Principles of Wireless Sensor Networks

https://www.kth.se/social/course/EL2745/

Lecture 6
February 4, 2013

Carlo Fischione
Associate Professor of Sensor Networks
e-mail: carlofi@kth.se
http://www.ee.kth.se/~carlofi/

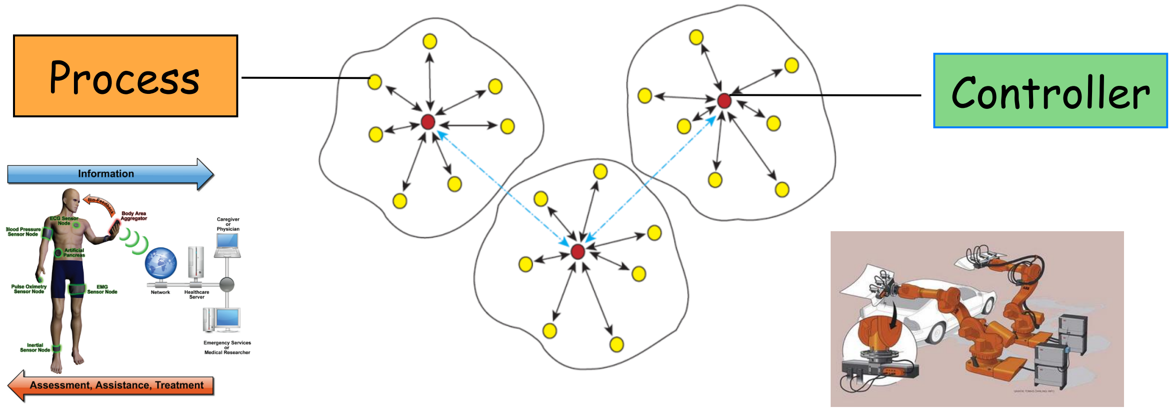
KTH Royal Institute of Technology
Stockholm, Sweden

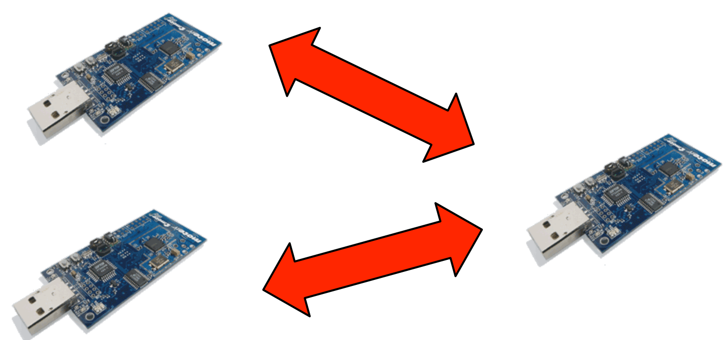
Course content

- Part 1
  - Lec 1: Introduction
  - Lec 2: Programming
- Part 2
  - Lec 3: The wireless channel
  - Lec 4: Physical layer
  - Lec 5: Mac layer
  - Lec 6: Routing
- Part 3
  - Lec 7: Distributed detection
  - Lec 8: Distributed estimation
  - Lec 9: Positioning and localization
  - Lec 10: Time synchronization
- Part 4
  - Lec 11: Networked control systems 1
  - Lec 12: Networked control systems 2
  - Lec 13: Summary and project presentations

