

Northwestern University

MITP 491: Selected Topics in Information Technology

Topic 3: Sensor Networks and RFIDs

Part 5

Instructor: Randall Berry

e-mail: rberry@ece.northwestern.edu

Review

- Introduction
 - Applications/technology trends.
- Single chip architecture
 - Hardware/software
- Design considerations
 - Energy efficient designs
- RFIDs
- IEEE 802.15.4/Zigbee

Today

- Discuss applications in HW.
- Loose ends from last time
- Networking issues:
 - MAC protocols
 - Topology formation
 - Routing
- Security
- Future trends

Homework Discussion

Loose ends

1. Wireless power (Thanks to Kevin)
2. Operating systems and power saving techniques.
3. I-Pass (semi-active RFID example).
4. RFIDs meet motes.

Wireless Power

- Technique developed by Powercast to re-charge batteries via RF transmissions.
- May be reasonable alternative for short-range sensor applications.
- For more details see wireless project presentations.

OS support for power savings

- Discussed a number of power saving techniques:
 - Dynamic power management
 - Dynamic voltage scaling
- Embedded OS's (e.g. tiny OS) may provide support for these.
 - Provide “hooks” for developers.
 - Automate these features.

OS support for power savings

tiny OS example.

- Key feature of any OS is a scheduler.
- In tiny OS serve jobs in FIFO order.
 - When no work, OS puts processor and peripherals into sleep state.
 - Also provides commands to set default sleep state (depends on hardware).
 - Likewise, sleep/wake commands provided for all peripherals.

OS support for power savings

What about DVS?

- Tiny OS does not currently support this.
- Any example of an OS that does is *eCos* (general purpose embedded OS).
- DVS can be implemented as part of the OS's scheduler
 - requires tasks to specify “deadlines.”

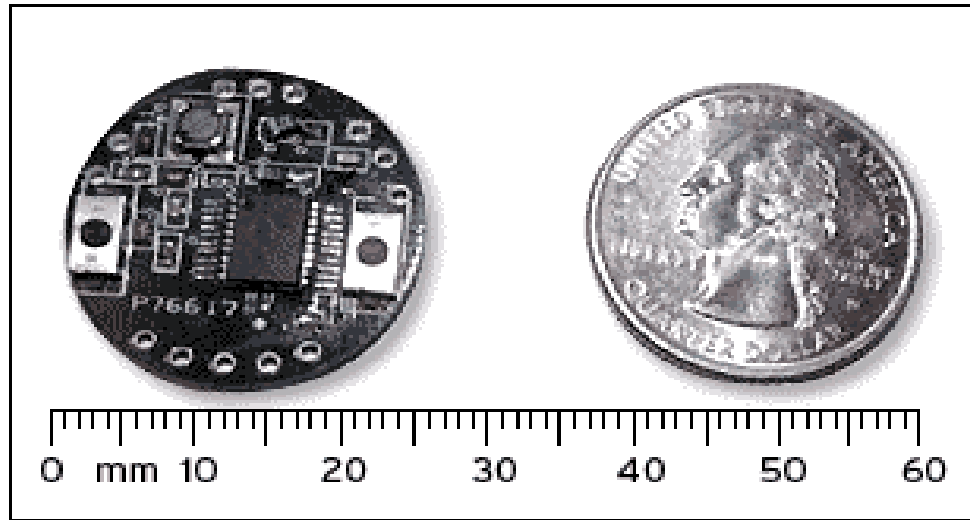
I-Pass example (RFID)

- An I-Pass is a *semi-active* RFID.
 - Powered by a Lithium battery (10 year design time.)
 - Operates at 900 MHz.
 - Sends 256 bit packets @ 500 kbps.
 - Has read/write capability.

From Computer Desktop Encyclopedia
© 2006 The Computer Language Company Inc.



RFIDs meet Motes



- **SkyeTek RFID reader:**
 - Reads tags 2.5” away with internal antenna (5” with external).
 - Can connect to Mica2Dot motes.
 - Uses?

