TCP/IP and the Internet

Eytan Modiano
Massachusetts Institute of Technology

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

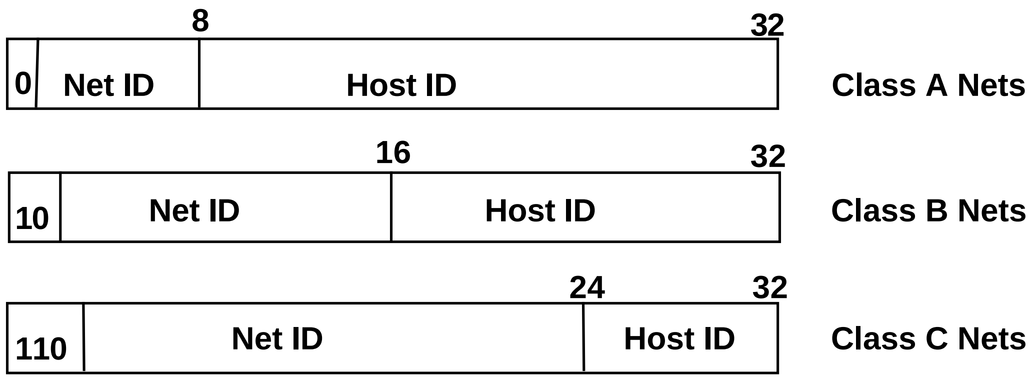
TCP - Transmission Control Protocol
UDP - User Datagram Protocol
IP - Internet Protocol

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

1	4	8	16	32
Ver	Header length	type of service	Total length (bytes)	
16 - bit identification			Flags	13 - bit fragment offset
TTL		Protocol	Header Checksum	
Source IP Address				
Destination IP Address				
Options (if any)				
Data				

Note that the minimum size header is 20 bytes; TCP also has 20 byte header

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

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**

	16 bit net ID	16 bit host ID	
Class B Address	140.252	Subnet ID	Host ID
Mask	111111 111 1111111	11111111	00000000

Classless inter-domain routing (CIDR)

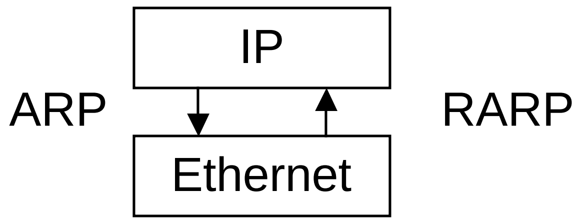
- Class A and B addresses allocate too many hosts to an organization while class C addresses don't allocate enough
  - This leads to inefficient assignment of address space
- Classless routing allows the allocation of addresses outside of class boundaries (within the class C pool of addresses)
  - Allocate 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

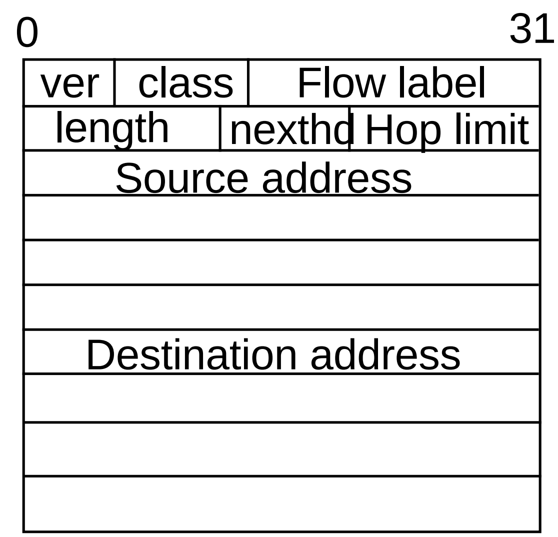
- The internet is divided into sub-networks, each under the control of a single authority known as an Autonomous System (AS)
- Routing algorithms are divided into two categories:
  - Interior protocols (within an AS)
  - Exterior protocols (between AS's)
- Interior Protocols use shortest path algorithms (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 (6 bytes)**
    - **1500 IP addresses per sq. ft. of earth!**
    - **Address partition similar to CIDR**
  - **Support for QoS via Flow Label field**
  - **Simplified header**

