Review of Networking and Design Concepts (I)

http://www.pde.rpi.edu/

Or

http://www.ecse.rpi.edu/Homepages/shivkuma/

Shivkumar Kalyanaraman
Rensselaer Polytechnic Institute
shivkuma@ecse.rpi.edu

Based in part upon slides of Prof. Raj Jain (OSU), S. Keshav (Cornell), L. Peterson (Princeton), J. Kurose (U Mass)

Overview

- Connectivity:
  - direct (pt-pt, N-users),
  - indirect (switched, inter-networked)
- Concepts: Topologies, Framing, Multiplexing, Flow/Error Control, Reliability, Multiple-access, Circuit/Packet-switching, Addressing/routing, Congestion control
- Data link/MAC layer:
  - SLIP, PPP, LAN technologies …
- Interconnection Devices
- Chapter 1,2,11 in Doug Comer book
- **Reading**: Saltzer, Reed, Clark: "End-to-End arguments in System Design"
- **Reading:** Clark: "The Design Philosophy of the DARPA Internet Protocols":
- **Reading:** RFC 2775: Internet Transparency: In HTML

Connectivity...

- Building Blocks
  - links: coax cable, optical fiber...
  - nodes: general-purpose workstations...
- *Direct* connectivity:
  - point-to-point
  - multiple access

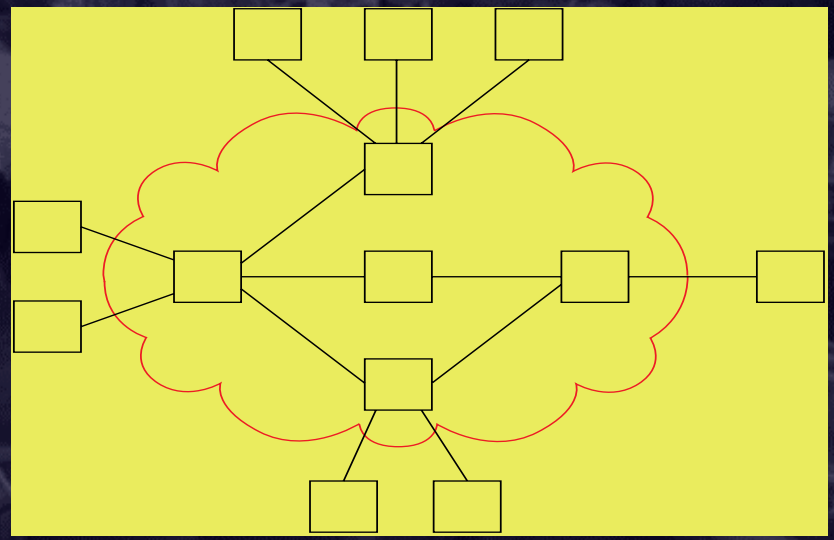
Connectivity… (Continued)

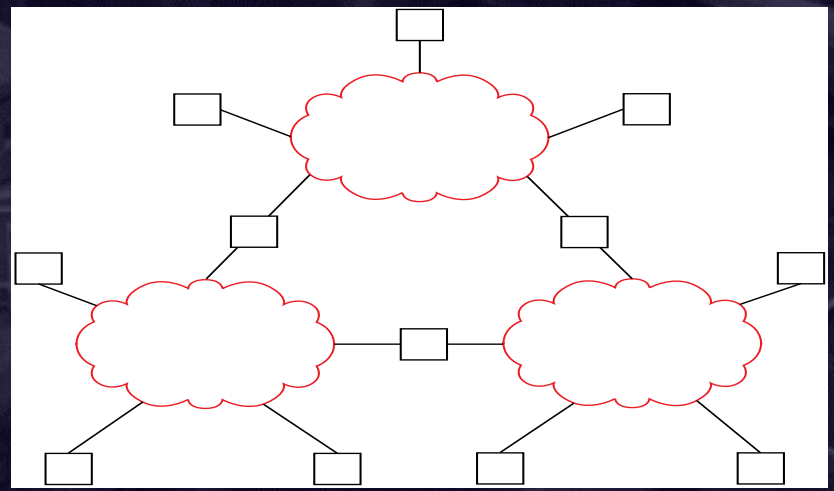
- *Indirect* Connectivity
  - switched networks

    => ***switches***

  - inter-networks

    => ***routers***



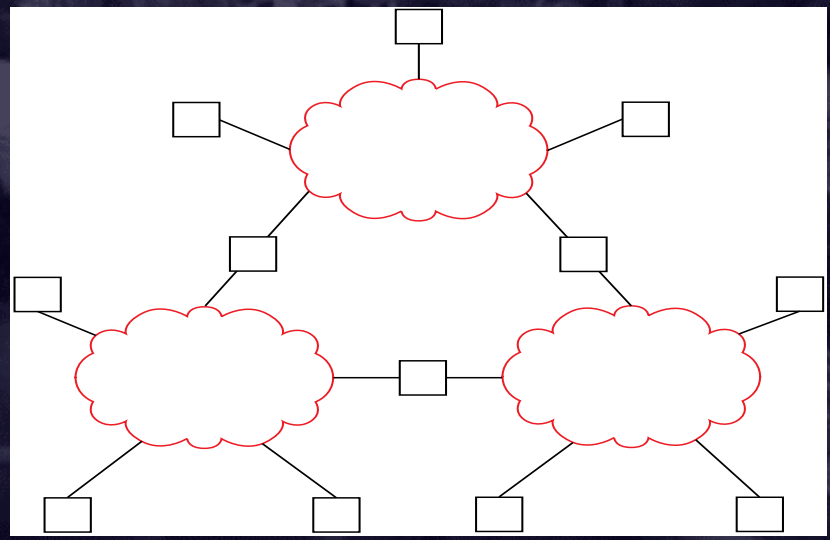


What is "*Connectivity*" ?

- *Direct or indirect access to every other node in the network*
- *Connectivity* is the *magic* needed to communicate if you do not have a link.
  - *Tradeoff: Performance characteristics worse!*

Connectivity …

- Internet:
  - *Best-effort (no performance guarantees)*
  - *Packet-by-packet*



- A pt-pt link:
  - *Always-connected*
  - *Fixed bandwidth*
  - *Fixed delay*
  - *Zero-jitter*

Point-to-Point Connectivity Issues

- **Physical** layer: coding, modulation etc
- **Link layer** needed if the link is shared bet'n apps; is unreliable; and is used sporadically
- *No need for protocol concepts* like addressing, names, routers, hubs, forwarding, filtering …

Link Layer: Serial IP (SLIP)

- Simple: only framing = Flags + byte-stuffing
- Compressed headers (CSLIP) for efficiency on low speed links for interactive traffic.
- Problems:
  - Need other end's IP address a priori (can't dynamically assign IP addresses)
  - No "type" field => no multi-protocol encapsulation
  - No checksum => all errors detected/corrected by higher layer.
- RFCs: 1055, 1144

Link Layer: PPP

- *Point-to-point protocol*
- Frame format similar to HDLC
- Multi-protocol encapsulation, CRC, dynamic address allocation possible
  - key fields: flags, protocol, CRC
- Asynchronous and synchronous communications possible

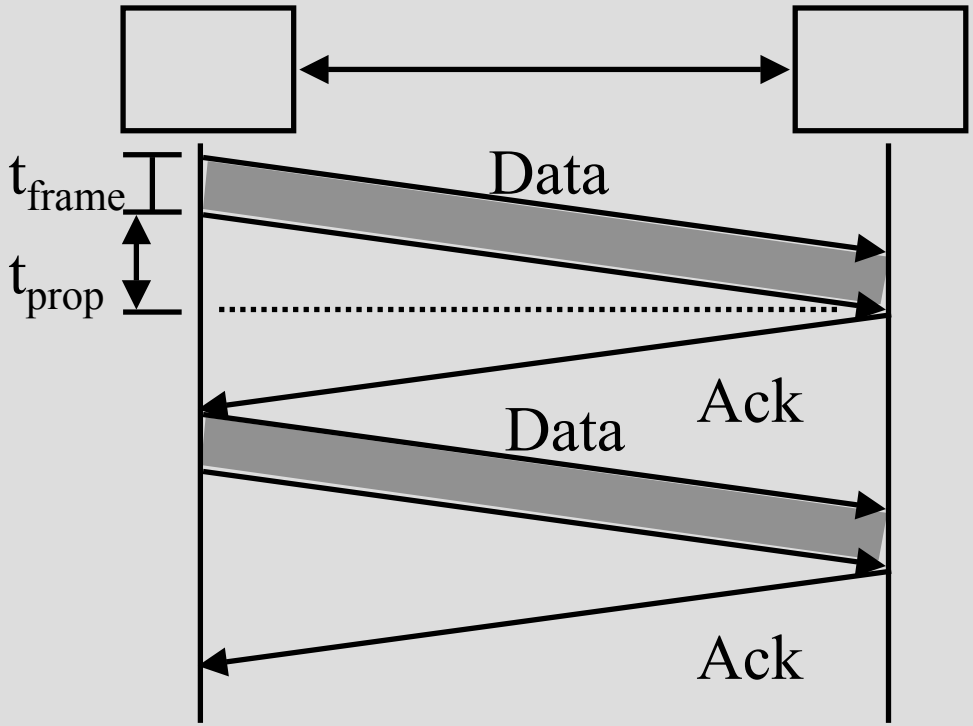
Link Layer: PPP (Continued)