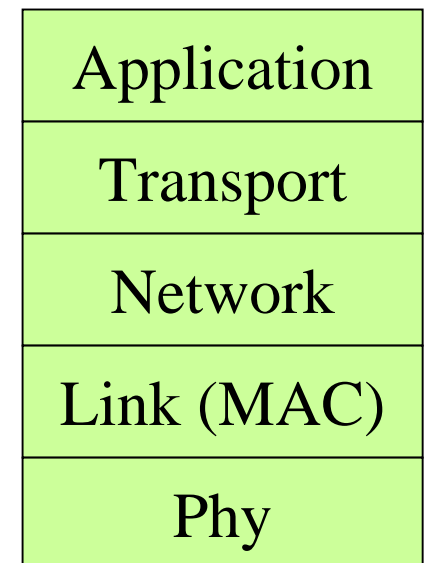
Networking issues

- Last time briefly looked at Zigbee protocol stack.
- Today - more general look at networking issues.
 - Will not only address Zigbee.

Sensor Network Architecture

- Network architecture refers to global view of how a network is designed.
- The Internet (TCP/IP) is a hugely successful network architecture.
 - Layered architecture.
 - IP “hourglass”

Why not just use this for sensor nets ?



Sensor Network Architecture

- Sensor network architectures **loosely** follow Internet protocol stack, but generally do **not** use TCP/IP.
 - At some point a SN will likely be connected to the Internet by one or more **gateway** nodes.
 - Issues of protocol/address translation are important here.
- A key reason is again **energy efficiency**
 - TCP/IP not designed with this in mind.
 - SN architectures are much more highly optimized.

Cross-layer design

- In traditional layered architectures, each layer provides a minimal “black-box” abstraction to next higher layer.
- Not clear this will suffice for sensor networks.
- Many designs relax strict layering and allow “layers” to exchange more information to “work together”
- Example:
 - Physical layer transmission energy to reach a node can be used by network layer routing algorithm.
- Requires balancing performance gains, with advantages of modularity.

Networking Issues

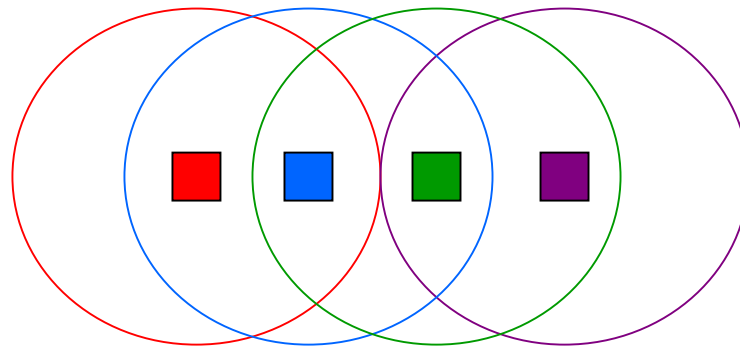
- MAC layer issues
- Topology control
- Routing

Medium Access Control

- MAC protocols are fundamental in any wireless network.
 - 30 + years of research
 - Many practical protocols (e.g. 802.11)
 - But not all are appropriate for sensor nets.
 - Due to energy/complexity concerns.

MAC fundamentals

- Basic problem – dividing a **shared** medium among **distributed** users.
 - In multi-hop networks complicated by **hidden/exposed terminals**.



Classes of MAC protocols

- Fixed assignment protocols
 - *Long-term* assignment of resource to users.
 - TDMA/FDMA/CDMA.
- Demand assignment protocols
 - Give all resources to one user based on demand.
 - E.g. polling, token rings.
- Random access protocols
 - Contention-based protocols
 - E.g. Aloha, CSMA-CA, RTS/CTS.

MAC Energy considerations

1. Collisions:

- Colliding packets waste transmission energy.

2. Overhearing:

- Receiving another node's packet uses receive energy.

3. Protocol asymmetry:

- Higher energy burden on “Leader” nodes.

4. Protocol overhead:

- Energy used to transmit/receive all overhead.
 - Includes RTS/CTS, synchronization, etc.

5. Idle listening:

- Listening in an idle state consumes energy.

Example MAC protocols

- Low duty cycle random access protocols.
- Schedule-based protocols
- 802.15.4 MAC revisited.

Low duty cycle protocols

Idea: Try to sleep as much as possible and only wake-up to transmit/receive packets.

