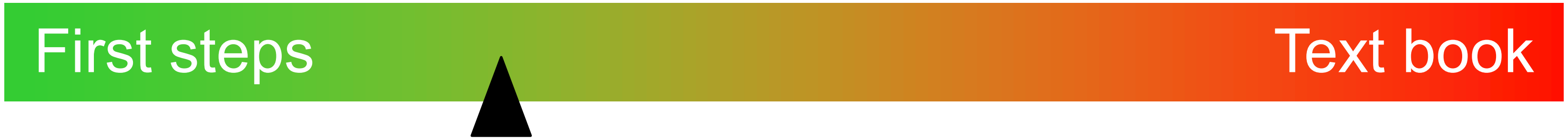
Media Access Control

Chapter 10

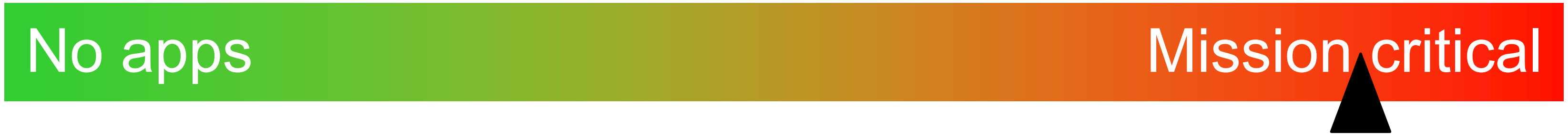
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Rating

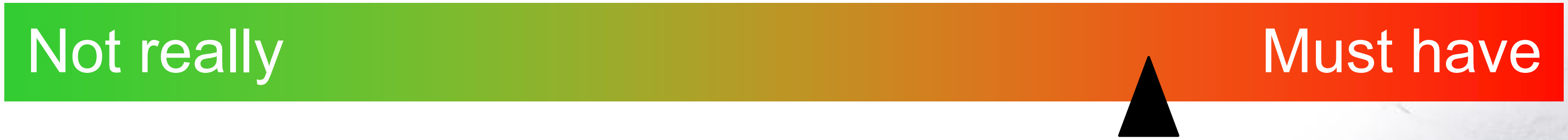
- Area maturity


First steps
Text book

- Practical importance


No apps
Mission critical

- Theoretical importance


Not really
Must have

Overview

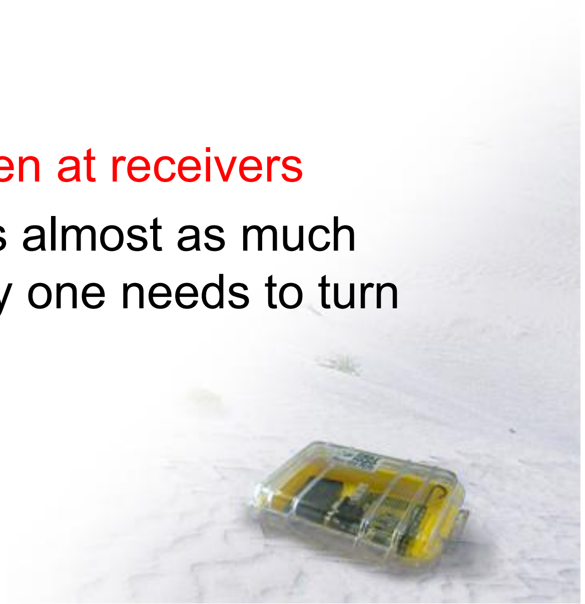
- Motivation
- Classification
- Case study: 802.11
- Other MAC layer techniques
- The broadcast problem

Motivation

- Can we apply media access methods from fixed networks?

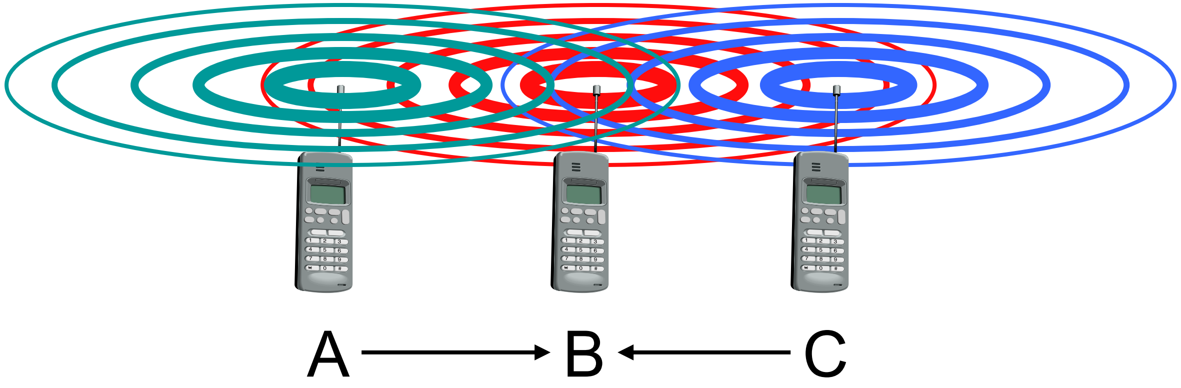
- Example CSMA/CD
  - **C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **D**etection
  - send as soon as the medium is free, listen into the medium if a collision occurs (original method in IEEE 802.3)

- Problems in wireless networks
  - signal strength decreases quickly with distance
  - senders apply CS and CD, but the collisions happen at receivers
  - Energy efficiency: having the radio turned on costs almost as much energy as transmitting, so to seriously save energy one needs to turn radio off!



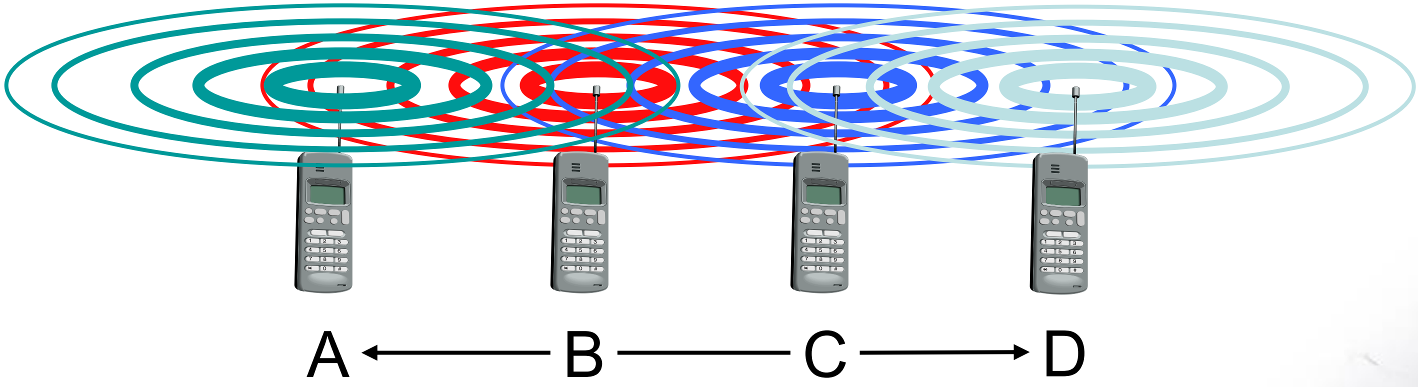
Motivation – Hidden terminal problem

- A sends to B, C cannot receive A
- C wants to send to B, C senses a “free” medium (CS fails)
- collision at B, A cannot receive the collision (CD fails)
- A is “hidden” for C



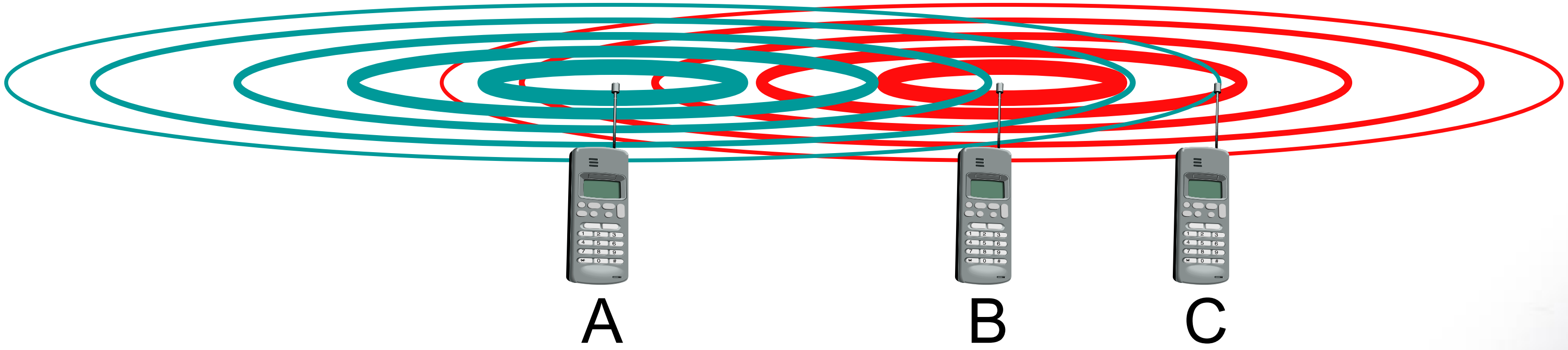
Motivation – Exposed terminal problem

- B sends to A, C wants to send to D
- C has to wait, CS signals a medium in use
- since A is outside the radio range of C waiting is not necessary
- C is “exposed” to B



Motivation - near and far terminals

- Terminals A and B send, C receives
  - the signal of terminal B hides A’s signal
  - C cannot receive A



- This is also a severe problem for CDMA networks
- precise power control

MAC Alphabet Soup

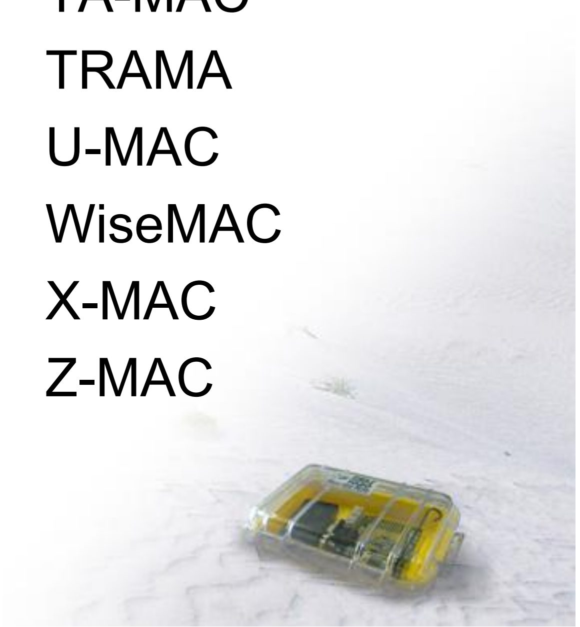
[TU Delft]

μ-MAC
Aloha
AI-LMAC
B-MAC
BitMAC
BMA
CMAC
Crankshaft
CSMA-MPS
CSMA/ARC
DMAC
E2-MAC
EMACs

f-MAC
FLAMA
Funneling-MAC
G-MAC
HMAC
LMAC
LPL
MMAC
nanoMAC
O-MAC
PACT
PCM
PEDAMACS

PicoRadio
PMAC
PMAC‘
Preamble sampling
Q-MAC
Q-MAC’
QMAC
RATE EST
RL-MAC
RMAC
RMAC’
S-MAC
S-MAC/AL

SMACS
SCP-MAC
SEESAW
Sift
SS-TDMA
STEM
T-MAC
TA-MAC
TRAMA
U-MAC
WiseMAC
X-MAC
Z-MAC



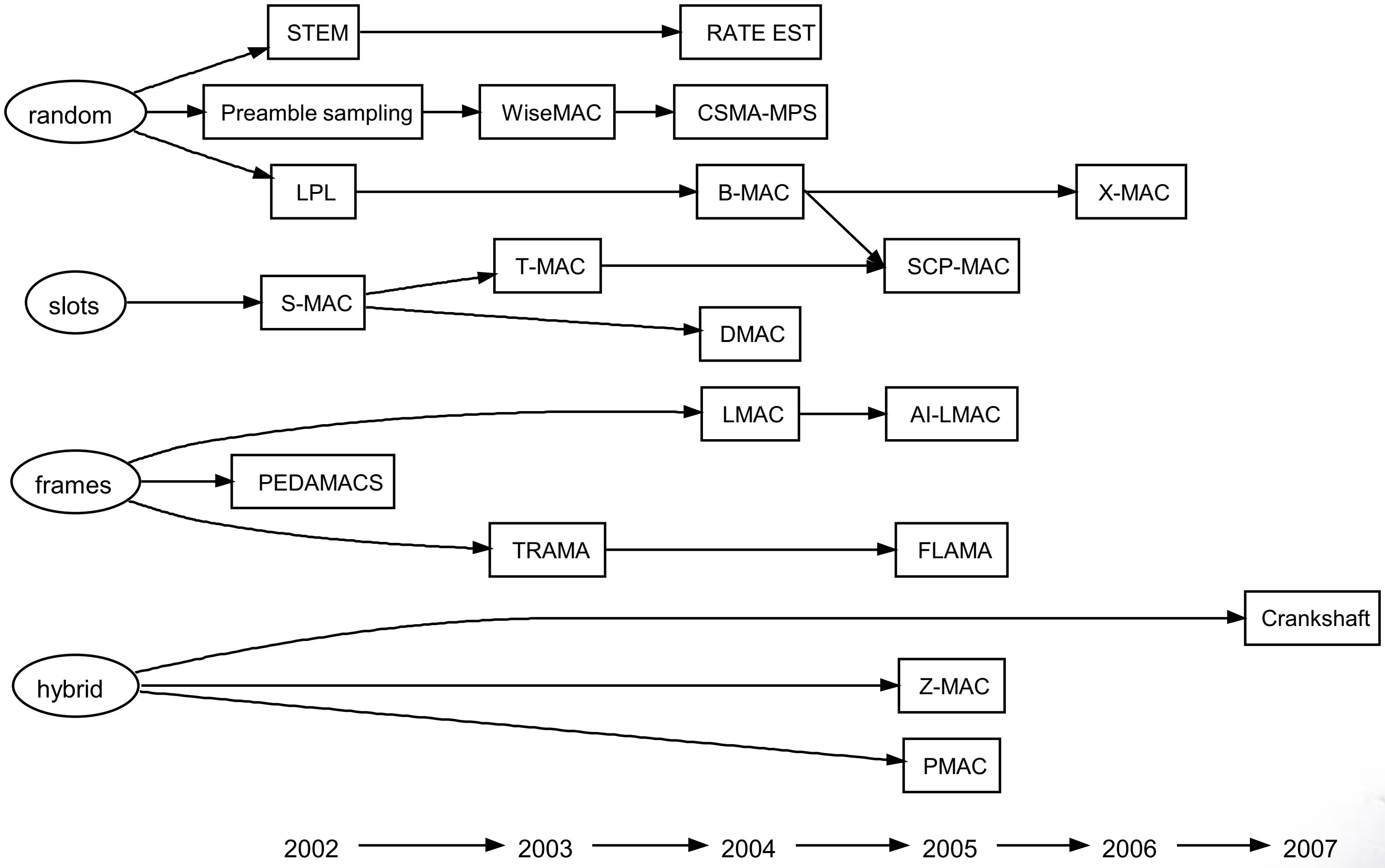
Traditional MAC protocol classification

- Contention Protocols
  - Transmit when you feel like transmitting
  - Retry if collision, try to minimize collisions, additional reservation modes
  - Problem: Receiver must be awake as well

- Scheduling Protocols
  - Use a “pre-computed” schedule to transmit messages
  - Distributed, adaptive solutions are difficult

- Other protocols
  - Hybrid solutions, e.g. contention with reservation → scheduling
  - Specific (“cross-layer”) solutions, e.g. Dozer for data gathering

Alternative view…



Access methods SDMA/FDMA/TDMA

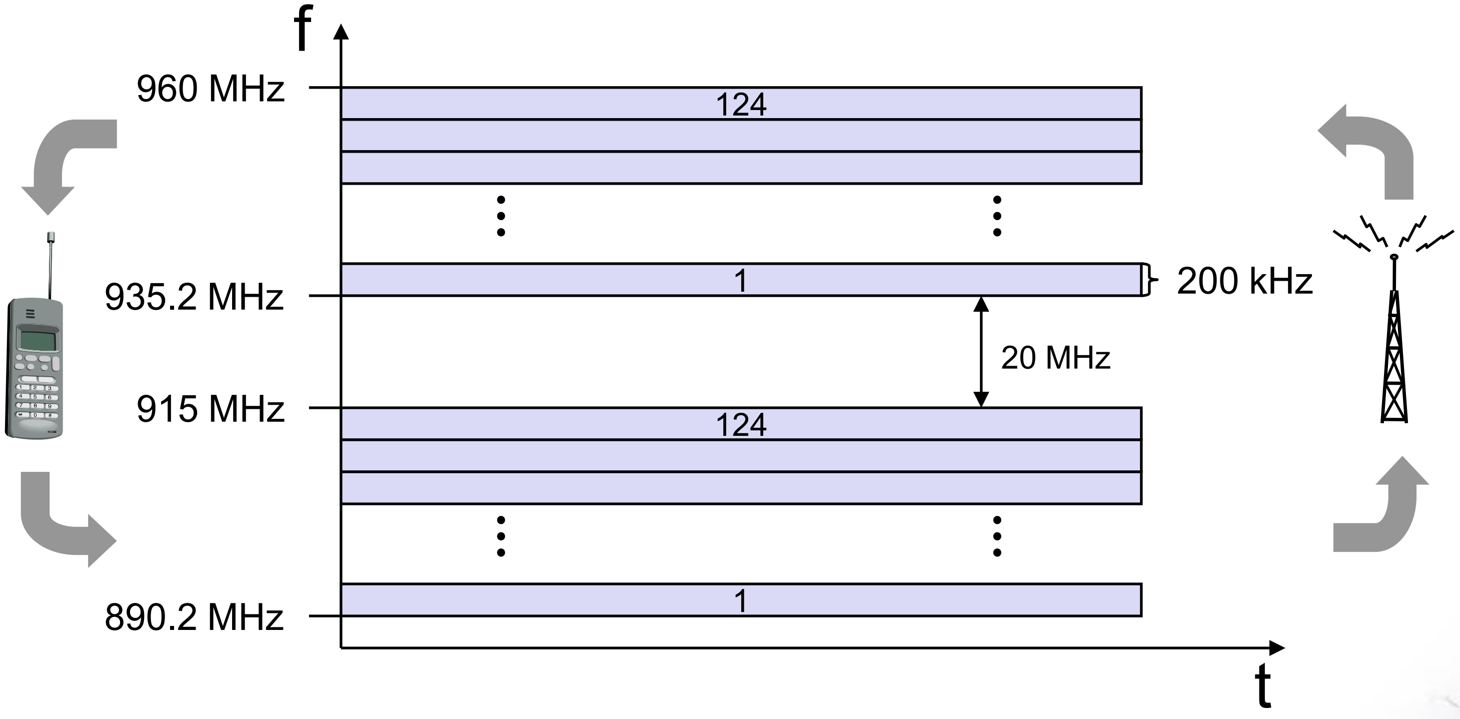
- SDMA (Space Division Multiple Access)
  - segment space into sectors, use directed antennas
  - Use cells to reuse frequencies
- FDMA (Frequency Division Multiple Access)
  - assign a certain frequency to a transmission channel
  - permanent (radio broadcast), slow hopping (GSM), fast hopping (FHSS, Frequency Hopping Spread Spectrum)
- TDMA (Time Division Multiple Access)
  - assign a fixed sending frequency for a certain amount of time
- CDMA (Code Division Multiple Access)
- Combinations!

