CS4222 Wireless Network - Bennett Clement

Wireless Network History

- 1893: Tesla give a public demo of "wireless" radio communication
- 1906: AM Radio, first audio broadcast of music/voice from shore to ship
- Radio / Audio Broadcasting → 1916: regular weather broadcast using morse code, 1917 - 1919: clear transmission of voice, 1935: FM Radio, 1980s: Digital Audio Broadcasting (DAB)
- TV Broadcast → 1941: National TV System Committee (NTSC), 1962: Telstar Satellite, 1996: HDTV
- Voice communications → 1983: AMPS, 1989: GSM - Groupe Special Mobile (EUR), 1993: CDMA (US)

Evolution of Cellular Wireless Network

1G (AMPS)

Advanced Mobile Phone Service (AMPS) provides analog traffic channels

- two 25-MHz bands allocated, one each for uplink and downlink. Each band is divided into smaller channels of 30KHz. Data rate: 10kbps. Spectral Efficiency : 10kbps / 30kHz = 0.33 bps/Hz
- Designed for voice services, TDMA, analog signal, circuit-based.
- Small coverage area (1-16 miles)
- Neighboring base stations use different frequency to avoid interference.
- Low blocking (2%) during busy hour
- Low power mobile transmitters (<4W) and medium power base stations (10-100W)

2G (GSM)

Big Move from analog → digital :

- improves spectral efficiency, increase capacity by operating with smaller cells
- digital transmission allows for modern error detection and correction techniques
- provide robust network security tech based on encryption and secure key distribution stored in SIM (Subscriber Identity Module) card.
- Support roaming: settle how to charge between telcos.
- Support for non-voice services, e.g. SMS

GSM (Global System for Mobile Communications) - Europe

- Still makes use of BSS and Public Switched Telephone Network (PSTN)
- Has many channels
  - Broadcast channels (base-mobile) - TDM based: FCCH (Frequency Correction) → tell which frequency to use, SCH (Synchronization) → for time, BCCH (Broadcast Control)
  - Common control channels
    - base-mobile: PCH (Paging), AGCH (Access Grant)

    - mobile-base: RACH (Random Access) → if mobile wants to talk to base, use RACH, which uses ALOHA. Base station will then prepare resources for dedicated control channels and send ACK on AGCH
  - Dedicated control channels (2-way): SDCCH (Standalone), SACCH (Slow Associated Control), FACCH (Fast Associated Control)
- Has Full Rate traffic channels or Half-Rate with FEC. Slower rate increase reliability
- **GSM Short Message Service (SMS)** provides a connectionless transfer of messages with low capacity (up to 140 octets, because it uses control channel, i.e. GSM SDCCH)

We also have transitional technologies (2.5G) : GPRS/EDGE

3G (CDMA-based)

Big Move: circuit-based → packet based

- Provides high-speed wireless communication for multimedia
  - Voice: quality comparable to PSTN
  - Data: 144kbps (high mobility) - 2Mbps (low mobility)

We have transitional technologies: HSPA (3.5G)

4G

- Target peak data rates of up to 100Mbps (high mobility) - 1Gbps (low mobility)
- CDMA spread spectrum technology used in 3G is replaced by frequency-domain equalization schemes, e.g. multi-carrier transmission such as OFDMA
- Include other techniques such as MIMO (multiple antennas: Multiple In Multiple Out), dynamic channel allocation and channel-dependent scheduling (for fairness, as channel state is always changing), carrier aggregation (using non-consecutive spectrum for higher bandwidth)

Move into LTE (Long Term Evolution). Each year / 2 years have a new release.

5G

Try to achieve throughput (for data applications e.g. VR), scalability (thousands of sensors, IoT) and reliability and low latency (for self-driving car).

More work is done on the cloud server, more programmable parts of the network.

Wireless Transmission Basics

Propagation Modes

- Ground Wave Propagation : f < 2MHz (medium-wave AM Radio 520kHz - 1,6MHz)
  - Follows contours of the earth. low bitrate.
- Sky Wave Propagation: 3 - 30 MHz (Short-wave AM Radio)
  - Signal "reflected" between ionized layer of atmosphere and earth's surface back and forth. low bitrate due to noise.

- Line-of-Sight Propagation: f > 30 MHz. (signals w/ this freq is not reflected by ionosphere)
  - transmitting and receiving antennas within line of sight.
  - Not really line-of-sight, because there might be obstacles, e.g. high-rise buildings.

Line-Of-Sight Propagation

Maximum distance (in km) between 2 antennas for LOS propagation: $3.57(\sqrt{Kh_1}+\sqrt{Kh_2})$ where K is an adjustment factor to account for refraction (defaults to 4/3) and h_i are height of the antennas (in m).

Gain $G=\frac{4\pi A_e}{\lambda^2}$ where $\lambda=\frac{c}{f}$ and $A_e=\pi r^2$ (in meters).

Watt to decibels: $10\log_{10}(P)$. Path Loss in dB = $10\log(P_t/P_r)$.

Free Space Propagation Model.

$$P_r=G_rG_t\left(\frac{c}{4\pi f_c d}\right)^{\alpha}P_t$$

where P are the power in which the signal is received/transmitted, f_c is the center frequency, c : speed of light, d: distance between transmitter and receiver, α (path loss component): a function of environment (defaults to 2, may increase due to obstacles or decrease due to constructive refraction), G is the antenna gain.

An efficient antenna (half-wave dipole antenna) has length half of the signal wavelength its transmitting.

Observations:

- transmitting using a higher frequency results in lower power, but it can be compensated with a higher gain.
- Atmospheric absorption: water vapour and oxygen absorbs, rain and fog scatter
- longer antenna → lower frequency → longer range, better obstacle penetration
- higher frequency → higher bandwidth

Mobile Radio Propagation

Challenges:

- attenuation: longer distance, weaker signals
- blockage
- interference from another transmitter
- multipath propagation → receiver receives multiple copies of same signal.
  - reflection, e.g. earth surface, buildings
  - diffraction: wave bends around obstacle
  - scattering: obstacles smaller than wavelength (e.g. foliage) scatter signal into several weaker signals
- Fading: variation of received signal power over time caused by changes in the transmission medium / paths.

- Large scale: environment changes, path loss
- Small scale: Doppler spread due to movement of users and obstacles, multipath fading (different combination of delayed signal arrivals)
  - Flat Fading → constant gain, decrease in SNR, seen in narrowband channels. Happens when channel bandwidth should be greater than the bandwidth of the transmitted signal, and period > multipath delay spread.
  - Selective Fading → happens when bandwidth is smaller than the bandwidth of the transmitted signal, and period < multipath delay spread.

Effects of Multipath propagation:

- Intersymbol interference (ISI): delayed copies of a pulse arrive at the same time as primary pulse for the next bit.
- Destructive out-of-phase arrival of copies → more difficult detection

Electromagnetic Spectrum

- (in decreasing frequency): Visible Light : ~10^{15} Hz, 400-700 nm, Infrared , Microwave: ~ $10^9 - 10^{11}$ Hz
- Radio:
  - AM (Medium Wave): 500 - 1600kHz with 9-10 kHz spacing
  - FM: 87.5 MHz - 108MHz, channel spacing > 100KHz
  - TV (UHF): 470 - 806 MHz, 6 MHz spacing
  - Data Transmission 300MHz - 6GHz, e.g. WiFi (2.4GHz, 5GHz)
  - Cellphone: ~700MHz - 2GHz.

Shannon Capacity

$C = B \times \log_2(1 + SNR)$ where C is the theoretical capacity of the channel (max bits sent or data rate), B is the available bandwidth, i.e. channel spacing and SNR is the signal-to-noise ratio (quality of the channel, dependent on P_r as noise is out of our control)

Combining with the Free Space Propagation Model, we have the following observations:

- Power is usually a constraint
- To increase data rate C, we can increase throughput by using
  - a larger bandwidth
  - higher frequencies → tradeoff: range decreases
  - Improve design to increase gain G, e.g. OFDM, MIMO

ISM Band

The common ISM band are:

- 433 MHz : support ~1Kb bitrate. only suitable for disaster recovery
- 900MHz: support up to 19.2Kb. suitable for most long-range sensors. Can reach several kilometres.
- 2.4GHz: WiFi, bluetooth, other appliances
- 5GHz: very high bitrate ~1Gbps but short range.

