

Introduction to Networks and the Internet

CMPE 80N

Winter 2004

Lecture 7



Layer 2: Data Link Layer



Data Link Layer



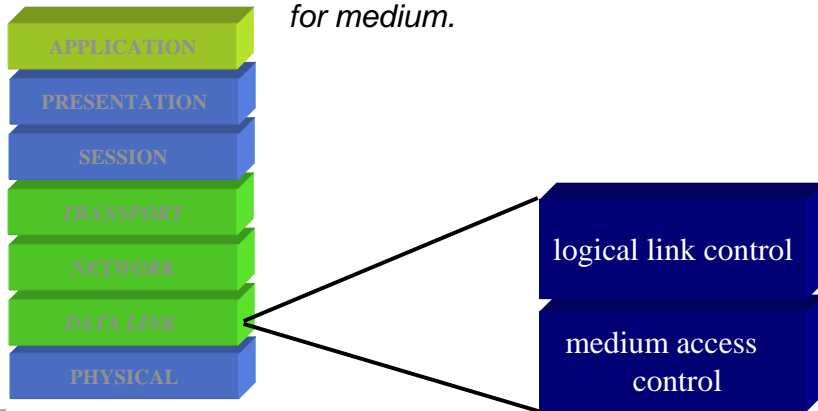
Data Link Layer

- So far, sending signals over transmission medium.
- Data link layer: responsible for error-free (reliable) communication between adjacent nodes.
- Functions: framing, error control, flow control, addressing, and medium access (in shared networks).

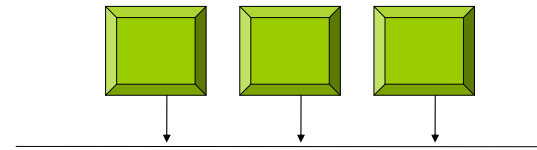


Medium Access Control

Coordinate competing requests
for medium.



Shared-medium networks



Multiple access / shared medium

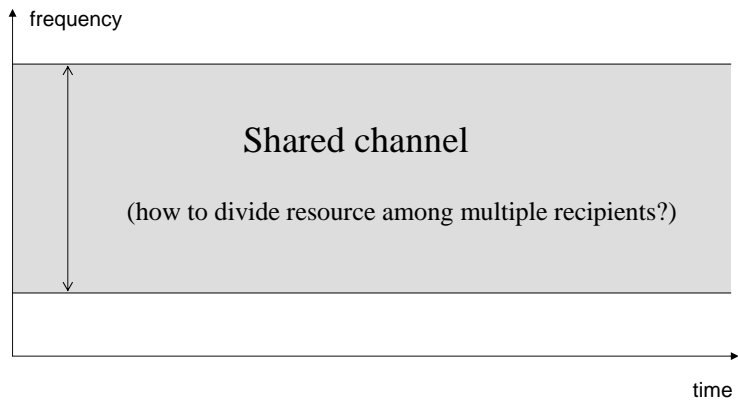
Media Access Control

- **Problem:**
 - Only one computer can transmit at a time.
 - If two computers try to use the same line at the same time, their messages get garbled.
 - Collision!
 - How can we organize transmissions so that all computers are given an opportunity to exchange messages?

Medium Access Control

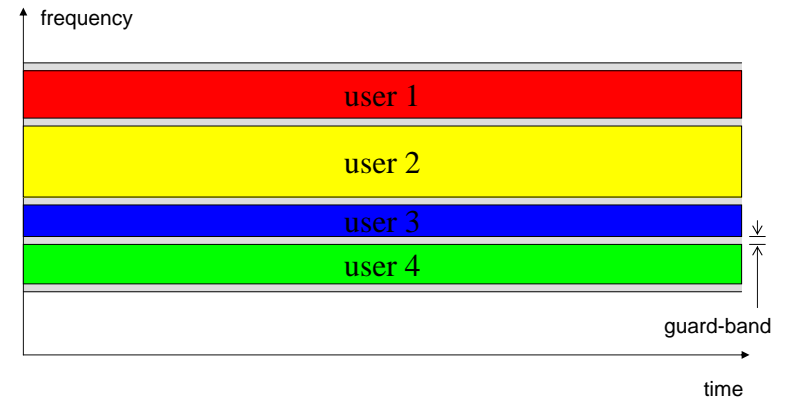
- Control access to shared medium.
- How?

