

Internet Protocols: IP and TCP

George Varghese

May 22, 2002

1.0 Internet Protocols, Old and New

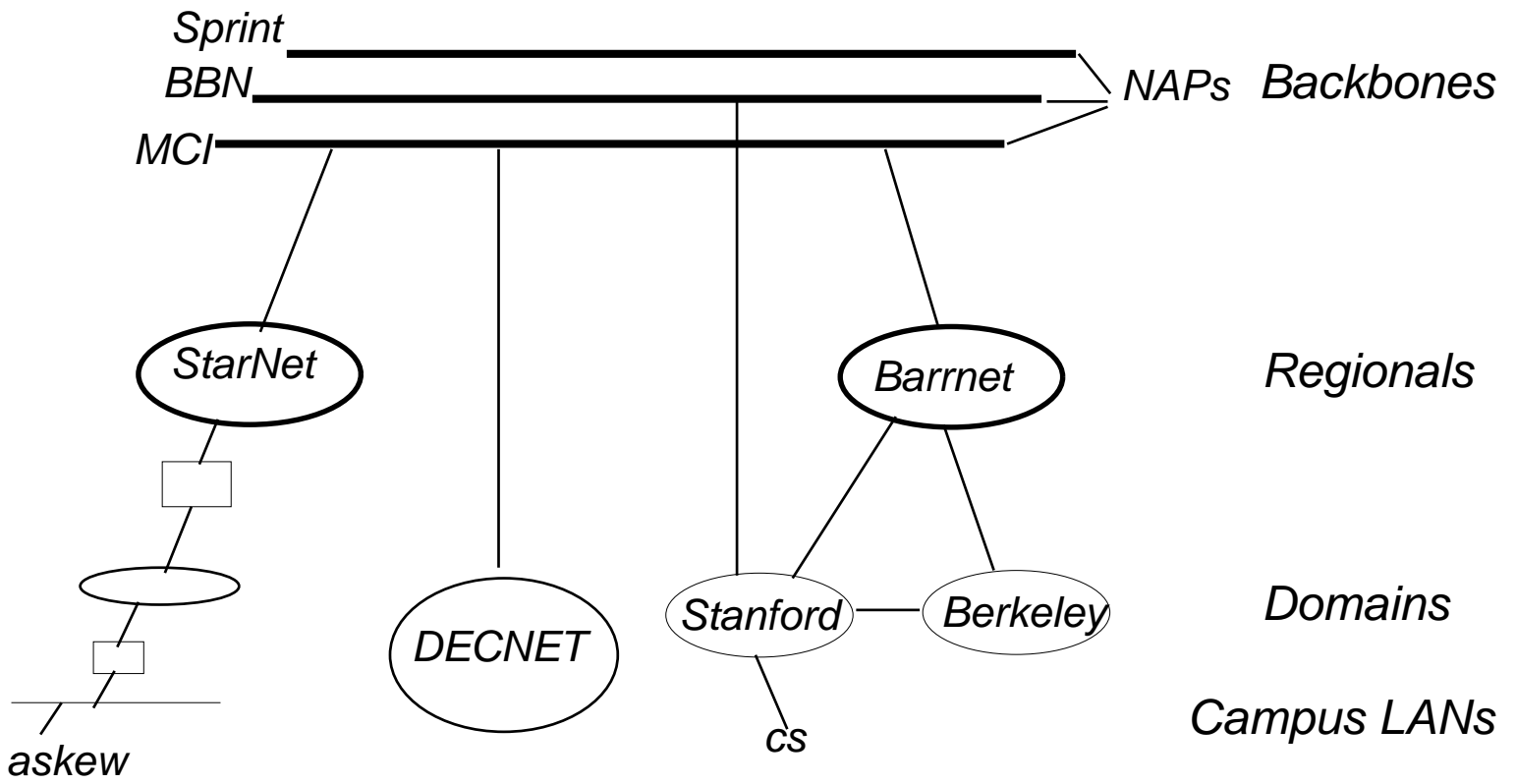
- **1.1** Internet Basics
- **1.2** Intradomain Protocols
- **1.3** Interdomain Protocols and Policy Routing
- *Segue:* Internet Future Requirements and Goals
- **1.4** Multicast Routing
- **1.5** Mobility Support
- **1.7** The Next Generation Internet
- *Segue:* The Domain Name Service
- **1.9** Internet Transport Protocol (TCP)
- **1.10** New Transport Protocols for multicast, video etc,

1.1 Internet Basics

Outline

- Basic Internetworking in IP: network numbers, datagram service, and fragmentation and reassembly
- IP evolution from ARPANET to Multiple Service Providers
- IP addressing from Classful to Classless Addressing using Subnetting and CIDR to IPv6

Topology



Basic Internetworking

- Internetworking was a specific goal of IP (unlike say DECNET, SNA etc.). Starts with an implicit hierarchy of physical networks (with network specific routing that IP does not care about) and an internetwork of physical networks. Each physical network has a network number.
- IP's job is to route to the right physical network based on the network number. Offers a datagram service with possible fragmentation and reassembly to deal with different MTU sizes.
- IP always configured with a companion protocol called ICMP for error messages (header checksum failed, TTL reached, redirect)

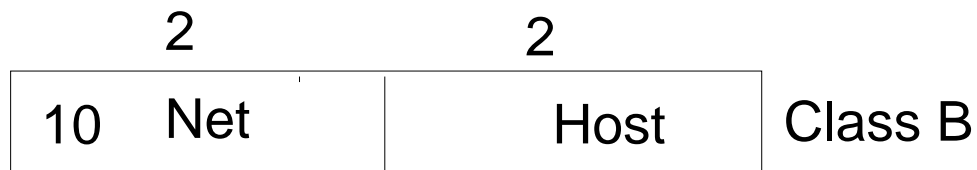
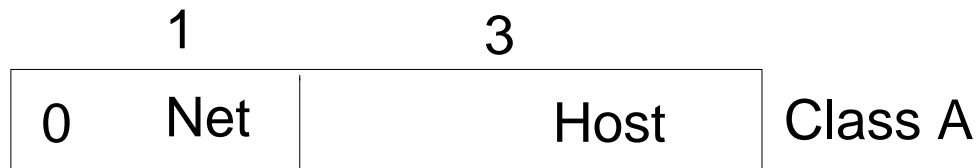
IP Evolution

- Started with the ARPANET linking sites in the 1970's
- In 1983, ARPANET splits up into MILNET and ARPANET. In 1984, NSF establishes NSFNET to be backbone. Campuses attached to backbone via regional networks (NYSERNET, BARTNET etc.). Strict hierarchy breaks down because of direct connections between providers.
- By late 1980's, Internet becomes worldwide. Still mostly hierarchical. Research network to production quality. Multiple autonomous service providers that need to control resource sharing.

Names and addresses

- When you send to a domain name like *cs.berkeley.edu*, a resolver in your host translates the name to a 32 bit IP address *128.32.35.123*. All messages carry IP destination and source addresses.
- Translation done using DNS; will study later.

Original IP addresses: Classful Addressing



- Small number of large networks (class A), moderate number of campus networks (class B), and large numbers of LANs (class C)
- Hierarchical address with a moveable partition for flexibility. Routers keep state only for network numbers.

Old IP Forwarding

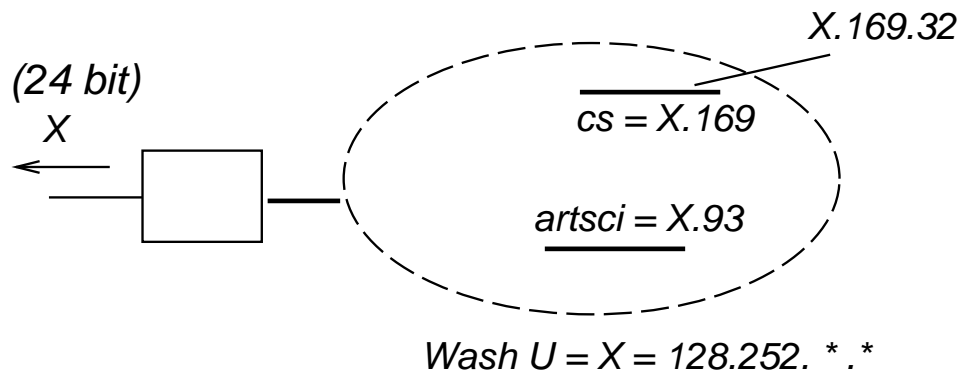
- Extract Network number of destination address in packet.
- If (Network Num of Dest = Network Number of this routers local interfaces) deliver packet on that interface. Map to local address using ARP or some such protocol.
- Else if (Network Number is in Router Forwarding Table) then deliver packet to NextHop Router
- Else deliver packet to default router.

Moving to a Global Internet

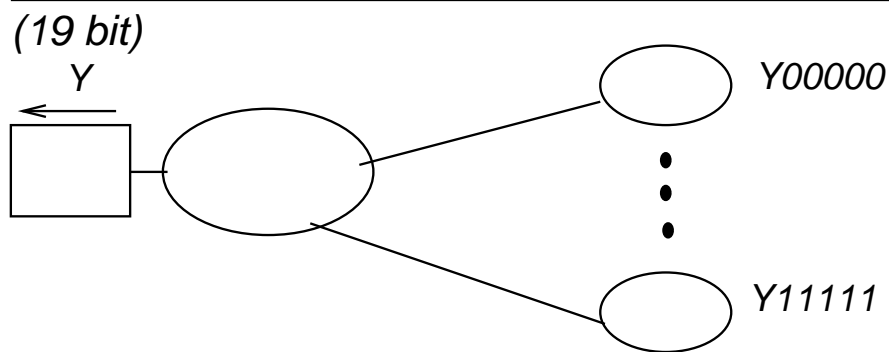
One level of hierarchy good but we have two scaling problems:

- *Inefficient address usage*: Any network with more than 255 addresses needs a class B address. One address network wastes 254 addresses, 256 address network wastes 64,000 addresses! Class B addresses are running out.
- *Routing table growth* Assigning lots of Class C network numbers requires every backbone router to know about these net numbers. Increases forwarding table size, router control traffic, search times.