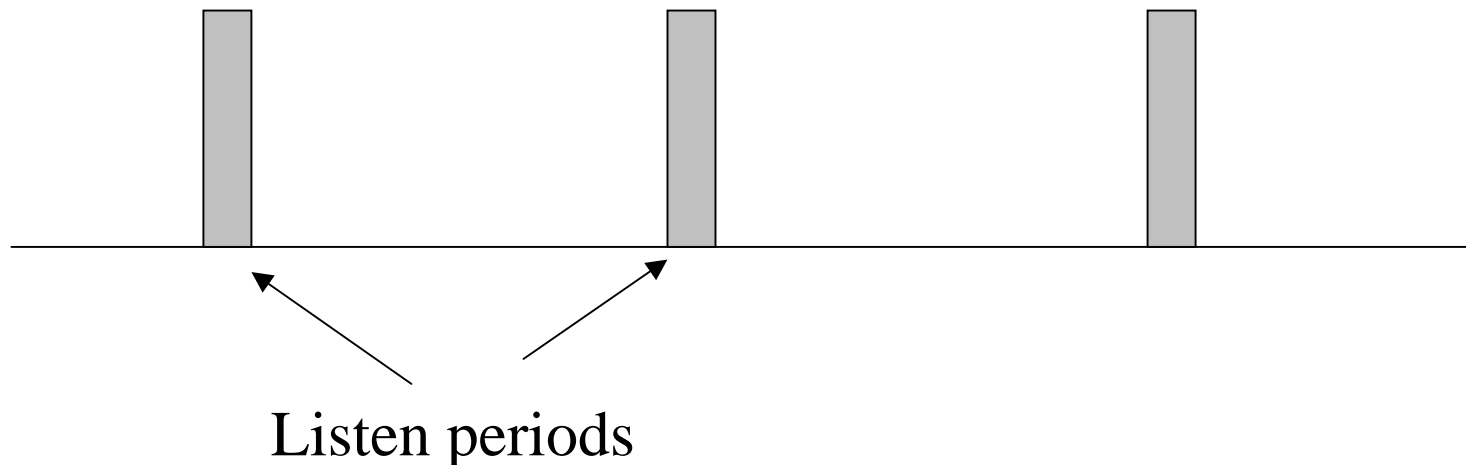
- save on idle listening/overhearing.

Basic problem: when a transmitter wakes up, how can it be sure that the receiver is also awake?

Main solution:

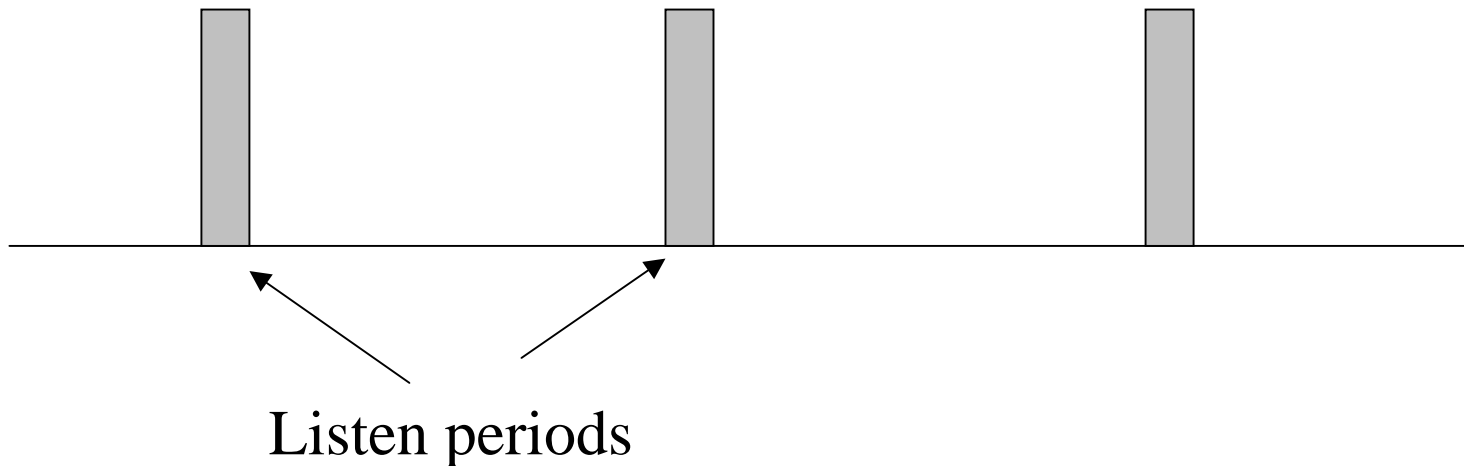
Periodic wake-up

- A Node wakes up for a listen period/if no packet then goes to sleep – otherwise, stays awake.
- **Wake-up period** = time from start of one listen period to start of next.
- **Duty-cycle** = listen period/ wake-up period.



