology; dynamic bandwidth sharing by multiple users

• 1970:
  – ALOHANet: Single-hop wireless packet network linking together the universities of the Hawaiian islands.

• 1972:
  – DARPA PRNet (Packet Radio Network) for military applications: protocols for sharing the radio channel and support for mobility

• 1983:
  – DARPA SURAN (Survivable Radio Networks) with small, low-cost, low-power devices and improved scalability and survivability
Characteristics

• A mobile wireless ad hoc network (MANET) is a network of mobile nodes that
  – are connected via wireless links and
  – exchange packets along multi-hop paths
  – without the support of a fixed communication infrastructure

• There is no centralized control. Each node can act as router and forward data for other nodes.

• Nodes are free to move randomly and organize themselves arbitrarily.

• The network’s topology may change rapidly and unpredictably.
Cellular vs. ad hoc wireless networks

Infrastructure-dependent

Basestation

Basestation

Basestation

Switching Center + Gateway

Infrastructure-less


Applications

• Smart cities
  – Smart transport
  – Smart buildings

• Environmental sensor networks
  – Forests, rivers, lakes, ocean, volcanoes

• Quality control and efficiency
  – Industrial processes
  – Energy sector

• Safety and security
  – Emergency
  – Military
Issues and challenges (I)

• **Physical layer:** how to transmit bits on the wireless medium
  – Use of electromagnetic spectrum
  – Radio propagation mechanisms and models
  – Converting data to electromagnetic waves for transmission
Issues and challenges (II)

- **Medium access control**: how to share the wireless medium among competing nodes
  - Distributed operation (no centralized coordination)
  - Time synchronization among nodes
  - High channel utilization by minimizing packet collisions and packet control overhead
  - Low delay in packet transmissions (particularly important in time-sensitive applications)
  - Fairness (equal / weighted share of bandwidth to competing nodes)
  - Bandwidth sharing in the presence of node mobility
  - Capability for power control and adaptive rate control
  - Use of directional antennas
Issues and challenges (III)

- Routing: how to find a feasible path to a destination based on criteria such as hop length, power consumption, path reliability, etc.
  - Distributed operation (no centralized control)
  - Path breaks and stale routing information caused by node mobility
  - Links are error-prone and bandwidth-constrained
  - Nodes are energy and memory constrained
  - Load balancing to avoid channel contention
  - Loop avoidance
  - Quick route acquisition and reconfiguration
  - Minimum control overhead
  - Scalability
  - ...
Issues and challenges (IV)

- Other issues
  - Sensing
  - Query processing
  - Storage management
  - Positioning
  - Controlled mobility
Summary

- Back in the 1970s, researchers started laying the foundations of ad hoc wireless networks with projects like ALOHANet, PRNet, etc.
- Mobile ad hoc networks consist of mobile nodes communicating wirelessly in a multi-hop manner.
- They have many interesting applications, ranging from military to sensor networks.
- They have many research challenges in terms of:
  - communication
  - positioning
  - data management
Plan

Monday
  1. Fundamentals of wireless communication
  2. Medium Access Control (MAC)

Tuesday
  3. Routing protocols

Wednesday
  4. Data management protocols
  5. Radio based positioning

Thursday
  6. Magnetic based positioning

Friday
  7. Mobile and delay-tolerant sensor networks
Related reading

• Chapter 5 of textbook:
Assessment

• Based on four exercises to be completed during lab sessions:
  – Communication (Tue afternoon)
  – Radio-based positioning (Wed afternoon)
  – Magnetic positioning (Thursday afternoon)
  – Mobile sensing (Friday afternoon)
Wireless networks: fundamentals of wireless communication

Electromagnetic spectrum, radio propagation mechanisms, path loss, fading, interference, transmission rate constraints, modulation techniques.
Electromagnetic spectrum

- Wireless communication: broadcast and reception of electromagnetic waves
- Wave frequency ($f$): number of cycles (oscillations) per second
- Wavelength ($\lambda$): distance between two consecutive minima or maxima in the wave
- Speed of propagation ($c$): varies from medium to medium (in a vacuum it is equal to the speed of light $3 \times 10^8$ m/s)

$$c = \lambda \times f$$
Electromagnetic spectrum

Long wavelength
Low frequency

Short wavelength
High frequency

Radio waves
Microwaves
Infra-red
Visible Light
Ultra-violet
X-rays
Gamma rays
Electromagnetic spectrum

**Frequency bands**

- **A band** is a small section of the spectrum of radio communication frequencies, typically used for the same purpose.
- Bands are divided at wavelengths of $10^n$ metres, or frequencies of $3\times10^n$ hertz.

<table>
<thead>
<tr>
<th>Band Number</th>
<th>Symbols</th>
<th>Frequency Range</th>
<th>Wavelength Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>VLF</td>
<td>3 to 30 kHz</td>
<td>10 to 100 km</td>
</tr>
<tr>
<td>5</td>
<td>LF</td>
<td>30 to 300 kHz</td>
<td>1 to 10 km</td>
</tr>
<tr>
<td>6</td>
<td>MF</td>
<td>300 to 3000 kHz</td>
<td>100 to 1000 m</td>
</tr>
<tr>
<td>7</td>
<td>HF</td>
<td>3 to 30 MHz</td>
<td>10 to 100 m</td>
</tr>
<tr>
<td>8</td>
<td>VHF</td>
<td>30 to 300 MHz</td>
<td>1 to 10 m</td>
</tr>
<tr>
<td>9</td>
<td>UHF</td>
<td>300 to 3000 MHz</td>
<td>10 to 100 cm</td>
</tr>
<tr>
<td>10</td>
<td>SHF</td>
<td>3 to 30 GHz</td>
<td>1 to 10 cm</td>
</tr>
<tr>
<td>11</td>
<td>EHF</td>
<td>30 to 300 GHz</td>
<td>1 to 10 mm</td>
</tr>
<tr>
<td>12</td>
<td>THF</td>
<td>300 to 3000 GHz</td>
<td>0.1 to 1 mm</td>
</tr>
<tr>
<td>Band Name</td>
<td>Common Use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Very Low Frequency (VLF)</td>
<td>(Sub-)Marine communications, wireless, heart rate monitors, geophysics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Frequency (LF)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium Frequency (MF)</td>
<td>AM (medium-wave) broadcasts, amateur radio, avalanche beacons..</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High Frequency (HF)</td>
<td>Long-distance aircraft/ship communications…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Very High Frequency (VHF)</td>
<td>FM, television broadcasts, aircraft comms…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ultra High Frequency (UHF)</td>
<td>Television broadcasts, microwave links, mobile phones, wireless LAN, Bluetooth, Zigbee, GPS, …</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Super High Frequency (SHF)</td>
<td>Satellite communications, microwave links, radars, satellite TV, radio astronomy…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extremely High Frequency (EHF)</td>
<td>Radio astronomy, microwave radio relay, microwave remote sensing…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terahertz or Tremendously High Frequency (THF)</td>
<td>Terahertz imaging, condensed matter physics …</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Radio propagation basics

- Radio transmission is omnidirectional (waves spread out from the transmitting antenna in all directions)
- Low frequencies: radio waves pass through obstacles easily
- High frequencies: radio waves get reflected by obstacles and are more prone to absorption by rain drops
Radio propagation mechanisms

- **Reflection**: When the propagating radio wave hits an object which is very large compared to its wavelength, the wave gets reflected by that object.

- **Diffraction**: When a wave hits an impenetrable object of size comparable to its wavelength, the wave bends at the edges of the object, thereby propagating in different directions.