Subnetting and Supernetting



Subnetting a Class B address X



Supernetting Class C addresses Y0–Y31

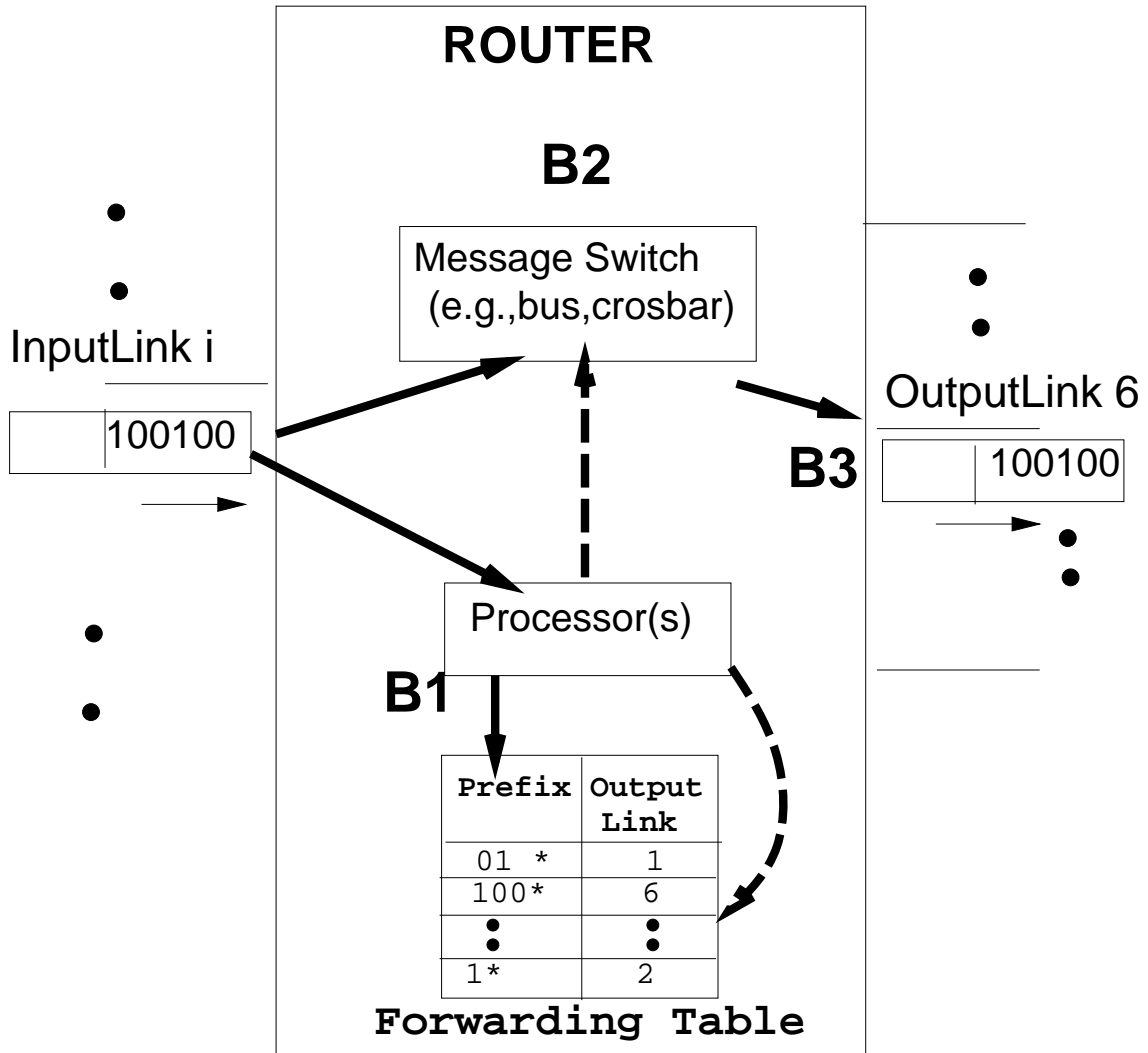
- Supernetting can be done recursively (CIDR). Leads to backbone routers having to deal with 45,000 prefixes of lengths 8-32. Temporary measures: need IPv6 and 128 bit addresses.

New IP Forwarding

- Find best matching prefix P of destination address in message.
- If P is nil, forward on default route. Else if next hop associated with P is a local interface, map to a local address (using say ARP) and deliver packet on that interface.
- Else send packet to NextHop Router associated with P .

Backbone routers in default-free zone keep 45,000 or so prefixes; enterprise routers tend to have 1000 or less prefixes because of heavy use of default routes.

Router Model



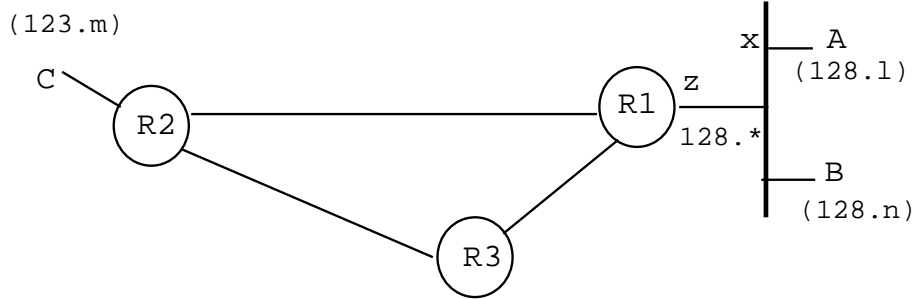
1.2 Basic Intra-domain Routing

Outline

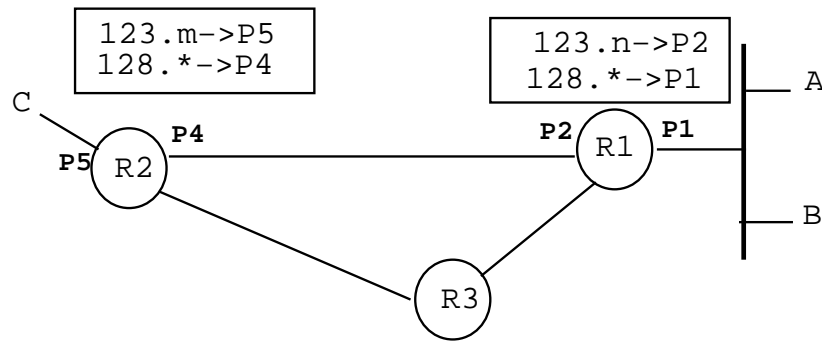
- Dealing with the last hop (e.g., ARP)
- Distance vector routing (e.g., RIP)
- Link state routing (e.g., OSPF)