Wireless Network Characteristics

- Range: longer range needs lower frequency
- Data Rate: higher data rate needs higher frequency → consumes more power to compensate range usually.
- Power
- Frequency Band

Addendum

Modulation vs Multiplexing vs Multiple Access

- Modulation: alter original (modulating / base band) signal to use a higher frequency (carrier frequency) for transmission. The resulting signal is called the modulated / band pass signal.
  - Require additional hardware (modulator, demodulator)
  - Example: AM, FM, QPSK, PCM, QAM
- Multiplexing: Transmit more than 1 source of signal together.
  - Time Domain (our frame will contain some packets form source A and some packets from source B and so on), Frequency Domain (OFDM)
  - In multiplexing we do not provide a dedicated resource to a single source.
  - Require additional hardware (multiplexer, demultiplexer) to combine and separate signals from different sources.
- Multiple Access : multiple users access the channel or link
  - Don't require additional hardware.
  - Example: TDMA, FDMA, CDMA, OFDMA

Sensor Networks

802.11 → WiFi, 802.15.4 → ZigBee

- Very small nodes → extremely low computation power, low bit-rate
- Design to execute a specific task and densely deployed (highly redundant) in large quantities.
- Tasks are a combination of sensing, computation and communication e.g. detection, classification, tracking
- Motivation: used in places where placing wires is expensive / cumbersome / not possible → early uses: wildlife tracker

IoT

- interconnection of uniquely identifiable embedded computing devices within the existing internet infrastructure (Wikipedia)
- Examples: personal devices (smart watch, TraceTogether), home automation (smart light bulb, smart power meter, Google Nest, security systems etc)

Sensors:

- 9-axis IMU Sensor: Gyroscope (detect rotation about X,Y,Z axis in deg per second), Accelerometer (output for 3 axis in g), Magnetometer (detect terrestrial magnetism in 3 axis) → used for localization, dead reckoning[1], motion detection
- Ambient Light Sensor: measures intensity of visible light (in lux) → used for backlight control, home automation, cameras etc.
- Altimeter / Pressure sensor: mesasures barometric pressure in hPA → used in weather forecast, navigation, enhance GPS

Energy Consumption

- Voltage (Volts), Current (Ampere). Power (W) = volts (V) x amps (A). Energy (J) = Power x Time
- Mobile nodes are battery powered → energy constraint is fundamental
- Energy is used by network, screen / monitor, computation, sensing

Key Idea:

Devices **rarely** transmit continuously → save energy by putting network interfaces into **lower power states** when they are not active (duty cycling). However, state transition is sometimes slow due to driver / software overhead.

Desirable properties for low-powered radio: very quick state transitions

How to know when and how often to wakeup → device waits and listens for more incoming packets → cellular "tail" behaviour → very inefficient for infrequent transfer of small amount of data.

Energy savings mechanisms:

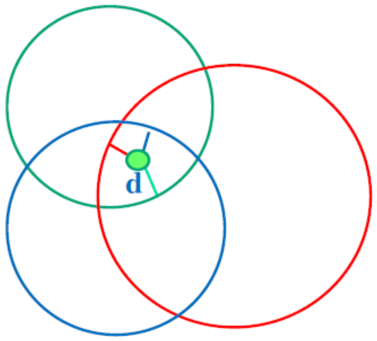
- reduce sampling frequency
- use low power sensors whenever possible
- offload to hardware (sensor batching)

Duty cycle is the ratio of time when the device is ON compared to the time it is OFF.

Localization

Localization: to determine the physical coordinates of a group of sensor nodes. Coordinates can be global, e.g. GPS, or relative, i.e. some "rigid transformation" (rotation, reflection, translation) away from global coordinate systems.

Multilateration



- Basic Idea: Trilateration using beacon / anchor / landmark.
- We need at least dimension + 1 beacons.
- Beacon placement improves accuracy if placed in a convex hull.

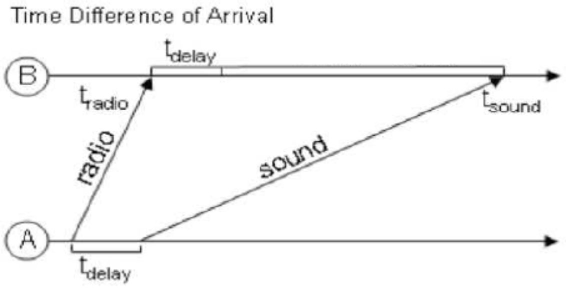
If there is no point (x, y) that satisfies all constraints, then we try to minimize the estimation error $\sum_{i=1}^{m}(d_{ij} - \hat{d_{ij}})^2$, i.e. solving the linear equation $-2x_ix - 2y_iy + R = d_i^2 - (x_i^2 + y_i^2)$.

We solve $A^TA\hat{x} = A^Tb$ where $A = [-2x_i, -2y_i, 1], x = [x; y; R], b = [d_i^2 - (x_i^2 + y_i^2)]$

Time of Arrival

Know speed of signal propagation → calculate distance from clock difference
Clock accuracy needed for meter accuracy is $1/v$ → it is much easier to use speed of sound than light (3ns vs 3ms clock accuracy)

Time Difference of Arrival (TDoA)



- Exploit differences in speed, say RF and audio.
- Works well in Line-of-Sight environment
- Example: Cricket
  - Transmitter sends a radio msg, (optionally) waits t_{delay} and sends audio "chirp" sound.
  - Listener hears the radio signal and record time t_{radio}.
  - Listener hears the audio signal and record time t_{sound}
  - $t_{\text{sound}} - t_{\text{radio}} - t_{\text{delay}} = d/v_{\text{sound}} - d/v_{\text{radio}}$.

GPS

A set of satellites orbiting the Earth. Each has very precise clock.

Equation form by GPS receiver for satellite i: $\Delta t = e + d_i/c$ where e is the clock drift (between receiver and satellite), Δt is the difference between local time of msg reception and the timestamp of statllite msg, and c the speed of light. We can express d_i as $\sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2}$ → minimum 4 satellite readings.

RSSI

- Received Signal-Strength Indication → In theory, RSSI ~ $1/d^\alpha$
- In practice, *RSSI measurements contain noise and is highly non-uniform* due to (1) different radio propagation over different surface and (2) obstacles. One common assumption: closer nodes have stronger RSSI than those further.

Wifi Fingerprinting

- Data collection: Measure signal strength (SS) and SNR in each location. Record (time, x, y, direction). Compute mean over 20 samples per beacon.
- Usage: Compute the average of real-time SS for all beacons for mobile host. Compare SS vectors to database acquired during data collection → find SS that minimizes error to obtain the most probable location

Mobility

Overview

Mobility is the pattern on which devices move.

- No mobility: using same access point
- Intermediate mobility: connecting / disconnecting from network using DHCP
- high mobility: passing through multiple AP while maintaining ongoing connections

Problems:

1. How do you locate the mobile device?
2. If communication is in progress, how do you route packets to the mobile node?

Vocabulary:

- Home network: permanent home of mobile
- Home agent: entity in home network that will perform mobility functions on behalf of mobile when mobile is remote
- Permanent address: address in home network (would not change even if mobile is remote). Can always be used to reach mobile
- Visited network: network in which mobile currently resides
- Foreign agent: entity in visited network that performs mobility functions on behalf of mobile
- Care-of-address: address in visited network
- Correspondent: others who want to communicate with mobile

Approaches

- let routing handle it: routers advertise permanent address of mobile-nodes-in-transit via usual routing table exchange → not scalable
- let end-systems handle it:
  - indirect routing: every action goes through home agent
  - direct routing: correspondent communicate directly with mobile

Overview on how correspondent can communicate with mobile:

- Ask the home network where the mobile node is
- Use direct / indirect routing to communicate with mobile node.

Registration

For the home network to know where the mobile node is,

1. mobile contacts foreign agent on entering visited network
2. foreign agent contacts home agent: "this mobile is resident in my network"

Indirect Routing

1. Correspondent addresses packets to home address of mobile
2. Home agent intercepts and forwards packets to foreign agent (via care-of addr)
3. Foreign agent receives and forwards packets to mobile
4. Mobile replies directly to correspondent

Observation:

- Mobile location is transparent to correspondent
- Triangle routing (correspondent - home net - mobile) is inefficient

Suppose mobile user moves to another network:

- rerun the registration process
- home agent updates care-of-address for mobile and packets continue to be forwarded (but with new care-of-address)

Direct Routing

1. Correspondent requests and receives foreign address of mobile
2. Correspondent sends packets to foreign address of mobile
3. Foreign agent receives and forwards packets to mobile
4. Mobile replies directly to correspondent

Observation:

- overcome triangle routing problem
- Mobile location is not-transparent to correspondent

Suppose mobile user moves to another network:

- Data always routed first to anchor FA (FA in first visited network)
- When mobile moves, new FA arranges to have data forwarded from old FA (chaining)

on the Internet

Node Identification: DHCP to bootstrap (get IP & DNS), then use IP. But IPv4 are hard to use for mobile node as it is tied to a location.

Mobile IP (RFC 3344): has many features discussed above

- indirect routing of datagrams
- agent discovery
- registration with home agent

in Cellular Network

Node Identification: Phone number, somewhat tied to location, e.g. country code

Handling mobility in cellular networks is quite similar to the ones on the internet.

Vocabulary:

- Home network: network of cellular provider you subscribe to (Singtel, Starhub)
- HLR (Home location Register): database in home network containing permanent cellphone number, profile information, current location info, etc.
- VLR (Visitor Location Register): database with entry for each user currently in the network.
- MSC (Mobile Switching Center), BSS (Base Station Subsystem)

GSM: indirect routing to mobile

1. Call from correspondent routed to home network
2. Home MSC consults HLR, gets roaming number of mobile in visited network
3. Home MSC sets up second part of call to MSC in visited network
4. MSC in visited network completes call through base station to mobile

Paging

How do MSC knows which base stations the mobile is associated to?

- When the device moves "far enough" (out of a **paging area**), inform the base station → infrequent communication → save energy
- With (predefined) paging areas, MSC only needs to search all base station in the paging area.

