BLM6109– Advanced Computer Networks

Slides taken from
Holger Karl
(Protocols and Architectures for Wireless Sensor Networks)

Outline

- Motivation & Applications
- Single node architecture
- Network Architecture
- Physical layer
- Medium access control protocols
- Link layer protocols

Chapter 1: Motivation & Applications

Holger Karl

Goals of This Chapter

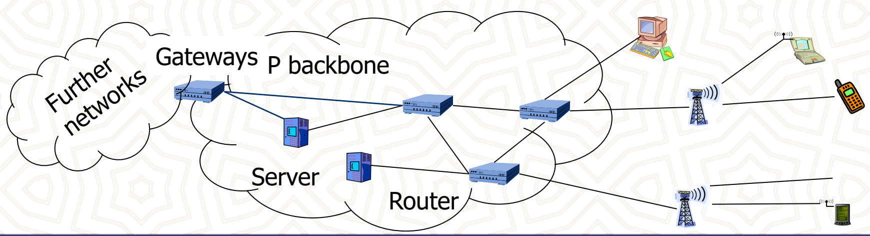
- Give an understanding what ad hoc & sensor networks are good for, what their intended application areas are
- Commonalities and differences
  - Differences to related network types
- Limitations of these concepts

Outline

- **Infrastructure for wireless?**
- (Mobile) ad hoc networks
- Wireless sensor networks
- Comparison

Infrastructure-Based Wireless Networks

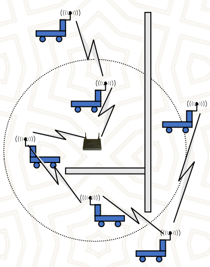
- Typical wireless network: Based on infrastructure
  - E.g., GSM, UMTS, …
  - Base stations connected to a wired backbone network
  - Mobile entities communicate wirelessly to these base stations
  - Traffic between different mobile entities is relayed by base stations and wired backbone
  - Mobility is supported by switching from one base station to another
  - Backbone infrastructure required for administrative tasks

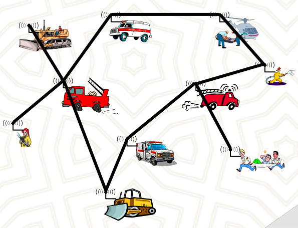


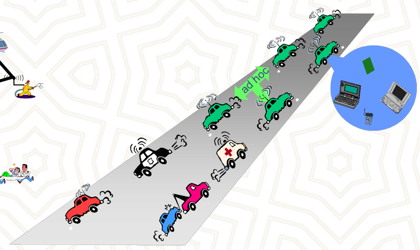
Infrastructure-Based Wireless Networks – Limits?

- What if …
  - No infrastructure is available? – E.g., in disaster areas
  - It is too expensive/inconvenient to set up? – E.g., in remote, large construction sites
  - There is no time to set it up? – E.g., in military operations

Possible Applications for Infrastructure-Free Networks







- Factory floor automation
- Disaster recovery
- Car-to-car communication

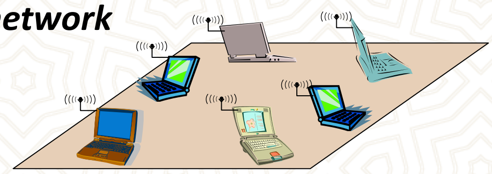
- Military networking: Tanks, soldiers, …
- Finding out empty parking lots in a city, without asking a server
- Search-and-rescue in an avalanche
- Personal area networking (watch, glasses, PDA, medical appliance, …)
- …

Outline

- Infrastructure for wireless?
- **(Mobile) ad hoc networks**
- Wireless sensor networks
- Comparison

Solution: (Wireless) Ad Hoc Networks

- Try to construct a network without infrastructure, using networking abilities of the participants
  - This is an ***ad hoc network*** – a network constructed "for a special purpose"
- Simplest example: Laptops in a conference room – a ***single-hop ad hoc network***



Problems/Challenges for Ad Hoc Networks

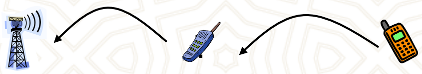
- Without a central infrastructure, things become much more difficult
- Problems are due to
  - Lack of central entity for organization available
  - Limited range of wireless communication
  - Mobility of participants
  - Battery-operated entities

No Central Entity ! Self-Organization

- Without a central entity (like a base station), participants must organize themselves into a network (***self-organization***)
- Pertains to (among others):
  - Medium access control – no base station can assign transmission resources, must be decided in a distributed fashion
  - Finding a route from one participant to another

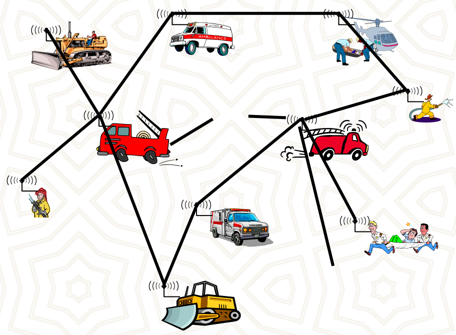
Limited Range ! Multi-Hopping

- For many scenarios, communication with peers outside immediate communication range is required
  - Direct communication limited because of distance, obstacles, …
  - Solution: ***multi-hop network***



Mobility ! Suitable, adaptive protocols

- In many (not all!) ad hoc network applications, participants move around
  - In cellular network: simply hand over to another base station
- In ***mobile ad hoc networks (MANET)***:
  - Mobility changes neighborhood relationship
  - Must be compensated for
  - E.g., routes in the network have to be changed
- Complicated by scale
  - Large number of such nodes difficult to support



Battery-Operated Devices ! Energy-Efficient Operation

- Often (not always!), participants in an ad hoc network draw energy from batteries
- Desirable: long run time for
  - Individual devices
  - Network as a whole

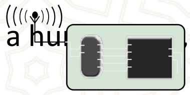
! Energy-efficient networking protocols
  - E.g., use multi-hop routes with low energy consumption (energy/bit)
  - E.g., take available battery capacity of devices into account
  - How to resolve conflicts between different optimizations?

Outline

- Infrastructure for wireless?
- (Mobile) ad hoc networks
- **Wireless sensor networks**
  - **Applications**
  - Requirements & mechanisms
- Comparison

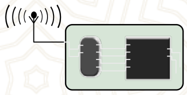
Wireless Sensor Networks

- Participants in the previous examples were devices close to a hu
  interacting with humans



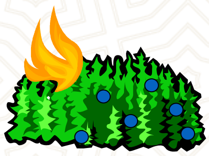
- Alternative concept:

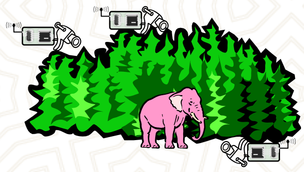
  Instead of focusing interaction on humans, focus on interacting with ***environment***
  - Network is ***embedded*** in environment
  - Nodes in the network are equipped with ***sensing*** and ***actuation*** to measure/influence environment
  - Nodes process information and communicate it wirelessly



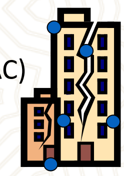
! ***Wireless sensor networks*** (WSN)
  - Or: ***Wireless sensor & actuator networks*** (WSAN)

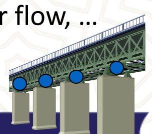
WSN Application Examples





- Disaster relief operations
  - Drop sensor nodes from an aircraft over a wildfire
  - Each node measures temperature
  - Derive a “temperature map”
- Biodiversity mapping
  - Use sensor nodes to observe wildlife
- Intelligent buildings (or bridges)
  - Reduce energy wastage by proper humidity, ventilation, air conditioning (HVAC) control
  - Needs measurements about room occupancy, temperature, air flow, …
  - Monitor mechanical stress after earthquakes





WSN Application Scenarios

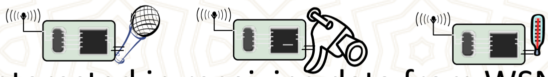
- Facility management
  - Intrusion detection into industrial sites
  - Control of leakages in chemical plants, …
- Machine surveillance and preventive maintenance
  - Embed sensing/control functions into places no cable has gone before
  - E.g., tire pressure monitoring
- Precision agriculture
  - Bring out fertilizer/pesticides/irrigation only where needed
- Medicine and health care
  - Post-operative or intensive care
  - Long-term surveillance of chronically ill patients or the elderly

WSN Application Scenarios

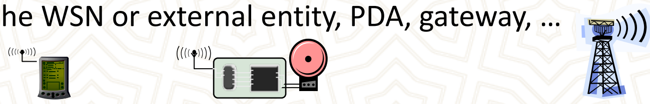
- Logistics
  - Equip goods (parcels, containers) with a sensor node
  - Track their whereabouts – ***total asset management***
  - Note: passive readout might suffice – compare RF IDs
- Telematics
  - Provide better traffic control by obtaining finer-grained information about traffic conditions
  - ***Intelligent roadside***
  - Cars as the sensor nodes

Roles of Participants in WSN

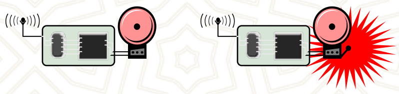
- ***Sources*** of data: Measure data, report them "somewhere"
  - Typically equip with different kinds of actual sensors



- ***Sinks*** of data: Interested in receiving data from WSN
  - May be part of the WSN or external entity, PDA, gateway, …



- ***Actuators***: Control some device based on data, usually also a sink



Structuring WSN Application Types

- ***Interaction patterns*** between sources and sinks classify application types
  - ***Event detection***: Nodes locally detect events (maybe jointly with nearby neighbors), report these events to interested sinks
    - ***Event classification*** additional option
  - ***Periodic measurement***
  - ***Function approximation:*** Use sensor network to approximate a function of space and/or time (e.g., temperature map)
  - ***Edge detection:*** Find edges (or other structures) in such a function (e.g., where is the zero degree border line?)
  - ***Tracking***: Report (or at least, know) position of an observed intruder ("pink elephant")

Deployment Options for WSN

- How are sensor nodes deployed in their environment?
  - Dropped from aircraft ! ***Random deployment***
    - Usually uniform random distribution for nodes over finite area is assumed
    - Is that a likely proposition?
  - Well planned, fixed ! ***Regular deployment***
    - E.g., in preventive maintenance or similar
    - Not necessarily geometric structure, but that is often a convenient assumption
  - ***Mobile*** sensor nodes
    - Can move to compensate for deployment shortcomings
    - Can be passively moved around by some external force (wind, water)
    - Can actively seek out “interesting” areas

Maintenance Options

- Feasible and/or practical to maintain sensor nodes?
  - E.g., to replace batteries?
  - Or: unattended operation?
  - Impossible but not relevant? Mission lifetime might be very small

- Energy supply?
  - Limited from point of deployment?
  - Some form of recharging, energy scavenging from environment?
    - E.g., solar cells

Outline

- Infrastructure for wireless?
- (Mobile) ad hoc networks
- **Wireless sensor networks**
  - Applications
  - **Requirements & mechanisms**
- Comparison

Characteristic Requirements for WSNs

- Type of service of WSN
  - Not simply moving bits like another network
  - Rather: provide ***answers*** (not just numbers)
  - Issues like geographic scoping are natural requirements, absent from other networks
- Quality of service
  - Traditional QoS metrics do not apply
  - Still, service of WSN must be "good": Right answers at the right time
- Fault tolerance
  - Be robust against node failure (running out of energy, physical destruction, …)
- Lifetime
  - The ***network*** should fulfill its task as long as possible – definition depends on application
  - Lifetime of individual nodes relatively unimportant
  - But often treated equivalently

Characteristic Requirements for WSNs

- Scalability
  - Support large number of nodes
- Wide range of densities
  - Vast or small number of nodes per unit area, very application-dependent
- Programmability
  - Re-programming of nodes in the field might be necessary, improve flexibility
- Maintainability
  - WSN has to adapt to changes, self-monitoring, adapt operation
  - Incorporate possible additional resources, e.g., newly deployed nodes

Required Mechanisms to Meet Requirements

- Multi-hop wireless communication
- Energy-efficient operation
  - Both for communication and computation, sensing, actuating
- Auto-configuration
  - Manual configuration just not an option
- Collaboration & in-network processing
  - Nodes in the network collaborate towards a joint goal
  - Pre-processing data in network (as opposed to at the edge) can greatly improve efficiency

Required Mechanisms to Meet Requirements

- Data centric networking
  - Focusing network design on ***data***, not on ***node identifies*** (id-centric networking)
  - To improve efficiency
- Locality
  - Do things locally (on node or among nearby neighbors) as far as possible
- Exploit tradeoffs
  - E.g., between invested energy and accuracy

Outline

- Infrastructure for wireless?
- (Mobile) ad hoc networks
- Wireless sensor networks
- **Comparison**

MANET vs. WSN

- Many commonalities: Self-organization, energy efficiency, (often) wireless multi-hop
- Many differences
  - ***Applications, equipment***: MANETs more powerful (read: expensive) equipment assumed, often "human in the loop"-type applications, higher data rates, more resources
  - ***Application-specific***: WSNs depend much stronger on application specifics; MANETs comparably uniform
  - ***Environment interaction***: core of WSN, absent in MANET
  - ***Scale***: WSN might be much larger (although contestable)
  - ***Energy***: WSN tighter requirements, maintenance issues
  - ***Dependability/QoS***: in WSN, individual node may be dispensable (network matters), QoS different because of different applications
  - ***Data centric*** vs. id-centric networking
  - ***Mobility***: different mobility patterns like (in WSN, sinks might be mobile, usual nodes static)

