Principles of Wireless Sensor Networks

http://www.ee.kth.se/~carlofi/teaching/pwsn-2011/wsn_course.shtml

Lecture 8
Stockholm, November 11, 2011

Routing, Zigbee, and RPL

Carlo Fischione

Royal Institute of Technology - KTH
Stockholm, Sweden
e-mail: carlofi@kth.se

Today's learning goals

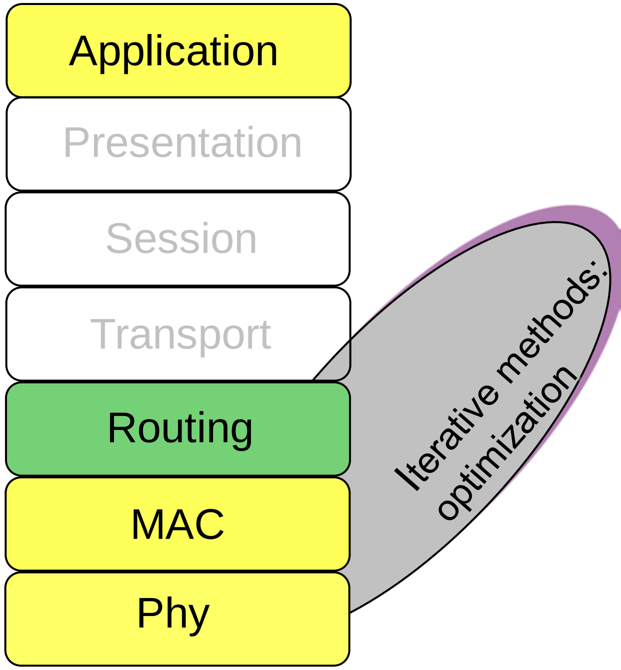
- What are the basic routing options for sensornet?
- How to compute the shortest path?
- How to compute multipath routings?
- Which routing is used in standard protocols?

Today's lecture

- Previous lecture
  - MAC protocols, IEEE 802.15.4

- Today's lecture will focus on routing

Protocol stack



Application

Presentation

Session

Transport

Routing

MAC

Phy

Iterative methods: optimization

Outline

- **Classification of routing protocols for WSNs**
- The shortest path routing
- Multipath routing
- Routing algorithms in standardized protocol stack

Routing protocols

- Goal: Derive a mechanism that allows a packet sent from an arbitrary node to arrive at some arbitrary destination node
  - The routing & forwarding problem
  - Routing: Construct data structures (e.g., tables) that contain information how a given destination can be reached
  - Forwarding: Consult these data structures to forward a given packet to its next hop
- Challenges
  - Nodes may move around, neighborhood relations change
  - Optimization may be difficult

Ad hoc routing protocols – classification

- When does the routing protocol operate?

- Option 1: Routing protocol **always** tries to keep its routing tables up-to-date
  - Protocol is proactive (active before tables are actually needed)

- Option 2: Route is only determined when actually needed
  - Protocol operates on demand

- Option 3: Combine these behaviors
  - Hybrid protocols

Proactive protocols - DSDV

- **Destination Sequence Distance Vector (DSDV)**
  - Based on Bellman Ford procedure (see later on)
  - Add **aging** information to route information propagated by distance vector exchanges; helps to avoid routing loops
  - Periodically send full route updates
  - On topology change, send incremental route updates
  - Unstable route updates are delayed