Periodic wake-up

- How does a transmitter find a receiver's listen period?
- Size of duty-cycle/Sleep period?



Variations

- Use two channels
 - One for data/one for wake-up requests.
 - Wake-up channel radio may be lower power/performance.
- Synchronize all nodes to same sleep schedule (S-MAC).
 - Periodically send “synch” packets.
 - Need to account for clock drifts.
 - Nodes then use contention approach during listen period.
- Exploit asymmetries
 - have a “leader node” coordinate transmitters with receivers.
 - E.g. nodes notify leader when they have a packet to send, receivers check in with leader to see if anyone wants to talk to them.
 - Similar to Zigbee coordinator.

Wake-up radio

- An extreme version of the two-channel model.
- **Wake-up radio** monitors channel for signal to indicate when node is to wake up.
 - Very low power radio – e.g. pulse detector. No A/D, minimal processing.
- Another possibility is to use RF “trigger circuit” to wake-up
 - Similar to passive RFIDs.
- Can send little if any information in wake-up signal.
 - May wake-up wrong node.

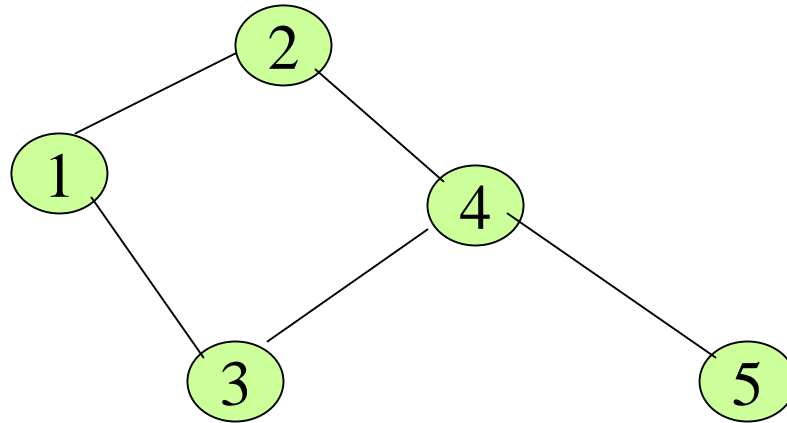
Scheduled MACs

Idea: Schedule all “links” so that they do **not interfere** with each other.

- Schedule nodes in TDMA manner.
- Avoids collisions.
- Nodes can sleep when not scheduled to receive.

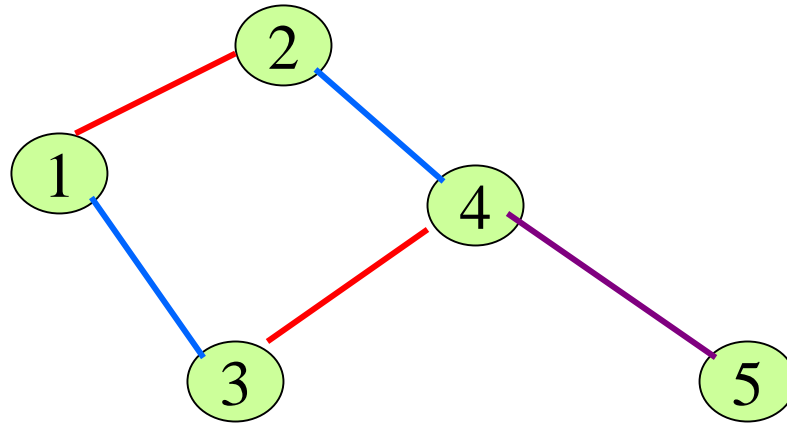
In multi-hop network, interference often *approximated* via an **interference graph**.

Interference graph



- Edge indicates nodes are “in range” with each other.
- Two “adjacent” edges interfere
 - Can not be scheduled at same time.
- Approximation?

Example Schedule



- Schedule using 6 time-slots (red/blue/purple).
 - Each edge corresponds to 2 time-slots/ one for each direction.
 - Alternative is to schedule transmitters (let transmitters send to any receiver in time-slot)

Scheduled MACs

- Issues:
 - Finding “optimal” schedules is often intractable.
 - Higher set-up overhead (signaling) than contention-based schemes.
 - For TDMA, clock drift is a concern.
 - Difficult to adapt schedules on fast-time scale.
 - Higher memory requirements per node.