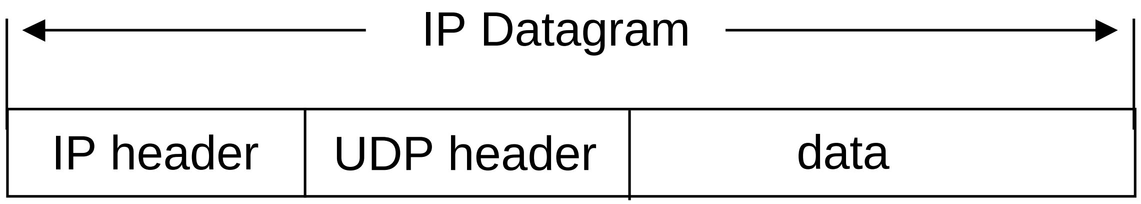
- **Most of the reasons for IPv6 have been taken care of in IPv4**
  - **Is IPv6 really needed?**
  - **Complex transition from V4 to V6**

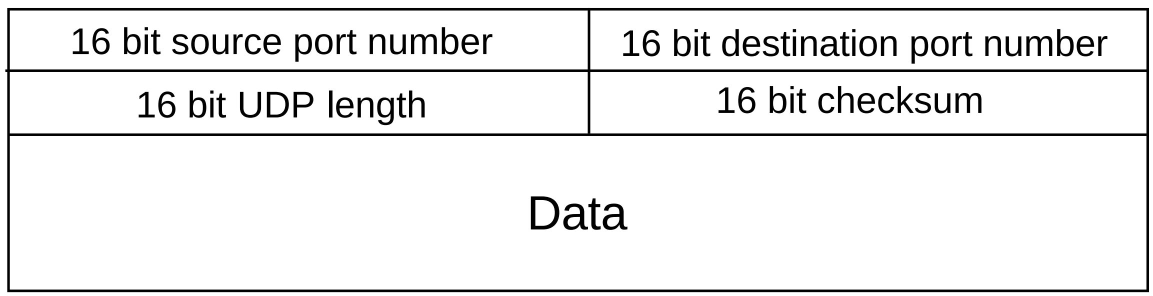


User Datagram Protocol (UDP)

