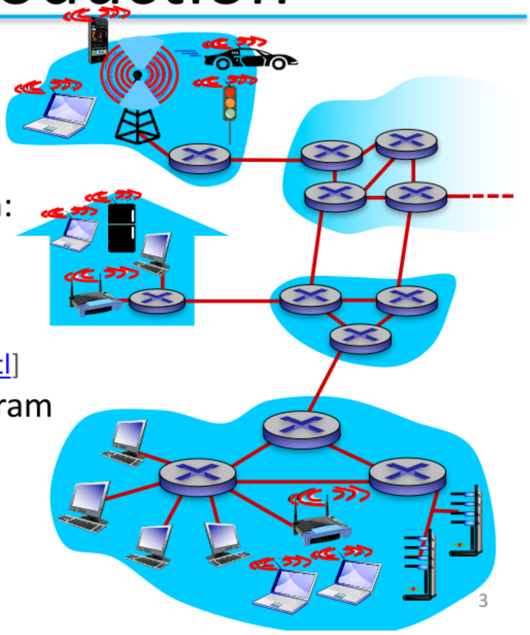
Link Layer: Introduction

Terminology:

- Hosts and routers: nodes
- Communication channels that connect adjacent nodes along communication path: links
  - Wired links
  - Wireless links
  - LANs [vs. WAN, MAN, etc. http://bit.ly/2BWmutl]
- Layer-2 packet: frame, encapsulates datagram



> Data-link layer has responsibility of transferring datagram from one node to physically adjacent node over a link

Verbindungsebene der Hardware
→ Verantwortung die Datagramme von Schicht 3 in Frames einzupacken und zum nächsten Link zu transportieren (Physikalisch benachbart).

Link Layer: Context

- Datagram transferred by different link protocols over different links: e.g.,
  - Ethernet on first link,
  - Frame relay on intermediate links,
  - 802.11 (WiFi) on last link
- Each link protocol provides different services, e.g.,
  - May or may not provide reliable data transfer over link

Transportation analogy:

- Trip from Princeton to Lausanne
  - Limo: Princeton to JFK
  - Plane: JFK to Geneva
  - Train: Geneva to Lausanne
- Tourist = datagram
- Transport segment = communication link
- Transportation mode = link layer protocol
- Travel agent = routing algorithm

Link Layer Services

- Framing, link access
  - Encapsulate datagram into frame, adding header, trailer
  - Channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, destination -- different from IP address!
- Reliable delivery between adjacent nodes
  - We learned how to do this already (chapter 4)!
  - Seldom used on low bit-error link (fiber, some twisted pair)
- Wireless links: high error rates
  - Q: Why both link-level and end-end reliability?

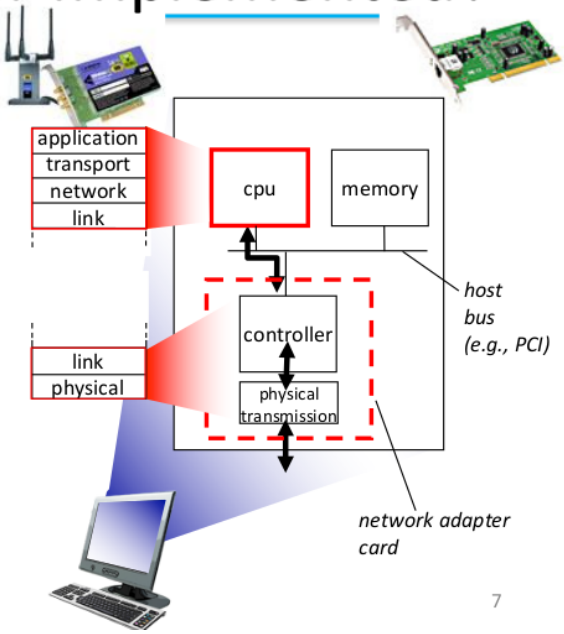
Reliable Datatransfer basiert auf einem der x (3 oben) Protokolle
– Shared Medium/Access ↔ Luft für WiFi
– Wenn nur End2End oder Link-level :: Nur Empfänger kann Fehler erkennen ↔ Alles muss neu gesendet werden.