On Demand protocols – AODV

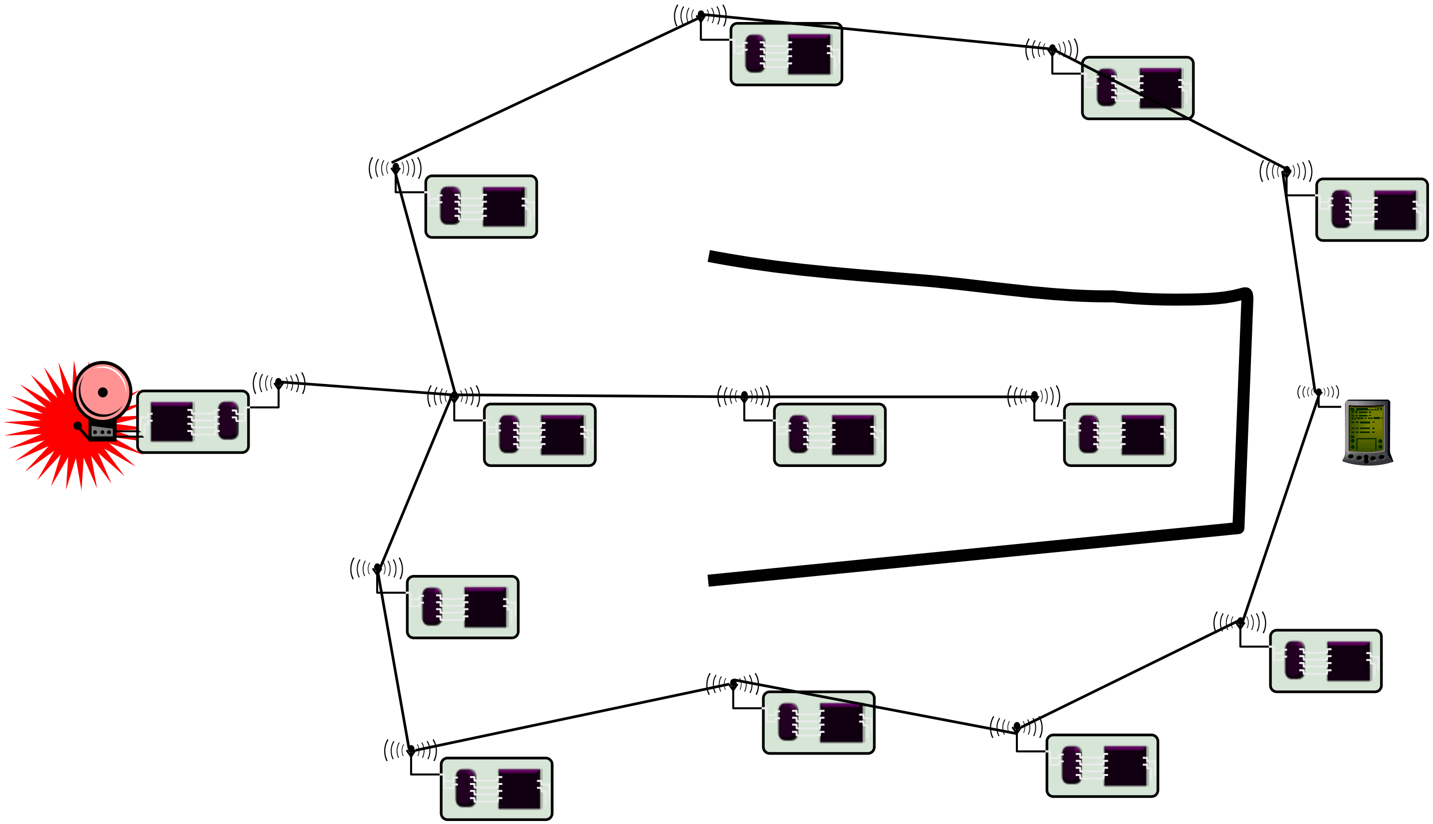
- **Ad hoc On Demand Distance Vector** routing (AODV)
  - Popular routing protocol on sensor networks
  - Route discovery procedure by special packets
  - Nodes maintain routing tables instead of source routing
  - Nodes remember from where a packet came and populate routing tables with that information

Geographic routing

- Routing tables contain information to which next hop a packet should be forwarded
  - Explicitly constructed
- Alternative: Implicitly **infer** this information from physical placement of nodes
  - Position of current node, current neighbors, destination known – send to a neighbor in the right direction as next hop
- Options
  - Send to any node in a given area – ***geocasting***
  - Use position information to aid in routing – **position-based routing**
    - Might need a **location service** to map node ID to node position

