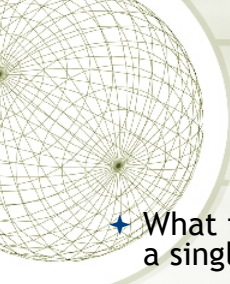


*Frames, Packets and
Topologies*

Info 341 Networking and
Distributed Applications




Frames and Packets




Motivating Packets

- ✦ What if a network required everything to be sent as a single chunk
 - ✦ What happens when you download a really large file, like a DVD?



Motivating Packets

- ✦ What if a network required everything to be sent as a single chunk
 - ✦ What happens when you download a really large file, like a DVD?
 - ✦ Does every one else just wait?
 - ✦ What happens if someone just sends?
 - ✦ What if there is an error in the middle of the transmission?



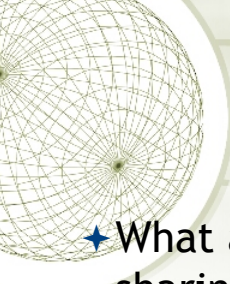
Outline

- ★ Motivating Packets
 - ◆ Sharing
- ★ Packets
 - ◆ Concept of the packet
 - ◆ Handling special content
 - ◆ Detecting transmission error



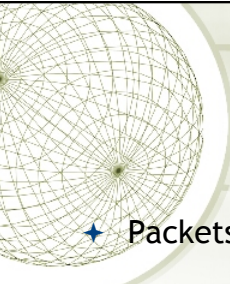
Sharing the Network

- ★ What are some ways we know of sharing a network?



Sharing the Network

- ★ What are some ways we know of sharing a network?
 - ◆ Multiplexing
 - ◆ Frequency Division Multiplexing
 - ◆ Time Division Multiplexing



What is a packet?

- ★ Packets are chunks of data
 - ★ Packets are binary data on a network
 - ◆ Contents are often written in hexadecimal
 - ★ Packets facilitate sharing the network
 - ◆ A form of time division multiplexing
 - ★ We'll use the term 'packet' for as the most general way of describing data on a network.



Packets = payload + overhead

- ★ Payload (the data)
- ★ Overhead
 - ★ Destination address
 - ★ Source address
 - ★ Packet format/type
 - ★ Error detection
- ★ Each device that communicates on a network needs a locally unique address



Payloads

- ★ Special content
 - ★ Some data values are special
 - ★ Special meaning, reserved
 - ★ Special non-printing ASCII characters
 - ★ soh, eot, esc
- ★ How do you send special data?

Byte Stuffing

- ★ Mechanism for transmitting special data

- ★ Example

- ★ Characters 'soh' and 'eot' have special meaning
 - ★ What if you want to send the 'soh' character as part of payload?

Byte Stuffing Example

- ★ Suppose your message contained a special character

c	o	n	t	a	i	n	s		s	o	h		h	e	r	e		□
---	---	---	---	---	---	---	---	--	---	---	---	--	---	---	---	---	--	---


- ★ In hexadecimal

63	6F	6E	74	61	69	6E	73	20	73	6F	68	20	68	65	72	65	20	01
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

- ★ Byte stuffing ensures that the special character *soh* (01 in hex) is not in the data

63	6F	6E	74	61	69	6E	73	20	73	6F	68	20	68	65	72	65	20	1B	78
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

- ★ Sometimes known as an 'escape sequence'