- **Scattering**: When a wave travels through a medium, which contains many objects with dimensions small when compared to its wavelength, the wave gets scattered into several weaker outgoing signals.
Path loss

• **Path Loss (Attenuation):** *Reduction in power density of an electromagnetic wave as it propagates from a transmitter to a receiver.*

• **It includes losses caused by:**
  – Natural expansion of the radio wave front in free space
  – Absorption (penetration) losses (when the radio passes through media not transparent to electromagnetic waves)
  – Diffraction/scattering losses etc.
Path loss

• The path loss depends on:
  – Radio frequency
  – Distance between transmitter and receiver
  – Antenna characteristics (height, location, gains)
  – Nature of the terrain (urban/rural, vegetation/clear)
  – Propagation medium (dry/moist air)
Radio propagation models

- **Free space model**: It assumes that there is only one clear line-of-sight path between the transmitter and receiver.

\[ P_r = P_t G_t G_r \left( \frac{\lambda}{4\pi d} \right)^2 \]

- received power
- transmitted power
- transmitter/receiver antenna gains
- distance between transmitter and receiver
Radio propagation models (contd.)

- **More general model**, where the propagation coefficient (a.k.a. **path loss exponent**) $\gamma$ varies between 2 (free-space propagation) and 5 (strong attenuation).

\[
P_r = P_t G_t G_r \left( \frac{\lambda}{4\pi} \right)^2 \frac{1}{d^\gamma}
\]
Radio propagation models (contd.)

- **Two-way (a.k.a. two-path) ground reflection model**: It assumes that the signal reaches the receiver through two paths, a line-of-sight path, and a non-line-of-sight reflected path.

\[
P_r = P_t G_t G_r \left( \frac{h_t h_r}{d^2} \right)^2
\]

- height of transmitter
- height of receiver
Radio propagation models (contd.)

• **Shadowing model:** It uses a close-in distance $d_0$ as a reference. The received power at certain distance is a random variable due to multipath propagation (fading) effects.

\[ P_r(d) = P_{r0} \times \left( \frac{d_0}{d} \right)^\gamma \]

predicted mean receive power at distance $d$

\[
\begin{bmatrix}
\frac{P_r(d)}{P_{r0}}
\end{bmatrix}_{dB} = -10 \times \gamma \times \log \left( \frac{d}{d_0} \right) + X_{dB}
\]
Fading

• Fading: fluctuations in signal strength at the receiver.
  – **Fast (Small-Scale) Fading**
    • Due to interference between multiple versions of the same transmitted signal arriving at the receiver at slightly different times
    • Recall that multi-path propagation may be caused by reflection, diffraction and scattering
  – **Slow (Large-Scale) Fading**
    • Obstacles between transmitter and receiver partially absorb the transmission
    • Example: Transmitter outside a building, receiver inside a building, walls blocking transmission
Interference

• **Adjacent Channel Interference**
  – Signals in nearby frequencies have components outside their allocated ranges, and these components may interfere with on-going transmissions in the adjacent frequencies

• **Co-Channel Interference**
  – Other nearby systems using the same transmission frequency

• **Inter-Symbol Interference**
  – Distortion in the received signal is caused by the temporal spreading and the consequent overlapping of individual pulses in the signal
Transmission rate constraints

• **Nyquist’s Theorem**: It gives the *maximum data rate (in bits per sec) possible on a noiseless channel*, as function of the bandwidth $B$ of the channel (in Hz) and the number of discrete signal levels (voltage values) used:

$$C = 2 \times B \times \log_2 L$$
Transmission rate constraints

• **Shannon’s Theorem:** It gives the *maximum data rate (in bits per sec) possible on a noisy channel*, as a function of the bandwidth $B$ of the channel (in Hz) and the ratio of signal power ($S$) to noise power ($N$):

$$C = B \times \log_2 \left(1 + \frac{S}{N}\right)$$
Modulation techniques

• How is data transmitted on the wireless medium?
• The modulation process alters certain properties of a radio wave, called a **carrier wave**.
• The carrier wave’s frequency is the same as the frequency of the wireless channel being used for the transmission.
  – **Analog modulation** techniques
    • For transmitting analog data
  – **Digital modulation** techniques
    • For transmitting digital data
Digital modulation techniques

- **Amplitude Shift Keying (ASK):** It represents digital data by varying the amplitude of the carrier wave.

- **Frequency Shift Keying (FSK):** It represents digital data by varying the frequency of the carrier wave.

- **Phase Shift Keying (PSK):** It represents digital data by varying the phase of the carrier wave.
Digital Modulation

BASEBAND

CARRIER

AMPLITUDE SHIFT KEYING

FREQUENCY SHIFT KEYING

PHASE SHIFT KEYING
Summary

- Wireless communication is based on broadcasting and receiving electromagnetic waves.
- ITU has defined various frequency bands in the electromagnetic spectrum used for different purposes.
- Radio transmission is omnidirectional, and suffers from attenuation, fading and interference.
- Nyquist’s and Shannon’s theorems define two important constraints that determine the maximum rate of data transmission on a channel.
- Transmission of data over the wireless medium is done by altering the properties (amplitude, frequency, phase) of a carrier wave.
Related reading

• Standard textbook:
  – Chapter 1
Medium Access Control

MAC protocols:
  design goals, challenges,
  contention-based
  and contention-free protocols
Why do we need MAC protocols?

- Wireless medium is shared
- Many nodes may need to access the wireless medium to send or receive messages
- Concurrent message transmissions may interfere with each other => collisions => message drops
- A MAC protocol is needed to allow the efficient sharing of the wireless medium by multiple nodes
Design goals

- To ensure reliable communication across wireless links (not end-to-end reliability, only 1-hop reliability)
- To maximize the use of available bandwidth (keep control overhead as low as possible)
- To ensure fair bandwidth allocation to contending nodes
- To minimize delay of sending/receiving messages
- To minimize energy-consumption of sending/receiving messages
Challenges

- Error-prone channel
- Limited bandwidth
- Limited communication range
- Limited energy (for remote battery-powered nodes)
- Node mobility
- Lack of central coordination
- Lack of tight time synchronisation
Protocol classification

• Contention-based MAC protocols
  – Contention arises when two or more nodes attempt to transmit at the same time over a shared channel.
  – A contention-based protocol assumes that packet collisions may occur, and tries to detect, avoid or deal with them.

• Contention-free MAC protocols
  – A contention-free protocol tries to divide the wireless channel into logical channels that do not interfere with each other (e.g. TDMA, FDMA, CDMA).
  – Nodes transmit packets using different logical channels, and as a result, there is no contention in the network.
  – Contention-free schemes are more applicable to fixed networks or networks with centralized control.

• In this course, we focus on contention-based MAC protocols
Some contention-based protocols

- ALOHA, slotted-ALOHA
- CSMA
- MACA
- MACAW
- IEEE 802.11 (DCF)
- ...
ALOHA, slotted-ALOHA

• Pure ALOHA
  – Nodes access the channel when they have data to transmit.
  – If a transmission is unsuccessful (e.g. no ACK is sent from the destination node), the source node retransmits after a random amount of time.

• Slotted-ALOHA
  – Time is divided into equal size slots
  – A node transmission always starts at the beginning of a slot
  – If a transmission is unsuccessful, the source node retransmits at a future slot with a certain probability.
ALOHA

Node 1
Node 2
Node 3
Node 4
Node 5
Node 6
Node 7

successful packet
unsuccessful packet

slotted-ALOHA

time
time
CSMA - physical sensing

• CSMA (Carrier Sense Multiple Access)
  – The sender first senses the wireless channel
  – If the channel is idle, the sender starts transmitting
  – If the channel is in use the sender refrains itself from transmission until the channel is idle.

