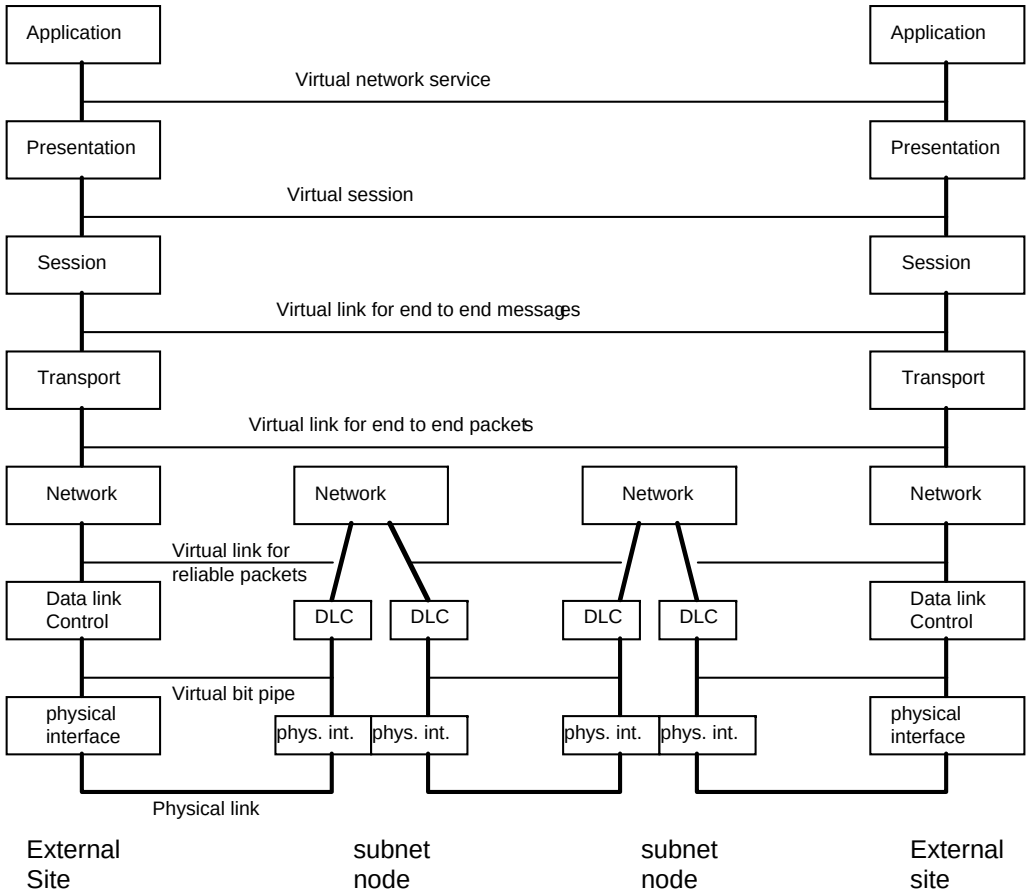
Higher Layer Protocols
TCP/IP and ATM

Eytan Modiano
Massachusetts Institute of Technology
Laboratory for Information and Decision Systems

Outline

- Network Layer and Internetworking
- The TCP/IP protocol suit
- ATM
- MPLS

Higher Layers



TCP, UDP

IP, ATM

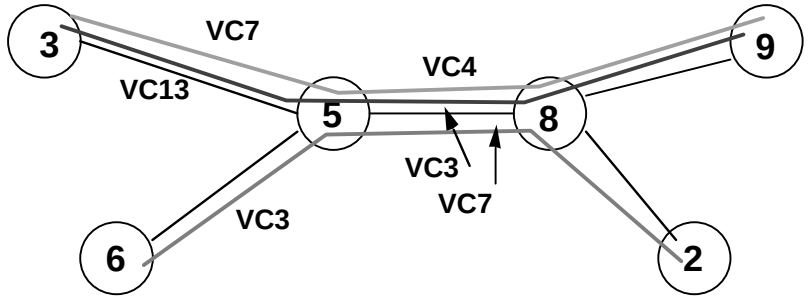
Packet Switching

- **Datagram packet switching**
  - **Route chosen on packet-by-packet basis**
  - **Different packets may follow different routes**
  - **Packets may arrive out of order at the destination**
  - **E.g., IP (The Internet Protocol)**

- **Virtual Circuit packet switching**
  - **All packets associated with a session follow the same path**
  - **Route is chosen at start of session**
  - **Packets are labeled with a VC# designating the route**
  - **The VC number must be unique on a given link but can change from link to link**
    - **Imagine having to set up connections between 1000 nodes in a mesh**
    - **Unique VC numbers imply 1 Million VC numbers that must be represented and stored at each node**
  - **E.g., ATM (Asynchronous transfer mode)**

Virtual Circuits Packet Switching

- For datagrams, addressing information must uniquely distinguish each network node and session
  - Need unique source and destination addresses
- For virtual circuits, only the virtual circuits on a link need be distinguished by addressing
  - Global address needed to set-up virtual circuit
  - Once established, local virtual circuit numbers can then be used to represent the virtual circuits on a given link: VC number changes from link to link
- Merits of virtual circuits
  - Save on route computation
    - Need only be done once at start of session
  - Save on header size
  - More complex
  - Less flexible



Node 5 table

(3,5) VC13 -> (5,8) VC3
(3,5) VC7 -> (5,8) VC4
(6,5) VC3 -> (5,8) VC7

The TCP/IP Protocol Suite

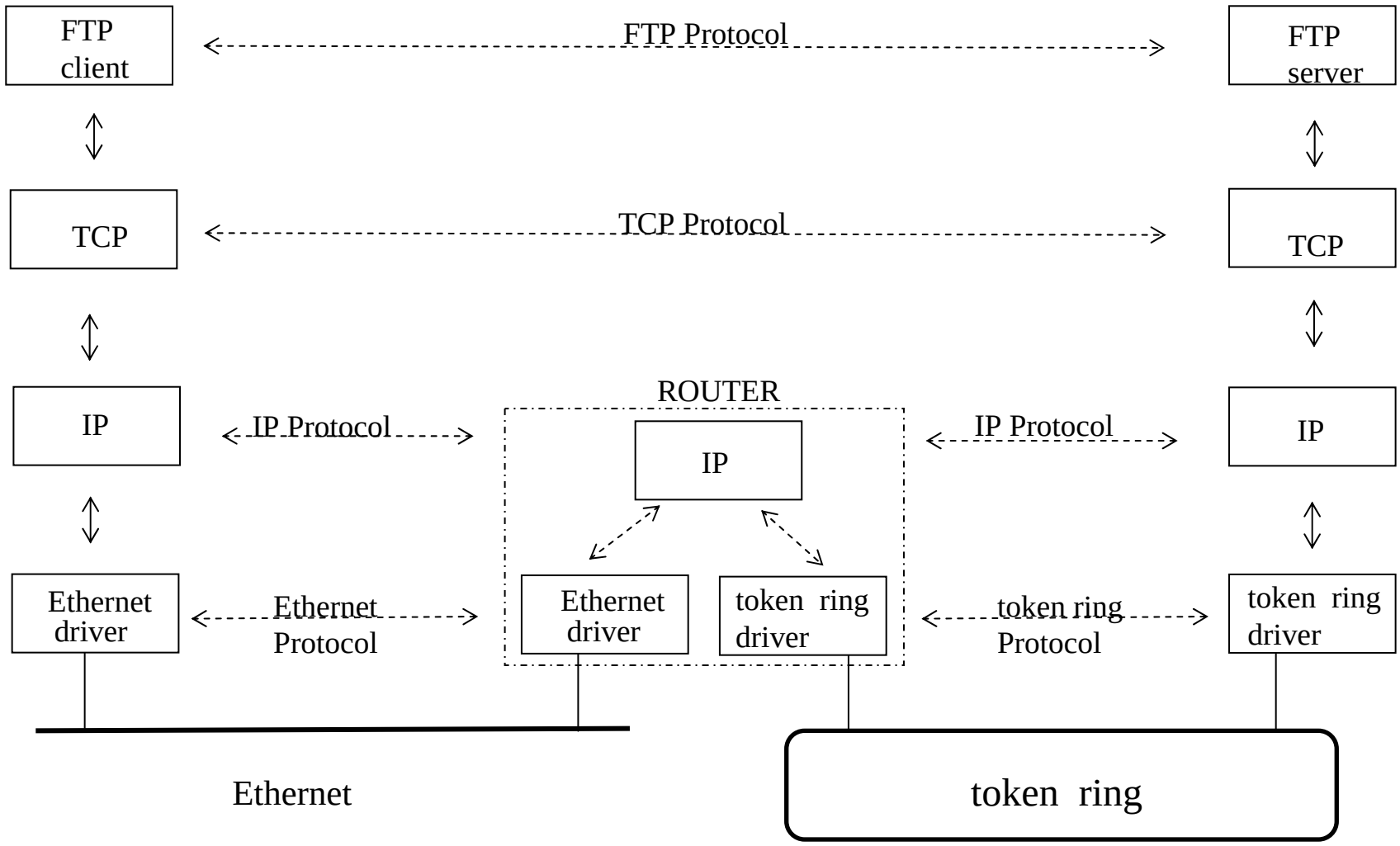
- **Transmission Control Protocol / Internet Protocol**
- **Developed by DARPA to connect Universities and Research Labs**