Previous lecture

Application
Presentation
Session
Transport
Routing
MAC
Phy

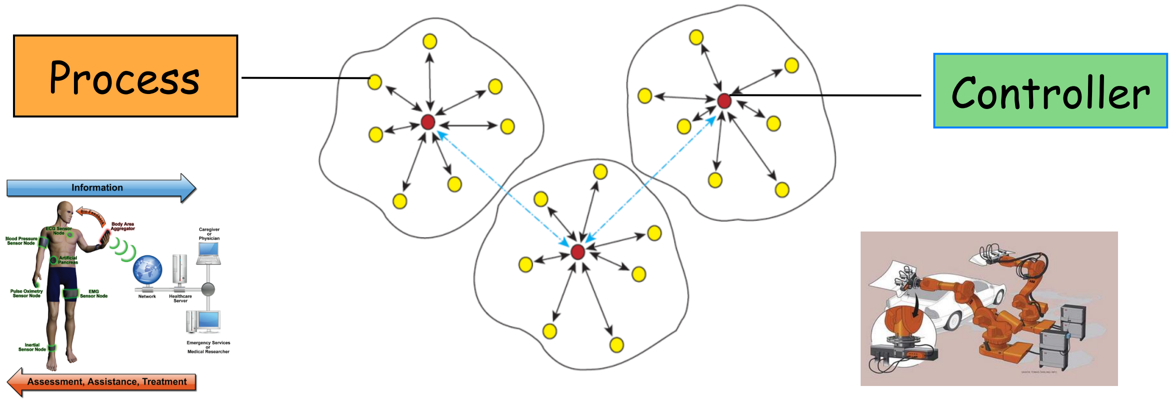


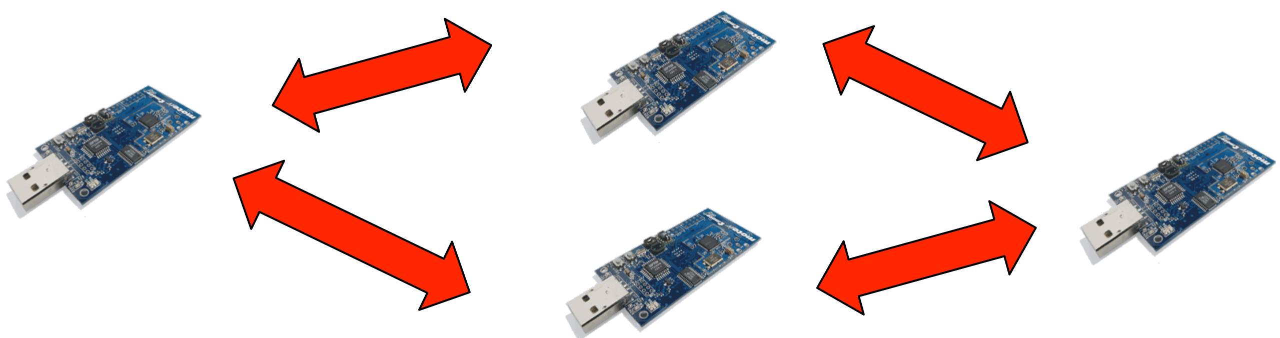


- When a node gets the right to transmit?
- What is the mechanism to get such a right?

Today's lecture

Application
Presentation
Session
Transport
Routing
MAC
Phy





On which path messages should be routed?

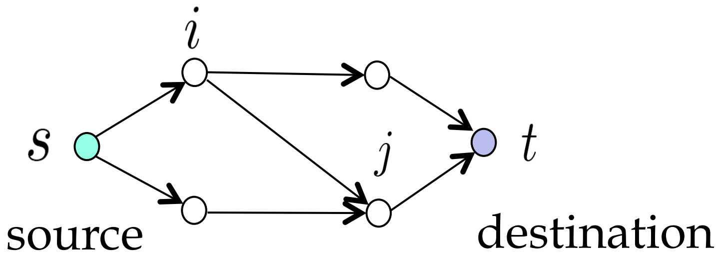
Today's learning goals

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

Outline

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

Routing protocols



- Derive a mechanism that allows a packet sent from an arbitrary node to arrive at some destination node

  - Routing information: data structures (e.g., tables) on how a given destination node can be reached by a source node

  - Forwarding: Consult these data structures to forward a given packet to its next hop node

- Challenges
  - Nodes may move, neighborhood relations change

Routing protocols classification

When the routing protocol operates?

1. **Proactive**: protocol always tries to keep its routing tables up-to-date and active before tables are actually needed
   - Example: Destination Sequence Distance Vector (DSDV), uses Bellman-Ford algorithm (see below)
2. **On demand**: route is only determined when needed by a node
   - Example: Ad hoc On Demand Distance Vector (AODV), nodes remember where packets came from and populate routing tables accordingly
3. **Hybrid**: combine the previous two