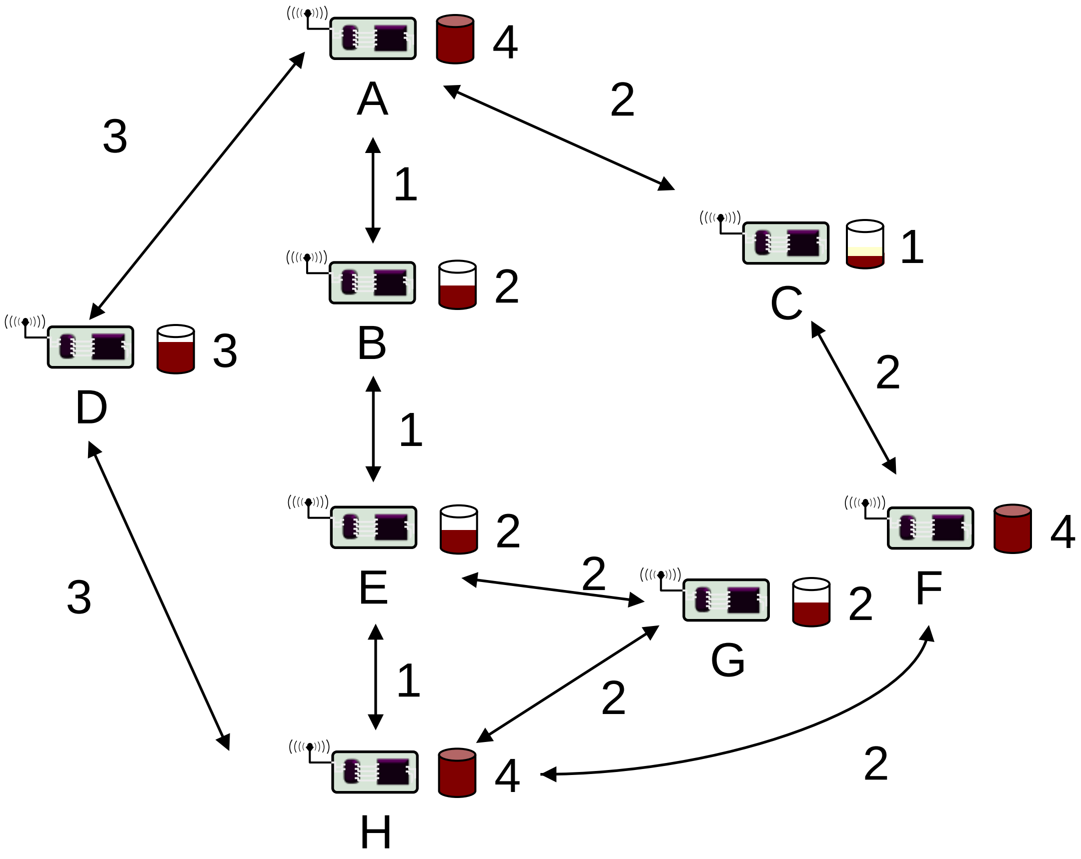
Problem: dead ends

- Simple strategies might send a packet into a dead end



Many options for routing

- Maximum total available battery capacity
  - Path metric: Sum of battery levels
  - Example: A-C-F-H
- Minimum battery cost routing
  - Path metric: Sum of reciprocal battery levels
  - Example: A-D-H
- Conditional max-min battery capacity routing
  - Only take battery level into account when below a given level
- Minimize variance in power levels
- Minimum total transmission power



Many options for routing

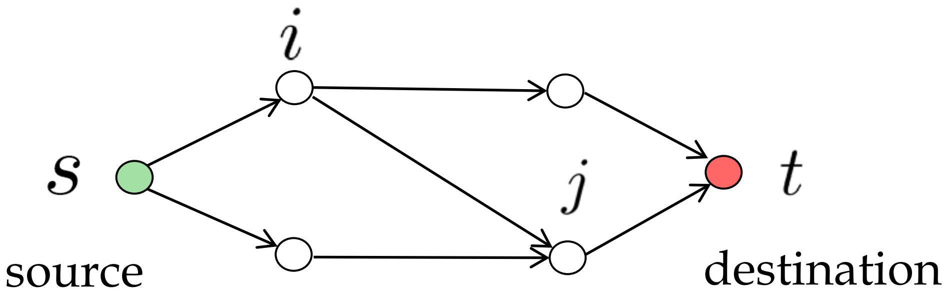
- Is there a basic way to model all these options?

- Yes, the shortest path problem...

Outline

- Classification of routing protocols for WSNs
- **The shortest path routing**
- Multipath routing
- Routing algorithms for standardized protocol stack

Some definitions



N	number of nodes
$\mathcal{N}$	set of nodes
$\mathcal{A} = \{(i,j)\}$	set of arcs
$\mathcal{G} = (\mathcal{N}, \mathcal{A})$	network
a_{ij}	routing cost on the link ij

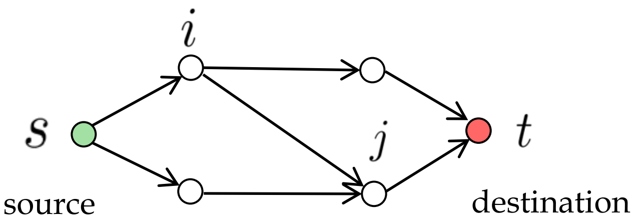
- Question: what is the shortest (minimum cost) path from source to destination?

The shortest path problem

$$\min_{\mathbf{x}} \quad \sum_{(i,j)\in\mathcal{A}} a_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in\mathcal{A}} x_{ij} - \sum_{j:(j,i)\in\mathcal{A}} x_{ij} = s_i \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ij} \geq 0 \quad \forall (i,j) \in \mathcal{A}$$



- x_{ij} binary variable. It can be also real, but remarkably if problem feasible, the unique optimal solution is binary

- The optimal solution gives the shorstest path source-destination

Applications of the shortest path problem

- The solution to this problem is very general and can be applied to

1. Routing over WSNs, used in WirelessHART, RPL,...
2. Dynamic programming
3. Project management
4. The paragraphing problem
5. ...

- In the following, we see the basic halgorithm to solve the problem

Generic shortest paths algorithm

- The solution of the problem can be achieved by combinatorial algorithms that don't use optimization

- We consider now a generic algorithm based on node labelling, the Generic shortest path algorithm

- The algorithm is the fundation of other more advanced algorithms widely used for routing such as

1. Dijkstra method
2. Bellman-Ford method

Complementaty slackness conditions for the shortest path problem

A label associated to a node $d_j = \begin{cases} \text{a scalar} \\ \infty \end{cases}$

Proposition: Let $d_1, d_2, \ldots, d_N$ be scalars such that

$$d_j \leq d_i + a_{ij}, \quad \forall (i,j) \in \mathcal{A}.$$

Let P be a path starting at a node i_1 and ending at a node i_k. If

$$d_j = d_i + a_{ij}, \quad \forall (i,j) \text{ of } P,$$

then P is a shortest path from i_1 to i_k.