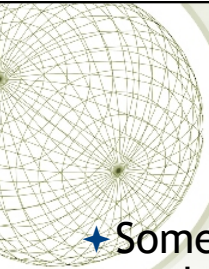
Byte Stuffing Example

- ★ Special character replaced with a sequence

Data	Sent
soh	esc x
eot	esc y
esc	esc z

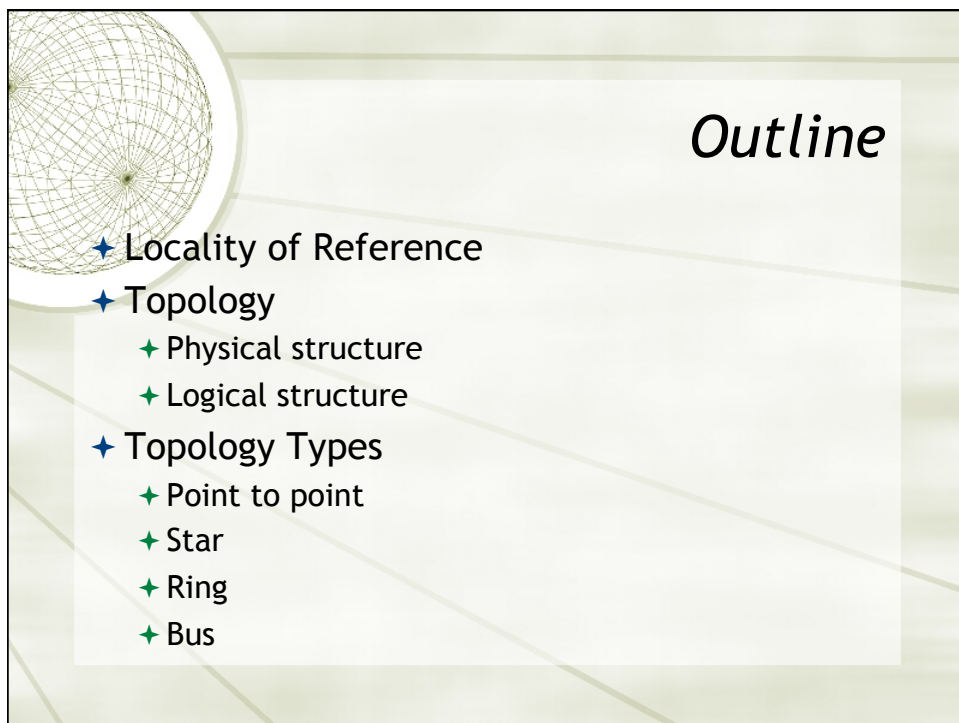
- ★ After byte stuffing data is safe in any packet

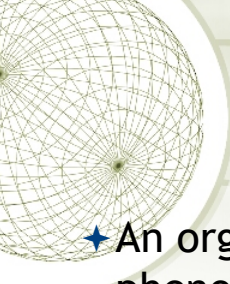
01	63	6F	6E	74	61	69	6E	73	20	73	6F	68	20	68	65	72	65	20	1B	78	04
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



Envelope Packets within Packets

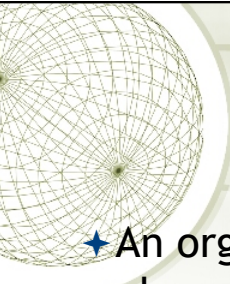
- ★ Sometimes the data in a packet - is a packet
 - ✦ Enveloping
- ★ Network hardware packet
 - ✦ The hardware defines what the 'native' packet format - called a **frame**
 - ✦ Outer most packet is always the hardware packet type





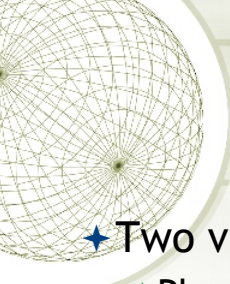
Locality of Reference

- ★ An organizing principle for many phenomenon
 - ✦ Consider 'client' files in a physical file



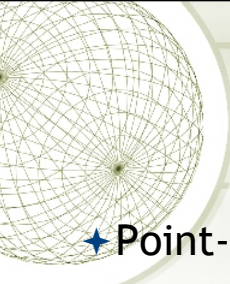
Locality of Reference

- ★ An organizing principle for many phenomenon
 - ✦ Consider 'client' files in a physical file
- ★ Two key types
 - ✦ Physical locality of reference
 - ✦ Temporal locality of reference



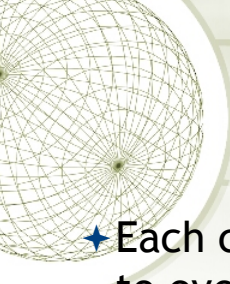
Topology

- ★ Two views of network topology
 - ◆ Physical topology
 - ◆ The actual layout of the network
 - ◆ How it is physically built
 - ◆ Logical topology
 - ◆ How data is organized and transported across the network.



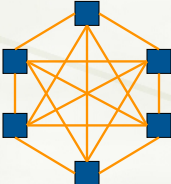
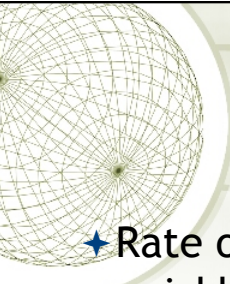
Topologies

- ★ Point-to-Point
- ★ Star
- ★ Ring
- ★ Bus



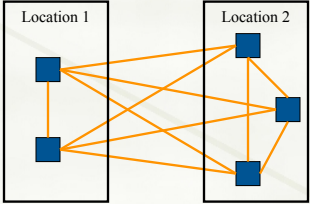
Point-to-Point

- ★ Each computer has a direct connection to every other computer in the network
- ★ Very robust
- ★ Highly redundant

How Many Connections?

- ★ Rate of interconnection increases quickly

$$\text{Connections} = (N^2 - N) / 2$$


Point-to-Point

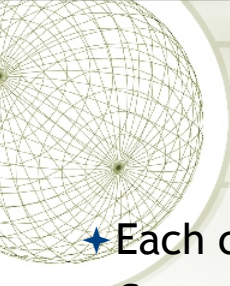
- ★ For long distance networks, point-to-point is not efficient
- ★ Maybe better to have one connection?

