

Digital Communications over Packet-Switched Networks

(ECE 446 – Lecture #8)

Sergio D. Servetto

School of Electrical and Computer Engineering – Cornell University

<http://cn.ece.cornell.edu/>

Outline

- Last week: TA lectures.
- Today: recap on layer 3 in IP networks.

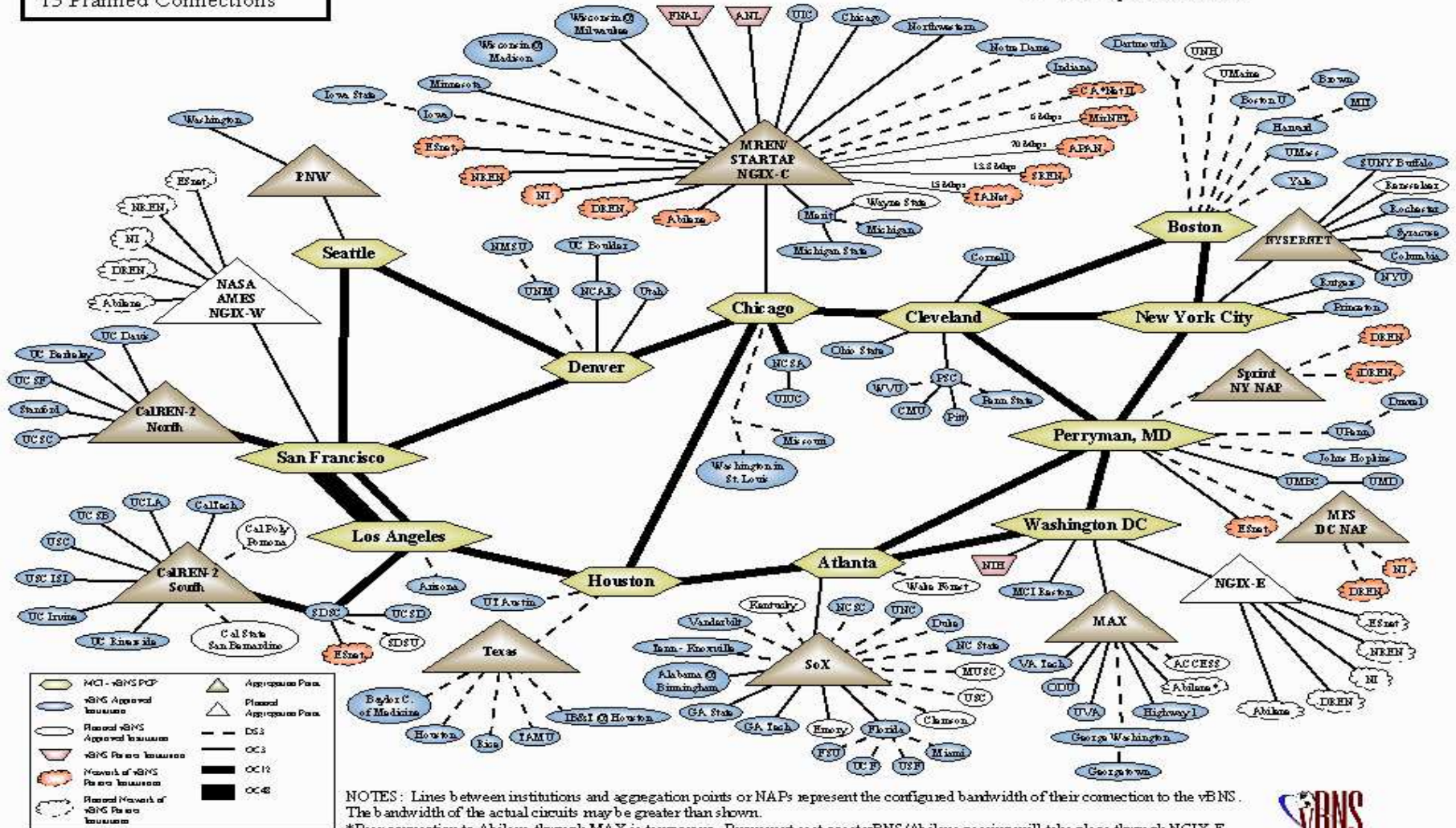
The Internet

- Origins: the ARPANET network sponsored by DoD in the 1960's:
 - Store-and-forward network, packet based.
 - DoD liked it because of its fault-tolerance properties.
 - Aimed at interconnecting heterogeneous networks.
- Network of networks:
 - Backbone of pt2pt links connecting Network Access Points (NAPs).