Four Layer model

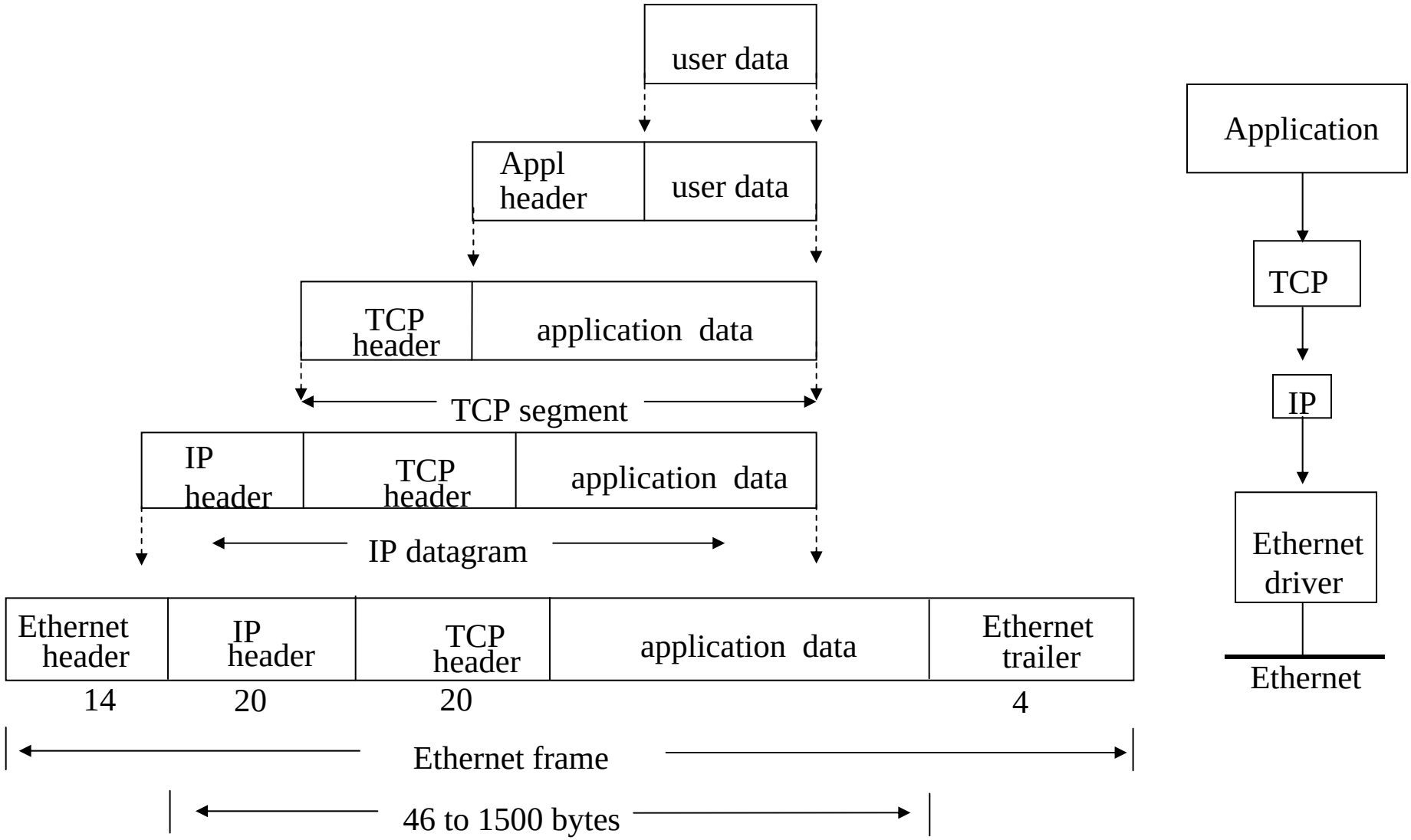
Applications	**Telnet, FTP, email, etc.**
Transport	**TCP, UDP**
Network	**IP, ICMP, IGMP**
Link	**Device drivers, interface cards**

TCP - Transmission Control Protocol
UDP - User Datagram Protocol
IP - Internet Protocol

Internetworking with TCP/IP



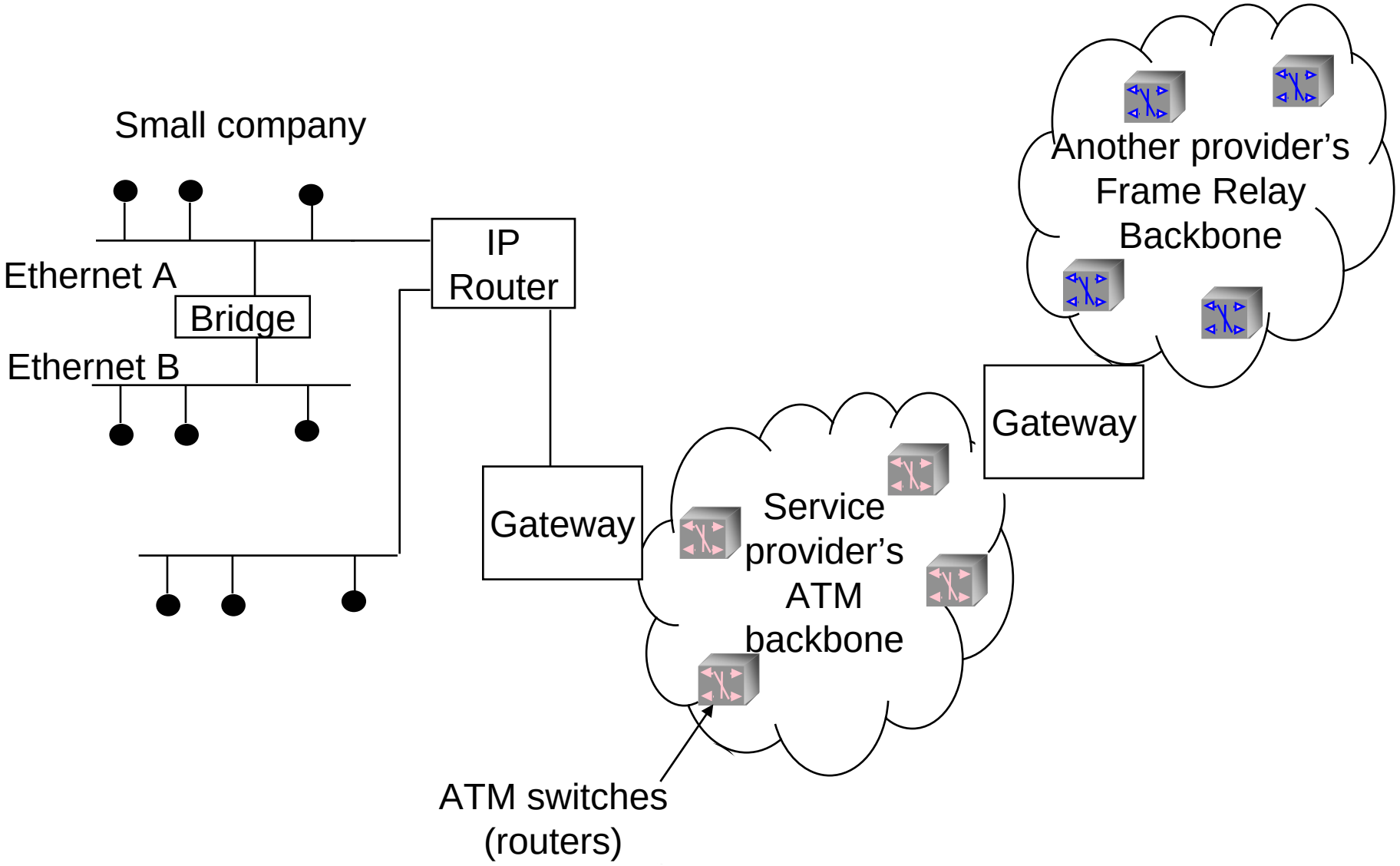
Encapsulation



Bridges, Routers and Gateways

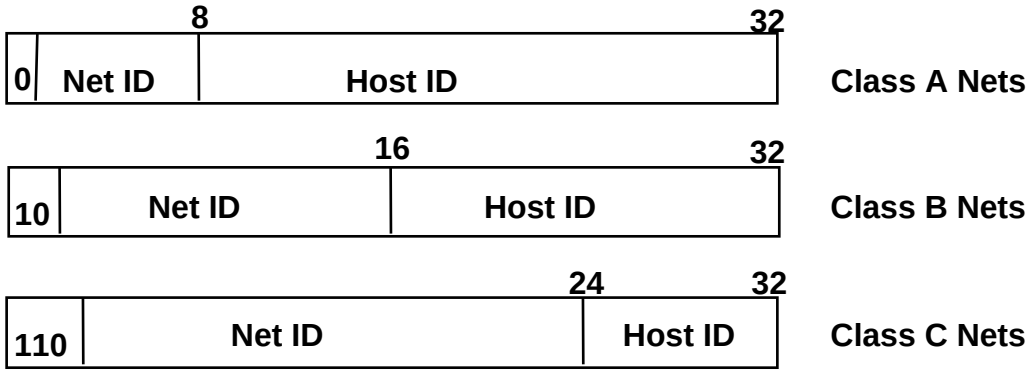
- A Bridge is used to connect multiple LAN segments
  - Layer 2 routing (Ethernet)
  - Does not know IP address
  - Varying levels of sophistication
    - Simple bridges just forward packets
    - smart bridges start looking like routers
- A Router is used to route connect between different networks using network layer address
  - Within or between Autonomous Systems
  - Using same protocol (e.g., IP, ATM)
- A Gateway connects between networks using different protocols
  - Protocol conversion
  - Address resolution
- These definitions are often mixed and seem to evolve!

Bridges, routers and gateways



IP addresses

- 32 bit address written as four decimal numbers
  - One per byte of address (e.g., 155.34.60.112)
- Hierarchical address structure
  - Network ID/ Host ID/ Port ID
  - Complete address called a socket
  - Network and host ID carried in IP Header
  - Port ID (sending process) carried in TCP header
- IP Address classes:



Class D is for multicast traffic

Host Names

- Each machine also has a unique name
- Domain name System: A distributed database that provides a mapping between IP addresses and Host names
- E.g., 155.34.50.112 => plymouth.ll.mit.edu

Internet Standards

- **Internet Engineering Task Force (IETF)**
  - **Development on near term internet standards**
  - **Open body**
  - **Meets 3 times a year**
- **Request for Comments (RFCs)**
  - **Official internet standards**
  - **Available from IETF web page: http://www.ietf.org**

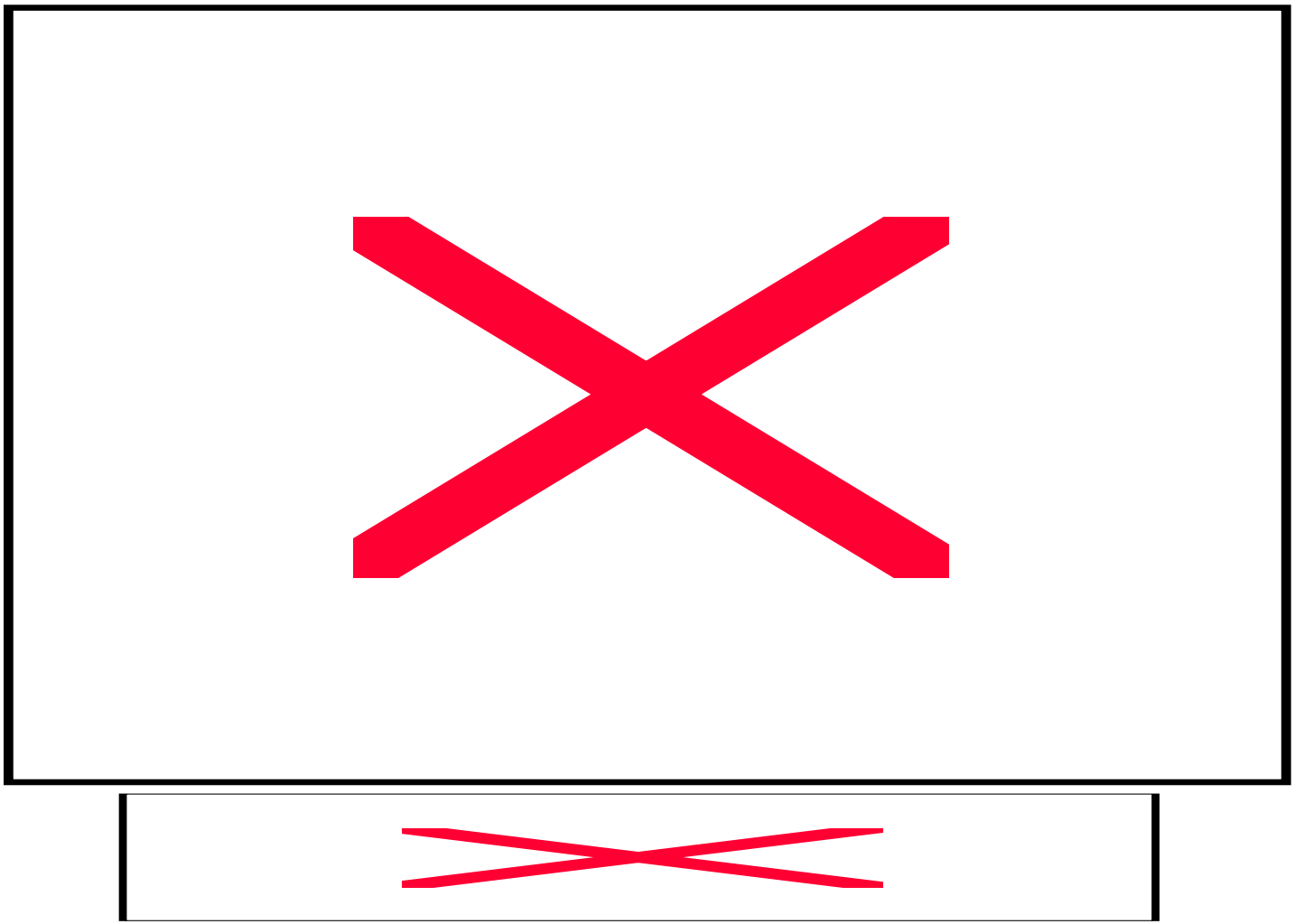
The Internet Protocol (IP)