Handoff

Goal: route call via new base station without interruption → stronger signal (continuing connectivity, less battery drain), load balance

With common MSC

1. old BSS informs MSC of impending handoff, provides list of >=1 new BSSes
2. MSC sets up path (allocates resources) to new BSS
3. New BSS allocates radio channel for use by mobile
4. New BSS signals MSC and old BSS: "ready"
5. Old BSS tells mobile: perform handoff to new BSS
6. Mobile signals to new BSS to activate new channel
7. Mobile signals via new BSS to MSC: handoff complete → MSC reroutes call
8. MSC - old BSS resources are released.

Between MSCs

Similar to direct routing, we define anchor MSC: first MSC visited during call. Then, we add new MSCs on to the end of MSC chain as mobile moves to new MSC.

Neighbor Discovery

Scenarios: detecting WiFi networks, sensors communicating with other sensors (in an energy efficient way)

Comment: Problem is easy to solve if there is no constraint and all nodes can be active all the time. Challenge → energy efficient protocol

Node A **discovers** node B when it hears a transmission from node B.

Synchronous Protocols

Synchronous protocols assume that all nodes are synchronized to the same clock/time.

Symmetric Nodes

- Nodes agree to wakeup periodically at the same time to transmit / receive packets.
- Assume time is divided into slots of 100 ms; slots cannot be too short because:
  - issue 1: clock drift
  - issue 2: amount of work to transmit & receive. how many nodes are we gonna transmit to?
  - issue 3: need time to do state transition (wake up / transmit / recv)

Drawbacks:

- synchronization is expensive, and periodic re-sync is needed to correct clock drift.
- If too many nodes wakeup (and transmit) simultaneously → increased collision

Asymmetric Nodes

In some topology, we can exploit the fact that some nodes are more powerful and is always "ON", e.g. WiFi APs (always ON) vs mobile device (stays in OFF state as much as possible)

Discovering nearby APs → since APs are always on, mobile device broadcasts a probe beacon and listens for probe replies. After a period of several cycles, device would have gotten result from all nearby APs

Clock synchronization → contact AP

WiFi Power Save Mode (PSM)

A mobile device will try to sleep as much as possible, but how do the device know when there is data from the AP if it is in the sleep state?

- When mobile goes into PSM, informs the AP
- Any packets arriving at the AP for the mobile in PSM will be buffered
- AP sends a beacon frame after every fixed time interval with a field called **traffic indication map (TIM)** (TIM for unicast, DTIM for broadcast packets) to inform the mobile in PSM of incoming packets
- Mobile in PSM wakes up periodically to listen to beacon frames and switch to active state if there are incoming packets

Asynchronous Protocols

Asynchronous protocols don't require clock to be time synchronized.

In general, time is still divided into time slots (e.g. 100ms), and each device decides which time slots to wakeup for transmission / listening

Birthday Protocol

Birthday paradox. Probability that there is at least one pair of people who have the same birthday: $1 - \frac{365!}{365^n(365-n)!}$

Nodes randomly choose to wakeup k time slots out of a total of n slots. What is the probability that 2 nodes wakeup at the same time slot at least once? $1 - \frac{\binom{n-k}{k}}{\binom{n}{k}}$

- easy to implement, good avg discovery latency (use bigger k and n to achieve same duty cycle), but unbounded worst case.

Deterministic Bound Protocols

The following, we discuss several protocols with deterministic worst-case bound. Delay bound is $\theta\left(\frac{T}{f^2}\right)$ where T is the time slot length and f is the duty cycle.

Quorum-based Protocol

1 Period is divided into n^2 slots ($n \times n$ grid). Each node chooses 1 row and 1 column to wakeup, i.e. $2n - 1$ active slots. Duty cycle = $\frac{2n-1}{n^2}$. Worst case latency is $\sim n^2$.

This protocol guarantees that within 1 period, 2 nodes will meet each other twice.

Limitations: Nodes all have same duty cycle.

DISCO Protocol

A node picks 2 prime numbers p and q with $p \neq q$. Duty cycle is $\frac{1}{p} + \frac{1}{q}$. Worst case latency is pq.

By choosing different primes, nodes can have different duty cycle.

Wireless Media Access Control (MAC)

Motivation: Multiple stations want to access a common channel. How to coordinate access?

The following protocols are classic MAC, i.e. not accounting for energy constraints.

Random-Access Protocols

These protocols try to reduce collision and "manage" retransmission

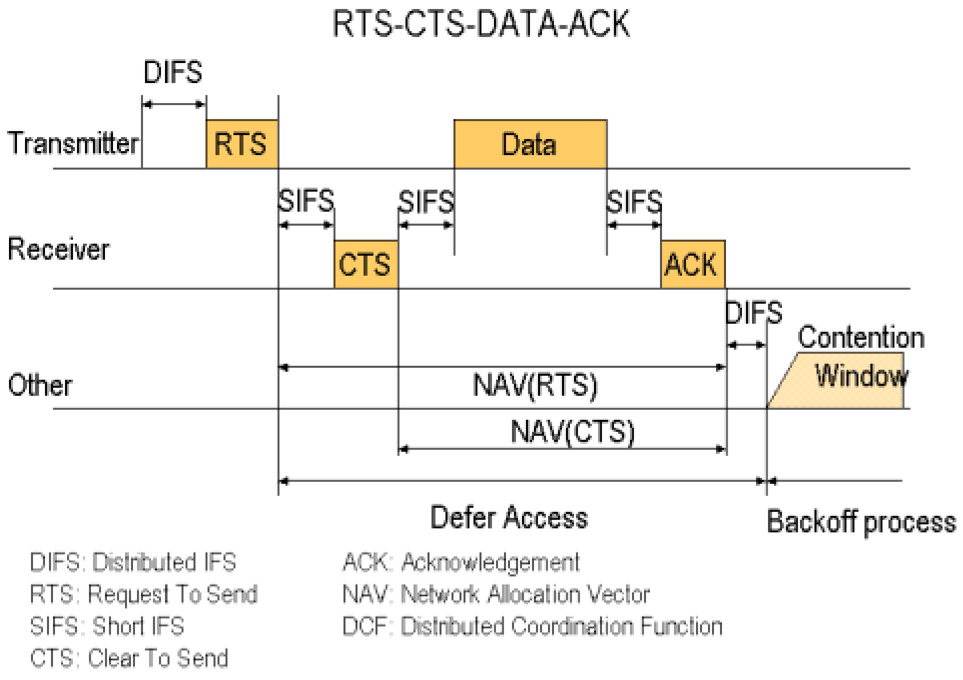
ALOHA

- No Coordination: Node immediately sends whenever it needs to. If it doesn't hear an ACK within a given duration, it resends after waiting a random (to avoid thundering herd) amount of time.
- Achieves low delay if load is light
- Let X be the packet transmission time and G be the average number of new or retransmitted packets in X seconds. Vulnerable period (when its signal can be interfered by others) of a transmission is $2X$. Model the number of total arrivals as Poisson process. The throughput S is given by $G \times \frac{(2G)^0}{0!}e^{-2G} = Ge^{-2G}$ → maximum throughput is $\frac{1}{2e}$.

Slotted ALOHA

- Performance of ALOHA can be improved by allowing transmission only at start of fixed intervals.
- Vulnerable period is now X as competition to send only occurs at the start of each slot → maximum throughput is $\frac{1}{e}$.
- Limitations: transmission is independent of channel state (busy / idle)

CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance)



Sense carrier, if idle → send, wait for ACK, if there isn't one assume a collision and retransmit.

1. If the channel is clear for DIFS, it defers transmission for a random interval
2. To solve hidden node problem,
    1. Before transmitting data, the transmitter first sent Request-To-Send (RTS).
    2. After SIFS, the receiving station responds with a Clear-To-Send (CTS).
    3. Stations that hear RTS/CTS will adjust their local Network Allocation Vector (NAV): time that must elapse before a station can sample channel again for idle status.
3. After SIFS from CTS, the transmitter start sending data packet
4. After SIFS from receiving data packet, the receiver sends ACK back to transmitter.

Inter-Frame Spaces has different length to allow priority of packets, e.g. CTS/ ACK are prioritized compared to RTS.

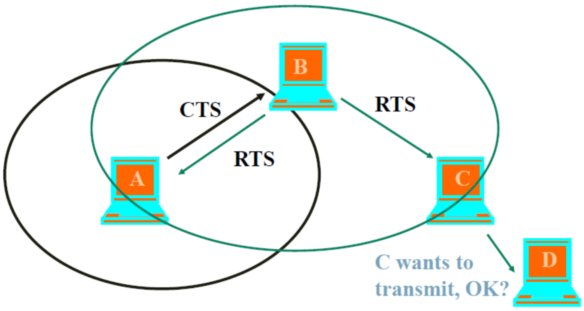
- DIFS: DCF (Distributed Coordinated Function) IFS. Longer than SIFS
- SIFS: Short IFS.

Note: WiFi defaults to CSMA/CA for data transmission. This is to minimize overhead. However, WiFi has another mode - "polling" a.k.a. PCF that's better to suit voice transmission. Switching between these 2 modes is left to the AP to decide.

Sharing/Allocation "Problems"

Capture Effect is more of a phenomenon.

Hidden Terminal Problem



Two stations that are not within range, but both simultaneously send a packet to a station that is located between them → transmissions overlap at the receiving node → data gets lost. However, both transmitters do not detect the collision and don’t start a new delivery attempt.

RTS/CTS solves the hidden terminal problem, but it adds overhead and creates the exposed terminal problem.