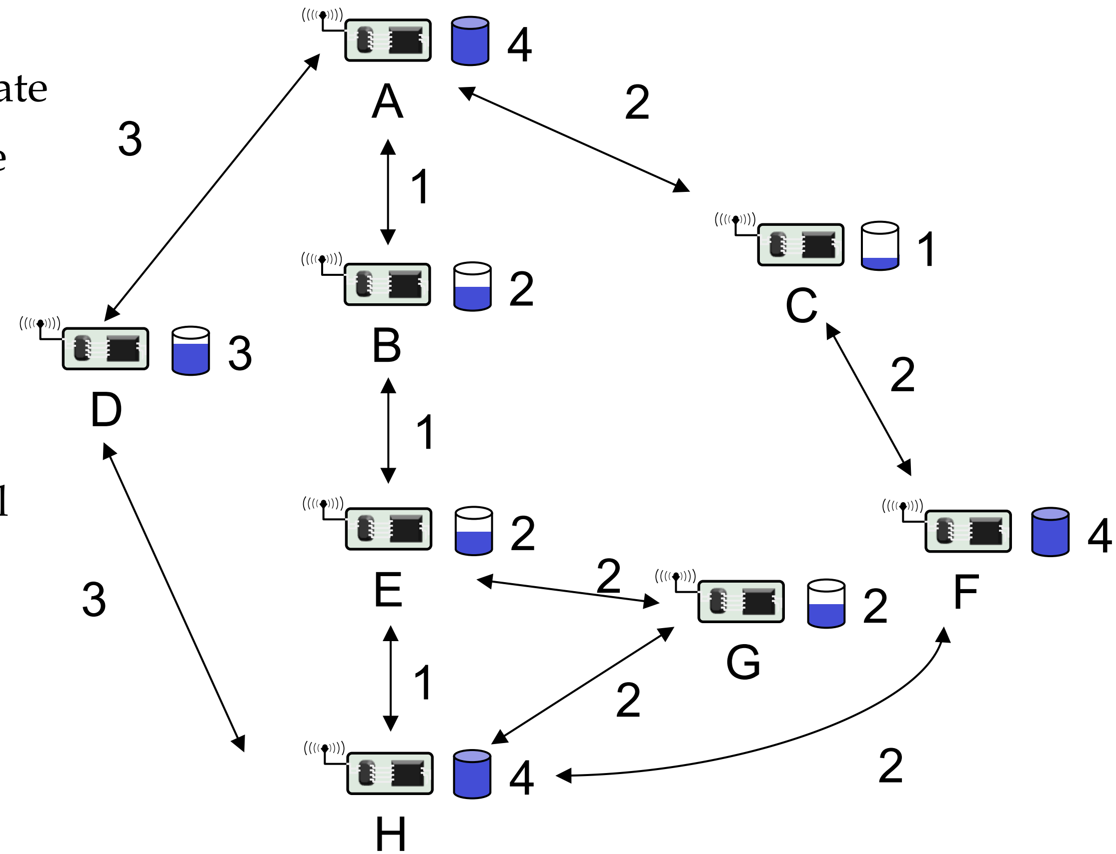
But how paths are built and chosen?

- We have seen a general classification of routing

- In practice,
  - how the routing structures (e.g., the tables) are built?
  - how the decision to select next hop is taken?

Many options for routing

1. Path with minimum delay
2. Path with minimum packet error rate
3. Path with maximum total available battery capacity
   - Path metric: Sum of battery levels
   - Example: A-C-F-H
4. Path with minimum battery cost
   - Path metric: Sum of reciprocal battery levels
   - Example: A-D-H
5. Path with conditional max-min battery capacity
   - Only take battery level into account when below a given level
6. Path with minimum variance in battery power levels
7. Path with minimum total transmission battery power



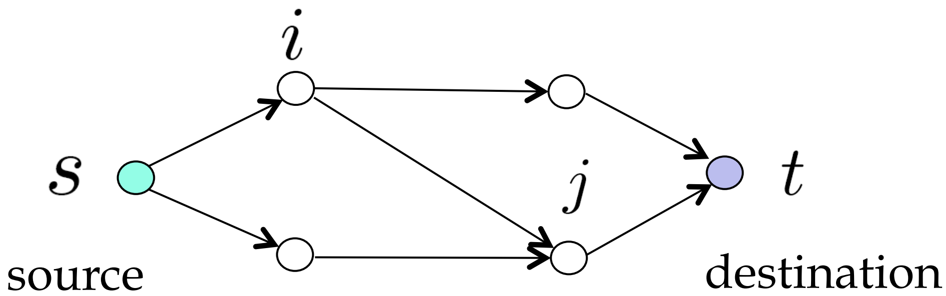
Many options for routing

- Is there a basic way to model all these options?
- Yes, the shortest path problem

Outline

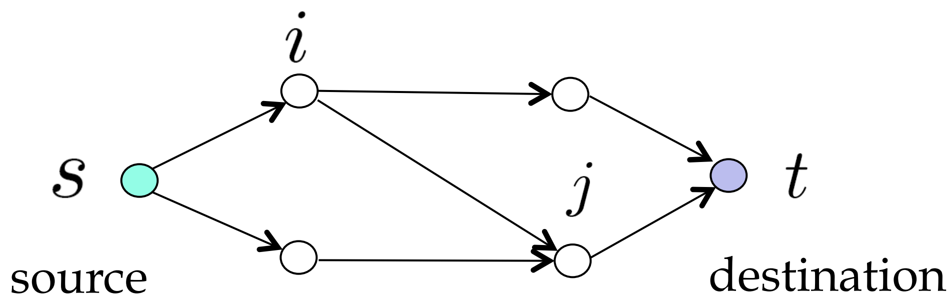
- Classification of routing protocols for WSNs
- **The shortest path routing**
- Routing algorithms for standardized protocol stack