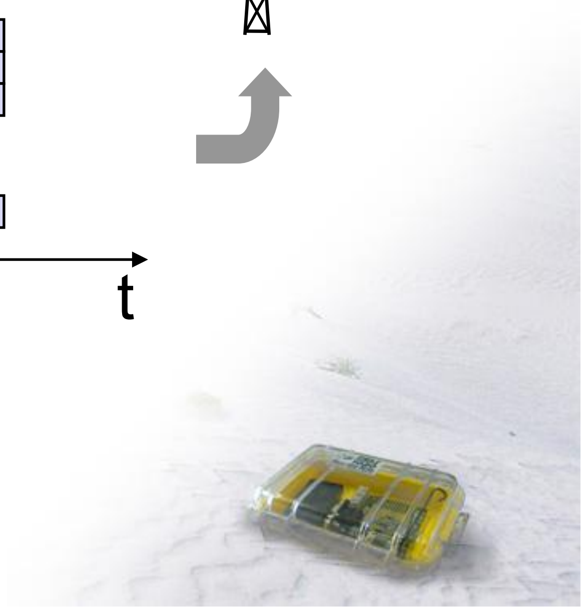
Comparison SDMA/TDMA/FDMA/CDMA

Approach	SDMA	TDMA	FDMA	CDMA
Idea	segment space into cells/sectors	segment sending time into disjoint time-slots, demand driven or fixed patterns	segment the frequency band into disjoint sub-bands	spread the spectrum using orthogonal codes
Terminals	only one terminal can be active in one cell/one sector	all terminals are active for short periods of time on the same frequency	every terminal has its own frequency, uninterrupted	all terminals can be active at the same place at the same moment, uninterrupted
Signal separation	cell structure, directed antennas	synchronization in the time domain	filtering in the frequency domain	code plus special receivers
Advantages	very simple, increases capacity per km²	established, fully digital, flexible	simple, established, robust	flexible, less frequency planning needed, soft handover
Dis-advantages	inflexible, antennas typically fixed	guard space needed (multipath propagation), synchronization difficult	inflexible, frequencies are a scarce resource	complex receivers, needs more complicated power control for senders
Comment	only in combination with TDMA, FDMA or CDMA useful	standard in fixed networks, together with FDMA/SDMA used in many mobile networks	typically combined with TDMA (frequency hopping patterns) and SDMA (frequency reuse)	still faces some problems, higher complexity, lowered expectations; will be integrated with TDMA/FDMA

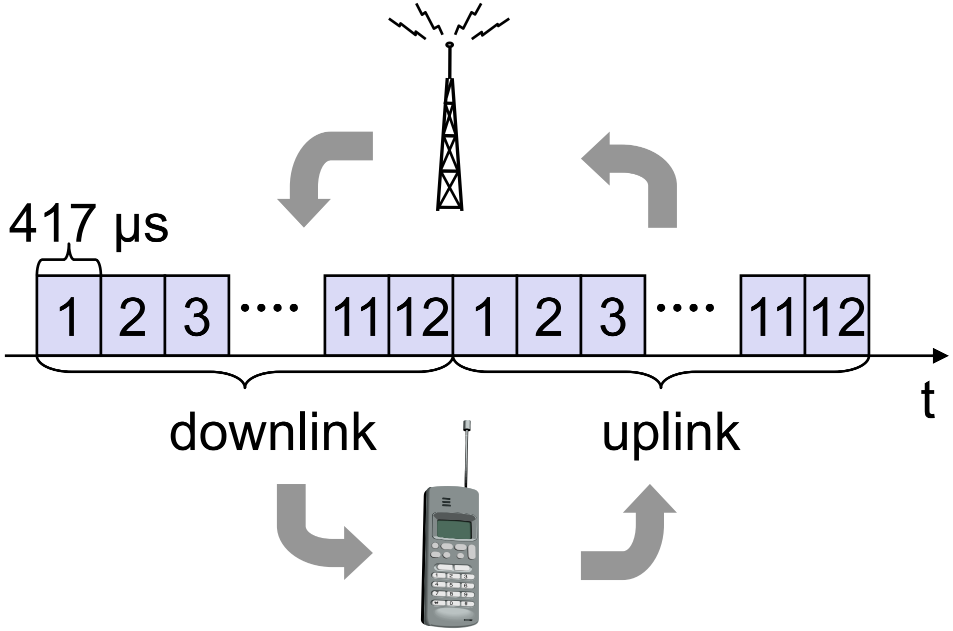
[J.Schiller]

FDD/FDMA - general scheme, example GSM @ 900Mhz

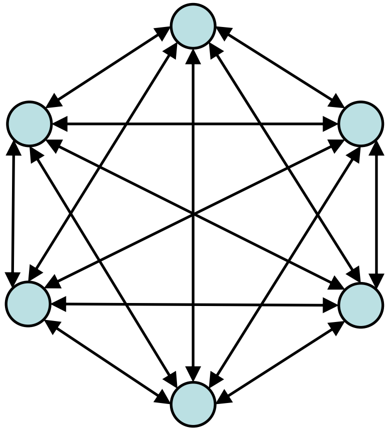




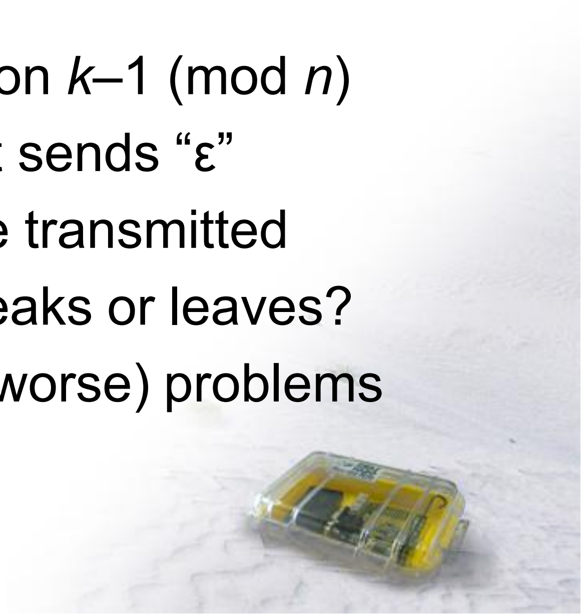
TDD/TDMA - general scheme, example DECT



TDMA – Motivation

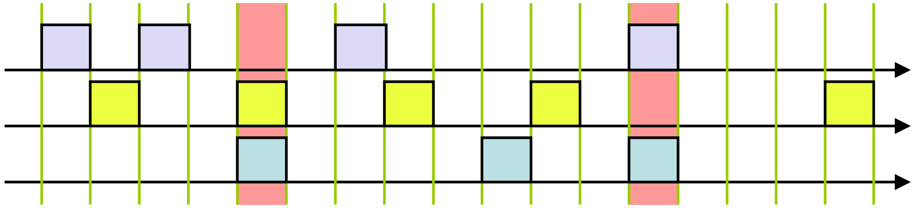


- We have a system with *n* stations (0,1,2,…,*n*–1) and one shared channel
- The channel is a perfect broadcast channel, that is, if any single station transmits alone, the transmission can be received by every other station. There is no hidden or exposed terminal problem. If two or more transmit at the same time, the transmission is garbled.
- Round robin algorithm: station *k* sends after station *k*–1 (mod *n*)
- If a station does not need to transmit data, then it sends “ε”
- There is a maximum message size *m* that can be transmitted
- How efficient is round robin? What if a station breaks or leaves?
- All deterministic TDMA protocols have these (or worse) problems



TDMA – Slotted Aloha

- We assume that the stations are perfectly synchronous
- In each time slot each station transmits with probability *p*.



$$P_1 = \Pr[\text{Station 1 succeeds}] \;=\; p(1-p)^{n-1}$$

$$P = \Pr[\text{any Station succeeds}] = nP_1$$

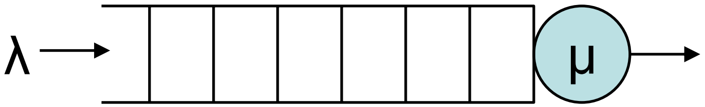
$$\text{maximize } P: \frac{dP}{dp} = n(1-p)^{n-2}(1-pn) \overset{!}{=} 0 \quad \Rightarrow \quad pn = 1$$

$$\text{then, } P = (1-\frac{1}{n})^{n-1} \geq \frac{1}{e}$$

- In slotted aloha, a station can transmit successfully with probability at least 1/*e*. How quickly can an application send packets to the radio transmission unit? This question is studied in queuing theory.

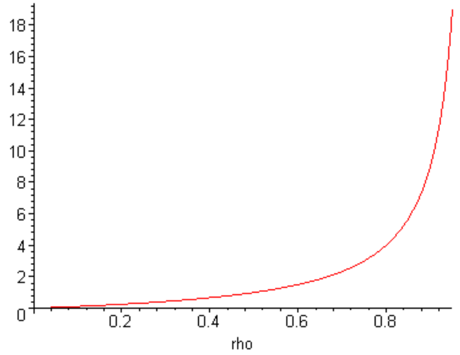
Queuing Theory – the basic basics in a nutshell

- Simplest M/M/1 queuing model (M=Markov):
- Poisson arrival rate λ, exponential service time with mean $1/\mu$



- In our time slot model, this means that the probability that a new packet is received by the buffer is λ; the probability that sending succeeds is μ, for any time slot. To keep the queue bounded we need $\rho = \lambda/\mu < 1$.

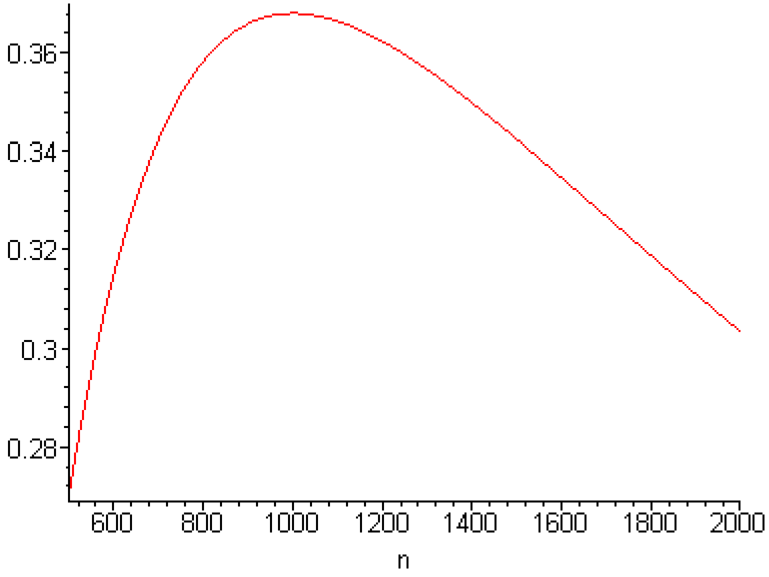
- In the equilibrium, the expected number of packets in the system is $N = \rho/(1-\rho)$, the average time in the system is $T = N/\lambda$.



Slotted Aloha vs. Round Robin

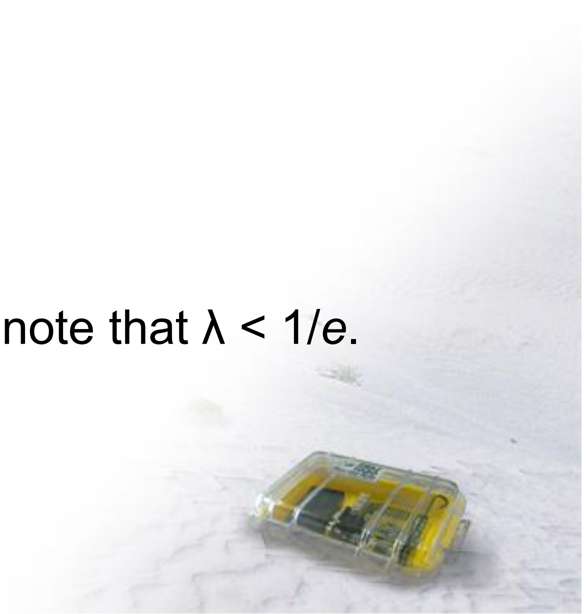
- – Slotted aloha uses not every slot of the channel; the round robin protocol is better.
- + What happens in round robin when a new station joins? What about more than one new station? Slotted aloha is more flexible.

- Example: If the actual number of stations is twice as high as expected, there is still a successful transmission with probability 30%. If it is only half, 27% of the slots are used successfully.



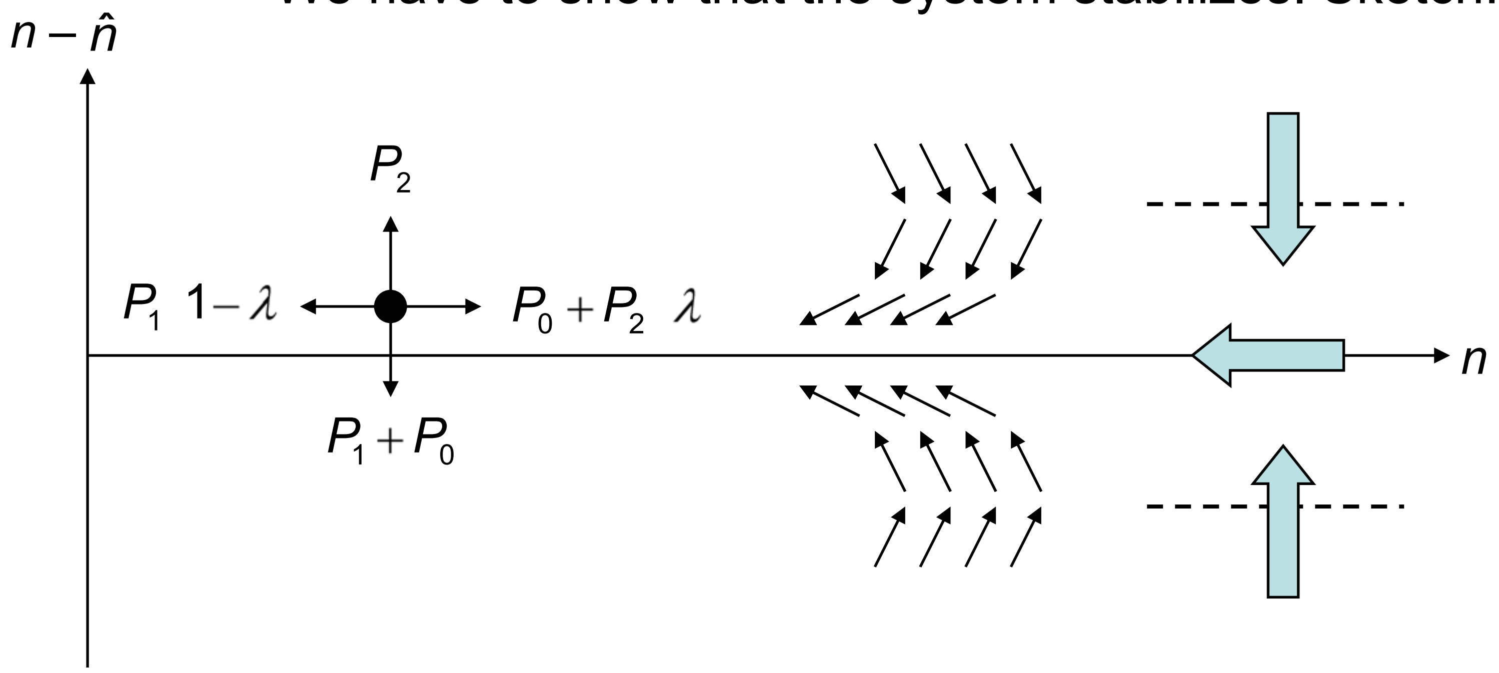
Adaptive slotted aloha

- Idea: Change the access probability with the number of stations
- How can we estimate the current number of stations in the system?
- Assume that stations can distinguish whether 0, 1, or more than 1 stations send in a time slot.
- Idea:
  - If you see that nobody sends, increase *p*.
  - If you see that more than one sends, decrease *p.*
- Model:
  - Number of stations that want to transmit: n.
  - Estimate of n: $\hat{n}$
  - Transmission probability: $p = 1/\hat{n}$
  - Arrival rate (new stations that want to transmit): λ; note that $\lambda < 1/e$.



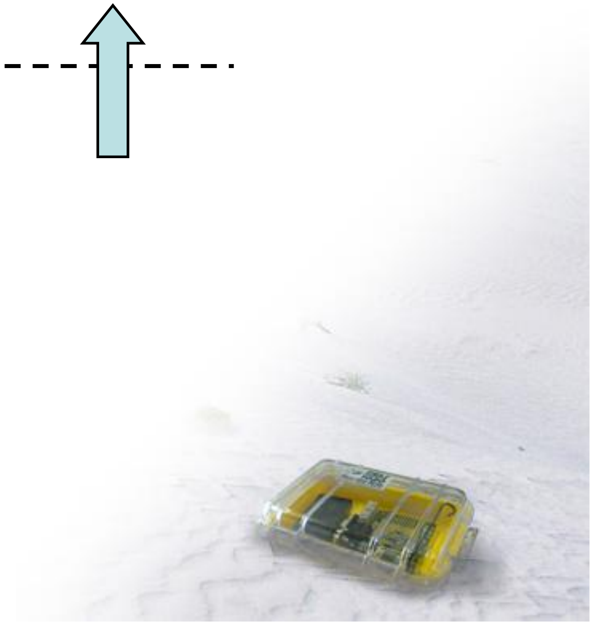
Adaptive slotted aloha 2

We have to show that the system stabilizes. Sketch:



$\hat{n} \leftarrow \hat{n}+\lambda-1$, if success or idle

$\hat{n} \leftarrow \hat{n}+\lambda+\frac{1}{e-2}$, if collision



Adaptive slotted aloha Q&A

Q: What if we do not know λ, or λ is changing?

A: Use λ = 1/e, and the algorithm still works

Q: How do newly arriving stations know $\hat{n}$?

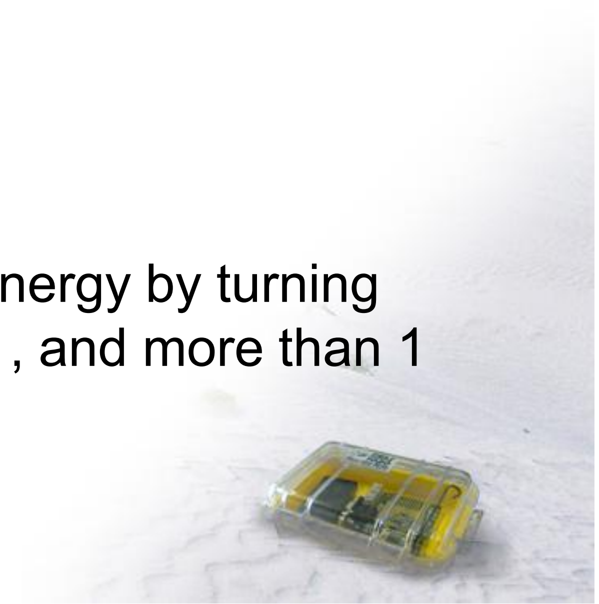
A: We send $\hat{n}$ with each transmission; new stations do not send before successfully receiving the first transmission.

Q: What if stations are not synchronized?

A: Aloha (non-slotted) is twice as bad

Q: Can stations really listen to all time slots (save energy by turning off)? Can stations really distinguish between 0, 1, and more than 1 sender?