Wireless fieldbuses and WSNs

- Fieldbus:
  - Network type invented for real-time communication, e.g., for factory-floor automation
  - Inherent notion of sensing/measuring and controlling
  - Wireless fieldbus: Real-time communication over wireless

! Big similarities

- Differences
  - Scale – WSN often intended for larger scale
  - Real-time – WSN usually not intended to provide (hard) real-time guarantees as attempted by fieldbuses

Enabling Technologies for WSN

- Cost reduction
  - For wireless communication, simple microcontroller, sensing, batteries
- Miniaturization
  - Some applications demand small size
  - “Smart dust” as the most extreme vision
- Energy scavenging
  - Recharge batteries from ambient energy (light, vibration, …)

Conclusion

- MANETs and WSNs are challenging and promising system concepts
- Many similarities, many differences
- Both require new types of architectures & protocols compared to “traditional” wired/wireless networks
- In particular, application-specificness is a new issue

Chapter 2: Single node architecture

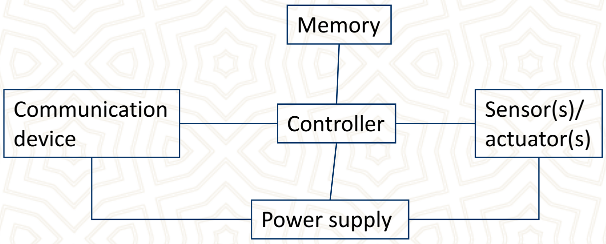
Holger Karl

Outline

- **Sensor node architecture**
- Energy supply and consumption
- Runtime environments for sensor nodes
- Case study: TinyOS

Sensor Node Architecture

- Main components of a WSN node
  - Controller
  - Communication device(s)
  - Sensors/actuators
  - Memory
  - Power supply



Controller

- Main options:
  - Microcontroller – general purpose processor, optimized for embedded applications, low power consumption
  - DSPs – optimized for signal processing tasks, not suitable here
  - FPGAs – may be good for testing
  - ASICs – only when peak performance is needed, no flexibility
- Example microcontrollers
  - Texas Instruments MSP430
    - 16-bit RISC core, up to 4 MHz, versions with 2-10 kbytes RAM, several DACs, RT clock, prices start at 0.49 US$
  - Atmel ATMega
    - 8-bit controller, larger memory than MSP430, slower

Communication device

- Which transmission medium?
  - Electromagnetic at radio frequencies? ✓
  - Electromagnetic, light?
  - Ultrasound?
- Radio transceivers transmit a bit- or byte stream as radio wave
  - Receive it, convert it back into bit-/byte stream

Transceiver Characteristics

- Capabilities
  - Interface: bit, byte, packet level?
  - Supported frequency range?
    - Typically, somewhere in 433 MHz – 2.4 GHz, ISM band
  - Multiple channels?
  - Data rates?
  - Range?
- Energy characteristics
  - Power consumption to send/receive data?
  - Time and energy consumption to change between different states?
  - Transmission power control?
  - Power efficiency (which percentage of consumed power is radiated?)
- Radio performance
  - Modulation? (ASK, FSK, …?)
  - Noise figure? NF = SNR_I/SNR_O
  - Gain? (signal amplification)
  - Receiver sensitivity? (minimum S to achieve a given E_b/N_0)
  - Blocking performance (achieved BER in presence of frequency-offset interferer)
  - Out of band emissions
  - Carrier sensing & RSSI characteristics
  - Frequency stability (e.g., towards temperature changes)
  - Voltage range

Transceiver States

- Transceivers can be put into different operational ***states***, typically:
  - ***Transmit***
  - ***Receive***
  - ***Idle*** – ready to receive, but not doing so
    - Some functions in hardware can be switched off, reducing energy consumption a little
  - ***Sleep*** – significant parts of the transceiver are switched off
    - Not able to immediately receive something
    - ***Recovery time*** and ***startup energy*** to leave sleep state can be significant
- Research issue: Wakeup receivers – can be woken via radio when in sleep state (seeming contradiction!)

Wakeup Receivers

- Major energy problem: ***RECEIVING***
  - Idling and being ready to receive consumes considerable amounts of power
- When to switch on a receiver is not clear
  - Contention-based MAC protocols: Receiver is always on
  - TDMA-based MAC protocols: Synchronization overhead, inflexible
- Desirable: Receiver that can (only) check for incoming messages
  - When signal detected, wake up main receiver for actual reception
  - Ideally: ***Wakeup receiver*** can already process simple addresses
  - Not clear whether they can be actually built, however

Sensors as such

- Main categories
  - Any energy radiated? Passive vs. active sensors
  - Sense of direction? Omidirectional?

  - Passive, omnidirectional
    - Examples: light, thermometer, microphones, hygrometer, …
  - Passive, narrow-beam
    - Example: Camera
  - Active sensors
    - Example: Radar
- Important parameter: Area of coverage
  - Which region is adequately covered by a given sensor?

Outline

- Sensor node architecture
- **Energy supply and consumption**
- Runtime environments for sensor nodes
- Case study: TinyOS

Energy Supply of Mobile/Sensor Nodes

- Goal: provide as much energy as possible at smallest cost/volume/weight/recharge time/longevity
  - In WSN, recharging may or may not be an option
- Options
  - Primary batteries – not rechargeable
  - Secondary batteries – rechargeable, only makes sense in combination with some form of energy harvesting
- Requirements include
  - Low self-discharge
  - Long shelf live
  - Capacity under load
  - Efficient recharging at low current
  - Good relaxation properties (seeming self-recharging)
  - Voltage stability (to avoid DC-DC conversion)

Energy Scavenging

- How to recharge a battery?
  - A laptop: easy, plug into wall socket in the evening
  - A sensor node? – Try to ***scavenge*** energy from environment
- Ambient energy sources
  - Light ! solar cells – between 10 μW/cm^2 and 15 mW/cm^2
  - Temperature gradients – 80 μ W/cm^2 @ 1 V from 5K difference
  - Vibrations – between 0.1 and 10000 μ W/cm^3
  - Pressure variation (piezo-electric) – 330 μ W/cm^2 from the heel of a shoe
  - Air/liquid flow
    (MEMS gas turbines)

Energy scavenging – overview

Energy source	Energy density
Batteries (zinc-air)	$1050 - 1560\,\mathrm{mWh/cm^3}$
Batteries (rechargable lithium)	$300\,\mathrm{mWh/cm^3}$ (at $3 - 4\,\mathrm{V}$)

Energy source	Power density
Solar (outdoors)	$15\,\mathrm{mW/cm^2}$ (direct sun)
	$0.15\,\mathrm{mW/cm^2}$ (cloudy day)
Solar (indoors)	$0.006\,\mathrm{mW/cm^2}$ (standard office desk)
	$0.57\,\mathrm{mW/cm^2}$ ($< 60\,\mathrm{W}$ desk lamp)
Vibrations	$0.01 - 0.1\,\mathrm{mW/cm^3}$
Acoustic noise	$3 \cdot 10^{-6}\mathrm{mW/cm^2}$ at $75\,\mathrm{Db}$
	$9,6 \cdot 10^{-4}\mathrm{mW/cm^2}$ at $100\,\mathrm{Db}$
Passive human-powered systems	$1.8\,\mathrm{mW}$ (shoe inserts)
Nuclear reaction	$80\,\mathrm{mW/cm^3}$, $10^6\,\mathrm{mWh/cm^3}$

Energy consumption

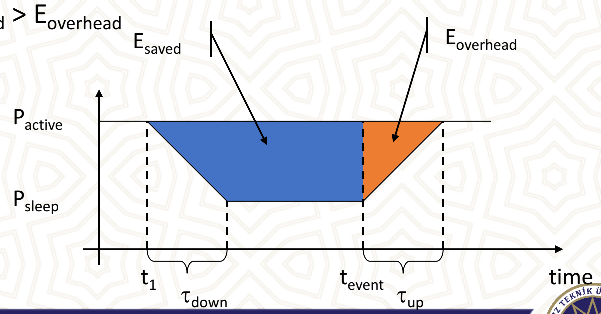
- A “back of the envelope” estimation
- Number of instructions
  - Energy per instruction: 1 nJ
  - Small battery (“smart dust”): 1 J = 1 Ws
  - Corresponds: 10^9 instructions!
- Lifetime
  - Or: Require a single day operational lifetime = 24¢60¢60 =86400 s
  - 1 Ws / 86400s ¼ ***11.5 µW*** as max. sustained power consumption!
- Not feasible!

Multiple Power Consumption Modes

- Way out: Do not run sensor node at full operation all the time
  - If nothing to do, switch to ***power safe mode***
  - Question: When to throttle down? How to wake up again?
- Typical modes
  - Controller: Active, idle, sleep
  - Radio mode: Turn on/off transmitter/receiver, both
- Multiple modes possible, "deeper" sleep modes
  - Strongly depends on hardware
  - TI MSP 430, e.g.: four different sleep modes
  - Atmel ATMega: six different modes

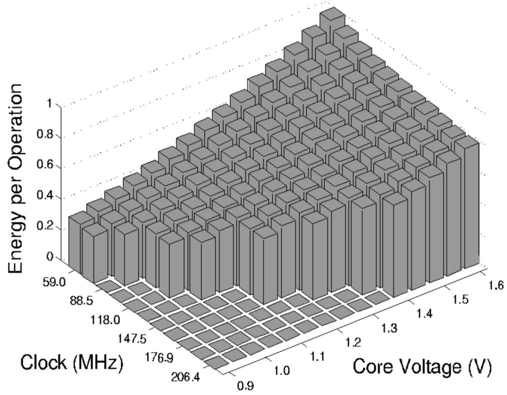
Switching Between Modes

- Simplest idea: Greedily switch to lower mode whenever possible
- Problem: Time and power consumption required to reach higher modes not negligible
  - Introduces overhead
  - Switching only pays off if $E_{saved} > E_{overhead}$
- Example: Event-triggered wake up from sleep mode
- Scheduling problem with uncertainty (exercise)



Alternative: Dynamic Voltage Scaling

- Switching modes complicated by uncertainty how long a sleep time is available
- Alternative: Low supply voltage & clock
  - ***Dynamic voltage scaling*** (DVS)
- Rationale:
  - Power consumption P depends on
    - Clock frequency
    - Square of supply voltage
    - *P / f V*²
  - Lower clock allows lower supply voltage
  - Easy to switch to higher clock
  - But: execution takes longer



Memory Power Consumption

- Crucial part: FLASH memory
  - Power for RAM almost negligible
- FLASH writing/erasing is expensive
  - Example: FLASH on Mica motes
  - Reading: ¼ 1.1 nAh per byte
  - Writing: ¼ 83.3 nAh per byte

Transmitter Power/Energy Consumption for n bits

- Amplifier power: $P_{amp} = \alpha_{amp} + \beta_{amp} P_{tx}$
  - P_{tx} ***radiated power***
  - α_{amp}, β_{amp} constants depending on model
  - Highest efficiency ($\eta = P_{tx} / P_{amp}$) at maximum output power
- In addition: transmitter electronics needs power P_{txElec}
- Time to transmit n bits: $n / (R ¢ R_{code})$
  - R nomial data rate, R_{code} coding rate
- To leave sleep mode
  - Time T_{start}, average power P_{start}

! $E_{tx} = T_{start} P_{start} + n / (R ¢ R_{code}) (P_{txElec} + \alpha_{amp} + \beta_{amp} P_{tx})$

- Simplification: Modulation not considered

Yıldız Teknik Üniversitesi - Bilgisayar Mühendisliği Bölümü

53

Receiver Power/Energy Consumption for n bits

- Receiver also has startup costs
  - Time T_{start}, average power P_{start}
- Time for n bits is the same $n / (R ¢ R_{code})$
- Receiver electronics needs P_{rxElec}
- Plus: energy to decode n bits $E_{decBits}$

! $E_{rx} = T_{start} P_{start} + n / (R ¢ R_{code}) P_{rxElec} + E_{decBits} (R)$

Yıldız Teknik Üniversitesi - Bilgisayar Mühendisliği Bölümü

54

Controlling Transceivers

- Similar to controller, low duty cycle is necessary
  - Easy to do for transmitter – similar problem to controller: when is it worthwhile to switch off
  - Difficult for receiver: Not only time when to wake up not known, it also depends on ***remote*** partners

  ! Dependence between MAC protocols and power consumption is strong!
- Only limited applicability of techniques analogue to DVS
  - Dynamic Modulation Scaling (DSM): Switch to modulation best suited to communication – depends on channel gain
  - Dynamic Coding Scaling – vary coding rate according to channel gain
  - Combinations

Computation vs. Communication Energy Cost

- Tradeoff?
  - Directly comparing computation/communication energy cost not possible
  - But: put them into perspective!
  - Energy ratio of "sending one bit" vs. "computing one instruction": Anything between 220 and 2900 in the literature
  - To communicate (send & receive) one kilobyte
    = computing three million instructions!
- Hence: try to compute instead of communicate whenever possible
- Key technique in WSN – ***in-network processing!***
  - Exploit compression schemes, intelligent coding schemes, …

Outline

- Sensor node architecture
- Energy supply and consumption
- ***Runtime environments for sensor nodes***
- Case study: TinyOS

Operating System Challenges in WSN

- Usual operating system goals
  - Make access to device resources abstract (virtualization)
  - Protect resources from concurrent access
- Usual means
  - Protected operation modes of the CPU – hardware access only in these modes
  - Process with separate address spaces
  - Support by a memory management unit
- Problem: These are not available in microcontrollers
  - No separate protection modes, no memory management unit
  - Would make devices more expensive, more power-hungry

! ???