- Flow control
  - Pacing between adjacent sending and receiving nodes
- Error detection
  - Errors caused by signal attenuation, noise
  - Receiver detects presence of errors
    - Signals sender for retransmission or drops frame
- Error correction
  - Receiver identifies and corrects bit error(s) without resorting to retransmission
- Half-duplex and full-duplex
  - With half duplex, nodes at both ends of link can transmit, but not at same time

– Flow Control: Empfänger benachrichtig zu hohe Datenrate
– Error Detection: Checksum & Bitfehler
– Half/Full-Duplex: Wann überträgt wer Daten.
→ Full: Beide
→ Half: Einer von beiden spricht zu einer Zeit (Funkgerät)

Where is the Link Layer Implemented?

- In each and every host
- Link layer implemented in “adaptor” (aka network interface card; NIC) or on a chip
  - Ethernet card, 802.11 card, Ethernet chipset
  - Implements link, physical layer
- Attaches into host’s system buses
- Combination of
  - Hardware
  - Software
  - Firmware



– Link & Physical Layer liegen zusammen in einem Adapter/Router

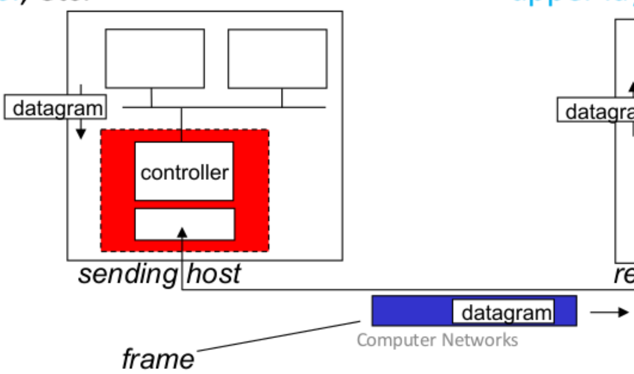
Adaptors Communicating

Sending side

- Encapsulates datagram in frame
- Adds error checking bits, rdt, flow control, etc.

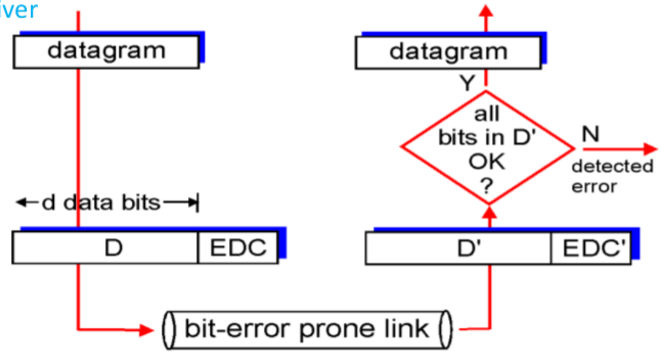
Receiving side

- Looks for errors, rdt, flow ctrl, etc.
- Extracts datagram, passes to upper layer at receiving side



Error Detection and Correction

- General idea: add redundancy used by the receiver
  - Systematic: add bits to original data
  - Non-systematic: transform original data
- Error detection
  - Parity bits/checks
  - Checksum
  - Cyclic redundancy checks (CRCs)
  - Cryptographic hash functions
  - Error correction codes (forward error correction)
- Error correction
  - Automatic repeat request (ARQ)
  - Error correction codes (forward error correction)
  - Hybrid ARQ: combination of ARQ and FEC
- Error detection and correction may not 100% reliable!
  - Protocol may miss some errors, but rarely
  - Larger EDC field yields better detection and correction



EDC = Error Detection and Correction bits (redundancy)

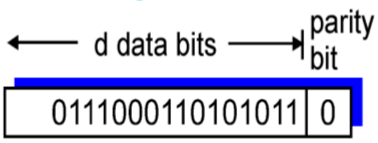
D = Data protected by error checking, may include header fields

– EDC als redundante Daten werden hinzugefügt ↔ Systematisch/Unsystematisch
– ARQ: End2End ↔ Ack/Seq, …
– Problem: Overhead