Generic shortest path algorithm: the idea

- Complementary Slackness conditions (CS) is the fundation of the generic shortest path algorithm

- Some initial vector of labels is assigned $(d_1, d_2, \ldots, d_N)$

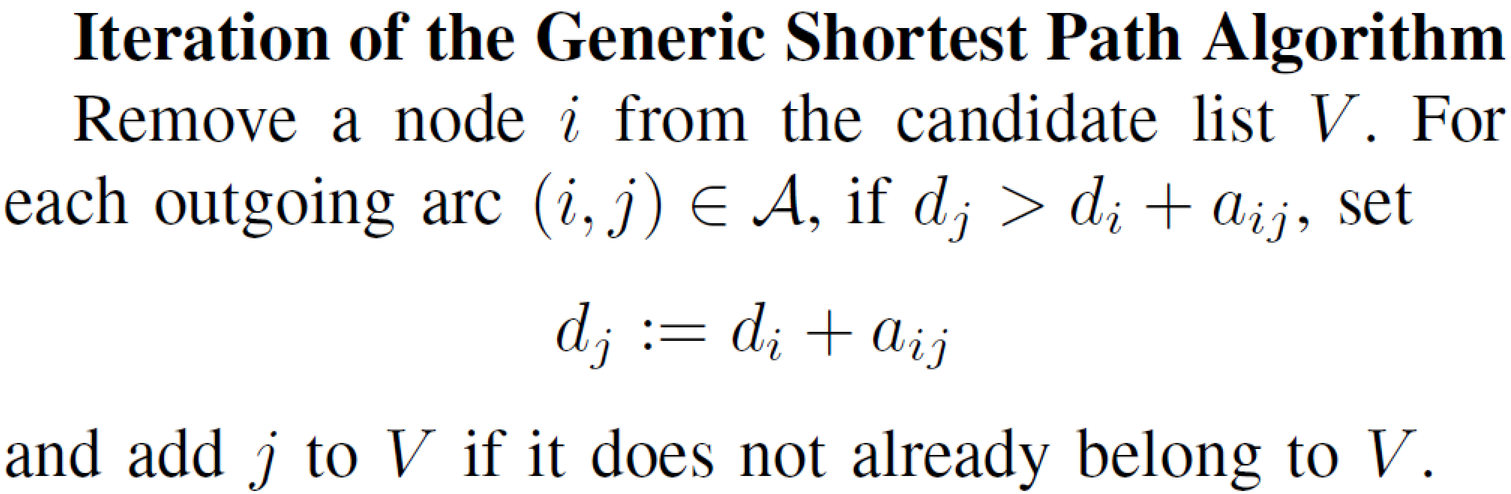
- The arcs (i, j) that violate the CS condition $d_j > d_i + a_{ij}$ are selected and their lables redefined so that

$$d_j := d_i + a_{ij}$$

- This is continued until the CS condition $d_j \leq d_i + a_{ij}$ is satisfied for all arcs (i, j)

Iterations of the algorithm

- Let initially be $V = \{1\} \quad d_1 = 0,\, d_i = \infty,\, \forall i \neq 1.$



Iteration of the Generic Shortest Path Algorithm

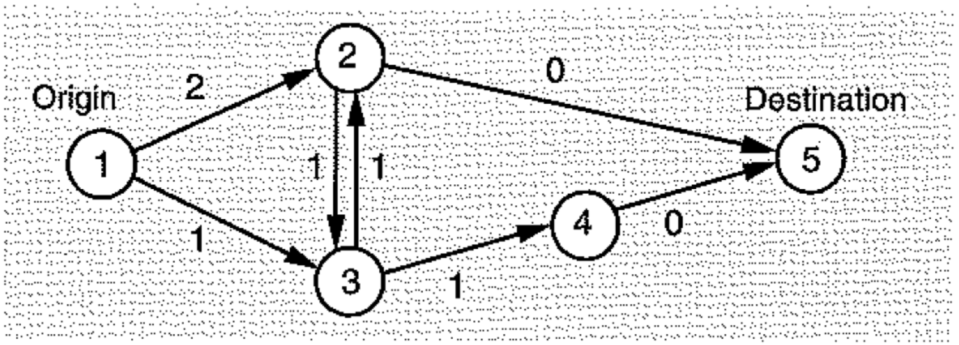
Remove a node i from the candidate list V. For each outgoing arc $(i, j) \in \mathcal{A}$, if $d_j > d_i + a_{ij}$, set

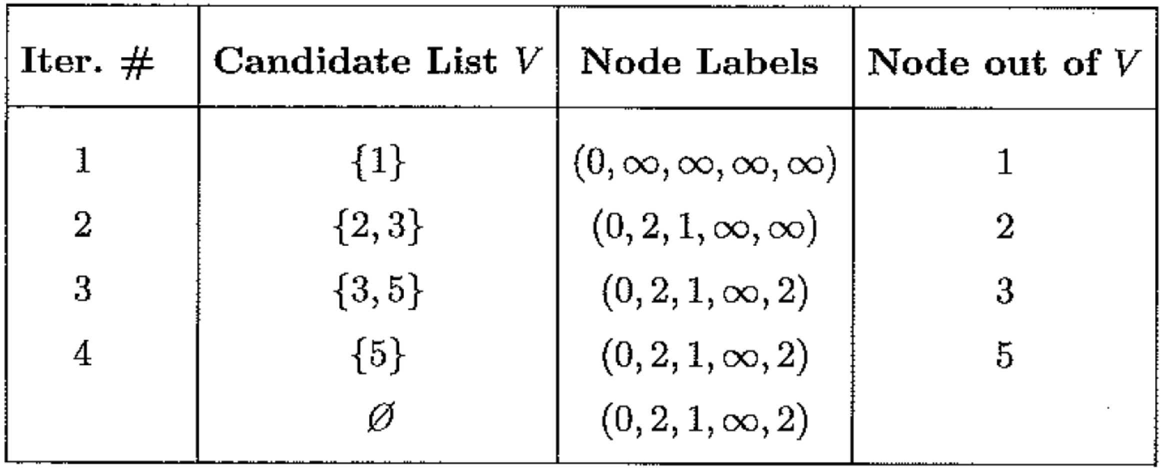
$$d_j := d_i + a_{ij}$$

and add j to V if it does not already belong to V.

