Routing

- Finding a (good) path between two remote ends in the network
- Need one or more algorithms that are most likely distributed amongst a set of hosts and routers
- Routing involves the following elements:
  - Protocols: defines how to exchange information using which to do routing decisions
  - Algorithms: use the data exchanged by the nodes to select good routes (a distributed process)
  - Routing tables: where the exchanged information or the resulting routing information is kept in routers
- Routing domain
  - A set of routers under same administration running the same routing protocol
- *Routing vs forwarding: what is the difference?*

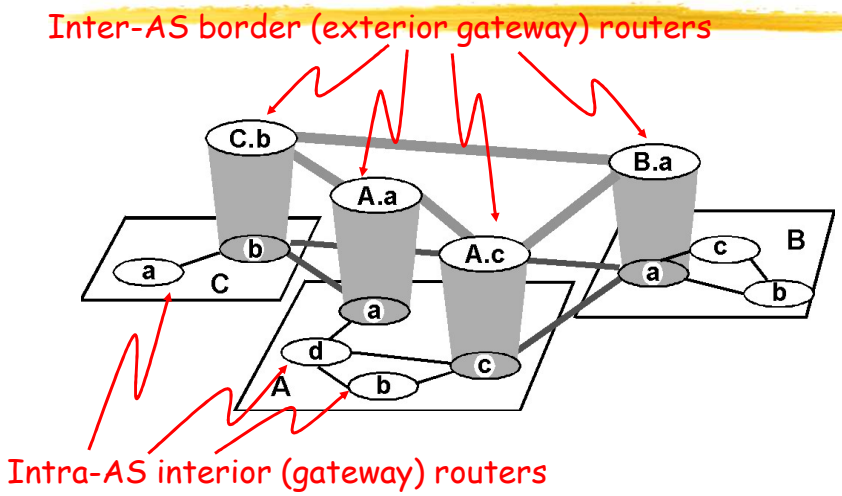
Routing

- Routing is used in all the three packet switching methods (*Datagrams*, *Virtual Circuits,* and *Source Routing*), although to a varying degree.
- The main objective of routing is the acquisition of the information for the forwarding (routing) tables.
- The distinction between a *forwarding table* (network prefix, output port, and MAC information) and a *routing table* (network prefix, next hop).
- A main issue in *distributed routing* is *consistency* and *convergence.*

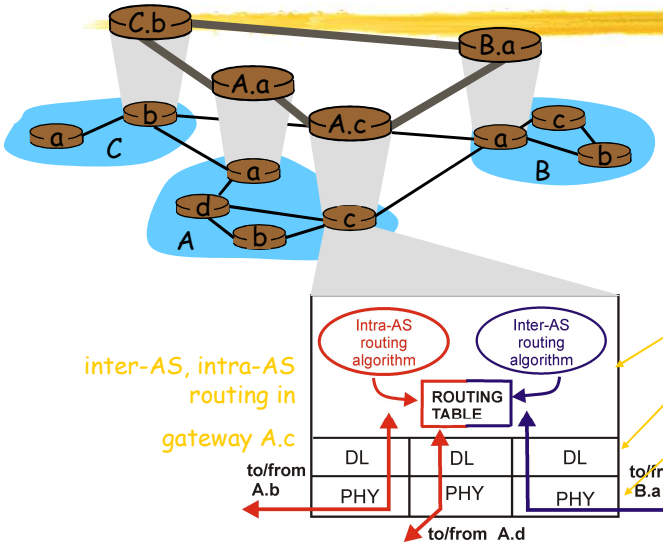
Routing in the Internet

- The Global Internet consists of Autonomous Systems (AS) interconnected with each other:
  - ***Stub AS****:* small corporation: one connection to other AS's
  - ***Multihomed AS****:* large corporation (no transit): multiple connections to other AS's
  - ***Transit AS****:* provider, hooking many AS's together
- Two-level routing:
  - ***Intra-AS:*** administrator responsible for choice of routing algorithm within network
  - ***Inter-AS:*** unique standard for inter-AS routing: BGP

Internet AS Hierarchy



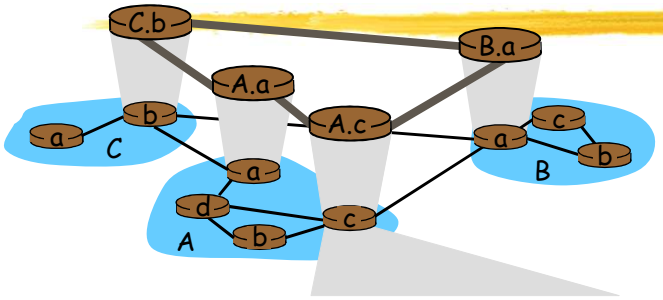
Intra-AS and Inter-AS routing



Gateways:

- perform inter-AS routing amongst themselves
- perform intra-AS routing with other routers in their AS