- Routing of packet across the network
- Unreliable service
  - Best effort delivery
  - Recovery from lost packets must be done at higher layers
- Connectionless
  - Packets are delivered (routed) independently
  - Can be delivered out of order
  - Re-sequencing must be done at higher layers

- Current version V4

- Future V6
  - Add more addresses (40 byte header!)
  - Ability to provide QoS

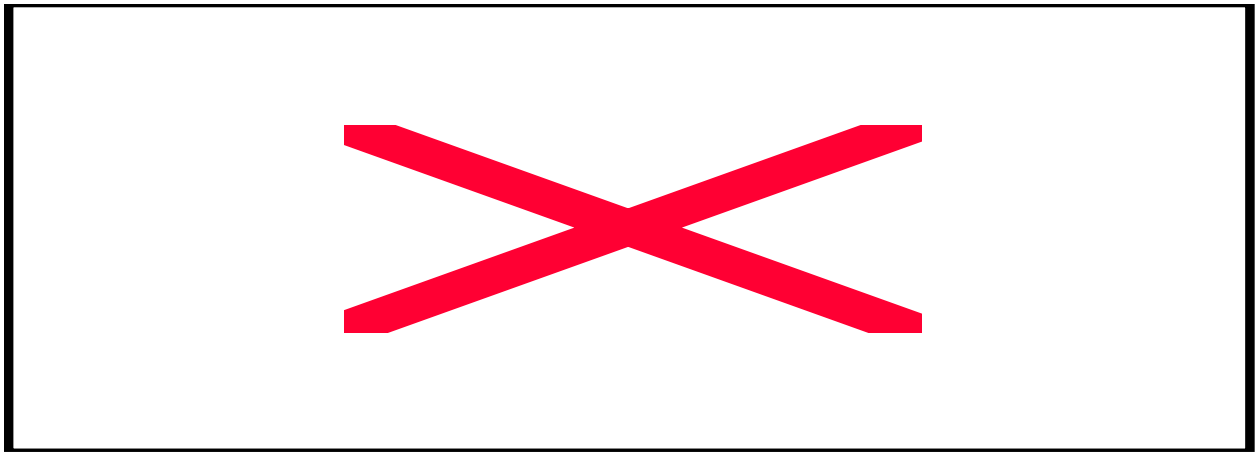
Header Fields in IP



IP HEADER FIELDS

- **Vers:** Version # of IP (current version is 4)
- **HL:** Header Length in 32-bit words
- **Service:** Mostly Ignored
- **Total length** Length of IP datagram
- **ID** Unique datagram ID
- **Flags:** NoFrag, More
- **FragOffset:** Fragment offset in units of 8 Octets
- **TTL:** Time to Live in "seconds" or Hops
- **Protocol:** Higher Layer Protocol ID #
- **HDR Cksum:** 16 bit 1's complement checksum (on header only!)
- **SA & DA:** Network Addresses
- **Options:** Record Route,Source Route,TimeStamp

FRAGMENTATION



- A gateway fragments a datagram if length is too great for next network (fragmentation required because of unknown paths).
- Each fragment needs a unique identifier for datagram plus identifier for position within datagram
- In IP, the datagram ID is a 16 bit field counting datagram from given host

POSITION OF FRAGMENT

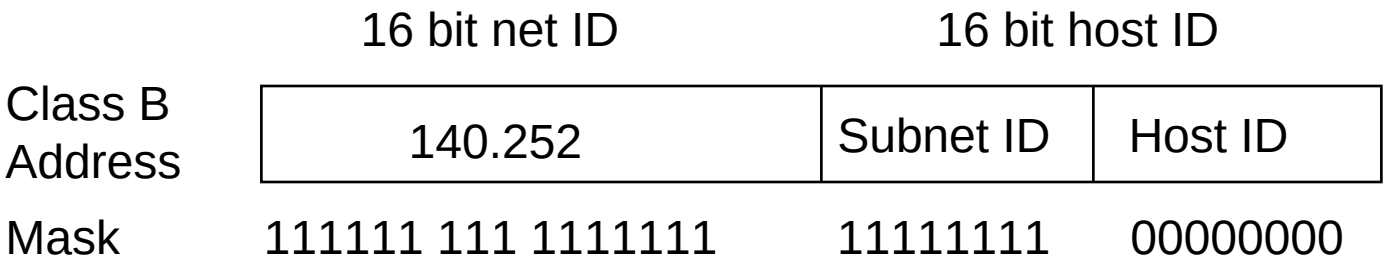
- Fragment offset field gives starting position of fragment within datagram in 8 byte increments (13 bit field)
- Length field in header gives the total length in bytes (16 bit field)
  - Maximum size of IP packet 64K bytes
- A flag bit denotes last fragment in datagram
- IP reassembles fragments at destination and throws them away if one or more is too late in arriving

IP Routing

- Routing table at each node contains for each destination the next hop router to which the packet should be sent
  - Not all destination addresses are in the routing table
    - Look for net ID of the destination “Prefix match”
    - Use default router
- Routers do not compute the complete route to the destination but only the next hop router
- IP uses distributed routing algorithms: RIP, OSPF
- In a LAN, the “host” computer sends the packet to the default router which provides a gateway to the outside world

Subnet addressing

- **Class A and B addresses allocate too many hosts to a given net**
- **Subnet addressing allows us to divide the host ID space into smaller “sub networks”**
  - **Simplify routing within an organization**
  - **Smaller routing tables**
  - **Potentially allows the allocation of the same class B address to more than one organization**
- **32 bit Subnet “Mask” is used to divide the host ID field into subnets**
  - **“1” denotes a network address field**
  - **“0” denotes a host ID field**



Classless inter-domain routing (CIDR)

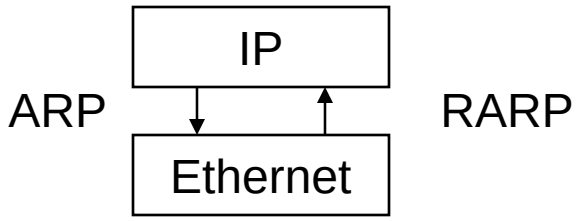
- Class A and B addresses allocate too many hosts to an organization while class C addresses don't allocate enough
  - This leads to inefficient assignment of address space
- Classless routing allows the allocation of addresses outside of class boundaries (within the class C pool of addresses)
  - Allocate a block of contiguous addresses
    - E.g., 192.4.16.1 - 192.4.32.155
    - Bundles 16 class C addresses
    - The first 20 bits of the address field are the same and are essentially the network ID
  - Network numbers must now be described using their length and value (I.e., length of network prefix)
  - Routing table lookup using longest prefix match
- Notice similarity to subnetting - "supernetting"

Dynamic Host Configuration (DHCP)

- Automated method for assigning network numbers
  - IP addresses, default routers
- Computers contact DHCP server at Boot-up time
- Server assigns IP address
- Allows sharing of address space
  - More efficient use of address space
  - Adds scalability
- Addresses are “least” for some time
  - Not permanently assigned

Address Resolution Protocol

- IP addresses only make sense within IP suite
- Local area networks, such as Ethernet, have their own addressing scheme
  - To talk to a node on LAN one must have its physical address (physical interface cards don't recognize their IP addresses)
- ARP provides a mapping between IP addresses and LAN addresses
- RARP provides mapping from LAN addresses to IP addresses
- This is accomplished by sending a "broadcast" packet requesting the owner of the IP address to respond with their physical address
  - All nodes on the LAN recognize the broadcast message
  - The owner of the IP address responds with its physical address
- An ARP cache is maintained at each node with recent mappings

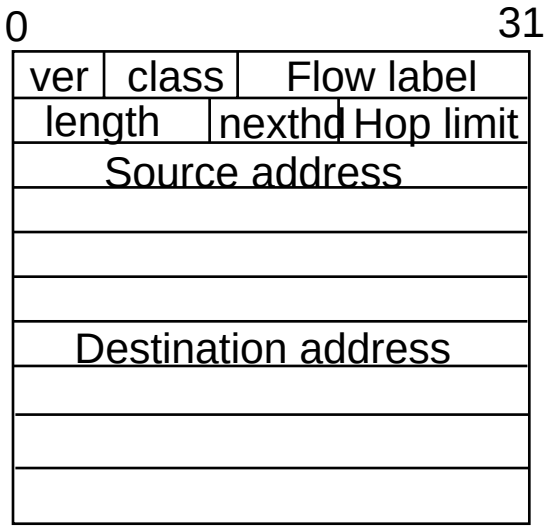


Routing in the Internet

- The internet is divided into sub-networks, each under the control of a single authority known as an Autonomous System (AS)
- Routing algorithms are divided into two categories:
  - Interior protocols (within an AS)
  - Exterior protocols (between AS's)
- Interior Protocols use shortest path algorithms (more later)
  - Distance vector protocols based on Bellman-ford algorithm
    - Nodes exchange routing tables with each other
    - E.g., Routing Information Protocol (RIP)
  - Link state protocols based on Dijkstra's algorithm
    - Nodes monitor the state of their links (e.g., delay)
    - Nodes broadcast this information to all of the network
    - E.g., Open Shortest Path First (OSPF)
- Exterior protocols route packets across AS's
  - Issues: no single cost metric, policy routing, etc..
  - Routes often are pre-computed
  - Example protocols: Exterior Gateway protocol (EGP) and Border Gateway protocol (BGP)

IPv6

- **Effort started in 1991 as IPng**
- **Motivation**
  - **Need to increase IP address space**
  - **Support for real time application - “QoS”**
  - **Security, Mobility, Auto-configuration**
- **Major changes**
  - **Increased address space (16 bytes)**
    - **1500 IP addresses per sq. ft. of earth!**
    - **Address partition similar to CIDR**
  - **Support for QoS via Flow Label field**
  - **Simplified header**
- **Most of the reasons for IPv6 have been taken care of in IPv4**
  - **Is IPv6 really needed?**
  - **Complex transition from V4 to V6**



Resource Reservation (RSVP)