- Link and Network Control Protocols (LCP, NCP) for flexible control & peer-peer negotiation
- Can be mapped onto low speed (9.6Kbps) and high speed channels (SONET)
- RFCs: 1548, 1332

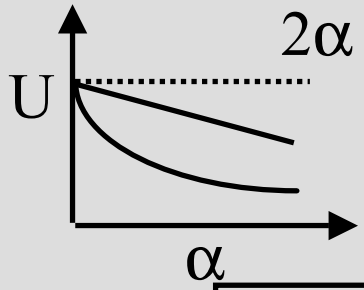
Reliability Mechanisms

- Mechanisms:
  - Checksum: detects corruption *in pkts & acks*
  - ACK: "packet correctly received"
  - Duplicate ACK: "packet *in*correctly received"
  - Sequence number: identifies packet or ack
    - 1-bit sequence number used *both in forward & reverse* channel
  - Timeout only at sender
- Reliability capabilities achieved:
  - An *error-free* channel
  - A *forward & reverse* channel with *bit*-errors
  - Detects *duplicates* of packets/acks
  - *NAKs eliminated*
  - A *forward & reverse* channel with *packet*-errors (loss)

Stop and Wait Flow Control



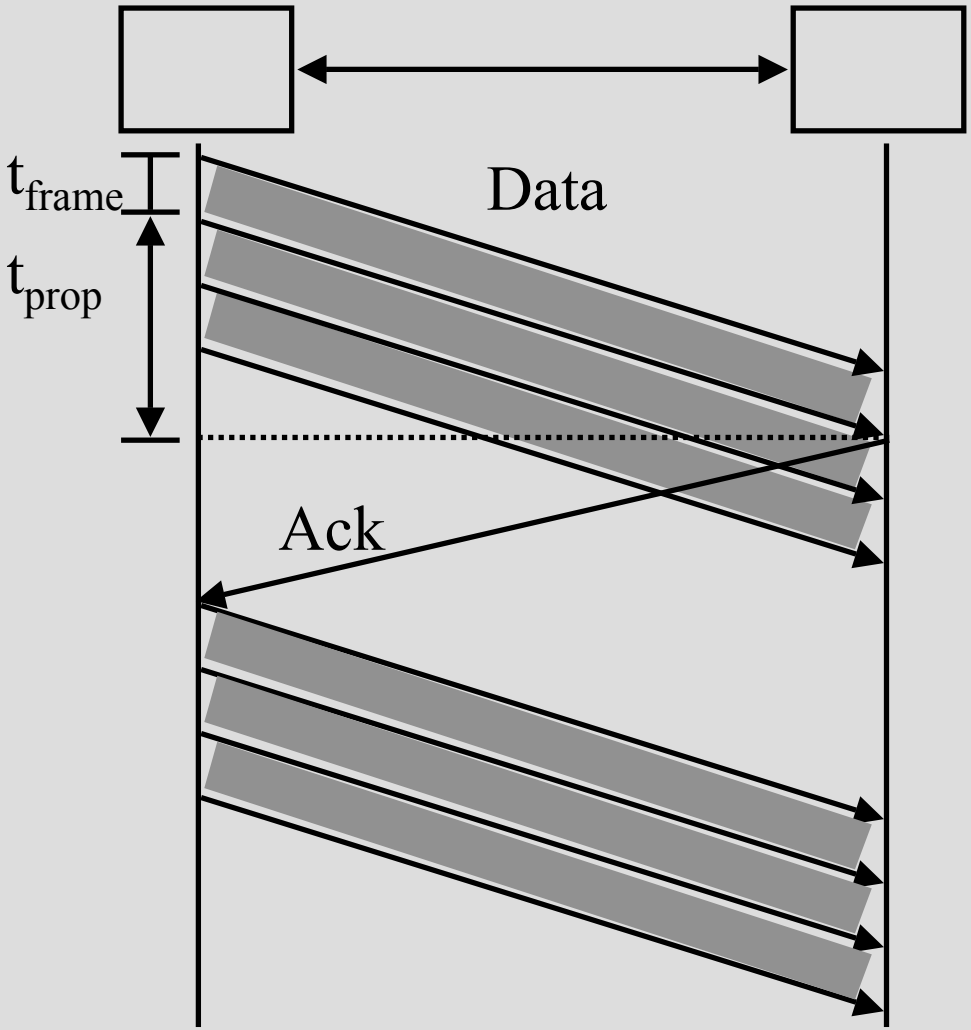
$$U = \frac{t_{frame}}{2t_{prop} + t_{frame}} = \frac{1}{2\alpha + 1}$$



$$\alpha = \frac{t_{prop}}{t_{frame}} = \frac{\text{Distance/Speed of Signal}}{\text{Frame size /Bit rate}} = \frac{\text{Distance} \times \text{Bit rate}}{\text{Frame size} \times \text{Speed of Signal}}$$

Light in vacuum
= 300 m/μs
Light in fiber
= 200 m/μs
Electricity
= 250 m/μs

Sliding Window Protocols



$$U = \frac{Nt_{frame}}{2t_{prop}+t_{frame}}$$

$$= \begin{cases} \dfrac{N}{2\alpha+1} \\ 1 \text{ if } N>2\alpha+1 \end{cases}$$

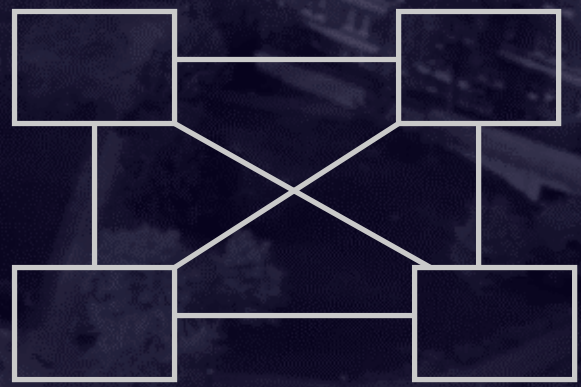
List of Issues

- Connectivity (direct/indirect)
- Pt-Pt connectivity:
  - Framing
  - Error control/Reliability (optional)
  - Flow control (optional)

Connecting N users: Directly…

- Pt-pt: connects only two users directly…
- How to connect **N** users **directly** ?

Bus

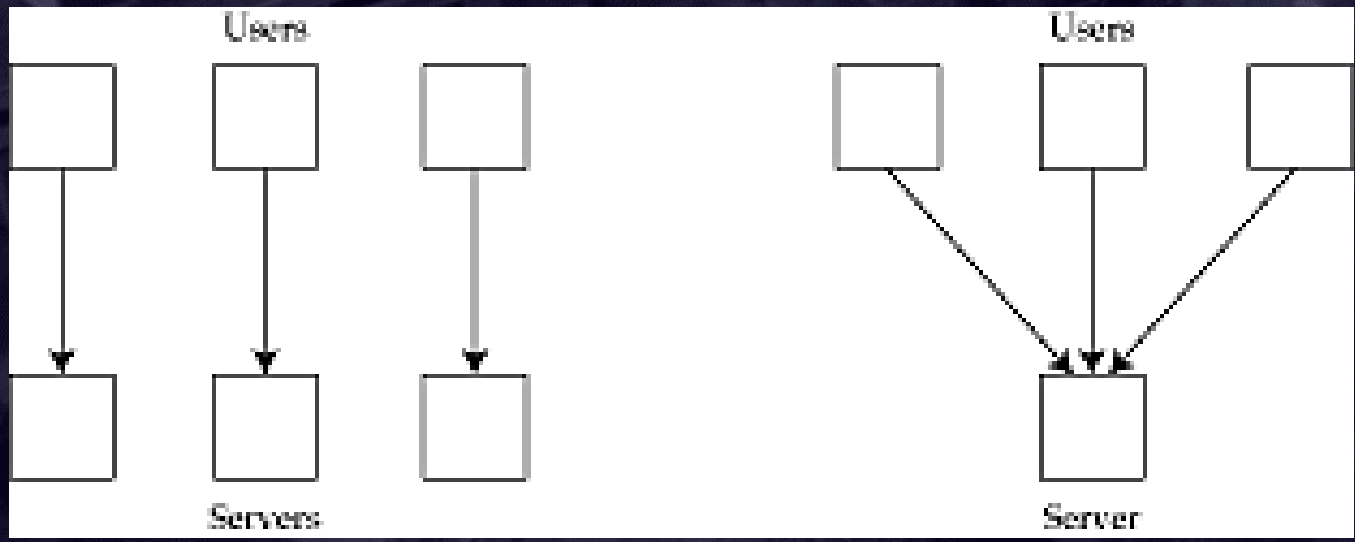


Full mesh

- What are the costs of each option?
- Does this method of connectivity scale ?