 - Points of Presence (PoPs): routers attached to NAPs.
 - Subscribers connect to PoPs (phone, DSL, cable, leased line).
 - Peering agreement among backbones maintained by \neq providers.
- vBNS: a backbone that was around when I wrote my PhD thesis:

91 Operational Connections
13 Planned Connections

vBNS Logical Network Map

Last Updated 04/26/99



© 1999 MCIWORLD.COM



- Internet protocols:

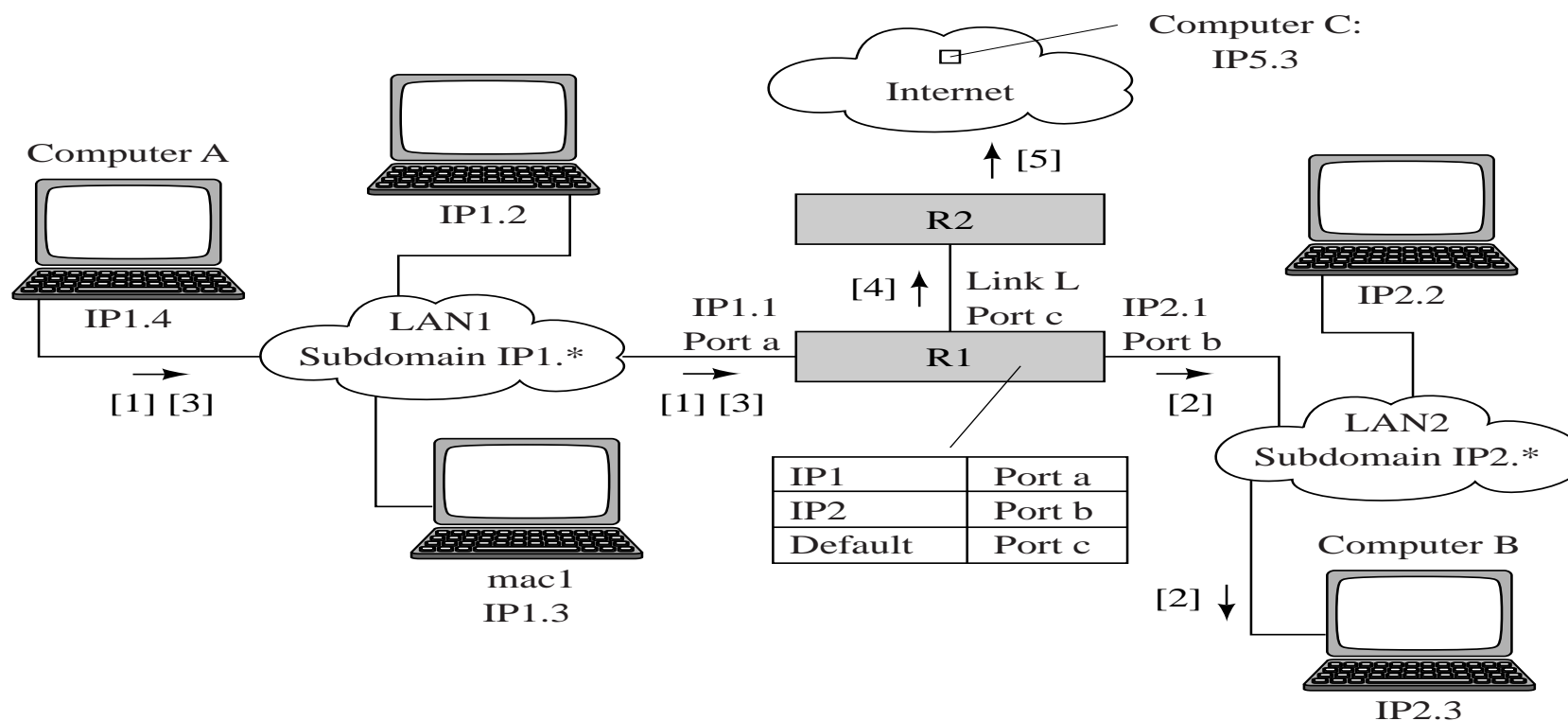
H T T P	T e l n e t	FTP	S M T P	S N M P	RCP	R S H	r l o g i n	T F T P	7
TCP								UDP	6
IP, ICMP, ARP, RARP									5
X.25, Ethernet, token ring, FDDI, T1, ...									4
									3
									1, 2