Operating System Challenges in WSN

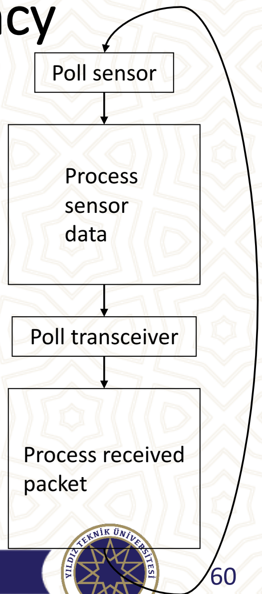
- Possible options
  - Try to implement "as close to an operating system" on WSN nodes
    - In particular, try to provide a known programming interface
    - Namely: support for processes!
    - Sacrifice protection of different processes from each other

    ! Possible, but relatively high overhead
  - Do (more or less) away with operating system
    - After all, there is only a single "application" running on a WSN node
    - No need to protect malicious software parts from each other
    - Direct hardware control by application might improve efficiency
- Currently popular verdict: no OS, just a simple run-time environment
  - Enough to abstract away hardware access details
  - Biggest impact: Unusual programming model

Main Issue: How to Support Concurrency

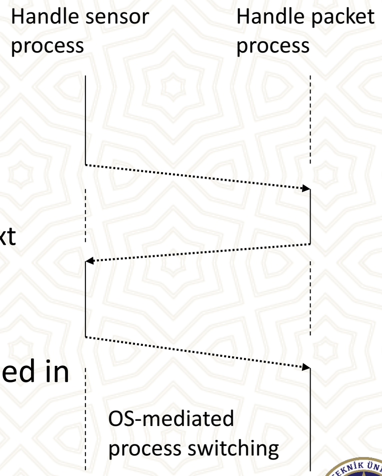
- Simplest option: No concurrency, sequential processing of tasks
  - Not satisfactory: Risk of missing data (e.g., from transceiver) when processing data, etc.

  ! Interrupts/asynchronous operation has to be supported
- Why concurrency is needed
  - Sensor node's CPU has to service the radio modem, the actual sensors, perform computation for application, execute communication protocol software, etc.



Traditional Concurrency: Processes

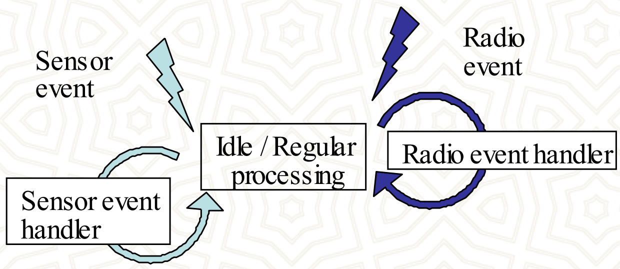
- Traditional OS: processes/threads
  - Based on interrupts, context switching
  - But: not available – memory overhead, execution overhead
- But: concurrency mismatch
  - One process per protocol entails too many context switches
  - Many tasks in WSN small with respect to context switching overhead
- And: protection between processes not needed in WSN
  - Only one application anyway



Event-Based Concurrency

- Alternative: Switch to ***event-based programming model***
  - Perform regular processing or be idle
  - React to events when they happen immediately
  - Basically: interrupt handler
- Problem: must not remain in interrupt handler too long
  - Danger of loosing events
  - Only save data, post information that event has happened, then return

  ! ***Run-to-completion*** principle
  - Two contexts: one for handlers, one for regular execution



Components Instead of Processes

- Need an abstraction to group functionality
  - Replacing "processes" for this purpose
  - E.g.: individual functions of a networking protocol
- One option: ***Components***
  - Here: In the sense of TinyOS
  - Typically fulfill only a single, well-defined function
  - Main difference to processes:
    - Component does not have an execution
    - Components access same address space, no protection against each other
  - NOT to be confused with component-based programming!

API to an Event-Based Protocol Stack

- Usual networking API: sockets
  - Issue: blocking calls to receive data
  - Ill-matched to event-based OS
  - Also: networking semantics in WSNs not necessarily well matched to/by socket semantics
- API is therefore also event-based
  - E.g.: Tell some component that some other component wants to be informed if and when data has arrived
  - Component will be posted an event once this condition is met
  - Details: see TinyOS example discussion below

Dynamic Power Management

- Exploiting multiple operation modes is promising
- Question: When to switch in power-safe mode?
  - Problem: Time & energy overhead associated with wakeup; greedy sleeping is not beneficial (see exercise)
  - Scheduling approach
- Question: How to control dynamic voltage scaling?
  - More aggressive; stepping up voltage/frequency is easier
  - Deadlines usually bound the required speed form below
- Or: Trading off fidelity vs. energy consumption!
  - If more energy is available, compute more accurate results
  - Example: Polynomial approximation
    - Start from high or low exponents depending where the polynomial is to be evaluated

Chapter 3: Network Architecture

Goals of this chapter

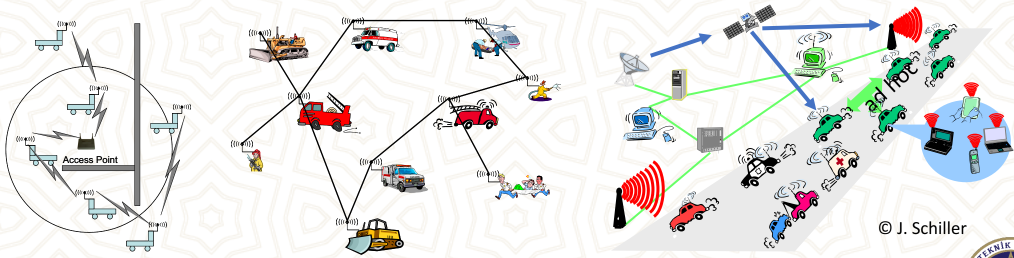
- Having looked at the individual nodes in the previous chapter, we look at general principles and architectures how to put these nodes together to form a meaningful network
- We will look at design approaches to both the more conventional ad hoc networks and the non-standard WSNs

Outline

- **Network scenarios**
- Optimization goals
- Design principles
- Service interface
- Gateway concepts

Basic Scenarios: Ad Hoc Networks

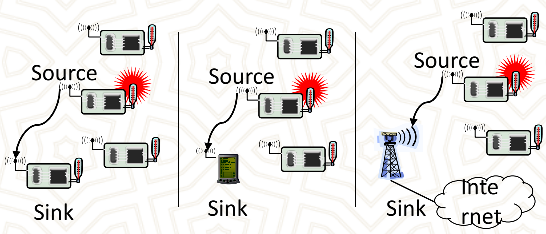
- (Mobile) ad hoc scenarios
  - Nodes talking to each other
  - Nodes talking to “some” node in another network (Web server on the Internet, e.g.)
    - Typically requires some connection to the fixed network
  - Applications: Traditional data (http, ftp, collaborative apps, …) & multimedia (voice, video) ! humans in the loop



© J. Schiller

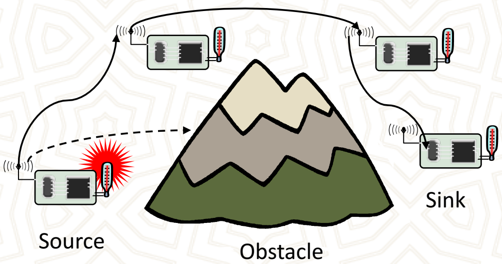
Basic Scenarios: Sensor Networks

- Sensor network scenarios
  - ***Sources***: Any entity that provides data/measurements
  - ***Sinks***: Nodes where information is required
    - Belongs to the sensor network as such
    - Is an external entity, e.g., a PDA, but directly connected to the WSN
      - Main difference: comes and goes, often moves around, …
    - Is part of an external network (e.g., internet), somehow connected to the WSN
  - Applications: Usually, machine to machine, often limited amounts of data, different notions of importance



Single-Hop vs. Multi-Hop Networks

- One common problem: limited range of wireless communication
  - Essentially due to limited transmission power, path loss, obstacles
- Option: multi-hop networks
  - Send packets to an intermediate node
  - Intermediate node forwards packet to its destination
  - ***Store-and-forward*** multi-hop network
- Basic technique applies to both WSN and MANET
- Note: Store&forward multi-hopping NOT the only possible solution
  - E.g., collaborative networking, network coding
  - Do not operate on a per-packet basis

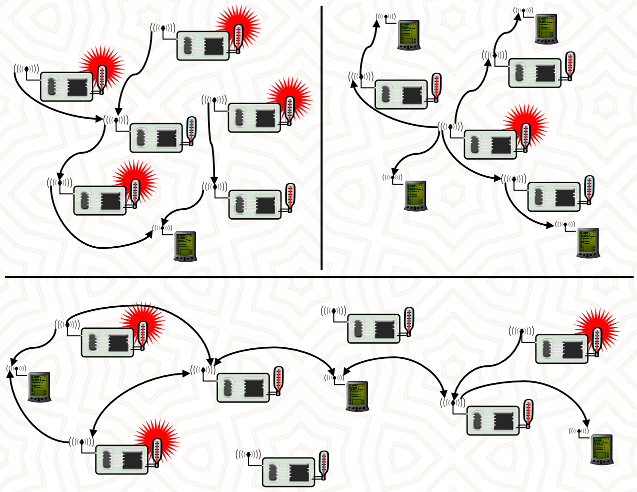


Energy Efficiency of Multi-Hopping?

- Obvious idea: Multi-hopping is more energy-efficient than direct communication
  - Because of path loss $\alpha > 2$, energy for distance d is reduced from cd^α to $2c(d/2)^\alpha$
    - c some constant
- However: This is usually wrong, or at least very over-simplified
  - Need to take constant offsets for powering transmitter, receiver into account
  - Details see exercise, chapter 2

! Multi-hopping for energy savings needs careful choice

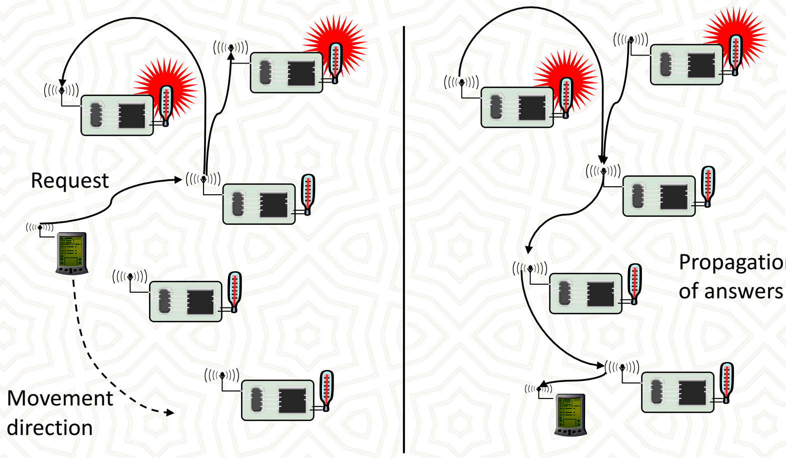
WSN: Multiple Sinks, Multiple Sources



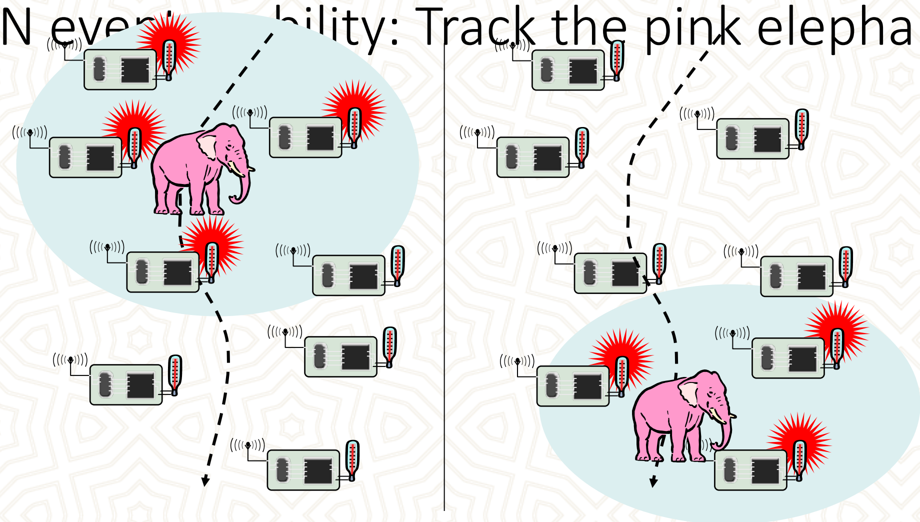
Different Sources of Mobility

- Node mobility
    - A node participating as source/sink (or destination) or a relay node might move around
    - Deliberately, self-propelled or by external force; targeted or at random
    - Happens in both WSN and MANET
- Sink mobility
    - In WSN, a sink that is not part of the WSN might move
    - Mobile requester
- Event mobility
    - In WSN, event that is to be observed moves around (or extends, shrinks)
    - Different WSN nodes become "responsible" for surveillance of such an event

WSN Sink Mobility



WSN event mobility: Track the pink elephant



Here: Frisbee model as example

Outline

- Network scenarios
- **Optimization goals**
- Design principles
- Service interface
- Gateway concepts

Optimization Goal: Quality of Service

- In MANET: Usual QoS interpretation
  - Throughput/delay/jitter
  - High perceived QoS for multimedia applications
- In WSN, more complicated
  - Event detection/reporting probability
  - Event classification error, detection delay
  - Probability of missing a periodic report
  - Approximation accuracy (e.g, when WSN constructs a temperature map)
  - Tracking accuracy (e.g., difference between true and conjectured position of the pink elephant)
- Related goal: robustness
  - Network should withstand failure of some nodes

Optimization Goal: Energy efficiency

- Umbrella term!
- Energy per correctly received bit
  - Counting all the overheads, in intermediate nodes, etc.
- Energy per reported (unique) event
  - After all, information is important, not payload bits!
  - Typical for WSN
- Delay/energy tradeoffs
- Network lifetime
  - Time to first node failure
  - Network half-life (how long until 50% of the nodes died?)
  - Time to partition
  - Time to loss of coverage
  - Time to failure of first event notification

Optimization Goal: Scalability

- Network should be operational regardless of number of nodes
  - At high efficiency
- Typical node numbers difficult to guess
  - MANETs: 10s to 100s
  - WSNs: 10s to 1000s, maybe more (although few people have seen such a network before…)
- Requiring to scale to large node numbers has ***serious*** consequences for network architecture
  - Might not result in the most efficient solutions for small networks!
  - Carefully consider actual application needs before looking for n ! 1 solutions!