From Routing to Forwarding

IP ROUTING



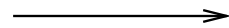
STEP 1: FIND NEIGHBORS



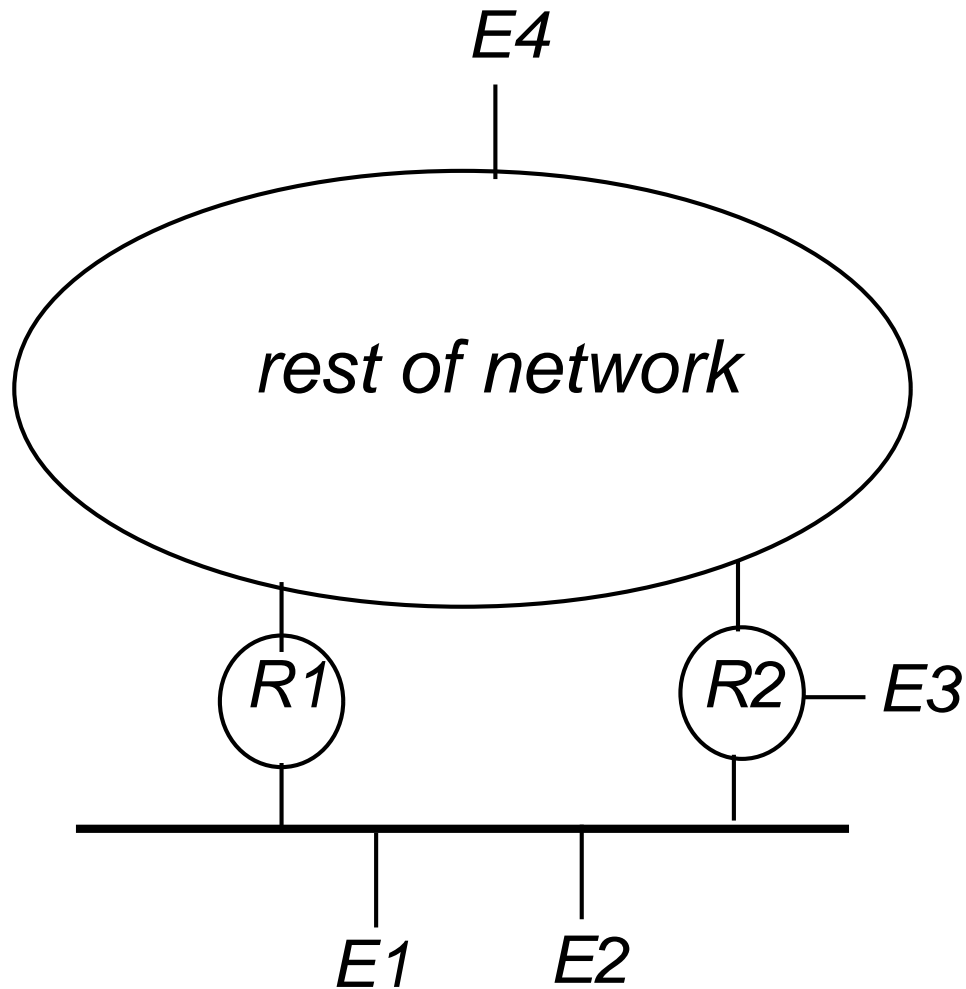
STEP 2: COMPUTE ROUTES



STEP 3: FORWARD

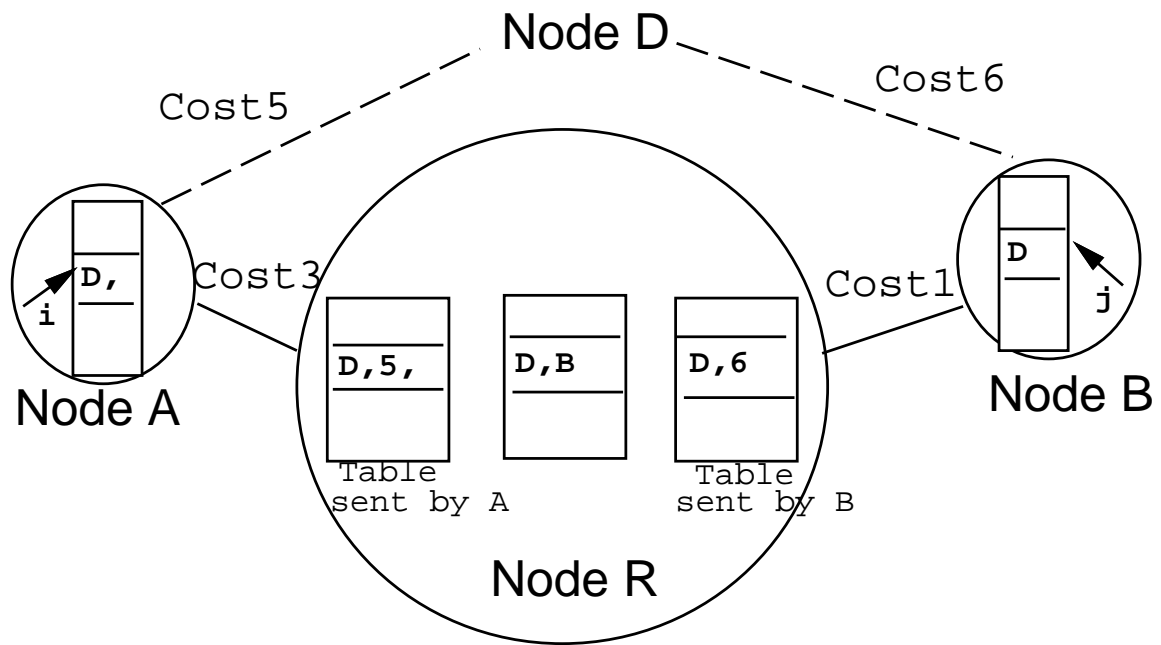


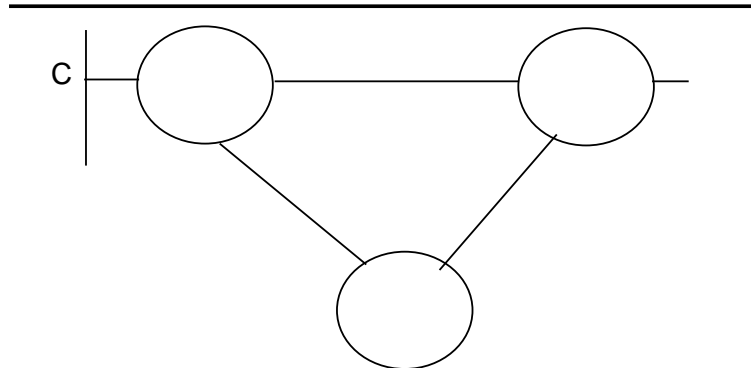
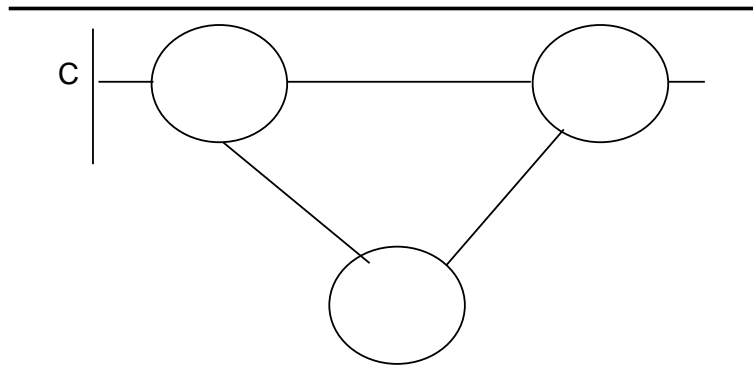
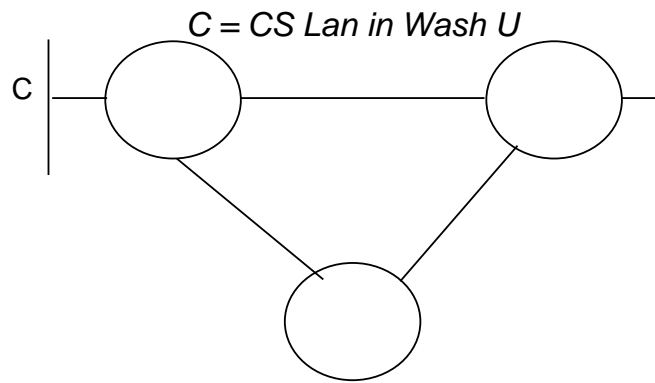
4 Problems for LAN Endnodes



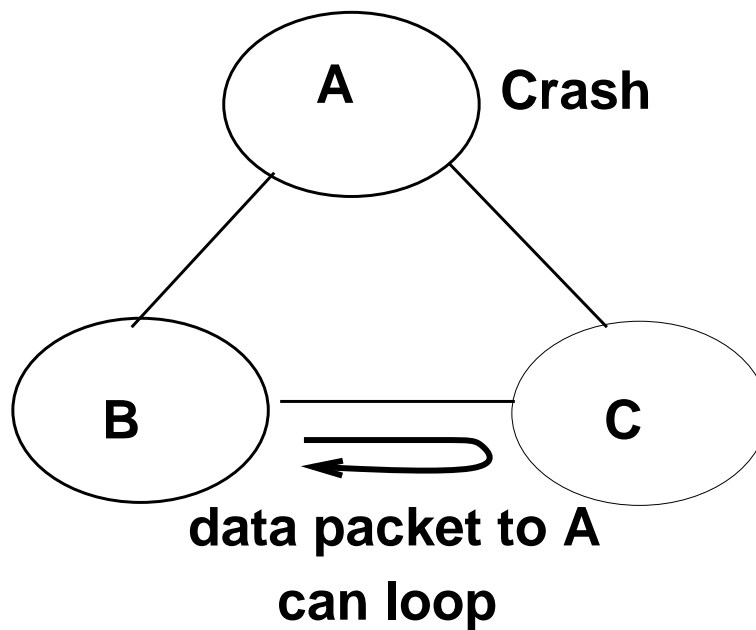
- Routers need Data Link addresses of endnodes
- Endnodes need Data Link address of at least one router
- *E1* and *E2* should be able to communicate without a router.
- *E1* to *E3* traffic should go through *R2* eventually.

Using Distance Vector in a domain





Data Packet Looping

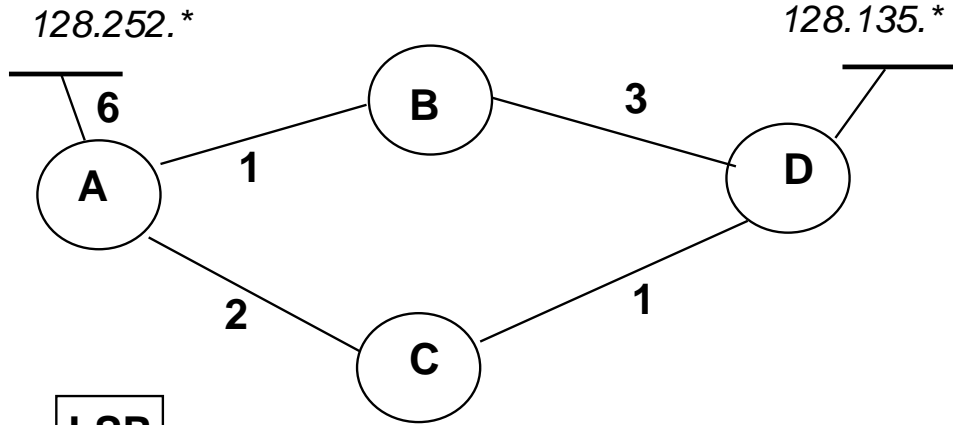


- After A crashes, B and C keep thinking the best way to get to A is through each other. Loop detected eventually by hop count.

Link State: the basic idea

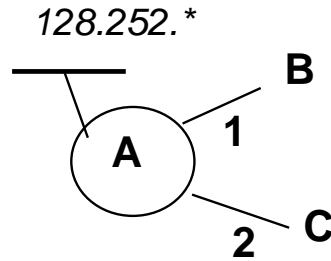
- Each router R knows the default (or manager settable) cost of its outgoing links. Place UP router neighbors directly reachable subnets, and link costs in a Link State Packet (LSP) for R .
- R broadcasts its LSP to *all* other nodes using a primitive flooding mechanism.
- R uses Dijkstra's algorithm to compute the next hop router on the shortest path from R to every other subnet D .

LSP Generation

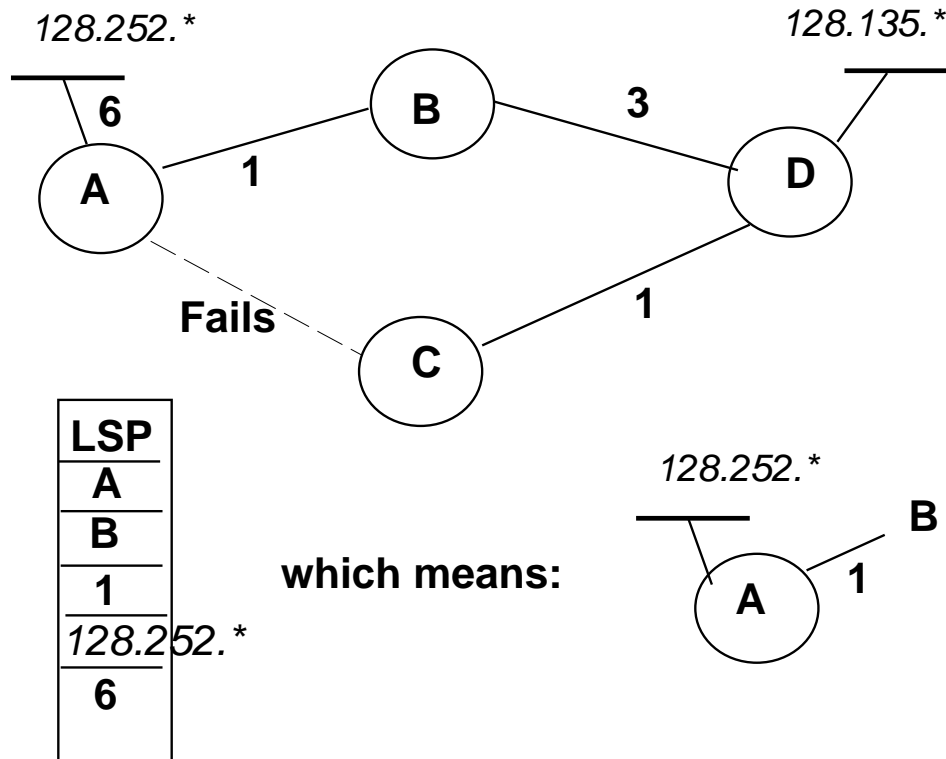


LSP
A
B
1
C
2
128.252.*
6

which means:

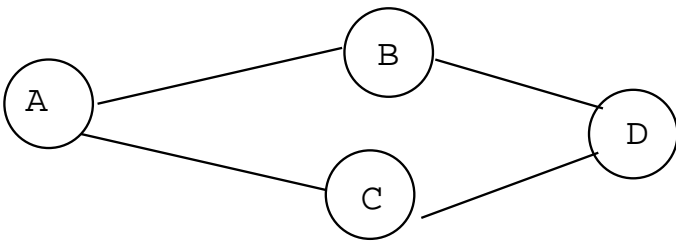
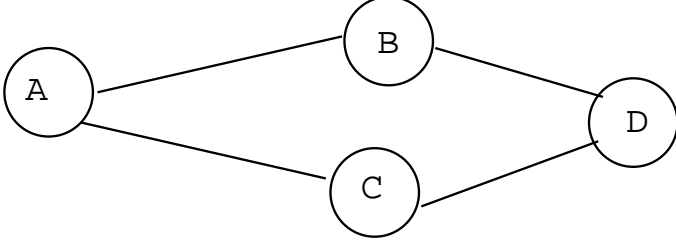
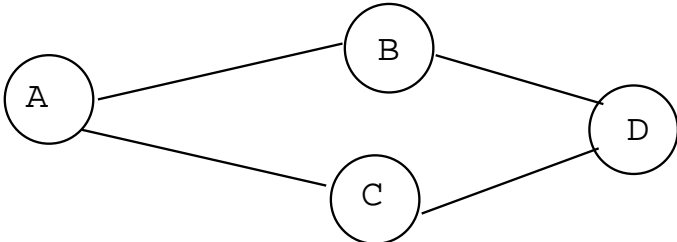
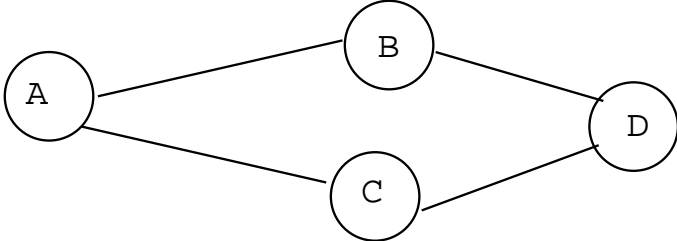


LSP Generation on Failure



- If link AC fails, A and C will eventually timeout link.
- Only A and C recompute their LSP values and broadcast their LSPs again to all other nodes. Other nodes do not recompute or rebroadcast their LSPs.