Parity Checking

Single bit parity

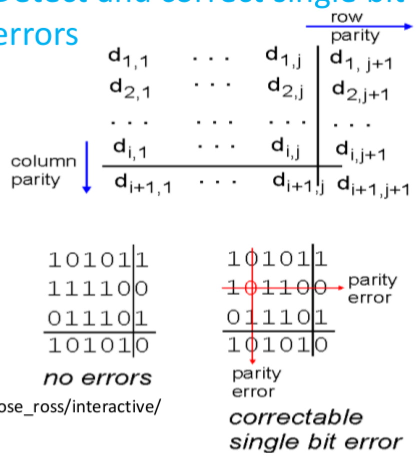
- Detect single bit errors



- Even (odd) parity scheme
  - # of 1s in d+1 bits is always even (odd)
  - Receiver checks this
  - Can detect 1 bit error
  - Insufficient for usual error-bursts

Two-dimensional bit parity

- Detect and correct single bit errors



* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Internet Checksum (review)

Goal: detect “errors” (e.g., flipped bits) in transmitted packet
(note: used at transport layer only)

Sender

- Treat segment contents as sequence of 16-bit integers
- Checksum: addition (1’s complement sum) of segment contents
- Sender puts checksum value into UDP/TCP checksum field

Receiver

- Compute checksum of received segment
- Check if computed checksum equals checksum field value
  - NO – error detected
  - YES – no error detected. But maybe errors nonetheless?

Cyclic Redundancy Check (CRC)

- More powerful error-detection coding
- View data bits, D, as a binary number
- Choose r+1 bit pattern (generator), G
- Goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - Receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - Can detect all burst errors less than r+1 bits
- Widely used in practice (Ethernet, 802.11 WiFi, ATM)

← d bits → ← r bits →

D: data bits to be sent	*R:* CRC bits

bit pattern

$D * 2^r \quad XOR \quad R$ *mathematical formula*

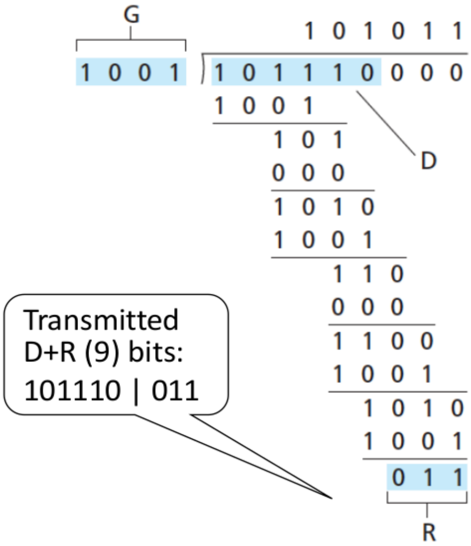
D ° R konkatiniert als 1 Binärzahl, dividiert durch Generator ↔ Wenn Rest 0 ↔ Kein Fehler.
Wenn Rest 1 ↔ Fehler

Generator G ist generalisiert.

CRC Example

- Want
  - $D*2^r$ XOR R = nG
- Equivalently
  - $D*2^r$ = nG XOR R
- Equivalently
  - If we divide $D*2^r$ by G, want remainder R to satisfy:

$$\boxed{R = remainder\frac{D * 2^r}{G}}$$

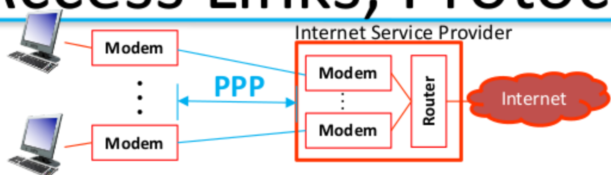


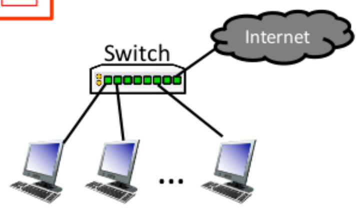
* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Multiple Access Links, Protocols

Two types of "links":

- Point-to-point
  - PPP for dial-up access
  - Point-to-point link between Ethernet switch and hosts
- Broadcast (shared wire or medium)
  - Old-fashioned Ethernet
  - Upstream hybrid fiber-coaxial (HFC)
  - 802.11 wireless LAN