Outline

- Network scenarios
- Optimization goals
- **Design principles**
- Service interface
- Gateway concepts

Distributed Organization

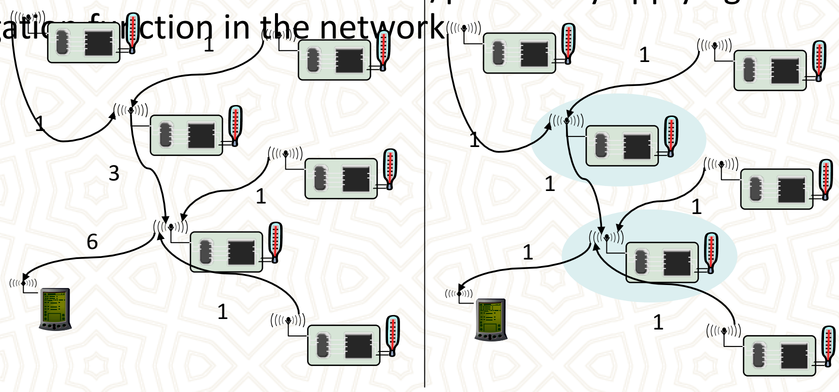
- Participants in a MANET/WSN should cooperate in organizing the network
  - E.g., with respect to medium access, routing, …
  - Centralistic approach as alternative usually not feasible – hinders scalability, robustness
- Potential shortcomings
  - Not clear whether distributed or centralistic organization achieves better energy efficiency (when taking all overheads into account)
- Option: "limited centralized" solution
  - Elect nodes for local coordination/control
  - Perhaps rotate this function over time

In-Network Processing

- MANETs are supposed to deliver bits from one end to the other
- WSNs, on the other end, are expected to provide information, not necessarily original bits
  - Gives addition options
  - E.g., ***manipulate*** or ***process*** the data in the network
- Main example: aggregation
  - Apply composable aggregation functions to a convergecast tree in a network
  - Typical functions: minimum, maximum, average, sum, …
  - Not amenable functions: median

In-Network Processing: Aggregation example

- Reduce number of transmitted bits/packets by applying an aggregation function in the network



In-Network Processing: Signal Processing

- Depending on application, more sophisticated processing of data can take place within the network
  - Example edge detection: locally exchange raw data with neighboring nodes, compute edges, only communicate edge description to far away data sinks
  - Example tracking/angle detection of signal source: Conceive of sensor nodes as a distributed microphone array, use it to compute the angle of a single source, only communicate this angle, not all the raw data
- Exploit ***temporal*** and ***spatial correlation***
  - Observed signals might vary only slowly in time ! no need to transmit all data at full rate all the time
  - Signals of neighboring nodes are often quite similar ! only try to transmit differences (details a bit complicated, see later)

Adaptive Fidelity

- Adapt the effort with which data is exchanged to the currently required accuracy/fidelity
- Example event detection
  - When there is no event, only very rarely send short "all is well" messages
  - When event occurs, increase rate of message exchanges
- Example temperature
  - When temperature is in acceptable range, only send temperature values at low resolution
  - When temperature becomes high, increase resolution and thus message length

Data Centric Networking

- In typical networks (including ad hoc networks), network transactions are addressed to the ***identities*** of specific nodes
  - A "node-centric" or "address-centric" networking paradigm
- In a redundantly deployed sensor networks, specific source of an event, alarm, etc. might not be important
  - Redundancy: e.g., several nodes can observe the same area
- Thus: focus networking transactions on the data directly instead of their senders and transmitters ! ***data-centric networking***
  - Principal design change

Implementation Options for Data-Centric Networking

- Overlay networks & distributed hash tables (DHT)
  - Hash table: content-addressable memory
  - Retrieve data from an unknown source, like in peer-to-peer networking – with efficient implementation
  - Some disparities remain
    - Static key in DHT, dynamic changes in WSN
    - DHTs typically ignore issues like hop count or distance between nodes when performing a lookup operation
- Publish/subscribe
  - Different interaction paradigm
  - Nodes can ***publish*** data, can ***subscribe*** to any particular kind of data
  - Once data of a certain type has been published, it is delivered to all subscribes
  - Subscription and publication are decoupled in time; subscriber and published are agnostic of each other (decoupled in identity)
- Databases

Further design principles

- Exploit location information
  - Required anyways for many applications; can considerably increase performance
- Exploit activity patterns
- Exploit heterogeneity
  - By construction: nodes of different types in the network
  - By evolution: some nodes had to perform more tasks and have less energy left; some nodes received more solar energy than others; …
- Cross-layer optimization of protocol stacks for WSN
  - Goes against grain of standard networking; but promises big performance gains
  - Also applicable to other networks like ad hoc; usually at least worthwhile to consider for most wireless networks

Outline

- Network scenarios
- Optimization goals
- Design principles
- **Service interface**
- Gateway concepts

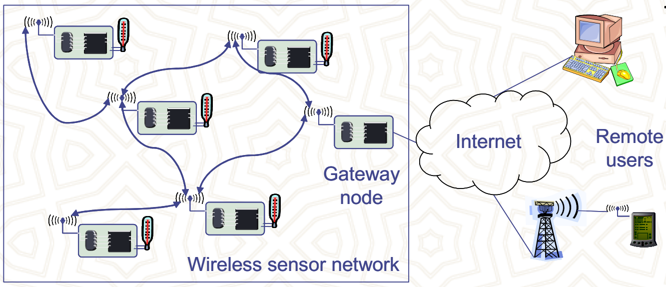
Interfaces to protocol stacks

- The world's all-purpose network interface: sockets
  - Good for transmitting data from one sender to one receiver
  - Not well matched to WSN needs (ok for ad hoc networks)
- Expressibility requirements
  - Support for simple request/response interactions
  - Support for asynchronous event notification
  - Different ways for identifying addressee of data
    - By location, by observed values, implicitly by some other form of group membership
    - By some semantically meaningful form – "room 123"
  - Easy accessibility of in-network processing functions
    - Formulate complex events – events defined only by several nodes
  - Allow to specify accuracy & timeliness requirements
  - Access node/network status information (e.g., battery level)
  - Security, management functionality, …
- No clear standard has emerged yet – many competing, unclear proposals

Outline

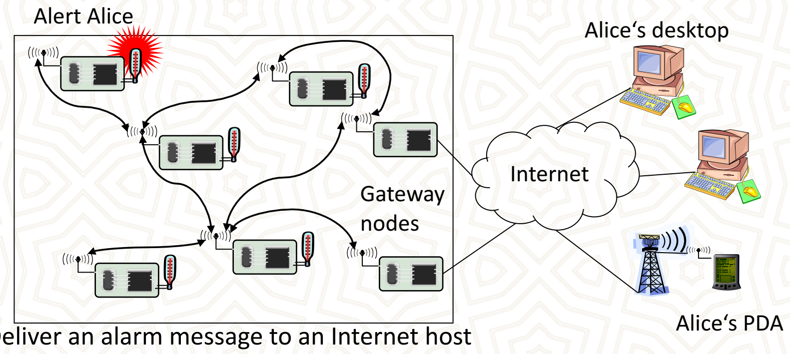
- Network scenarios
- Optimization goals
- Design principles
- Service interface
- **Gateway concepts**

Gateway concepts for WSN/MANET



- Gateways are necessary to the Internet for remote access to/from the WSN
  - Same is true for ad hoc networks; additional complications due to mobility (change route to the gateway; use different gateways)
  - WSN: Additionally bridge the gap between different interaction semantics (data vs. address-centric networking) in the gateway
- Gateway needs support for different radios/protocols, …

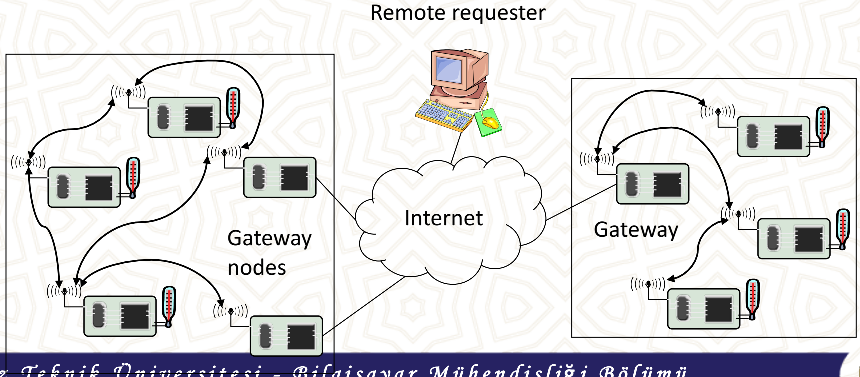
WSN to Internet Communication



- Example: Deliver an alarm message to an Internet host
- Issues
  - Need to find a gateway (integrates routing & service discovery)
  - Choose "best" gateway if several are available
  - How to find Alice or Alice's IP?

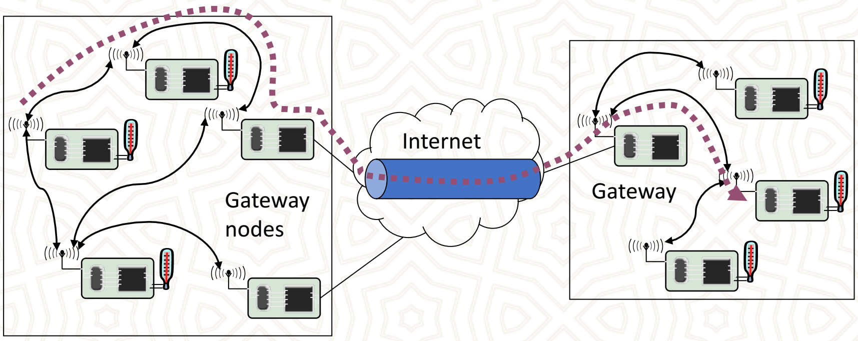
Internet to WSN communication

- How to find the right WSN to answer a need?
- How to translate from IP protocols to WSN protocols, semantics?



WSN tunneling

- Use the Internet to “tunnel” WSN packets between two remote WSNs



Summary

- Network architectures for ad hoc networks are – in principle – relatively straightforward and similar to standard networks
  - Mobility is compensated for by appropriate protocols, but interaction paradigms don't change too much
- WSNs, on the other hand, look quite different on many levels
  - Data-centric paradigm, the need and the possibility to manipulate data as it travels through the network opens new possibilities for protocol design
- The following chapters will look at how these ideas are realized by actual protocols

Chapter 4: Physical layer

Goals of This Chapter

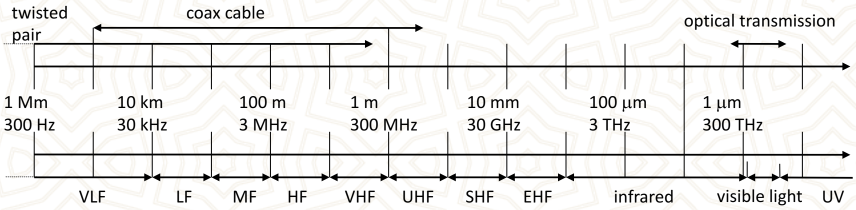
- Get an understanding of the peculiarities of wireless communication
  - "Wireless channel" as abstraction of these properties – e.g., bit error patterns
  - Focus is on radio communication
- Impact of different factors on communication performance
  - Frequency band, transmission power, modulation scheme, etc.
  - Some brief remarks on transceiver design
- Understanding of energy consumption for radio communication
- Here, differences between ad hoc and sensor networks mostly in the required performance
  - Larger bandwidth/sophisticated modulation for higher data rate/range

Overview

- **Frequency bands**
- Modulation
- Signal distortion – wireless channels
- From waves to bits
- Channel models
- Transceiver design

Radio Spectrum for Communication

- Which part of the electromagnetic spectrum is used for communication
  - Not all frequencies are equally suitable for all tasks – e.g., wall penetration, different atmospheric attenuation (oxygen resonances, …)



- VLF = Very Low Frequency
- LF = Low Frequency
- MF = Medium Frequency
- HF = High Frequency
- VHF = Very High Frequency

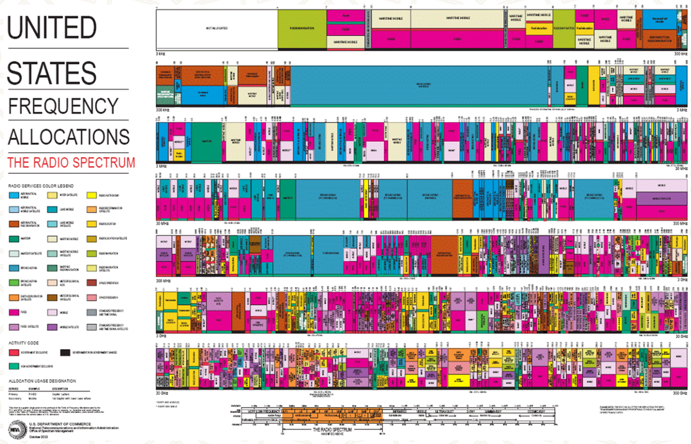
UHF = Ultra High Frequency
SHF = Super High Frequency
EHF = Extra High Frequency
UV = Ultraviolet Light

Frequency Allocation

- Some frequencies are allocated to specific uses
  - Cellular phones, analog television/radio broadcasting, DVB-T, radar, emergency services, radio astronomy, …
- Particularly interesting: ISM bands ("Industrial, scientific, medicine") – license-free operation

Some typical ISM bands	
Frequency	Comment
13,553-13,567 MHz	
26,957 – 27,283 MHz	
40,66 – 40,70 MHz	
433 – 464 MHz	Europe
900 – 928 MHz	Americas
2,4 – 2,5 GHz	WLAN/WPAN
5,725 – 5,875 GHz	WLAN
24 – 24,25 GHz	

Example: US Frequency Allocation



Overview

- Frequency bands
- **Modulation**
- Signal distortion – wireless channels
- From waves to bits
- Channel models
- Transceiver design

Transmitting Data Using Radio Waves

- Basics: Transmit can send a radio wave, receive can detect whether such a wave is present and also its parameters
- Parameters of a wave = sine function:

$$s(t) = A(t)\sin(2\pi f(t)t + \phi(t))$$

  - Parameters: amplitude A(t), frequency f(t), phase ϕ(t)
- Manipulating these three parameters allows the sender to express data; receiver reconstructs data from signal
- Simplification: Receiver “sees” the same signal that the sender generated – not true, see later!