• CSMA/CA (CSMA with Collision Avoidance)
  – If the channel is busy before transmission then the transmission is deferred for a "random" interval.
  – This reduces the probability of collisions on the channel.
CSMA/CA implementation

- Before transmitting a frame, a node senses the channel.
- If the channel is idle for longer than DIFS (Distributed Interframe Space), the node continues with its transmission.
- Otherwise, if the medium is busy, the transmission is deferred until the end of the ongoing transmission. Then a backoff procedure is started.

Node 1 finds the channel idle and transmits

Node 2 finds the channel busy and defers transmission

Node 3 finds the channel busy and defers transmission
CSMA/CA implementation - backoff

- A random interval - backoff time - is selected and used to initialize the backoff timer.
- The backoff timer is:
  - decreased as long as the channel is idle,
  - paused when a transmission is detected, and
  - reactivated when the channel is sensed as idle again for more than DIFS
- When the backoff timer reaches 0, the node retries to send its frame.
CSMA/CA implementation - backoff

• Binary Exponential Backoff (BEB)
  
  – The backoff time is an integer number of slots uniformly chosen in the interval (0,\(CW-1\)), where \(CW\) is the Contention Window.

  – At the first transmission attempt, \(CW=CW_{\text{min}}\) and it is doubled at each retransmission up to \(CW_{\text{max}}\)

  – After a successful transmission, \(CW\) is reset to \(CW_{\text{min}}\)
CSMA/CA with ACK

- Acknowledgement scheme:
  - The destination node waits for SIFS (Short Interframe Space) after receiving a frame. Typically, SIFS is shorter than DIFS.
  - The destination node sends an ACK to the source node.

- If the source node does not get an ACK, it considers the transmission to be unsuccessful, waits for EIFS (Extended Interframe Space) and activates the backoff algorithm.
The exposed node problem

*CSMA may cause nodes to unnecessarily refrain from accessing the medium.*

B transmits to A.
C hears the transmission from B to A.
C *unnecessarily* refrains from sending a message to D even though no collision would occur.
The hidden node problem

CSMA does not avoid the hidden node problem.

A transmits to B. B receives the message. C does not hear the transmission.

A tries to transmit to B. C also tries to transmit to B. Both messages are dropped at B.
MACA – virtual sensing

• MACA (Multiple Access with Collision Avoidance) [Karn 1990]

• Nodes reserve the channel using control messages (virtual sensing):
  – The sender first expresses its wish to transmit by sending a Request-To-Send (RTS) message
  – The receiver allows this transmission by sending a Clear-To-Send (CTS) message
  – The sender then sends the Data message
A sends RTS to B.
B sends a CTS to A (C overhears it).
A sends Data to B.

Both RTS and CTS carry information about the duration of the Data transmission.
RTS-CTS handshake

• If control (RTS-CTS) messages collide with each other or with data packets, a backoff procedure is activated (backoff is binary exponential).

• RTS-CTS helps to avoid some cases of the hidden and exposed node problems, because:
  – All neighbors of the sender hear the RTS.
  – All neighbors of the receiver hear the CTS.

• However, it does not always avoid these problems!
MACAW

- MACAW [Bharghavan et al. 1994] extends MACA
  - RTS-CTS-**DS**-Data-**ACK**
MACAW

- MACAW extends MACA with
  - acknowledgements
  - an improved backoff mechanism
    => fair allocation of the medium to contending nodes
  - DS (Data Sending) message:
    - Say that a neighbor of the sender overhears an RTS but not a CTS (from the receiver)
    - In this case it can’t tell if RTS-CTS was successful or not
    - When it overhears the DS, it realizes that the RTS-CTS was successful, and it defers its own transmission
• Standard MAC and physical protocol for wireless LANs
• The MAC layer offers two types of services:
  – Distributed Coordination Function (DCF)
  – Point Coordination Function (PCF)

DCF combines
  – physical sensing (CSMA/CA) and
  – virtual sensing (RTS-CTS-Data-ACK)
IEEE 802.11 DCF - virtual sensing

- RTS and CTS include the busy channel duration
- Nodes that overhear RTS/CTS set timer NAV (Network Allocation Vector) to the busy channel duration
- A node starts backoff after NAV becomes zero
802.11 standards

• 802.11 (1997): First WLAN standard by IEEE
  – 2.4GHz (unregulated)
  – 2 Mbps – too slow for most applications
• 802.11b (1999)
  – 2.4 GHz, interference from microwave ovens, cordless phones, etc.
  – 11 Mbps
• 802.11g (2002-2003)
  – 2.4 GHz, backwards compatible with 802.11b
  – 54 Mbps
• 802.11n (2009)
  – MIMO: multiple wireless antennas in tandem to transmit and receive
  – +100 Mbps
• 802.11ac (2014)
  – 5 GHz band, up to 7Gbps
  – Multi user MIMO
Contestation-free multiple access

- **TDMA (Time Division Multiple Access)**
  - Time is divided into timeslots
  - Nodes transmit one after the other using their own timeslot
  - TDMA requires good time synchronization
- **FDMA (Frequency Division Multiple Access)**
  - The available bandwidth is divided into multiple frequency channels / bands.
  - A transmitter-receiver pair uses a dedicated frequency channel for communication
- **CDMA (Code Division Multiple Access)**
  - Every transmitter uses the entire spectrum (not a specific frequency)
  - The transmissions are differentiated through a unique code assigned to each node (that is independent of the data being transmitted)
IEEE 802.11 PCF

- IEEE 802.11 PCF (Point Coordination Function)
- One node, called Access Point (AP), coordinates the transmissions of its neighbors
- The AP polls neighbors one after the other, and allows them to transmit in a round robin manner
- PCF is not suitable for large multi-hop networks
Bluetooth

- Piconet: One node, called the master can communicate with up to 7 nodes called the slaves
- Bluetooth uses 79 channels (each 1 MHz wide) and changes channels up to 1600 times per second
- Each channel is divided into time slots of 625 µsecs
- The master switches from slave to slave in a round-robin fashion
  - Time-Division Duplex (TDD): master (downlink) and slave (uplink) transmissions occur in alternative slots
  - Slaves can talk back to the master immediately after they are polled by the master
Summary

Contention-based protocols
- Aloha, slotted-Aloha: no physical sensing
- CSMA: physical sensing
- CSMA/CA: physical sensing and backoff
- MACA and MACAW: virtual sensing using control packets (RTS/CTS)
- IEEE 802.11 DCF: physical and virtual sensing

Contention-free protocols
- IEEE 802.11 PCF
- Bluetooth
- ...
Related reading

• Standard textbook:
  – Section 2.3
  – Section 2.5
  – Sections 6.1-6.5.1

• Internet sources:
  – http://www.utdallas.edu/~mxw013200/MAC_ADHOC.html
  – http://attila.sdsu.edu/~kumar/MAC_Survey.pdf
Routing

design goals, challenges,
link state vs. distance-vector routing,
proactive vs. reactive routing
Why do we need routing protocols?

Nodes are not fully connected => Need for *multi-hop communication*

Say N7 wants to send a message to N3. Several options:
- N7 -> N1 -> N2 -> N3
- N7 -> N1 -> N5 -> N2 -> N3
- N7 -> N1 -> N5 -> N4 -> N3  etc.
Why do we need routing protocols?

Routing problem:
-> For each node find the best (shortest/fastest/most robust) path to each destination node.

Distributed version of the routing problem:
-> For each node find the next hop in the best path to each destination node.
Why do we need routing protocols?

- They help nodes identify the preferred neighbor (next hop) in the optimal path to a destination.