- **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**

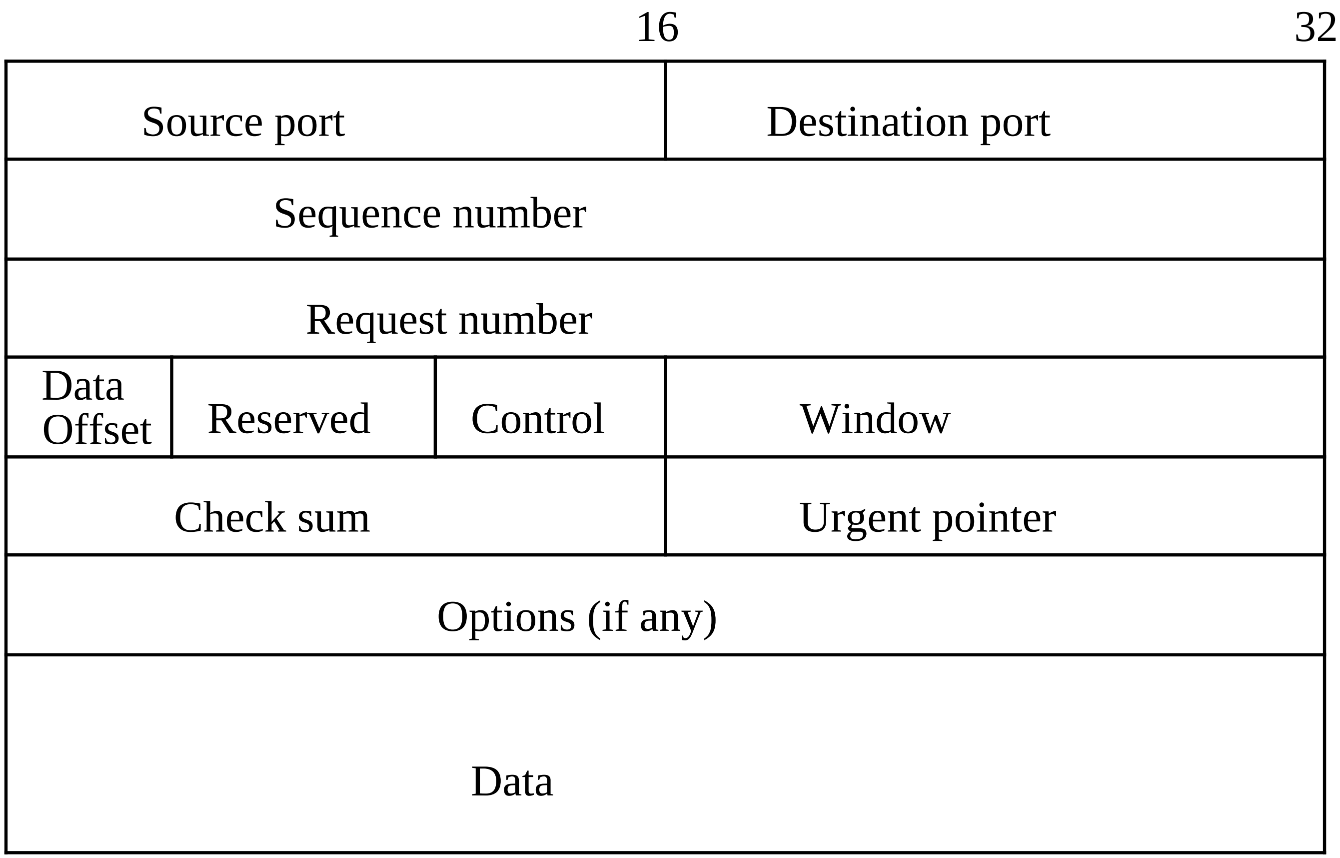
Transmission Control Protocol (TCP)

- **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 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

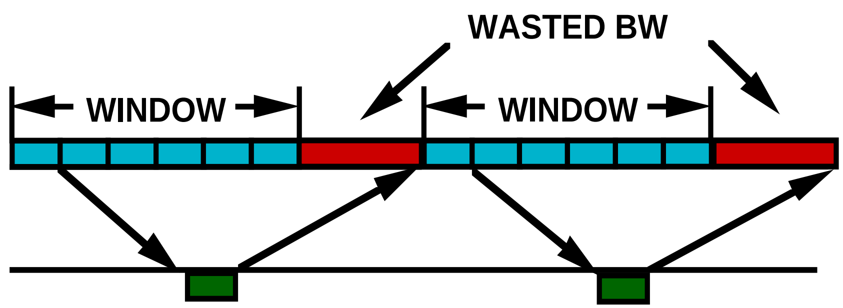
- Transmission rate when using window flow control is equal to one window of packets every round trip time