Modulation and Keying

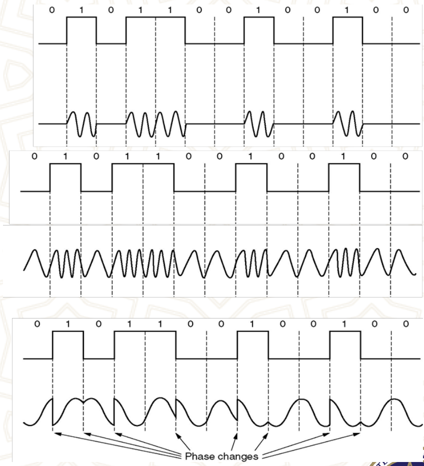
- How to manipulate a given signal parameter?
  - Set the parameter to an arbitrary value: analog modulation
  - Choose parameter values from a finite set of legal values: digital keying
  - Simplification: When the context is clear, modulation is used in either case
- Modulation?
  - Data to be transmitted is used select transmission parameters as a function of time
  - These parameters modify a basic sine wave, which serves as a starting point for modulating the signal onto it
  - This basic sine wave has a center frequency fc
  - The resulting signal requires a certain bandwidth to be transmitted (centered around center frequency)

Modulation (keying!) Examples

- Use data to modify the amplitude of a carrier frequency ! Amplitude Shift Keying

- Use data to modify the frequency of a carrier frequency ! Frequency Shift Keying

- Use data to modify the phase of a carrier frequency ! Phase Shift Keying



Receiver: Demodulation

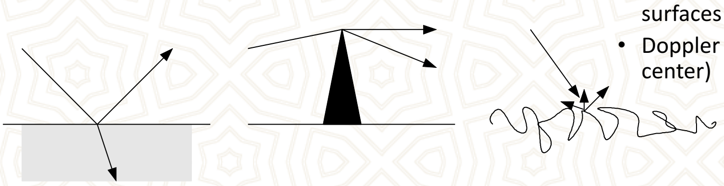
- The receiver looks at the received wave form and matches it with the data bit that caused the transmitter to generate this wave form
  - Necessary: one-to-one mapping between data and wave form
  - Because of channel imperfections, this is at best possible for digital signals, but not for analog signals
- Problems caused by
  - Carrier synchronization: frequency can vary between sender and receiver (drift, temperature changes, aging, …)
  - Bit synchronization (actually: symbol synchronization): When does symbol representing a certain bit start/end?
  - Frame synchronization: When does a packet start/end?
  - Biggest problem: Received signal is not the transmitted signal!

Overview

- Frequency bands
- Modulation
- **Signal distortion – wireless channels**
- From waves to bits
- Channel models
- Transceiver design

Transmitted Signal <> Received Signal!

- Wireless transmission distorts any transmitted signal
  - Received <> transmitted signal; results in uncertainty at receiver about which bit sequence originally caused the transmitted signal
  - Abstraction: Wireless channel describes these distortion effects
- Sources of distortion
  - Attenuation – energy is distributed to larger areas with increasing distance
  - Reflection/refraction – bounce of a surface; enter material
  - Diffraction – start "new wave" from a sharp edge
  - Scattering – multiple reflections at rough surfaces
  - Doppler fading – shift in frequencies (loss of center)



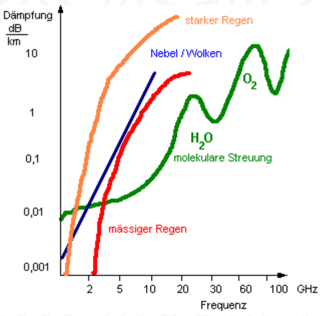
Attenuation Results in Path Loss

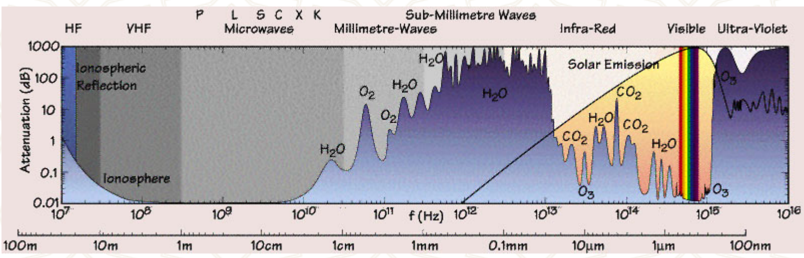
- Effect of attenuation: received signal strength is a function of the distance d between sender and transmitter
- Captured by Friis free-space equation
  - Describes signal strength at distance d relative to some reference distance d0 < d for which strength is known
  - d0 is far-field distance, depends on antenna technology

$$P_{\text{recv}}(d) = \frac{P_{\text{tx}} \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 \cdot d^2 \cdot L} = \frac{P_{\text{tx}} \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 \cdot d_0^2 \cdot L} \cdot \left(\frac{d_0}{d}\right)^2 = P_{\text{recv}}(d_0) \cdot \left(\frac{d_0}{d}\right)^2$$

Suitability of Different Frequencies – Attenuation

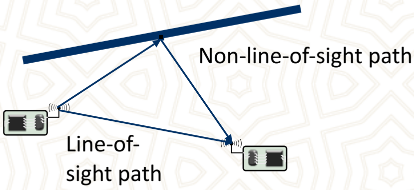
- Attenuation depends on the used frequency
- Can result in a frequency-selective channel
  - If bandwidth spans frequency ranges with different attenuation properties

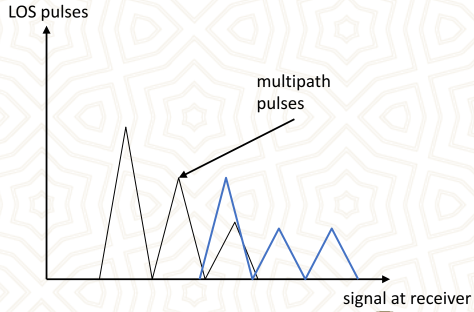




Distortion Effects: Non-line-of-sight Paths

- Because of reflection, scattering, …, radio communication is not limited to direct line of sight communication
  - Effects depend strongly on frequency, thus different behavior at higher frequencies

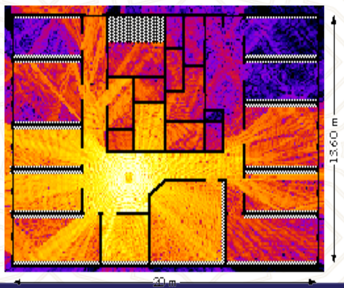


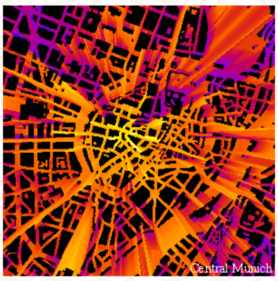


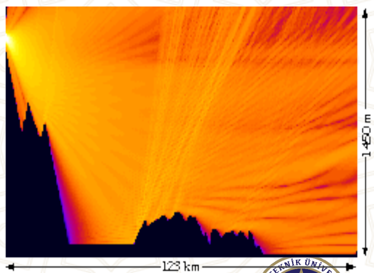
- Different paths have different lengths = propagation time
  - Results in delay spread of the wireless channel
  - Closely related to frequency-selective fading properties of the channel
  - With movement: fast fading

Wireless Signal Strength in a Multi-Path Environment

- Brighter color = stronger signal
- Obviously, simple (quadratic) free space attenuation formula is not sufficient to capture these effects







Generalizing the Attenuation Formula

- To take into account stronger attenuation than only caused by distance (e.g., walls, …), use a larger exponent $\gamma > 2$
  - γ is the path-loss exponent

$$P_{\text{recv}}(d) = P_{\text{recv}}(d_0) \cdot \left(\frac{d_0}{d}\right)^{\gamma}$$

  - Rewrite in logarithmic form (in dB):

$$\text{PL}(d)[\text{dB}] = \text{PL}(d_0)[\text{dB}] + 10\gamma \log_{10}\left(\frac{d}{d_0}\right)$$

- Take obstacles into account by a random variation
  - Add a Gaussian random variable with 0 mean, variance $\sigma 2$ to dB representation
  - Equivalent to multiplying with a lognormal distributed r.v. in metric units ! lognormal fading

$$\text{PL}(d)[\text{dB}] = \text{PL}(d_0)[\text{dB}] + 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma[\text{dB}]$$

Overview

- Frequency bands
- Modulation
- Signal distortion – wireless channels
- **From waves to bits**
- Channel models
- Transceiver design

Noise and Interference

- So far: only a single transmitter assumed
  - Only disturbance: self-interference of a signal with multi-path "copies" of itself
- In reality, two further disturbances
  - Noise – due to effects in receiver electronics, depends on temperature
    - Typical model: an additive Gaussian variable, mean 0, no correlation in time
  - Interference from third parties
    - Co-channel interference: another sender uses the same spectrum
    - Adjacent-channel interference: another sender uses some other part of the radio spectrum, but receiver filters are not good enough to fully suppress it
- Effect: Received signal is distorted by channel, corrupted by noise and interference
  - What is the result on the received bits?

Symbols and Bit Errors

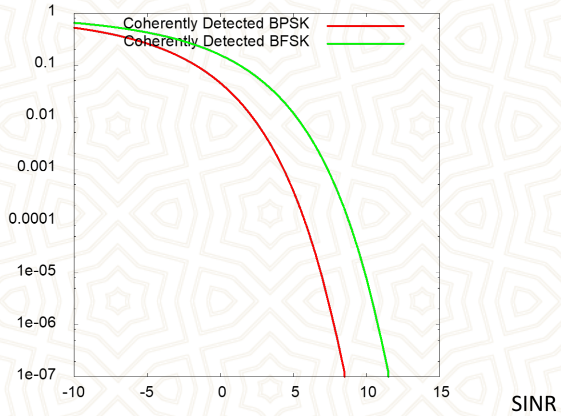
- Extracting symbols out of a distorted/corrupted wave form is fraught with errors
  - Depends essentially on strength of the received signal compared to the corruption
  - Captured by signal to noise and interference ratio (SINR)

$$\mathrm{SINR} = 10 \log_{10} \left(\frac{P_{\mathrm{recv}}}{N_0 + \sum_{i=1}^{k} I_i} \right)$$

- SINR allows to compute bit error rate (BER) for a given modulation
  - Also depends on data rate (# bits/symbol) of modulation
  - E.g., for simple DPSK, data rate corresponding to bandwidth:

$$\mathrm{BER}(\mathrm{SINR}) = 0.5 e^{-\frac{E_b}{N_0}}$$
$$E_b / N_0 = \mathrm{SINR} \cdot \frac{1}{R}$$

Examples for SINR ! BER mappings



Overview

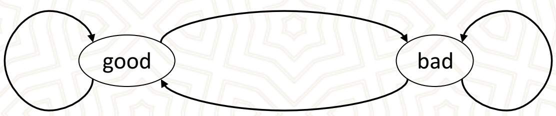
- Frequency bands
- Modulation
- Signal distortion – wireless channels
- From waves to bits
- **Channel models**
- Transceiver design

Channel Models – Analog

- How to stochastically capture the behavior of a wireless channel
  - Main options: model the SNR or directly the bit errors
- Signal models
  - Simplest model: assume transmission power and attenuation are constant, noise an uncorrelated Gaussian variable
    - Additive White Gaussian Noise model, results in constant SNR
  - Situation with no line-of-sight path, but many indirect paths: Amplitude of resulting signal has a Rayleigh distribution (Rayleigh fading)
  - One dominant line-of-sight plus many indirect paths: Signal has a Rice distribution (Rice fading)