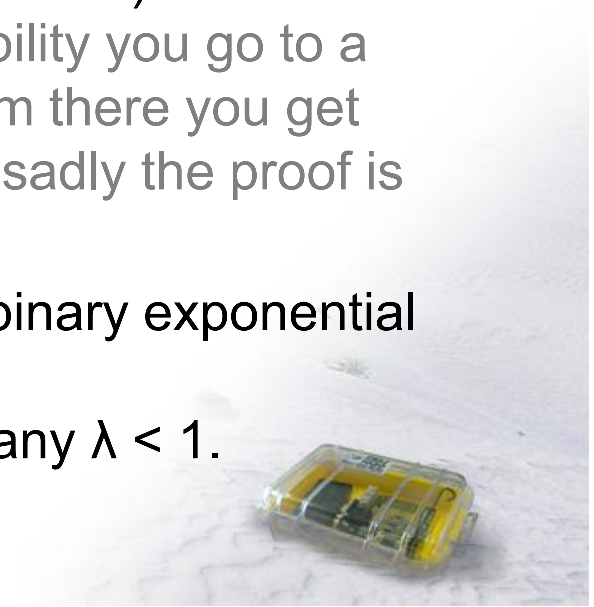
A: Maybe. One can use systems that only rely on acknowledgements…



Backoff Protocols

- Backoff protocols rely on acknowledgements only.
- Binary exponential backoff, for example, works as follows:
- If a packet has collided *k* times, we set $p = 2^{-k}$
  Or alternatively: wait from random number of slots in $[1..2^{k}]$

- It has been shown that binary exponential backoff is not stable for any λ > 0 (if there are infinitely many potential stations)
  [Proof sketch: with very small but positive probability you go to a bad situation with many waiting stations, and from there you get even worse with a potential function argument – sadly the proof is too intricate to be shown in this course ☺]
- Interestingly when there are only finite stations, binary exponential backoff becomes unstable with λ > 0.568;
  Polynomial backoff however, remains stable for any λ < 1.

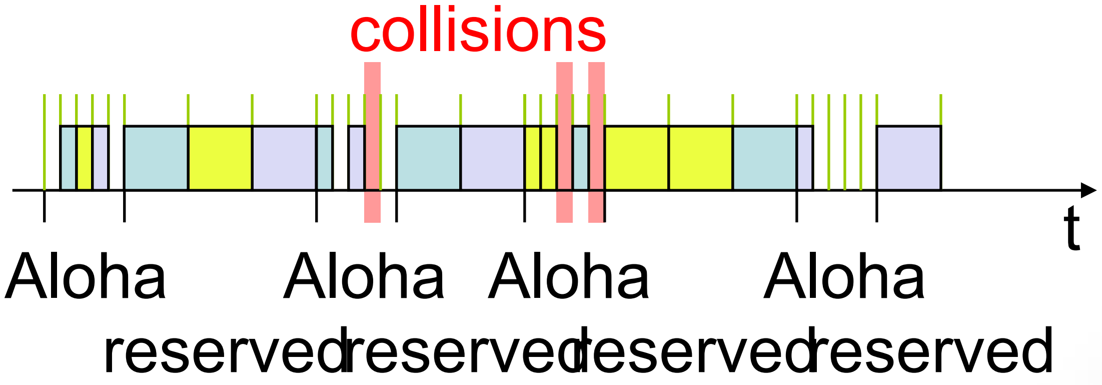


Demand Assigned Multiple Access (DAMA)

- Channel efficiency only 36% for Slotted Aloha, and even worse for Aloha or backoff protocols.
- Practical systems therefore use reservation whenever possible. But: Every scalable system needs an Aloha style component.
- Reservation:
  - a sender *reserves* a future time-slot
  - sending within this reserved time-slot is possible without collision
  - reservation also causes higher delays
  - typical scheme for satellite systems
- Examples for reservation algorithms:
  - Explicit Reservation (Reservation-ALOHA)
  - Implicit Reservation (PRMA)
  - Reservation-TDMA
  - Multiple Access with Collision Avoidance (MACA)

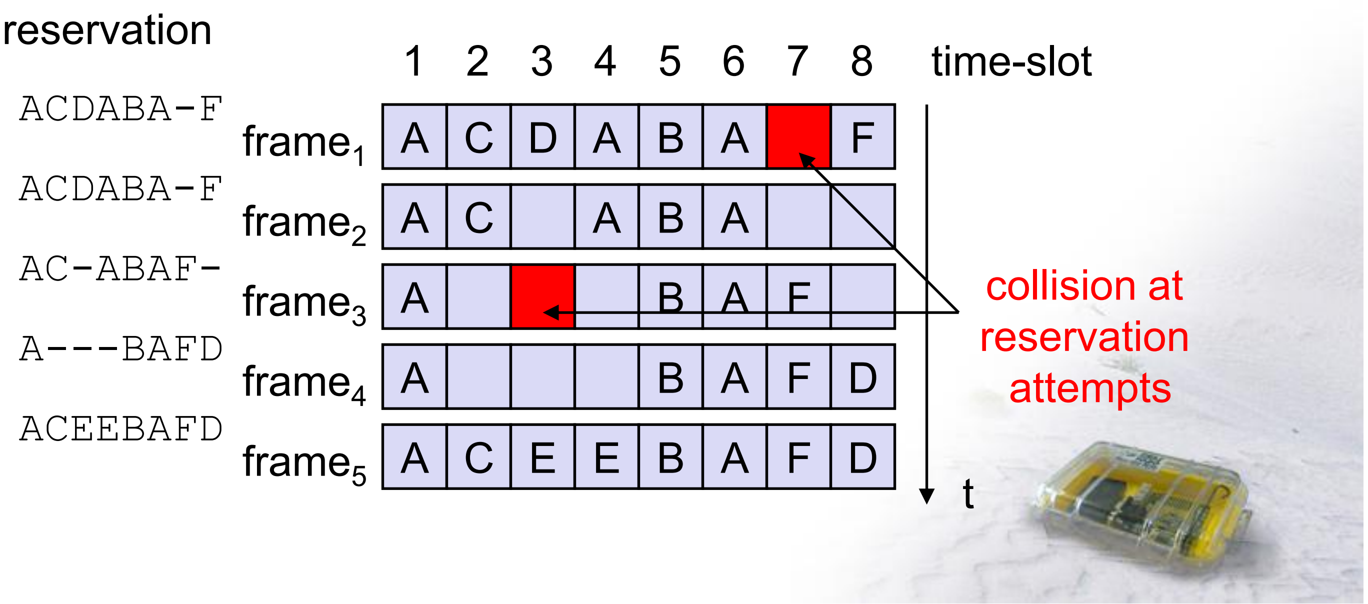
DAMA: Explicit Reservation

- *Aloha mode* for reservation: competition for small reservation slots, collisions possible
- *reserved mode* for data transmission within successful reserved slots (no collisions possible)
- it is important for all stations to keep the reservation list consistent at any point in time and, therefore, all stations have to synchronize from time to time



DAMA: Packet Reservation MA (PRMA)

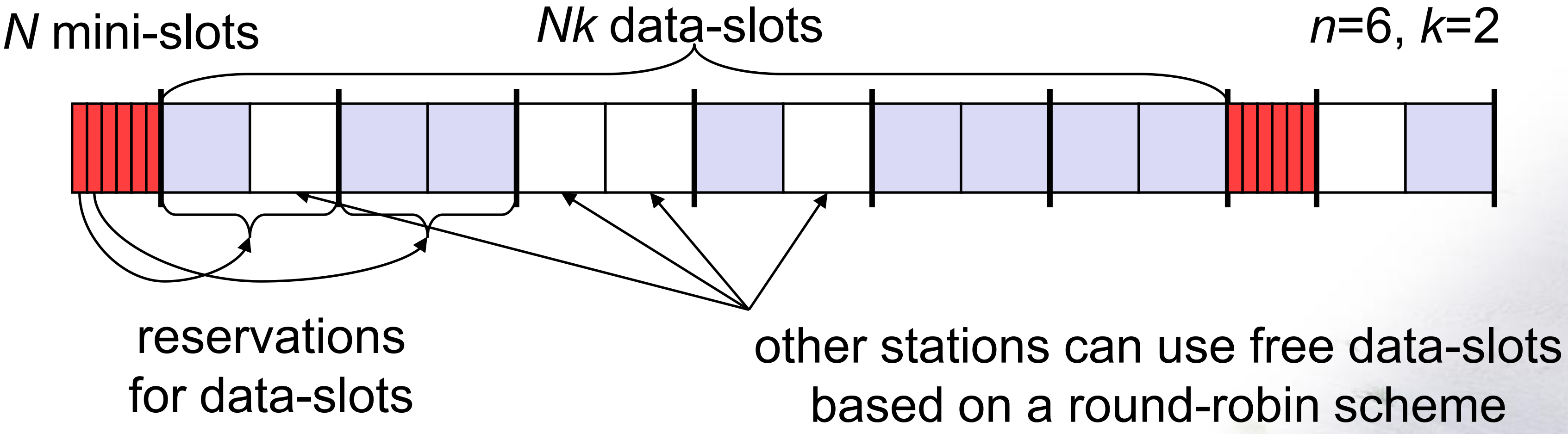
- a certain number of slots form a frame, frames are repeated
- stations compete for empty slots according to the slotted aloha principle
- once a station reserves a slot successfully, this slot is automatically assigned to this station in all following frames as long as the station has data to send
- competition for this slots starts again as soon as the slot was empty in the last frame

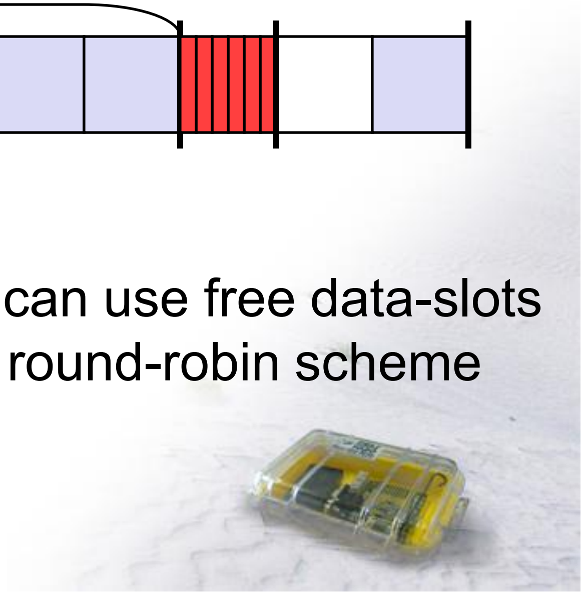


reservation		1	2	3	4	5	6	7	8
ACDABA-F	frame_1	A	C	D	A	B	A	🟥	F
ACDABA-F	frame_2	A	C		A	B	A		
AC-ABAF-	frame_3	A		🟥		B	A	F	
A---BAFD	frame_4	A				B	A	F	D
ACEEBAFD	frame_5	A	C	E	E	B	A	F	D

DAMA: Reservation TDMA

- every frame consists of *n* mini-slots and *x* data-slots
- every station has its own mini-slot and can reserve up to *k* data-slots using this mini-slot (i.e. *x* = *nk*).
- other stations can send data in unused data-slots according to a round-robin sending scheme (best-effort traffic)


N mini-slots
Nk data-slots
n=6, *k*=2
reservations for data-slots
other stations can use free data-slots based on a round-robin scheme



Multiple Access with Collision Avoidance (MACA)

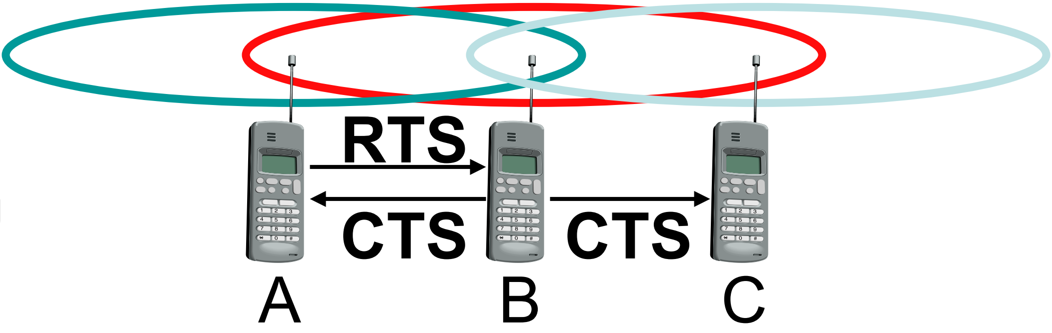
- Use short signaling packets for collision avoidance
  - Request (or ready) to send RTS: a sender requests the right to send from a receiver with a short RTS packet before it sends a data packet
  - Clear to send CTS: the receiver grants the right to send as soon as it is ready to receive

- Signaling packets contain
  - sender address
  - receiver address
  - packet size

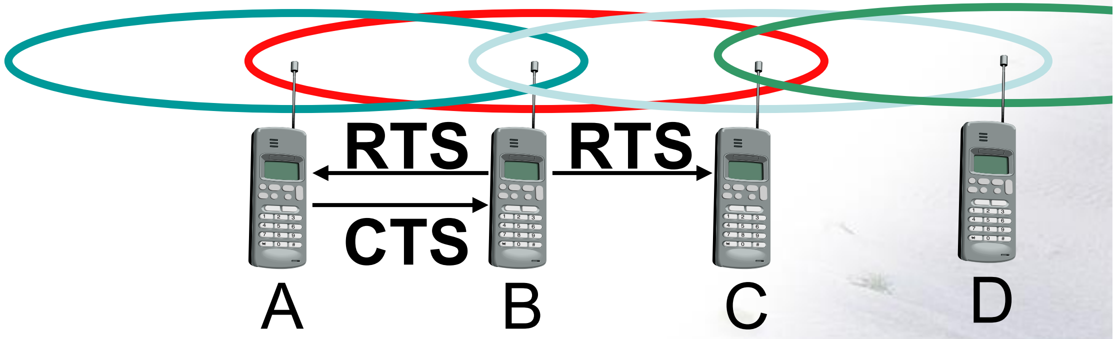
- Example: Wireless LAN (802.11) as DFWMAC

MACA examples

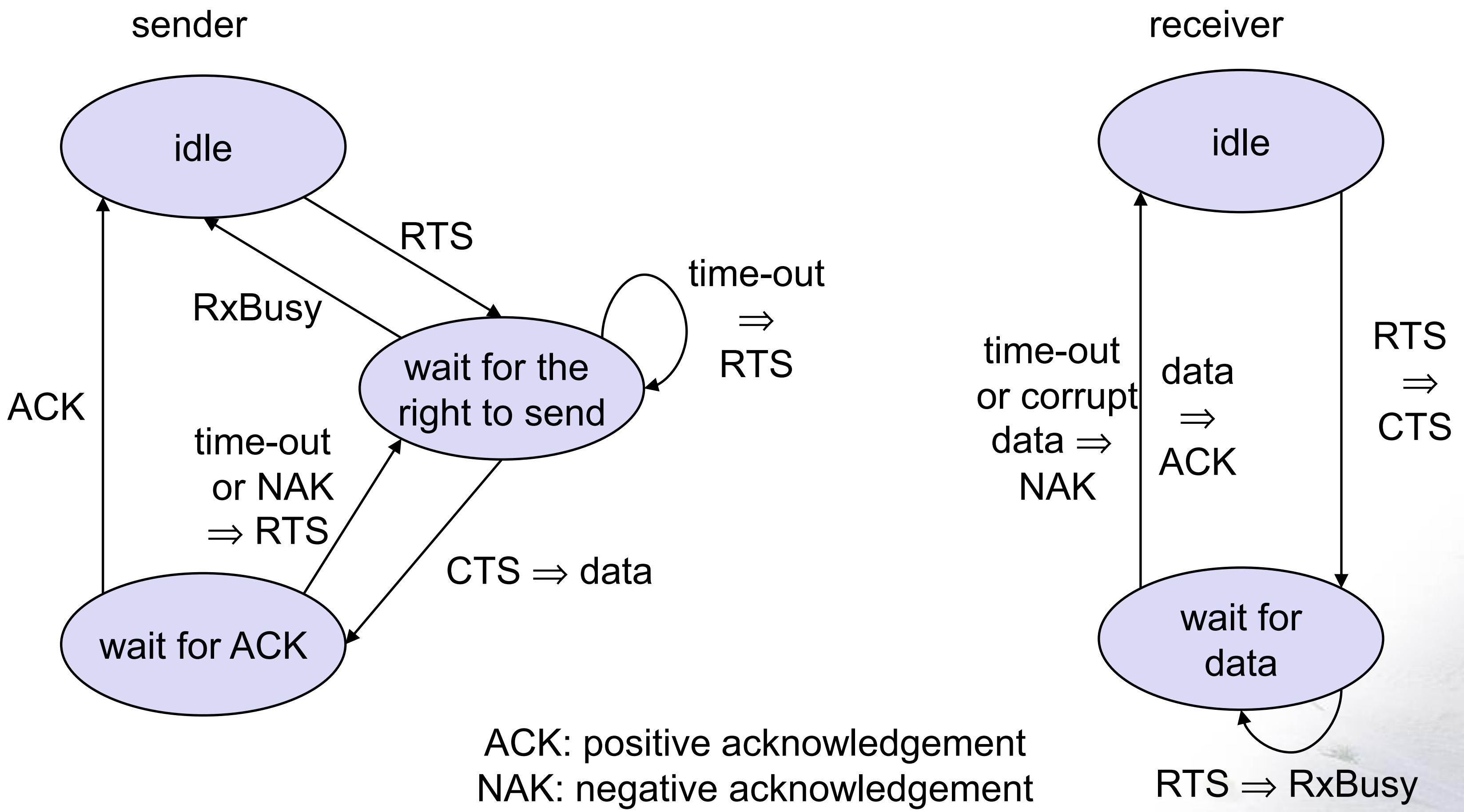
- MACA avoids the problem of hidden terminals
  - A and C want to send to B
  - A sends RTS first
  - C waits after receiving CTS from B



- MACA avoids the problem of exposed terminals
  - B wants to send to A, and C to D
  - now C does not have to wait as C cannot receive CTS from A



MACA variant: DFWMAC in IEEE802.11



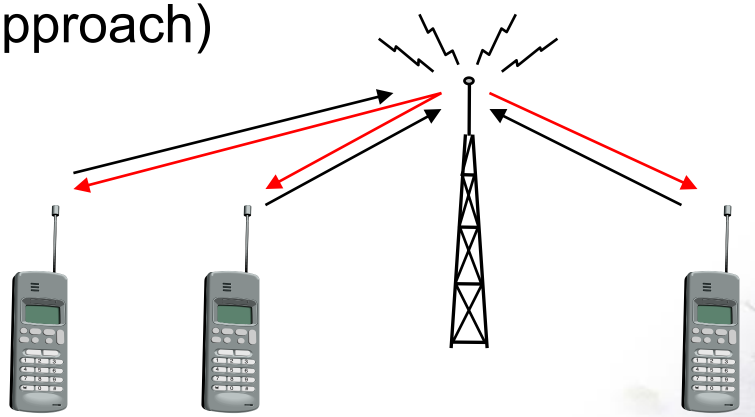
ACK: positive acknowledgement
NAK: negative acknowledgement
RxBusy: receiver busy

Polling mechanisms

- If one terminal can be heard by all others, this “central” terminal (a.k.a. base station) can poll all other terminals according to a certain scheme
  - Use a scheme known from fixed networks
  - The base station chooses one address for polling from the list of all stations
  - The base station acknowledges correct packets and continues polling the next terminal
  - The cycle starts again after polling all terminals of the list
  - An aloha-style component is needed to allow new stations join

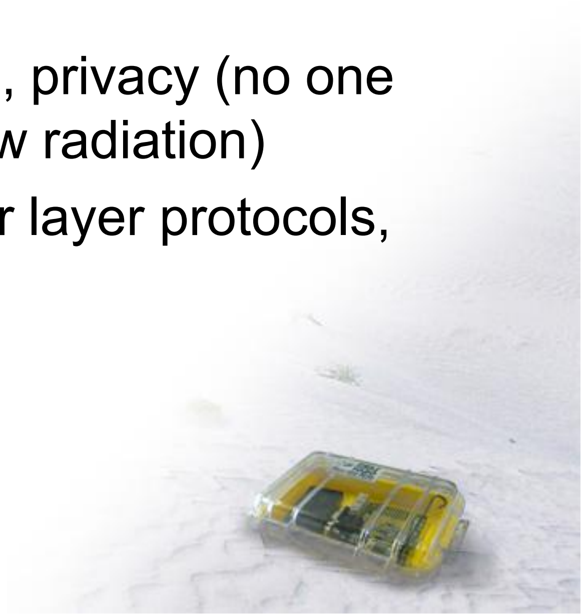
Inhibit Sense Multiple Access (ISMA)