The Multiplexing Problem



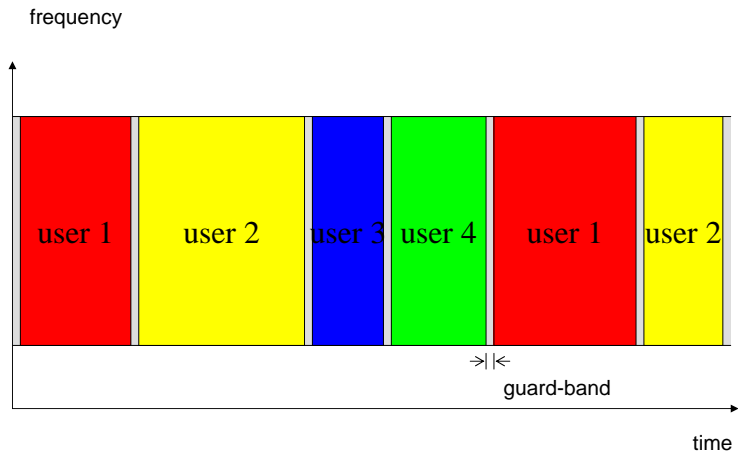
Analogy: a highway shared by many users

Frequency-Division Multiplexing



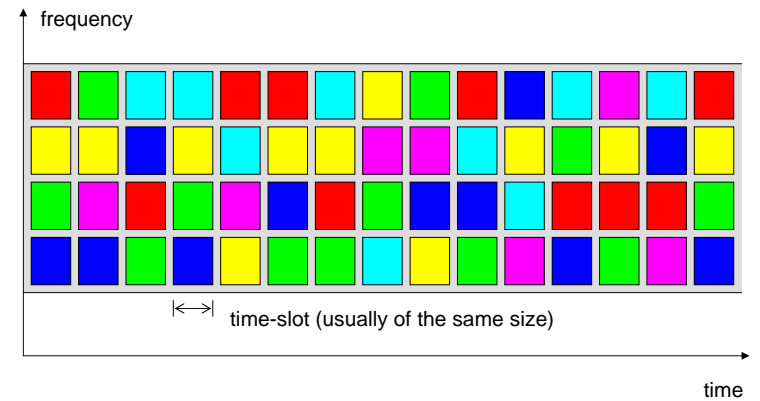
Analogy: a highway has multiple lanes

Time-Division Multiplexing



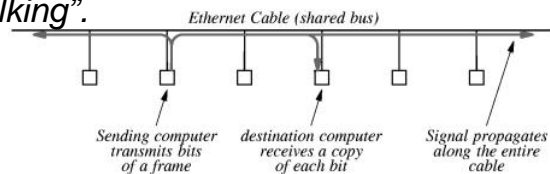
Requirement: precise time coordination

Frequency-Time-Division



Solution 1 - Ethernet

- Works on a **bus topology**.
- When a computer needs to send a message (**frame**), it first “listens” to the line.
 - Carrier sensing.
- When the line is “free” (no other computer is transmitting), it sends its message.
 - Nobody can “talk” while someone else is “talking”.



Ethernet (cont'd)

- What if two computers start “talking” at the same time?
 - There is a “collision” - the messages they send are useless!
 - The computers immediately realize that they are interfering with each other and abort transmission.
 - Collision detection.
- The computer must start transmitting the message all over again
 - Each computer waits for a random period of time, and then tries again: exponential backoff.

Ethernet (cont'd)

- What if a computer transmits a very long message?
 - It keeps the line busy for very long time, while all other computers must wait for the long message to end.
- **Rule #1 of resource sharing:** all “messages” must be “small”, to allow other computers to access the line.
 - For Ethernet, the maximum size of the payload is 1,500 bytes.

Ethernet (cont'd)

- What is the expected **performance** of Ethernet?
 - When only one computer needs to transmit: it can immediately access the line.
 - When many computers want to access (high traffic):
 - The average waiting time is high.
 - There is high probability of “collision”.
 - For every collision, the two computer must start sending the message all over again
- **Conclusion:** the expected **delay** depends on the traffic on network!

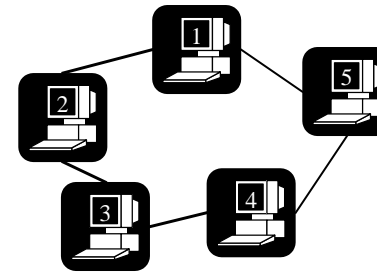
Solution 2 - Reservation

- Suppose there are M stations connected to the channel.
- In the beginning, each station transmits (in turn) a special “reservation packet” that says whether it has data ready to transmit.
- Then, all the stations that communicated that they have data ready, will in turn transmit their data packets (one packet per station).
- After that, all stations again transmit in turn their “reservation packet”, and so on.



Solution 3 - Token Passing

- Example: **Token Ring**.
- Works on a **ring topology**.



Token Passing (cont'd)

- Consider the following analogy:
 - A group of friends sitting in a circle
 - A ball is passed from friend to friend
 - When somebody receives the ball, it passes to the friend to his/her left
 - A person is allowed to talk only when s/he has the ball in his/her hands
 - This guarantees that only one person talks at a time!



Token Passing (cont'd)

- Let's make the game more difficult:
 - A person that receives the ball and has something to say, rather than saying it, s/he writes it on a letter
 - Including the name of the addressee
 - Before passing along the ball, s/he passes along the letter
 - Everyone who receives the paper passes it to the person to his/her left
 - If s/he is the recipient of the letter, s/he signs it after reading it
 - Once the letter arrives back to the sender, s/he throws it away
 - The ball is still circulating in the circle



Token Passing (cont'd)

- *Token Passing works similarly:*
 - *A special pattern (3-bytes word) of bits called **token** moves from one computer to the next*
 - *If a computer does not have a message to send, it just passes the token along*
 - *Otherwise, it “seizes the token” and transmits its message (including the address)*
 - *The message is passed from one computer to the next, until it arrives back to the sender, which “destroys” it (does not pass it along anymore)*
 - *The addressee may “write” something on the message so that the sender knows it has been received correctly*
 - *Once the computer is done transmitting the message, it “releases” (transmits) the token*



Token Ring (cont'd)

- *What is the expected **performance** of Token Passing?*
 - *It is a very fair resource sharing mechanism*
 - *Each computer is given in turn an opportunity to transmit, even when the traffic is high*
 - *However, even if only one computer has a message to send, it has to wait that it receives the token.*
- *Again, long messages should not be allowed, because otherwise one computer may “hold the token” for too long time.*



Ethernet versus Token Ring

- *Token ring:*
 - *Efficient at heavy traffic.*
 - *Guaranteed delay.*
 - *Fair.*
 - *But, ring/token maintenance overhead.*
 - *But, under light traffic?*
- *Ethernet is simple!*



Standardized MACs

Topologies

Techniques	Bus	Ring
Round robin	Token bus (802.4) Polling (802.11)	Token ring (802.5; FDDI)
Scheduled	DQDB (802.6)	
Contention	CSMA/CD (802.3) CSMA/CA(802.11)	