- Node removal rule gives Dijkstra method or Bellman-Ford method

An example





Iter. #	Candidate List V	Node Labels	Node out of V
1	$\{1\}$	$(0, \infty, \infty, \infty, \infty)$	1
2	$\{2, 3\}$	$(0, 2, 1, \infty, \infty)$	2
3	$\{3, 5\}$	$(0, 2, 1, \infty, 2)$	3
4	$\{5\}$	$(0, 2, 1, \infty, 2)$	5
	$\emptyset$	$(0, 2, 1, \infty, 2)$	

Convergence of the algorithm (a)

Proposition: Consider the generic shortest path algorithm.

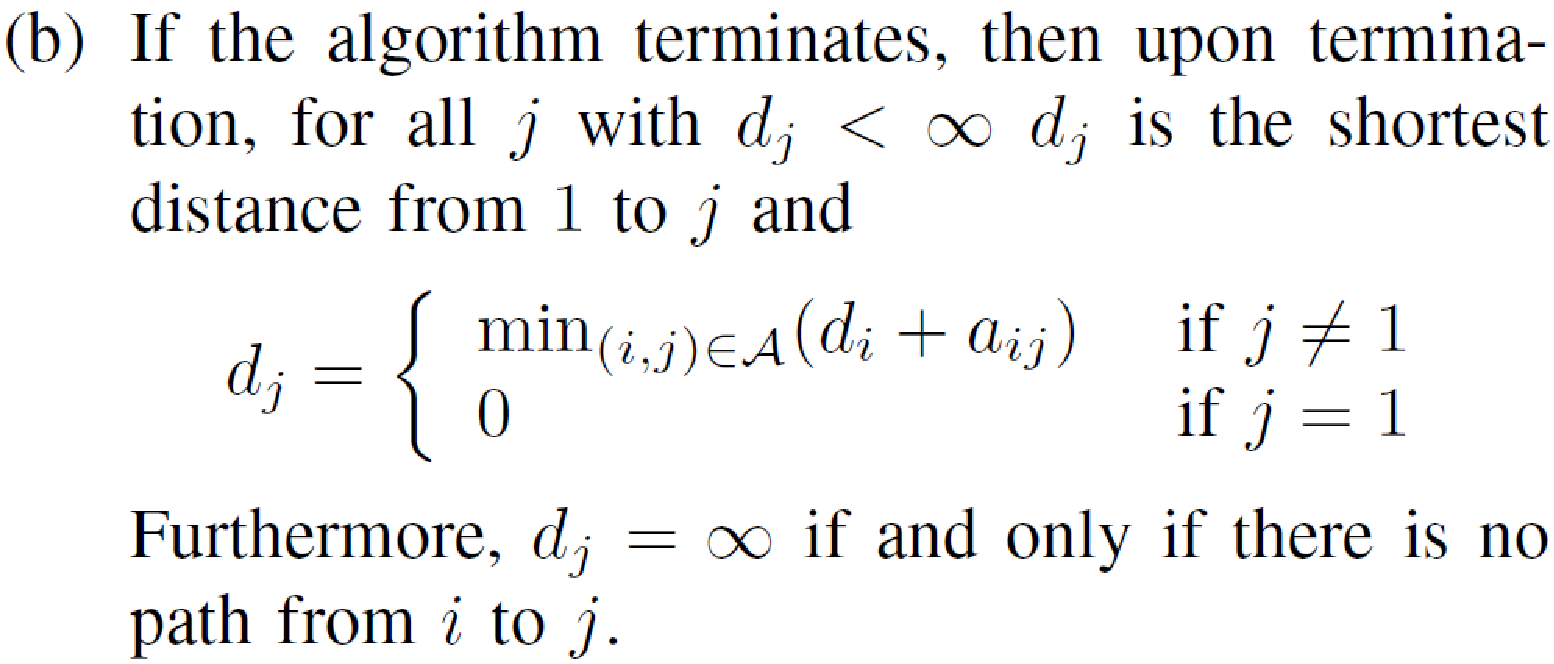
(a) At the end of each iteration, the following conditions hold:

(i) If $d_j < \infty$, then d_j is the length of some path that starts at 1 and ends at j.

(ii) If $i \ni V$, then either $d_i = \infty$ or else

$$d_j \leq d_i + a_{ij}, \quad \forall j \text{ such that } (i, j) \in \mathcal{A}.$$

Convergence of the algorithm (b)



(b) If the algorithm terminates, then upon termination, for all j with $d_j < \infty$ d_j is the shortest distance from 1 to j and

$$d_j = \begin{cases} \min_{(i,j)\in\mathcal{A}}(d_i + a_{ij}) & \text{if } j \neq 1 \\ 0 & \text{if } j = 1 \end{cases}$$

Furthermore, $d_j = \infty$ if and only if there is no path from i to j.