INTELLIGENT FLOODING



1.3 Policy Routing between Domains

Why Interdomain Routing

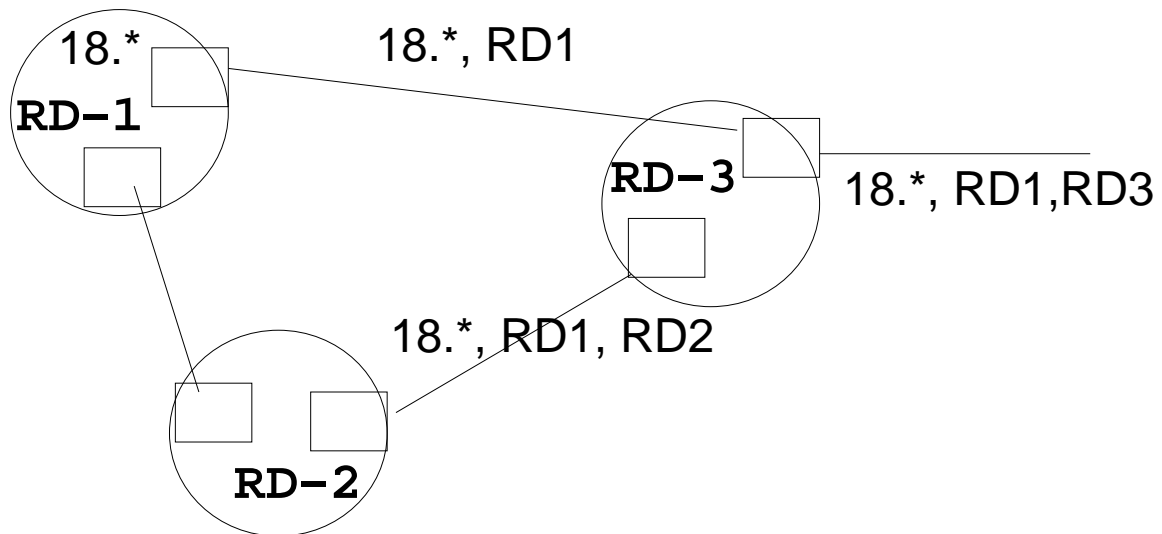
Why not one happy melting pot of a network:

- Multiple providers (see IP evolution) implies need for independence and independent policies.
- Different metrics, trust patterns, different charging policies (hot potato, cold potato), different administrative and legal requirements (e.g., ARPANET only for government business, Canadian traffic stays within Canada).
- Not very well developed. Basic conflict between abstraction and hierarchies (for scaling) and ability to specify arbitrary policies.

Possible Policies

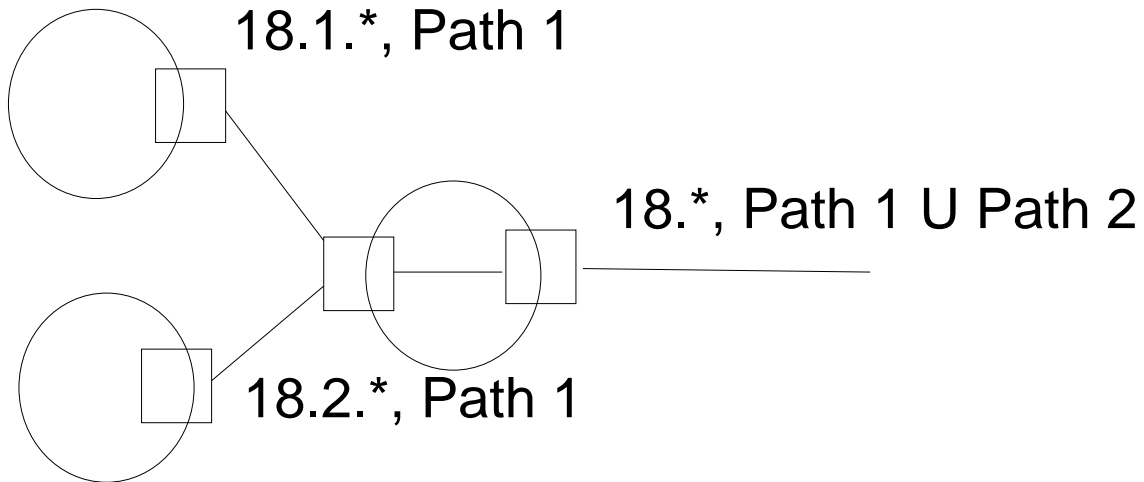
- Never use Routing Domain X for any destination.
- Never use domains X and Y.
- Don't use X to get to a destination in domain Y.
- Use X only as a last resort.
- Minimize number of domains in path.
- Government messages can traverse the ARPANET but not others.
- Use the set of domains whose combined cost is least.

The Border Gateway Protocol(BGP)



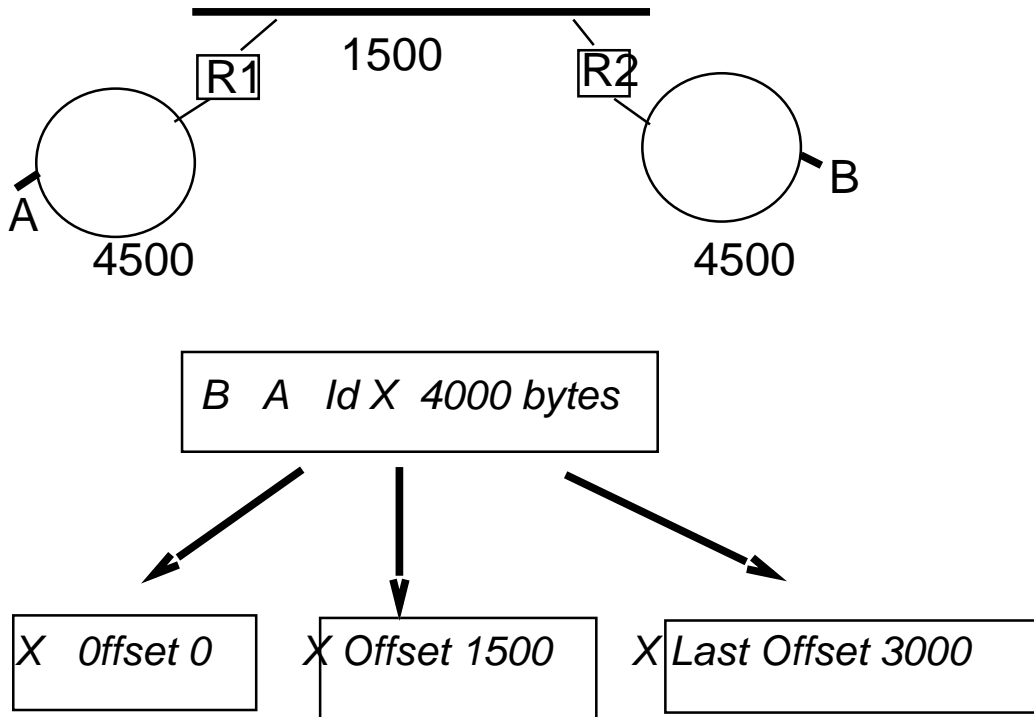
- Path vector scheme allows both policy flexibility and loop free routing.
- Per source domain policy can cause information to be lost.

Some BGP features



- Can aggregate addresses to improve routing table scalability by taking unions of paths. Uses TCP between BGP neighbors for incremental updates.
- Conclusion: Some loss of flexibility. But does it even *work*?

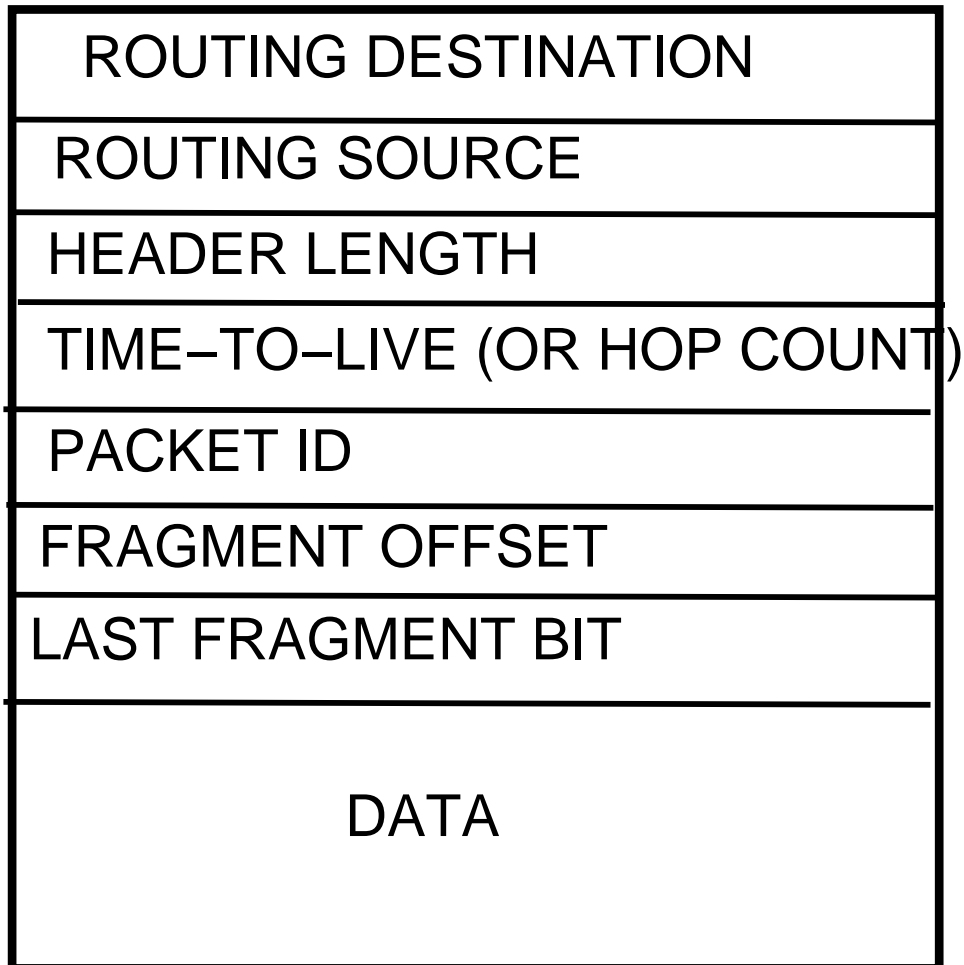
Fragmentation and Reassembly



Strong move to avoid fragmentation by having source determine optimal packet size. Determined by setting don't fragment bit, and routers send ICMP messages to source.