Scheduled MACs

- Typical protocols involve some form of “neighbor discovery” and some information exchange to set-up schedule.
- Neighbor discovery:
 - send period “hello” packets with node ID.
 - add any received ID into neighbor table (soft state).

Schedule establishment

- **SMACS** (Self-organizing MAC for sensor networks)
 - Send neighbor a copy of available channels.
 - Neighbor responds with mutually available channel.
- **TRAMA** (Traffic-adaptive medium access)
 - Exchange 2-hop neighbors.
 - Also exchange list of receivers for packets in queue
 - Calculate schedule using global hash function of node ID's and time.

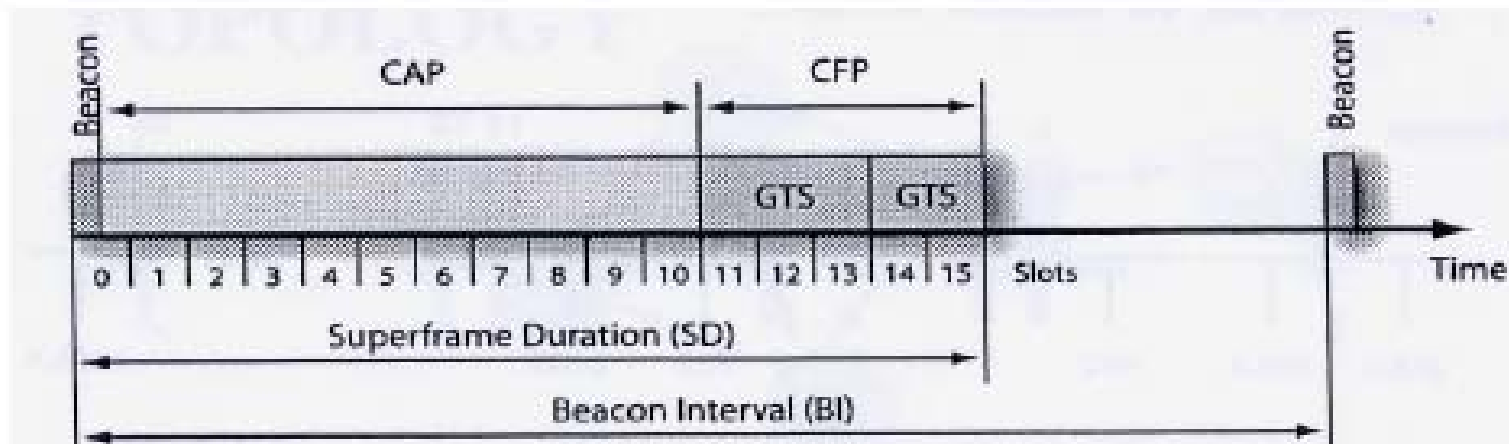
Variations

- Use combination of FDMA/CDMA with TDMA for schedule
 - Constraints depend on tunability of radios (FDMA)
 - Also “duplexing” constraints.
- Again a “leader” node can be used to establish schedules, esp. when leader has more capability.
- Can be combined with contention-based approaches (e.g. 802.15.4)

802.15.4 MAC revisited

- Recall, each network has exactly one **coordinator** device.
- Two modes for MAC:
 - **Beaconed** mode/**non-beaconed** mode.

802.15.4 Beacons mode (star topology)



- Time divided into superframes.
 - Each contains 1 active period & 1 inactive period.
- Each active period contains 16 time-slots.
- First time-slot contains beacon (sent by coordinator).
 - Remaining time-slots partitioned into contention access period (CAP) & upto 7 guaranteed time slots (GTS).

802.15.4 Beaconed mode

- Beacon includes superframe specification.
 - Lengths of various time-slots and overall frame are configurable.
 - Time-slots large enough to send small data packet as well as receive ACK
- GTS slots are allocated to particular device (by coordinator) – could be send or receive slots.
 - Devices request a GTS during the CAP.
- During CAP devices can use slotted CSMA-CA for either control or data packets.
 - Each time-slot further divided into smaller back-off periods.