Channel Models – Digital

- Directly model the resulting bit error behavior
  - Each bit is erroneous with constant probability, independent of the other bits ! binary symmetric channel (BSC)
  - Capture fading models' property that channel be in different states ! Markov models – states with different BERs
    - Example: Gilbert-Elliot model with "bad" and "good" channel states and high/low bit error rates



  - Fractal channel models describe number of (in-)correct bits in a row by a heavy-tailed distribution

WSN-Specific Channel Models

- Typical WSN properties
  - Small transmission range
  - Implies small delay spread (nanoseconds, compared to micro/milliseconds for symbol duration)
  - ! Frequency-non-selective fading, low to
    - Coherence bandwidth often > 50 MHz
- Some example measurements
  - γ path loss exponent
  - Shadowing variance $\sigma 2$
  - Reference path loss at 1 m

Location	Average of γ	Average of σ^2[dB]	Range of PL(1m)[dB]
Engineering Building	1.9	5.7	$[-50.5, -39.0]$
Apartment Hallway	2.0	8.0	$[-38.2, -35.0]$
Parking Structure	3.0	7.9	$[-36.0, -32.7]$
One-sided Corridor	1.9	8.0	$[-44.2, -33.5]$
One-sided patio	3.2	3.7	$[-39.0, -34.2]$
Concrete canyon	2.7	10.2	$[-48.7, -44.0]$
Plant fence	4.9	9.4	$[-38.2, -34.5]$
Small boulders	3.5	12.8	$[-41.5, -37.2]$
Sandy flat beach	4.2	4.0	$[-40.8, -37.5]$
Dense bamboo	5.0	11.6	$[-38.2, -35.2]$
Dry tall underbrush	3.6	8.4	$[-36.4, -33.2]$

Wireless Channel Quality – Summary

- Wireless channels are substantially worse than wired channels
  - In throughput, bit error characteristics, energy consumption, …
- Wireless channels are extremely diverse
  - There is no such thing as THE typical wireless channel
- Various schemes for quality improvement exist
  - Some of them geared towards high-performance wireless communication – not necessarily suitable for WSN, ok for MANET
    - Diversity, equalization, …
  - Some of them general-purpose (ARQ, FEC)
  - Energy issues need to be taken into account!

Overview

- Frequency bands
- Modulation
- Signal distortion – wireless channels
- From waves to bits
- Channel models
- **Transceiver design**

Some Transceiver Design Considerations

- Strive for good power efficiency at low transmission power
  - Some amplifiers are optimized for efficiency at high output power
  - To radiate 1 mW, typical designs need 30-100 mW to operate the transmitter
    - WSN nodes: 20 mW (mica motes)
  - Receiver can use as much or more power as transmitter at these power levels
    - ! Sleep state is important
- Startup energy/time penalty can be high
  - Examples take 0.5 ms and ¼ 60 mW to wake up
- Exploit communication/computation tradeoffs
  - Might payoff to invest in rather complicated coding/compression schemes

Choice of Modulation

- One exemplary design point: which modulation to use?
  - Consider: required data rate, available symbol rate, implementation complexity, required BER, channel characteristics, …
  - Tradeoffs: the faster one sends, the longer one can sleep
    - Power consumption can depend on modulation scheme
  - Tradeoffs: symbol rate (high?) versus data rate (low)
    - Use m-ary transmission to get a transmission over with ASAP
    - But: startup costs can easily void any time saving effects
    - For details: see example in exercise!
- Adapt modulation choice to operation conditions
  - Akin to dynamic voltage scaling, introduce Dynamic Modulation Scaling

Summary

- Wireless radio communication introduces many uncertainties and vagaries into a communication system
- Handling the unavoidable errors will be a major challenge for the communication protocols
- Dealing with limited bandwidth in an energy-efficient manner is the main challenge
- MANET and WSN are pretty similar here
  - Main differences are in required data rates and resulting transceiver complexities (higher bandwidth, spread spectrum techniques)

Chapter 5: Medium access control protocols

Goals of This Chapter

- Controlling when to send a packet and when to listen for a packet are perhaps the two most important operations in a wireless network
  - Especially, idly waiting wastes huge amounts of energy
- This chapter discusses schemes for this medium access control that are
  - Suitable to mobile and wireless networks
  - Emphasize energy-efficient operation

Overview

- **Principal options and difficulties**
- Contention-based protocols
- Schedule-based protocols
- IEEE 802.15.4

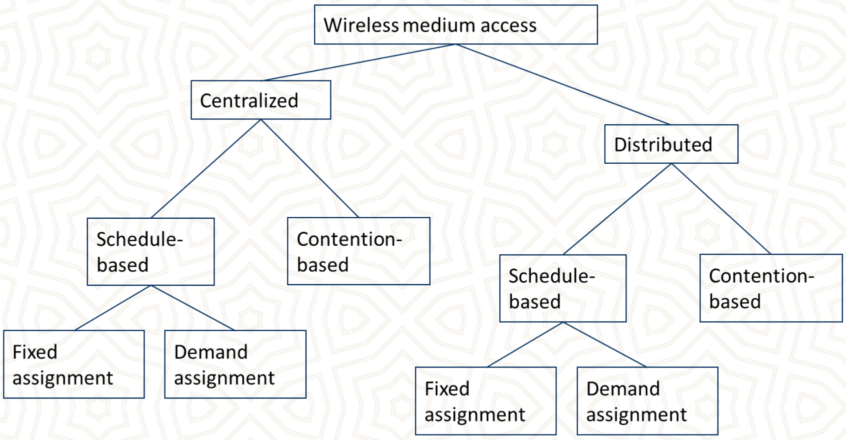
Principal options and difficulties

- Medium access in wireless networks is difficult mainly because of
  - Impossible (or very difficult) to sende and receive at the same time
  - Interference situation at receiver is what counts for transmission success, but can be very different from what sender can observe
  - High error rates (for signaling packets) compound the issues
- Requirement
  - As usual: high throughput, low overhead, low error rates, …
  - Additionally: energy-efficient, handle switched off devices!

Requirements for Energy-Efficient MAC Protocols

- Recall
  - Transmissions are costly
  - Receiving about as expensive as transmitting
  - Idling can be cheaper but is still expensive
- Energy problems
  - Collisions – wasted effort when two packets collide
  - Overhearing – waste effort in receiving a packet destined for another node
  - Idle listening – sitting idly and trying to receive when nobody is sending
  - Protocol overhead
- Always nice: Low complexity solution

Main Options



Centralized Medium Access

- Idea: Have a central station control when a node may access the medium
  - Example: Polling, centralized computation of TDMA schedules
  - Advantage: Simple, quite efficient (e.g., no collisions), burdens the central station
- Not directly feasible for non-trivial wireless network sizes
- But: Can be quite useful when network is somehow divided into smaller groups
  - Clusters, in each cluster medium access can be controlled centrally – compare Bluetooth piconets, for example
- ! Usually, distributed medium access is considered

Schedule- vs. Contention-Based MACs

- Schedule-based MAC
  - A schedule exists, regulating which participant may use which resource at which time (TDMA component)
  - Typical resource: frequency band in a given physical space (with a given code, CDMA)
  - Schedule can be fixed or computed on demand
    - Usually: mixed – difference fixed/on demand is one of time scales
  - Usually, collisions, overhearing, idle listening no issues
  - Needed: time synchronization!
- Contention-based protocols
  - Risk of colliding packets is deliberately taken
  - Hope: coordination overhead can be saved, resulting in overall improved efficiency
  - Mechanisms to handle/reduce probability/impact of collisions required
  - Usually, randomization used somehow

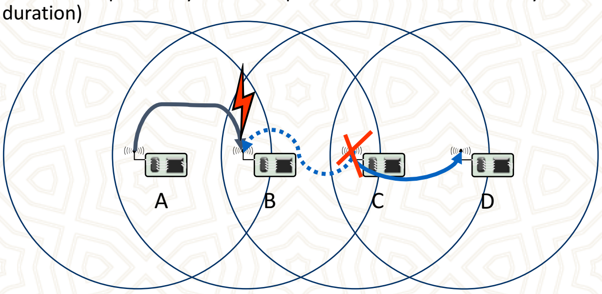
Overview

- Principal options and difficulties
- **Contention-based protocols**
  - MACA
  - S-MAC, T-MAC
  - Preamble sampling, B-MAC
  - PAMAS
- Schedule-based protocols
- IEEE 802.15.4

Distributed, Contention-Based MAC

- Basic ideas for a distributed MAC
  - ALOHA – no good in most cases
  - Listen before talk (Carrier Sense Multiple Access, CSMA) – better, but suffers from sender not knowing what is going on at receiver, might destroy packets despite first listening for a
- ! Receiver additionally needs some possibility to inform possible senders in its vicinity about impending transmission (to “shut them up” for this duration)

Hidden terminal scenario:



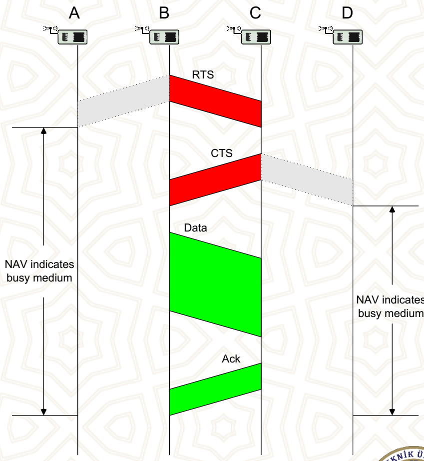
Also: recall exposed terminal scenario

Main Options to Shut up Senders

- Receiver informs potential interferers while a reception is on-going
  - By sending out a signal indicating just that
  - Problem: Cannot use same channel on which actual reception takes place
  - ! Use separate channel for signaling
  - Busy tone protocol
- Receiver informs potential interferers before a reception is on-going
  - Can use same channel
  - Receiver itself needs to be informed, by sender, about impending transmission
  - Potential interferers need to be aware of such information, need to store it

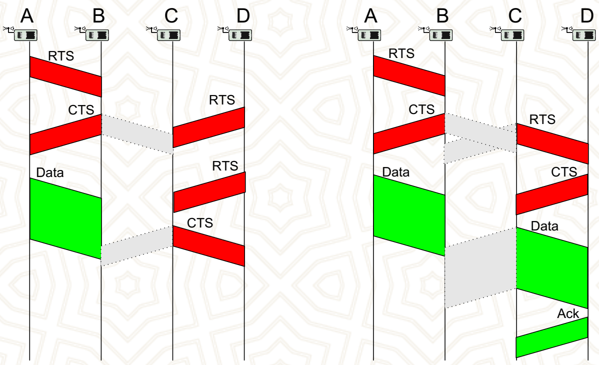
Receiver Informs Interferers Before Transmission – MACA

- Sender B asks receiver C whether C is able to receive a transmission Request to Send (RTS)
- Receiver C agrees, sends out a Clear to Send (CTS)
- Potential interferers overhear either RTS or CTS and know about impending transmission and for how long it will last
  - Store this information in a Network Allocation Vector
- B sends, C acks
- ! MACA protocol (used e.g. in IEEE 802.11)



RTS/CTS

- RTS/CTS ameliorate, but do not solve hidden/exposed terminal problems
- Example problem cases:

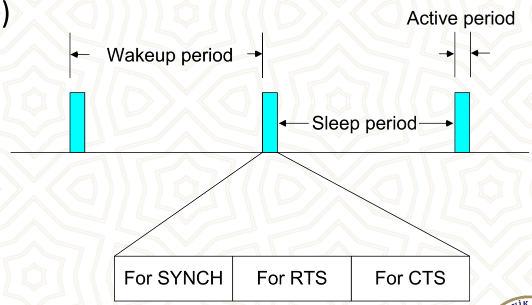


MACA Problem: Idle Listening

- Need to sense carrier for RTS or CTS packets
  - In some form shared by many CSMA variants; but e.g. not by busy tones
  - Simple sleeping will break the protocol
- IEEE 802.11 solution: ATIM windows & sleeping
  - Basic idea: Nodes that have data buffered for receivers send traffic indicators at pre-arranged points in time
  - Receivers need to wake up at these points, but can sleep otherwise
- Parameters to adjust in MACA
  - Random delays – how long to wait between listen/transmission attempts?
  - Number of RTS/CTS/ACK re-trials?
  - …

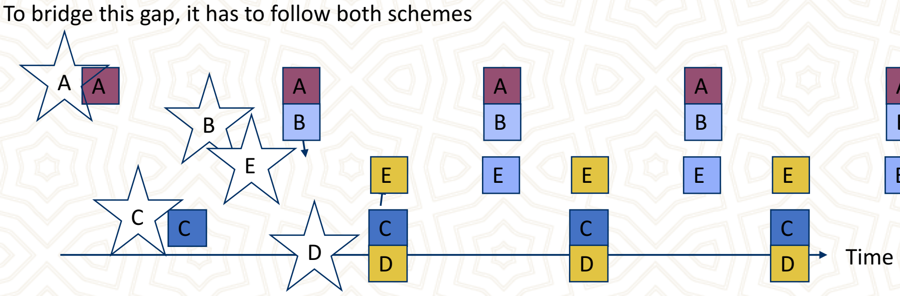
Sensor-MAC (S-MAC)

- MACA's idle listening is particularly unsuitable if average data rate is low
  - Most of the time, nothing happens
- Idea: Switch nodes off, ensure that neighboring nodes turn on simultaneously to allow packet exchange (rendez-vous)
  - Only in these ***active periods***, packet exchanges happen
  - Need to also exchange wakeup schedule between neighbors
  - When awake, essentially perform RTS/CTS
- Use SYNCH, RTS, CTS phases