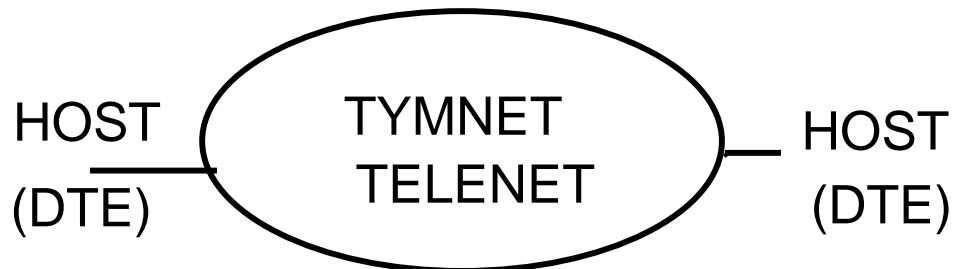
Generic IP or OSI Header

*GENERIC ROUTING HEADER
(IP or OSI)*



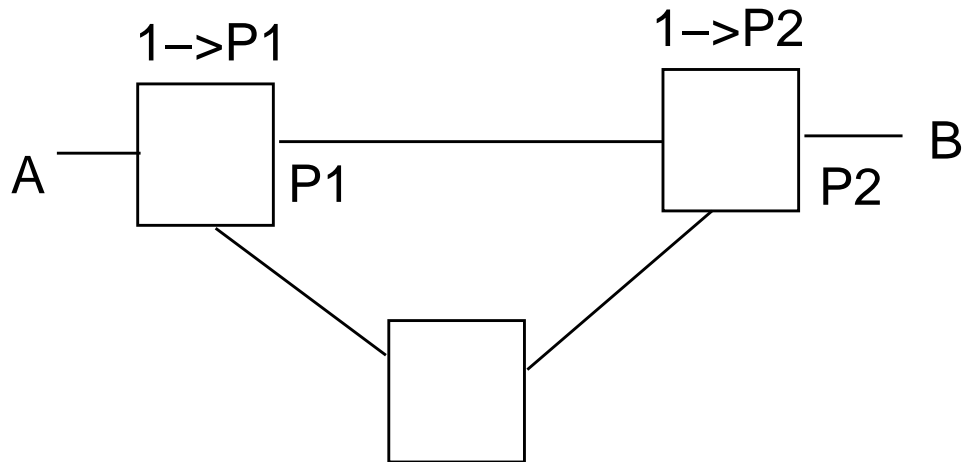
Virtual Circuit Networks

Early Virtual Circuit Networks



- 1960's and 70's. Higher speed than modems. Needed to bill users to make a profit.
- So natural to use a telephone (i.e., virtual circuit) model. Some people felt that it was also more reliable than datagrams. Used hop-by-hop reassembly and flow control.

Assigning Call Numbers



- How do you assign call numbers uniquely for all possible source destination pairs that wish to communicate?
- Need a new trick

More details

- In X.25 packets are sent reliably and in FIFO order. ATM is FIFO but not reliable.
- Thus all we need is a simple bit for the last fragment. Called the More bit which is set to 1 for all but the last fragment. Same idea used in ATM.
- ATM uses fixed size (53 byte cells) in order to avoid data packet forwarding from affecting latency of video and voice cells. (X.25 allowed larger packets).
- Call set up done by first choosing a good route (using any datagram scheme like LSPs) and then sending a call set up packet on the route. At each hop, each router picks an unused VC for the outgoing hop and sends the call set up packet on with the new VC number. A Call reply packet then comes back on same path to source: then the VC is set up. Similarly for Call Teardown.

X.25 to ATM

THE WHEEL OF TIME

- Early commercial (60's - Early 70's):
Point-to-point links, connection oriented (IBM's SNA, Tymnet).
- Middle Commercial (70's - 90's): Datagram LANs (IP, OSI).
- Future Commercial (90's - ?): ATM
(Point-to-point), connection oriented.
- ATM falling out of favor in LAN but quite heavily used in backbone.

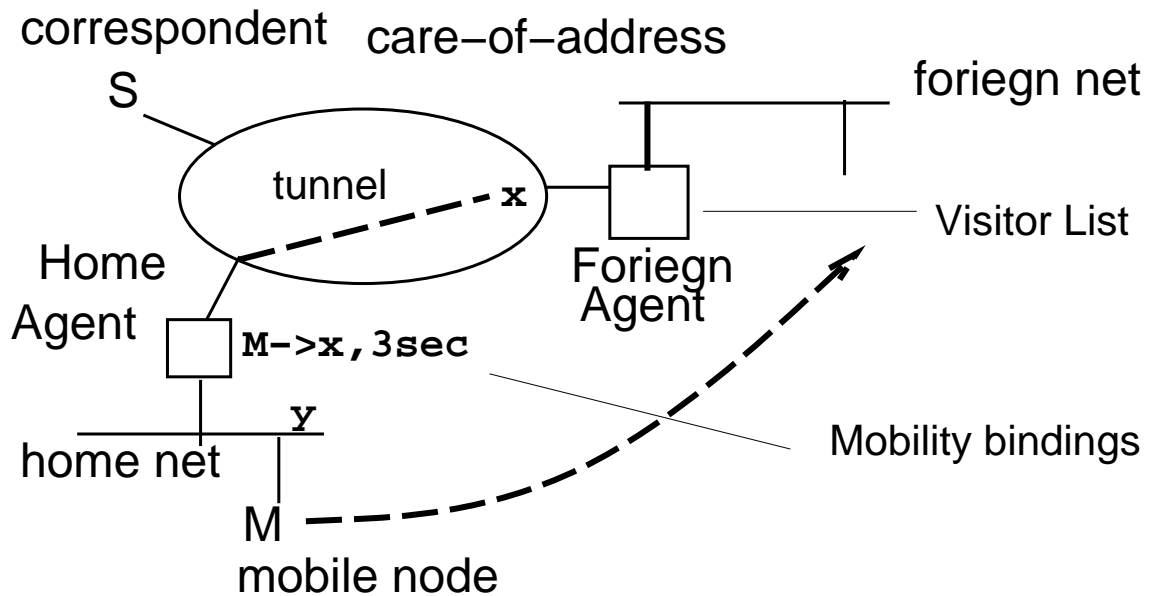
Driving factors for ATM

- Telephone company getting Digital anyway. Wants new markets and desires to be a major player in the data network. (Data jacks just like phone jacks? Have the infrastructure.) Standard TDM channels inefficient for bursty data.
- Fibre providing fast point-to-point links. Fast crossbar switches allow more parallel switching in the “routers” than the traditional LAN wiring concentrator.
- Easy to forward at high speeds once a VC is set up because we just have to lookup the VC and all the forwarding info is set up at VC set up time. Essentially a table lookup that can be done in hardware in a few clock cycles. (Prefix forwarding on the other hand is complex). Easy billing.

Segue: Aspects of the Future

- Multicasting *MBONE protocols*
- Mobility and Ubiquitous Computing (Laptops, PDAs) *Mobile IP*
- Increased Scale (users, required and available bandwidth) *IP v6*
- Integrated Services (voice, video, data, telemetry) *DiffServ, fair queuing*
- Accounting *TBD*
- Better ways to access information and use the network *TBD*

Mobile IP



- Agent advertisements, agent solicitation, agent discovery, registration, security bindings

Quality of Service

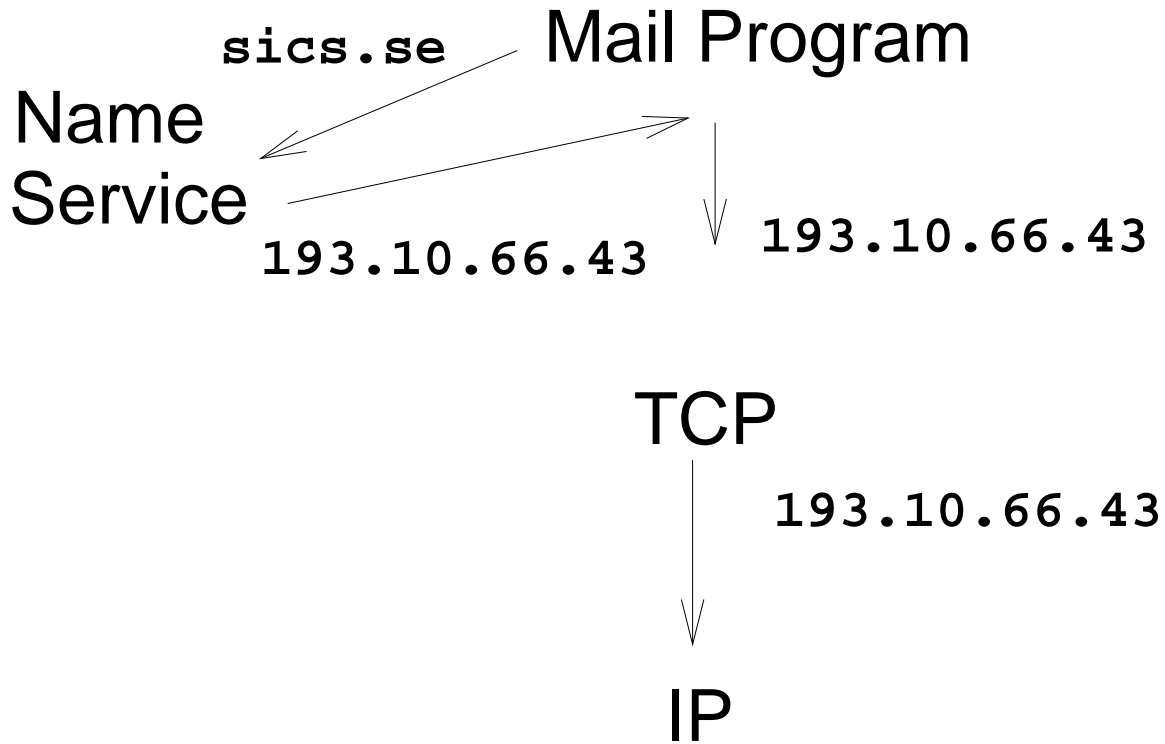
- With the advent of multicast video and audio, it is important to have routers modified to give guarantees to such stringent applications using reservations.
- DiffServ is a proposed means by which edge routers mark packets using a few bits in the IP TOS field so that they can be treated differently in the backbone. Allows us to have premium service in ISPs that can be charged more but guaranteed resources.
- Also involves routers doing some form of fairness among different types of flows using traffic shaping or round robin techniques.

IPv6

- Subnetting and BGP-4 (which helped CIDR to be deployed) have contained scaling problems. Still 4 billion hosts are not going to be enough, especially with address inefficiencies.
- IETF began looking at the need for larger addresses in 1991. Requires a new version: huge change. While doing so, might as well fix old problems (autoconfiguration) and add new features (real time, security, mobility). Requires graceful upgrade from v4 to v5.
- IETF appointed a committee called IPng and made a spec for 128 bit IPv6 addresses. Not doing well right now because of so-called NAT boxes that allow many devices in a small office to share an IP address if there is some way to tell the different applications apart.