- They help nodes forward packets hop-by-hop from the source to the destination:
  - The data packet contains the destination node in its header.
  - When a node receives a data packet, it forwards it to the preferred neighbor in the optimal path to the destination.
Challenges I

- Mobile hosts can move
- Mobile hosts can be dynamically added or removed from the network
- Wireless links often have intermittent connectivity

=> The network connectivity changes dynamically
Challenges II

- Mobile nodes often have stringent energy and memory constraints
- Wireless links have limited bandwidth, and time-varying link error probability
- Some applications require real-time data delivery
- Some applications require robust data delivery
Design goals

• Fully distributed - no centralized control
• Adaptivity to topology changes
• Loop-free paths
• Fast route discovery (for real-time applications)
• Low control overhead (for bandwidth-/energy-constrained applications)
• Fault-tolerance (for security/emergency applications)
• ...
Classification of routing protocols

Routing protocols

- Link-state
- Distance-vector
  - Proactive
  - Reactive
  - Hybrid
In link-state protocols, each node:
• has a complete view of the network topology
• propagates the costs of its outgoing links to all other nodes
• uses Dijkstra’s algorithm to find the optimal path to other nodes

In distance-vector protocols, each node $i$:
• maintains for each destination $x$ a set of distances (costs) $d_{i-j-...-x}$ of paths from $i$ to $x$ through neighbor node $j$:
• selects to forward a data packet through neighbor $k$ such that $d_{i-k-...-x} = \min_j \{d_{i-j-...-x}\}$
• propagates to neighbors current estimates of optimal distances $d_{i-k-...-x}$ to each destination $x$
• uses the distributed Bellman Ford algorithm to update local estimates of optimal distances, based on those of its neighbors
Example of route information stored at node N1
Example of route information sent from node N1

The route info that “the neighbors of N1 are N2 and N3” is sent to all nodes

The route info that “the costs of paths from N1 to N2,N3,N4,N5,N6 are 1,1,2,2,2 respectively” is sent only to neighbor nodes N2 and N3
Dijkstra’s algorithm by example

- http://www.dgp.toronto.edu/people/JamesStewart/270/9798s/Laffra/DijkstraApplet.html
Distributed Bellman-Ford by example

Initially, nodes know distances to their neighbors only. They assume distances to other nodes are $\infty$.

<table>
<thead>
<tr>
<th>Info stored at node ...</th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
<th>N4</th>
<th>N5</th>
<th>N6</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>0</td>
<td>2-N2</td>
<td>7-N3</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>N2</td>
<td>2-N1</td>
<td>0</td>
<td>$\infty$</td>
<td>3-N4</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>N3</td>
<td>7-N1</td>
<td>$\infty$</td>
<td>0</td>
<td>1-N4</td>
<td>3-N5</td>
<td>2-N6</td>
</tr>
<tr>
<td>N4</td>
<td>$\infty$</td>
<td>3-N2</td>
<td>1</td>
<td>0</td>
<td>1-N5</td>
<td>$\infty$</td>
</tr>
<tr>
<td>N5</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>3-N3</td>
<td>1-N4</td>
<td>0</td>
<td>$\infty$</td>
</tr>
<tr>
<td>N6</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>2-N3</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>0</td>
</tr>
</tbody>
</table>

Let each node broadcast its route info periodically to its neighbors.
Distributed Bellman-Ford by example

For example, let N2 broadcast its route info first (highlighted in bold). The neighbors of N2 (N1 and N4) receive this route info.

Nodes N1 and N4 update their local route info based on the received route info.

- N1 discovers a shorter path to N4 of length 5 via next-hop N2
- N4 discovers a shorter path to N1 of length 5 via next-hop N2
Distributed Bellman-Ford by example

In the next step, let N4 broadcast its route info (highlighted in bold). The neighbors of N4 (N2, N3 and N5) hear this route info.

N2, N3 and N5 update their local info as underlined above.
Distributed Bellman-Ford by example

Nodes continue to broadcast their route info to their neighbors, and after several iterations, nodes identify optimal paths to all other nodes.

Distance to node ...

<table>
<thead>
<tr>
<th></th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
<th>N4</th>
<th>N5</th>
<th>N6</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Assume that link N1<--> N2 breaks.

N3 is not aware of the link failure. N3 broadcasts that it knows a route to N1 of length 5 (which is false).

Hearing N3, node N2 thinks that there exists a path from N2 to N1, via next-hop N3, of length 3+5=8.

When node N2 decides to send data to N1, the data travels on loop N2->N3->N4->N5->N6->N2->N3->N4->N5->N6-> ... etc.
Distance-vector protocols

• **Proactive protocols** identify optimal paths to destination nodes in advance, so that they are already there whenever needed.
  – e.g. DSDV

• **Reactive protocols** identify optimal paths to destination nodes on-demand (when needed).
  – e.g. AODV

• **Hybrid protocols** use the proactive approach to find paths to nodes within a nearby zone, and the reactive approach to find paths to nodes beyond this zone.
  – e.g.: ZRP

• DSDV and AODV use an enhanced version of the Distributed Bellman Ford algorithm, in which route information is annotated with an indication of its freshness to prevent / quickly resolve loops.
DSDV: Destination-Sequenced Distance-Vector protocol

- Each node maintains locally a routing table
- Each entry of the routing table includes routing information for a destination node:
  - the next hop in the optimal path to the destination
  - the cost of the optimal path to the destination
  - the freshness (sequence no) of the path to the destination
- The node advertises the local routing table to its neighbors
  - Periodically
  - When topology changes are detected
- On receiving routing information from a neighbor, a node uses it to update its own local routing table
DSDV: Destination-Sequenced Distance-Vector protocol

A few entries in N1’s routing table

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>Metric</th>
<th>Sequence Number</th>
<th>Install</th>
<th>Stable Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>N2</td>
<td>N2</td>
<td>1</td>
<td>S212_N2</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>N3</td>
<td>N2</td>
<td>2</td>
<td>S302_N3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N4</td>
<td>N5</td>
<td>2</td>
<td>S100_N4</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Sequence number is generated at the destination.
What if N1 receives new routing information (Dest=N6, Metric=2, SeqNo=S200_N6) from N5?

DSDV: Destination-Sequenced Distance-Vector protocol

N1 Routing Table

<table>
<thead>
<tr>
<th>Destin</th>
<th>Next Hop</th>
<th>Metric</th>
<th>Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>N6</td>
<td>N2</td>
<td>4</td>
<td>S200_N6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

N1 Routing Table (updated)

<table>
<thead>
<tr>
<th>Destin</th>
<th>Next Hop</th>
<th>Metric</th>
<th>Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>N6</td>
<td>N5</td>
<td>3</td>
<td>S200_N6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>3</td>
<td>S200_N6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

...
Any routing information that N1 receives regarding Dest=N6 that has sequence number smaller than 200 (S200_N6) is considered stale, and it is ignored by N1.
What if N1 receives new routing information (Dest=N6, Metric=2, SeqNo=S202_N6) from N5?

DSDV: Destination-Sequenced Distance-Vector protocol

### N1 Routing Table

<table>
<thead>
<tr>
<th>Destin</th>
<th>Next Hop</th>
<th>Metric</th>
<th>Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>N6</td>
<td>N5</td>
<td>3</td>
<td>S200_N6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### N1 Routing Table (updated)

<table>
<thead>
<tr>
<th>Destin</th>
<th>Next Hop</th>
<th>Metric</th>
<th>Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>N6</td>
<td>N5</td>
<td>3</td>
<td>S202_N6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

...23
DSDV: Destination-Sequenced Distance-Vector protocol

N1 Routing Table

<table>
<thead>
<tr>
<th>Destin</th>
<th>Next Hop</th>
<th>Metric</th>
<th>Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>N6</td>
<td>N5</td>
<td>3</td>
<td>S202_N6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

What if the link between N1 and N5 breaks?