Multiplexing vs Have it all

- Multiplexing = sharing
  - Allows system to achieve "economies of scale"
  - ***Cost***: waiting time (delay), buffer space & loss
  - ***Gain***: Money ($$) => Overall system costs less

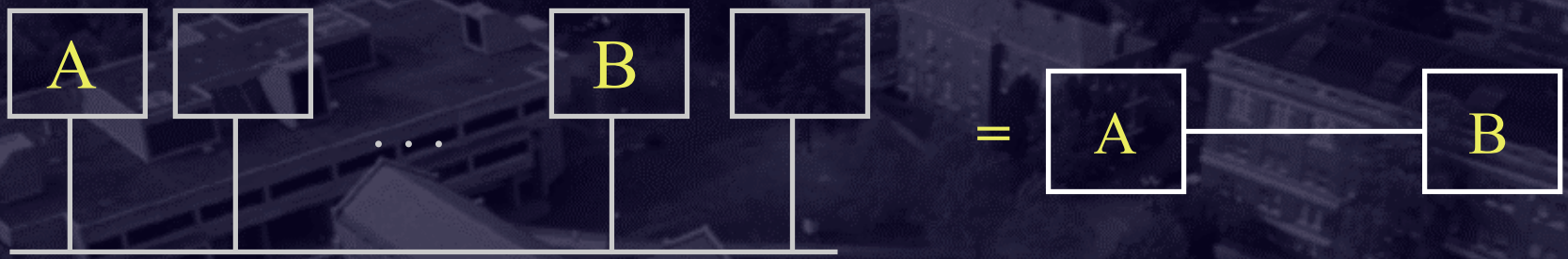


Full Mesh

Bus

Virtualization

- The multiplexed shared resource with a level of indirection will seem like a *unshared virtual resource*!
  - I.e. Multiplexing + indirection = virtualization



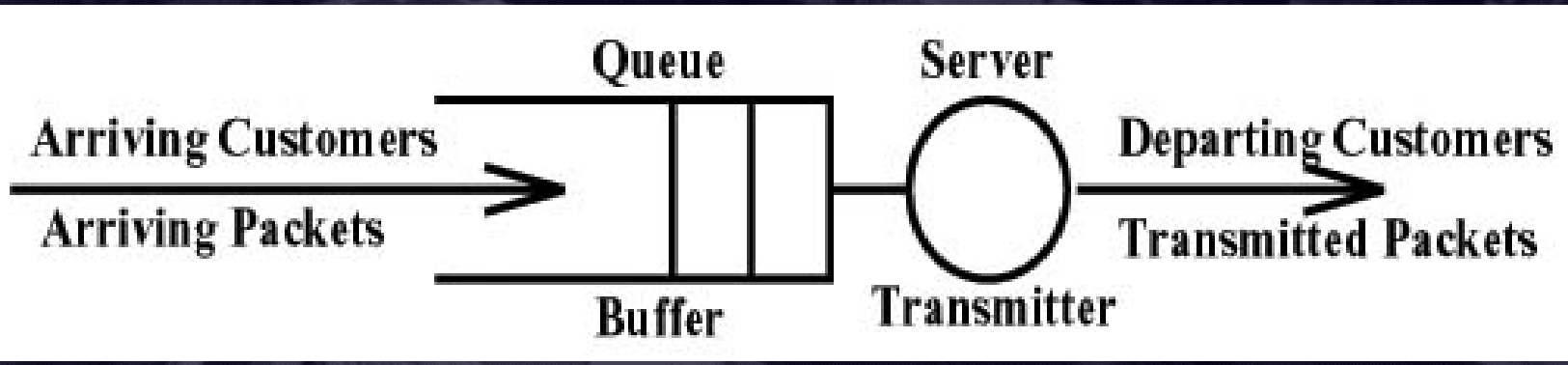
  - We can "refer" to the virtual resource as if it were the physical resource.
    - Eg: virtual memory, virtual circuits…
  - Connectivity: a virtualization created by the Internet!
  - Indirection requires *binding and unbinding*…
    - Eg: use of packets, slots, tokens etc.

Statistical Multiplexing

- Reduce resource requirements (eg: bus capacity) by *exploiting statistical knowledge* of the system.
  - Eg: average rate <= service rate <= peak rate
  - If service rate < average rate, then system becomes unstable!!
    - First design to ensure system stability!!
  - Then, for a stable multiplexed system:
    - *Gain* = peak rate/service rate.
    - *Cost*: buffering, queuing delays, losses.
- *Useful only if peak rate differs significantly from average rate.*
  - Eg: if traffic is smooth, fixed rate, no need to play games with capacity sizing…

Stability of a Multiplexed System

Average Input Rate > Average Output Rate
=> system is unstable!



How to ensure stability ?

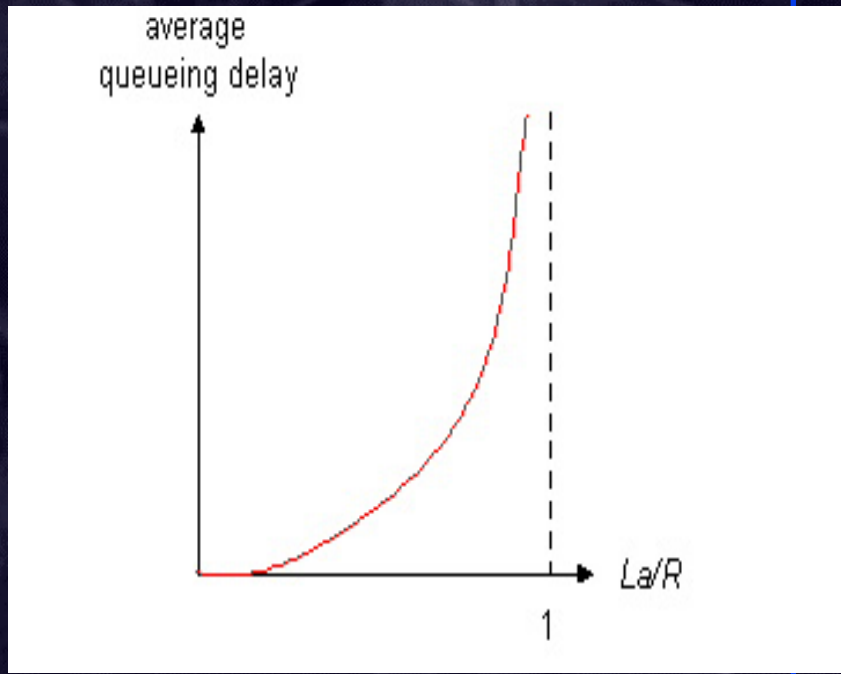
1. Reserve enough capacity so that demand is less than reserved capacity
2. Dynamically detect overload and adapt either the demand or capacity to resolve overload

What's a performance *tradeoff* ?

- **A situation where you cannot get something for nothing!**
- **Also known as a zero-sum game.**

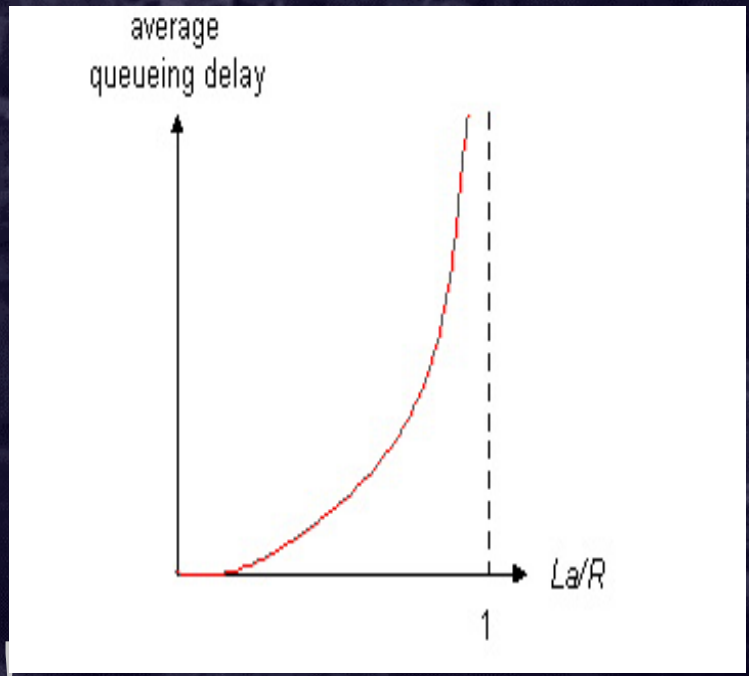
- R=link bandwidth (bps)
- L=packet length (bits)
- a=average packet arrival rate

Traffic intensity = La/R



What's a performance *tradeoff* ?