The shortest path routing



- The shortest path routing problem is a general optimization problem that models ALL the cases above for routing
- In the following, we study the basic version, when in the network there is one source and one destination
- Multiple sources multiple destinations scenarios are a simple extension

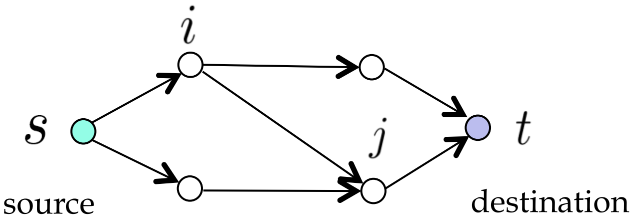
Definitions



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
	Examples: 1) MAC delay $i \rightarrow j$ 2) packet error rate $i \rightarrow j$

What is the shortest (minimum cost) path from source s to destination t ?

The Shortest Path optimization Problem



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

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

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

$$\mathbf{x} = [x_{12}, x_{13}, \ldots, x_{i_n i_{n+1}}, \ldots]$$

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

- The optimal solution gives the shorstest path source-destination

Applications of the Shortest Path Problem

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

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

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

- This problem is much more general and can be applied to
  1. Routing over WSNs, used in ROLL RPL, WirelessHART…
  2. Project management
  3. The paragraphing problem
  4. Dynamic programming
  5. …

How to solve the Shortest Path Problem

- Since it is an optimization problem, one could use standard techniques of optimization theory, such as Lagrangian methods

- However, the solution can be achieved by combinatorial algorithms that don't use optimization theory at all

- We consider now such a combinatorial solution algorithm, the Generic shortest path algorithm

- The Generic shortest path algorithm is the foundation of other more advanced algorithms widely used for routing (e.g., in ROLL RPL) such as
  1. Bellman-Ford method (see exercises)
  2. Dijkstra method (see exercises)

Complementary 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 in-nuce

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

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

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

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

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

Iterations of the Generic Shortest Path Algorithm

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

Iteration of the Generic Shortest Path Algorithm

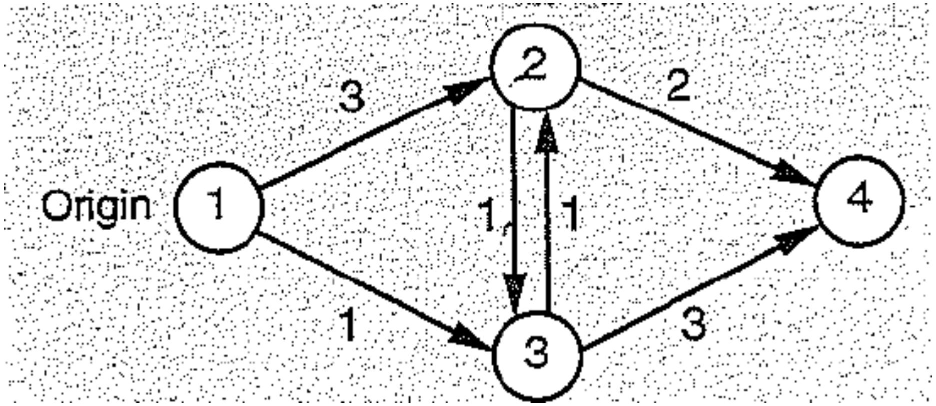
Remove a node i from the candidate list V. For each outgoing arc $(i, j) \in \mathcal{A}$, if $d_j > d_i + a_{ij}$, set

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

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

- The removal rule gives
  - Bellman-Ford method
  - Dijkstra method

An example



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

Figure 2.3: Illustration of the generic shortest path algorithm. The numbers next to the arcs are the arc lengths. Note that node 2 enters the candidate list twice. If in iteration 2 node 3 was removed from V instead of node 2, each node would enter V only once. Thus, the order in which nodes are removed from V is significant.

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.

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.