Exposed Terminal Problem

Upon receiving RTS / CTS, all nodes within range will stop from communicating except the pair. This will block anyone from communicating with these other nodes, even though communication might not interfere with the original pair's communication.

Rate Asymmetry

Achievable wireless transmission rate depends on (1) channel quality and (2) technology used.

Notion of fairness

In standard CSMA/CA, nodes send (almost) same number of packets on average (fair). However, with heterogeneous nodes, this drags down the average transmission rate by a lot → System is fair, but efficiency is low.

To improve efficiency, we can define fairness as: each node allocated the same transmission duration.

Capture Effect

Collision does not always lead to packet loss. If a receiver hears from 2 or more stations and the signal from 1 station is much stronger than the rest, the receiver can receive the signal correctly "only" from this station → capture effect.

With capture effect, adding new transmitters of the same packet may improve reliability.

Controlled-Access Protocols

In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations.
Thus, these protocols avoid collision.

Reservation

A station needs to make a reservation before sending data → system is more efficient if many data frames are sent after one reservation.

Example:

- Time is divided into reservation and data slots
- Reservation uses slotted ALOHA. Stations that want to send should broadcast their request during reservation slots.
- If there are stations that want to send, reservation slots will be followed by data slots. Number of data slots vary according to the number of stn that needs to send.

Polling

One primary, e.g. AP, and many secondary stations, e.g. end nodes.

- Select: primary station sends to a secondary station
- Poll: primary station ask (round-robin) if a secondary station has data to send

Advantages: can be used in time-sensitive applications, e.g. VoIP.
Limitation: Cannot have too many end nodes, otherwise it takes to long to get polled.

Token Passing

Pass a token around. Nodes can only send if it holds the token

Example topology: physical ring, dual ring (2-way ring), bus ring, star ring.

Channelization Protocols

These protocols "statically" allocate resources.

Pros: Resource always available, performance guarantee, no interference
Cons: Inefficient if resource is not utilized, slow to adjust to traffic demand changes.

FDMA (Frequency Division Multiple Access)

Similar to broadcast radio and TV, assign a different carrier frequency for diff nodes. Guard channels are used to make sure adjacent frequency spectrum don't interfere.

OFDMA (Orthogonal Frequency Division Multiple Access)

OFDMA is a variant of FDMA.

- In FDMA, the entire frequency spectrum is dedicated to carry data from a single source.
- In OFDMA, data is divided into many (slower) bit streams and multiple subcarriers (each with a smaller spectrum) is used. There is no guard channels in OFDMA. Instead, channels are orthogonal to one another, i.e. the peak of a bit stream happens at the null of other bit streams.

- OFDMA is robust to selective fading and overcomes inter-symbol interference in a multi-path environment because data rate per subcarrier is lower.

Note on OFDM vs OFDMA: In OFDM systems, only a single user can transmit on all of the sub-carriers at any given time. In order to support multiple users time and/or frequency division access techniques are used in OFDM. The major setback to this static multiple access scheme is the fact that the different users see the wireless channel differently is not being utilized. OFDMA, on the other hand, allows multiple users to transmit simultaneously on the different sub-carriers per OFDM symbol.

TDMA (Time Division Multiple Access)

- Assign a time slot to diff nodes. Data transmission can be done on multiple freq.
- User buffers data on the sender side. It sends data at an accelerated rate (as it can use the whole channel) when its time comes. Data is then stored at receiver and played back at original (slower) rate.
- Good for **frequent**, 2-way communication.

FDMA and TDMA can be combined to fit changing needs of users and spread out bad frequencies

CDMA (Code Division Multiple Access)

A multiplexing technique use with spread spectrum.

Spread Spectrum

- methods to deliberately spread a signal in the frequency band resulting in a signal with wider bandwidth
- Used to establish secure communications, increase resistance to interference, jamming and interception (signals are difficult to interpret, similar to background noise)

Code Design

- Synchronous CDMA: Used to send from BS to mobile. Use orthgonal codes
- Asynchronous CDMA: Used to send from mobile to BS. Use Pseudorandom noise (PN) to encode the bits.

Others

- Sharing of channel when there is two-way communications (duplex)
  - FDD (Frequency Division Duplex): different freq for uplink and downlink. Used in cellular systems: AMPS, CDMA, GSM
  - TDD (Time Division Duplex): Used in cordless telephone systems

Energy Efficient MAC

Difference from traditional MAC → most sensor network don't have many stuff to send (low channel utilization), although we still want fairness, low latency.

Usually, to reduce latency and improve channel utilization, nodes sense continuously. If energy is constrained → higher latency.

The main sources of energy consumption in any contention-based MAC protocol:

1. overhearing: get packets that is not addressed to the node → because wireless is broadcast in nature
2. control packet overhead: e.g. RTS/CTS
3. data transmission / reception
4. idle listening: just listening to the channel without any traffic.

In a sensor network, idle listening (and as an extension, overhearing) consumes most energy as traffic load decreases. Since we want to be as energy-efficient as possible, we need to do **Duty Cycle MAC**.

Challenge: How to coordinate the wakeup schedule, i.e. if A wants to send to B, how does B wake up at the right time?

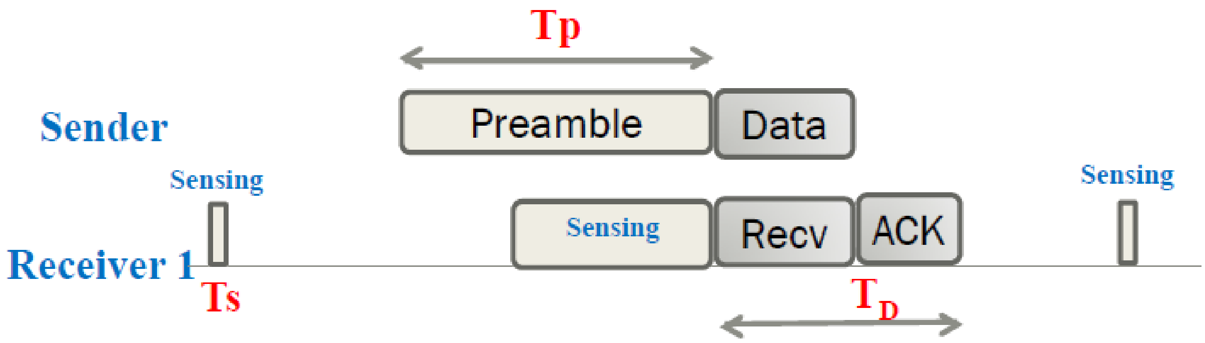
If nodes are asymmetric, an example is WiFi PSM. We now discuss several energy-efficient MAC protocols for symmetric energy-constrained nodes.

Asynchronous MAC Protocols

B-MAC

Does the simplest MAC protocol, ALOHA works?

- since there is low channel utilization → no contention (great!)
- requires receiver to be listening (not always the case with duty cycle)



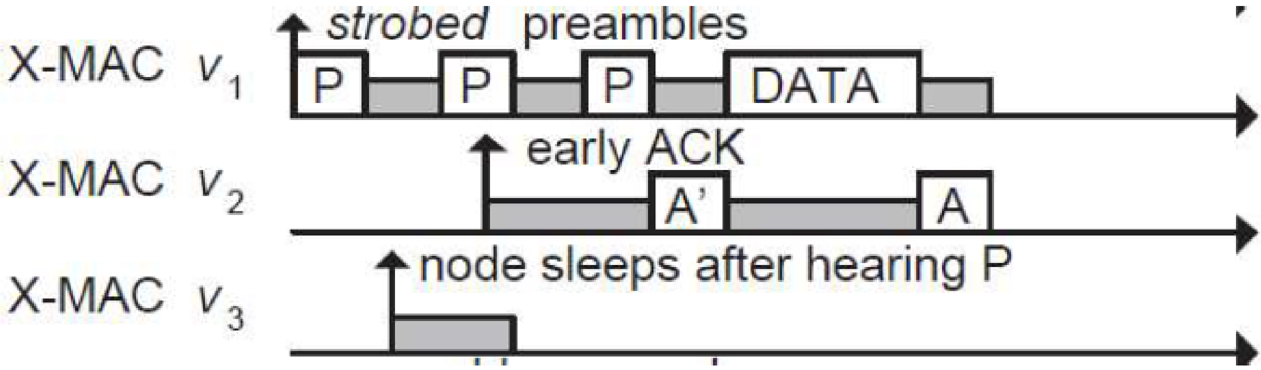
To tackle the second problem, we add a preamble to the sender prior to sending.

- Each receiver periodically (every T_p) wakes up to check if channel is busy for a duration of T_s. ($T_p > T_s$)
  - If channel is busy, stays in listening mode until destination of data is known.
  - If it is not for the node, goes back to sleep until the next period.
- Sender transmits a long preamble for T_p and then send data packet at the end of the preamble.
  - The preamble guarantees that the intended receiver will have woken up when data packet is sent. (Side effect: all nodes are waken up)

Energy consumption:

- When there is transmission: Transmitter: $T_p + T_d$, Receiver: $0.5\,T_p(+T_d)$.
- When there is no transmission: T_s.
- Latency to send a packet is: $T_p + T_d$ (not great) and max. utilization is $T_d/(T_p + T_d)$.