- Current state of the medium is signaled via a “busy tone”
- the base station signals on the downlink (base station to terminals) whether the medium is free
- terminals must not send if the medium is busy
- terminals can access the medium as soon as the busy tone stops
- the base station signals collisions and successful transmissions via the busy tone and acknowledgements, respectively (media access is not coordinated within this approach)
- Example: for CDPD
  (USA, integrated into AMPS)



802.11 Design goals

- Global, seamless operation
- Low power consumption for battery use
- No special permissions or licenses required
- Robust transmission technology
- Simplified spontaneous cooperation at meetings
- Easy to use for everyone, simple management
- Interoperable with wired networks
- Security (no one should be able to read my data), privacy (no one should be able to collect user profiles), safety (low radiation)
- Transparency concerning applications and higher layer protocols, but also location awareness if necessary

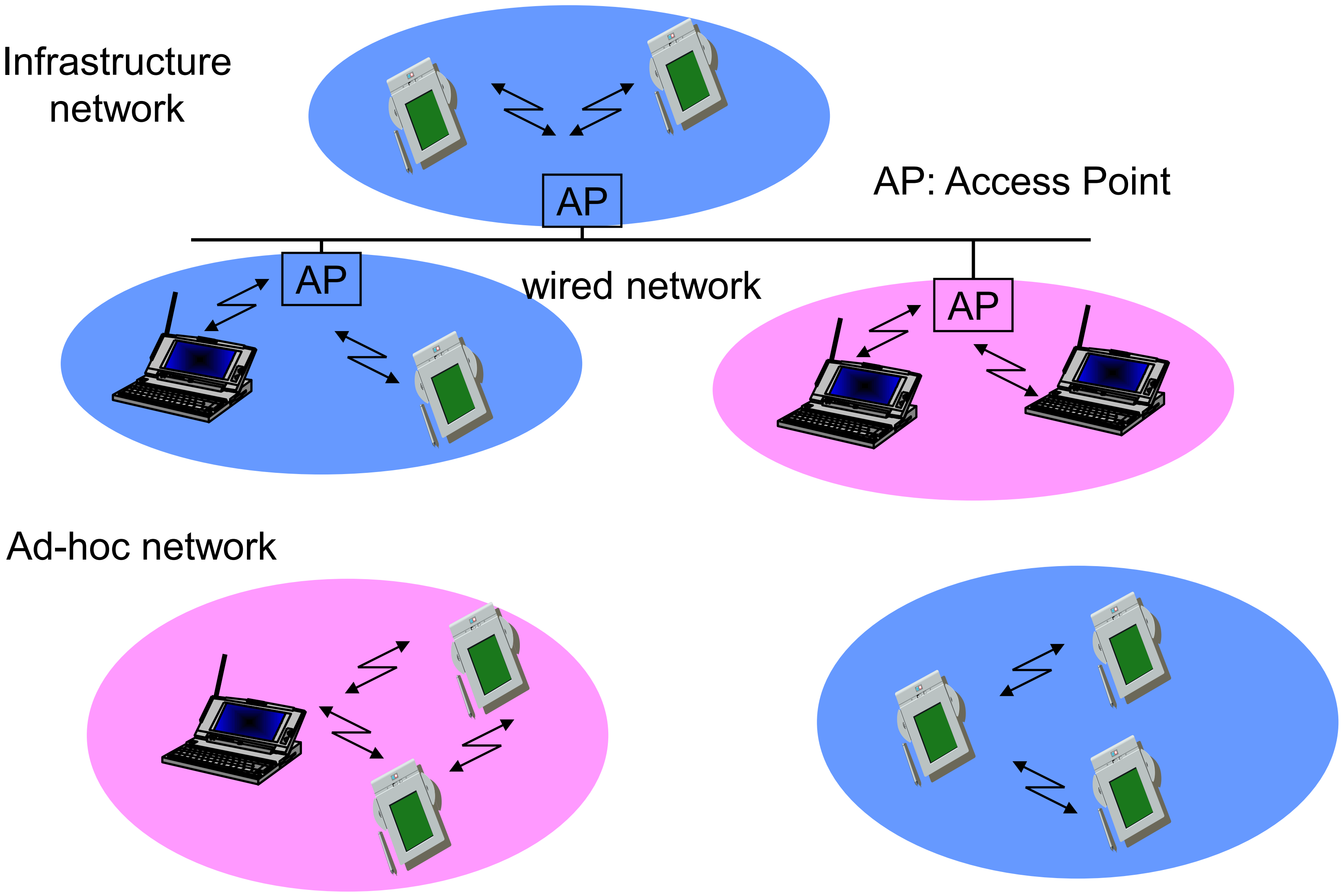


802.11 Characteristics

+ Very flexible (economical to scale)
+ Ad-hoc networks without planning possible
+ (Almost) no wiring difficulties (e.g. historic buildings, firewalls)
+ More robust against disasters or users pulling a plug

– Low bandwidth compared to wired networks (10 vs. 100[0] Mbit/s)
– Many proprietary solutions, especially for higher bit-rates, standards take their time
– Products have to follow many national restrictions if working wireless, it takes a long time to establish global solutions (IMT-2000)
– Security
– Economy

802.11 Infrastructure vs. ad hoc mode



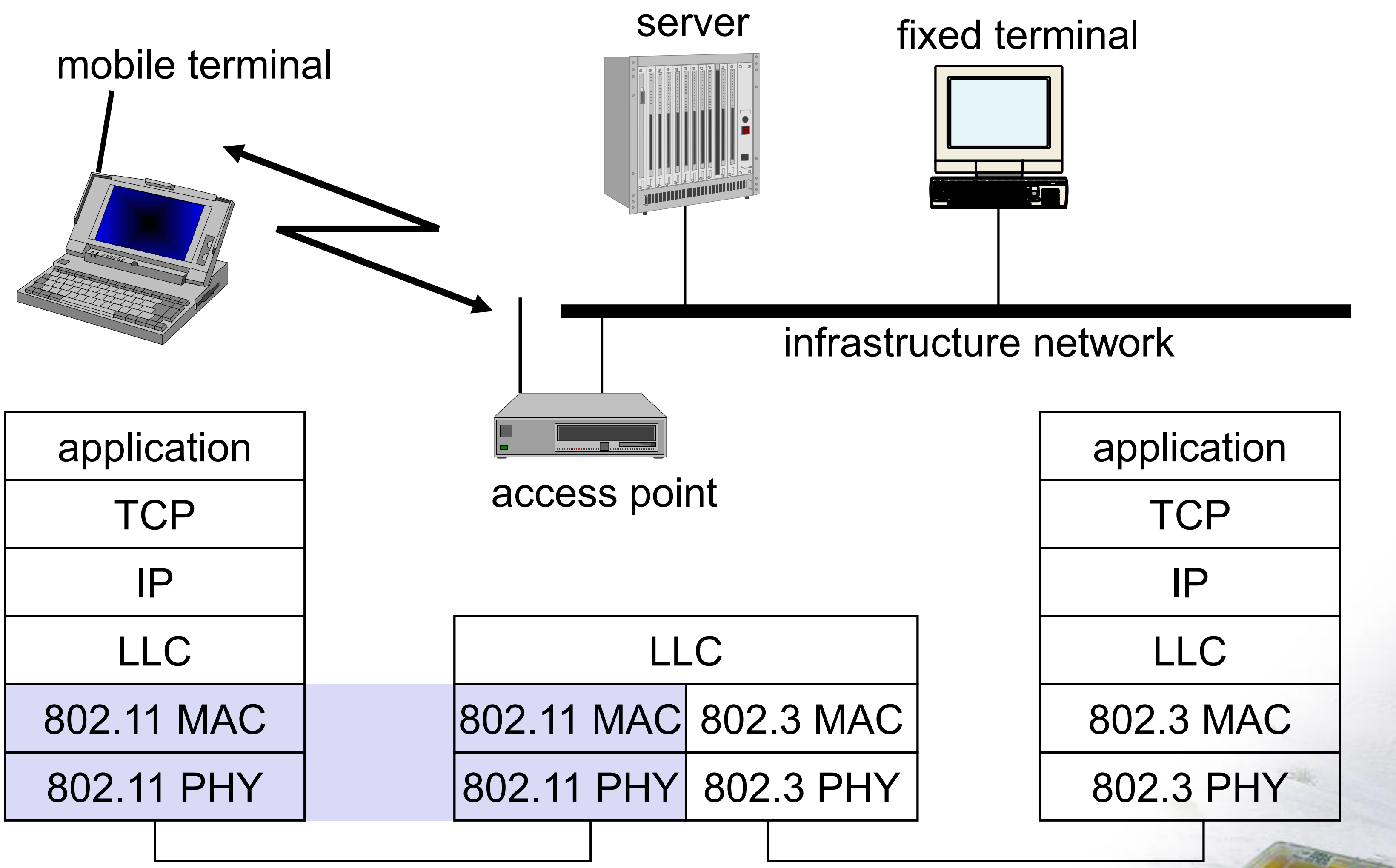
Infrastructure network

AP: Access Point

wired network

Ad-hoc network

802.11 – Protocol architecture

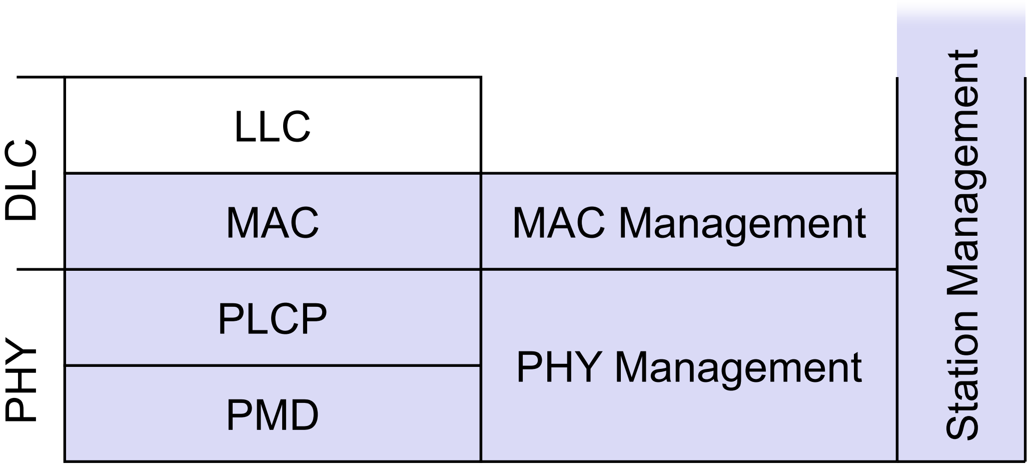


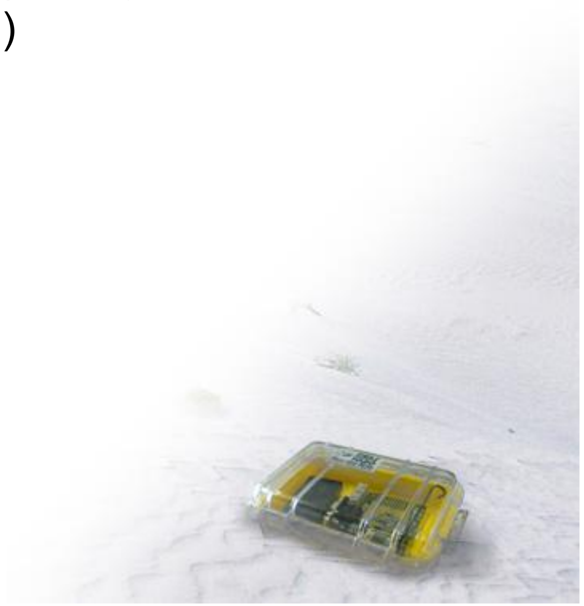
application			application
TCP			TCP
IP			IP
LLC	LLC		LLC
802.11 MAC	802.11 MAC	802.3 MAC	802.3 MAC
802.11 PHY	802.11 PHY	802.3 PHY	802.3 PHY

802.11 – The lower layers in detail

- PMD (Physical Medium Dependent)
  - modulation, coding
- PLCP (Physical Layer Convergence Protocol)
  - clear channel assessment signal (carrier sense)
- PHY Management
  - channel selection, PHY-MIB
- Station Management
  - coordination of all management functions

- MAC
  - access mechanisms
  - fragmentation
  - encryption
- MAC Management
  - Synchronization
  - roaming
  - power management
  - MIB (management information base)



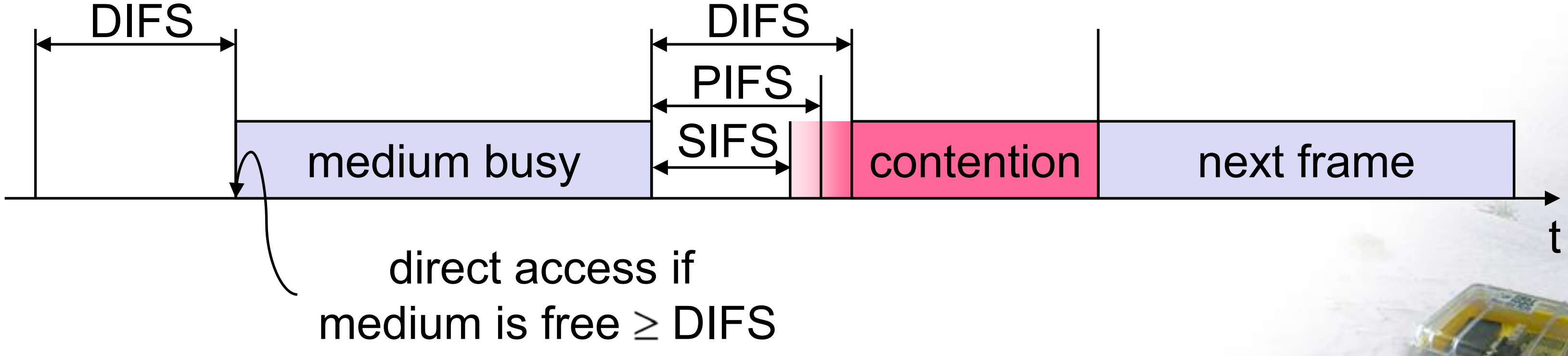


MAC layer: DFWMAC

- Traffic services
  - Asynchronous Data Service (mandatory)
    - exchange of data packets based on “best-effort”
    - support of broadcast and multicast
  - Time-Bounded Service (optional)
    - implemented using PCF (Point Coordination Function)
- Access methods
  - DFWMAC-DCF CSMA/CA (mandatory)
    - collision avoidance via binary exponential back-off mechanism
    - minimum distance between consecutive packets
    - ACK packet for acknowledgements (not used for broadcasts)
  - DFWMAC-DCF w/ RTS/CTS (optional)
    - avoids hidden terminal problem
  - DFWMAC-PCF (optional)
    - access point polls terminals according to a list

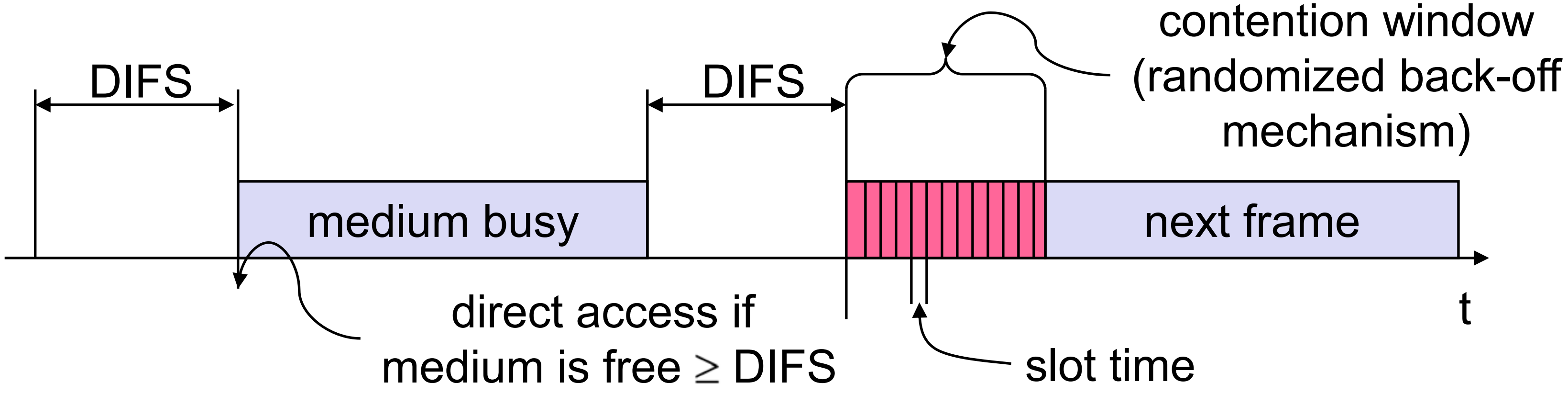
MAC layer

- defined through different inter frame spaces
- no guaranteed, hard priorities
- SIFS (Short Inter Frame Spacing)
  - highest priority, for ACK, CTS, polling response
- PIFS (PCF IFS)
  - medium priority, for time-bounded service using PCF
- DIFS (DCF, Distributed Coordination Function IFS)
  - lowest priority, for asynchronous data service



direct access if
medium is free ≥ DIFS

CSMA/CA



DIFS

DIFS

contention window (randomized back-off mechanism)

medium busy

next frame

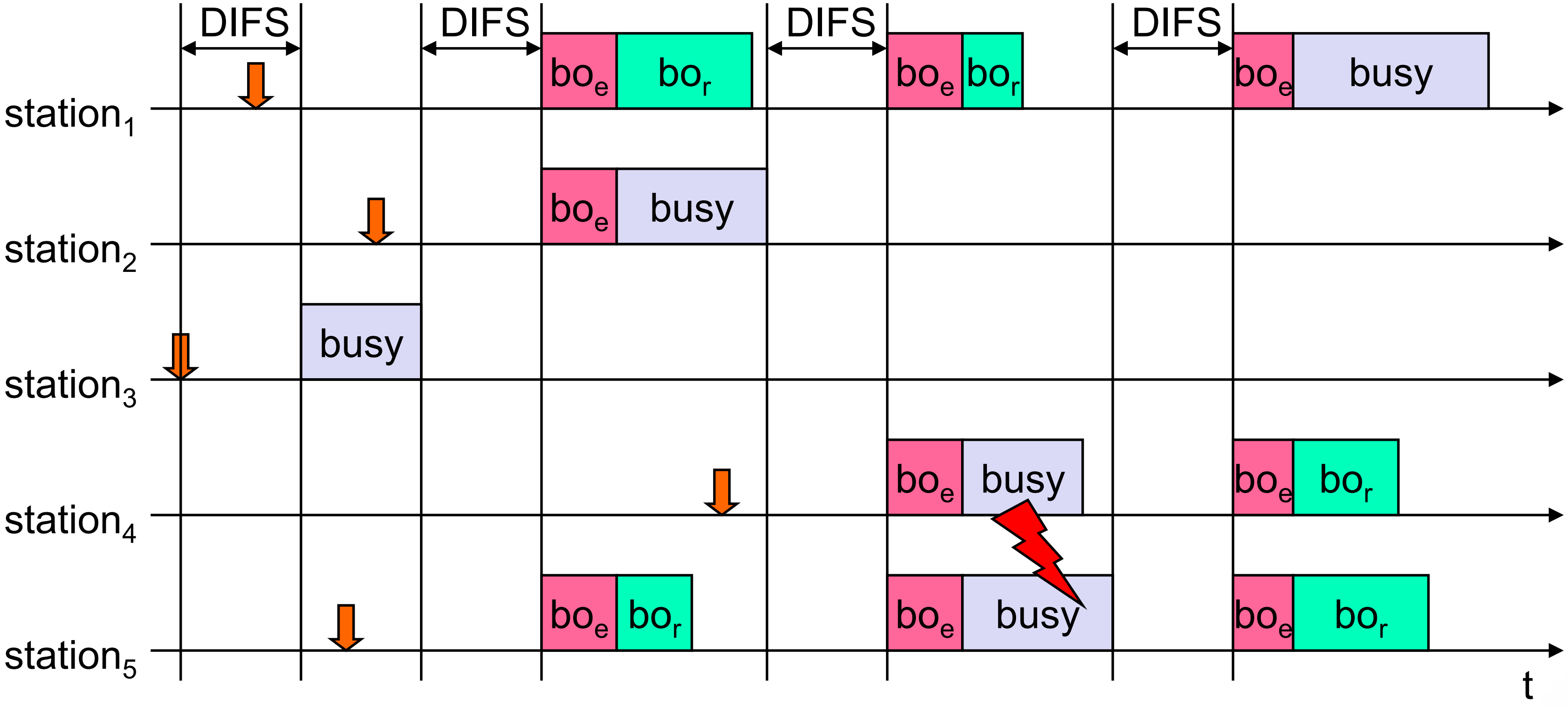
direct access if medium is free ≥ DIFS

slot time

t

- station ready to send starts sensing the medium (Carrier Sense based on CCA, Clear Channel Assessment)
- if the medium is free for the duration of an Inter-Frame Space (IFS), the station can start sending (IFS depends on service type)
- if the medium is busy, the station has to wait for a free IFS, then the station must additionally wait a random back-off time (collision avoidance, multiple of slot-time)
- if another station occupies the medium during the back-off time of the station, the back-off timer stops (fairness)

Competing stations - simple example



busy medium not idle (frame, ack etc.)