- La/R ~ 0: average queuing delay small
- La/R -> 1: delays become large
- La/R > 1: average delay infinite (*service degrades unboundedly => instability*)!

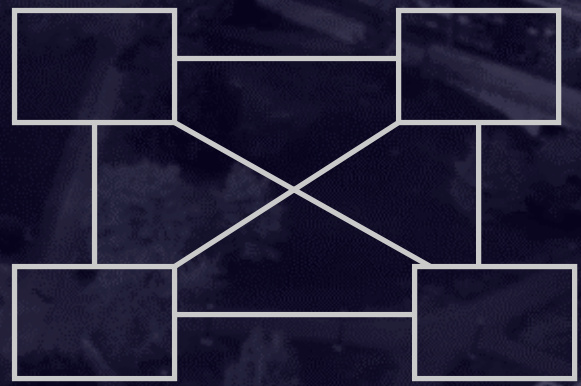


Summary: Multiplexing using bus topologies has both direct resource costs and intangible costs like potential instability, buffer/queuing delay.

Connecting N users: *Directly* ...

- ***Bus***: Low cost vs broadcast/collisions, MAC complexity
- ***Full mesh***: High cost vs simplicity
- *New concept:*
  - Address to identify nodes.
  - Needed if we want the receiver alone to consume the packet!

Bus



Full mesh

- Problem: Direct connectivity does not "scale"….

How to build Scalable Networks?

- Scaling: system allows the increase of a key parameter. Eg: let N increase…
  - *Inefficiency limits scaling* …
- Direct connectivity is *inefficient* & hence does not scale
  - Mesh: *inefficient* in terms of # of links
  - Bus architecture: 1 expensive link, N cheap links. *Inefficient* in bandwidth use

Filtering, forwarding …

- Filtering: choose a subset of elements from a set
  - Don't let information go where its not supposed to…
  - *Filtering => More efficient => more scalable*

Filtering is the key to efficiency & scaling

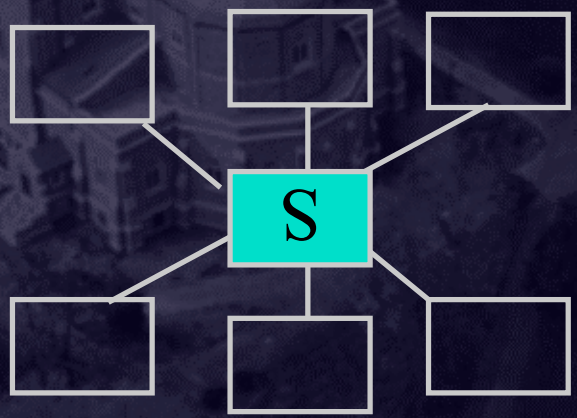
- Forwarding: actually sending packets to a filtered subset of link/node(s)
  - Packet sent to one link/node => efficient

- Solution: Build nodes which focus on filtering/forwarding and achieve indirect connectivity

"switches" & "routers"

Connecting N users: *Indirectly*

- Star: One-hop path to any node, reliability, forwarding function
- "Switch" S can filter and forward!
  - Switch may forward multiple pkts in parallel for additional efficiency!



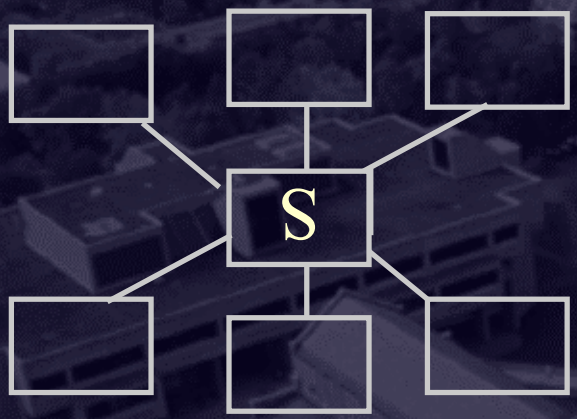
Star

Connecting N users: *Indirectly* …

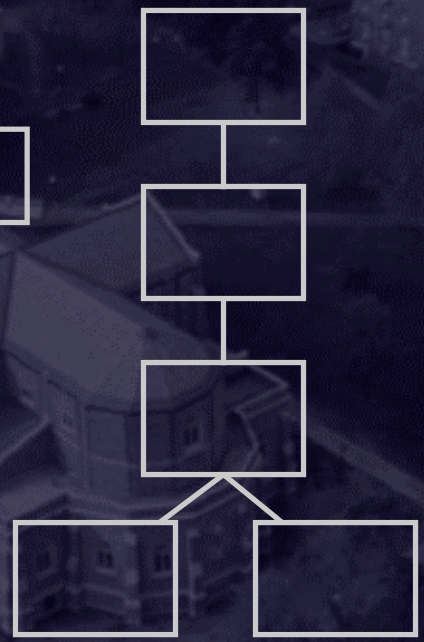
- **Ring:** *Reliability* to link failure, *near-minimal* links
- All nodes need "forwarding" and "filtering"
- Sophistication of forward/filter lesser than switch

Ring

Topologies: Indirect Connectivity



Star



Tree

Ring

Multi-Access LANs

- ***Hybrid topologies:***
  - Uses directly connected topologies (eg: bus), or
  - Indirectly connected with simple filtering components (switches, hubs).
    - Limited scalability due to limited filtering

- ***Medium Access Protocols***:
  - ALOHA, CSMA/CD (Ethernet), Token Ring …
  - *Key: Use a single protocol in network*

- ***Concepts***: address, forwarding (and forwarding table), bridge, switch, hub, token, medium access control (MAC) protocols

MAC Protocols: a taxonomy

Three broad classes:

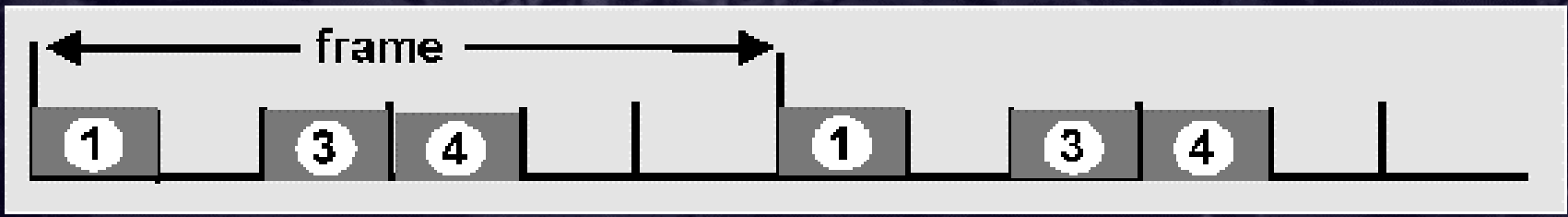
- **Channel Partitioning**
  - divide channel into smaller "pieces" (time slots, frequency)
  - allocate piece to node for exclusive use
- **Random Access**
  - allow collisions
  - "recover" from collisions
- **"Taking turns": Token-based**
  - tightly coordinate shared access to avoid collisions