- Service classes (defined by IETF)
  - Best effort
  - Guaranteed service
    - Max packet delay
  - Controlled load
    - emulate lightly loaded network via priority queueing mechanism
- Need to reserve resources at routers along the path
- RSVP mechanism
  - Packet classification
    - Associate packets with sessions (use flow field in IPv6)
  - Receiver initiated reservations to support multicast
  - “soft state” - temporary reservation that expires after 30 seconds
    - Simplify the management of connections
    - Requires refresh messages
  - Packet scheduling to guarantee service
    - Proprietary mechanisms (e.g., Weighted fair queueing)
- Scalability Issues
  - Each router needs to keep track of large number of flows that grows with the size (capacity) of the router

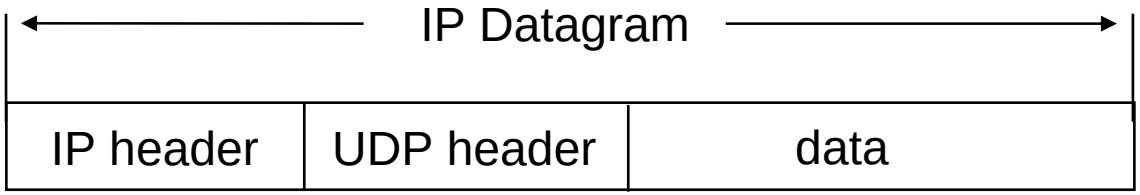
Differentiated Services (Diffserv)

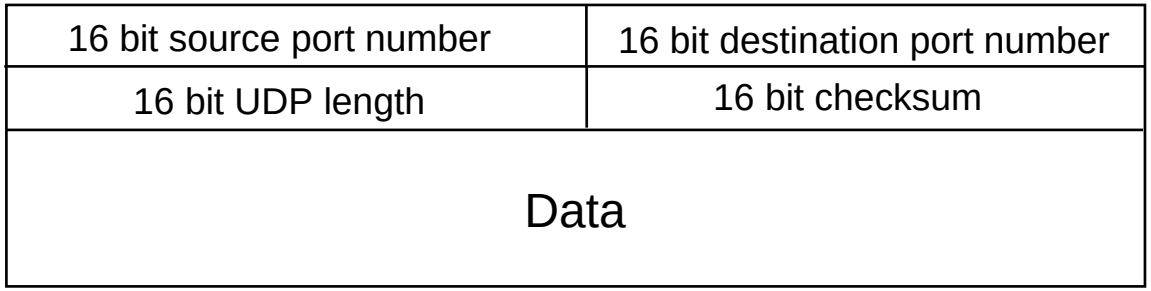
- **Unlike RSVP Diffserv does not need to keep track of individual flows**
  - **Allocate resources to a small number of classes of traffic**
    - **Queue packets of the same class together**
  - **E.g., two classes of traffic - premium and regular**
    - **Use one bit to differential between premium and regular packets**
  - **Issues**
    - **Who sets the premium bit?**
    - **How is premium service different from regular?**
- **IETF propose to use TOS field in IP header to identify traffic classes**
  - **Potentially more than just two classes**

User Datagram Protocol (UDP)

- Transport layer protocol
  - Delivery of messages across network
- Datagram oriented
  - Unreliable
    - No error control mechanism
  - Connectionless
  - Not a “stream” protocol
- Max packet length 65K bytes
- UDP checksum
  - Covers header and data
  - Optional
    - Can be used by applications
- UDP allows applications to interface directly to IP with minimal additional processing or protocol overhead

UDP header format





- **The port numbers identifie the sending and receiving processes**
  - **I.e., FTP, email, etc..**
  - **Allow UDP to multiplex the data onto a single stream**
- **UDP length = length of packet in bytes**
  - **Minimum of 8 and maximum of 2^16 - 1 = 65,535 bytes**
- **Checksum covers header and data**
  - **Optional, UDP does not do anything with the checksum**

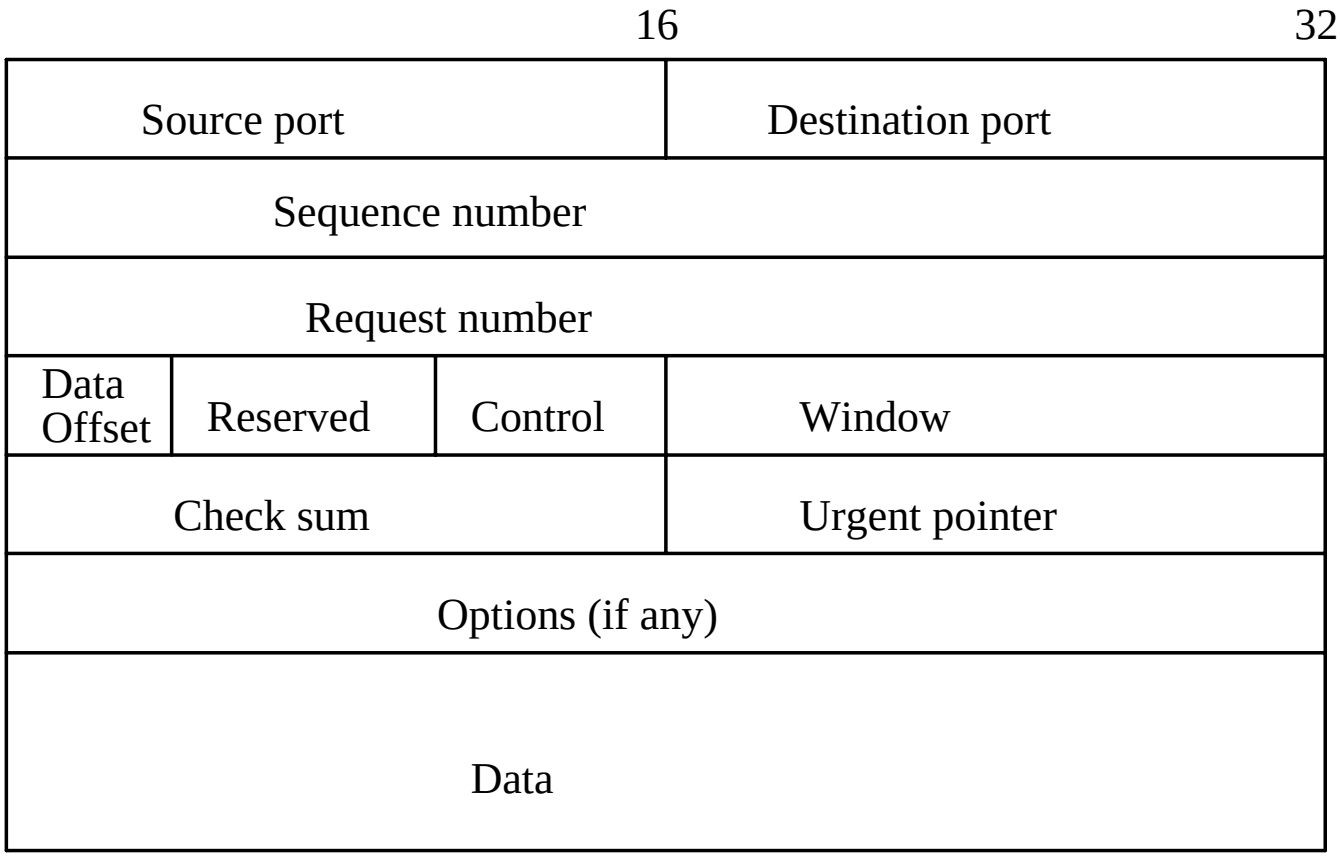
Transmission Control Protocol (TCP)

- **Transport layer protocol**
  - **Reliable transmission of messages**
- **Connection oriented**
  - **Stream traffic**
  - **Must re-sequence out of order IP packets**
- **Reliable**
  - **ARQ mechanism**
  - **Notice that packets have a sequence number and an ack number**
  - **Notice that packet header has a window size (for Go Back N)**
- **Flow control mechanism**
  - **Slow start**
    - **Limits the size of the window in response to congestion**

Basic TCP operation

- At sender
  - Application data is broken into TCP segments
  - TCP uses a timer while waiting for an ACK of every packet
  - Un-ACK'd packets are retransmitted
- At receiver
  - Errors are detected using a checksum
  - Correctly received data is acknowledged
  - Segments are reassembled into their proper order
  - Duplicate segments are discarded
- Window based retransmission and flow control

TCP header fields



TCP header fields

- Ports number are the same as for UDP
- 32 bit SN uniquely identify the application data contained in the TCP segment
  - SN is in bytes!
  - It identify the first byte of data
- 32 bit RN is used for piggybacking ACK's
  - RN indicates the next byte that the received is expecting
  - Implicit ACK for all of the bytes up to that point
- Data offset is a header length in 32 bit words (minimum 20 bytes)
- Window size
  - Used for error recovery (ARQ) and as a flow control mechanism
    - Sender cannot have more than a window of packets in the network simultaneously
  - Specified in bytes
    - Window scaling used to increase the window size in high speed networks
- Checksum covers the header and data

TCP error recovery

- Error recovery is done at multiple layers
  - Link, transport, application
- Transport layer error recovery is needed because
  - Packet losses can occur at network layer
    - E.g., buffer overflow
  - Some link layers may not be reliable
- SN and RN are used for error recovery in a similar way to Go Back N at the link layer
  - Large SN needed for re-sequencing out of order packets
- TCP uses a timeout mechanism for packet retransmission
  - Timeout calculation
  - Fast retransmission

TCP timeout calculation

- **Based on round trip time measurement (RTT)**
  - **Weighted average**

$$RTT\_AVE = a*(RTT\_measured) + (1-a)*RTT\_AVE$$

- **Timeout is a multiple of RTT_AVE (usually two)**
  - **Short Timeout would lead to too many retransmissions**
  - **Long Timeout would lead to large delays and inefficiency**

- **In order to make Timeout be more tolerant of delay variations it has been proposed (Jacobson) to set the timeout value based on the standard deviation of RTT**

$$Timeout = RTT\_AVE + 4*RTT\_SD$$

- **In many TCP implementations the minimum value of Timeout is 500 ms due to the clock granularity**

Fast Retransmit

- When TCP receives a packet with a SN that is greater than the expected SN, it sends an ACK packet with a request number of the expected packet SN
  - This could be due to out-of-order delivery or packet loss
- If a packet is lost then duplicate RNs will be sent by TCP until the packet it correctly received
  - But the packet will not be retransmitted until a Timeout occurs
  - This leads to added delay and inefficiency
- Fast retransmit assumes that if 3 duplicate RNs are received by the sending module that the packet was lost
  - After 3 duplicate RNs are received the packet is retransmitted
  - After retransmission, continue to send new data
- Fast retransmit allows TCP retransmission to behave more like Selective repeat ARQ
  - Future option for selective ACKs (SACK)

TCP congestion control

- TCP uses its window size to perform end-to-end congestion control
  - More on window flow control later
- Basic idea
  - With window based ARQ the number of packets in the network cannot exceed the window size (CW)

    Last_byte_sent (SN) - last_byte_ACK'd (RN) <= CW

- Transmission rate when using window flow control is equal to one window of packets every round trip time