↓ packet arrival at MAC

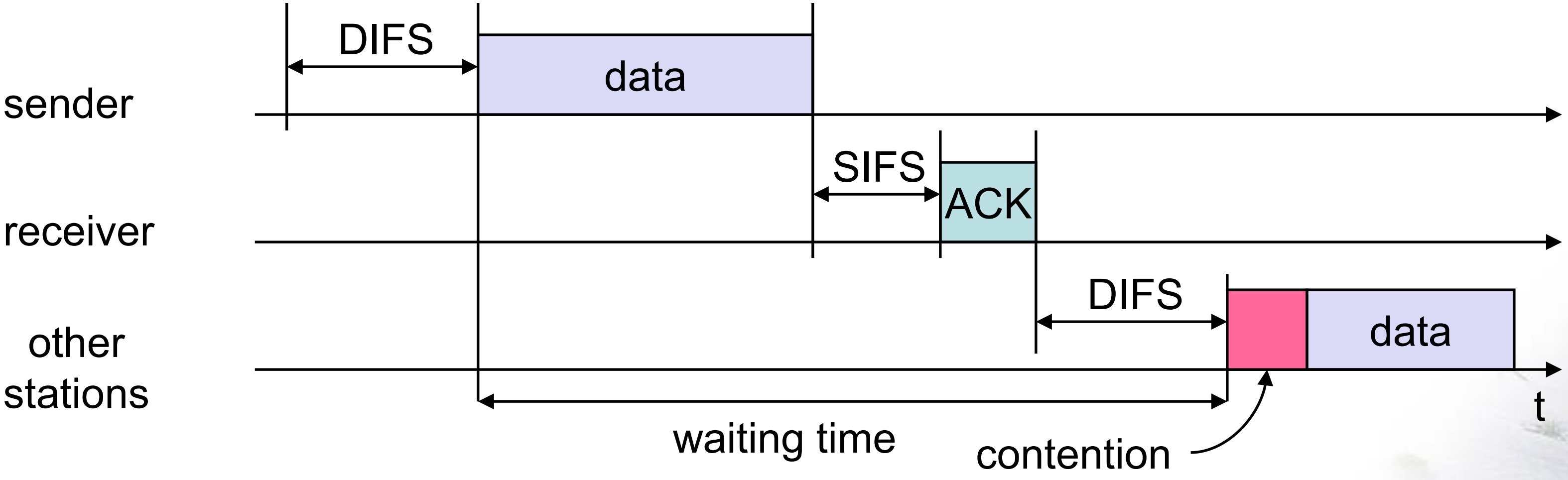
bo$_e$ elapsed backoff time

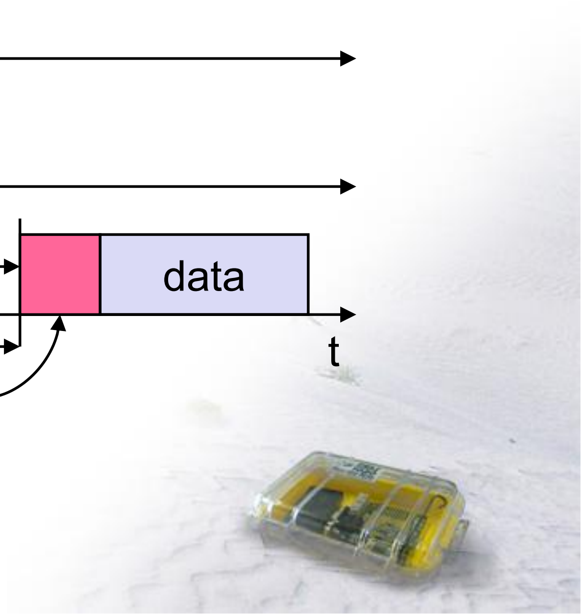
bo$_r$ residual backoff time

} backoff

CSMA/CA 2

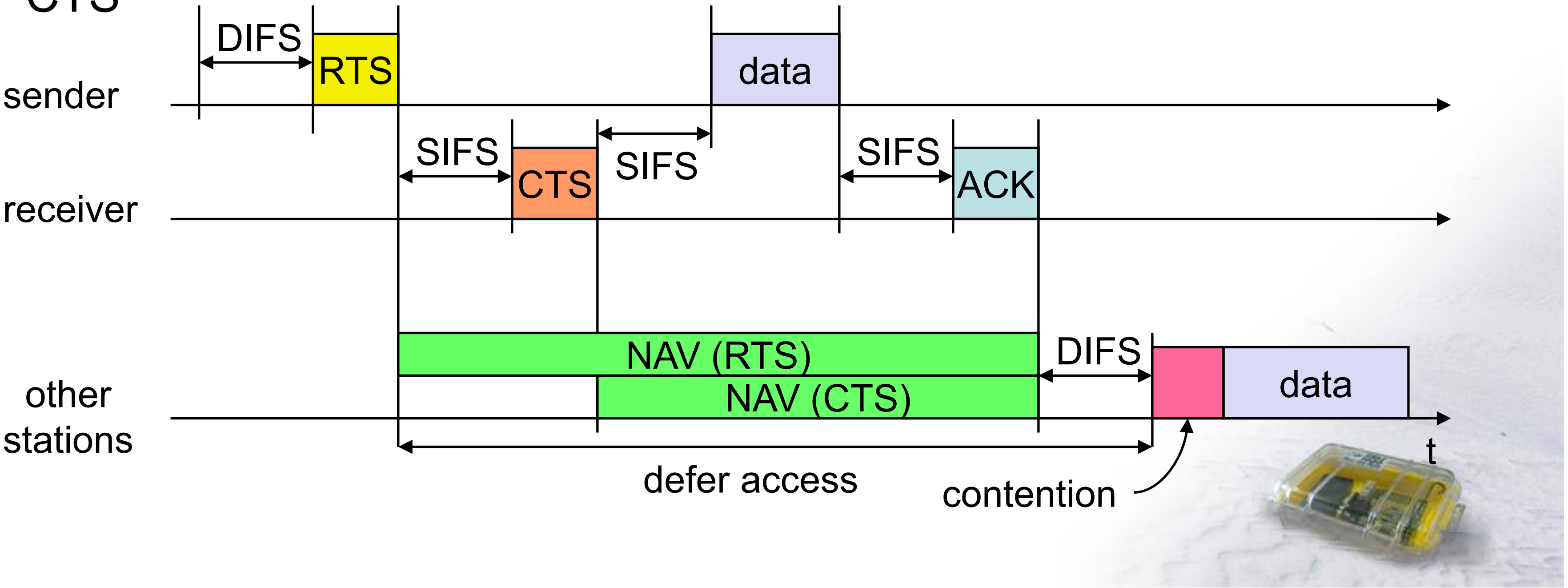
- Sending unicast packets
  - station has to wait for DIFS before sending data
  - receivers acknowledge at once (after waiting for SIFS) if the packet was received correctly (CRC)
  - automatic retransmission of data packets in case of transmission errors





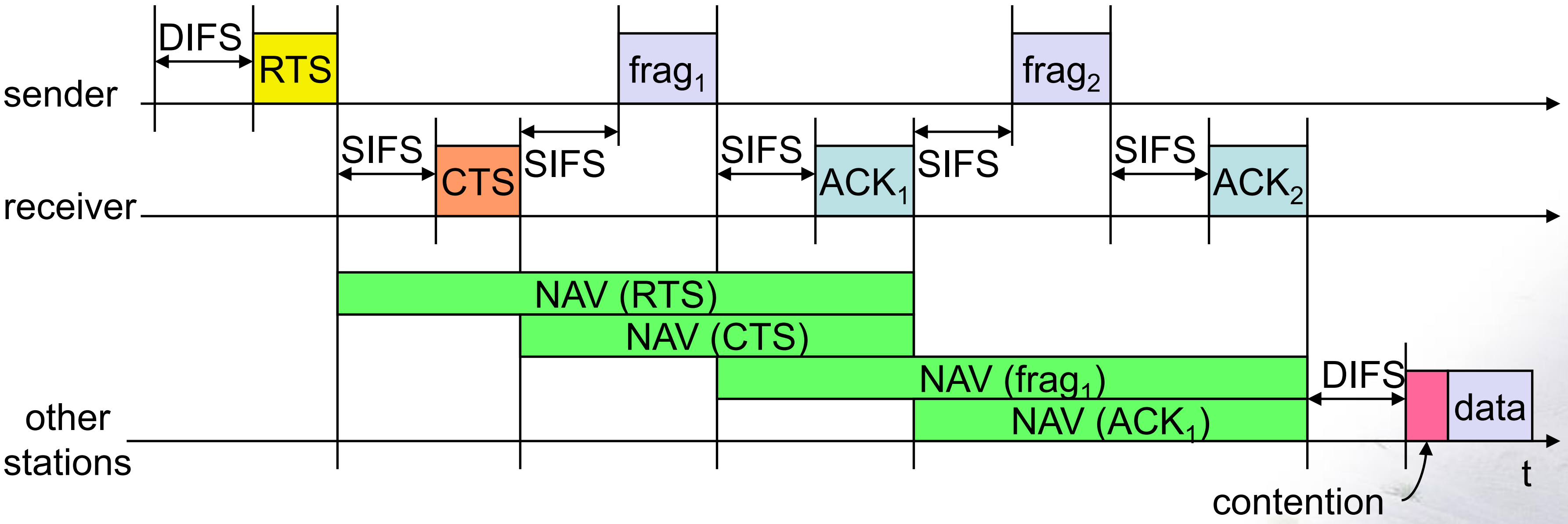
DFWMAC

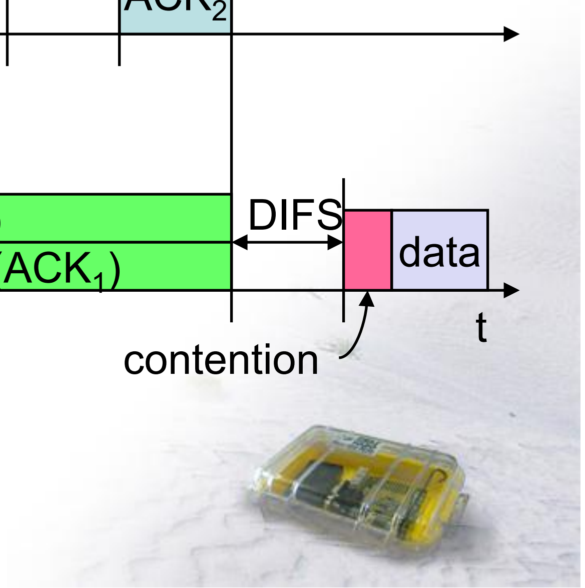
- station can send RTS with reservation parameter after waiting for DIFS (reservation determines amount of time the data packet needs the medium)
- acknowledgement via CTS after SIFS by receiver (if ready to receive)
- sender can now send data at once, acknowledgement via ACK
- other stations store medium reservations distributed via RTS and CTS



Fragmentation

- If packet gets too long transmission error probability grows
- A simple back of the envelope calculation determines the optimal fragment size





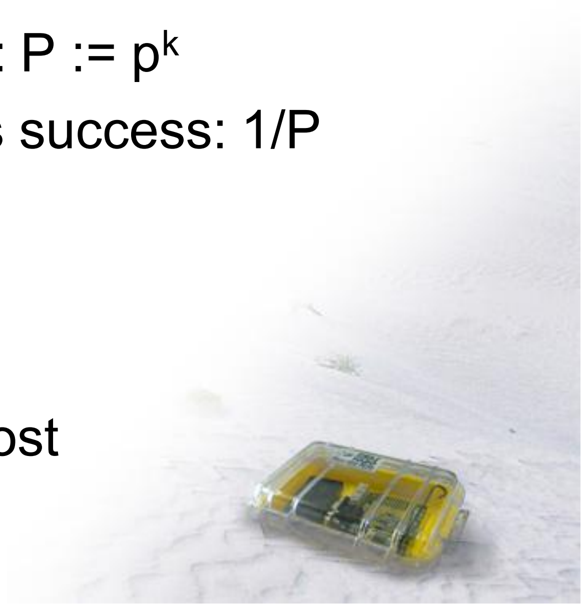
Fragmentation: What fragment size is optimal?

- Total data size: D bits
- Overhead per packet (header): h bits
- Overhead between two packets (acknowledgement): a “bits”
- We want f fragments, then each fragment has $k = D/f + h$ data + header bits

- Channel has bit error probability $q = 1-p$
- Probability to transmit a packet of k bits correctly: $P := p^k$
- Expected number of transmissions until packet is success: $1/P$

- Expected total cost for all D bits: $f \cdot (k/P+a)$

- Goal: Find a $k > h$ that minimizes the expected cost

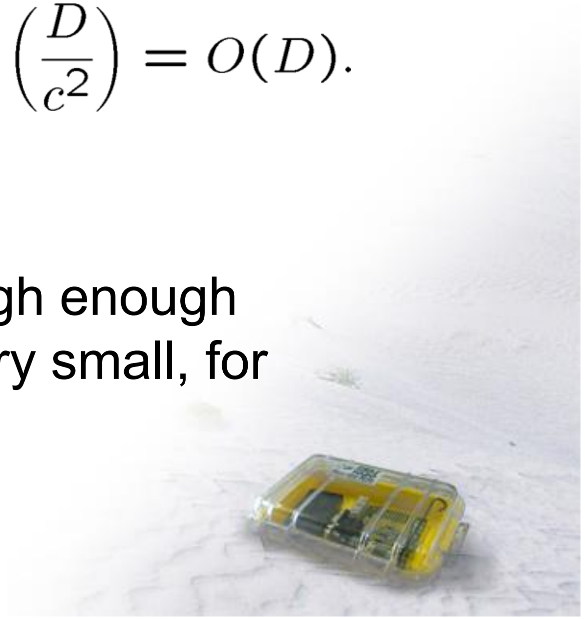


Fragmentation: What fragment size is optimal?

- For the sake of a simplified analysis we assume a = O(h)
- If we further assume that a header can be transmitted with constant probability c, that is, $p^h = c$.
- We choose k = 2h; Then clearly D = f·h, and therefore expected cost

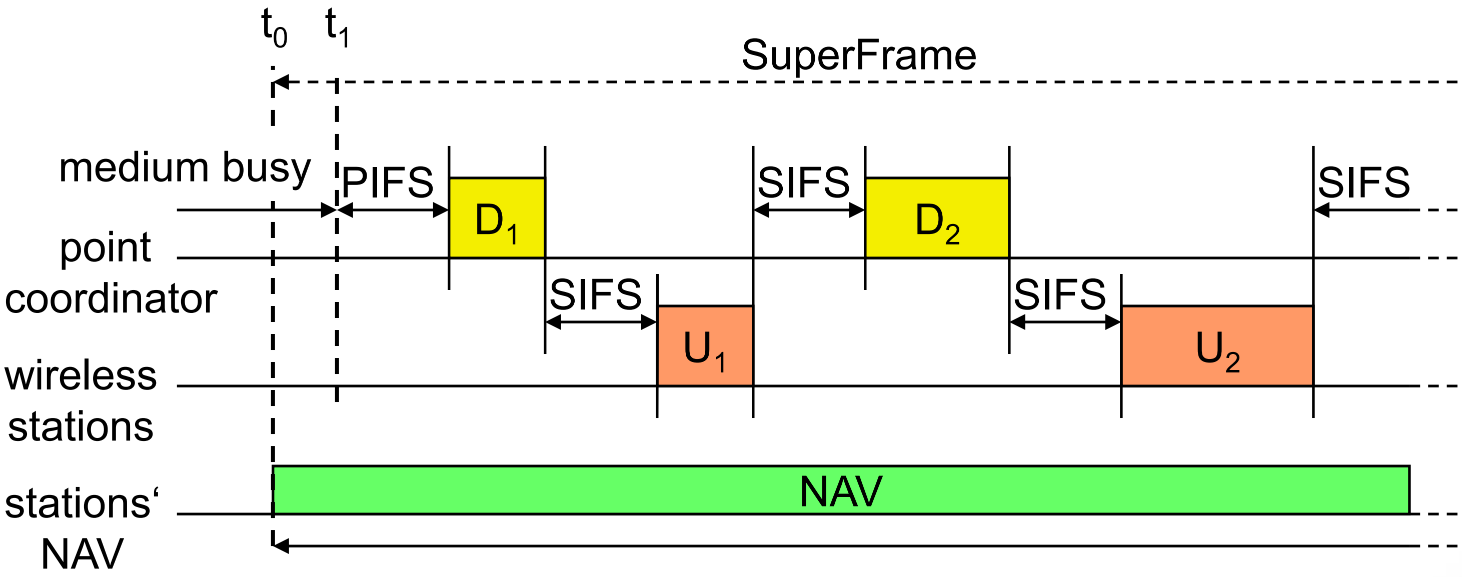
$$f \cdot \left(\frac{k}{P} + a\right) = \frac{D}{h}\left(\frac{2h}{p^{2h}} + O(h)\right) = O\left(\frac{D}{p^{h^2}}\right) = O\left(\frac{D}{c^2}\right) = O(D).$$

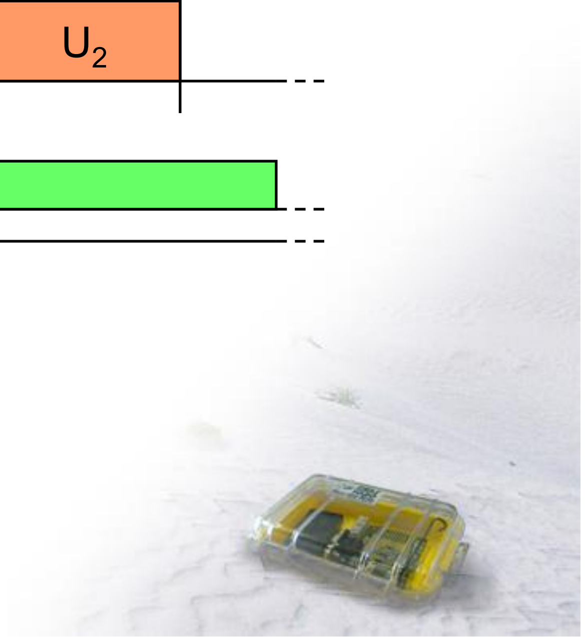
- If already a header cannot be transmitted with high enough probability, then you might keep the message very small, for example k = h + 1/q