Goal: efficient, fair, simple, decentralized

Channel Partitioning MAC protocols. Eg: TDMA

TDMA: time division multiple access

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

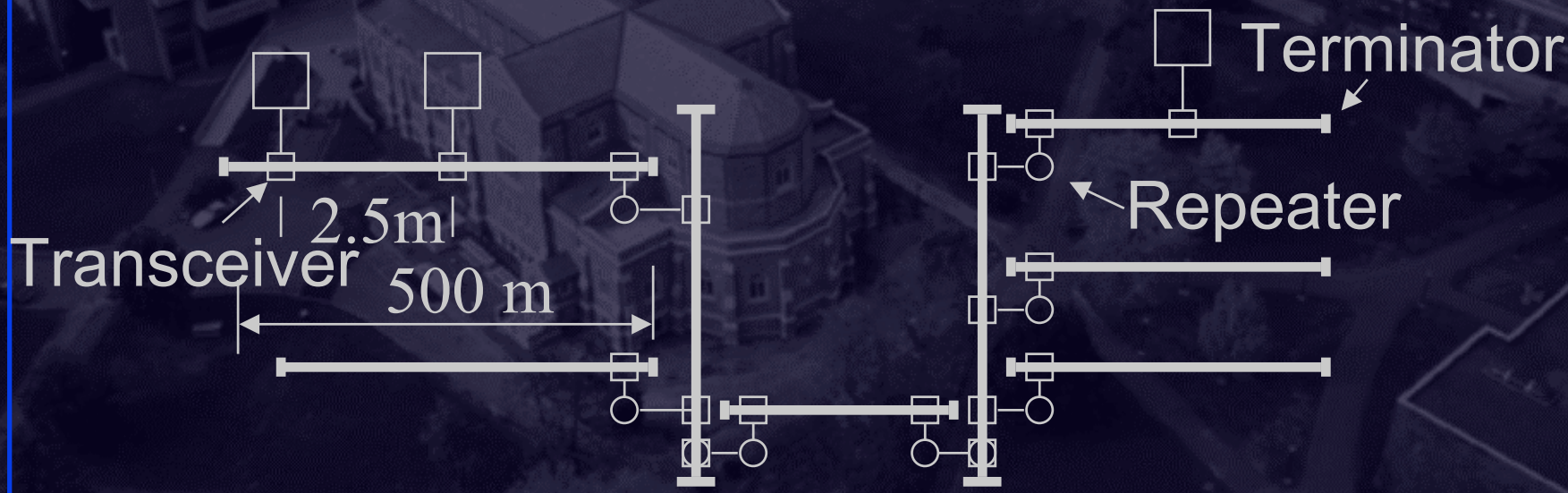


Review: *Multiple Access Protocols*

- Aloha at University of Hawaii:
  Transmit whenever you like
  Worst case utilization = 1/(2e) =18%
- CSMA: Carrier Sense Multiple Access
  Listen before you transmit
- CSMA/CD: CSMA with Collision Detection
  Listen while transmitting.
  Stop if you hear someone else.
- Ethernet uses CSMA/CD.
  Standardized by IEEE 802.3 committee.

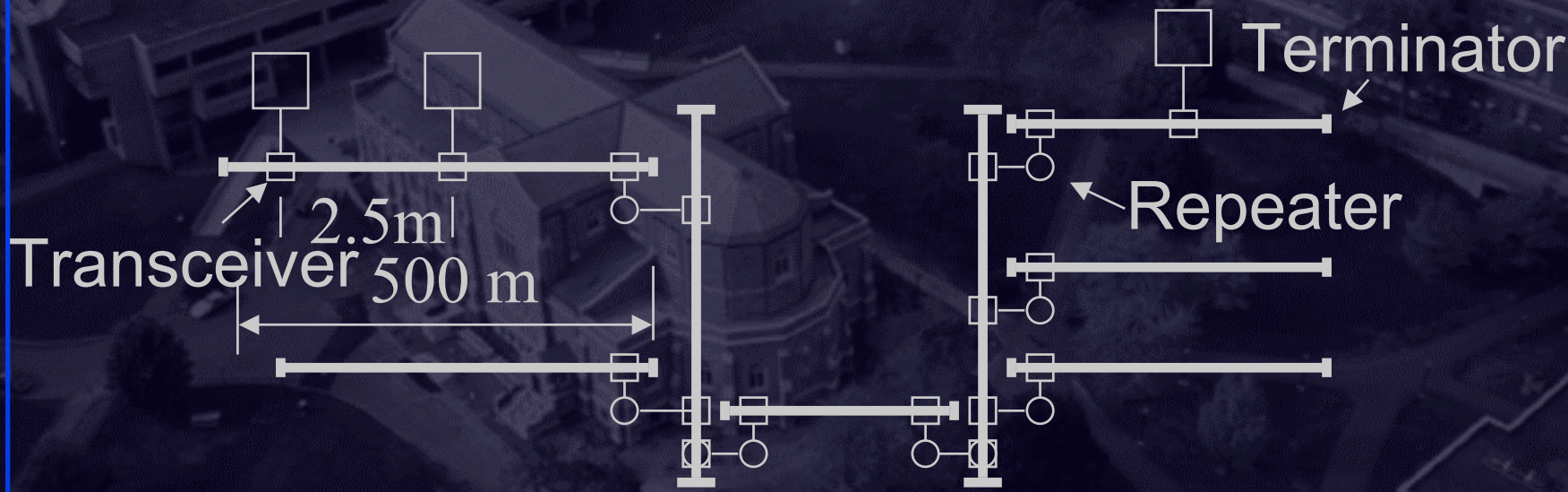
10Base5 Ethernet *Cabling* Rules

- Thick coax
- Length of the cable is limited to 2.5 km, no more than 4 repeaters between stations
- No more than 500 m per segment ⇒ "10Base5"



10Base5 *Cabling* Rules (Continued)

- No more than 2.5 m between stations
- Transceiver cable limited to 50 m



Inter-connection Devices

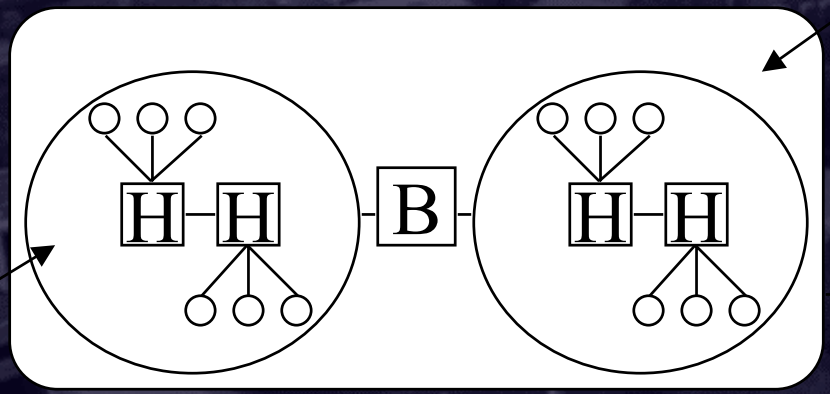
- **Repeater:** Layer 1 (PHY) device that restores data and collision signals: a digital amplifier
- **Hub:** Multi-port repeater + fault detection
  - Note: broadcast at layer 1
- **Bridge:** Layer 2 (Data link) device connecting two or more *collision domains*.
  - Key: a bridge attempts to filter packets and forward them from one collision domain to the other.
  - It snoops on passing packets and learns the interface where different hosts are situated, and builds a L2 forwarding table
  - MAC multicasts propagated throughout "*extended LAN*."
  - Note: Limited filtering intelligence and forwarding capabilities at layer 2

Interconnection Devices (Continued)

- **Router:** Network layer device. IP, IPX, AppleTalk. Interconnects *broadcast domains*.
  - Does not propagate MAC multicasts.
- **Switch:**
  - Key: has a switch fabric that allows parallel forwarding paths
  - *Layer 2 switch:* Multi-port bridge w/ fabric
  - *Layer 3 switch:* Router w/ fabric and per-port ASICs

These are functions. Packaging varies.

Interconnection *Devices*



LAN= **Collision Domain**

Extended LAN =**Broadcast domain**

Router

Application	Gateway	Application
Transport		Transport
Network	Router	Network
Datalink	Bridge/Switch	Datalink
Physical	Repeater/Hub	Physical

Ethernet (IEEE 802) *Address Format*

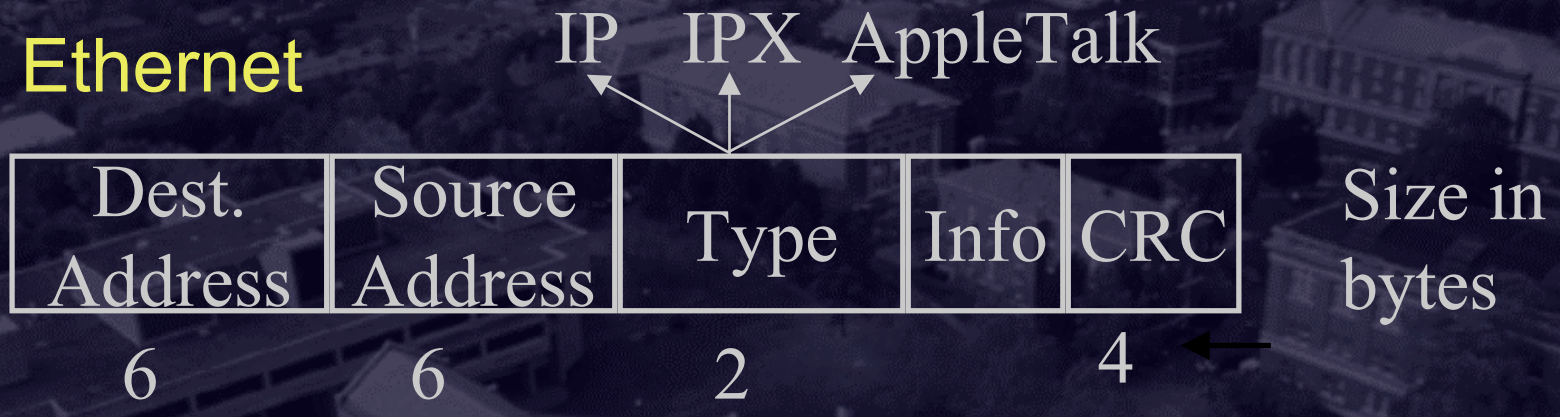
- 48-bit flat address => no hierarchy to help forwarding
  - Hierarchy only for administrative/allocation purposes
  - Assumes that all destinations are (logically) directly connected.
  - *Address structure does not explicitly acknowledge indirect connectivity*
    - => Sophisticated filtering cannot be done!