The diagram illustrates two network topologies between two locations, Location 1 and Location 2. In the left diagram, each location contains two nodes (blue squares). All nodes in Location 1 are connected to all nodes in Location 2, creating a dense mesh of connections. In the right diagram, each location also contains two nodes, but only one node in each location is connected to a single node in the other location, representing a single connection between the two locations.

Star

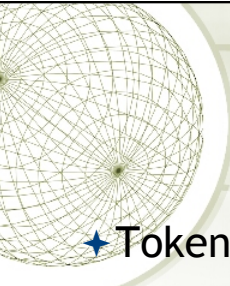
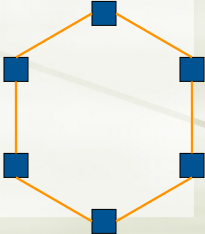
- ★ Each computer has one connection to a central connector device
 - ★ A central point (hub/switch) receives the data and communicates it to the destination
 - ★ Small number of connections
 - ★ If central point fails, everything stops

The diagram shows a star network topology. A central blue circle represents a hub or switch. Six blue squares, representing computers, are arranged around the central circle, each connected to it by a single line. This illustrates a central point where all data flows.




Ring

- ★ Each computer has two connections
- ★ Connected in series, if one computer fails ring breaks
- ★ Adding computers slows network
- ★ Shared medium




Ring

- ★ Token Ring
- ★ FDDI (Fiber Distributed Data Interface)
 - ◆ Double-counter directional rings

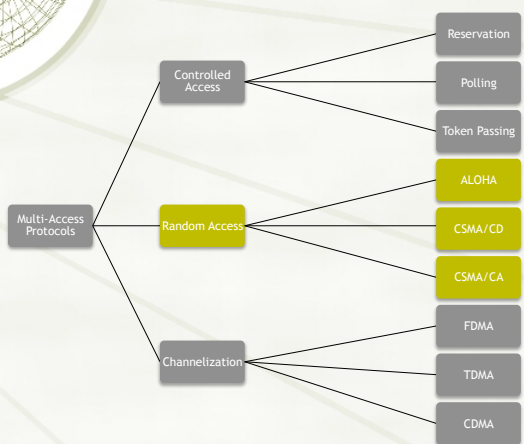


Bus

- ★ Each computer needs one connection
- ★ All computers receive the data
- ★ Shared medium

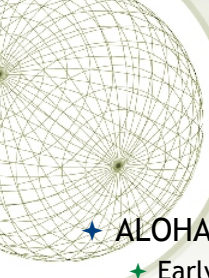


Protocols for Shared Access



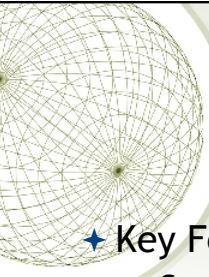
```

graph LR
    MAP[Multi-Access Protocols] --> CA[Controlled Access]
    MAP --> RA[Random Access]
    MAP --> CH[Channelization]
    CA --> R[Reservation]
    CA --> P[Polling]
    CA --> TP[Token Passing]
    RA --> A[ALOHA]
    RA --> CSMA_CD[CSMA/CD]
    RA --> CSMA_CA[CSMA/CA]
    CH --> FDMA[FDMA]
    CH --> TDMA[TDMA]
    CH --> CDMA[CDMA]
    
```



Random Access Protocols

- ✦ ALOHA
 - ✦ Early radio network protocol
- ✦ CSMA/CD - Carrier Sense Multi-Access/ Collision Detection
 - ✦ Ethernet
- ✦ CSMA/CA - Carrier Sense Multi-Access/Collision Avoidance
 - ✦ WiFi




CSMA/CD


- ✦ Key Features of CSMA/CD
 - ✦ Carrier Sense
 - ✦ Detect if a packet is being sent - if you only see the carrier, then no packet is currently being sent
 - ✦ Collision Detection
 - ✦ Listen when sending, if signal is different from what is sent, then collision
 - ✦ Exponential Backoff
 - ✦ When collision happens calculate a random time to wait before sending

CSMA/CA


- ★ Key Features
 - ★ Collision Avoidance handles “hidden” node problem
 - ★ Computer 2 sees 1 and 3, but 1 is hidden from 3 because of signal attenuation



Computer 1




Computer 2




Computer 3

CSMA/CA


- ★ Key Features
 - ★ Computer 1 sends a request to send
 - ★ Computer 2 sends a clear to send signal
 - ★ Notifies Computer 1 *and* Computer 3
 - ★ Computer 1 sends data
 - ★ Does not prevent a collision



Computer 1



Computer 2



Computer 3