- Basic architectural principles for the Internet:

- Best-effort packet delivery service.
- Put IP on every network technology, build all applications on top of IP.

Internet Protocols – IPv4

- Overview of IP routing and naming:



- Send [data1] from A to B.
- Use DNS to translate B to its IP address.
- Create an IP packet: [IP1.4 – IP2.3|data1].
- Send packet to R1 (IP2.3 doesn't fit IP1.*, route to default) over LAN1.
- Create an Ethernet frame:
[mac(IP1.4) – mac(IP1.1)|IP1.4 – IP2.3|data1|CRC].
- R1 receives Ethernet frame, recovers [IP1.4 – IP2.3|data1], checks routing table, determines that addresses IP2.* are on port b.
- R1 generates new Ethernet frame, sends packet out on port b.
- B receives frame, removes Ethernet and IP headers, extracts [data1].

(If more hops needed, similar steps are followed.)

- Key elements:

- Addresses (address format, DNS).
- Determination of routes and routing mechanism.

Internet Protocols – IPv4 Addresses

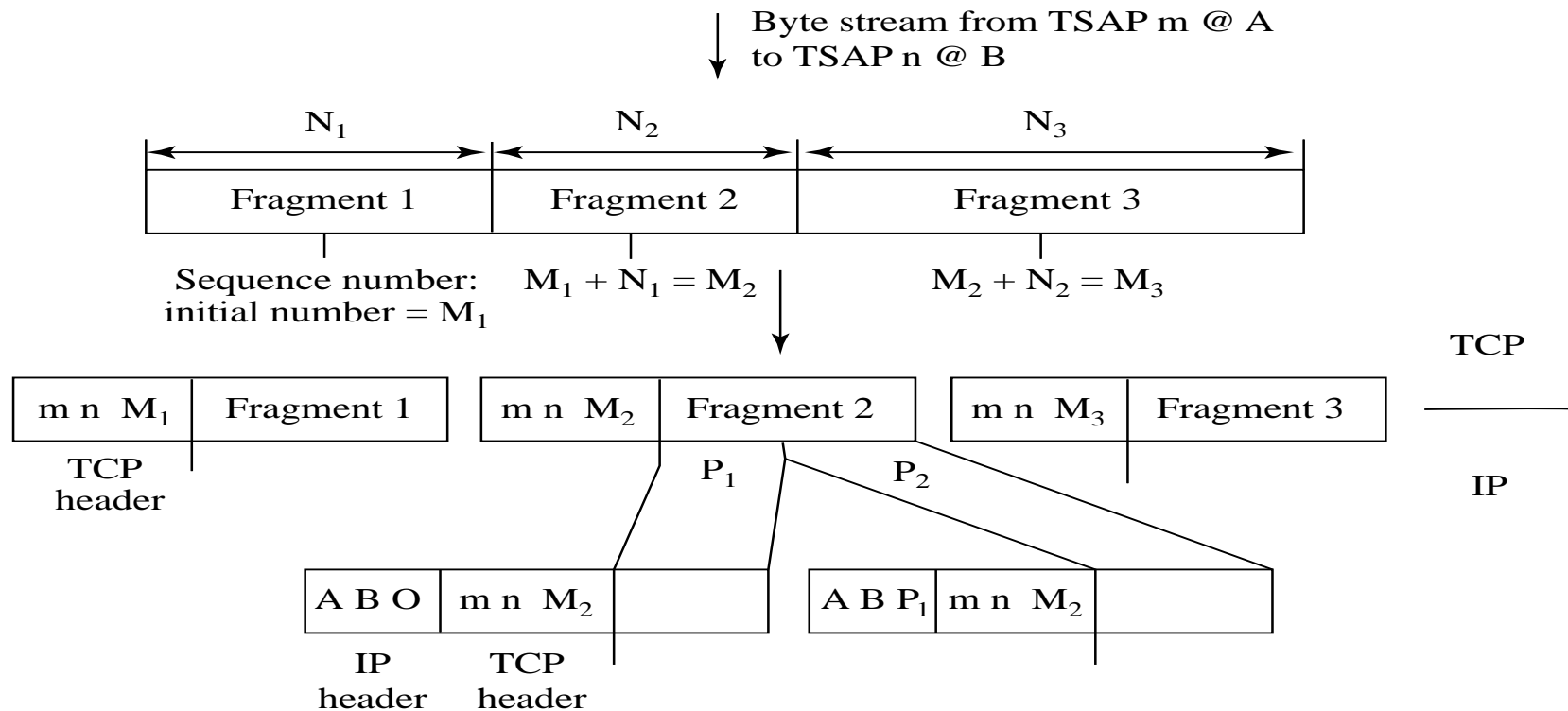
Four types of addresses: MAC, IP, domain-based, and URLs.

- MAC (layer 2).
 - Very different across hardware platforms (Ethernet, Token Ring, ...).
 - Need a standard way of identifying hosts / network interfaces.
- IP (layer 3) – e.g., 128.84.81.131 / `ushuaia.ece.cornell.edu`.
 - Network number: first two digits (e.g., at Cornell: 128.84, 128.253).
 - Host number: last two digits (e.g.: `ushuaia` is 81.131 within 128.84).
 - DHCP: dynamic allocation of IP addresses.