Shared wire (e.g., cabled Ethernet)

Shared RF (e.g., 802.11 WiFi)

Shared RF (satellite)

Humans at a cocktail party (shared air, acoustical)

– Zur Übertragung werden Bits codiert (XOR!)
– P2P-Protokoll ↔ PPP … als Modem/Switch

Multiple Access Protocols

- Single shared broadcast channel
  - Two or more simultaneous transmissions by nodes: interference
  - Collision if node receives two or more signals at the same time
- Multiple access protocol
  - Distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
  - Communication about channel sharing must use channel itself! No out-of-band channel for coordination
- Ideal multiple access protocol
  - Given: broadcast channel of rate R bps
  - Desiderata: (Latin: "desired things")
    1. When one node wants to transmit, it can send at rate R.
    2. When M nodes want to transmit, each can send at average rate R/M
    3. Fully decentralized: no special node to coordinate transmissions, no synchronization of clocks, slots
    4. Simple

– Knoten machen sich untereinander aus wann Daten zum Access Point gesendet werden

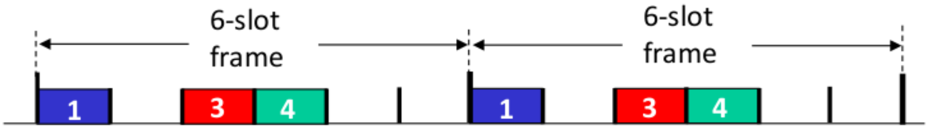
Three broad classes of MAC (Medium Access Control) protocols:

- Channel partitioning
  - Divide channel into smaller "pieces" (time slots, frequency, code)
  - Allocate piece to node for exclusive use
- Random access
  - Channel not divided, allow collisions
  - "Recover" from collisions
- "Taking turns"
  - Nodes take turns, but nodes with more to send can take longer turns

Channel Partitioning: TDMA

TDMA: time division multiple access

- Access to channel in "rounds"
- Each station gets fixed length slot (length = packet transmission time) in each round
- Unused slots go idle
- Example:
  - 6-station LAN
  - 1,3,4 have packets to send
  - slots 2,5,6 idle

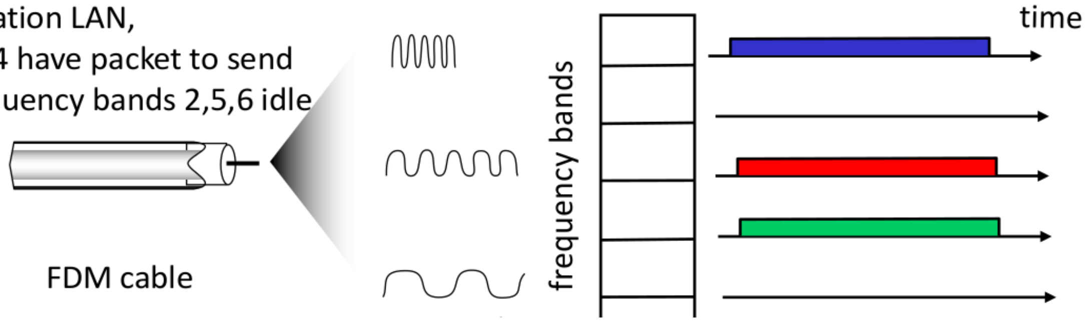


→ Kanal wird aufgeteilt, jeder kann diesen irgendwann nutzen … Es wird zeitlich in Slots aufgeteilt und jeder Knoten kann den Kanal zu einem bestimmten Zeitpunkt nutzen.

Channel Partitioning: FDMA

FDMA: frequency division multiple access

- Channel spectrum divided into frequency bands
- Each station assigned fixed frequency band
- Unused transmission time in frequency bands go idle
- Example:
  - 6-station LAN,
  - 1,3,4 have packet to send
  - Frequency bands 2,5,6 idle



Channel Partitioning: CDMA

CDMA: code division multiple access

- Senders may send simultaneously, coding helps to separate different sources
- All users share same frequency, but each user has own chipping sequence (i.e., code) to encode data
- Allows code set partitioning (if codes are orthogonal)
- Used mostly in wireless broadcast channels (cellular, satellite, etc)
- The code needs high frequency (chipping frequency)
- Encoded signal = (original data) X (chipping sequence)
- Decoding: inner-product of encoded signal and chipping sequence
- Multiple users can “coexist” with minimal interference