N1 Routing Table (updated)
New routing entry is broadcast

<table>
<thead>
<tr>
<th>Destin</th>
<th>Next Hop</th>
<th>Metric</th>
<th>Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>N6</td>
<td>N5</td>
<td>∞</td>
<td>S203_N6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

... 24
DSDV: Destination-Sequenced Distance-Vector protocol

- Compare new routing information with the information available in the local routing table
- Prefer routes with more recent sequence numbers
- Discard routes with older sequence numbers
- Prefer routes with sequence number equal to an existing entry if it has a better metric value
- Newly recorded routes are scheduled for immediate broadcasting
- Updated routes only with a new sequence number are scheduled for advertisement at a later time
AODV: Ad Hoc On-Demand Distance-Vector protocol

• AODV
  – does not maintain routes from every node to every other node in the network.
  – discovers routes on-demand (reactively, not proactively)
  – provides unicast, multicast and broadcast communication ability
• We will consider only unicast route discovery.
AODV: Ad Hoc On-Demand Distance-Vector protocol

Unicast routing
A node wishes to send a packet to a destination node D. It first checks whether it has a valid route to D.

- If yes, it sends the packet to the next hop towards the destination.
- If not, it initiates a route discovery process.
AODV: Ad Hoc On-Demand Distance-Vector protocol

Unicast routing: Route Discovery Process

- The node creates a RREQ (RouteRequest) packet
- The node broadcasts the RREQ
- The node sets a timer to wait for a reply

<table>
<thead>
<tr>
<th>SrcID</th>
<th>BcastID</th>
<th>DestID</th>
<th>DestSeqNo</th>
<th>HopCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>c</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
AODV: Ad Hoc On-Demand Distance-Vector protocol

Unicast routing: Route Discovery Process

• When a node receives a RREQ, it ignores it if it has seen another routing packet with the same <SrcID, BcastID> pair.
• Otherwise, the node sets up a reverse routing entry in its routing table:
  – SrcID
  – BcastID
  – HopsToSource
  – PrevHopToSource
• Route entries that exceed their lifetime are deleted.
AODV: Ad Hoc On-Demand Distance-Vector protocol

- SrcID: a
- BcastID: 0
- HopsToSrc: 1
- PrevHopToSrc: a
AODV: Ad Hoc On-Demand Distance-Vector protocol

Unicast routing: Route Discovery Process

• A node responds to an RREQ if it has
  – an unexpired entry for the destination in its route table
  – with sequence no >= RREQ’s DestSeqNo
  By unicasting a RREP back to the source.

• If a node cannot respond to an RREQ, it increments the RREQ’s HopCount and then broadcasts the packet to its neighbors.
AODV: Ad Hoc On-Demand Distance-Vector protocol

<table>
<thead>
<tr>
<th>SrcID</th>
<th>BcastID</th>
<th>DestID</th>
<th>DestSeqNo</th>
<th>HopCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>c</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
AODV: Ad Hoc On-Demand Distance-Vector protocol

**Unicast routing: Route Replies (RREPs)**
- If the destination node receives a RREQ, it creates a RREP packet with the current destination sequence number.
- The RREP’s hop count to destination value is set to 0.
- The RREP’s lifetime value is initialized. Routes to destination are invalidated at the end of the lifetime.

- The RREP is sent to the neighbor that first sent the RREQ.

<table>
<thead>
<tr>
<th>DestID</th>
<th>DestSeqNo</th>
<th>HopCount</th>
<th>Lifetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>1</td>
<td>0</td>
<td>lifeValue</td>
</tr>
</tbody>
</table>
AODV: Ad Hoc On-Demand Distance-Vector protocol

Unicast routing: Route Replies (RREPs)

- If an intermediate node receives a RREP, it sets up a forward path entry to the destination in its routing table.
- It then sends the RREP back towards the source.

<table>
<thead>
<tr>
<th>DestID</th>
<th>DestSeqNo</th>
<th>HopCount</th>
<th>Lifetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>1</td>
<td>1</td>
<td>lifeValue</td>
</tr>
</tbody>
</table>
AODV: Ad Hoc On-Demand Distance-Vector protocol

- Once the source receives the first RREP packet, it updates its routing table and can now forward data to the destination.
Unicast routing: Route maintenance

• A discovered route between a source and destination is maintained as long as needed by the source node - we call these active paths.
• If the source node moves during an active session, it can reinitiate route discovery to establish a new path to the destination.
• When the destination or an intermediate node of an active path moves, a Route Error (RERR) message is initiated by the node closer to the path break and is forwarded back to the source node, which then reinitiates route discovery.
DSDV vs. AODV

- In DSDV, nodes continuously maintain routes to all destinations, even if they don’t use them frequently. In AODV, nodes identify and maintain routes on-demand - when they need to send packets to a destination.
- DSDV has a heavy control overhead during high mobility, due to continuous route updates caused by broken links. AODV deals with broken links more efficiently since it informs only those nodes that have been using them. In the frequent case that a link is idle before it breaks, no control messages are sent.
- DSDV routes data packets to a destination faster than AODV, since it does not need to discover routes on demand - routes are already established.
Summary

• The aim of routing protocols is to identify min-cost multi-hop paths to forward data from a source node to a destination node.
• Routing is a challenging problem in ad hoc networks due to node mobility, node/link failures, and resource constraints (bandwidth, energy, computation and storage are often limited resources).
• Routing protocols are divided into link-state and distance-vector protocols. Link-state protocols require each node to know the costs of links in the entire network. Distance-vector protocols require each node to know the distance (and next hop) of paths to destination nodes.
• Distance-vector protocols are divided into proactive (e.g. DSDV) and reactive (e.g. AODV). Proactive protocols discover and maintain optimal paths from sources to destinations in advance, independent of whether these paths are used, whereas reactive protocols discover paths from sources to destinations on demand - when needed.
Related Reading

• Standard textbook:
  – Sections 7.1-7.4.1
  – Section 7.5.2


Routing protocols for sensor networks
Sensor node

Node with

- computation,
- storage,
- communication and
- sensing capabilities.

Example: Tmote

- 8 MHz Processor
- 10k RAM, 48k Flash
- 250kbps 2.4GHz Radio
  - 50m range indoors / 125m range outdoors
- Integrated Humidity, Temperature, and Light sensors
- Programming and data collection via USB
- TinyOS support
Sensor network

- A collection of sensor nodes deployed in an area and connected through a multi-hop wireless network.
Simple deployment

Gateway

queries

sensor readings
Hierarchical deployment

Internet
Other hybrid deployments
Sensor network applications

Examples:

• habitat monitoring
• chemical and biological sensors
• fire, earthquake emergencies
• vehicle tracking, traffic control
• surveillance of city districts
• defense-related networks
• alerts to terrorist threats
• ...
Routing protocols for sensor networks

- Routing protocols must be localized, lightweight and scalable.
- They must avoid the overhead of storing routing tables or other information that is expensive to update such as link costs or routes to every destination.
- Routing decisions must be made not based on destination addresses (classical approach), but based on destination attributes like:
  - Location
  - Type of sensors
  - Certain range of values in a certain type of sensed data
- New mechanisms are needed to route packets using destination attributes.
Routing protocols for sensor networks

• In this course, we will describe three approaches:
  – Geographic routing
  – Attribute-based routing
  – Tree-based routing
Geographic routing

- The goal is to route packets to a node with a known geographic location (or to all nodes within a geographic region)
- All nodes know their geographic location
- Each node knows its one-hop neighbors
Geographic routing

• Greedy forwarding:
  – Let s be the source node and d the destination node
  – Let the packet currently be held by intermediate node x
  – Node x learns its neighbors’ positions from beacons
  – Node x sends its packet to neighbor node y that is geographically closest to the destination d
Geographic routing