The Bellman-Ford equation

Convergence of the algorithm (c) (d)

(c) If the algorithm does not terminate, then there exists some node j and a sequence of paths that start at 1, ends at j, and have a length diverging to $-\infty$.

(d) The algorithms terminates if and only if there is no path that starts at 1 and contains a cycle with negative length.

Outline

- Classification of routing protocols for WSNs
- The shortest path routing
- **Multipath routing**
- Routing algorithms for standardized protocol stack

Multipath routing

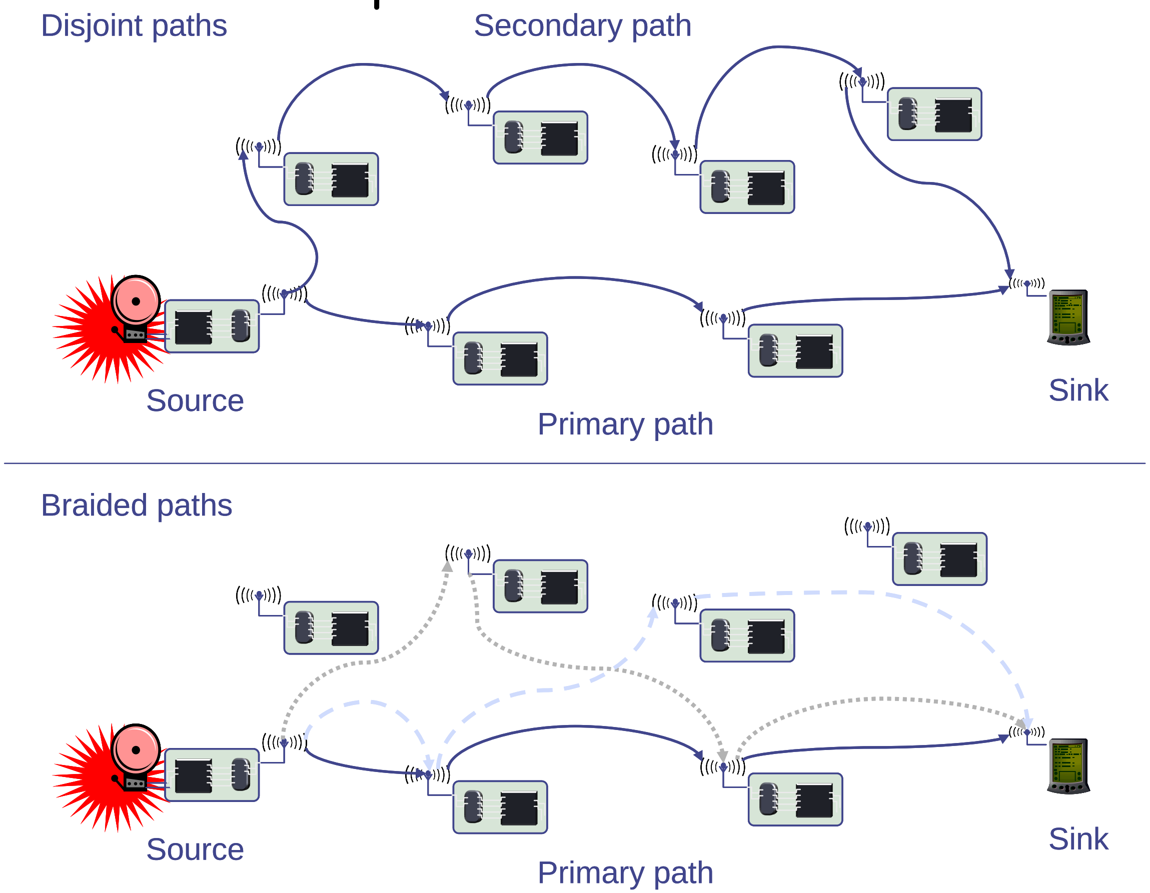
- The shortest path algorithm gives the best path from source to destination.

- The shortest path algorithm can be used to compute the first best, the second best, and so on paths

- Sending messages over multiple paths introuces diversity

Multipath routing

- Instead of only a single path, it can be useful to compute multiple paths between a source-destination pair



How to split optimally the messages over multiple paths?

- Of course, see Bertsekas and Titsiklis, Parallel and Distributed Computation, Numerical Methods:

1. Constrained optimization, pages 210-212

2. Distributed routing protocol over data networks, pages 414-418

Outline

- Classification of routing protocols for WSNs
- The shortest path routing
- Multipath routing
- **Routing algorithms for standardized protocol stack**

ZigBee

- http://www.zigbee.org/
- ZigBee covers the networking and application layers on top of IEEE 802.15.4
- Nodes:
  - IEEE 802.15.4 nodes
  - ZigBee coordinator: starts the network
  - ZigBee router
- Networks:
  - Star network
  - Tree network
  - Mesh network
- Routing
  - No transport protocol for end-to-end reliability (only hop-by-hop).
  - Tree routing: packets are sent to the coordinator, and then to the destination.
  - Mesh routing: AODV protocol for route discovery

ISA SP-100

- http://www.isa.org

- Standard for non critical process applications tolerating delays up to 100ms.