X-MAC



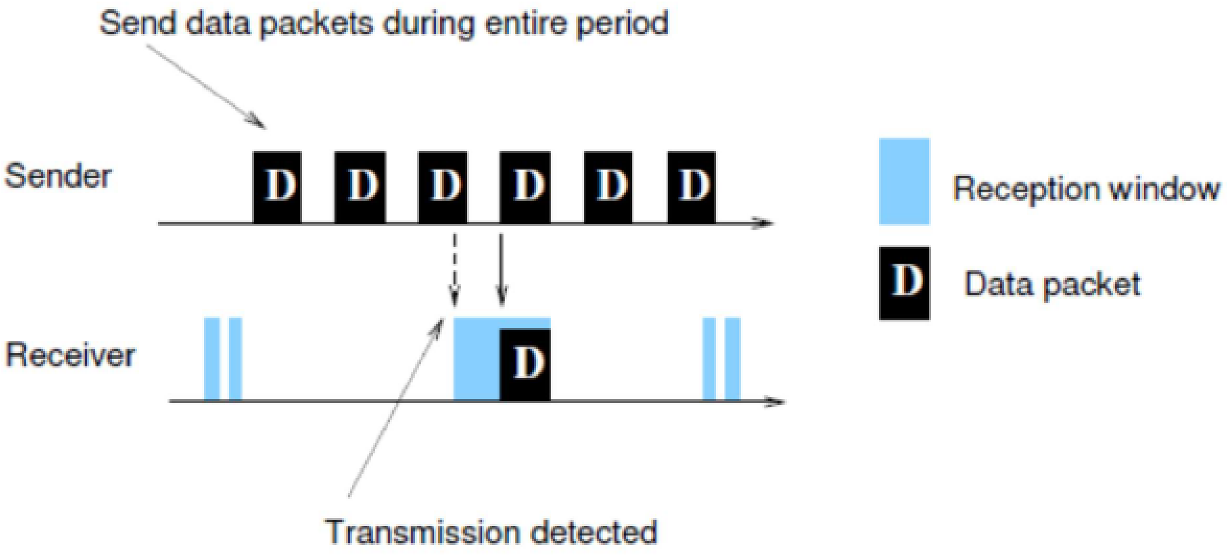
X-MAC improves over B-MAC: There is no need to wake up all nodes!

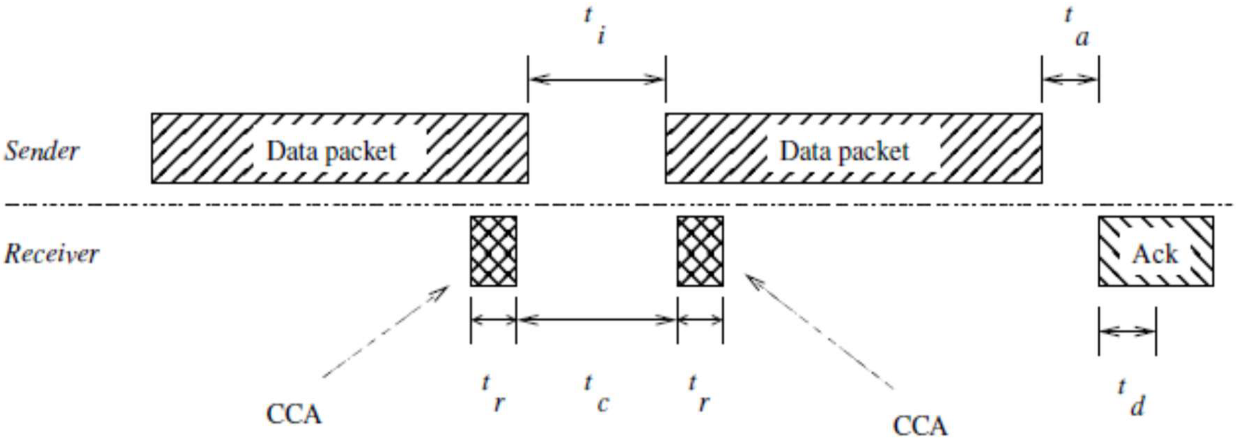
- Sender transmits many "probes", each containing address of intended receiver and taking T_p to transmit, with gaps instead of a long preamble.
  - On average, the preambles are half as long as B-MAC's preamble + more energy efficient since there is gaps in between
- Receiver checks address embedded in probe, sends ACK during the gap period between probes if it is the destination, otherwise go back to sleep. Note that if it wakes up in the middle of the probe, it would idle listen and wait for the next probe.
  - Worst case for receiver sensing time is $2\,T_p$ + gap (just missed the first probe)

ContikiMAC

A variant of X-MAC implemented in ContikiOS. The difference is

- the entire payload is immediately sent as probes (as it is small < 128 bytes)
- receiver has a pair of reception window, instead of one continuous window.





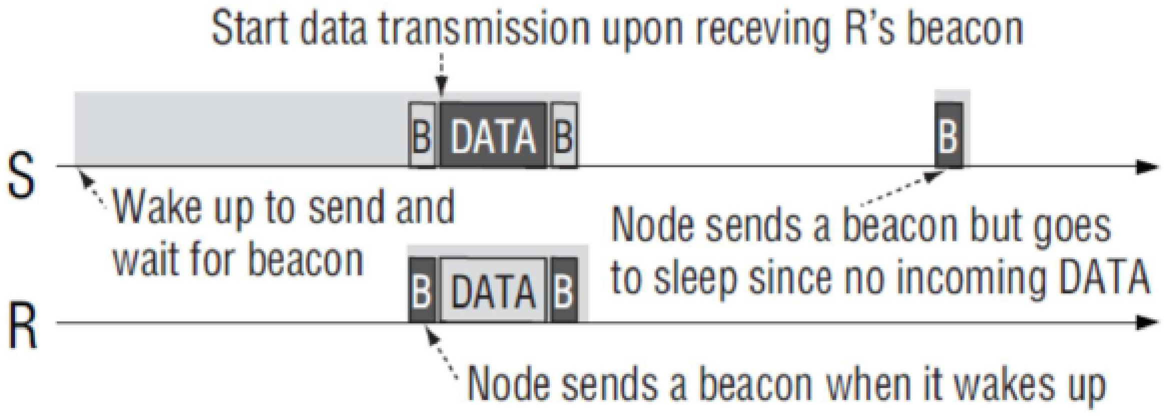
$$t_a + t_d < t_i < t_c < t_c + 2t_r < t_s.$$

Sender: t_i is interval b/w each pkt transmission, t_d is time required to successfully detect an ACK from receiver, t_s is the time to send a pkt.
Receiver: t_r is time required for stable RSSI, needed for stable CCA (Clear Channel Assessment) indication, t_c is interval between each CCA, t_a is time between receiving a pkt and sending ACK.

In CC2420, t_r = 0.192ms, t_i = 0.4ms, t_c = 0.5ms → min t_s = 0.884ms. With a bitrate of 250kbps and a 7 byte overhead / pkt, min pkt size is 21 bytes.

RI-MAC

In B-MAC, X-MAC and ContikiMAC, sender is responsible for sending a long preamble. Can we switch from sender-initiated to receiver-initiated?


Start data transmission upon receving R's beacon
S
B
DATA
B
B
Wake up to send and wait for beacon
Node sends a beacon but goes to sleep since no incoming DATA
R
B
DATA
B
Node sends a beacon when it wakes up

Whenever a transmitter wants to send, it wakes up and listens for a period of at most T_p. Receiver periodically wakes up every T_p and sends a beacon. Transmitter initiates transmission if the intended receiver wakes up.

Advantage: less transmission → less collision, may consume less energy.

Synchronous MAC Protocols

S-MAC

If we have clock synchronization, why not let all nodes wakeup at the same time?

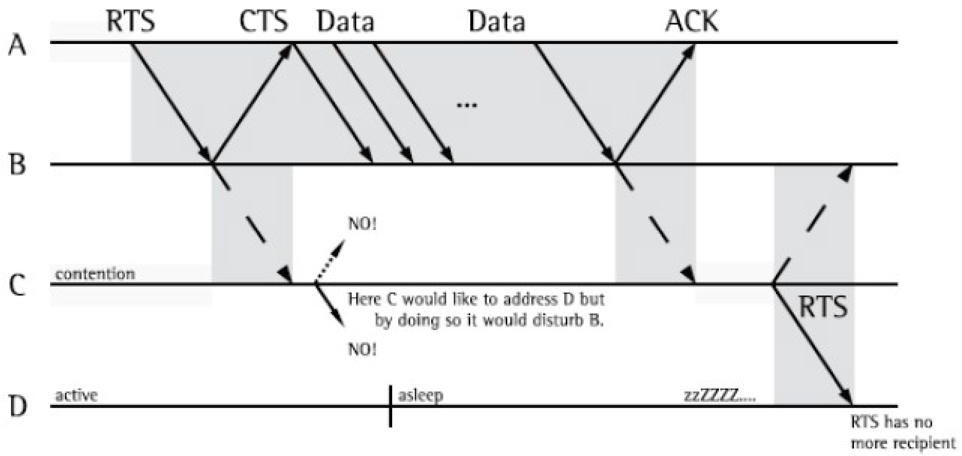
A node cycle between 2 phases:

- Listen phase (regular CSMA/CA)
  - The node compress all its activity within these phase → might increase collision.
  - Divided into SYNC and DATA phase. Before sending DATA, we employ RTS/CTS to win the shared channel.
- Sleep phase
  - If a node receive RTS/CTS not destined for the node, it will immediately sleep to avoid interfering with another communication