Random Access Protocols

- When node has packet to send
  - Transmit at full channel data rate R
  - No a priori coordination among nodes
- Two or more transmitting nodes ➜ "collision"
- Random access protocol specifies:
  - How to detect/avoid collisions
  - How to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - Slotted ALOHA, ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted Additive Links On-line Hawaii Area (ALOHA)

Pure ALOHA (1971), Slotted ALOHA (1975), Univ. of Hawaii

Assumptions

- All frames same size
- Time divided into equal size slots (time to transmit 1 frame)
- Nodes start to transmit only slot beginning
- Nodes are synchronized
- If 2 or more nodes transmit in slot, all nodes detect collision

Operation

When node obtains fresh frame, transmits in next slot

- If no collision: node can send new frame in next slot
- If collision: node retransmits frame in each subsequent slot with prob. p until success

→ Daten kommen von oberer Schicht, werden in Frame eingepackt und in einem Slot beginnt die Übertragung.

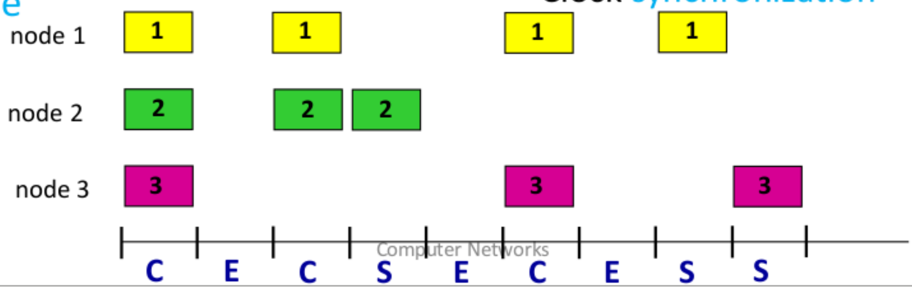
Slotted ALOHA

Pros

- Single active node can continuously transmit at full rate of channel
- Highly decentralized: only slots in nodes need to be in sync
- Simple

Cons

- Collisions, wasting slots
- Idle slots
- Nodes may be able to detect collision in less than time to transmit packet
- Clock synchronization



1. Kollission ↔ Mit WSK P wird per Packet gewartet

Slotted ALOHA: Efficiency

Efficiency: long-run fraction of successful slots (many nodes, all with many frames to send)

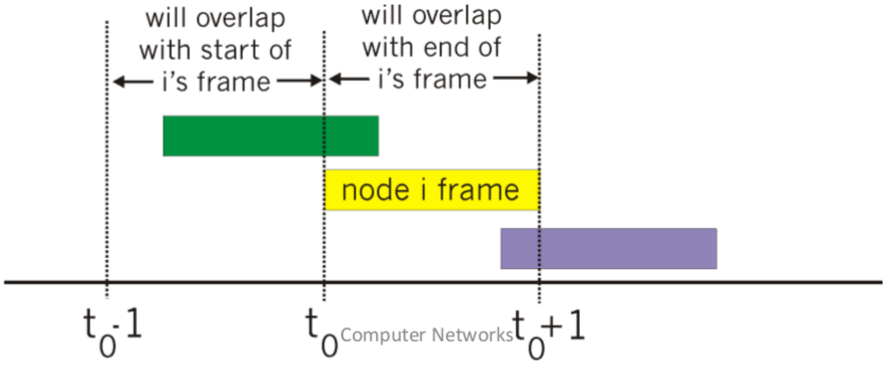
- Suppose: N nodes with many frames to send, each transmits in slot with probability p
- Probability that given node has success in a slot $= p(1-p)^{N-1}$
- Probability that any node has a success $= Np(1-p)^{N-1}$
- Max efficiency: find p* that maximizes $Np(1-p)^{N-1}$
- For many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:
  max efficiency = 1/e = .37

At best: channel used for useful transmissions 37% of time!