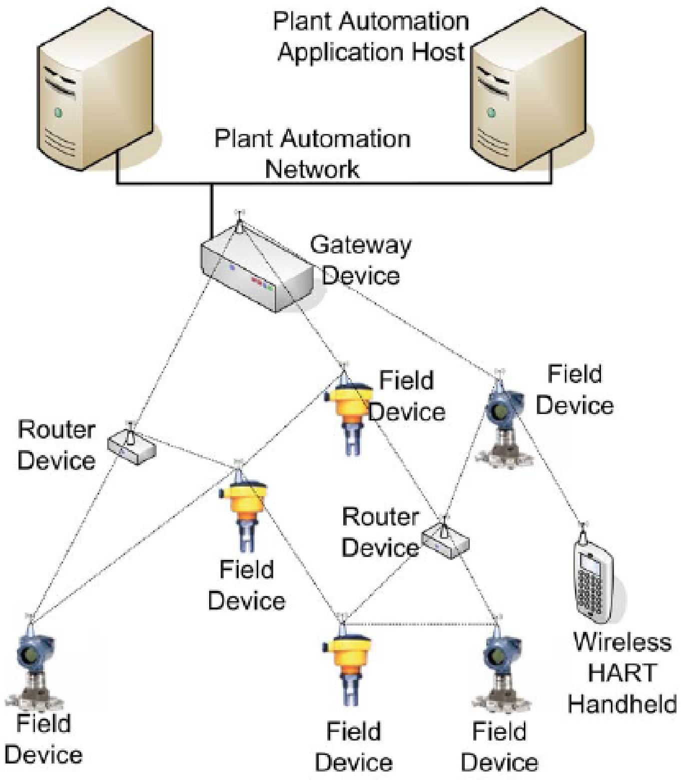
- It is based on IEEE 802.15.4 plus a new data link layer and adaptation layer between MAC and data link
  - Frequency hopping

WirelessHART

- http://www.hartcomm.org/
- Released in September 2007 as part of HART 7 specifications
- An open communication standard designed for process measurements and control applications
  - strict timing requirements
  - security concerns

WirelessHART network

- Field device, attached to the plant process
- Handheld, a portable computer to configure devices, run diagnostic and perform calibrations
- Gateway, that connect host applications to field devices
- Network manager, responsible for configuring the network, scheduling and managing communication



WirelessHART network

- Topology: Star, Cluster, Mesh

- Central network manager: maintains up-to-date routes and communication schedules for the network

- Basic functionalities:
  - Timer
  - Network wide synchronization
  - Communication security
  - Reliable mesh networking
  - Central network management

A Five Layers Architecture

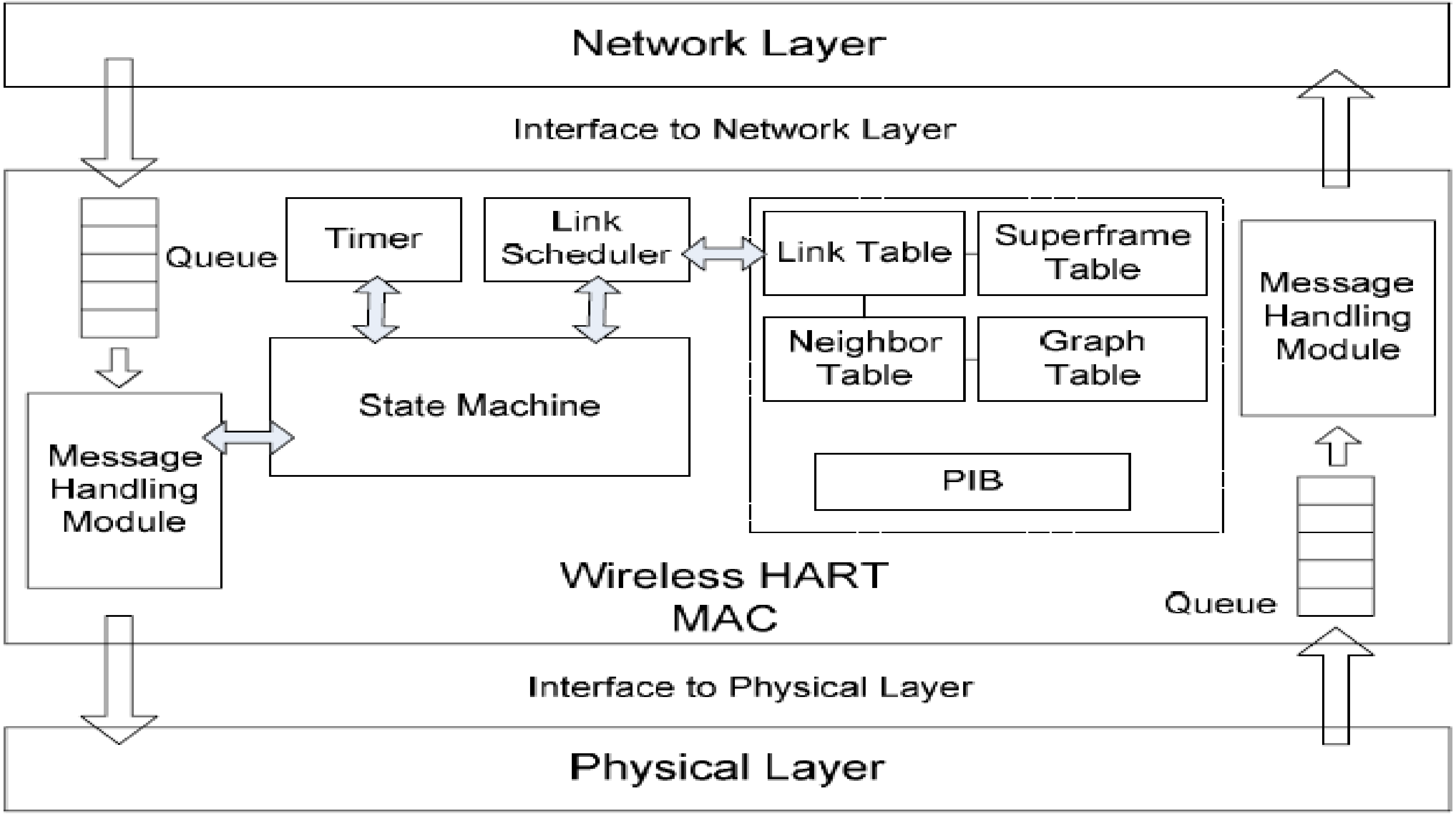
OSI Layer	HART	
Application	Command Oriented. Predefined Data Types and Application Procedures	
Presentation		
Session		
Transport	Auto-Segmented transfer of large data sets, reliable stream transport, Negotiated Segment sizes	
Network		Power-Optimized Redundant Path, Mesh to the edge Network
Data Link	A Binary, Byte Oriented, Token Passing, Master/Slave Protocol	Secure, Time Synched TDMA/ CSMA, Frequency Agile with ARQ
Physical	Simultaneous Analog & Digital Signaling 4-20mA Copper Wiring	2.4 GHz Wireless, 802.15.4 based radios, 10dBm Tx Power
	Wired FSK/PSK & RS 485	**Wireless 2.4 GHz**