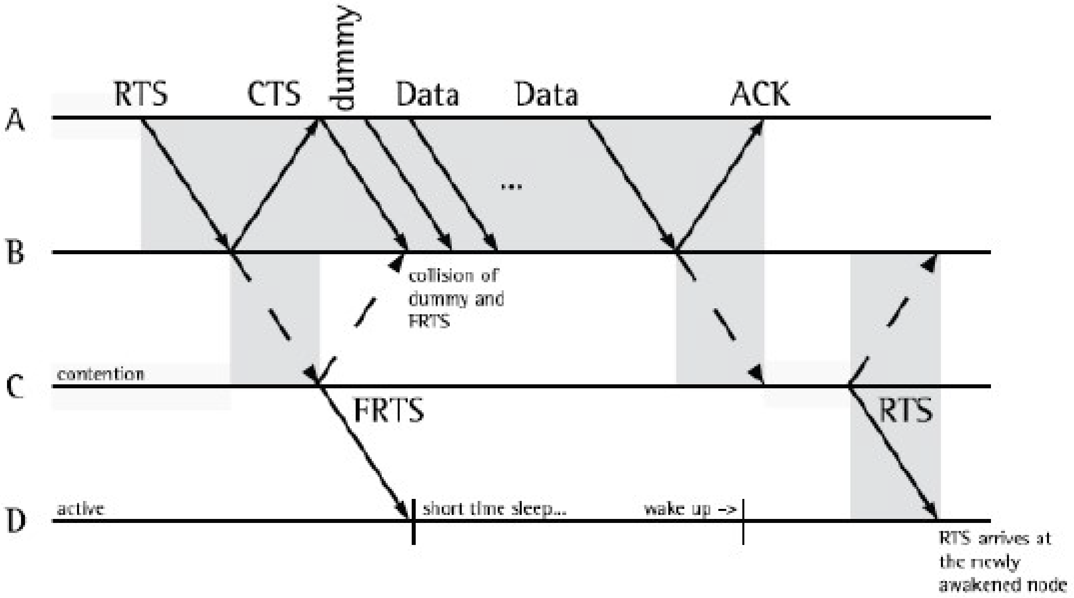
Challenge: How to synchronize the schedule of all nodes (assuming the nodes self-bootstrap itselves)?

Initially nodes are deployed to start after a random time interval. The node whose timer expires first is the synchronizer. For boundary nodes, that can hear multiple nodes claiming to be the synchronizer, it adopts multiple schedule.

Issue:


RTS
CTS
Data
Data
ACK
A
B
C
D
...
NO!
contention
Here C would like to address D but by doing so it would disturb B.
NO!
RTS
active
asleep
zzZZZZ....
RTS has no more recipient

T-MAC



T-MAC improves on S-MAC using 2 proposals:

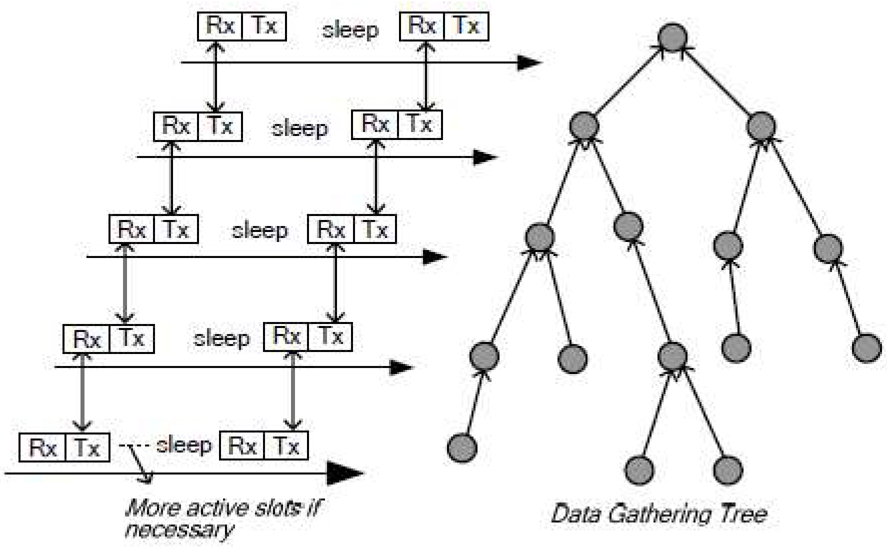
- Future Ready to Send (FRTS)
  - FRTS might interfere (might not if it is a hidden terminal) with B's receiving
- Full buffer priority - sends unsolicited RTS

Key Idea: We broke RTS/CTS in SMAC by keeping D awake (or D can sleep first and then wake up on an estimated time after B finish communication) even after the LISTEN phase. This info is contained in FRTS.

The proposal helps for 2 reasons:

1. Helps to reduce latency, because in SMAC, if you lose RTS/CTS, you need to wait until another listen phase before you can start sending.
2. Buffer might be full, if node C continuously lose RTS priority and keep receiving.

D-MAC



For specific application such as data gathering, topology information can be exploted, i.e. synchronize schedule based on node's position from root.

This reduces latency and collision, however if an event occurs too early, it needs to wait for the next cycle to be sent.

WiseMAC

If we have stronger assumption on what is known / can be learn, then we can achieve further improvements.

If sender knows when receiver wakes up to listen → start transmitting preamble just before receiver wakes up!

Receiver's schedule can be learnt if the node informs its neighbors of its wakeup schedule through the ACK packet, and other nodes learn and refresh their neighbors' sleep schedule. This is cheap if the schedule is periodic.

Issues: needs synchronization, overhead in fluctuative wakeup schedule, change in receivers (link quality, node failure/addition, etc).

PEGASIS

Power-Efficient Gathering in Sensor Information Systems (PEGASIS) is designed for synchronous and periodic traffic from all nodes with static topology.

- All sensors are chained together so that each sensor receives only one msg and transmits only 1 msg till message reaches the destination or the BS.
- Each msg is very small, so 1 packet can contain message from many nodes.
- Good for dense topology, e.g. indoor parking slot, farming, soil quality
- Good if there are delay constraints.

A^2-MAC

So far, 1 sender transmits to 1 receiver. If we allow multiple possible receiver (anycast), it could help.

A^2-MAC is an asynchronous, slot-based anycast MAC. It adapts X-MAC to stop sending probes after we receive the first ACK. It has been shown to have better energy efficiency (~2% duty cycle with 4 forwarders).

Drawback: Since there are multiple receivers, a packet are replicated many times. Although the latency is low, but it will increase energy used to send 1 packet → affect the throughput. To tackle this, we need to efficiently detect and remove (stop transmitting) duplicates.

Multi-Channel Protocols

So far, we have been discussing on protocols that send on a single channel. Now, we explore the possibility of using multiple channels.

Motivation:

- Increased throughput: more device can send at the same time.
- Increased reliability: when one channel is blocked caused by another send-recv, we can simply use another good channel to send-recv.

Challenges: harder to coordinate which channels to listen to, since now every device needs to jump to the correct channel at the right time. Previously, this was not a challenge as all nodes listens to just 1 channel.

Wifi Channels

Wifi has 14 22MHz-channels designated in the 2.4GHz range spaced 5 MHz apart (with the exception of a 12 MHz spacing before channel 14). These channels are overlapping. We have at most 3 non-overlapping channels (1,6,11 etc.)

ZigBee Channels

ZigBee has 26 non-overlapping 3MHz-channels designated in the 2.4GHz range spaced 5MHz apart. In practice, we can combine 4 subchannels into 1 bigger channel to achieve higher bitrate.

Most channels are full. Source of channel interference is WiFi, Bluetooth (channel hopping).

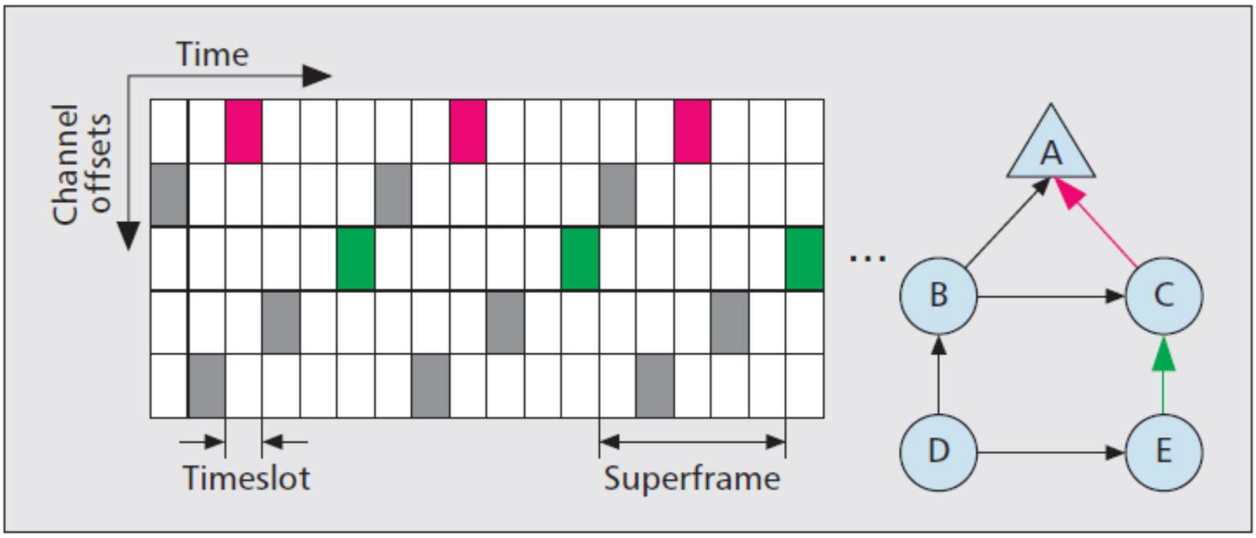
Approaches

1. Use a control channel:
    - a default channel for time synchronization and control message exchange
    - nodes switch to a different data channel for data communication
2. All channels are used for data communication
    - Nodes switches to another channel when link quality is bad.
    - We can find a "good" channel to switch to by
        - sampling the channels
        - keep a dynamic whitelist/blacklist of channels to use

    - needs a node as a scheduler so that the nodes do not keep switching back and forth to sample → keep the list and convey info.