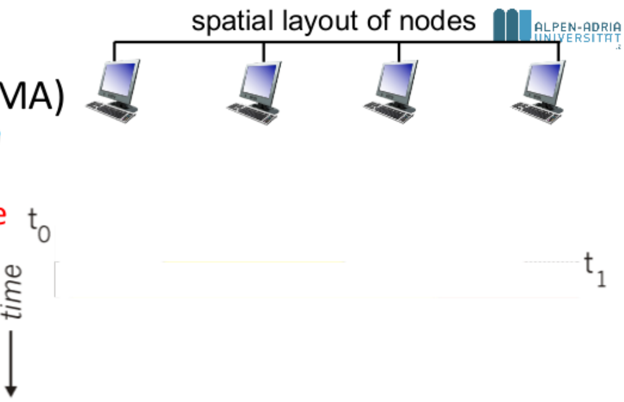
!

Pure (unslotted) ALOHA

- Unslotted Aloha: simpler, no synchronization
- When frame first arrives, transmit immediately
- Collision probability increases:
  - Frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$
- Efficiency: half of that of slotted ALOHA: 18%



Carrier Sense Multiple Access (CSMA)

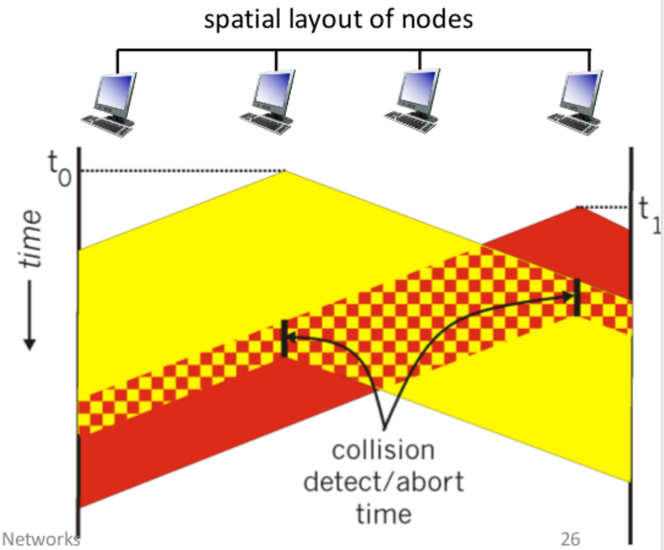


- CSMA: listen before transmit
  - If channel sensed idle: transmit entire frame
  - If channel sensed busy, defer transmission
  - Human analogy: don't interrupt others!
- Collisions can still occur
  - Propagation delay means two nodes may not "hear" each other's transmission
  - Entire packet transmission time wasted
- Distance & propagation delay play role in determining collision probability

Spatial Layout of nodes:
t0: 2 Sendet
t1: 4 sendet ↔ Modem merkt dass Kanal belegt ist
↔ Sendungen kollidieren ab t2 → Siehe Collision Detection
… Alle PCs hören mit ↔ Carrier Sense

CSMA/CD (Collision Detection)

- CSMA/CD: carrier sensing, deferral as in CSMA
  - Collisions detected within short time
  - Colliding transmissions aborted, reducing channel wastage
- Collision detection:
  - Easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - Difficult in wireless LANs: received signal strength overwhelmed by local transmission strength
  - Human analogy: the polite conversationalist



Die Gesamte Zeit der Übertragung, ab der Kollition, ist verloren. Erkennung ist erst ab eingezeichnetem Punkt.

CSMA/CD Efficiency

- T_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

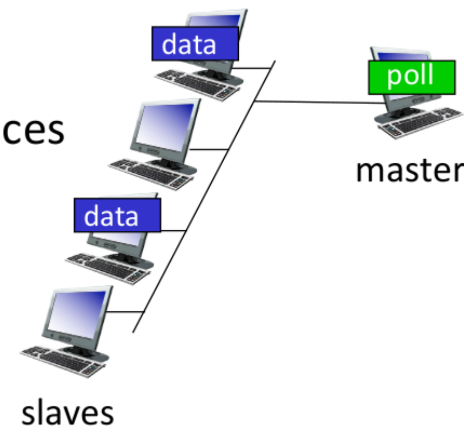
$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