    R = CW/RTT

- By controlling the window size TCP effectively controls the rate

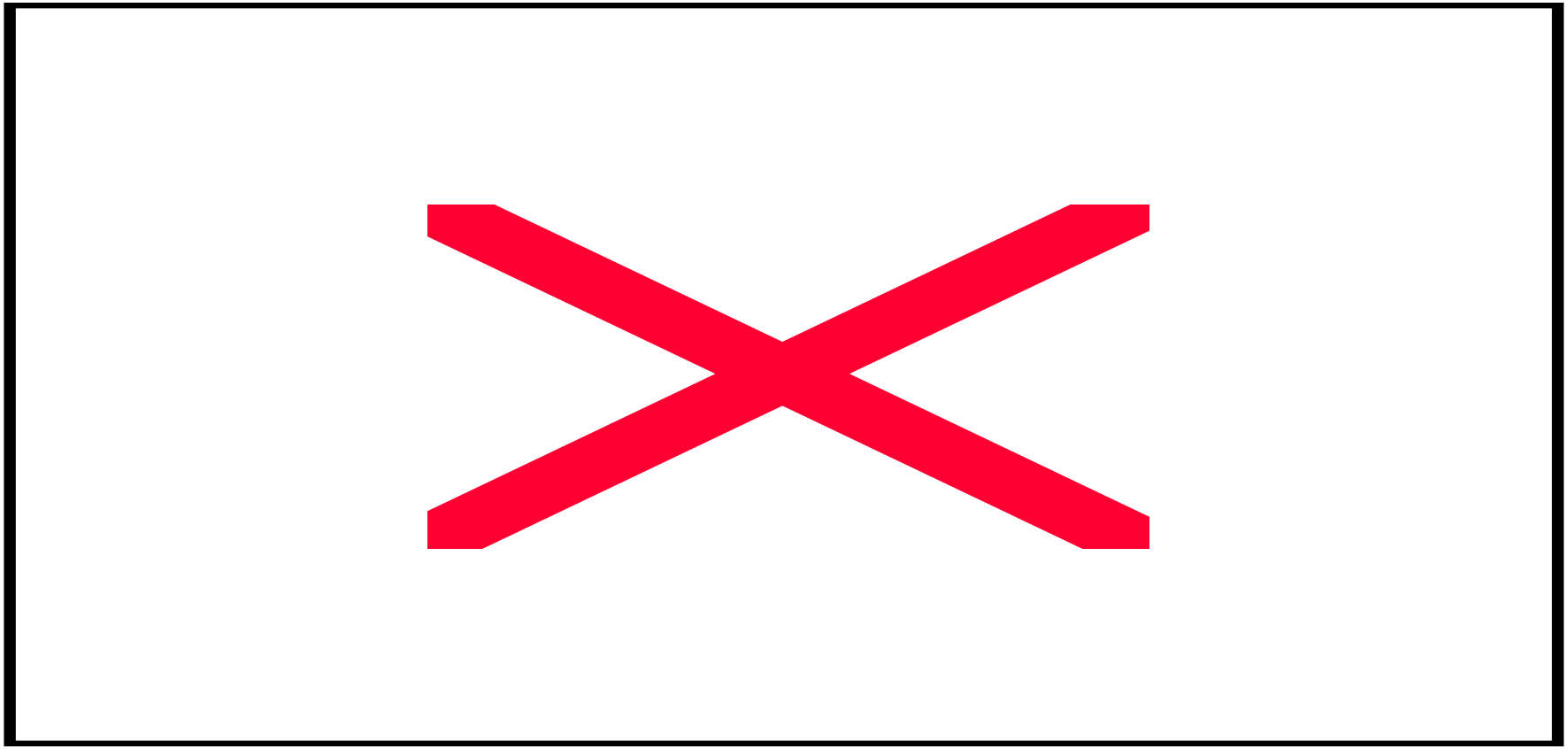
Effect Of Window Size

- **The window size is the number of bytes that are allowed to be in transport simultaneously**

- **Too small a window prevents continuous transmission**

- **To allow continuous transmission window size must exceed round-trip delay time**

Length of a bit (traveling at 2/3C)



Dynamic adjustment of window size

- TCP starts with CW = 1 packet and increases the window size slowly as ACK's are received
  - Slow start phase
  - Congestion avoidance phase
- Slow start phase
  - During slow start TCP increases the window by one packet for every ACK that is received
  - When CW = Threshold TCP goes to Congestion avoidance phase
  - Notice: during slow start CW doubles every round trip time

    Exponential increase!
- Congestion avoidance phase
  - During congestion avoidance TCP increases the window by one packet for every window of ACKs that it receives
  - Notice that during congestion avoidance CW increases by 1 every round trip time - Linear increase!
- TCP continues to increase CW until congestion occurs

Reaction to congestion

- Many variations: Tahoe, Reno, Vegas
- Basic idea: when congestion occurs decrease the window size
- There are two congestion indication mechanisms
  - Duplicate ACKs - could be due to temporary congestion
  - Timeout - more likely due to significant congstion

- TCP Reno - most common implementation

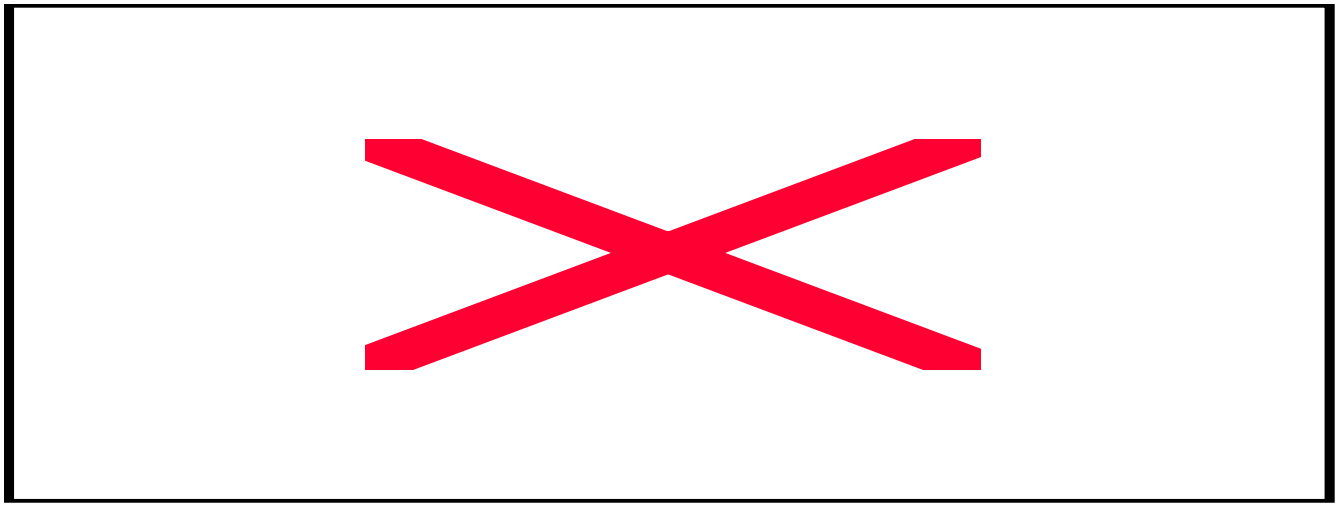
  - If Timeout occurs, CW = 1 and go back to slow start phase

  - If duplicate ACKs occur CW = CW/2 stay in congestion avoidance phase

Understanding TCP dynamics

- Slow start phase is actually fast
- TCP spends most of its time in Congestion avoidance phase
- While in Congestion avoidance
  - CW increases by 1 every RTT
  - CW decreases by a factor of two with every loss

"Additive Increase / Multiplicative decrease"

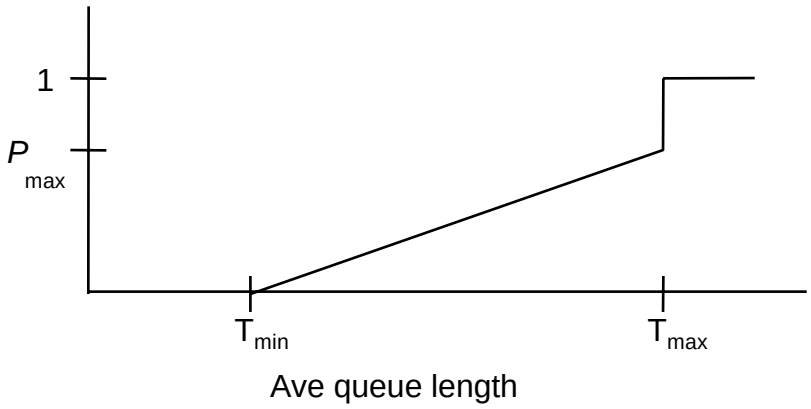


Random Early Detection (RED)

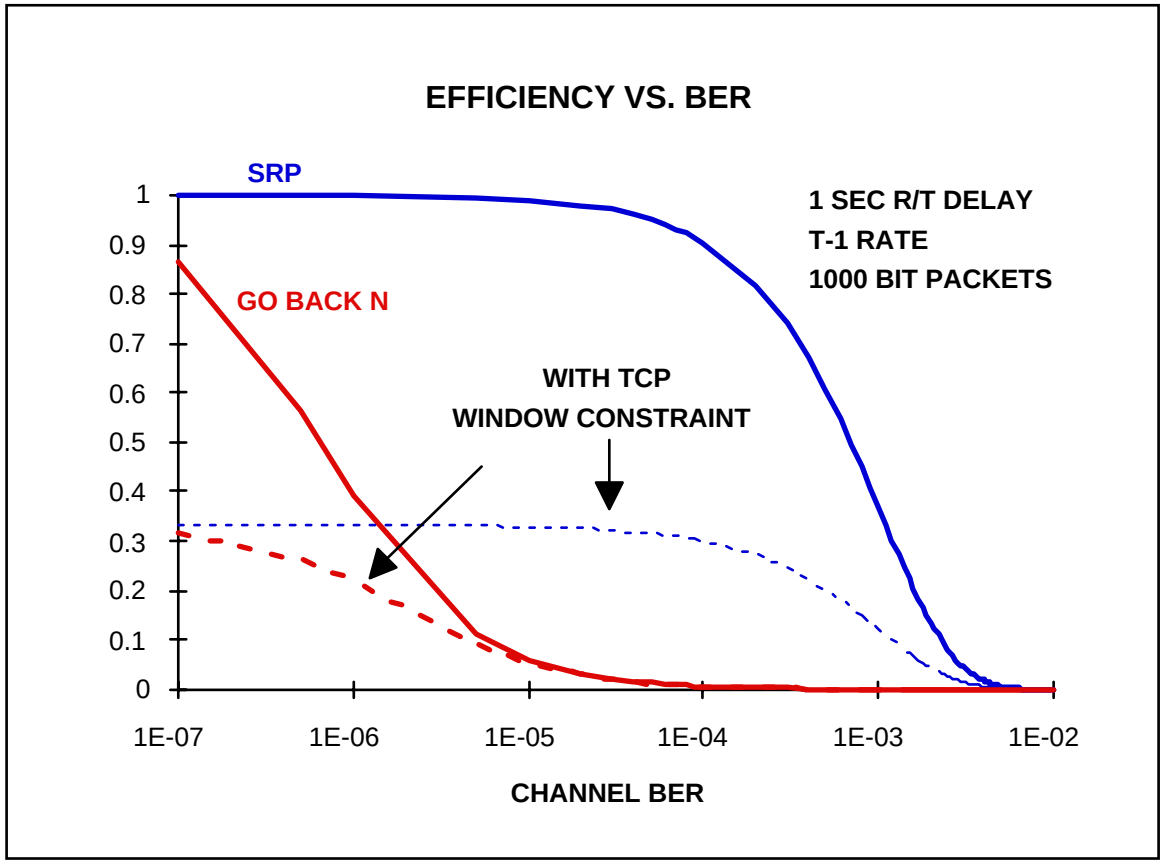
- **Instead of dropping packet on queue overflow, drop them probabilistically earlier**