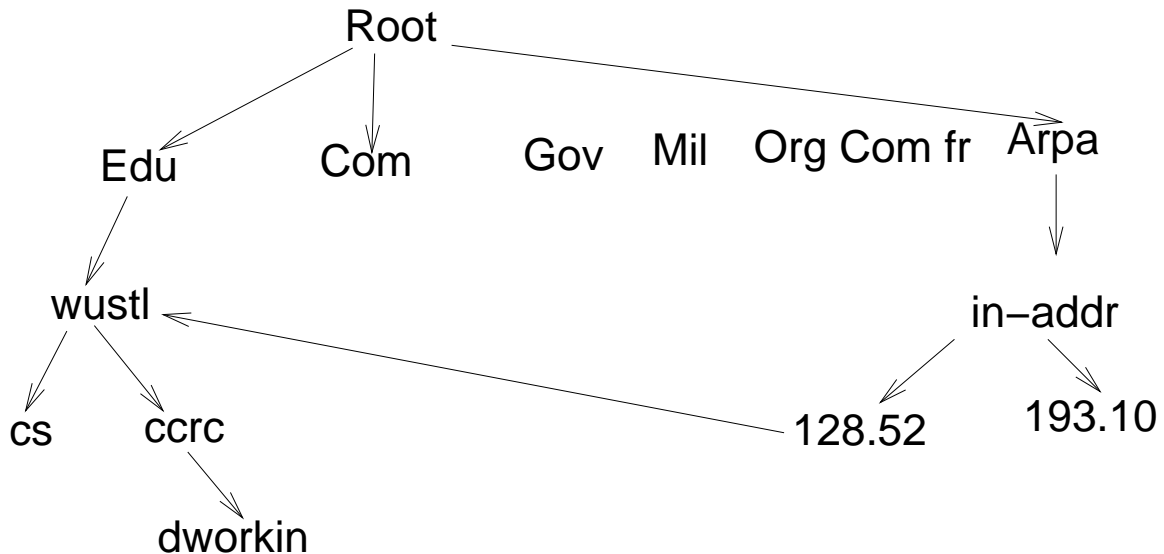
Segue 1.8 Domain Name Service (DNS)

DNS Basics



- The name service is part of application (linked in) using routines like *gethostbyname*. TCP and IP deal only with IP addresses

DNS Hierarchy



- Name space is naturally hierarchical. Want to exploit hierarchy by dividing natural hierarchy into subblocks called zones and implement all information in a zone in two or more name servers.
- ARPA domain is for reverse lookups.

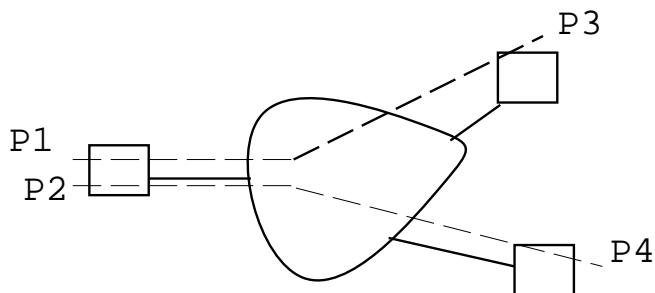
Procedure

- Start in file `/etc/resolv.conf` Gives name of nameserver (sometimes on machine itself or points to local nameserver). Also gives domain name so we can expand local short hand into a FQDN.
nameserver 128.252.169.2 dworkin nameserver 128.252.169.1 ccrc nameserver 128.252.120.1 wustl
- Each name server will either return answer (in domain or cached) or will go to next higher server in subtree. Every server has the hardwired address of one root server (`ns.internic.net`). Then work downward till find authoritative name server.

1.9 TCP

BASIC QUESTIONS

- 1) What function does a transport provide?
- 2) What is a connection?
- 3) Why not have just one connection for all data?
- 4) Why not keep connections up always?
- 5) How do we address the receiving process in the receiver machine?



P1,..P4 denote processes (mail, ftp)

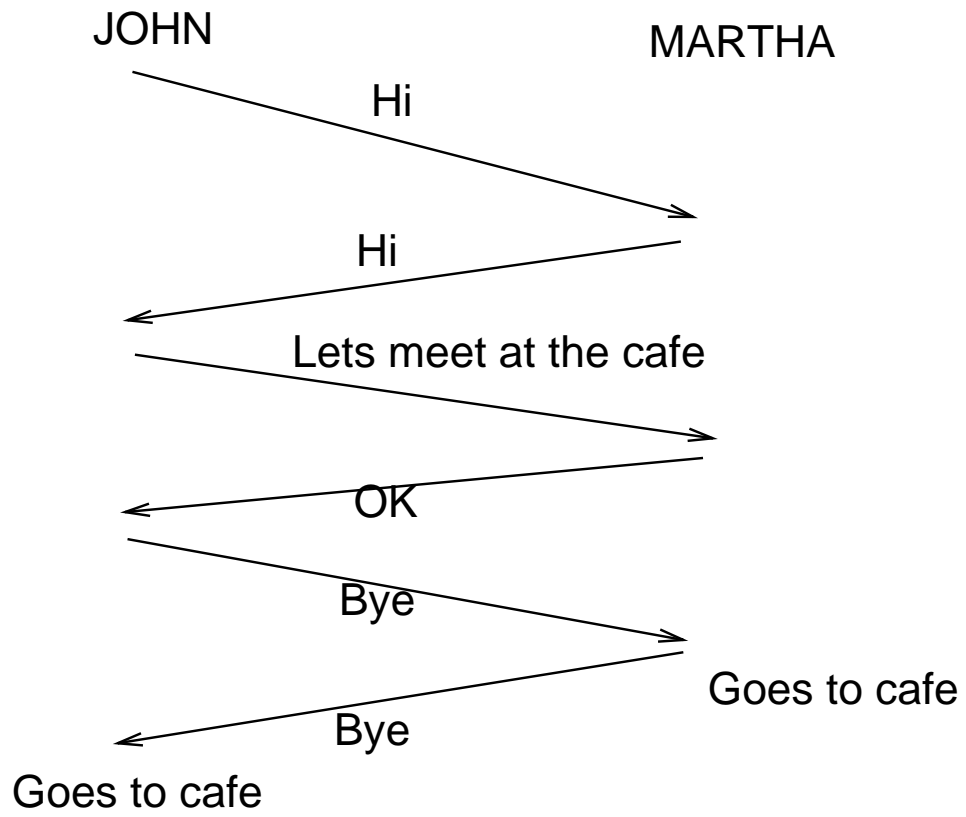
--- lines denote connections

Differences between Data Link and Transport

- Network instead of single FIFO link
 - packets can be delayed for large amounts of time
 - duplicates can be created by packet looping: delayed duplicates imply need for large sequence numbers.
 - packets can be reordered by route changes.
- Connection management
 - Only done for Data Link when a link crashes or comes up
 - Lots of clients dynamically requesting connections
 - HDLC didn't work: here more at stake, have to do it right.
- Data link only needs speed matching between receiver and sender (flow control). Here we also need speed matching between sender and network (congestion control)
- Transport needs to dynamically round-trip delay to set retransmit timers.

Delayed Duplicates

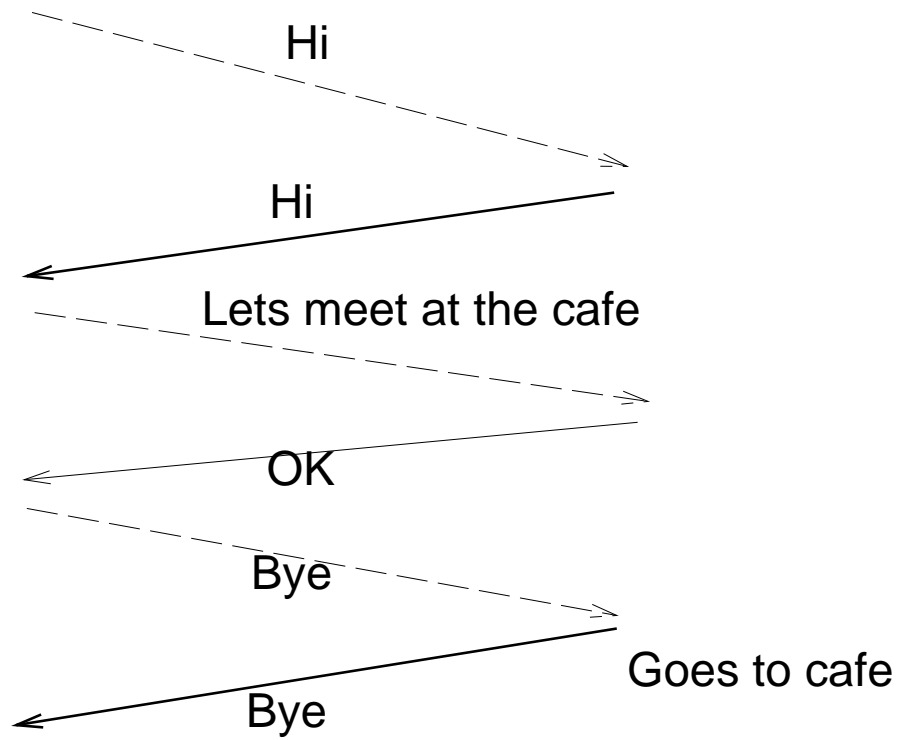
- Because of non-FIFO delivery and repeated sending at sender, receiver can get delayed duplicates
- Delayed duplicates can cause serious damage for connection management as we see from the following John and Martha story.

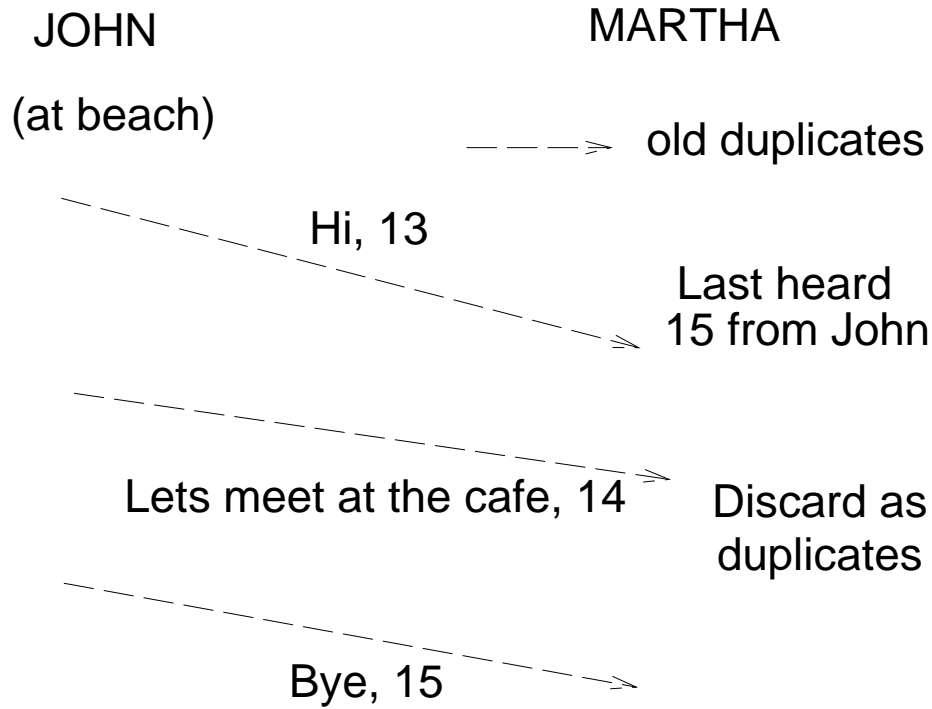


JOHN
(at beach)