- Greedy forwarding is not always possible
  - A packet may encounter a *void* or *hole* in the way to the destination, for example, none of x’s neighbors is closer to d than x
Geographic routing

- Face routing (or face traversal) is invoked to recover from local minima
  - It works correctly on a network graph that has no crossing links - a planar graph
  - A planar graph consists of faces, enclosed polygonal regions bounded by links
  - Face routing uses two primitives to traverse a planar graph:
    1. the right-hand rule and
    2. face changes
Geographic routing

- The right-hand rule:
  - y receives a packet from x and forwards it to its first neighbor counterclockwise about itself, z.
Geographic routing

- Face changes in face routing

- Line \( sd \) line cuts a series of faces in the planar graph.
- Face traversal successively traverses the faces cut by this line.
- The right hand rule is augmented with a rule that describes when to change to an adjacent face.
- A face change takes place just before traversing an edge that crosses the \( sd \) line.
Geographic routing

- Face routing works correctly on planar graphs - graphs where no two edges cross
- How to planarize graphs?
  - Relative Neighborhood Graph (RNG)
  - Gabriel Graph (GG)
Geographic routing

- The Relative Neighborhood Graph (RNG) is planar
  - For edge \((u,v)\) to be included in the RNG graph, the intersection area must contain no witness \(w\)
  - Note that when we begin with a connected graph, and remove edges not part of the RNG, we cannot disconnect the graph.
Geographic routing

• The Gabriel Graph (GG) is planar
  – For edge \((u,v)\) to be included in the GG graph, the shaded circle must contain no witness \(w\)

  
  ![Diagram of the Gabriel Graph](image)

  – Note that when we begin with a connected graph, and remove edges not part of the RNG, we cannot disconnect the graph.
  – RNG is a subset of the GG
Geographic routing

• GPSR: Greedy Perimeter Stateless Routing
  – Greedy Forwarding on the full network graph
  – Face Routing (Perimeter Forwarding) on the planarized network graph, where Greedy Forwarding is not possible

Routing protocols for sensor networks

- Geographic routing
- **Attribute-based routing**
- Tree-based routing
Attribute-based routing

- We want to get information from the sensor network about some type of event without knowing
  - the ids of the nodes nor
  - the location of nodes
  - that generate relevant sensor data.

- We can express our request as a series of attribute-value pairs, e.g.:
  - type = animal
  - instance = horse
  - interval = 30 min
Attribute-based routing

- Sinks: nodes requesting information
- Sources: nodes providing information
- Interests: records indicating a desire for certain type of information

- A typical interest record contains an ‘interval’ attribute field, which indicates the frequency with which the sink wishes to receive information about objects matching the other record attributes.
Attribute-based routing: Directed diffusion

- Directed diffusion is suitable for addressing attribute-value requests.
- It finds good paths between sources and sinks.
- The cost of finding good paths is amortized over the period of use of the paths (assuming a long-lived request).
Attribute-based routing: Directed diffusion

- Sinks generate interests that *diffuse* through the sensor network
Attribute-based routing: Directed diffusion

- Each node that receives an *interest* message stores:
  - the interest record
  - the neighbor who sent it
  - the data rate in which results are requested in a local *interest cache*.

- The initial requested data rate is set to a very low value.
Attribute-based routing: Directed diffusion

Interest cache of node N6:

<table>
<thead>
<tr>
<th>Interest</th>
<th>Neighbor</th>
<th>Data Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>animal=elephant,</td>
<td>N0</td>
<td>every 1 hour</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>animal=elephant,</td>
<td>N4</td>
<td>every 1 hour</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Event
Attribute-based routing: Directed diffusion

- A source computes the highest data rate among all its gradients for a certain interest.

- It sends event records to all neighbors for which it has a gradient for a particular event interest.

- A node that receives an event record checks its cache to see if it has matching interests.
  - If not, it drops the message
  - Otherwise it processes the message as follows:
    - The node caches recently seen event records. If the event record has not already arrived at the node through another path, the node forwards it to interested neighbors.
Event records are propagated back to the sink through multiple paths at the initially low data rate.
The sink **reinforces min-delay path**, by sending the same interest to neighbor N4 with higher requested data rate.
N4 reinforces min-delay path, by sending the same interest to N1 with higher requested data rate.
Attribute-based routing: Directed diffusion

The source propagates event records to the sink along the reinforced path with a high frequency.
Attribute-based routing: Directed diffusion

- Directed diffusion is a distributed algorithm – it works only with neighbor-to-neighbor interaction.
- The gradient mechanism is used to reinforce routing along the best paths.

- Details of the algorithm can be found below:
Routing protocols for sensor networks

• Geographic routing
• Attribute-based routing
• Tree-based routing
Tree-based routing

- Sensor network applications often require periodic gathering of data from multiple sensor nodes to a single base-station
  - many-to-one communication
- How to identify a suitable paths from source sensor nodes to the base-station?
- The base-station initiates a tree construction mechanism to establish paths from sensor nodes to itself.
- Sensor nodes then forward their readings along the paths of the constructed tree.

Tree-based routing

- Nodes initially set their hop-count to the base-station to infinity.
- The base-station broadcasts a beacon with beaconHopCount = 0.
- When a node hears a beacon, it compares its own nodeHopCount with the beaconHopCount.
  - If nodeHopCount > beaconHopCount + 1
    - nodeHopCount = beaconHopCount + 1
    - nodeParent = beaconSender
    - beaconHopCount++
    - rebroadcast beacon
  - Otherwise ignore beacon
- In this way, each node selects a parent in the shortest path to the base-station.
- After tree construction, each node forwards data via its parent to the base-station.
Summary

• Specialized MAC and routing protocols have been designed for sensor networks. They typically aim at preserving energy and adapting to rapid changes in the network topology.

• S-MAC is a medium access control protocol that tries to minimize energy consumption by scheduling nodes to turn off the radio periodically.

• Common approaches to routing data in sensor networks are: geographic, attribute-based and tree-based.
Related Reading


  http://www-csag.ucsd.edu/teaching/cse291s03/Readings/directed_diffusion.pdf

Data management for sensor networks

Distributed database approach, query processing, storage management
Distributed data management

- Each node is equipped with sensors and generates data
- The network is viewed as a distributed database **
- Research challenges:
  - How to query data?
  - How to store data?

** e.g. TinyDB: [http://www.cs.berkeley.edu/~franklin/Papers/TinyDBTods.pdf](http://www.cs.berkeley.edu/~franklin/Papers/TinyDBTods.pdf)
How to query sensor data?

• A declarative approach to tasking sensors using an SQL-like language:

• Example of a long-running query:
  
  ```sql
  SELECT nodeid, light, temp
  FROM sensors
  SAMPLE INTERVAL 1s
  FOR 10s
  ```
How to query sensor data?

• Example of a snapshot query:
  
  ```sql
  SELECT nodeid, pressure
  FROM sensors
  where pressure > thresholdValue
  ```

• Example of an aggregate query:
  
  ```sql
  SELECT nodeid, AVG (pressure)
  FROM sensors
  WHERE location in (100,100,200,200)
  [SAMPLE INTERVAL 100s
  FOR 1000s]
  ```
How to query sensor data?

• Example of an aggregate query with grouping:

  SELECT room, AVG(volume)
  FROM sensors
  WHERE floor = 6
  GROUP BY room
  HAVING AVG(volume) > threshold
  SAMPLE INTERVAL 30s
How to query sensor data?