- **Motivation**
  - **Dropped packets are used as a mechanism to force the source to slow down**
    - **If we wait for buffer overflow it is in fact too late and we may have to drop many packets**
    - **Leads to TCP synchronization problem where all sources slow down simultaneously**
  - **RED provides an early indication of congestion**
    - **Randomization reduces the TCP synchronization problem**

- **Mechanism**
  - **Use weighted average queue size**
    - **If AVE_Q > T_{min} drop with prob. P**
    - **If AVE_Q > T_{max} drop with prob. 1**
  - **RED can be used with explicit congestion notification rather than packet dropping**
  - **RED has a fairness property**
    - **Large flows more likely to be dropped**
  - **Threshold and drop probability values are an area of active research**

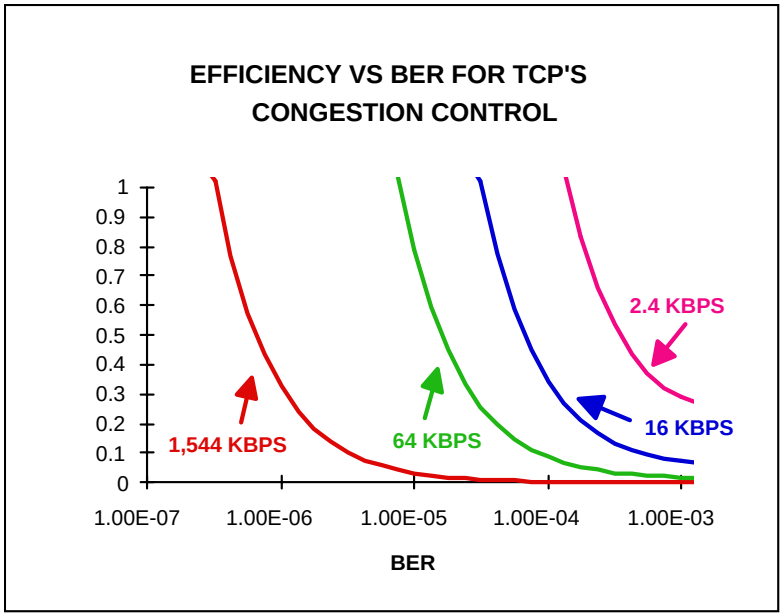


TCP Error Control



- Original TCP designed for low BER, low delay links
- Future versions (RFC 1323) will allow for larger windows and selective retransmissions

Impact of transmission errors on TCP congestion control

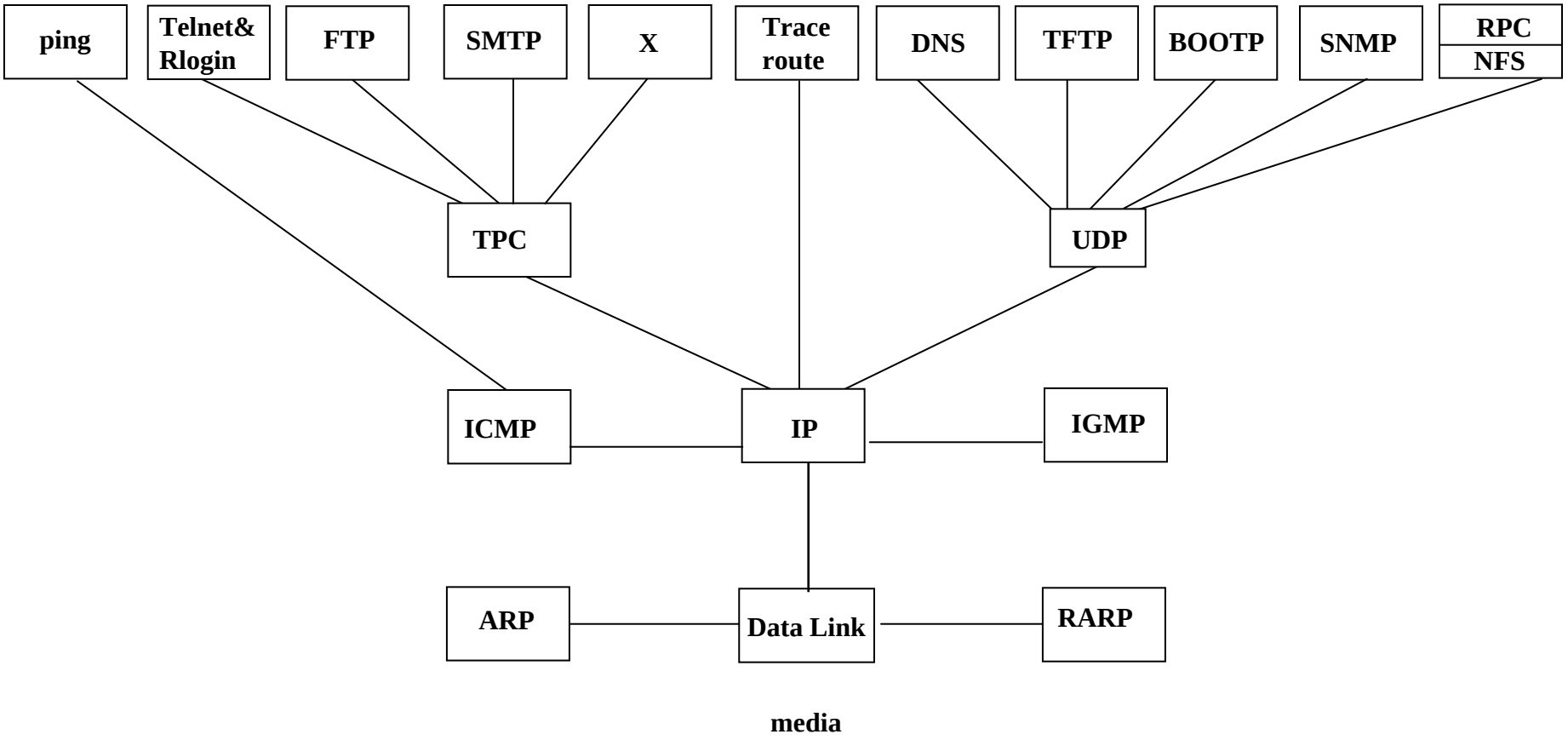


- TCP assumes dropped packets are due to congestion and responds by reducing the transmission rate
- Over a high BER link dropped packets are more likely to be due to errors than to congestion
- TCP extensions (RFC 1323)
  - Fast retransmit mechanism, fast recovery, window scaling

TCP releases

- TCP standards are published as RFC's
- TCP implementations sometimes differ from one another
  - May not implement the latest extensions, bugs, etc.
- The de facto standard implementation is BSD
  - Computer system Research group at UC-Berkeley
  - Most implementations of TCP are based on the BSD implementations
    SUN, MS, etc.
- BSD releases
  - 4.2BSD - 1983
    First widely available release
  - 4.3BSD Tahoe - 1988
    Slow start and congestion avoidance
  - 4.3BSD Reno - 1990
    Header compression
  - 4.4BSD - 1993
    Multicast support, RFC 1323 for high performance

The TCP/IP Suite

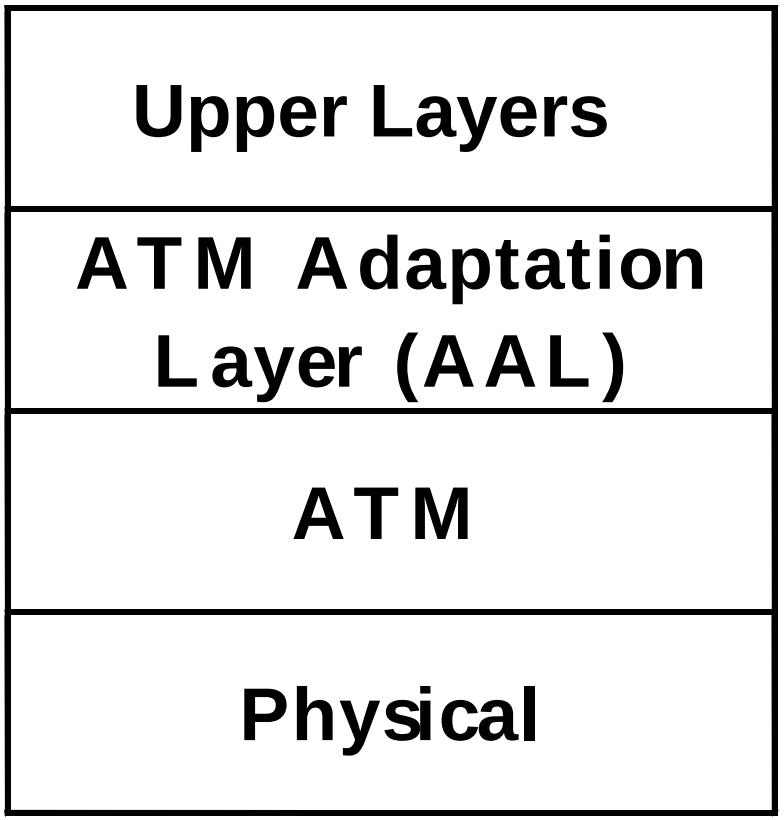


Asynchronous Transfer Mode (ATM)

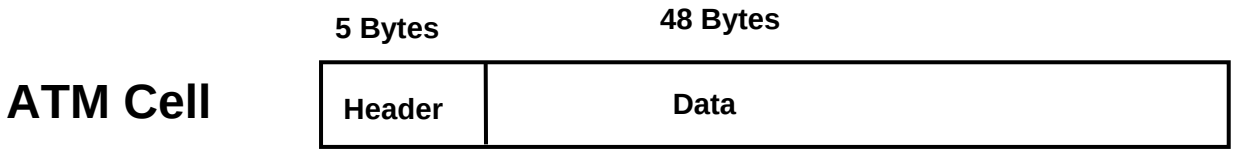
- 1980's effort by the phone companies to develop an integrated network standard (BISDN) that can support voice, data, video, etc.
- ATM uses small (53 Bytes) fixed size packets called "cells"
  - Why cells?
    - Cell switching has properties of both packet and circuit switching
    - Easier to implement high speed switches
  - Why 53 bytes?
  - Small cells are good for voice traffic (limit sampling delays)
    - For 64Kbps voice it takes 6 ms to fill a cell with data
- ATM networks are connection oriented
  - Virtual circuits

ATM Reference Architecture

- **Upper layers**
  - **Applications**
  - **TCP/IP**
- **ATM adaptation layer**
  - **Similar to transport layer**
  - **Provides interface between upper layers and ATM**
    - **Break messages into cells and reassemble**
- **ATM layer**
  - **Cell switching**
  - **Congestion control**
- **Physical layer**
  - **ATM designed for SONET**
    - **Synchronous optical network**
    - **TDMA transmission scheme with 125 μs frames**