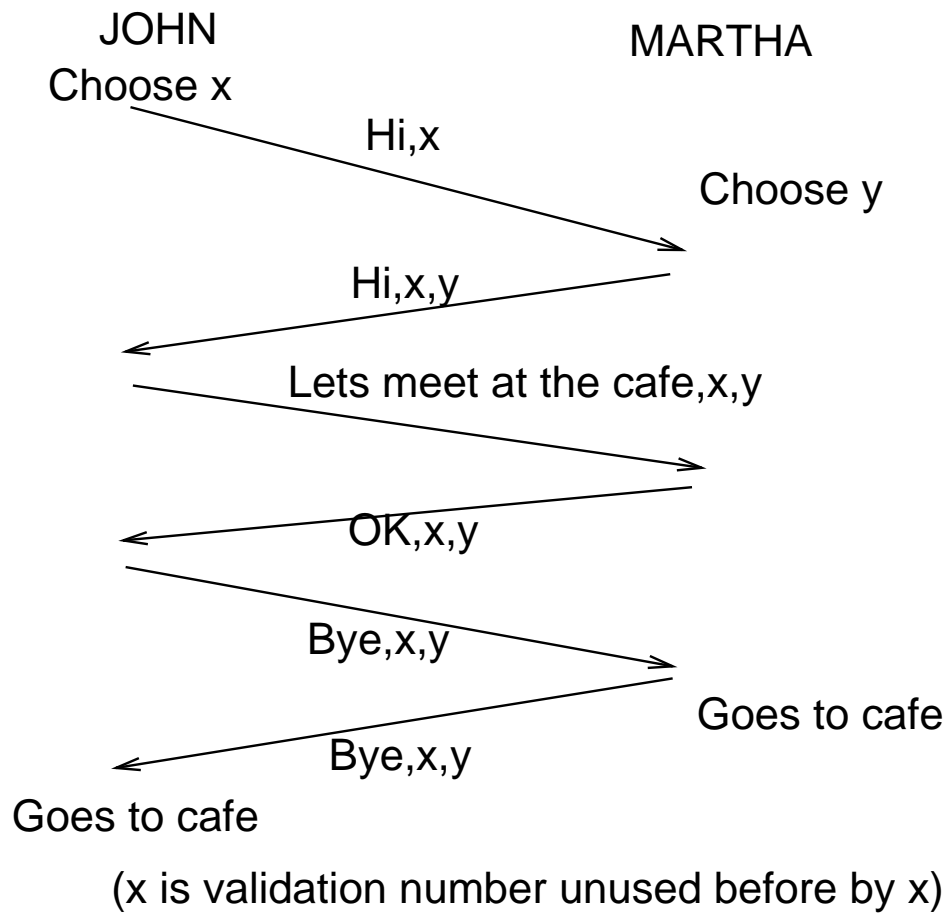
MARTHA

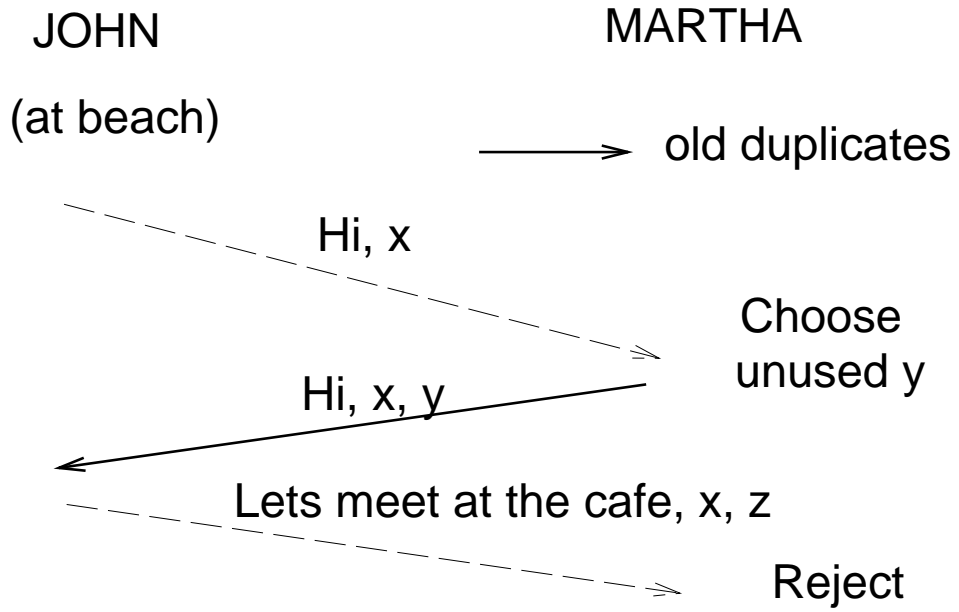
-----> old duplicates





Martha has to remember last number from John for worst-case packet lifetime. Called Timer-based connection management.





Crashes

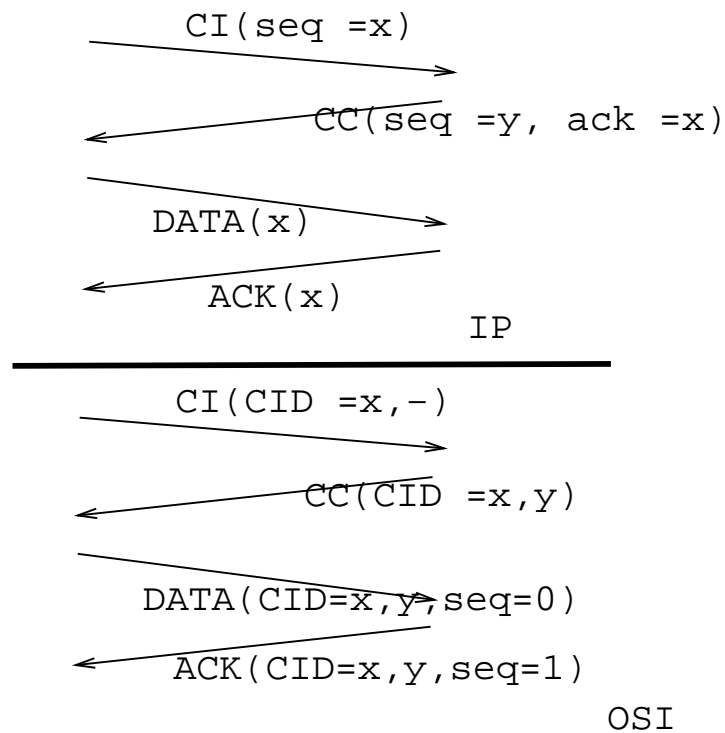
- Each send (e.g., John, Martha) needs to remember only one number on NVRAM that is incremented every time a new connection is set up. Without NVRAM, pick a random number from a large space.
- Note that Martha's validation number protects herself, and John's protects himself. Sauce for the goose is sauce for the gander.
- Same ideas used in TCP or OSI

Addressing

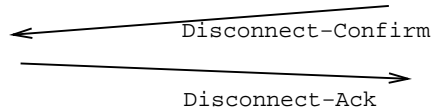
How do you know the address of the destination application? Can start by sending packets to well-known initial address with some indication of what service you want.

- TCP: Sender sends to well-known mail port say 25. Is delivered to mail process which picks unused port number (say 5) and returns number in Connect Confirm. Sender then uses specific port (i.e., 5).
- OSI: Same as TCP initially except that after initial setup, the connection identifier (combination of validation numbers) identifies connection.

IP VERSUS OSI: HANDSHAKES

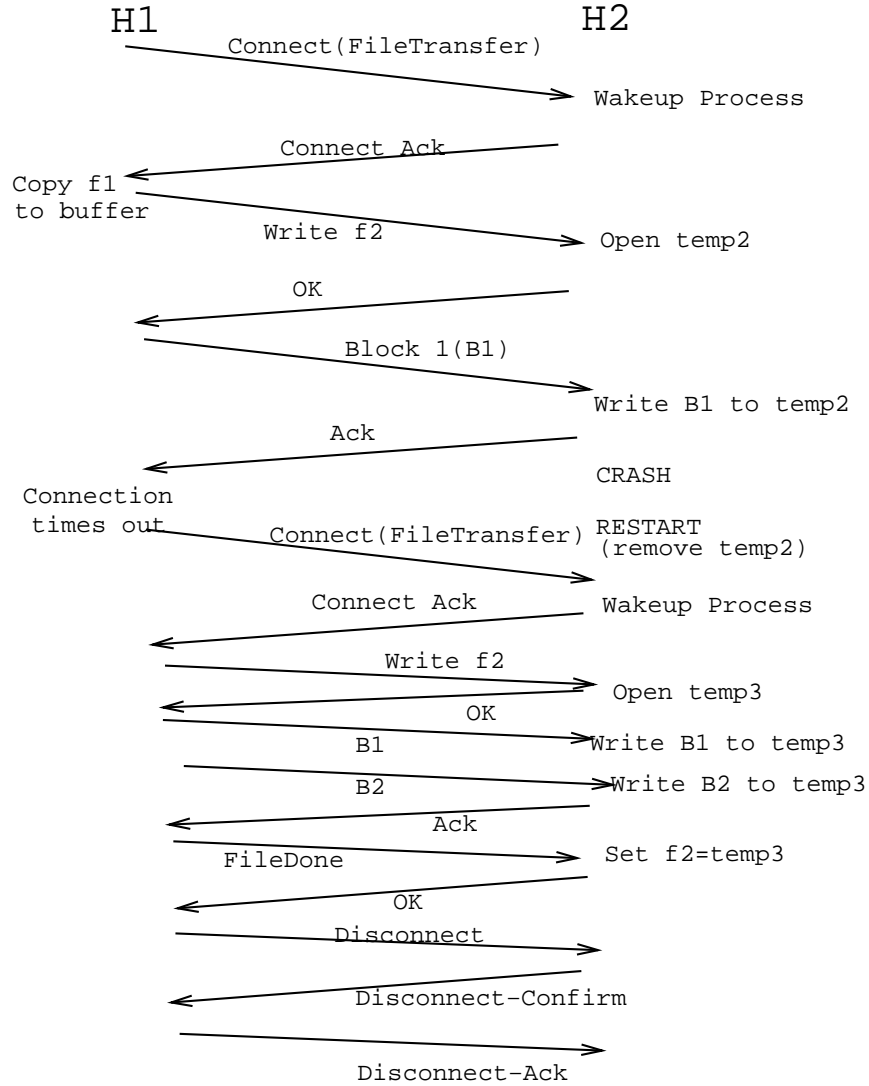


In IP the handshake sets the initial sequence numbers; in OSI, it sets up a CID and resets seq. numbers



TRANSFERRING A FILE

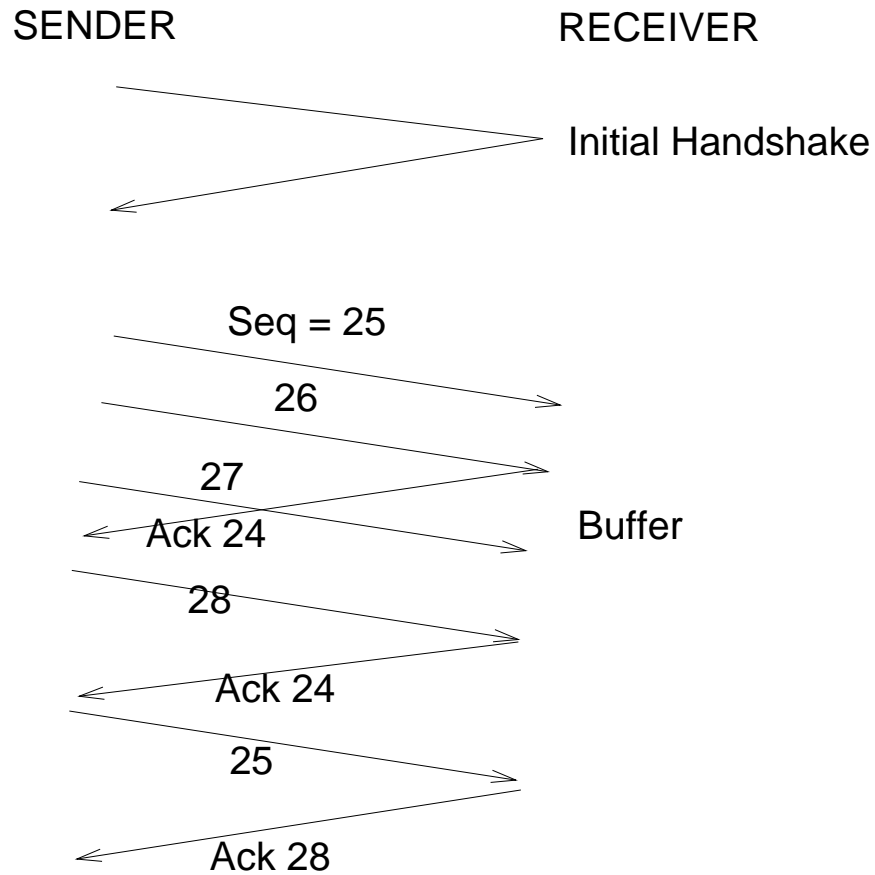
copy f1 f2@H2



DISCONNECTION

- 1) Need timers anyway to get rid of connection state to dead nodes.
- 2) However, timer should be large so that "keepalive" hello overhead is low.
- 3) If communication is working, would prefer graceful closing (so receiver process knows quickly) to long timers.
- 4) Hence 3 phase disconnect handshake
After sending disconnect and receiving disconnect ack, both sender and receiver set short timers.