- Domain-based addresses – e.g., `servetto@ece.cornell.edu`.
 - DNS: the domain name server.
- URLs – e.g., `http://people.ece.cornell.edu/servetto/`.

Internet Protocols – Fragmentation/Reassembly in IPv4

Streams of bytes have to be put into IP packets, and IP packets may need to be further broken into smaller packets by the networks.



Internet Protocols – IPv4 Routing

To route packets, routers maintain tables of next hops:

- Address Resolution Protocol (ARP): addresses on the same network.
 - Search for the MAC address of a computer given its name.
 - If not found, broadcast name and request a reply (MAC address).
 - Table entries have a time-to-live (TTL), purged upon expiration.
- Routing tables *within* Autonomous System (AS):
 - Shortest-path computations – will study algorithms in part II.
 - Internet Control Message Protocol (ICMP): monitoring services
 - * `ping`: ask a host to echo a packet.
 - * `traceroute`: route to host and delay on each hop.
 - * Unreachable destinations, ttl exceeded, ...

Internet Protocols – IPv4 Routing

Border gateway protocol:

- AS: network under a single administration.
- ASs interconnected by *border gateways* (routers).
- Each AS contains one designated router, the *BGP speaker*.
 - Use TCP to exchange routing tables (routes among speakers).
 - Exchange keep-alive messages, to identify paths that change.
- Main role: choose routes between ASs.
 - Can refuse to carry traffic generated by specific ASs.
 - Can choose better routes for own traffic.
 - Can ...