The convergence properties of the Generic Shortest Path Algorithm

- The convergence properties above are based on sound theoretical analysis

- They are the foundation over which routing protocols, such as the standardized ROLL RPL, are built

- Let's have a quick look at ROLL RPL and other standardized routing protocols

Outline

- Classification of routing protocols for WSNs
- The shortest path routing
- **Routing for standardized protocols**
  - **ROLL RPL**
  - ZigBee
  - ISA100
  - WirelessHART

ROLL: Routing over Low Power Lossy Networks

- ROLL is a Working Group of the Internet Engineering Task Force
  www.ietf.org/dyn/wg/charter/roll-charter.html

- ROLL RPL, IPv6 Routing Protocol for Low Power and Lossy Networks

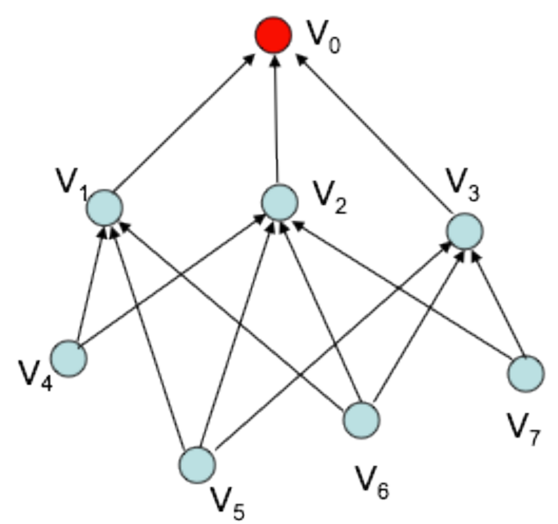
- RPL is intended for
  - Industrial and home automation
  - Healthcare
  - Smart grids

ROLL RPL assumptions

- Networks with many embedded nodes with limited power, memory, and processing
- Networks interconnected by a variety of protocols, such as EEE 802.15.4, Bluetooth, Low Power WiFi, wired or other low power Powerline communications
- End-to-end Internet Protocol-based solution to avoid the problem of non-interoperable networks interconnected by protocol translation gateways and proxies
- Traffic patterns
  1. Multipoint to Point (MP2P)
  2. Point to Multipoint (P2MP)
  3. Point-to-Point (P-2-P)

RPL is tree based

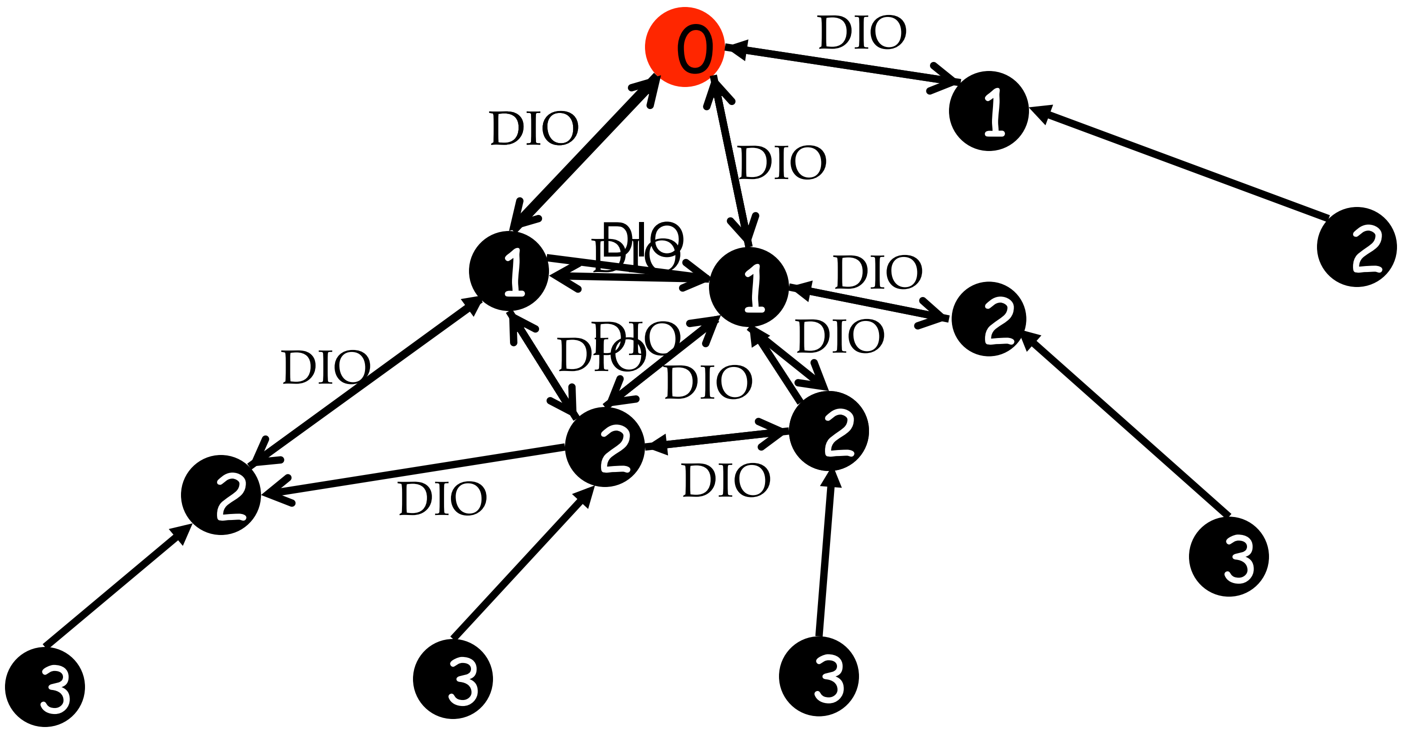
- RPL constructs destination-oriented directed acyclic graphs (DODAGs) i.e., trees sources-destinations
- Nodes build and maintain DODAGs by periodically multicasting messages, the DODAG Information Object (DIO), to their neighbors
- To join a DODAG, a node listens to the DIO messages sent by its neighbors and selects a subset of these nodes as its parents
- Packet forwarding metrics, the a_{ij} see above:
  1. Link reliability,
  2. Packet delay,
  3. Node energy consumption,
  4. Expected transmissions count (ETX)
  5. …