• Example of a temporal aggregate query:
  SELECT WINAVG (volume, 30s, 5s) 
  FROM sensors 
  SAMPLE INTERVAL 1s 
  (will report avg volume over the last 30 sec once every 5 sec 
  sampling once per second)

• Example of an event-based query:
  ON EVENT bird-detect (loc):
    SELECT AVG (light), AVG (temp), event.loc 
    FROM sensors AS s 
    WHERE dist(s.loc, event.loc) < 10m 
    SAMPLE INTERVAL 2s FOR 30s
How to query sensor data?

• Example of a lifetime-based query:
  
  ```
  SELECT nodeId, accel
  FROM sensors
  LIFETIME 30 days
  ```

  – The query above specifies that the network should run for at least 30 days, sampling acceleration sensors at a rate that is as quick as possible and still satisfies the lifetime goal.
Query processing

• Query dissemination:
  – Then queries are disseminated in a simple binary format into the sensor network.

• Query execution
  – Finally, queries are executed in a distributed manner, and results are propagated back to the gateway.
Query dissemination

• The query is first broadcast from the gateway (root of the network).

• When a node hears the query, it decides
  – if the query applies locally (if there is a non-zero probability that the node can produce pertinent results for this query)
  – If the query needs to be broadcast to the children of the current node in the routing tree. This depends on whether the query applies to any of the current node’s descendants.
Query dissemination
Semantic routing trees (SRTs)

- SRTs are designed to help nodes decide if any of their descendants needs to participate in a given query.

- Each node stores a single uni-dimensional interval representing the range of a sensor attribute $A$ beneath each of its children.

- When a query with a predicate $p(A)$ arrives at a node, the node checks if any child has a range of $A$ values some of which could satisfy $p(A)$. If so, the node forwards the query to that child.

- If the query also applies locally, the node starts executing the query.

- If the query does not apply at the current node, nor at any of its children, it is ignored.
Query dissemination
Semantic routing trees (SRTs)

select light
from sensors
where x>3 and x<7

![Semantic Routing Trees Diagram]

- **SRT(x)**
  - 2: [1,1]
  - 3: [5,10]
  - 4: [5,5]
  - 5: [10,10]
Query dissemination
Semantic Routing Trees

• How to build SRTs?
  – In tree-based routing, nodes typically select as their parent the neighbor node on the shortest (or most reliable) path to the root. Suppose that there are more than one such neighbor nodes, i.e. more than one candidate parents.

  – With SRTs, the choice of parent depends also on semantic properties, for example, how the values of the indexed attribute on the local node compares to those on the candidate parent nodes.

  – For example, in the closest-parent approach, nodes select among their candidate parents, the one whose attribute value is closest to their own.
Query execution

- During every epoch of query execution, each node performs several operations:
  - samples local sensors
  - receives data from its children
  - processes local and received data
  - sends intermediate results to its parent
  - stores data locally (optional) for robustness
Query execution
Prioritizing data delivery

• Sensor samples and incoming messages, once processed, are inserted into a radio queue for delivery.
• In networks with high data rates, this queue may overflow.
• Potential approaches to deal with queue overflow:
  ➢ *Naïve*: no tuple is considered more valuable than any other. Tuples are dropped if they do not fit in the queue.
  ➢ *Winavg*: the two results at the head of the queue are averaged to make room for new results.
  ➢ *Delta*: Tuples are prioritized dependent on their difference from the most recent value successfully transmitted from this node. At each point in time, the tuple with the highest score is delivered. The tuple with the lowest score is evicted when the queue overflows.
  ➢ Other approximation and compression schemes, e.g. Fourier
Aggregate query execution

Centralized processing:
• send raw data to gateway
• evaluate query centrally at the gateway

With centralized processing, results at each edge grow linearly in the number of descendant nodes.

```
SELECT SUM(s) FROM SensorData s WHERE s.nest = empty EVERY 60 min
```
Aggregate query execution

In-network processing:
• evaluate partial aggregate locally at each node
• send partial aggregate to parent node

With in-network processing, the number of results at each edge remains constant.
⇒ Reduces communication overhead
⇒ Reduces energy consumption
⇒ Increases network lifetime

SELECT SUM(s) FROM SensorData s WHERE s.nest = empty EVERY 60 min
Summary of issues in query processing for sensor networks

• Query language:
  – Declarative SQL-like queries; snapshot vs. long-running vs. event-based queries; aggregate vs. temporal aggregate vs. non-aggregate queries

• Query dissemination:
  – SRTs can help reduce the cost of query dissemination

• Query execution
  – Data prioritization; in-network aggregation

• http://www.cs.berkeley.edu/~franklin/Papers/TinyDBTods.pdf
Related reading

• Query processing for sensor networks
RF-based positioning
RF-based systems

• Existing 802.11 wireless LANs can be leveraged for indoor positioning

• Two classes of WLAN location systems:
  – Client-based
  – Infrastructure-based

• In a client-based system
  – APs transmit frames
  – Clients use signal strength of AP-transmitted frames to infer location
Phases of RF-based location system

- **Offline training phase**
  - Record the strength of signals received from APs at selected locations
  - Build a *radio map*

- **Online location determination phase**
  - Take signal strength samples at the client from APs
  - Search the radio map to estimate user location
Deterministic fingerprinting approach

1. Signal strength at each location is represented by a scalar value (e.g. mean value)

2. Deterministic approaches are used to estimate user location

Example: *Radar* system [Infocom’00] (it uses the nearest neighborhood technique)
Probabilistic fingerprinting approach

1. Signal strength at each location is represented by a probability distribution
2. Probabilistic approaches are used to estimate user location

Example: **HORUS** system (Mobisys’05)
Wireless channel variation
Temporal

Normalized histogram of the received signal strength from one AP over a period of 5 mins

Fig. 1 in (Mobisys’05)
Temporal variation

Autocorrelation function of the samples collected from one AP at a fixed position.

Do not assume independence when sampling from the same AP!

Fig. 2 in (Mobisys’05)
Monotonically increasing function between the average signal strength and number of samples received from an AP.

Fig. 3 in (Mobisys’05)
Wireless channel variation
Spatial

Figure 4: Large-scale variations: Average signal strength over distance.

HORUS [Mobisys’05]

Figure 5: Small-scale variations: Signal strength contours from an AP in a 30.4 cm (12 inches) by 53.3 cm (21 inches) area.
Localisation problem

Assumption
At each 2D location $x$ (in discrete or continuous location space $X$), we get signal strengths from $k$ access points

Problem
Given a signal strength vector $s = (s_1, ..., s_k)$, find the location $x$ that maximizes the probability $P(x/s)$
HORUS: offline phase

Map Builder

• During the offline phase, Horus estimates the signal strength histogram for each access point at each location

• It then uses an autoregressive model to capture the correlations between different samples from the same AP
HORUS: offline phase

Map Builder

The signal strength histogram is approximated by a Gaussian with mean $\mu$ and variance $\sigma^2$.