802.15.4 Beaconed mode

- For the coordinator to send data to device – it either allocates device a receive GTS or advertises the device address in beacon.
 - In later case, device must “fetch” data in CAP.
- During the GTS phase only the device who is allocated that slot needs to be awake (and coordinator).
- During the CAP phase only only devices who have something to send (or a packet to fetch) need be awake.
- Device with no GTS need not wake-up even every superframe.

Multi-hop topologies

- In principle beacons mode can also be used in **multi-hop** topologies (cluster tree, mesh)
 - Need Zigbee routers also send out beacons.
 - Some work on doing this in a cluster tree – Zigbee routers try to “synch” beacons with coordinator.
 - One issue is how to avoid collisions with Beacons or Beacons with data frames.
- More common approach in multi-hop networks is to use **non-beaconed mode**.

802.15.4 Nonbeaconed mode

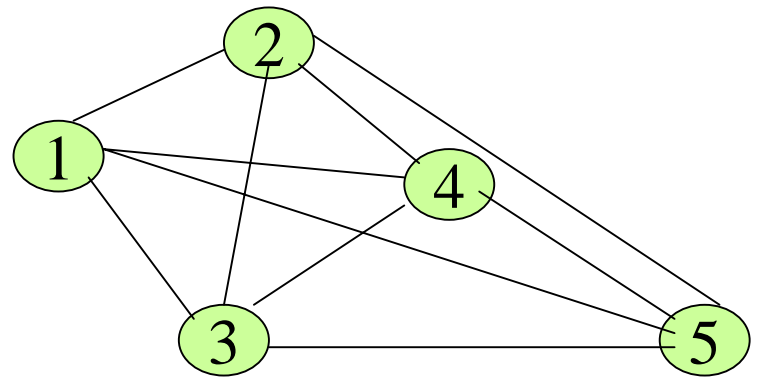
- No beacons/no GTS slots.
- Coordinators/Zigbee routers always on.
- All packets sent using **unslotted** CSMA-CA.
 - Why unslotted?

Networking Issues

- MAC layer issues
- Topology control
- Routing

Topology Control

- In principle in a wireless network every node can communicate with every other.
 - Fully connected topology



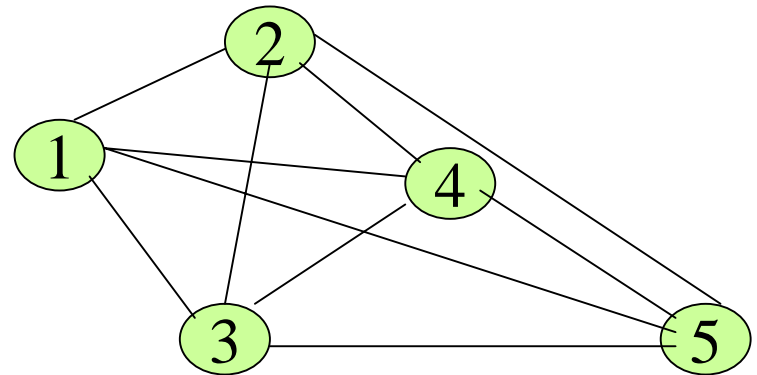
- Such a topology is often not very useful.
 - Why?

Topology Control

- **Topology control** refers to algorithms for restricting the network topology to overcome these difficulties.

- **Approaches:**

- power control.
- hierarchies (clustering).
- simply turning off nodes.