Multichannel MAC are more preferred in traffic pattern where there are more data to be sent, since switching channels cost can be amortized.

TSCH Network



TSCH (Time-Synchronized Channel Hopping) network is one of the example of multi-channel protocols. It assumes a fixed topology.

- Time is divided into slots, multiple slots form a slotframe.
- Sends to receiver on slots it is listening on (e.g. green to C, red to A)
- Frequency chosen for a timeslot is based on a pseudorandom pattern
- Scheduling policy is not standardized

Wireless Routing

Overview

Routing in Wired network can use switches to significantly reduce the number of links, however it made routing more complex than a one-hop network.

Routing protocols are complex, hard to maintain/debug and incurred significant overhead. More so for wireless networks as it needs to care about the link quality (vary across time, need to handle interference from multiple nodes talking at the same time)

That's why in practice, wireless network deployments are single-hop, i.e. direct communication between device and base station → routing is trivial. Example: WiFi, cellular network.

To cover a large area, one-hop communication can use:

- sufficiently long range radio → lower bandwidth
- large number of base stations → expensive

Multi-hop routing is strictly better in terms of reach/coverage for the same number of base stations; it also provides more routes → potentially more robust.

Typically, a single-hop is considered better because there are less overhead, and only need to transmit once. However, the power to transmit a packet over the same distance is much higher (a factor of polynomial in N where N is the number of base stations in a multi-hop network)

Wireless Mesh Routing

- Formed by a network of **static** wireless nodes, **no duty-cycling**
- Needs to select routes that minimize interference and error
  - Interference: msg crossing with one another
  - Self Interference: consider A --- B --- C and say we are transmitting multihop A → B -> C, then the earlier packet forwarded by A (which is now being forwarded from B to C) self-interferes with the new packet sent from A to B
  - Error: higher channel quality → less error

Some possible routing metric (the last 2 is wireless-specific):

1. hop count
2. delay: avoid busy nodes
3. reliability: use good links
4. energy: longer distance → more packet loss → retransmission (expensive)

ETX

Expected Number of Transmission (ETX) is a very commonly used routing metric. It correlates with Packet Reception Ratio (PRR).

- ETX = 1/PRR.
- Path ETX = sum of ETX of each link in the path

Conceptually, ETX is achieved when we continuously retransmit loss packets.

Assumption: uniform bitrate for all links, otherwise we are not comparing apple-to-apple.

Shortest Path Routing

Many practical routing algos are based on the notion of shortest path between 2 nodes.

- Each link is assigned a positive number called its length.
- A shortest path algo routes each packet along the minimum length path between source and destination.
- Length is always 1 → minimum hop routing
- We can define length using other metrics like cost, delay, etc.

Tree-Based Routing

The simplest routing protocol is a tree-based protocol (e.g. spanning tree). Moreover, tree is suitable for data collection application, which is a common application of Wireless Sensor Network (WSN).

- Select a root node, which would perform a broadcast and neighboring nodes obtained: (1) distance (hop count) from the root, (2) link quality from the root
- Each node repeats the process: nodes select the "preferred" node as parent by some criteria, e.g. minimum hopcount / maximal link quality.
- At the end of the cycle, all nodes have a path towards root

Mobile Ad-Hoc Routing

Now we consider the case where nodes are mobile. Common applications include military applications and disaster relief.

A **Mobile Ad-Hoc Network (MANET)** is an autonomous system of mobile nodes connected by wireless links. Since nodes are mobile and has limited transmitting power, the network's topology may change dynamically in an unpredictable manner. So, routing path maybe available but needs to be "discovered".

Assumption:

- nodes move, but not fast enough that it is out of a nodes transmission range suddenly
- all nodes are connected, can be discovered
- all nodes are awake (no duty cycle)

AdHoc routing protocols can be classified as:

- Table Driven Protocols:
  - Proactive: when a packet needs to be forwarded, the route is already known → might have stale data (similar to wired network)
  - Each node maintains routing info to all other nodes in the network
  - When topology changes, updates are propagated through the network
- Source-Initiated Protocols
  - Reactive: determine route only when there is data to send → need to do more work in a short period of time.

Destination Sequenced Distance Vector (DSDV)

- Example of Table Driven Protocols
- Based on the Bellman-Ford algorithm
- Each node maintains a routing table: <seq_nbr, destination, number of hops>
  - seq_nbr helps to distinguish stale routes from new ones → to avoid loops
- Routing table updates are periodically transmitted

Ad-Hoc On-Demand Distance Vector (AODV)

- Example of Source-Initiated Protocols
- Nodes not in a selected path don't maintain routing information / participate in routing table exchanges

- Source node initiates path discovery process to locate other intermediate nodes and the destination, by broadcasting a Route Request (RREQ) to its neighbors

Delay Tolerant Network (DTN)

Mostly disconnected mobile nodes (to save power) that exists within a short range. Nodes meet one another opportunistically (not always connected).

Typically, to improve delivery ratio, messages are duplicated to many (if not all nodes) and by estimating contact probabilities. Nodes cannot broadcast / build a table because usually when nodes wakeup, nobody is around.

Constraints are: (1) number of messages buffered, (2) contact duration.

Examples: Bus network, deep space network

A few routing algorithms that has been explored:

Epidemic (Vahdat 2000) sends all packet to every node one encounters

- best performance if there is no constraint on number of buffered messages and contact duration
- Enhancement:
  - Only relay a message up to n-times (control number of copy)
  - Use ACK message + timeout to remove delivered messages from buffer

Single Copy (Spyropoulos, 2004)

- Default: Source keep the message and only transmit to destination if the 2 nodes are in direct contact
- Additionally, relay the message to another node with probability p. This helps to extend the coverage of the packet due to node mobility pattern

Other possibilities

- Duplicate a few copies that balance buffer space vs contact probability
- Estimate contact probabilities based on:
  - last time of encounter: transmit to nodes that encounter destination nodes more recently than itself
  - last contact information (location & time): transmit to nodes which are closer to destination.

Sensor Network Routing

- Similar to Mesh Routing, design for static, connected network. However, nodes are energy-constrained and need to duty cycle.
- Similar to DTN, but we can design the sensors' mobility pattern.

Link Measurements

In a wireless environment, each node needs to **decide if a link is available** (neighbor discovery != link to neighbor is available, since neighbor might sleep, channel quality drop etc.). This decision is usually made through some form of measurements, i.e. by sending beacons. Common metrics to be measured is PRR and RSSI, fortunately there is a threshold of RSSI, where PRR > 0.9.

These measurements are a tradeoff between communication cost and reaction time to environment / channel quality changes.

A "simple" beacon-based protocol

1. Node broadcasts beacon messages
2. When node A receives a "HELLO" msg from node B, it identifies B as its *inbound neighbors*. It will then include B in the next "I-HEAR-YOU" broadcast msg
3. When node B receives "I-HEAR-YOU" msg from node A, i.e. B's id is included in the msg, B will identify A as its outbound neighbor.
4. In practice, the "I-HEAR-YOU" msg contains the list of inbound neighbors as well as their PRR values, and the "HELLO" msg are combined into 1 packet.

Link Asymmetry

In general most links are asymmetric. We might want to use such links where the worse direction is not too bad because we still need to hear ACKs , send control packets etc.

Neighbor Maintenance

To reduce memory and computation cost, neighbor table tends to only contain a small number (6-10) "good" neighbors. However, doing this might lead to network partition. To tackle this problem, we can enforce a routing tree over the network to guarantee connectedness.

Collection Tree Protocol (CTP)

Collection Tree Protocol CTP (stanford.edu)

CTP uses distance-vector protocol instead of link-state protocol as nodes only need to store path to root, where as in link-state protocol it needs to store a lot of state → more memory, more power.

CTP, as a tree-based routing, is great for topology with a **fixed sink.**

Major subcomponents: 4-bit link estimator, datapath validation, adaptive beaconing

Agile and Accurate Link Quality Estimation (using 4-bit link estimator)

- Estimation is agile and driven by the ACK-bit of a packet (use info from physical, data link and network layers)
- Keep a moving window of last N packets to calculate PRR

Data Path Validation

- A data packet carries its estimated ETX value to the root
- Receiver compares ETX value of packet received and its own estimate. If its estimate is smaller → stored topology is stale. Re-measure.
- Allows detection on the first data packet, rather than waiting routing protocol update

Adaptive Beaconing

- Balance between sending as few beacons as possible (energy efficiency) with quickly reacting to topology changes
- Start from a fixed and small interval T_0 (min interval). If there is no change, double interval till T_H (max interval). If a change is detected, drops to T_0. Change is defined as:
  - Receiver sees larger ETX from transmitter
  - ETX drops significantly (> 1.5)
  - Neighbor ask for update (e.g. a new node joins)

Point-to-Point Routing

Next, we move to discuss routing for topology **without a fixed sink**; any node can send message to any other node. To achieve this, we can use tree-based routing by routing packet through the root, similar to spanning tree protocol in Ethernet, but this is inefficient in terms of path length.