RPL DIO messages

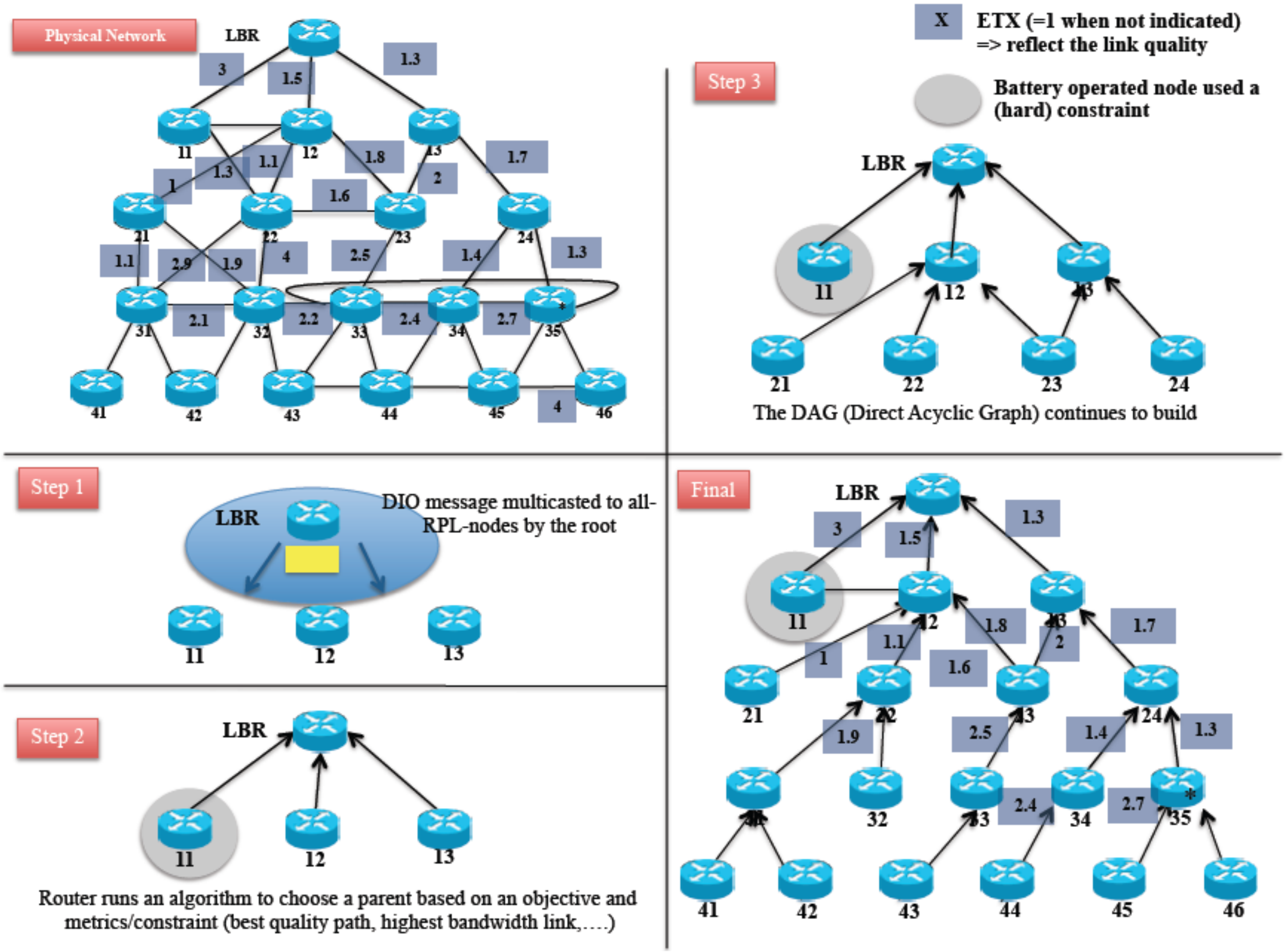
- DODAG minimizes the cost to go to the root (destination node) based on the Objective Function
- DIO messages are broadcast to build the tree; DIO includes
  - a node's rank (its level) d_j
  - packet forwarding metric a_{ij}
- A node selects a parent based on the received DIO message and calculates its rank
- Destination Advertisement Option (DAO) messages are sent periodically to notify parent about routes to children nodes