Layers

- Physical layer:
  - similar to IEEE 802.15.4
  - 2.4-2.4835 GHz, 26 channels, 250 Kbps per channel

- Data link layer:
  - Network wide synchronization (a fundamental functionality)
  - TDMA with strict 10ms time slots
  - Periodical superfames
  - Channel blacklisting: the network administrator removes the channels with high interference.
  - Pseudorandom change of the channel for robustness to fading
  - TDMA security: industry-standard AES-128 ciphers and keys

WirelessHART MAC



WirelessHART Routing

- Graph Routing:
  - A graph is a collection of paths that connect network nodes
  - The paths in each graph is created by the network manager and downloaded to each individual network device
  - To send a packet, the source device writes a specific graph ID (determined by the destination) in the network header
  - All network devices on the way to the destination must be pre-configured with graph information that specifies the neighbors to which the packets may be forwarded
- Source Routing:
  - supplement of the graph routing aiming at network diagnostics
  - To send a packet to its destination, the source device includes in the header an ordered list of devices through which the packet must travel
  - As the packet is routed, each routing device utilizes the next network device address in the list to determine the next hop until the destination device is reached

ROLL: Ruting over Low Power Lossy Networks

- http://www.ietf.org/dyn/wg/charter/roll-charter.html

- ROLL is a Working Group of the Internet Engineering Task Force
- RPL, routing protocols for low power and lossy networks in Industrial and home automation, healthcare, smart grids

- Network assumption
  - many embedded devices with limited power, memory, and processing
  - Interconnection by a variety of links, such as EEE 802.15.4, Bluetooth, Low Power WiFi, wired or other low power Powerline Communications
  - End-to-end IP-based solution to avoid the problem of on-interoperable networks interconnected by protocol translation gateways and proxies

Summary

- We have studied routing protocol
  - Proactive
  - Reactive
  - Geographic

- The basic mechanism of routing is the shortest path optimization problem

- Once shortest paths are available, multipath routing can increase diversity

- These mechanisms are included in standards such as ZigBee, ISA100, WirelessHart and RPL

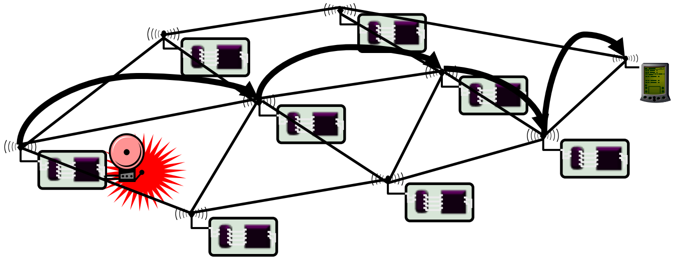
Project presentations, Nov. 15

- Make a presentation of max 15 minutes including questions
  - Phd Students
    - Motivation
    - Problem formulation
    - Approach to the solution
    - Results
    - Conclusions
  - Master Students:
    - Didactic overview of the investigated material

- Email by tonight title of the presentation and speakers

Basics of position-based routing

- "Most forward within range r" strategy
  - Send to that neighbor that realizes the most forward progress towards destination
  - NOT: farthest away from sender!



- Nearest node with (any) forward progress
  - Idea: Minimize transmission power
- Directional routing
  - Choose next hop that is angularly closest to destination
  - Choose next hop that is closest to the connecting line to destination
  - Problem: Might result in loops!