Internet Protocols – IP Multicast

Multicast: allows for the transmission of an IP packet to a group of hosts (instead of sending to a single destination):

- Multicast group identified by a common address:
 - Class D addresses: 224.0.0.0 to 239.255.255.255.
 - Tx writes to a socket with destination IP address = multicast address.
 - Rx's *join/leave* a group, by specifying an option on a socket.
 - After join and before leave, rx's can receive from the given address.
- Routers must be able to provide some functions in support of multicast:
 - Determination of local hosts part of a multicast group (routers do *not* maintain a list of hosts, but need to know which groups are active).

- Construct a multicast tree (shortest paths) connecting all routers with active groups, rooted at the transmitter.
 - Update the tree dynamically, as nodes join/leave groups.
 - Provide packet forwarding/replication along the tree.
 - Communication among routers over (layer 3) *IP tunnels*.
- MBone (IP Multicast Backbone): an overlay on the Internet for video/audio multicast.
 - Reliable multicast: packets must be delivered to all nodes in a group.
 - Applications: distribution of software updates, documents, ...
 - Dealing with acknowledgements is not trivial.

So, where is IP multicast today?

Internet Protocols – IP Multicast

Nowhere to be seen...

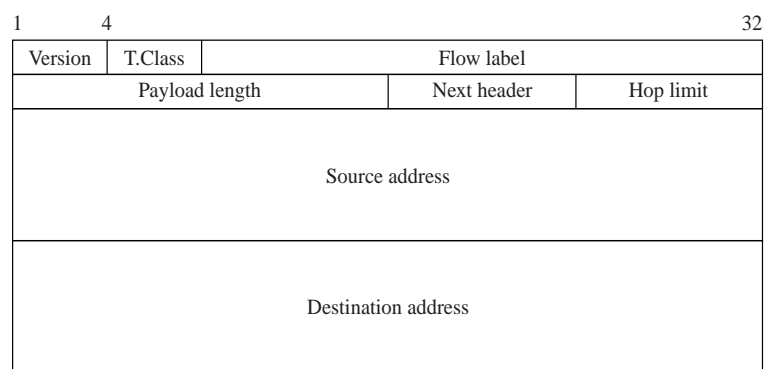
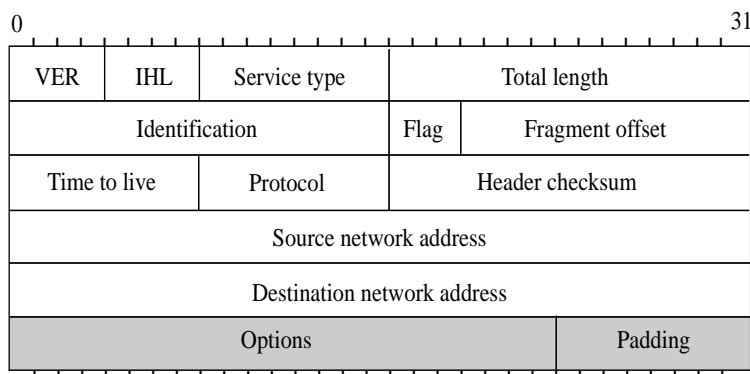
- Scalability problems:
 - The server implosion problem.
 - Routers have to do “too much work” as the number of rx’s grow.
- Reliability problems:
 - QoS on the MBone just not good enough for real applications.
- No major applications use multicast today:
 - No such thing as CNN/IP, or NYTimes/IP, or ...

A paper I wrote some years back on the subject: [\[Download\]](#).

Internet Protocols – IPv6

Shortcomings of IPv4:

- Limited number of addresses.
- Complex header.
- Difficult to extend to incorporate new services.
- Poor support for security and privacy.