General Sensor Routing

- Data centric: Addressing a node based on its properties, e.g. availability of information
  - Don't require unique address, e.g. Directed Diffusion
- Location based: A specialized class of data centric routing which utilizes location, i.e. each node knows its location. Example: GPSR, GHT

RPL

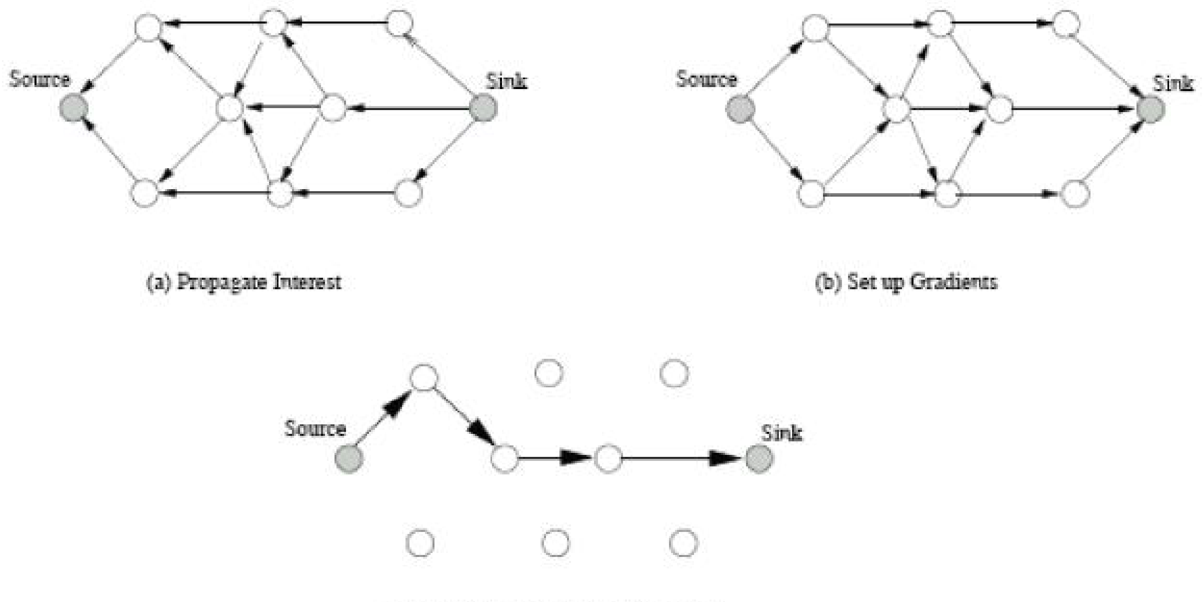
Routing over Low Power and Lossy Networks(RPL)

- Distance Vector (DV) Protocol:
  - Each router inform its neighbors of topology changes periodically
  - Each node maintains a vector (table) of minimum distance to every node.
  - DV protocol is more popular than link-state protocol because in low-powered networks we only want to go to root (less states to keep track off), whereas in link-state you need to keep track of the whole topology.
- Source Routing Protocol: allows a sender of a packet to specify the route the packet takes through the network
- Destination Oriented DAGs (DODAG)
  - Each node has a rank: define position relative to other nodes w.r.t. DODAG root
  - Version Number: a version is a specific iteration of a DODAG.
- Control Messages:
  - DODAG Information Solicitation (DIS): used to solicit a DIO from a RPL Node
  - DODAG Information Object (DIO): carries information to discover a RPL instance, learn its configuration parameters, select a DODAG parent set, maintain DODAG.

  - DIO is transmitted using a Trickle Timer, a dynamic timer that adjusts to the environment, similar to CTP.
  - Destination Advertisement Object (DAO): propagate destination info upward along DODAG
  - DAO-ACK
- Rank-based data-path validation mechanisms
  - Storing mode: unicast DAO msg from child to selected parent(s)
  - Non-storing mode: unicast DAO msg to the root.

Directed Diffusion

Similar to Ad-Hoc Network



(a) Propagate Interest

(b) Set up Gradients

(c) Send data and path Reinforcement

Node (sink) asks for a particular kind of information, but does not know if the information exists or not, e.g. is there a big vibration in construction tunnel, suspicious activities?

1. Node sends a (broadcast) query to express an interest for specific information
2. Neighboring nodes act if they were the sink and forward queries to their neighbors. Only 1 copy of interest/query is forwarded.
3. Source node floods the answer back to sink. Flooding / propagation stops when a (source) node can satisfy an interest / query.
4. Sink assumes that the node from which the first reply arrived is also the neighbor on the "shortest" path. It then **reinforce path** by sending more query on this path.

Observation:

- By using broadcast, if there is a failing node, we can simply trigger another path search.
- Path would converge over time and with different pairs of (source, sink), a distribution tree would emerged.
- Every node is informed about an event, even nodes that are not interested
- Propagation of data depends on reinforcement from interested sources
- Initial flooding expires after some time while reinforcement messages continue.

Issues:

- Need to store large amount of status information (which msg has arrived, which node sends first etc.)
- Substantial communication overhead in path discovery phase.
  - This is especially bad in wireless because of collisions. When path has reinforced, not so bad.
  - Need enough bandwidth (cannot use 900MHz) to support the flooding phase, otherwise low bitrate → cannot broadcast fast enough → retransmission → collision colapse.

Advantages:

- Should find a path as long as it exists
- Works well with (1) large amount of sensed events with few queries or (2) large amount of queries for same data.

GPSR

Motivation: Routing is expensive (many overheads, etx, many states to maintain). Can we do routing without these overheads? Yes, by maintaining some global information e.g. location via GPS.

GPSR (Greedy Perimeter Stateless Routing) is a very popular *location-based routing* protocol. Each node knows its location and the location of the destination node, e.g. via GPS. First deployment scenario is over water bodies (salinity measurement etc.)

There are many routing algorithm in this class.

GREEDY algorithm: to reach destination, node chooses the node that is "closest" to the destination as next hop.

- Problem: can get "trap" in a local minimum and cannot get out
  - Many "recovery" schemes have been proposed but may not always work.
- Works with anycast MAC
- GREEDY works very well in less-dense environment (only 1 way) or very-dense environment (there is almost no holes and it is very easy to find a shortest path). However, for medium-density environment (3-8 nodes), there is quite a number of options that lead to a local minima.

Getting global coordinates restricts the kind of deployment environment to outdoors, with a clear view of sky, etc.

To extend GPSR to more deployment scenario, e.g. indoors, we can construct virtual topologies (2D, 3D, 4D, etc) from the connectivity map (connectivity measured by link quality). Obviously, less dimension is cheaper, but might get stuck in local minima.

Geographic Hash Table

GHT combines *location-based routing* with data retrieval.

Motivation: Wants information generated by sensor nodes but don't know where to find them or which node generates them.

Assumption: nodes know their locations

Approach: combine location knowledge and storage capability.

Example:

- Initialization: divide area into a 10×10 grid, each node know which grid they belong to
- Insertion: Suppose we have an event `{ key: "apple", data }`. To store, we hash the key which return a coordinate of the grid, e.g. `(1,4)`. We then store `data` in one of the nodes in the grid `(1,4)`.
- Lookup: Hash the event key `apple` to get the coordinates `(1,4)`. Broadcast query to all nodes in grid `(1,4)`. Nodes that holds matching `data` will reply.

Issues:

- How to design the hash function to approximate / generate the coordinate
- What if there is no node at the specified location/coordinate
- What if some nodes leave and others join?
- What if a node fails? How to provide redundancy.

Hierarchical Routing

The network consists of multiple levels (from low power to high power):

- Sensors transmits only to their immediate cluster head (CH) to save energy. Sensors in a cluster have similar data.
- CH performs additional computation on data (aggregation) and sends to the next level CH. CH then communicates to a Base Station
- BS, away from the nodes, through which the end user can access data from the sensor network.

Contiki

Architecture:

- modular architecture
- event driven at kernel level with optional threading facilities
- OS facilities, e.g. sensor dat ahandling, communication, device drivers, etc. are provided in the form of services, each has its interface and implementation

Programming Model and Concurrency Support:

- sequential
- event-driven: on an event occurrence, kernel invokes handler code
- multi-threaded: thread runs until next blocking statement

Communication Protocol Support

- Provides an implementation of TCP/IP Protocol stack for small 8-bit microcontrollers (uIP)
- ContikiRPL: Provide IPv6 routing for low power and lossy networks as an implementation of RPL
- RIME: lightweight layered protocol stack → single hop unicast, single hop broadcast, multi-hop communication support, no support for multicast

---

1. Dead Reckoning: process of calculating current position of some moving object by using a previously determined position, or fix, and then incorporating estimates of speed, heading direction, and course over elapsed time↩