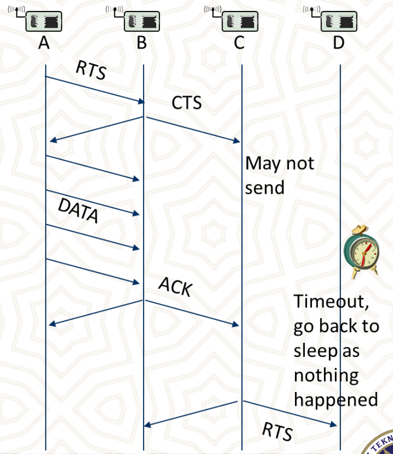
S-MAC Synchronized Islands

- Nodes try to pick up schedule synchronization from neighboring nodes
- If no neighbor found, nodes pick some schedule to start with
- If additional nodes join, some node might learn about two different schedules from different nodes
  - "Synchronized islands"
- To bridge this gap, it has to follow both schemes



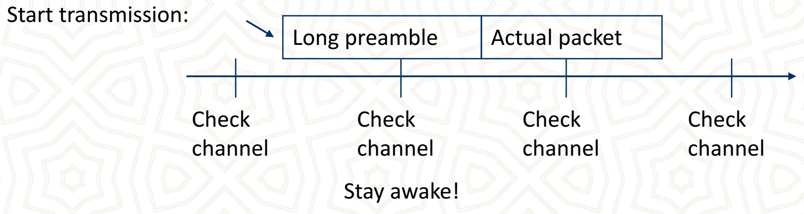
Timeout-MAC (T-MAC)

- In S-MAC, active period is of constant length
- What if no traffic actually happens?
  - Nodes stay awake needlessly long
- Idea: Prematurely go back to sleep mode when no traffic has happened for a certain time (=timeout) ! T-MAC
  - Adaptive duty cycle!
- One ensuing problem: Early sleeping
  - C wants to send to D, but is hindered by transmission A! B
  - Two solutions exist – homework!



Preamble Sampling

- So far: Periodic sleeping supported by some means to synchronize wake up of nodes to ensure rendez-vous between sender and receiver
- Alternative option: Don't try to explicitly synchronize nodes
  - Have receiver sleep and only periodically sample the channel
- Use long preambles to ensure that receiver stays awake to catch actual packet
  - Example: WiseMAC



B-MAC

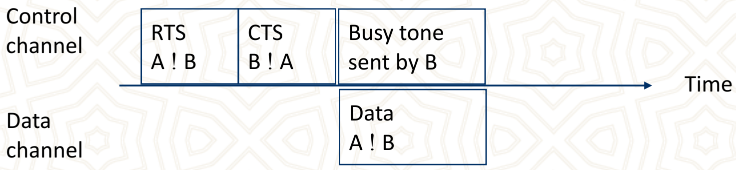
- Combines several of the above discussed ideas
  - Takes care to provide practically relevant solutions
- Clear Channel Assessment
  - Adapts to noise floor by sampling channel when it is assumed to be free
  - Samples are exponentially averaged, result used in gain control
  - For actual assessment when sending a packet, look at five channel samples – channel is free if even a single one of them is significantly below noise
  - Optional: random backoff if channel is found busy
- Optional: Immediate link layer acknowledgements for received packets

B-MAC II

- Low Power Listening (= preamble sampling)
  - Uses the clear channel assessment techniques to decide whether there is a packet arriving when node wakes up
  - Timeout puts node back to sleep if no packet arrived
- B-MAC does not have
  - Synchronization
  - RTS/CTS
  - Results in simpler, leaner implementation
  - Clean and simple interface
- Currently: Often considered as the default WSN MAC protocol

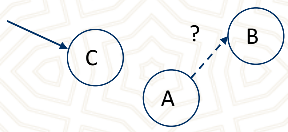
Power Aware Multiaccess with Signaling – PAMAS

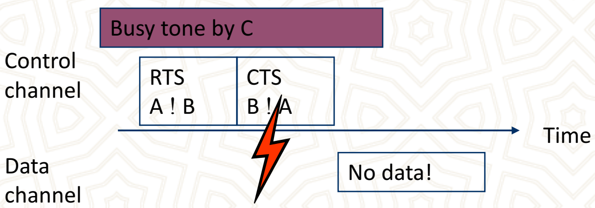
- Idea: combine busy tone with RTS/CTS
  - Results in detailed overhearing avoidance, does not address idle listening
  - Uses separate data and control channels
- Procedure
  - Node A transmits RTS on control channel, does not sense channel
  - Node B receives RTS, sends CTS on control channel if it can receive and does not know about ongoing transmissions
  - B sends busy tone as it starts to receive data



PAMAS – Already Ongoing Transmission

- Suppose a node C in vicinity of A is already receiving a packet when A initiates RTS
- Procedure
- A sends RTS to B
- C is sending busy tone (as it receives data)
- CTS and busy tone collide, A receives no CTS, does not send data





Similarly: Ongoing transmission near B destroys RTS by busy tone

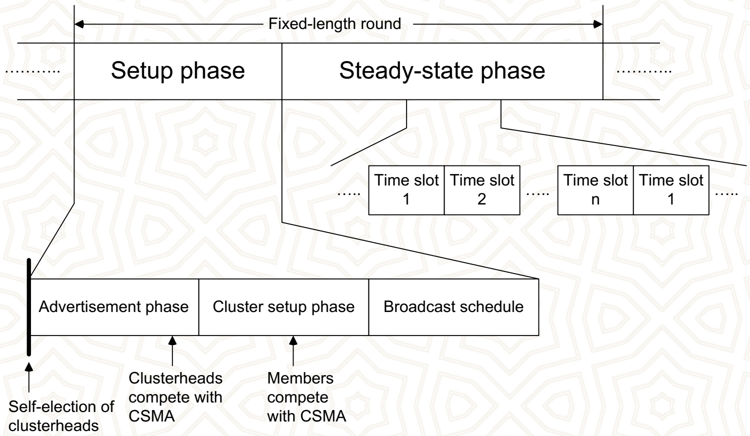
Overview

- Principal options and difficulties
- Contention-based protocols
- **Schedule-based protocols**
  - LEACH
  - SMACS
  - TRAMA
- IEEE 802.15.4

Low-Energy Adaptive Clustering Hierarchy (LEACH)

- Given: dense network of nodes, reporting to a central sink, each node can reach sink directly
- Idea: Group nodes into "clusters", controlled by clusterhead
  - Setup phase; details: later
  - About 5% of nodes become clusterhead (depends on scenario)
  - Role of clusterhead is rotated to share the burden
  - Clusterheads advertise themselves, ordinary nodes join CH with strongest signal
  - Clusterheads organize
    - CDMA code for all member transmissions
    - TDMA schedule to be used within a cluster
- In steady state operation
  - CHs collect & aggregate data from all cluster members
  - Report aggregated data to sink using CDMA

LEACH Rounds

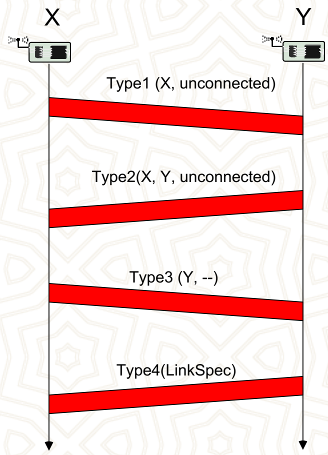


SMACS

- Given: many radio channels, superframes of known length (not necessarily in phase, but still time synchronization required!)
- Goal: set up directional links between neighboring nodes
  - Link: radio channel + time slot at both sender and receiver
  - Free of collisions at receiver
  - Channel picked randomly, slot is searched greedily until a collision-free slot is found
- Receivers sleep and only wake up in their assigned time slots, once per superframe
- In effect: a local construction of a schedule

SMACS Link Setup

- Case 1: Node X, Y both so far unconnected
  - Node X sends invitation message
  - Node Y answers, telling X that is unconnected to any other node
  - Node X tells Y to pick slot/frequency for the link
  - Node Y sends back the link specification
- Case 2: X has some neighbors, Y not
  - Node X will construct link specification and instruct Y to use it (since Y is unattached)
- Case 3: X no neighbors, Y has some
  - Y picks link specification
- Case 4: both nodes already have links
  - Nodes exchange their schedules and pick free slots/frequencies in mutual agreement



Message exchanges protected by randomized backoff

TRAMA

- Nodes are synchronized
- Time divided into cycles, divided into
  - Random access periods
  - Scheduled access periods
- Nodes exchange neighborhood information
  - Learning about their two-hop neighborhood
  - Using neighborhood exchange protocol: In random access period, send small, incremental neighborhood update information in randomly selected time slots
- Nodes exchange schedules
  - Using schedule exchange protocol
  - Similar to neighborhood exchange

TRAMA – Adaptive Election

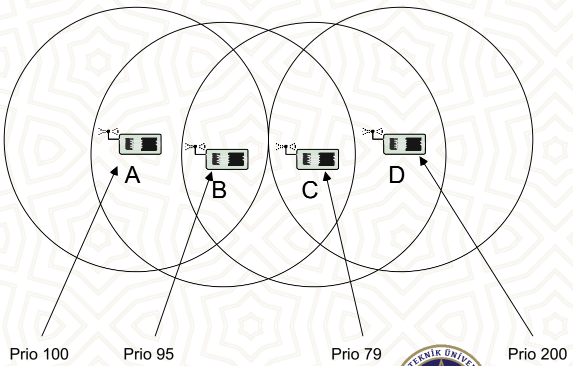
- Given: Each node knows its two-hop neighborhood and their current schedules
- How to decide which slot (in scheduled access period) a node can use?
  - Use node identifier x and globally known hash function h
  - For time slot t, compute priority p = h (x © t)
  - Compute this priority for next k time slots for node itself and all two-hop neighbors
  - Node uses those time slots for which it has the highest priority

Priorities of node A and its two neighbors B & C

	t = 0	t = 1	t = 2	t=3	t = 4	t = 5
A	14	23	9	56	3	26
B	33	64	8	12	44	6
C	53	18	6	33	57	2

TRAMA – Possible Conflicts

- When does a node have to receive?
- Easy case: one-hop neighbor has won a time slot and announced a packet for it
- What does B believe?
  - A thinks it can send
  - B knows that D has higher priority in its 2-hop neighborhood!
- Rules for resolving such conflicts are part of TRAMA



Comparison: TRAMA, S-MAC

- Comparison between TRAMA & S-MAC
  - Energy savings in TRAMA depend on load situation
  - Energy savings in S-MAC depend on duty cycle
  - TRAMA (as typical for a TDMA scheme) has higher delay but higher maximum throughput than contention-based S-MAC
- TRAMA disadvantage: substantial memory/CPU requirements for schedule computation

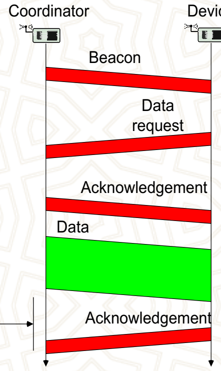
Overview

- Principal options and difficulties
- Contention-based protocols
- Schedule-based protocols
- **IEEE 802.15.4**

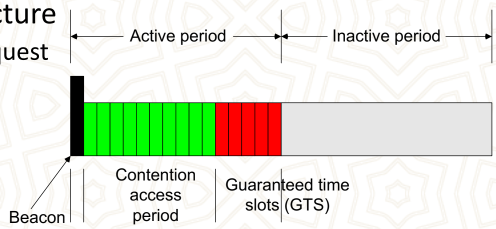
IEEE 802.15.4

- IEEE standard for low-rate WPAN applications
- Goals: low-to-medium bit rates, moderate delays without too stringent guarantee requirements, low energy consumption
- Physical layer
  - 20 kbps over 1 channel @ 868-868.6 MHz
  - 40 kbps over 10 channels @ 905 – 928 MHz
  - 250 kbps over 16 channels @ 2.4 GHz
- MAC protocol
  - Single channel at any one time
  - Combines contention-based and schedule-based schemes
  - Asymmetric: nodes can assume different roles

IEEE 802.15.4 MAC overview



- Star networks: devices are associated with coordinators
  - Forming a PAN, identified by a PAN identifier
- Coordinator
  - Bookkeeping of devices, address assignment, generate beacons
  - Talks to devices and peer coordinators
- Beacon-mode superframe structure
  - GTS assigned to devices upon request



Wakeup Radio MAC Protocols

- Simplest scheme: Send a wakeup "burst", waking up all neighbors ! Significant overhearing
  - Possible option: First send a short filter packet that includes the actual destination address to allow nodes to power off quickly
- Not quite so simple scheme: Send a wakeup burst including the receiver address
  - Wakeup radio needs to support this option
- Additionally: Send information about a (randomly chosen) data channel, CDMA code, … in the wakeup burst
- Various variations on these schemes in the literature, various further problems
  - One problem: 2-hop neighborhood on wakeup channel might be different from 2-hop neighborhood on data channel
  - Not trivial to guarantee unique addresses on both channels

Further Protocols

- MAC protocols for ad hoc/sensor networks is one the most active research fields
  - Tons of additional protocols in the literature
  - Examples: STEM, mediation device protocol, many CSMA variants with different timing optimizations, protocols for multi-hop reservations (QoS for MANET), protocols for multiple radio channels, …
  - Additional problems, e.g., reliable multicast
- This chapter has barely scratched the surface…