Internet Protocols – IPv6

Header structure:

- Version: 110 (6, in binary).
- T.Class: importance/urgency of the packet.
- Flow Label: characteristics of traffic (real time, bulk, ...).
- Payload Length: # of bytes after header.
- Next Header: which extension follows, and if part of TCP/UDP flows.
- Hop limit: max number of hops that a packet can go over.

Internet Protocols – IPv6

Header extensions:

- Hop-by-hop options: information for each router (not specified yet).
- Routing: up to 24 IPv6 addresses that this packet must visit.
- Fragmentation: which fragment of a larger packet this packet contains.
- Authentication: checksum to enable destination to authenticate sender.
- Encapsulating Security Payload: key number and encrypted payload.
- Destination Options: not specified yet.

Recall what the main goals were: more addresses and extensibility.

Internet Protocols – TCP and UDP

UDP (User Datagram Protocol):

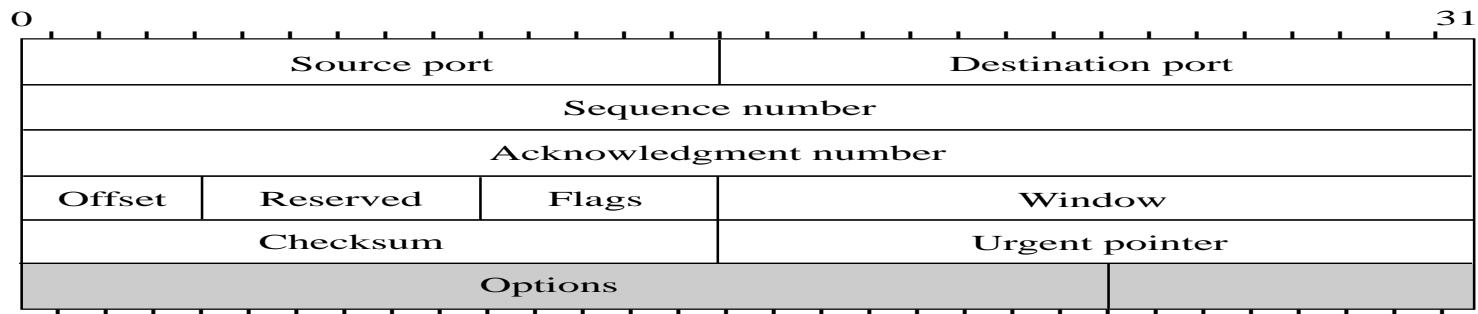
- Connectionless service (no state maintained at tx/rx for connection).
- Internet version of sending a letter by regular mail.
- No guaranteed delivery, no in order delivery, no duplicate prevention.
- Only multiplexing and error detection supported.

Useful for applications where reliability is not critical or with tight real-time delay constraints (retransmissions unaffordable).

Internet Protocols – TCP and UDP

TCP (Transmission Control Protocol):

- Connection-oriented, reliable, byte-stream service:
 - Tx sends a stream of bytes to rx, in segments put into IP packets.
 - Segments are delivered in order, reliably, without repetitions.
 - Go-back-N is used to guarantee reliability.
- Headers:



- Source, destination ports: multiplex connections, identify services.
 - Sequence, ack numbers: for go-back-N, and to sort out-of-order packets (acks are cumulative).
 - Offset: number of 32-bit words in a packet (where payload starts).
 - Some flags (URG, ACK, SYN, ...).
- Life of a TCP connection from A to B:
 1. Node A sends a TCP packet to open a connection:
 - Flag = SYN, set IP addresses and ports for tx/rx, set initial sn.
 2. B responds to A with initial sn.
 3. A sends first data packet, connection open.
 4. A sends packets, B acks correct packets with sn of next byte.
 5. When A or B want to close connection, send a FIN flag.