If we represent the signal strength time series as an autoregressive model

$$s_t = \alpha \cdot s_{t-1} + (1 - \alpha) \cdot \nu_t \quad 0 \leq \alpha \leq 1$$

the distribution of the average of $n$ correlated samples is a Gaussian with mean $\mu$ and variance

\[
\frac{1 + \alpha}{1 - \alpha} \sigma^2
\]

parameters stored in the radio map.
HORUS: online phase

Discrete Space Estimator

1. Average the value of n consecutive samples to get vector s
2. Then search for the location of maximum probability given s

\[
\text{argmax}_x P(x / s) = \text{argmax}_x P(s / x)
\]

using radio map to compute \( P(s/x) \) as follows:

\[
P(s / x) = \prod_{i=1}^{k} P(s_i / x)
\]
HORUS: online phase

Continuous Space Estimator

• Centre of mass technique
  Obtain the centre of mass of the top N most likely locations

• Time averaging technique
  Use a time window of size W
  Average the last W location estimates obtained by the discrete-space or continuous-space estimator

\[
x = \frac{\sum_{i=1}^{N} p(i)x(i)}{\sum_{i=1}^{N} p(i)}
\]
HORUS: online phase

Small-Scale Compensator
• The radio map does not capture small scale variations
• How to detect small-scale variations?
  – Exploit the fact that users’ locations cannot change too fast
• How to compensate for small-scale variations?
  – Slightly perturb the signal strength observations from each of k access points
  – Select the perturbed combination that is the nearest to the previous user location
Summary

• HORUS is an example of radio-based fingerprinting method used for indoor positioning
• Offline phase: used to build radio map
  – Online phase: used to estimate position given radio map
• Two flavours:
  – Discrete
  – Continuous
• Two optimisations:
  – Exploit temporal correlations in signal strength observations
  – Compensate for small scale signal strength variations
Related reading

Delay-Tolerant Networks (DTNs)

Characteristics, applications, research challenges, routing protocols
Characteristics of DTNs

- DTNs lack continuous network connectivity
- Not always possible to find a complete end-to-end route from a source to a destination
- Traditional routing protocols for ad hoc networks, like AODV, fail in DTNs
- Need for asynchronous message forwarding
- Users tolerate delay in end-to-end delivery
Examples of delay-tolerant networks

Vehicular DTN

DTN project of DARPA

CASSIOPE: satellite project of the Canadian Space Agency
Applications

• Terrestrial mobile networks
  – vehicular ad hoc networks
  – buses forwarding messages from villages to cities in developing countries

• Military ad hoc networks
  – soldiers in a hostile environment that often get disconnected, or where data traffic may have to wait for high-priority voice traffic

• Sensor / actuator networks
  – swarm of helicopters monitoring a region
  – fixed sensors with a low duty cycle to conserve power
Research challenges

• **Uncertainty about contact schedules** - when pairs of nodes will come within communication range

• **Uncertainty about contact capacity** - how much data can be exchanged between nodes when they meet

• **Limited buffer space** - intermediate routers require considerable buffer space to store messages waiting for future communication opportunities

• **Scarce node resources** – e.g. sensor nodes with limited energy and processing power
Routing protocols for DTNs

• Criteria used to classify routing protocols for DTNs:
  – Replication
    • Some protocols replicate messages to increase their chance of getting delivered within a delay bound.
    • Others only forward a single copy of each message
  – Knowledge of network state
    • Some protocols require nodes to make decisions based on local node contacts that are currently available.
    • Others require nodes to have more global knowledge of the network state (e.g. statistics, or exact knowledge of contact schedules across the network)
Routing protocols for DTNs

• Flooding protocols:
  – They rely on replicating messages to enough nodes so the destination receives them.

• Forwarding protocols:
  – They rely on knowledge about the network to select the best path to the destination. Typically, they do not use replication.
Flooding protocols

- **Direct contact:**
  - This protocol waits until the source comes into contact with the destination before forwarding the data.
  - It is a degenerate case of the flooding family of protocols, where the set of relay nodes contains only the destination.
  - It does not require any information about the network.
  - It uses exactly one message transmission.
  - It only works if the source contacts the destination.
Flooding protocols

- Direct contact

$t_1$ $t_2$ $t_3$ $t_4$

\begin{itemize}
  \item \textbf{source}
  \item \textbf{destination}
\end{itemize}
Flooding protocols

• Two-hop relay:
  – The source copies the message to the first \( n \) nodes that it contacts
  – The source and the relays hold the message and deliver it to the destination
  – There are at most \( n+1 \) copies of the message in the network
  – Compared to the direct contact protocol, the two-hop relay protocol has better chances of delivering the message, and with lower delay
  – However, if the \( n+1 \) nodes never reach the destination, the message will not be delivered
Flooding protocols

- Two-hop relay (n=3, copies=3+1)
Flooding protocols

• Tree-based flooding
  – It extends two-hop relay, by allowing relays to make copies of the message.
  – Tree-based flooding can deliver messages to destinations over multiple hops

• Variants:
  – Limit the depth of the tree: each node can make unlimited copies, but the message can travel a maximum of n hops from the source
  – Limit the depth and the breadth of the tree: each node can make at most m copies, and each message can travel at most n hops
  – Limit the total number of copies: when a node makes a copy it distributes the responsibility for making half of its current copies to the other node
Flooding protocols

- Tree-based flooding (depth=2)

  - t1
  - t2
  - t3
  - t4

  [Diagram showing the flooding protocol with source, relay, and destination nodes at different time steps.]

- [Legend for the diagram: source, relay, destination]
Flooding protocols

• Epidemic routing
  – Messages are tagged with unique IDs
  – When two nodes connect, they exchange summary vectors, i.e. info about the message IDs that they hold in their buffers
  – They then exchange the messages that they do not have, so that eventually they have the same messages in their buffers
  – Each message is sent over all paths in the network => high redundancy, but also robustness and low latency
  – Death certificates can be issued to stop propagating a message after it has been delivered
  – Buffer management is an important issue
Flooding protocols

- Epidemic routing

<table>
<thead>
<tr>
<th>MsgID</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>asadfj</td>
</tr>
<tr>
<td>2</td>
<td>sdfsss</td>
</tr>
<tr>
<td>5</td>
<td>sdfdss</td>
</tr>
<tr>
<td>8</td>
<td>34sd</td>
</tr>
<tr>
<td>10</td>
<td>sdfsf</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MsgID</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>asadfj</td>
</tr>
<tr>
<td>3</td>
<td>sss</td>
</tr>
<tr>
<td>5</td>
<td>sdfdss</td>
</tr>
<tr>
<td>7</td>
<td>xxxx</td>
</tr>
<tr>
<td>10</td>
<td>sdfsf</td>
</tr>
</tbody>
</table>
Forwarding protocols

• Location-based routing
  – A message is forwarded to a potential next hop if that node is closer to the coordinate space than the current custodian
  – Three problems:
    • Even if the distance between two nodes is small, there is no guarantee that they will be able to communicate.
    • The message can fall into a local minimum and not be able to reach the destination.
    • Nodes (including the destination) move continuously – hard to know their coordinates.
Forwarding protocols

• Gradient routing
  – Each node is assigned a weight denoting its suitability to deliver messages to a given destination
  – When a node contacts another node that has a higher weight, it passes the message to it.
  – The message follows a gradient of improving utility function values towards the destination.
  – Each node stores a metric for each destination based on:
    • time of last contact with the destination
    • remaining energy
    • mobility
  – Utility values take time to propagate through the network => It can initially take a long time for a good custodian to be found.
Forwarding protocols

- **Link metrics**
  - These protocols build a topology graph, assign weights to each link and run a shortest path algorithm to find best paths.
  - Different protocols use different performance metrics to find the best path: bandwidth, latency, delivery ratio.
  - They require a lot of information about the network, for example accurate contact schedules.
  - If contact schedules are not known, estimate link weights based on recently observed network connectivity.
  - In dynamic conditions, it is recommended to re-evaluate the best path dynamically at each relay node.
Summary of DTN routing protocols

• Delay-tolerant networks lack synchronous end-to-end connectivity between each source-destination pair
• New routing protocols have been devised that handle the lack of continuous connectivity
• To route messages in a DTN, they exploit either message redundancy or knowledge about the network connectivity
  – Flooding protocols generate multiple replicas of a message and deliver it without exploiting knowledge about the network connectivity.
  – Forwarding protocols typically forward one copy of the message and exploit knowledge about the network connectivity to forward it to the destination.