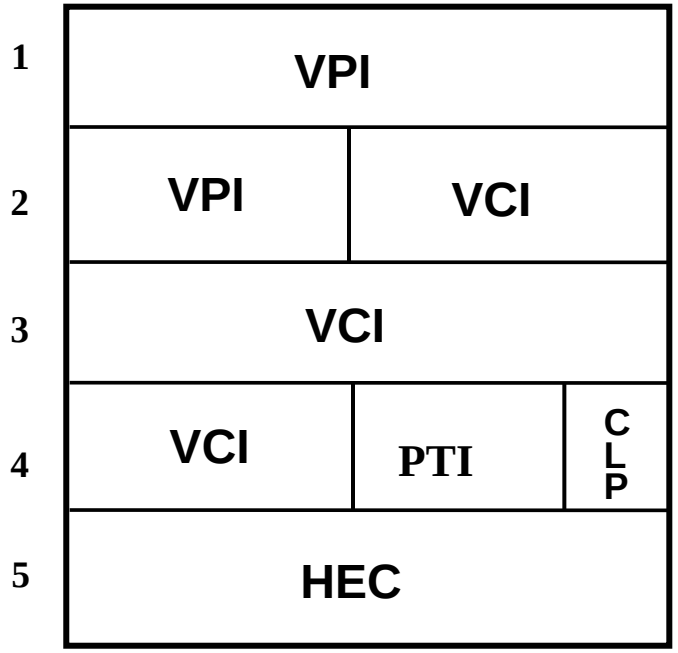


ATM Cell format



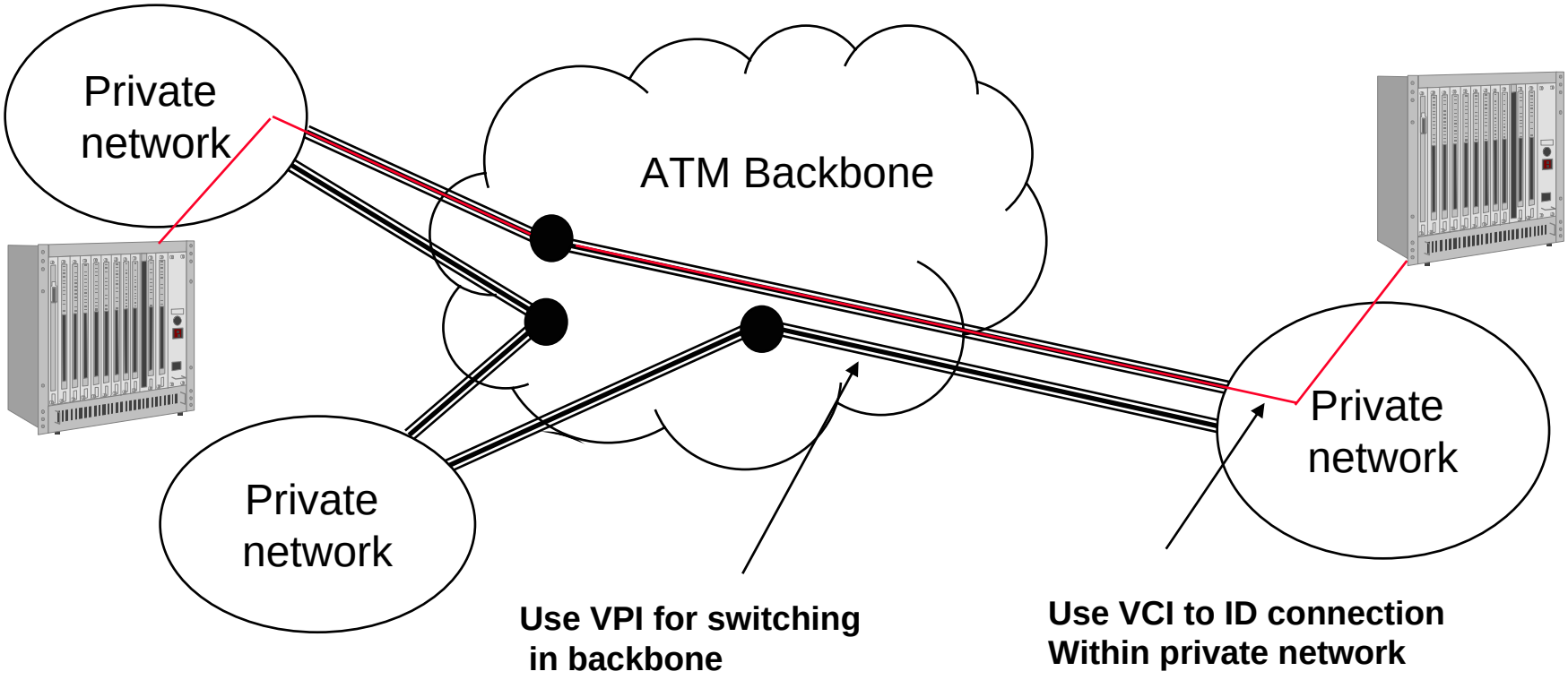
- Virtual circuit numbers (notice relatively small address space!)
  - Virtual channel ID
  - Virtual path ID
- PTI - payload type
- CLP - cell loss priority (1 bit!)
  - Mark cells that can be dropped
- HEC - CRC on header

ATM Header (NNI)



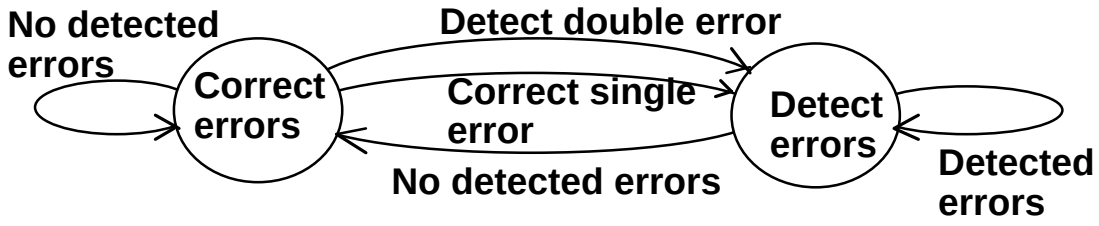
VPI/VCI

- **VPI identifies a physical path between the source and destination**
- **VCI identifies a logical connection (session) within that path**
  - **Approach allows for smaller routing tables and simplifies route computation**



ATM HEADER CRC

- ATM uses an 8 bit CRC that is able to correct 1 error
- It checks only on the header of the cell, and alternates between two modes
  - In detection mode it does not correct any errors but is able to detect more errors
  - In correction mode it can correct up to one error reliably but is less able to detect errors
- When the channel is relatively good it makes sense to be in correction mode, however when the channel is bad you want to be in detection mode to maximize the detection capability



ATM Service Categories

- **Constant Bit Rate (CBR) - e.g. uncompressed voice**
  - **Circuit emulation**
- **Variable Bit Rate (rt-VBR) - e.g. compressed video**
  - **Real-time and non-real-time**
- **Available Bit Rate (ABR) - e.g. LAN interconnect**
  - **For bursty traffic with limited BW guarantees and congestion control**
- **Unspecified Bit Rate (UBR) - e.g. Internet**
  - **ABR without BW guarantees and congestion control**

ATM service parameters (examples)

- Peak cell rate (PCR)
- Sustained cell rate (SCR)
- Maximum Burst Size (MBS)
- Minimum cell rate (MCR)
- Cell loss rate (CLR)
- Cell transmission delay (CTD)
- Cell delay variation (CDV)

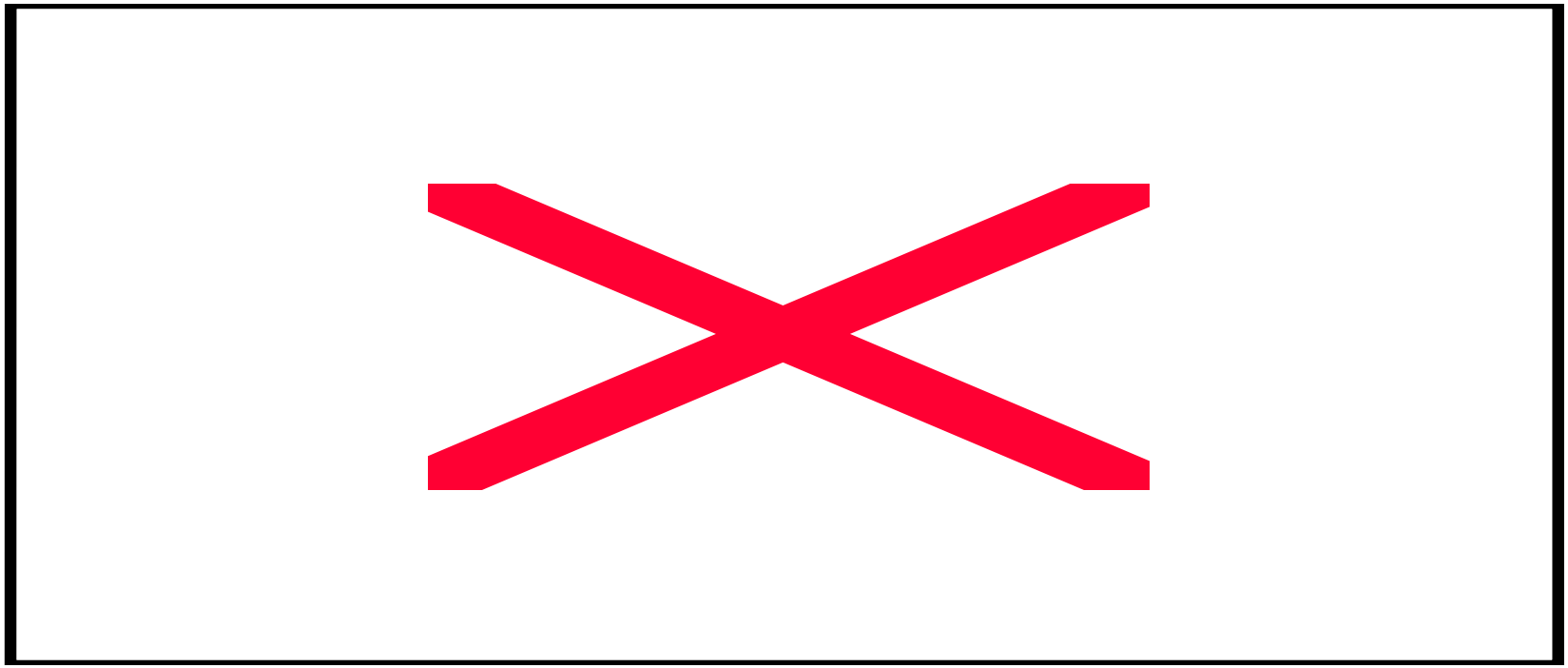
- Not all parameters apply to all service categories
  - E.g., CBR specifies PCR and CDV
  - VBR specifies MBR and SCR

- Network guarantees QoS provided that the user conforms to his contract as specified by above parameters
  - When users exceed their rate network can drop those packets
  - Cell rate can be controlled using rate control scheme (leaky bucket)