Ethernet (IEEE 802) *Address Format*

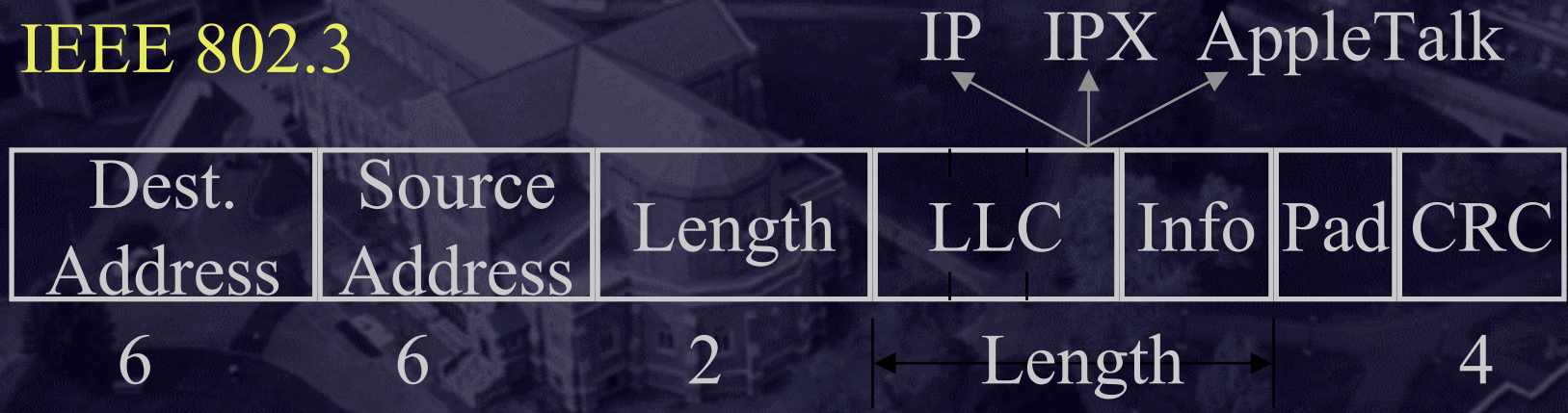
- G/L bit: *administrative*
  - Global: unique worldwide; assigned by IEEE
  - Local: Software assigned
- G/I: bit: *multicast*
  - I: unicast address
  - G: multicast address. Eg: “To all bridges on this LAN”

Ethernet & 802.3 *Frame Format*

- Ethernet



- IEEE 802.3



- Maximum Transmission Unit (MTU) = 1518 bytes
- Minimum = 64 bytes (due to CSMA/CD issues)

"Taking Turns" MAC protocols - 1

Channel partitioning MAC protocols:

- share channel efficiently at high load
- inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

Random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

"Taking turns" protocols

look for best of both worlds!

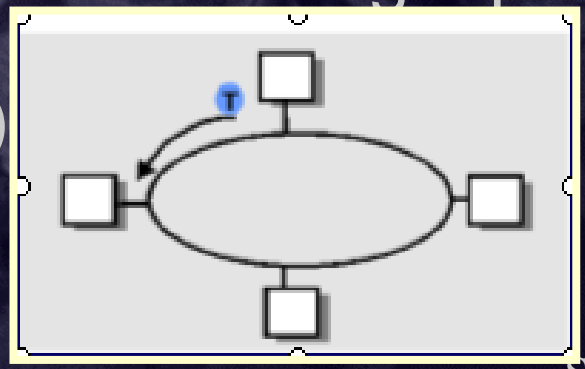
"Taking Turns" MAC protocols - 2

Polling:

- Master node "invites" slave nodes to transmit in turn
- Request to Send, Clear to Send messages
- Concerns:
  - polling overhead
  - latency
  - single point of failure (master)

Token passing:

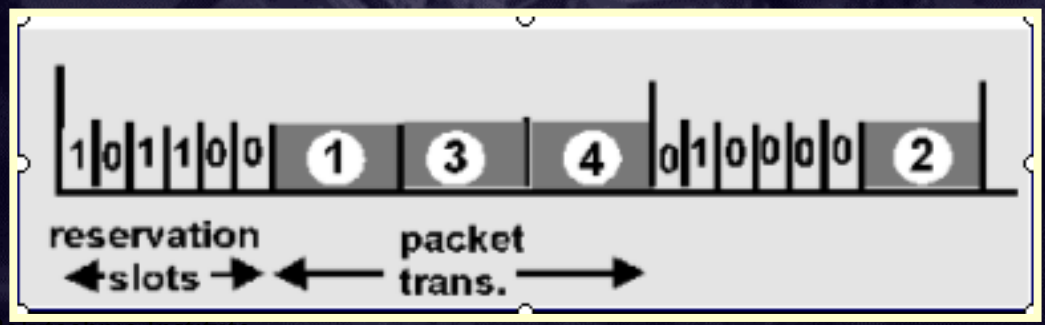
- Control **token** passed from one node to next sequentially.
- Token message
- Concerns:
  - token overhead
  - latency
  - single point of failure (token)



"Taking Turns" Protocols –3

Reservation-based a.k.a Distributed Polling:

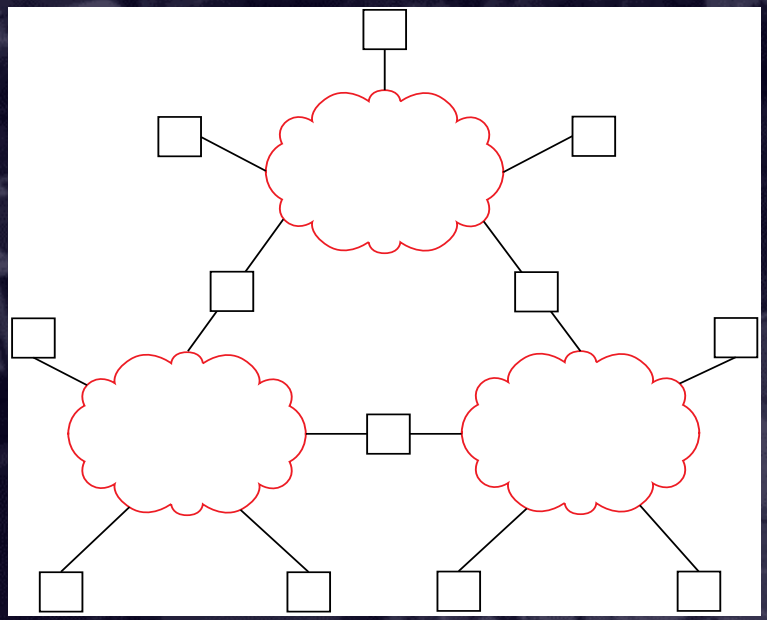
- Time divided into slots
- Begins with N short reservation slots
  - reservation slot time equal to channel end-end propagation delay
  - station with message to send posts reservation
  - reservation seen by all stations
- After reservation slots, message transmissions ordered by known priority