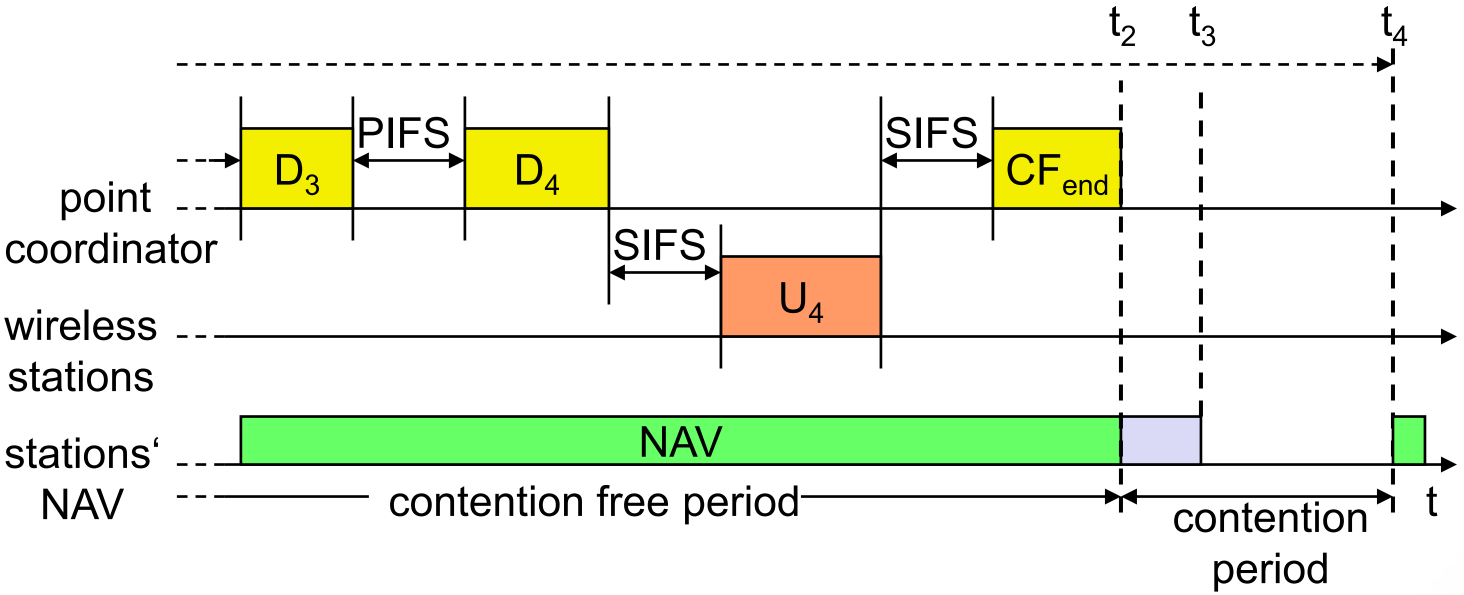
DFWMAC-PCF

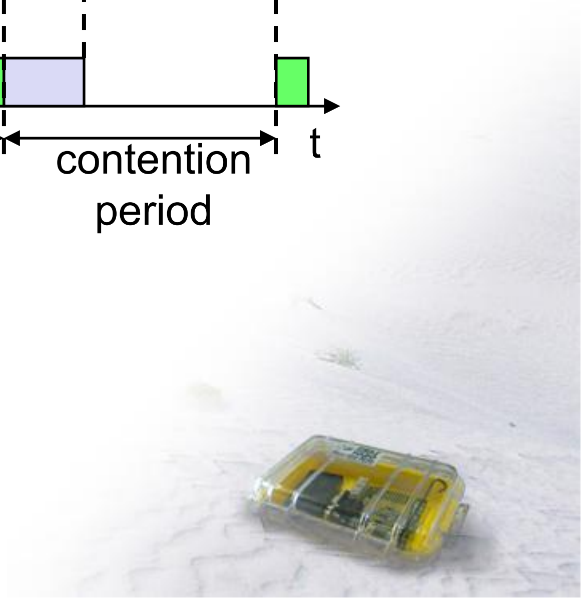
- An access point can poll stations





DFWMAC-PCF 2


t_2
t_3
t_4
D_3
PIFS
D_4
SIFS
CF_{end}
point coordinator
SIFS
U_4
wireless stations
stations' NAV
NAV
contention free period
contention period
t

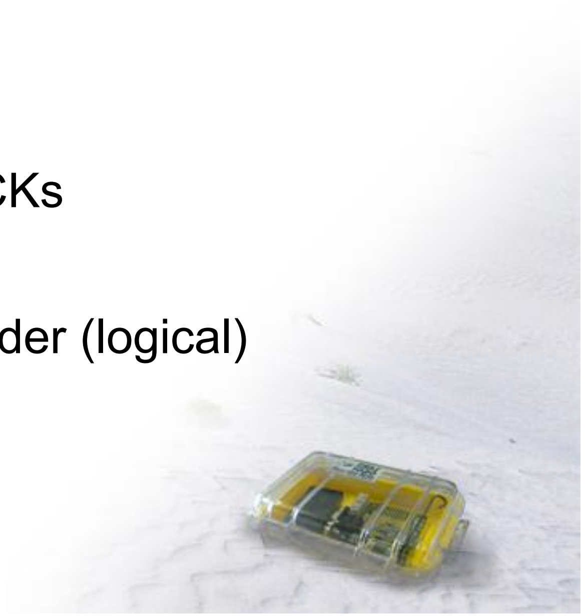


Frame format

2	2	6	6	6	2	6	0-2312	4 bytes
Frame Control	Duration ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	Data	CRC

Byte 1: version, type, subtype
Byte 2: two DS-bits, fragm., retry, power man., more data, WEP, order

- Type
  - control frame, management frame, data frame
- Sequence control
  - important against duplicated frames due to lost ACKs
- Addresses
  - receiver, transmitter (physical), BSS identifier, sender (logical)
- Miscellaneous
  - sending time, checksum, frame control, data



MAC address format

scenario	to DS	from DS	address 1	address 2	address 3	address 4
ad-hoc network	0	0	DA	SA	BSSID	-
infrastructure network, from AP	0	1	DA	BSSID	SA	-
infrastructure network, to AP	1	0	BSSID	SA	DA	-
infrastructure network, within DS	1	1	RA	TA	DA	SA

DS: Distribution System
AP: Access Point
DA: Destination Address
SA: Source Address
BSSID: Basic Service Set Identifier
RA: Receiver Address
TA: Transmitter Address

Special Frames: ACK, RTS, CTS

- Acknowledgement

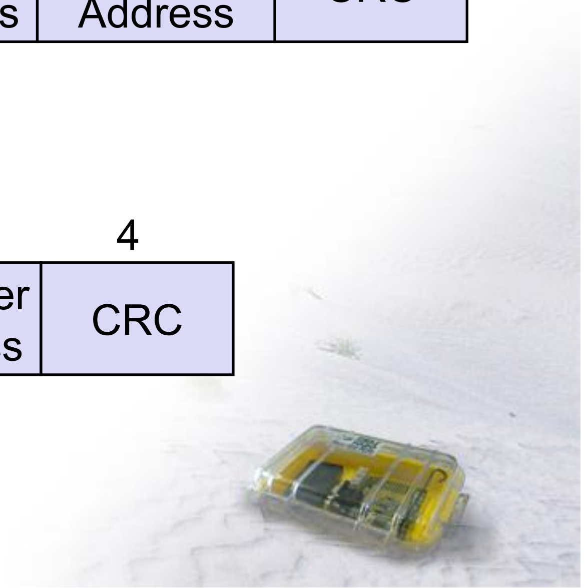
bytes	2	2	6	4
ACK	Frame Control	Duration	Receiver Address	CRC

- Request To Send

bytes	2	2	6	6	4
RTS	Frame Control	Duration	Receiver Address	Transmitter Address	CRC

- Clear To Send

bytes	2	2	6	4
CTS	Frame Control	Duration	Receiver Address	CRC

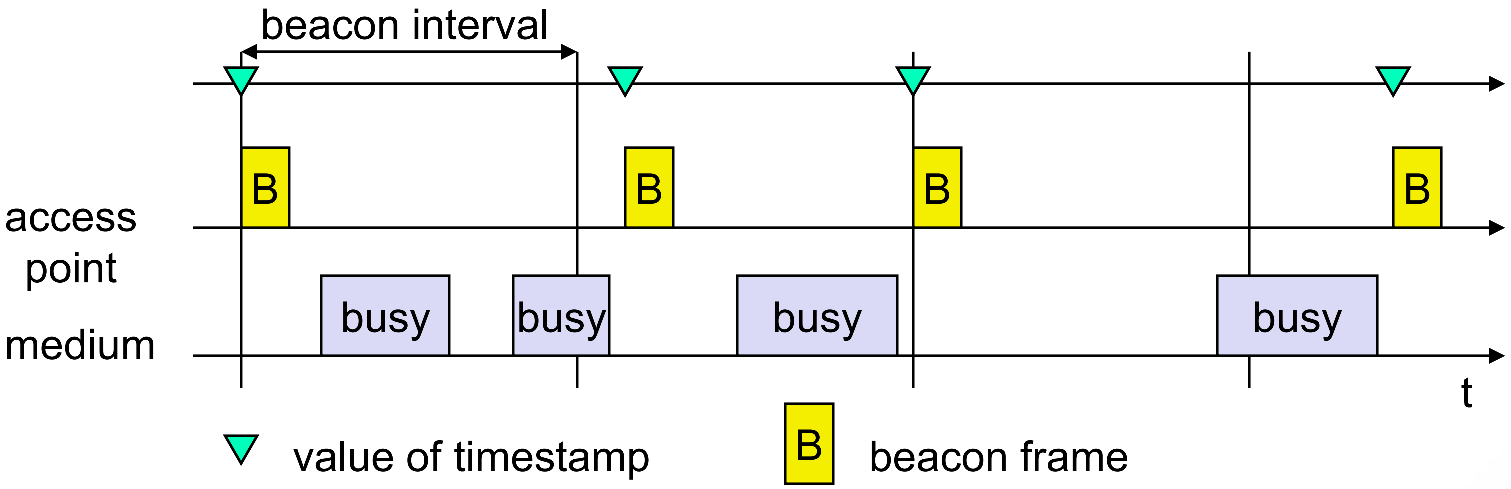


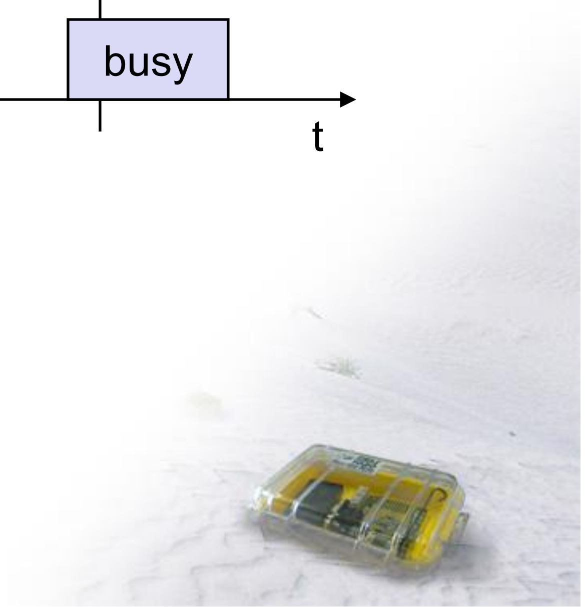
MAC management

- Synchronization
  - try to find a LAN, try to stay within a LAN
  - timer etc.
- Power management
  - sleep-mode without missing a message
  - periodic sleep, frame buffering, traffic measurements
- Association/Reassociation
  - integration into a LAN
  - roaming, i.e. change networks by changing access points
  - scanning, i.e. active search for a network
- MIB - Management Information Base
  - managing, read, write

Synchronization

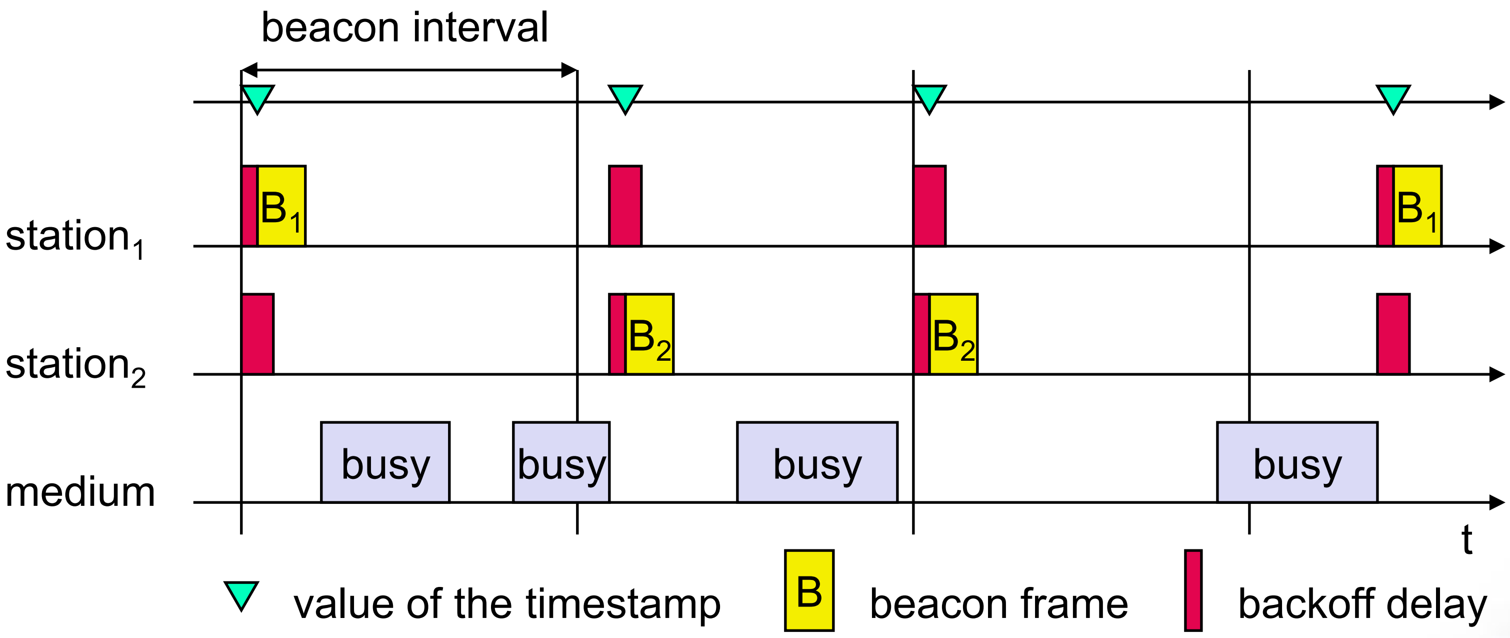
- In an infrastructure network, the access point can send a beacon





Synchronization

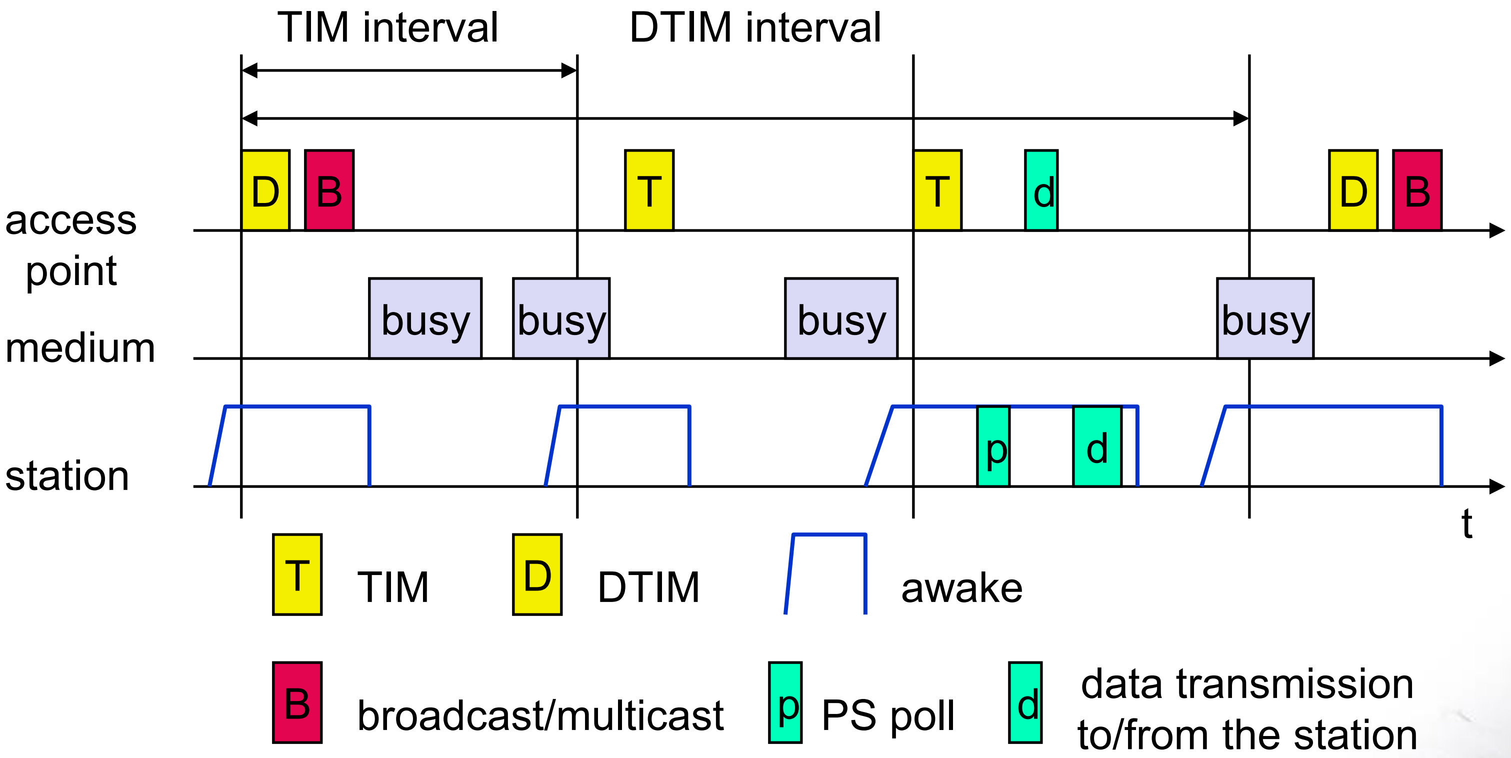
- In an ad-hoc network, the beacon has to be sent by any station