Flow control in ATM networks (ABR)

- ATM uses resource management cells to control rate parameters
  - Forward resource management (FRM)
  - Backward resource management (BRM)
- RM cells contain
  - Congestion indicator (CI)
  - No increase Indicator (NI)
  - Explicit cell rate (ER)
  - Current cell rate (CCR)
  - Min cell rate (MCR)
- Source generates RM cells regularly
  - As RM cells pass through the networked they can be marked with CI=1 to indicate congestion
  - RM cells are returned back to the source where
    - CI = 1 => decrease rate by some fraction
    - CI = 1 => Increase rate by some fraction
  - ER can be used to set explicit rate

End-to-End RM-Cell Flow

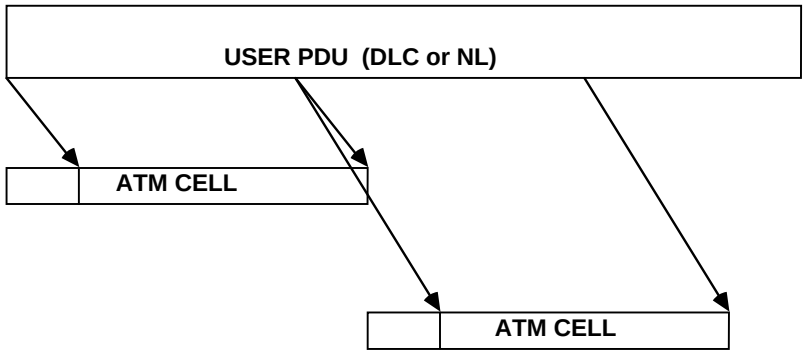


At the destination the RM cell is “turned around” and sent back to the source

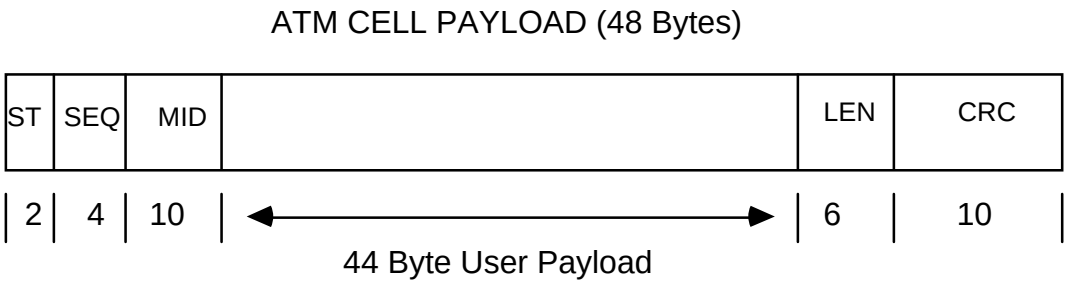
ATM Adaptation Layers

- Interface between ATM layer and higher layer packets
- Four adaptation layers that closely correspond to ATM's service classes
  - AAL-1 to support CBR traffic
  - AAL-2 to support VBR traffic
  - AAL-3/4 to support bursty data traffic
  - AAL-5 to support IP with minimal overhead
- The functions and format of the adaptation layer depend on the class of service.
  - For example, stream type traffic requires sequence numbers to identify which cells have been dropped.

Each class of service has A different header format (in addition to the 5 byte ATM header)



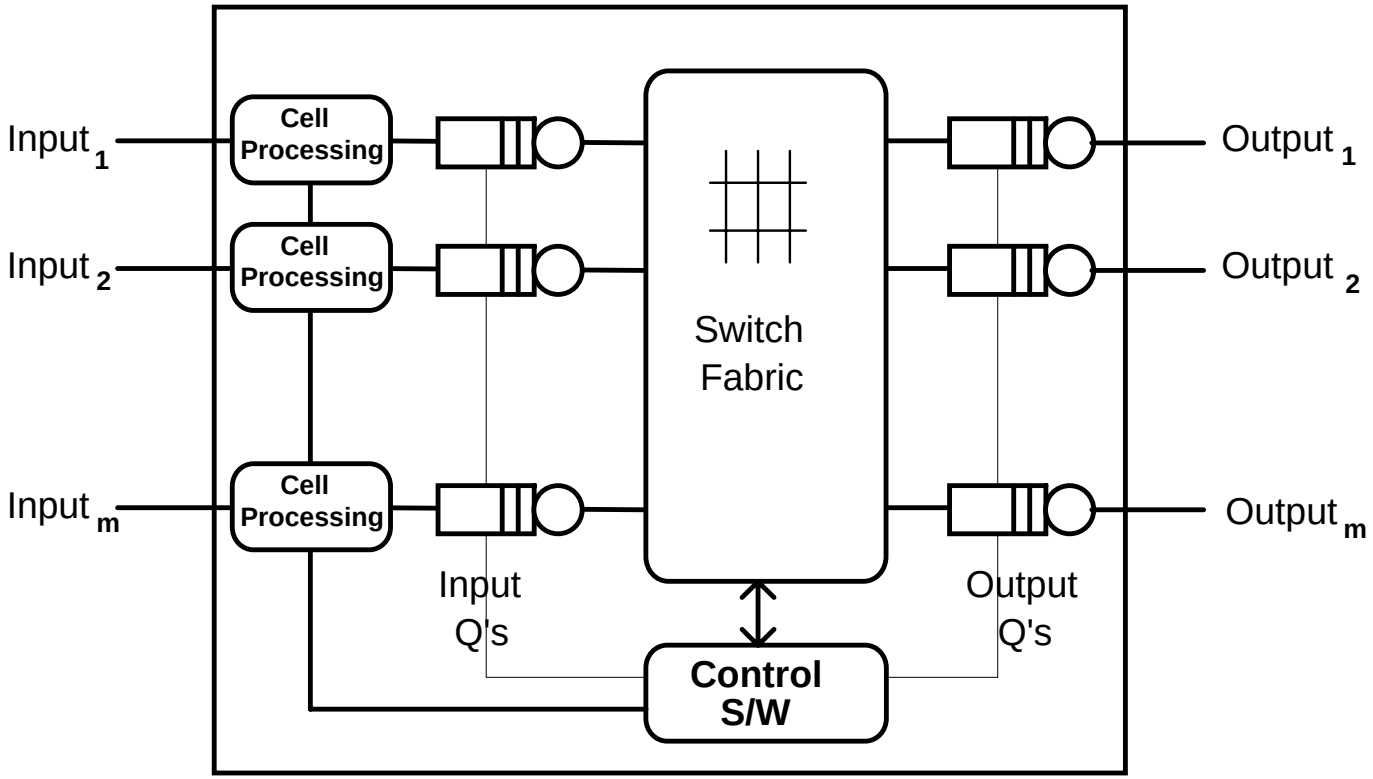
Example: AAL 3/4



- **ST: Segment Type (1st, Middle, Last)**
- **SEQ: 4-bit sequence number (detect lost cells)**
- **MID: Message ID (reassembly of multiple msgs)**
- **44 Byte user payload (~84% efficient)**
- **LEN: Length of data in this segment**
- **CRC: 10 bit segment CRC**

- **AAL 3/4 allows multiplexing, reliability, & error detection but is fairly complex to process and adds much overhead**
- **AAL 5 was introduced to support IP traffic**
  - **Fewer functions but much less overhead and complexity**

ATM cell switches



- **Design issues**
  - **Input vs. output queueing**
  - **Head of line blocking**
  - **Fabric speed**

ATM summary

- ATM is mostly used as a “core” network technology
- ATM Advantages
  - Ability to provide QoS
  - Ability to do traffic management
  - Fast cell switching using relatively short VC numbers
- ATM disadvantages
  - It not IP - most everything was design for TCP/IP
  - It’s not naturally an end-to-end protocol
    - Does not work well in heterogeneous environment
    - Was not design to inter-operate with other protocols
    - Not a good match for certain physical media (e.g., wireless)
  - Many of the benefits of ATM can be “borrowed” by IP
    - Cell switching core routers
    - Label switching mechanisms

Multi-Protocol Label Switching (MPLS)

“As more services with fixed throughput and delay requirements become more common, IP will need virtual circuits (although it will probably call them something else)”

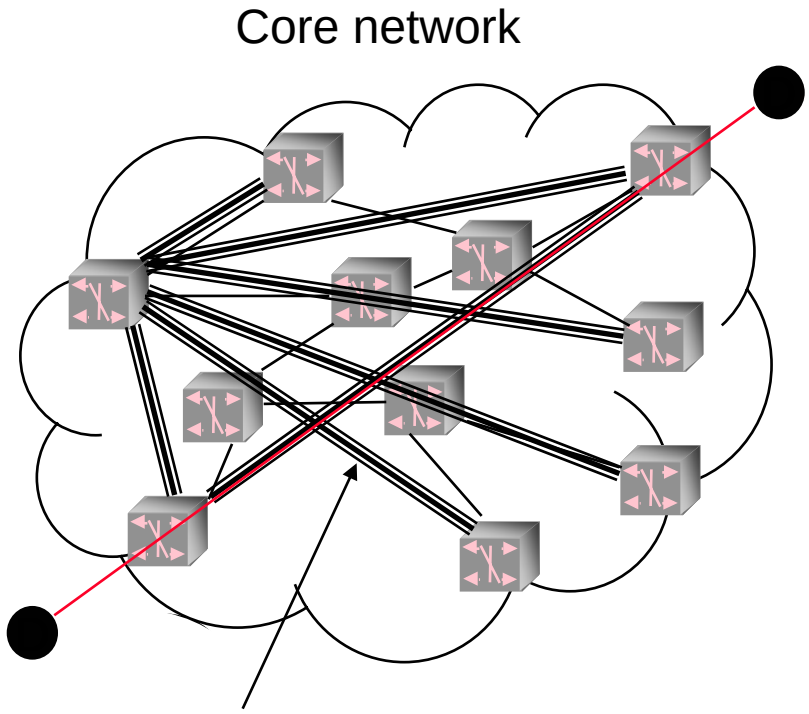
RG, April 28, 1994

Label Switching

- Router makers realize that in order to increase the speed and capacity they need to adopt a mechanism similar to ATM
  - Switch based on a simple tag not requiring complex routing table look-ups
  - Use virtual circuits to manage the traffic (QoS)
  - Use cell switching at the core of the router
- First attempt: IP-switching
  - Routers attempt to identify flows
    - Define a flow based on observing a number of packets between a given source and destination (e.g., 5 packets within a second)
  - Map IP source-destination pairs to ATM VC's
    - Distributed algorithm where each router makes its own decision
- Multi-protocol label switching (MPLS)
  - Also known as Tag switching
  - Does not depend on ATM
  - Add a tag to each packet to serve as a VC number
    - Tags can be assigned permanently to certain paths

Label switching can be used to create a virtual mesh with the core network

- Routers at the edge of the core network can be connected to each other using labels
- Packets arriving at an edge router can be tagged with the label to the destination edge router
  - "Tunneling"
  - Significantly simplifies routing in the core
  - Interior routers need not remember all IP prefixes of outside world
  - Allows for traffic engineering
    - Assign capacity to labels based on demand



References

- TCP/IP Illustrated (Vols. 1&2), Stevens
- Computer Networks, Peterson and Davie
- High performance communication networks, Walrand and Varaiya