TCP also implements an *adaptive* window mechanism – later.

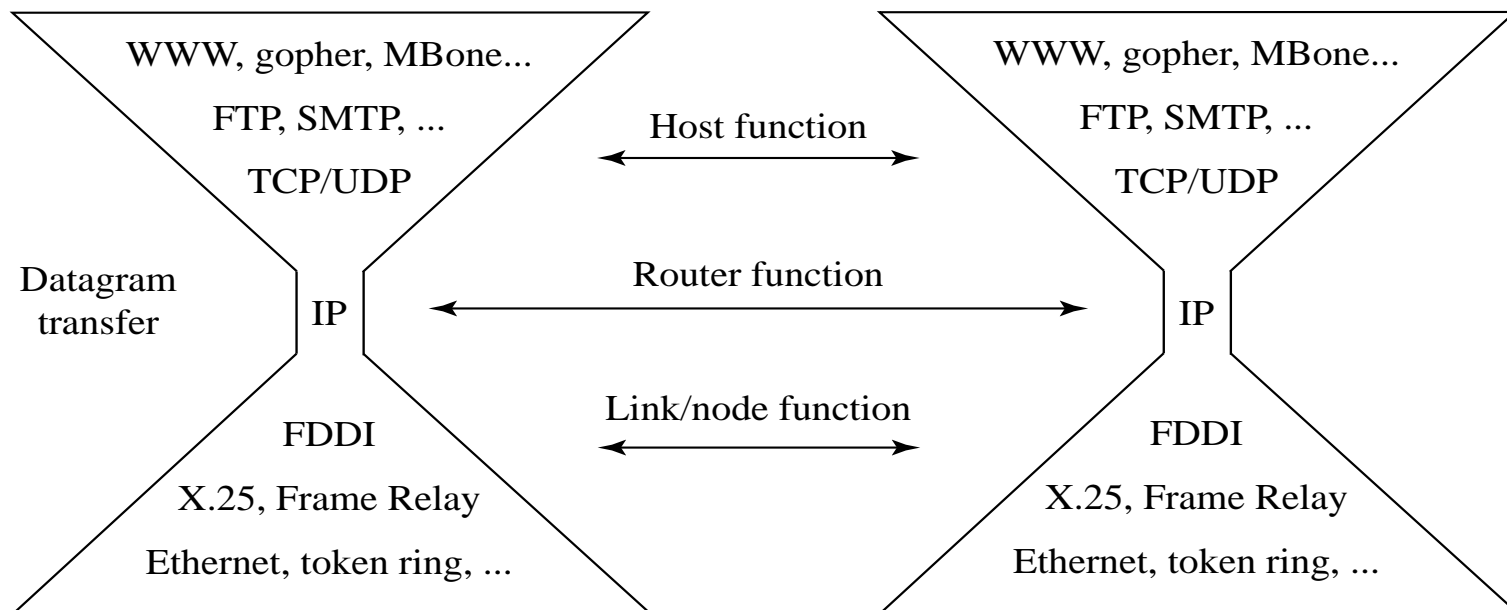
Examples of Applications using TCP/UDP

OS provides system calls (sockets) that applications can use:

- FTP: enable file transfer between computers.
 - Open TCP socket, copy a (compressed) file to the socket, close.
 - Need a client to write file, a server capable of receiving file.
 - Need mechanisms for client/server to exchange parameters.
- SMTP: email, runs on top of TCP.
- SSH: encrypted remote login, runs on top of TCP.
- HTTP: exchange of web pages, runs on top of TCP.
- Audio/Video streaming: runs on top of either TCP or UDP.

The Internet Architecture

- Applications built on top of the IP abstraction.
- The network provides best-effort service to deliver IP packets.
- IP implemented on all sorts of different platforms.



What Next

- Next time: circuit switching.
- Complete reading chapter 4, start browsing chapter 5.

Credits: most figures used in these slides are from our textbook.