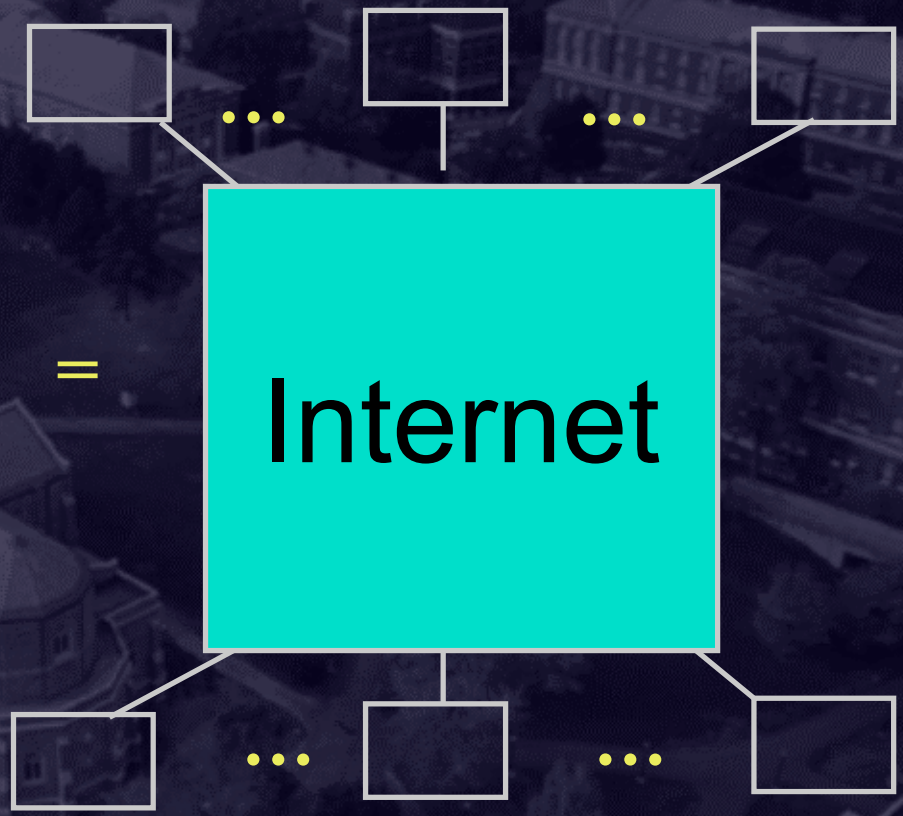
Additions to List of Issues

- Filtering techniques:
  - Learning, routing
- Multiple access
  - How to share a wire
  - Partitioning, Random Access, Taking Turns
- Switching, bridging, routing
- Addressing, Packet Formats

Inter-Networks: *Networks of Networks*



=



Our goal is to design this black box on the right

Inter-Networks: Networks of Networks

- What is it ?
  - "Connect many *disparate* physical networks and make them function as a coordinated unit … " - Douglas Comer
  - *Many => scale*
  - *Disparate => heterogeneity*

- Result: *Universal connectivity!*
  - The inter-network looks like one large switch,
  - User interface is sub-network independent

Inter-Networks: Networks of Networks

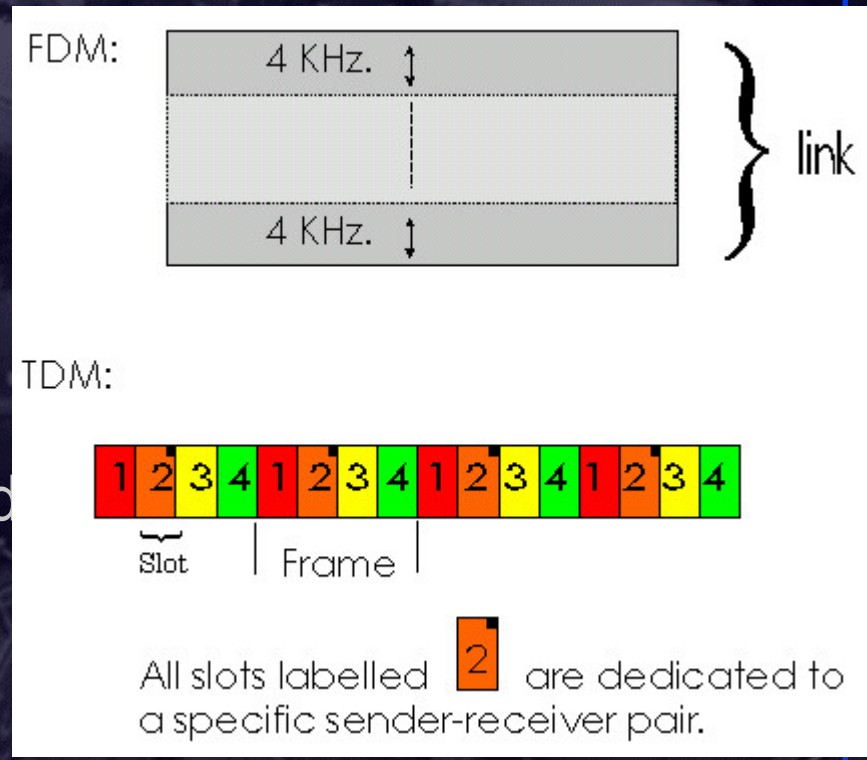
- Internetworking involves two fundamental problems: ***heterogeneity and scale***

- Concepts:
  - Translation, overlays, address & name resolution, fragmentation: to handle *heterogeneity*
  - Hierarchical addressing, routing, naming, address allocation, congestion control: to handle *scaling*

- Two broad approaches: circuit-switched and packet-switched

How to design large inter-networks?
Circuit-Switching

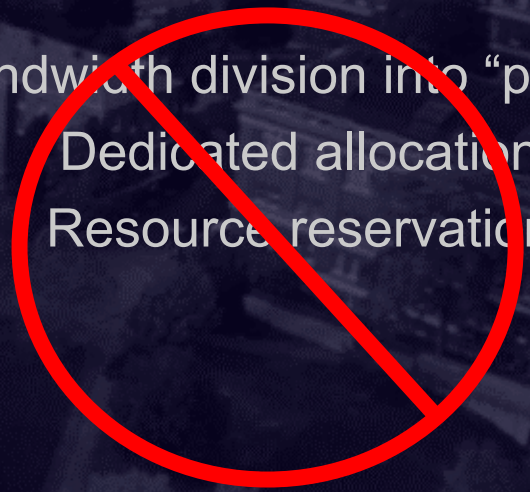
- Divide link bandwidth into "pieces"
- Reserve pieces on successive links and tie them together to form a "*circuit*"
- Map traffic into the reserved circuits
- Resources wasted if unused: *expensive*.



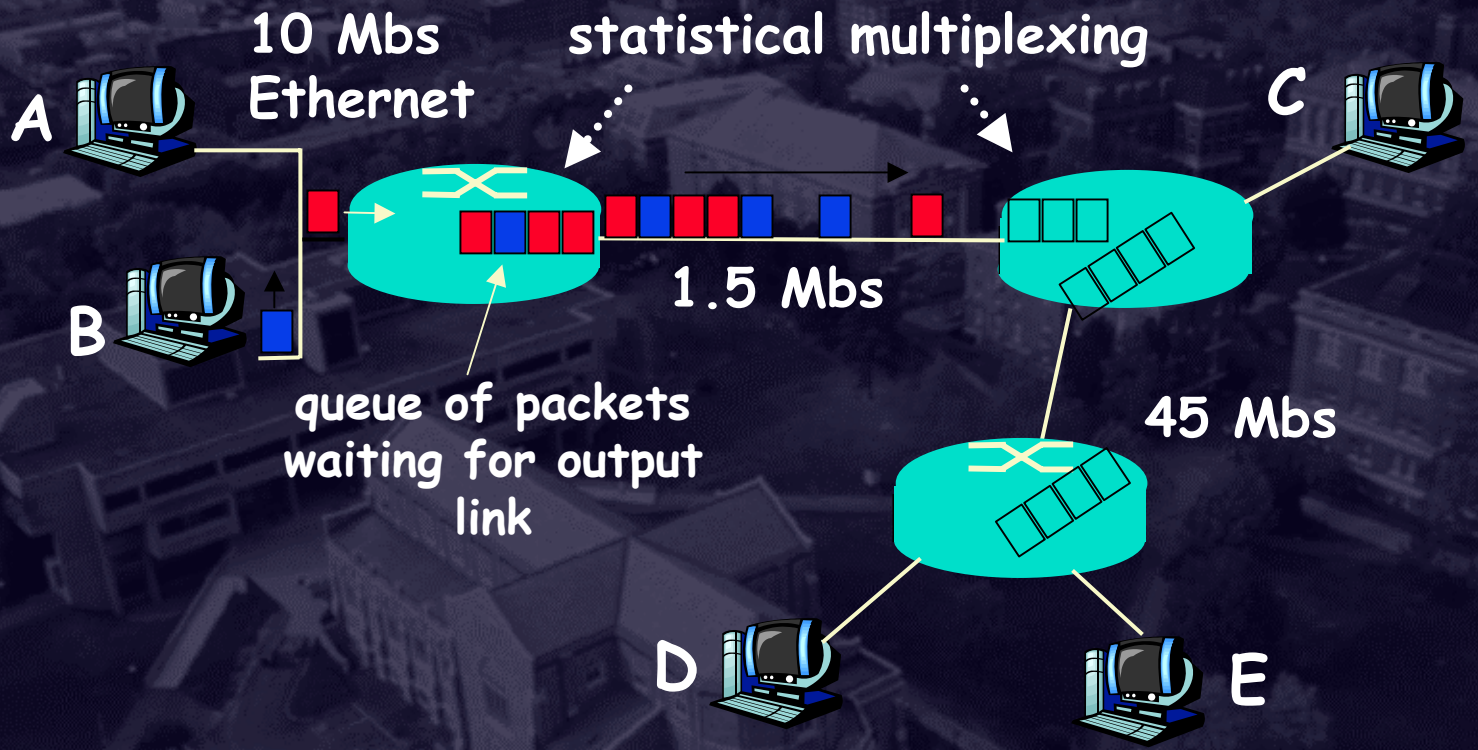
– **Mapping can be done without "headers".**
– **Everything inferred from timing.**

How to design large inter-networks?
Packet-Switching

- Chop up data (not links!) into "packets"
  - Packets: data + meta-data (header)
- "Switch" packets at intermediate nodes
  - *Store-and-forward* if bandwidth is not immediately available.