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

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

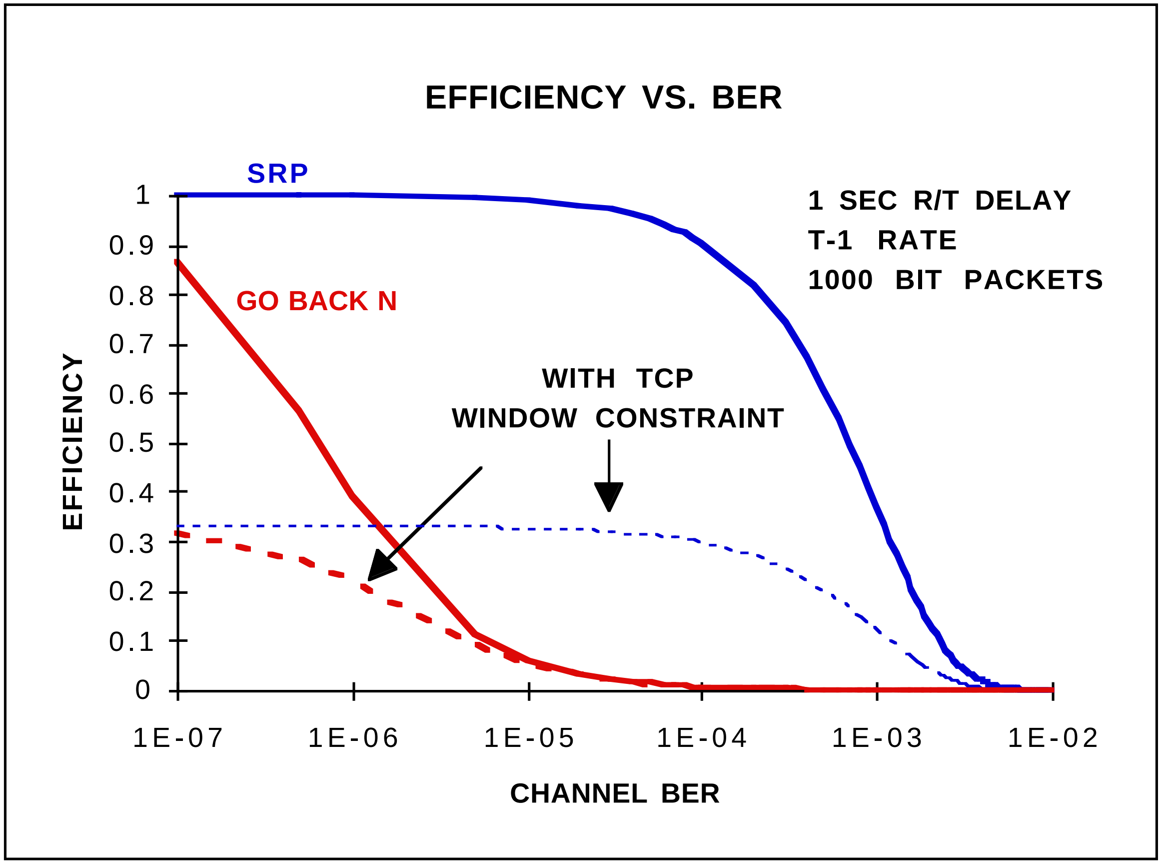
- Many variations: Tahoe, Reno, Vegas
- Basic idea: when congestion occurs decrease the window size
- There are two congestion indication mechanisms
  - Duplicate ACKs - could be due to temporary congestion
  - Timeout - more likely due to significant congstion

- TCP Reno - most common implementation

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

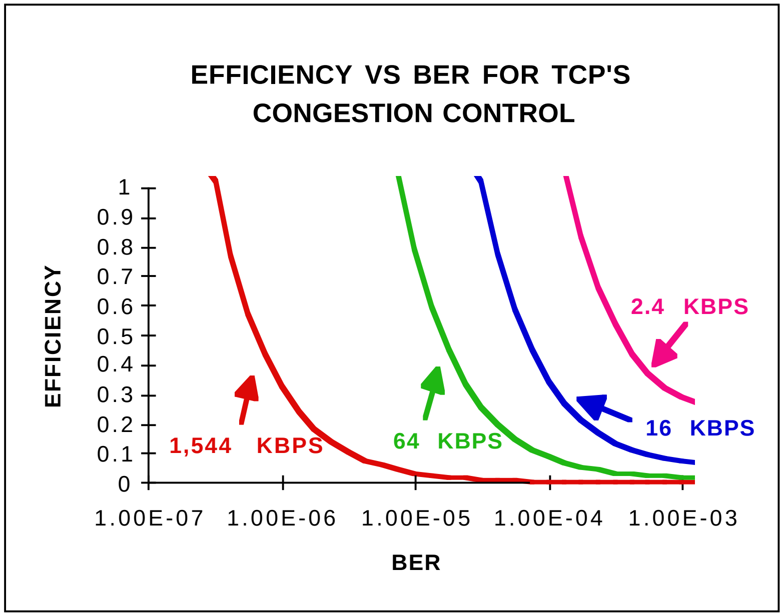
  - If duplicate ACKs occur CW = CW/2 stay in congestion avoidance phase

TCP Error Control over a GEO Satellite link



- **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