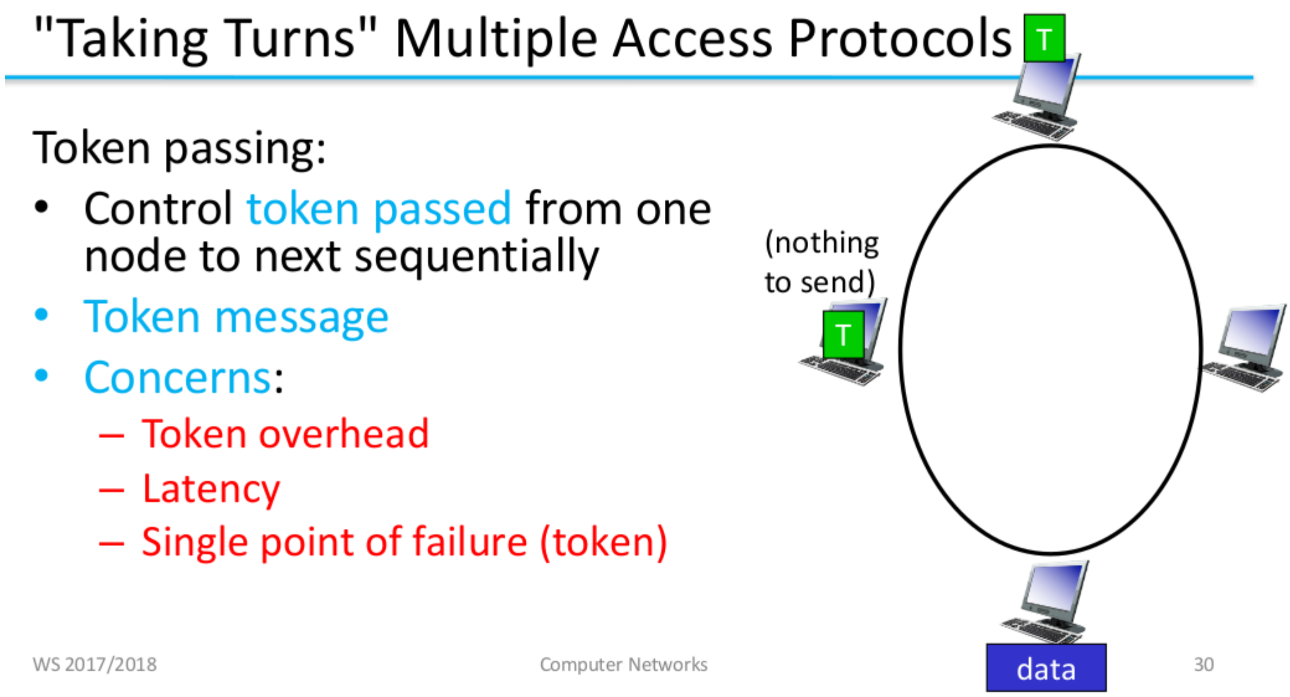
- Efficiency goes to 1
  - as t_{prop} goes to 0
  - as t_{trans} goes to infinity
- Better performance than ALOHA: and simple, cheap, decentralized!