- **Main issues:**

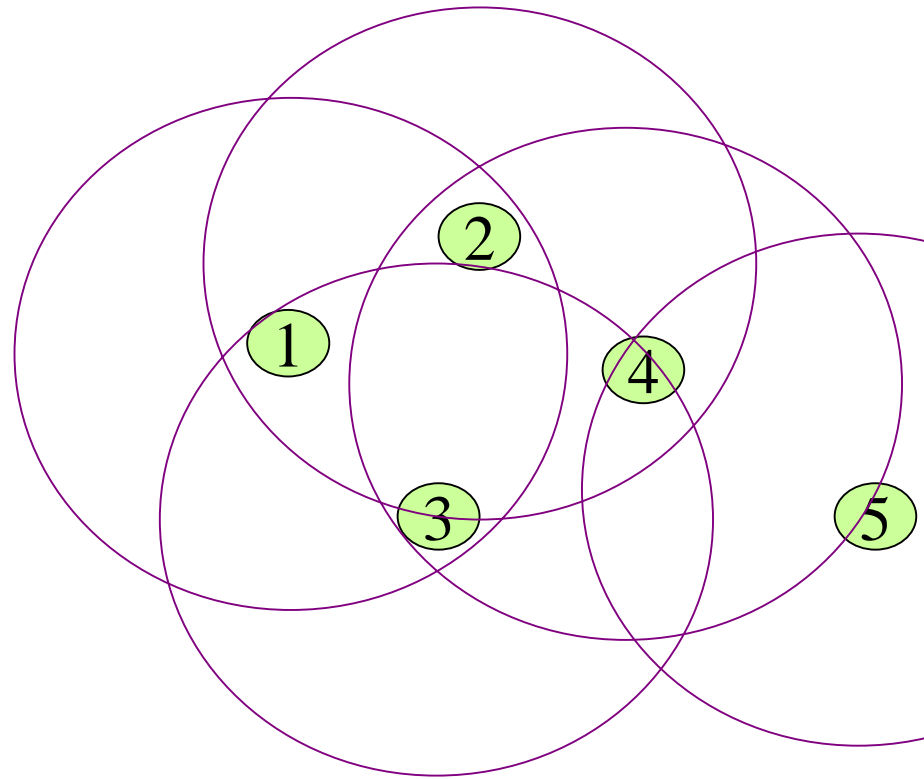
- Balancing connectivity/sensing performance with routing/MAC layer performance.

Topology Control

- The benefits of topology control will depend somewhat on the deployment scenario.
- Changing the topology will generally effect other protocols (e.g. MAC layer schedules or routing tables)
 - Therefore it should not be done on a fast time-scale.

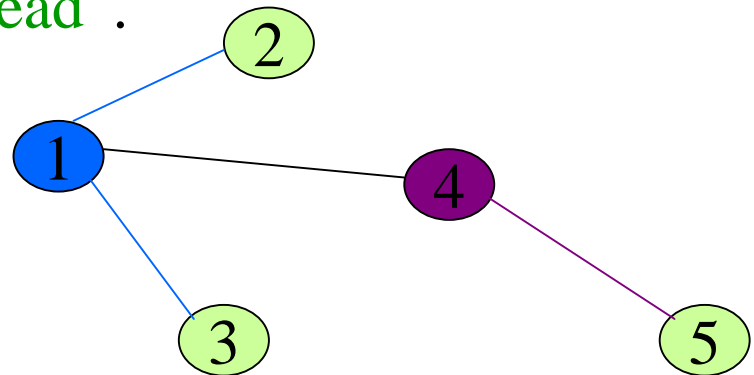
Power-based topology control

- For a given rate, transmission power determines neighbors.
- Decreasing power removes links.
- Simple approach is to vary power of “hello messages” until connected.



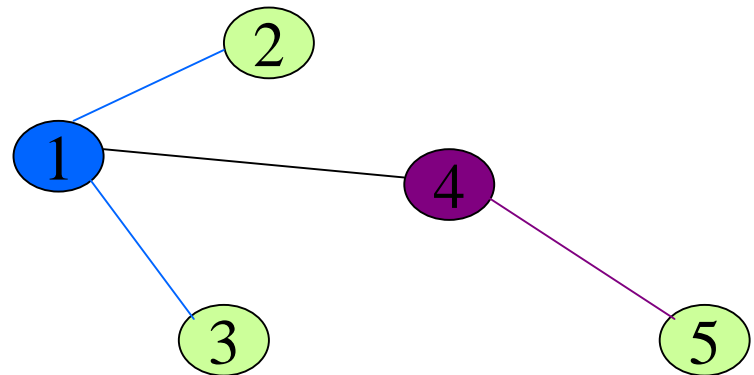
Hierarchical topology control

- Control topology by simply ignoring some neighbors.
- Which neighbors ignored specified by “hierarchy” among nodes.
- One example is via “**clustering**”
- Divide network into **clusters**.
- Each cluster has one “**clusterhead**”.



Hierarchical topology control

- Clusterheads manage resources for other nodes within clusters.
 - E.g. link assignments/ MAC schedules.
 - Performed by Zigbee routers in cluster tree topology.
- Lots of variations:
 - Overlapping clusters.
 - Inter-cluster communication.
 - Intra-cluster communication.
 - Static/dynamic cluster heads.
 - Multiple levels of clusters



Hierarchical topology control

- Establishing clusters
 - Choosing cluster-heads
 - Battery capacity/node capability
 - Choosing cluster assignments
 - Cluster size/ link quality