Packet Switching



- **Cost:** self-descriptive header per-packet, buffering and delays due to statistical multiplexing at switches.
- Need to either reserve resources or dynamically detect and adapt to overload for stability

Spatial vs Temporal Multiplexing

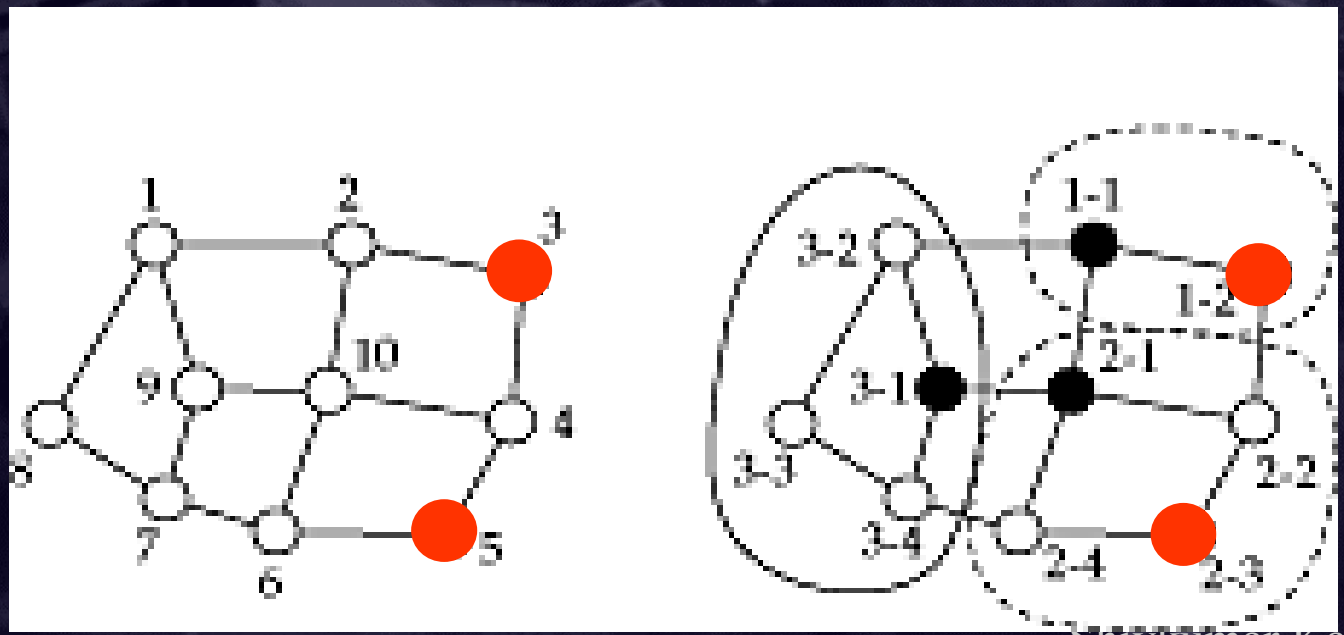
- Spatial multiplexing: Chop up resource into chunks. Eg: bandwidth, cake, circuits…
- Temporal multiplexing: resource is shared over time, I.e. queue up jobs and provide access to resource over time. Eg: FIFO queueing, packet switching
- Packet switching is designed to exploit *both* spatial & temporal multiplexing gains, provided performance tradeoffs are acceptable to applications.
- *Packet switching is potentially more efficient => potentially more scalable than circuit switching !*

Scalable Forwarding, Structured Addresses

- Address has structure which aids the forwarding process.
- Address assignment is done such that nodes which can be reached without resorting to L3 forwarding have the same prefix (network ID)
- A simple comparison of network ID of destination and current network (broadcast domain) identifies whether the destination is "directly" connected
  - I.e. Reachable through L2 forwarding only
- Within L3 forwarding, further structure can aid hierarchical organization of routing domains (because routing algorithms have other scalability issues)

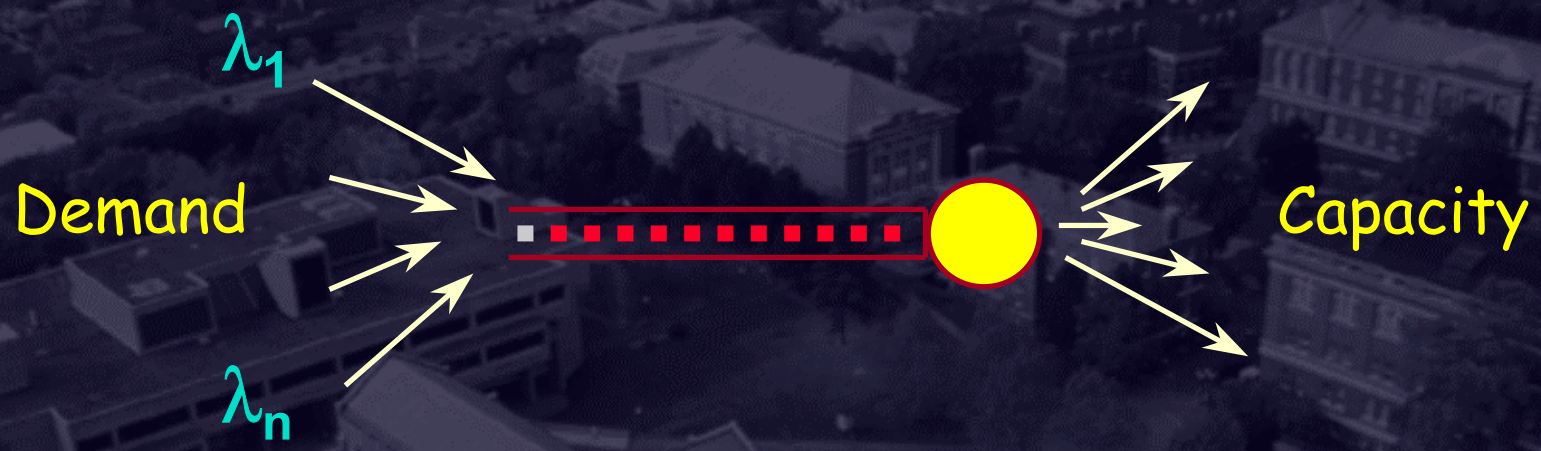
Flat vs Structured Addresses

- Flat addresses: no structure in them to facilitate scalable routing
  - Eg: IEEE 802 LAN addresses
- Hierarchical addresses:
  - Network part (prefix) and host part
  - Helps identify direct or indirectly connected nodes



The Congestion Problem

- ***Problem: demand outstrips available capacity***



- If information about λ_i , λ and μ is known in a central location where control of λ_i or μ can be effected with zero time delays,
  - the congestion problem is solved!

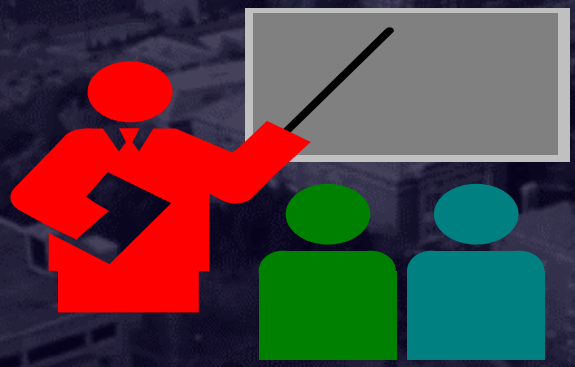
The Congestion Problem (Continued)

- Problems:
  - Incomplete information (eg: loss indications)
  - Distributed solution required
  - Congestion and control/measurement locations different
  - Time-varying, heterogeneous time-delay

Additions to Problem List

- Internetworking problems: heterogeneity, scale.
- Circuit Switching vs Packet Switching
- Heterogeneity:
  - Overlay model, Translation, Address Resolution, Fragmentation
- Scale:
  - Structured addresses, hierarchical routing
  - Naming, addressing
  - Congestion control

Summary: Laundry List of Problems



- Basics: Direct/indirect connectivity, topologies
- Link layer issues:
  - Framing, Error control, Flow control
- Multiple access & Ethernet:
  - Cabling, Pkt format, Switching, bridging vs routing
- Internetworking problems: Naming, addressing, Resolution, fragmentation, congestion control, traffic management, Reliability, Network Management