Example 1



0
DIO
1
DIO
DIO
2
DIO
DIO
1
1
DIO
2
DIO
DIO
DIO
DIO
DIO
2
2
2
DIO
DIO
3
3
3
3

Example 2



Physical Network
LBR
X
ETX (=1 when not indicated)
=> reflect the link quality
Battery operated node used a
(hard) constraint
Step 3
The DAG (Direct Acyclic Graph) continues to build
Step 1
DIO message multicasted to all-
RPL-nodes by the root
Step 2
Router runs an algorithm to choose a parent based on an objective and
metrics/constraint (best quality path, highest bandwidth link,….)
Final

Other standardized protocol stacks

- In the following there is mention of other protocol stacks with standardized routing

- If you have time, read them. There will not be question on them at the exam

ZigBee, www.zigbee.org

- ZigBee covers the networking and application layers on top of IEEE 802.15.4
- Nodes:
  1. IEEE 802.15.4 nodes
  2. ZigBee coordinator: starts the network
  3. ZigBee router
- Networks: star, tree, mesh
- Routing
  1. No transport protocol for end-to-end reliability (only hop-by-hop).
  2. Tree routing: packets are sent to the coordinator, and then to the destination.
  3. Mesh routing: AODV protocol for route discovery

ISA SP-100, www.isa.org

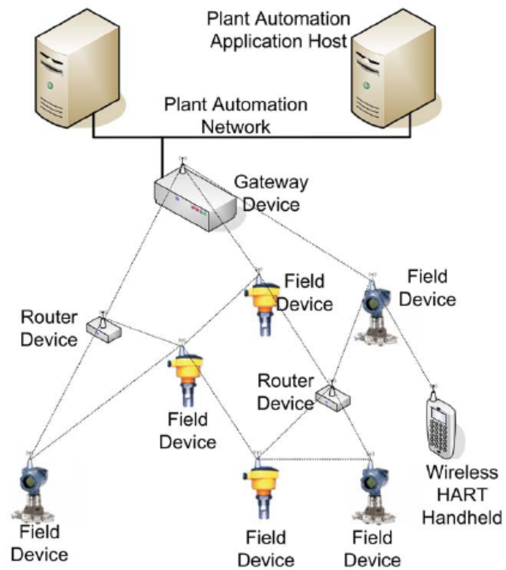
- Standard for non critical process applications tolerating delays up to 100ms.

- 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, www.hartcomm.org

- Released in September 2007 as part of HART 7 specifications
- An open communication standard designed for process measurements and control applications
  1. strict timing requirements
  2. 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:
  1. Timer
  2. Network wide synchronization
  3. Communication security
  4. Reliable mesh networking
  5. Central network management