Summary

- Many different ideas exist for medium access control in MANET/WSN
- Comparing their performance and suitability is difficult
- Especially: clearly identifying interdependencies between MAC protocol and other layers/applications is difficult
  - Which is the best MAC for which application?
- Nonetheless, certain “common use cases” exist
  - IEEE 802.11 DCF for MANET
  - IEEE 802.15.4 for some early “commerical” WSN variants
  - B-MAC for WSN research not focusing on MAC

Chapter 6: Link layer protocols

Goals of This Chapter – Link Layer Tasks in General

- Framing – group bit sequence into packets/frames
  - Important: format, size
- Error control – make sure that the sent bits arrive and no other
  - Forward and backward error control
- Flow control – ensure that a fast sender does not overrun its slow(er) receiver
- Link management – discovery and manage links to neighbors
  - Do not use a neighbor at any cost, only if link is good enough
- ! Understand the issues involved in turning the radio communication between two neighboring nodes into a somewhat reliable link

Overview

- **Error control**
- Framing
- Link management

Error Control

- Error control has to ensure that data transport is
  - Error-free – deliver exactly the sent bits/packets
  - In-sequence – deliver them in the original order
  - Duplicate-free – and at most once
  - Loss-free – and at least once
- Causes: fading, interference, loss of bit synchronization, …
  - Results in bit errors, bursty, sometimes heavy-tailed runs (see physical layer chapter)
  - In wireless, sometimes quite high average bit error rates – 10-2 … 10-4 possible!
- Approaches
  - Backward error control – ARQ
  - Forward error control – FEC

Backward Error Control – ARQ

- Basic procedure (a quick recap)
  - Put header information around the payload
  - Compute a checksum and add it to the packet
    - Typically: Cyclic redundancy check (CRC), quick, low overhead, low residual error rate
  - Provide feedback from receiver to sender
    - Send positive or negative acknowledgement
  - Sender uses timer to detect that acknowledgements have not arrived
    - Assumes packet has not arrived
    - Optimal timer setting?
  - If sender infers that a packet has not been received correctly, sender can retransmit it
    - What is maximum number of retransmission attempts? If bounded, at best a semi-reliable protocols results

Standard ARQ protocols

- Alternating bit – at most one packet outstanding, single bit sequence number
- Go-back N – send up to N packets, if a packet has not been acknowledged when timer goes off, retransmit all unacknowledged packets
- Selective Repeat – when timer goes off, only send that particular packet

How to Use Acknowledgements

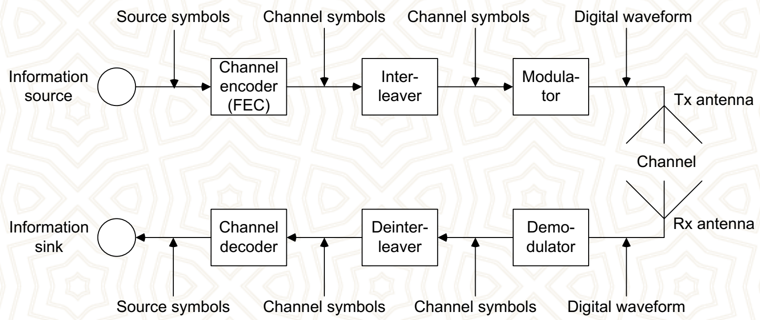
- Be careful about ACKs from different layers
  - A MAC ACK (e.g., S-MAC) does not necessarily imply buffer space in the link layer
  - On the other hand, having both MAC and link layer ACKs is a waste
- Do not (necessarily) acknowledge every packet – use cumulative ACKs
  - Tradeoff against buffer space
  - Tradeoff against number of negative ACKs to send

When to Retransmit

- Assuming sender has decided to retransmit a packet – when to do so?
  - In a BSC channel, any time is as good as any
  - In fading channels, try to avoid bad channel states – postpone transmissions
  - Instead (e.g.): send a packet to another node if in queue (exploit multi-user diversity)
- How long to wait?
  - Example solution: Probing protocol
  - Idea: reflect channel state by two protocol modes, "normal" and "probing"
  - When error occurs, go from normal to probing mode
  - In probing mode, periodically send short packets (acknowledged by receiver) – when successful, go to normal mode

Forward Error Control

- Idea: Endow symbols in a packet with additional redundancy to withstand a limited amount of random permutations
  - Additionally: interleaving – change order of symbols to withstand burst errors



Block-coded FEC

- Level of redundancy: blocks of symbols
  - Block: k p-ary source symbols (not necessarily just bits)
  - Encoded into n q-ary channel symbols
- Injective mapping (code) of pk source symbols ! qn channel symbols
- Code rate: (k ld p) / (n ld q)
  - When p=q=2: k/n is code rate
- For p=q=2: Hamming bound – code can correct up to t bit errors only if

$$2^{n-k} \geq \sum_{i=0}^{t} \binom{n}{i}$$

  - Codes for (n,k,t) do not always exist

Popular Block Codes

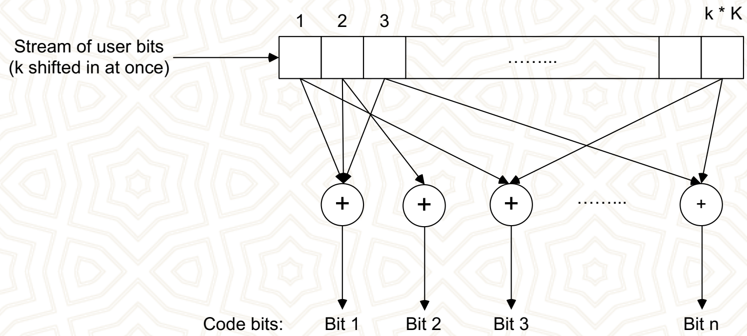
- Popular examples
  - Reed-Solomon codes (RS)
  - Bose-Chaudhuri-Hocquenghem codes (BCH)
- Energy consumption
  - E.g., BCH encoding: negligible overhead (linear-feedback shift register)
  - BCH decoding: depends on block length and Hamming distance (n, t as on last slide)

$$E_{\text{dec}} = (2nt + 2t^2) \cdot (E_{\text{add}} + E_{\text{mult}})$$

  - Similar for RS codes

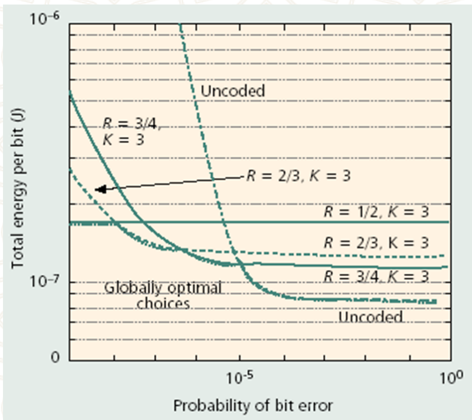
Convolutional Codes

- Code rate: ratio of k user bits mapped onto n coded bits
- Constraint length K determines coding gain
- Energy
  - Encoding: cheap
  - Decoding: Viterbi algorithm, energy & memory depends exponentially (!) on constraint length



Energy Consumption of Convolutional Codes

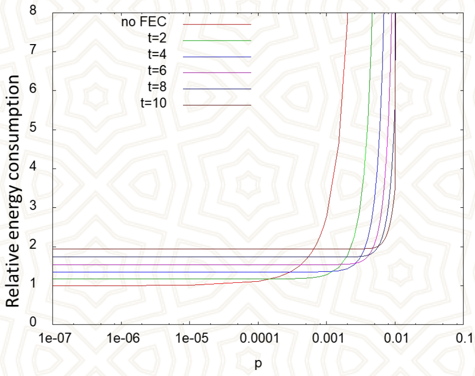
- Tradeoff between coding energy and reduced transmission power (coding gain)
- Overall: block codes tend to be more energy-efficient



Comparison: FEC vs. ARQ

- FEC
  - Constant overhead for each packet
  - Not (easily) possible to adapt to changing channel characteristics
- ARQ
  - Overhead only when errors occurred (expect for ACK, always needed)
- Both schemes have their uses ! hybrid schemes

t: error correction capacity



BCH + unlimited number of retransmissions

Power Control on a Link Level

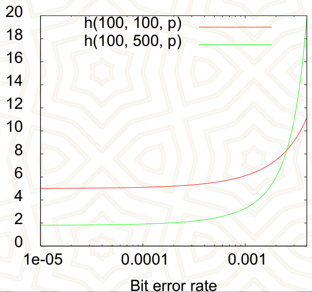
- Further controllable parameter: transmission power
  - Higher power, lower error rates – less FEC/ARQ necessary
  - Lower power, higher error rates – higher FEC necessary
- Tradeoff!

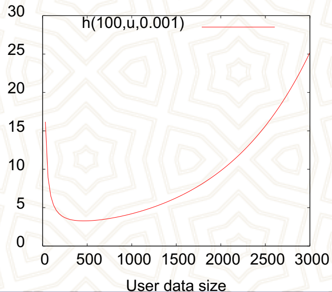
Overview

- Error control
- **Framing**
- Link management

Frame, Packet Size

- Small packets: low packet error rate, high packetization overhead
- Large packets: high packet error rate, low overhead
- Depends on bit error rate, energy consumption per transmitted bit
- Notation: h(overhead, payload size, BER)



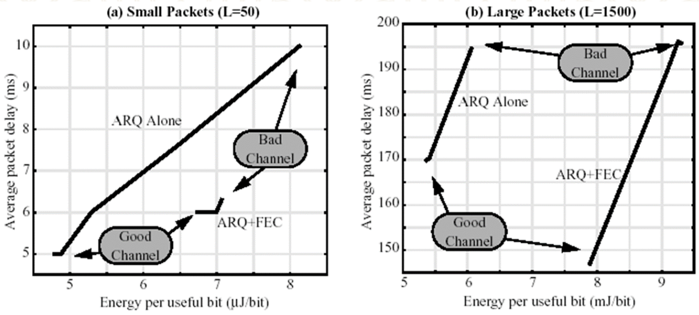


Dynamically Adapt Frame Length

- For known bit error rate (BER), optimal frame length is easy to determine
- Problem: how to estimate BER?
  - Collect channel state information at the receiver (RSSI, FEC decoder information, …)
  - Example: Use number of attempts T required to transmit the last M packets as an estimator of the packet error rate (assuming a BSC)
    - Details: homework assignment
- Second problem: how long are observations valid/how should they be aged?
  - Only recent past is – if anything at all – somewhat credible

Putting It Together: ARQ, FEC, Frame Length Optimization

- Applying ARQ, FEC (both block and convolutional codes), frame length optimization to a Rayleigh fading channel
  - Channel modeled as Gilbert-Elliot



Overview

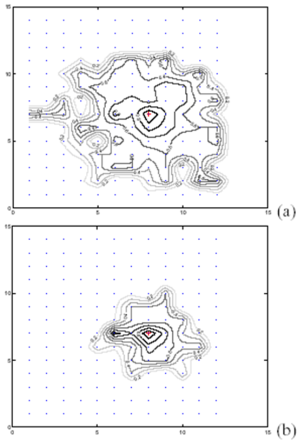
- Error control
- Framing
- **Link management**

Link Management

- Goal: decide to which neighbors that are more or less reachable a link should be established
  - Problem: communication quality fluctuates, far away neighbors can be costly to talk to, error-prone, quality can only be estimated
- Establish a neighborhood table for each node
  - Partially automatically constructed by MAC protocols

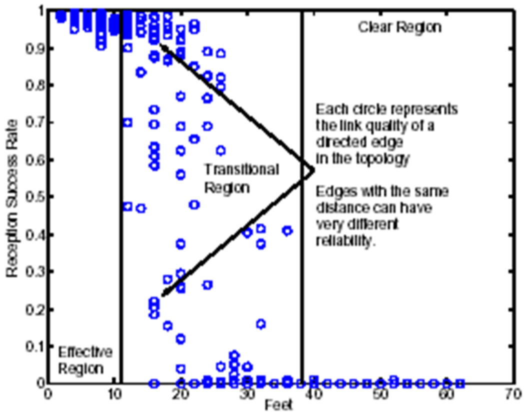
Link Quality Characteristics

- Expected: simple, circular shape of "region of communication" – not realistic
- Instead:
  - Correlation between distance and loss rate is weak; iso-loss-lines are not circular but irregular
  - Asymmetric links are relatively frequent (up to 15%)
  - Significant short-term PER variations even for stationary nodes



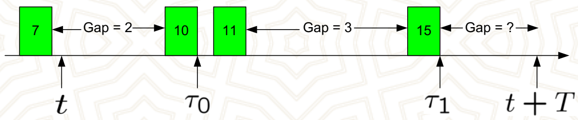
Three Regions of Communication

- Effective region: PER consistently < 10%
- Transitional region: anything in between, with large variation for nodes at same distance
- Poor region: PER well beyond 90%



Link Quality Estimation

- How to estimate, on-line, in the field, the actual link quality?
- Requirements
  - Precision – estimator should give the statistically correct result
  - Agility – estimator should react quickly to changes
  - Stability – estimator should not be influenced by short aberrations
  - Efficiency – Active or passive estimator



$$P_n = \alpha P_{n-1} + (1-\alpha)\frac{r_n}{r_n+f_n}$$

r_n: received packets in interval

f_n: packets identified as lost

- Example: WMEWMA only estimates at fixed intervals

Conclusion

- Link layer combines traditional mechanisms
  - Framing, packet synchronization, flow control
- with relatively specific issues
  - Careful choice of error control mechanisms – tradeoffs between FEC & ARQ & transmission power & packet size …
  - Link estimation and characterization