"Taking Turns" Multiple Access Protocols

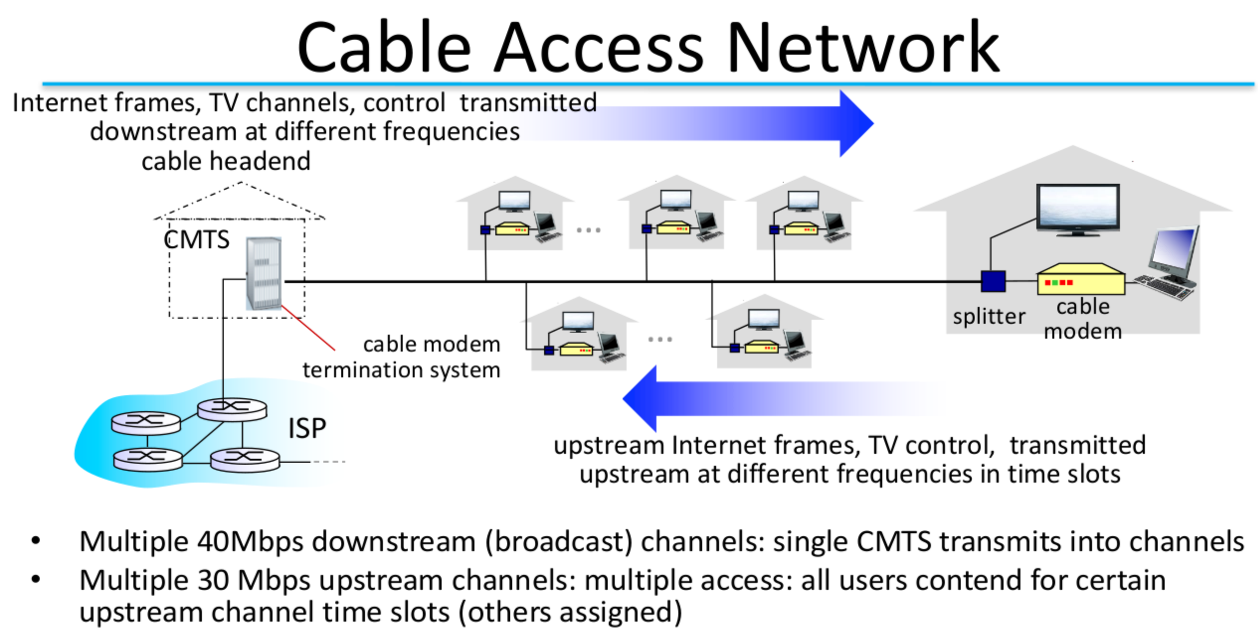
Polling:

- Master node "invites" slave nodes to transmit in turn
- Typically used with "dumb" slave devices
- Concerns:
  - Polling overhead
  - Latency
  - Single point of failure (master)



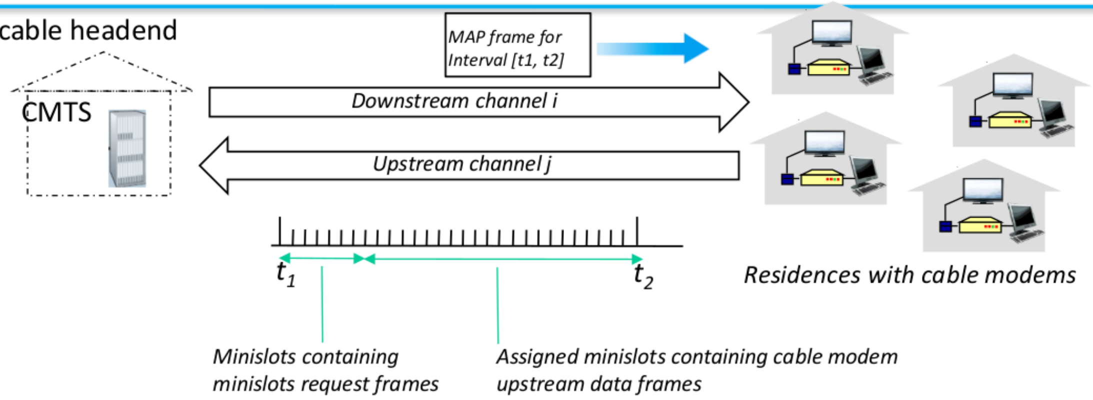


Masterknoten fragt ab: Gibt es etwas zu senden?



z.B. UPC
→ Upstream ist schwieriger zu Realisieren
→ Wann darf der User den Kanal nutzen?

Cable Access Network



DOCSIS: data over cable service interface specification

- FDM over upstream, downstream frequency channels
- TDM upstream: some slots assigned, some have contention
  - Downstream MAP frame: assigns upstream slots
  - Request for upstream slots (and data) transmitted random access (binary backoff) in selected slots

Summary of Multiple Access Protocols

- Channel partitioning, by time, frequency or code
  - Time Division, Frequency Division, Code Division
- Random access (dynamic)
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - Carrier sensing: easy in some technologies (wire), hard in others (wireless)
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11
- Taking turns
  - Polling from central site, token passing
  - Bluetooth, Fiber Distributed Data Interface (FDDI), token ring