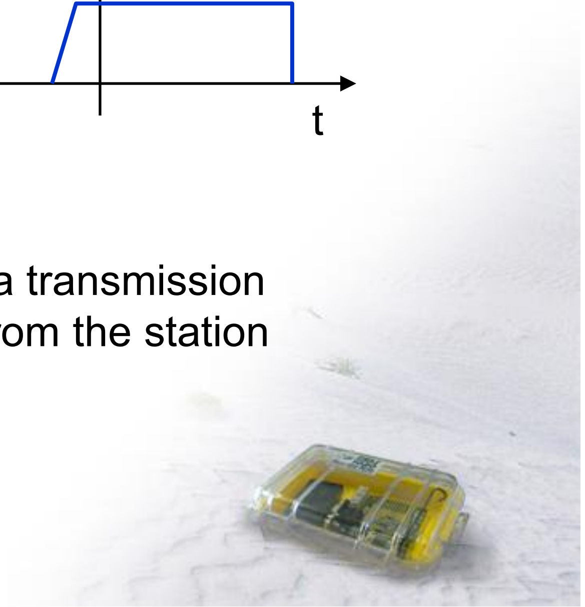


Power management

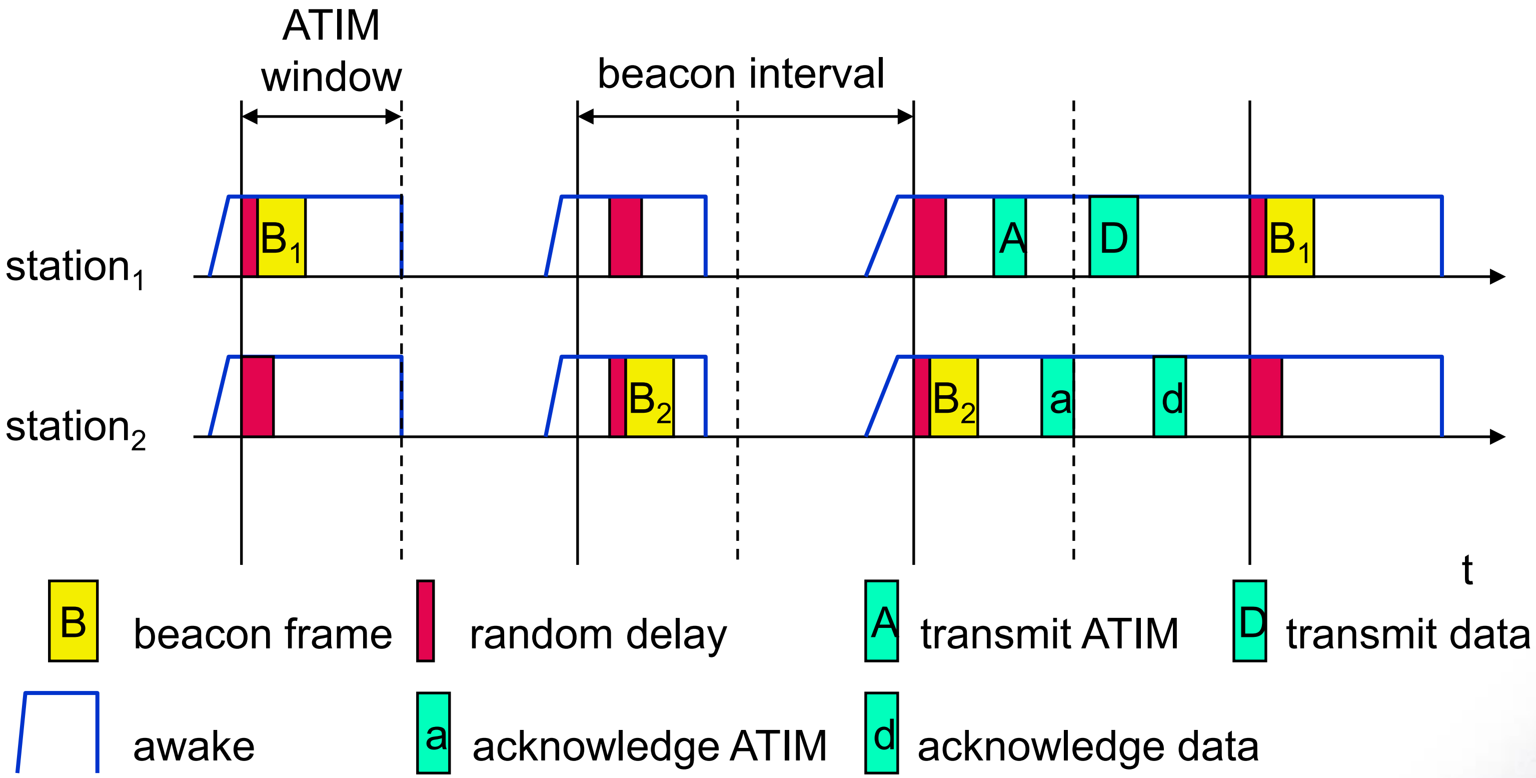
- Idea: if not needed turn off the transceiver
- States of a station: sleep and awake
- Timing Synchronization Function (TSF)
  - stations wake up at the same time
- Infrastructure
  - Traffic Indication Map (TIM)
    - list of unicast receivers transmitted by AP
  - Delivery Traffic Indication Map (DTIM)
    - list of broadcast/multicast receivers transmitted by AP
- Ad-hoc
  - Ad-hoc Traffic Indication Map (ATIM)
    - announcement of receivers by stations buffering frames
    - more complicated - no central AP
    - collision of ATIMs possible (scalability?)

Power saving with wake-up patterns (infrastructure)





Power saving with wake-up patterns (ad-hoc)



WLAN: IEEE 802.11b

- Data rate
  - 1, 2, 5.5, 11 Mbit/s, depending on SNR
  - User data rate max. approx. 6 Mbit/s
- Transmission range
  - 300m outdoor, 30m indoor
  - Max. data rate <10m indoor
- Frequency
  - Free 2.4 GHz ISM-band
- Security
  - Limited, WEP insecure, SSID
- Cost
  - Low
- Availability
  - Declining

WLAN: IEEE 802.11b

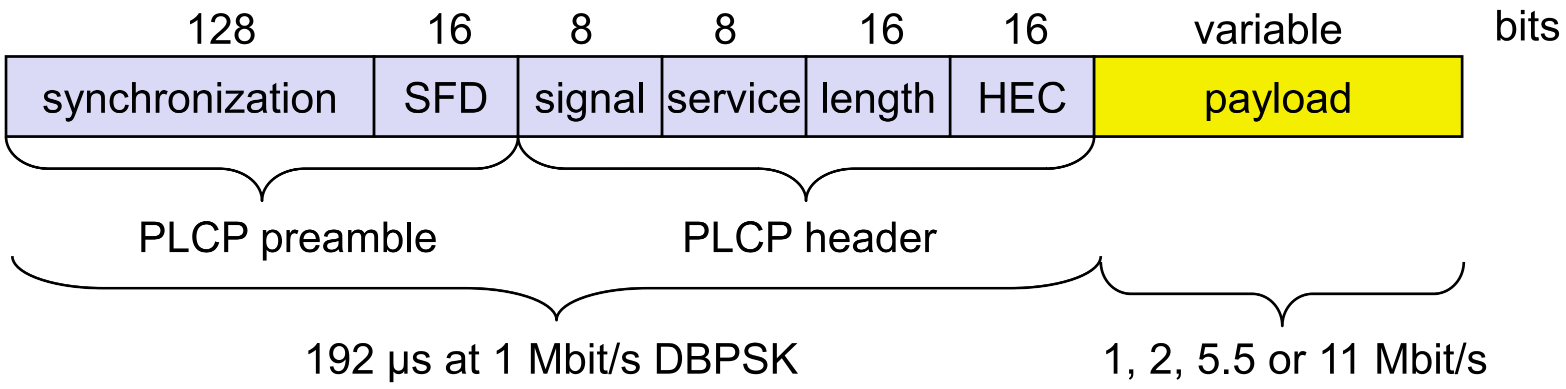
- Connection set-up time
  - Connectionless/always on
- Quality of Service
  - Typically best effort, no guarantees
  - unless polling is used, limited support in products
- Manageability
  - Limited (no automated key distribution, sym. encryption)

+ Advantages: many installed systems, lot of experience, available worldwide, free ISM-band, many vendors, integrated in laptops, simple system

– Disadvantages: heavy interference on ISM-band, no service guarantees, slow relative speed only

IEEE 802.11b – PHY frame formats

Long PLCP PPDU format



128	16	8	8	16	16	variable	bits
synchronization	SFD	signal	service	length	HEC	payload	

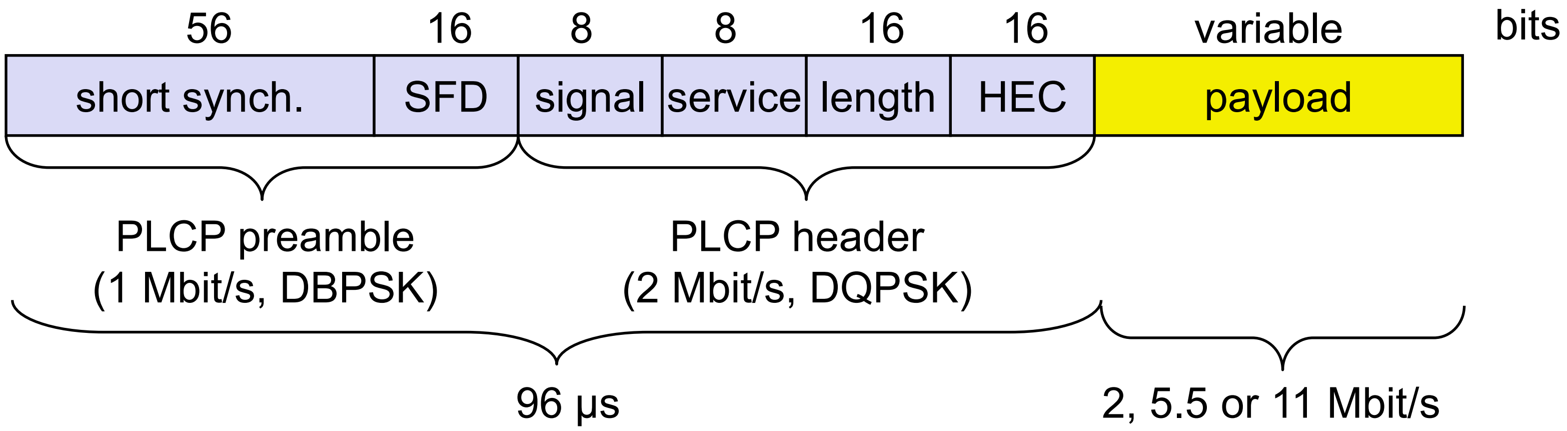
PLCP preamble

PLCP header

192 µs at 1 Mbit/s DBPSK

1, 2, 5.5 or 11 Mbit/s

Short PLCP PPDU format (optional)



56	16	8	8	16	16	variable	bits
short synch.	SFD	signal	service	length	HEC	payload	

PLCP preamble
(1 Mbit/s, DBPSK)

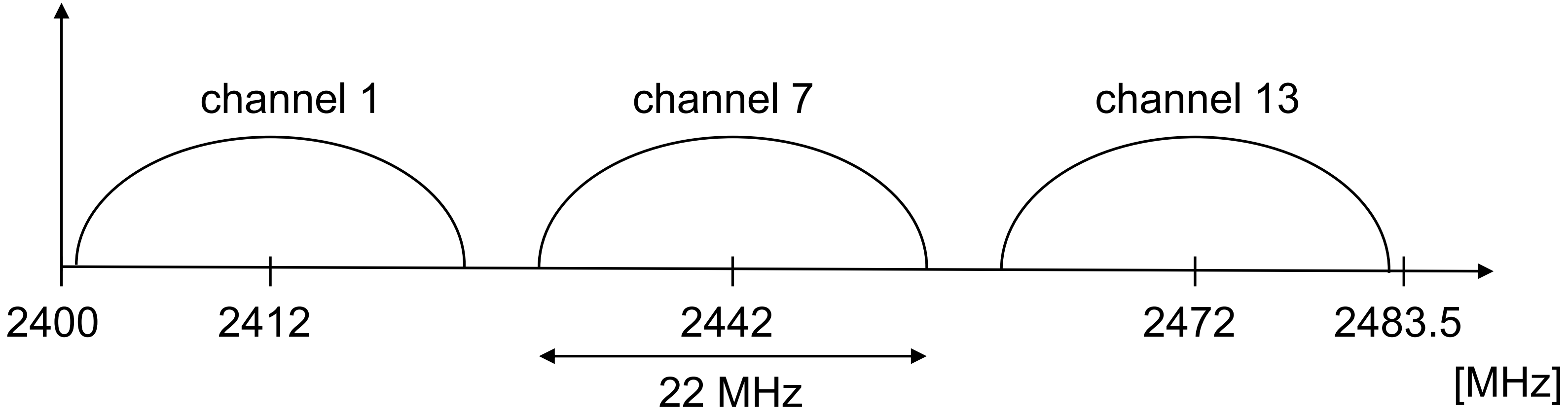
PLCP header
(2 Mbit/s, DQPSK)

96 µs

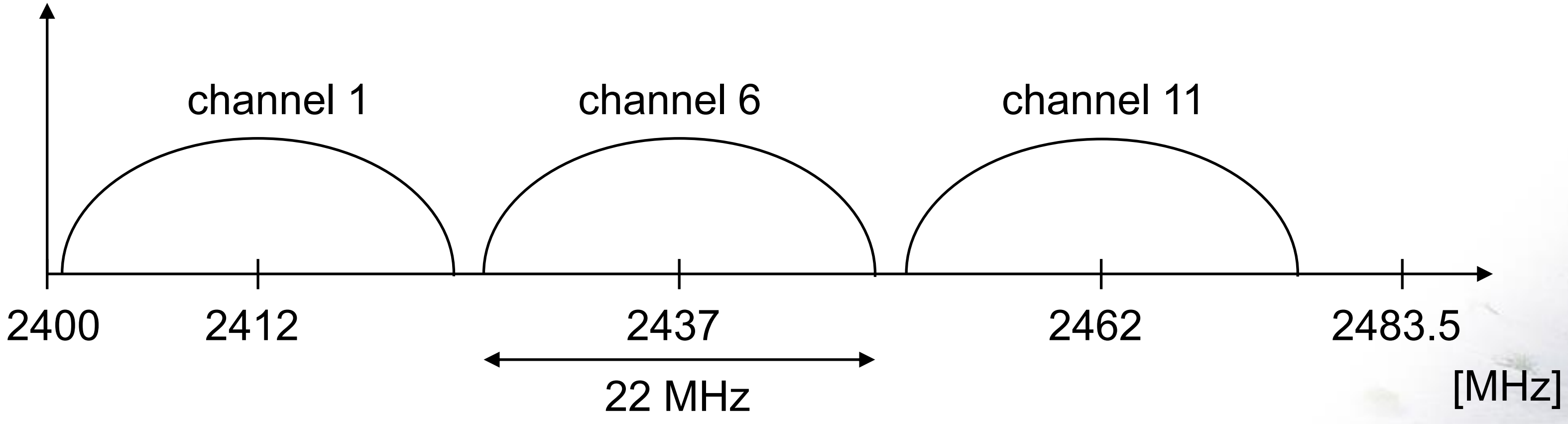
2, 5.5 or 11 Mbit/s

Channel selection (non-overlapping)

Europe (ETSI)



US (FCC)/Canada (IC)



WLAN: IEEE 802.11a

- Data rate
  - 6, 9, 12, 18, 24, 36, 48, 54 Mbit/s, depending on SNR
  - User throughput (1500 byte packets): 5.3 (6), 18 (24), 24 (36), 32 (54)
  - 6, 12, 24 Mbit/s mandatory
- Transmission range
  - 100m outdoor, 10m indoor: e.g., 54 Mbit/s up to 5 m, 48 up to 12 m, 36 up to 25 m, 24 up to 30m, 18 up to 40 m, 12 up to 60 m
- Frequency
  - Free 5.15-5.25, 5.25-5.35, 5.725-5.825 GHz ISM-band
- Security
  - Limited, WEP insecure, SSID
- Cost
  - $50 adapter, $100 base station, dropping
- Availability
  - Some products, some vendors
  - Not really deployed in Europe (regulations!)

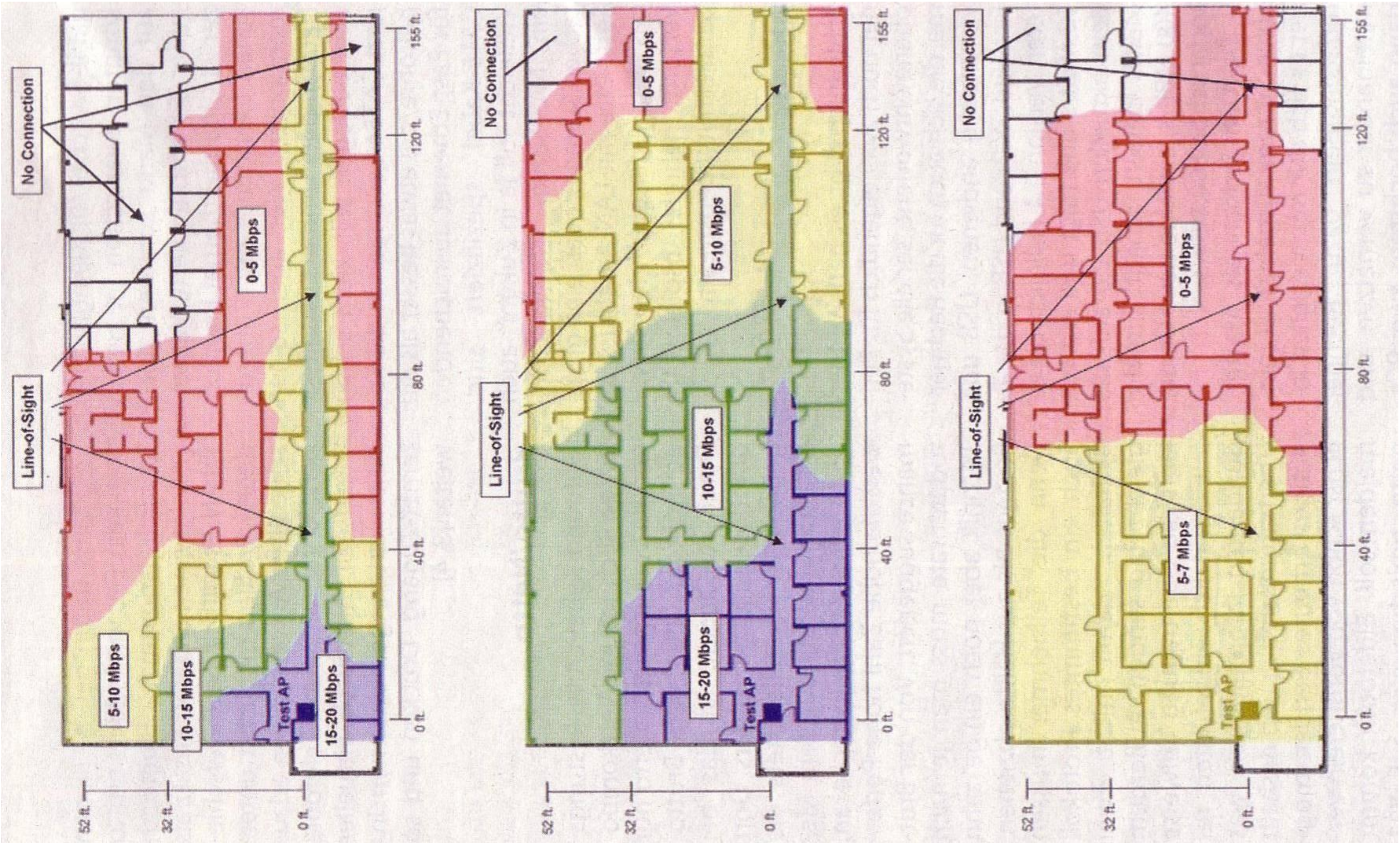
WLAN: IEEE 802.11a

- Connection set-up time
  - Connectionless/always on
- Quality of Service
  - Typically best effort, no guarantees (same as all 802.11 products)
- Manageability
  - Limited (no automated key distribution, sym. Encryption)

+ Advantages: fits into 802.x standards, free ISM-band, available, simple system, uses less crowded 5 GHz band

– Disadvantages: stronger shading due to higher frequency, no QoS

Quiz: Which 802.11 standard?