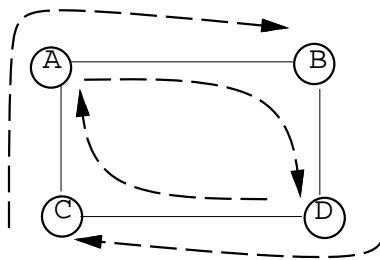
TCP Fast Retransmission



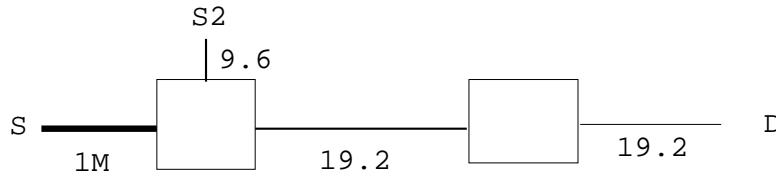
- Buffers out of order segments
- Guesses based on duplicate acks that 1 packet is lost. Move to TCP with Selective ACKS (TCP-SACK)

CONGESTION CONTROL: ISSUES

- Dynamic Problem, not static.
- Can't be solved by rerouting traffic within network; must stop admission to network.
- In worst case, throughput can collapse to 0. (Congestion collapse).
- 2 models for dealing with finite buffers:
 - Drop packets when buffers get full — — > LIVELOCK.
 - Wait at previous hop till next hop buffers empty. Highway model — — — > GRIDLOCK.



FEEDBACK CONGESTION CONTROL



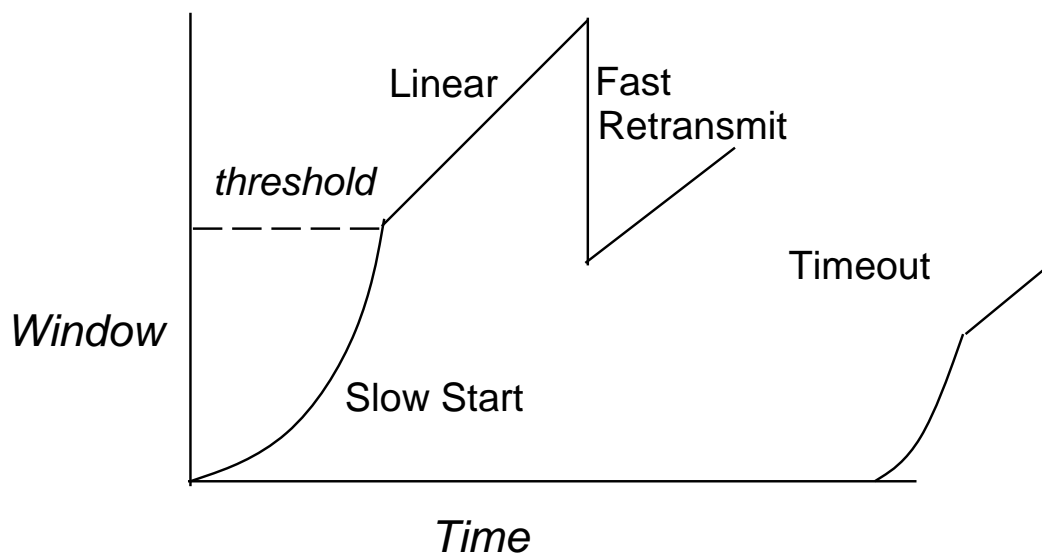
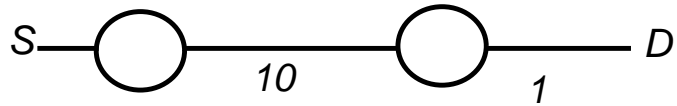
- 1) DETECT CONGESTION
- 2) FEEDBACK INFORMATION TO THE SOURCE
- 3) SOURCE ADJUSTS WINDOW:
INCREASE POLICY
DECREASE POLICY

TWO INTERESTING CASES:

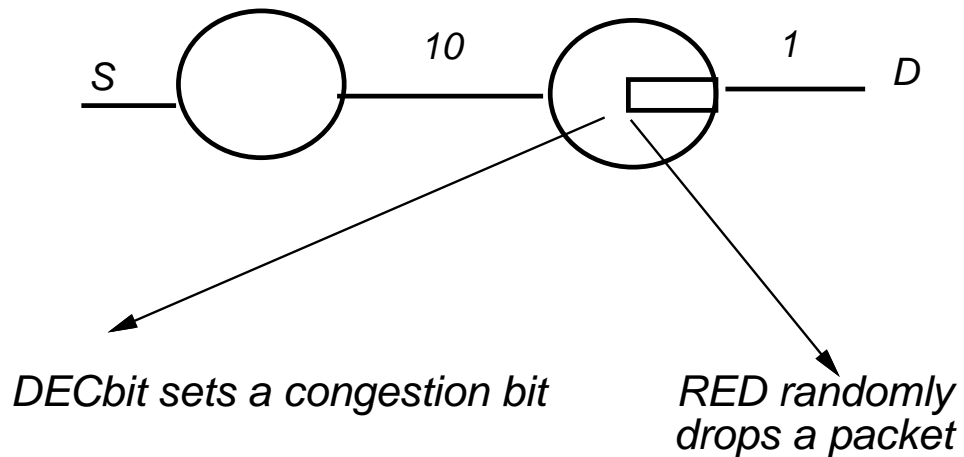
- a) HOW A SOURCE REACHES STEADY STATE.
- b) HOW A SOURCE REACTS TO A NEW SOURCE
TO PROVIDE A FAIR ALLOCATION.

CONGESTION AVOIDANCE (OSI) VERSUS
CONGESTION CONTROL (TCP)

TCP Congestion Control



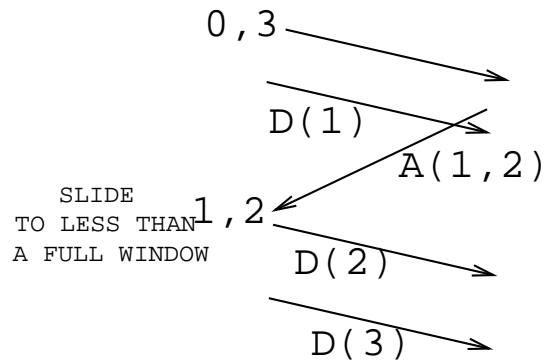
Warning Sources of Impending Congestion



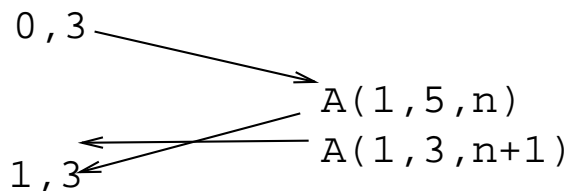
- Today's IP has no room for a congestion bit, so they randomly drop packets with small probability when average queue size passes a threshold (RED).
- Proposal on table for a ECN bit for IPv6.

FLOW CONTROL

- Windows provide static flow control. Can provide dynamic flow control if receiver acks indicate what receiver will buffer.

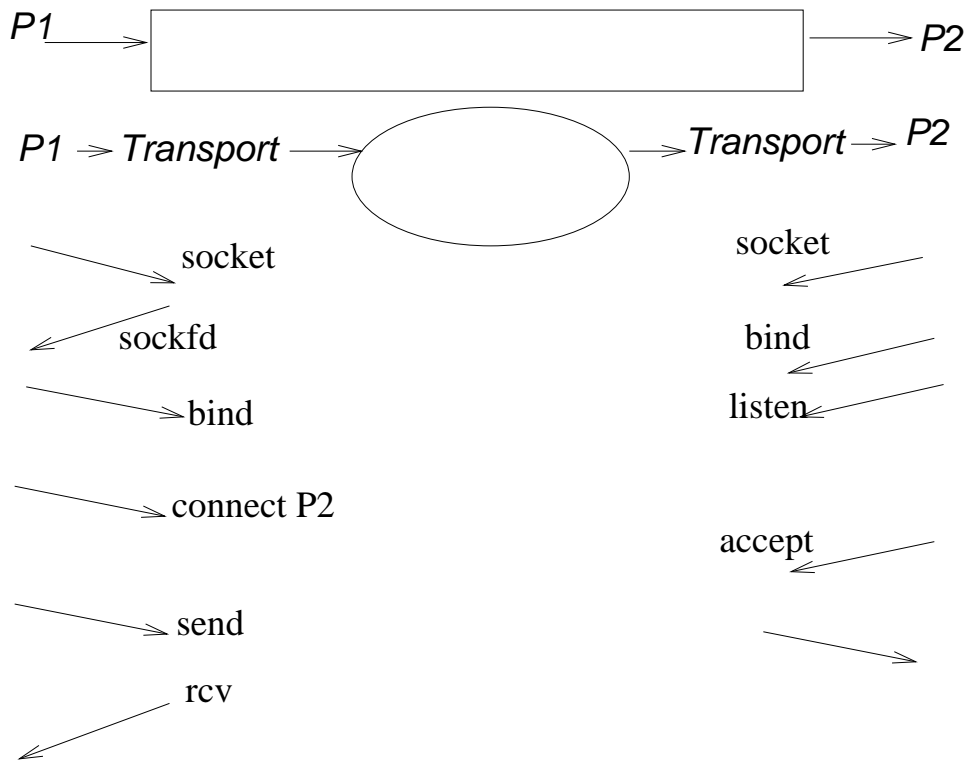


- Need to avoid deadlock if window is reduced to 0 and then increase to $c > 0$. In OSI, receiver keeps sending c . In IP, sender periodically probes an empty window.
- In OSI, receiver can retract window from say 5 to 3. To detect out of order acks, use subsequence numbers.



- Real Window = Min Flow Control, Congestion Control windows.

BSD Shared Queue (Socket) Abstraction for TCP



Real Time Transport Protocol (RTP)

- For all audio and video applications like vic and vat over MBONE.
- Sends packets with a time stamp and a sequence number. Allows data packets to be placed in order and to be discarded when late and resynchronized (but RTP does not specify).
- Audio and video use separate connections that can be synchronized using timestamps. Allows mixers and translators.