A Five Layers Architecture

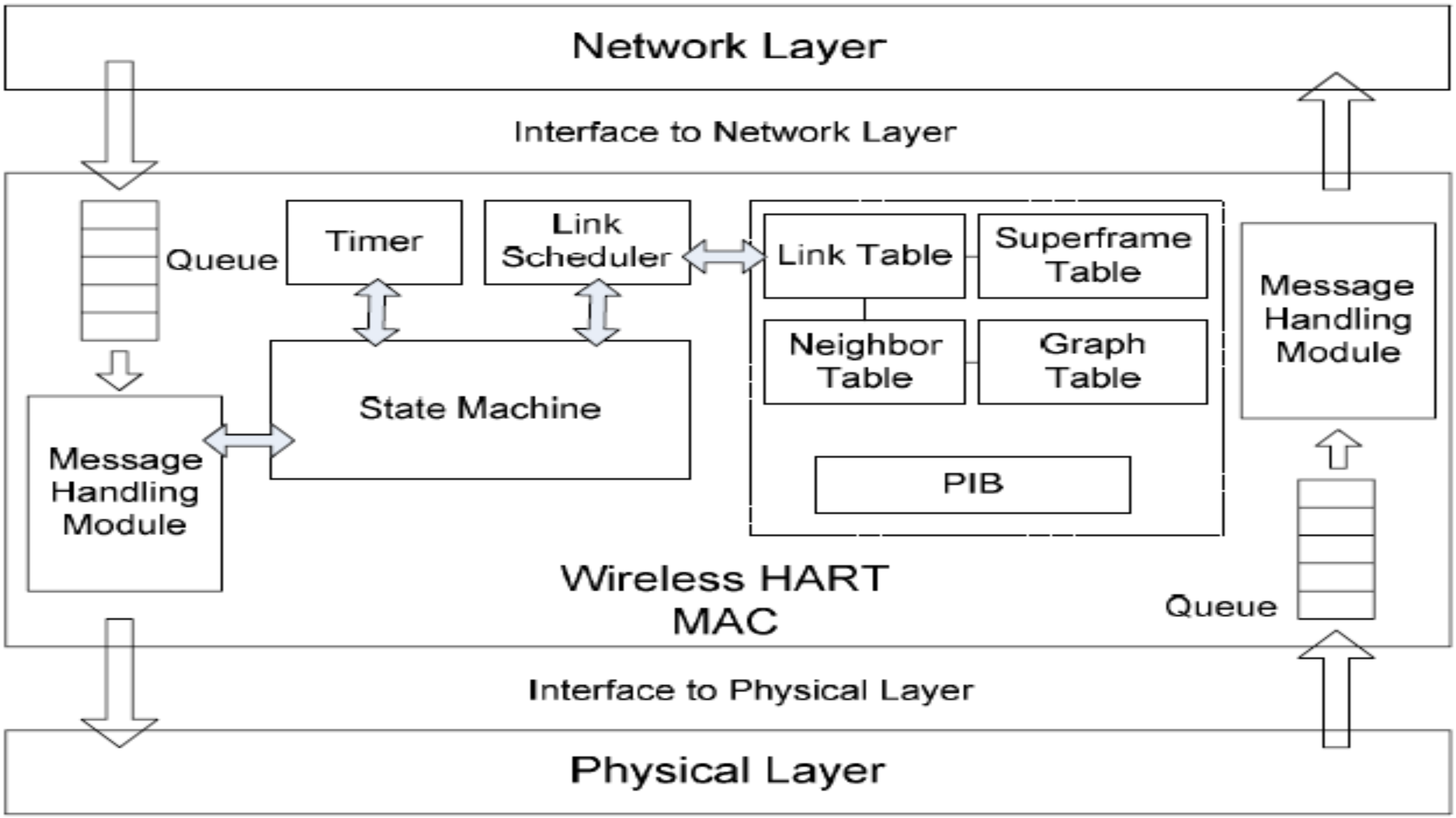
OSI Layer	HART	
Application	Command Oriented. Predefined Data Types and Application Procedures	
Presentation		
Session		
Transport	Auto-Segmented transfer of large data sets, reliable stream transport, Negotiated Segment sizes	
Network		Power-Optimized Redundant Path, Mesh to the edge Network
Data Link	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 Graph Routing

- Graph: collection of paths that connect network nodes
- Graph's paths are created by the network manager and downloaded to each node
- To send a packet, the source node 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

WirelessHART Source Routing

- Source Routing is a supplement of the graph routing aiming at network diagnostics
- To send a packet to destination, the source node 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

Summary

- We have studied routing protocols

- The theoretical foundation of routing is the shortest path optimization problem

  - It gives the basic mechanisms that aim at optimal routing

  - These mechanisms are included in existing WSN standards such as ROLL RPL, ZigBee, ISA100, WirelessHart

Next lecture

- By this lecture, we concluded the networking part of the course

- We move to the signal processing part

  - Distributed detection, i.e., how to reliably detect the happening of events by a WSN