Pimp my MAC protocol

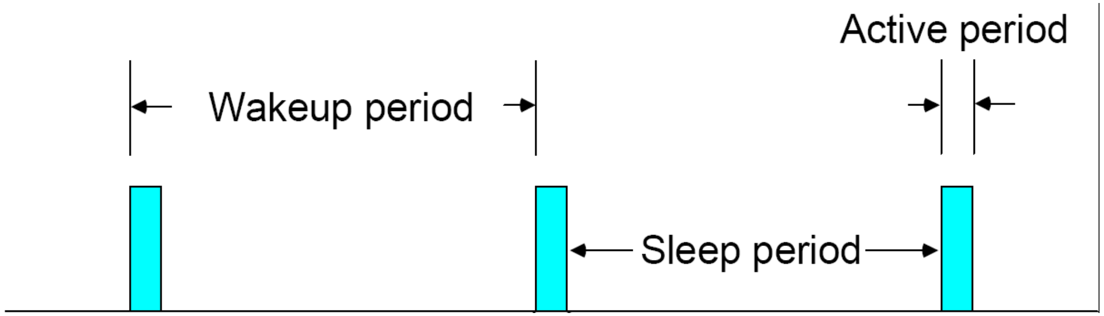
- Some general techniques to improve MAC protocols. In the following we present a few ideas, stolen from a few known protocols such as
  - S-MAC
  - T-MAC
  - B-MAC
  - Dozer
  - WiseMAC
  - RFID

- Many of the hundreds of MAC protocols that were proposed have similar ideas…

Energy vs. Delay (e.g. S-MAC)

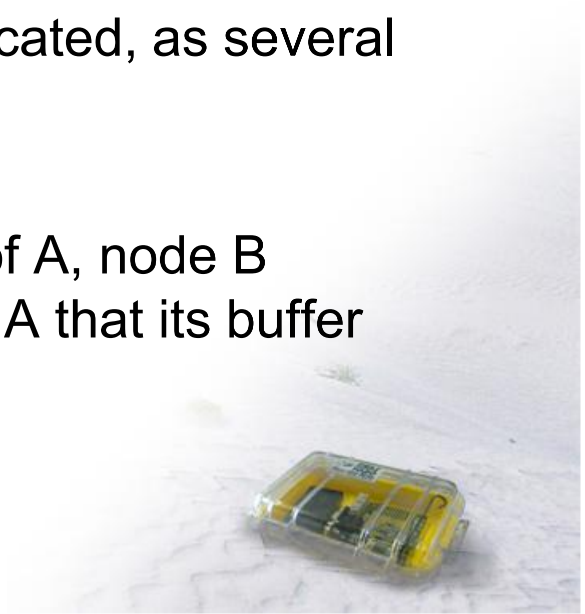
- Compute a connected dominating set (CDS)
- Nodes in the CDS choose and announce an awake schedule, and synchronize to an awake schedule of their neighbor CDS nodes.
- The other nodes synchronize to the awake schedule of their dominator (if they have more than one dominator, an arbitrary dominator can be chosen)
- Then use active periods to initiate communication (through RTS/CTS), and potentially communicate during sleep period

- Problems: Large overhead because of connecting domains, may potentially eat up a lot of the savings…



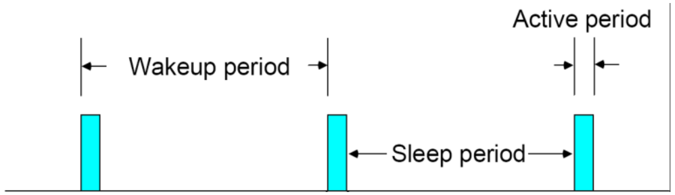
Adaptive periods (e.g. T-MAC)

- More traffic → higher duty cycles
- Control problems: Assume linked list network A → B → C. Assume that AB and BC have very low duty cycle. Now A needs to send data to C, thus increasing duty cycle of AB. Then A might send B a lot of data before B has a chance to increase duty cycle of BC.
- This is even worse when network is more complicated, as several nodes may want to start to use channel BC…
- T-MAC proposal: When receiving the next RTS of A, node B immediately answers with an RTS itself to signal A that its buffer needs to be emptied first.



Long preambles (e.g. B-MAC)

- As idle listening costs about as much energy as transmitting, we might to try to reduce idle listening. Nodes still have their sleeping cycles as before.

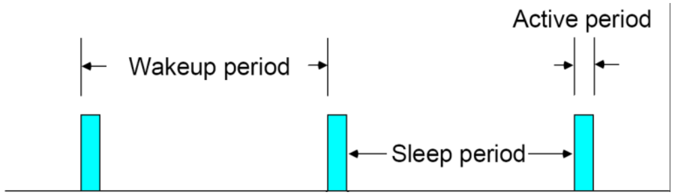


- If sender wants to transmit message, it attaches a preamble of the size of a sleep period to make sure that the receiver wakes up during preamble.

- Problem: Receiver needs to wait for whole preamble to finish, even if it wakes up early in the preamble.
  - Solution 1: Send wake-up packets instead of preamble, wake-up packets tell when data is starting so that receiver can go back to sleep as soon as it received one wake-up packet.
  - Solution 2: Just send data several times such that receiver can tune in at any time and get tail of data first, then head.

Synchronize to receiver (e.g. Dozer)

- Maybe sender knows wake-up pattern of receiver. Then it can simply start sending at the right time, almost without preamble



- Problem: How to know the wake-up pattern?
  - Dozer solution: Integrate it with higher-layer protocol, continuously exchange information, restrict number of neighbors (or align many of them to reduce information)
  - Other solutions, e.g. WiseMAC: First send long preamble; receiver then ACKs packet, and encodes its wake-up schedule in ACK for future use.

Two radios

- Nodes have two radios, a regular (high-power) radio to exchange data, and a low-power radio to sense transmissions.

- Utopia: Maybe it is even possible to send a high-power pulse over some distance which can wake up receiver (e.g. RFID)
  - Problem: Sender must be exceptionally high-power; may lead to very asymmetric design such as in RFID where the reader is orders of magnitudes larger than a passive RFID chip. This may not be feasible in ad hoc or sensor networks.

The best MAC protocol?!?

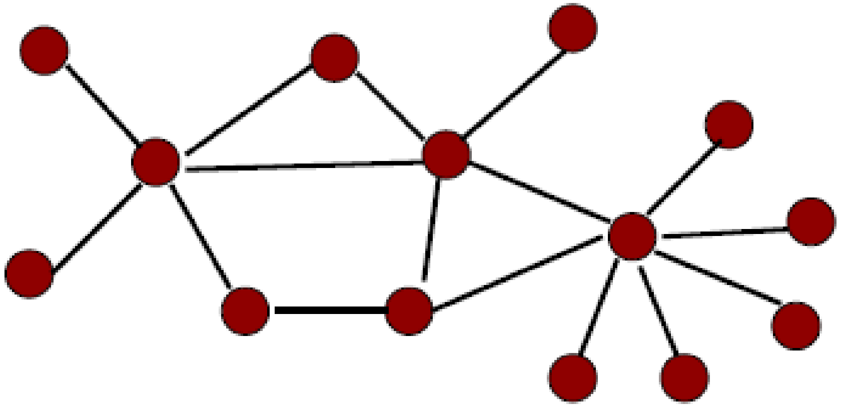
- Energy-efficiency vs. throughput vs. delay
- Worst-case guarantees vs. best-effort
- Centralized/offline vs. distributed/online
- Random topology vs. worst-case graph vs. worst-case UDG vs. …
- Communication pattern
  - Network layer: local broadcast vs. all-to-all vs. broadcast/echo
  - Transport layer: continuous data vs. bursts vs. …

- So, clearly, there cannot be a best MAC protocol!

- … but we don't like such a statement
  - We study the "broadcasting" problem

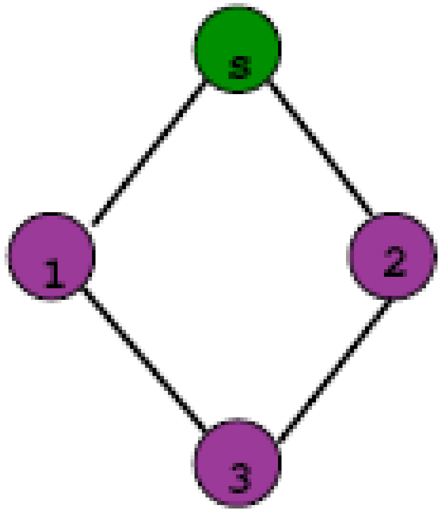
Model

- Network is an undirected graph
  - Nodes do not know topology of graph
- Synchronous rounds
  - Nodes can either transmit or receive (not both, not sleep)
- Message is received if exactly one neighbor transmits
  - No collision detection: That is, a node cannot distinguish whether 0 or 2 or more neighbors transmit
- We study broadcasting problem
  - sort of MAC layer, not quite
  - Initially only source has message
  - finally every node has message
- How long does this take?!?



Deterministic algorithms (anonymous)

- If nodes are anonymous (they have no node IDs), then one cannot solve the broadcast problem
  - For the graph on the right nodes 1 and 2 always have the same input, and hence always do the same thing, and hence node 3 can never receive the message



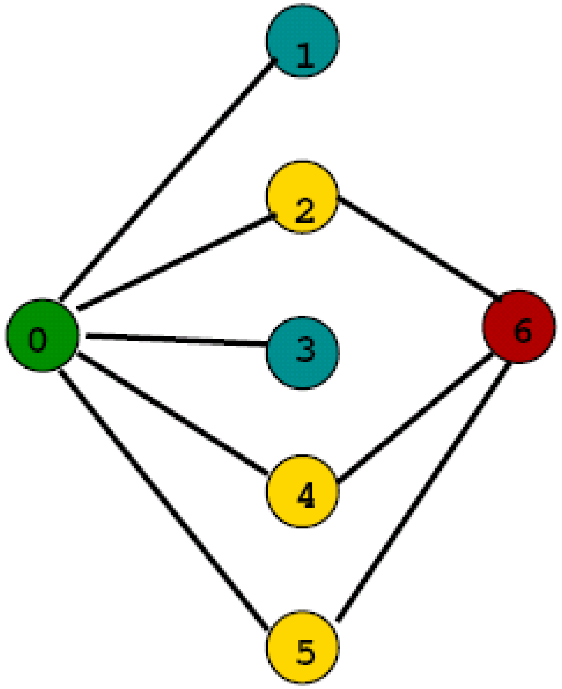
- So the nodes need IDs.

Deterministic algorithms (not anonymous)

- Consider the following network family:

- *n*+2 nodes, 3 layers
  - First layer: source node (green)
  - Last layer: final node (red)
  - Middle layer: all other nodes (*n*)
- Source connected to all nodes in middle layer
- Middle layer consists of golden and blue nodes
- Golden nodes connect to red node,
  blue nodes don't.

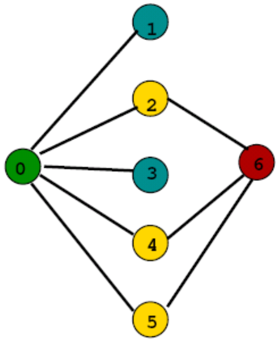
- Clearly, in one single step all middle nodes know message. But then…? (The problem is that we don't know the golden nodes!)



How to choose golden nodes?

- Task:
  - Given deterministic algorithm, e.g. *n*-1 sets M_i of nodes
  - Choose golden and blue nodes, such that no set M_i contains a single golden node.

- Construction of golden set
  - We start with golden set S being all middle nodes
  - While $\exists$ M_i such that $|M_i \cap S| = 1$ do $S := S \setminus \{M_i \cap S\}$



- Any deterministic algorithm needs at least *n* rounds
  - In every iteration a golden node intersecting with M_i is removed from S; set M_i does not have to be considered again afterwards.
  - Thus after *n*-1 rounds we still have one golden node left and all sets M_i do not contain exactly one golden node.

Improvement through randomization?

- If in each step a random node is chosen that would not help much, because a single golden node still is only found after about *n*/2 steps. So we need something smarter…

- Randomly select $n^{i/k}$ nodes, for i=0…k-1 also chosen randomly.
  - Assume that there are about $n^{s/k}$ golden nodes.
  - Then the chance to randomly select a single golden node is about

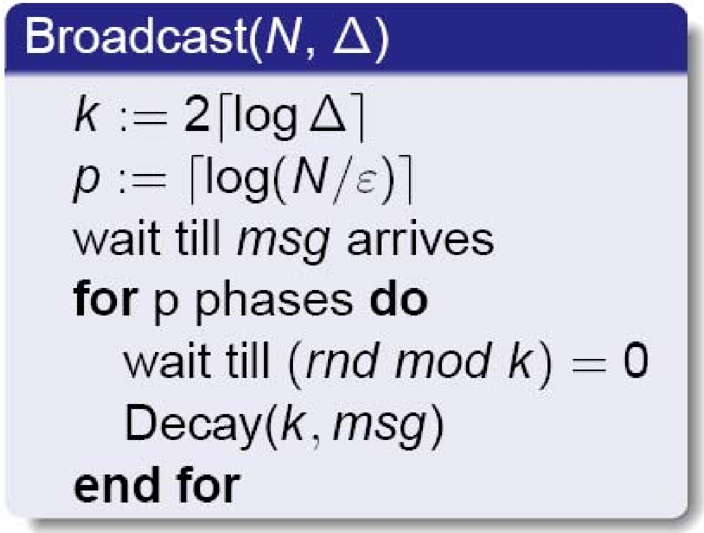
$$Pr(\text{success}) = \underbrace{n^{i/k}}_{\text{Positions for golden node}} \cdot \underbrace{n^{s/k-1}}_{\text{Probability for golden node}} \cdot \underbrace{(1 - n^{s/k-1})^{n^{i/k}-1}}_{\text{All others are not golden}}$$

  - If we are lucky and $k = i+s$ this simplifies to

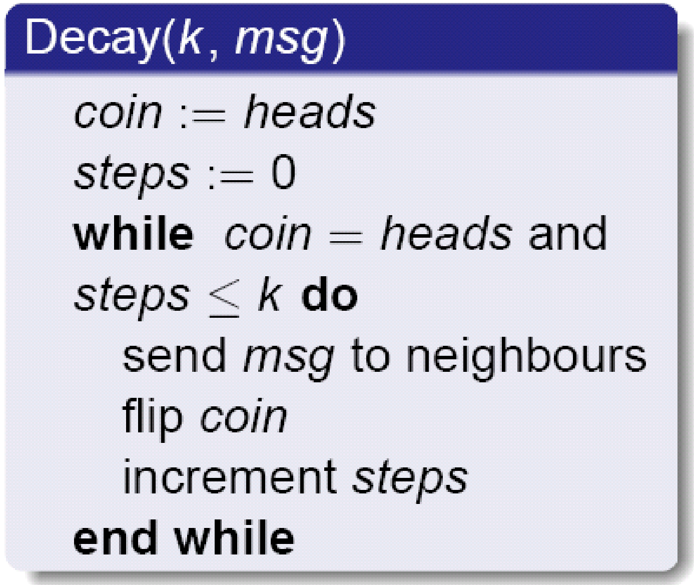
$$Pr(\text{success}) \approx 1 \cdot \left(1 - \frac{1}{n^{i/k}}\right)^{n^{i/k}} \approx 1/e$$

  - If we choose k = log n and do the computation correctly, we have polylogarithmic trials to find a single golden node.

Randomized protocol for arbitrary graphs



```
Broadcast(N, Δ)
  $k := 2\lceil \log \Delta \rceil$
  $p := \lceil \log(N/\varepsilon) \rceil$
  wait till msg arrives
  for p phases do
      wait till $(rnd \bmod k) = 0$
      Decay(k, msg)
  end for
```



```
Decay(k, msg)
  coin := heads
  steps := 0
  while coin = heads and
  steps ≤ k do
      send msg to neighbours
      flip coin
      increment steps
  end while
```

- O(D·log n + log$^2 n$)
- N: upper bound on node number
- Δ: upper bound on max degree
- ϵ: Failure probability, think $\epsilon = 1/n$
- N, Δ, ϵ are globally known
- D: diameter of graph
- Algorithm runs in synchronous phases, nodes always transmit slot number in every message

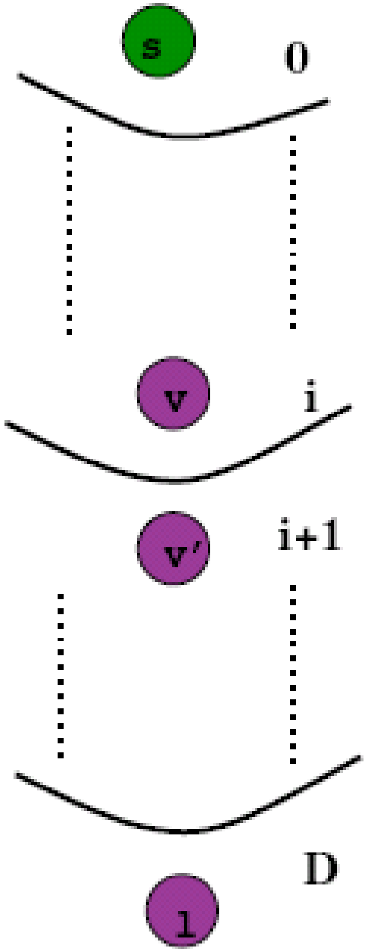
Proof overview

- During one execution of Decay a node can successfully receive a message with probability $p \geq 1/(2e)$

- Iterating *Decay* $c \cdot \log n$ times we get a very high success probability of $p \geq 1/n^c$

- Since a single execution of *Decay* takes $\log n$ steps, all nodes of the next level receive the message after $c \cdot \log^2 n$ steps (again, with very high probability).

- Having D layers a total of $O(D \cdot \log^2 n)$ rounds is sufficient (with high probability).



Proof of the first step

- During one execution of Decay a node can successfully receive a message with probability $p \geq 1/(2e)$:

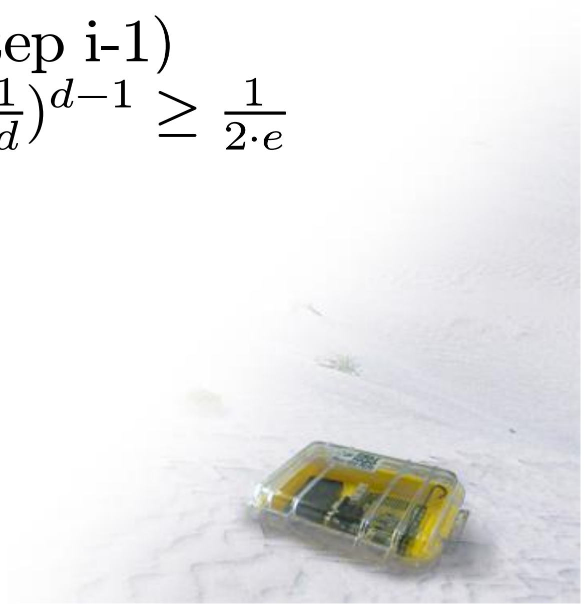
- At the start of Decay d nodes try to reach our target node. About half of them fail each step. More formally, after step i, s.t. $2^{i-1} < d \leq 2^i$

$$\frac{1}{2d} < Pr(\text{node transmits in step i-1}) = \frac{1}{2^i} \leq \frac{1}{d}$$

- And hence $Pr(\text{exactly 1 node transmits in step i-1})$

$$\geq d \cdot \frac{1}{2d} \cdot (1 - \frac{1}{d})^{d-1} \geq \frac{1}{2 \cdot e}$$

- (Step i does exist since k = 2 log Δ.)



Fastest algorithm

- Known lower bound Ω(*D*·log(*n*/*D*) + log²*n*)
- Fastest algorithm matches lower bound. Sketch of one case:

for any $k \in \{0, 1, \ldots, \log n\}$, we define

= loglog *n*

$$\alpha_k = \begin{cases} 2^{-(k+1)} & \text{for } 1 \le k \le \mathcal{LL}(n) , \\ \frac{1}{2 \log n} & \text{for } \mathcal{LL}(n) \le k \le \log n , \\ 1 - \sum_{i=1}^{\log n} \alpha_i & \text{for } k = 0 . \end{cases}$$

```
Input: Network $\mathcal{N} = (V, E)$.
        Randomized sequence $\mathfrak{I} = \langle I_1, I_2, \ldots \rangle$ such that
         $\mathbf{Pr}[I_r = k] = \alpha_k \; \forall r \in \mathbb{N}, \forall k \in \{0, 1, 2, \ldots, \log n\}$
for $r = 1$ to $T$ do { round number $r$ }
    for each active node $v \in V$ independently do
        node $v$ transmits with probability $2^{-I_r}$
```

Node that received message from source

Open Problem

- Although the MAC alphabet soup is constantly growing, the tradeoffs delay, throughput, energy-efficiency, locality, dynamics, fairness, … are still not understood. Maybe the nicest open problems are about lower bounds:

- We are looking for a non-trivial lower bound using some of the ingredients above, e.g.
  - local communication model
  - realistic model with interference, e.g. two-radii
  - some kind of edge dynamics/churn
  - and still guarantees for delay/throughput/etc.