Intra-AS and Inter-AS routing



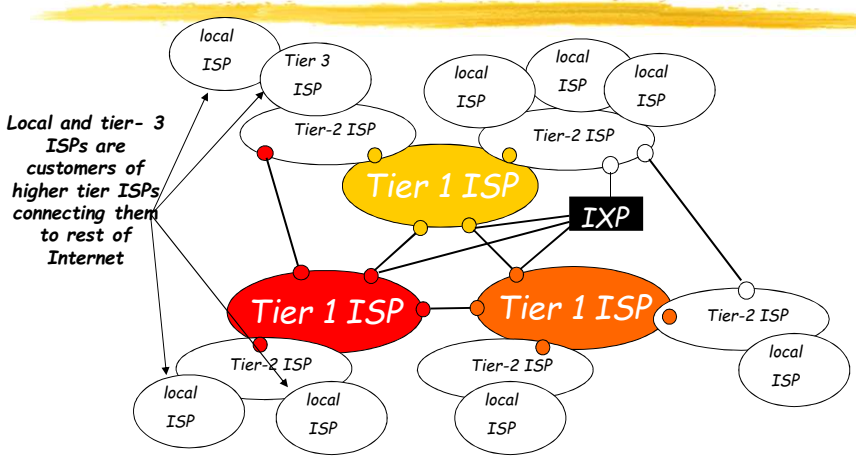
Gateways:

- perform inter-AS routing amongst themselves
- perform intra-AS routing with other routers in their AS

Intra-domain routing protocols (Interior gateway protocols (IGPs)) vs. *Inter-domain routing protocols* – it's the size of the network

The key issue: SCALABILITY

Internet structure: network of networks



Domain = Autonomous system (AS): each AS has an AS number, different from its network address

Internet Protocol (IP)

- Implements an *IP virtual network* on top of different types of hardware
  - HW is hidden by the network layer (except for link MTU)
- Service Model – (what IP promises to provide)
  - An addressing scheme to uniquely identify each node
  - Connectionless (datagram-based) service
  - Best-effort delivery
    - No reliability guarantees
      - Packets may be lost, delayed for long time
      - Packets may be duplicated, delivered out of order
    - No bandwidth or delay guarantees

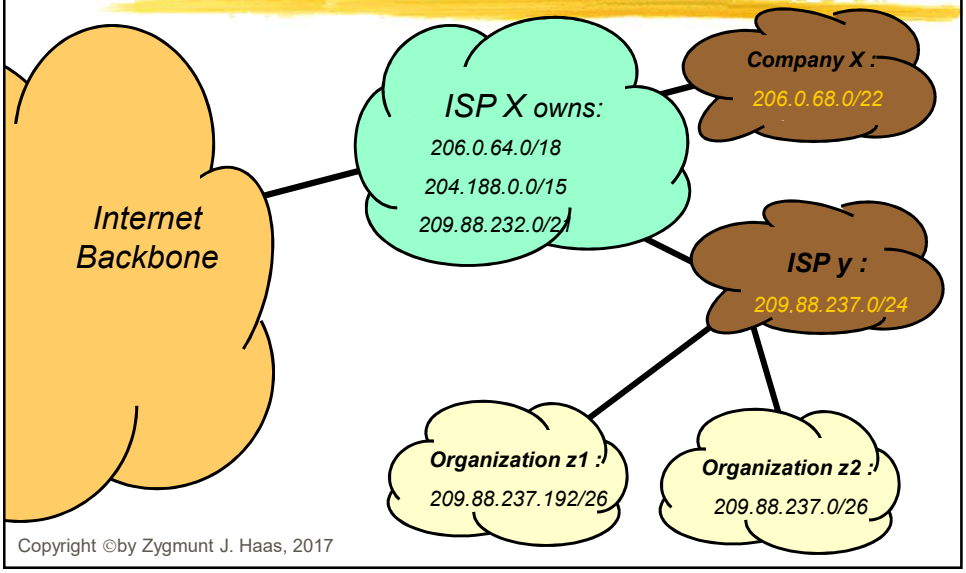
Forwarding Alternatives for IP

- IP forwarding based on global IP addresses
  - Packets carry destination IP address
  - Routers forward based on destination IP address
- Source routing
  - Source node includes the path into the packet
- Virtual circuit based IP forwarding
  (i.e. MPLS forwarding – this is below IP)
  - Build virtual circuits to route packets over
    - Can be pre-computed or in run time
  - Packets will carry tags to be forwarded over the circuits

Forwarding based on Global IP Addresses

- Each packet includes a full destination address
  - Remember that IP addresses are globally unique
- Switches/routers include a "forwarding" table as
  destination -> next hop
- What is the granularity of the information in the "forwarding" tables
  - One entry per host on the Internet?
    - How many such entries are there globally?
  - One entry per LAN?
    - How big the resulting forwarding table would be?
  - One entry per domain (i.e. organizational network)?
    - How useful is this?

Granularity of forwarding table info



Forwarding based on Global IP Addresses

- Advantages
  - Stateless
  - Packet size efficiency – better than some others (i.e., source routing)
- Disadvantages
  - Switches/routers need to know how to reach each destination
    - Need to have a careful address assignment to avoid large tables
  - Packets are independently routed
    - What would be the problem with that?

Source routing

- Source host puts the entire (or partial) path in the packet header
  - Strict source routing: the entire path is defined and used
    - What is the main disadvantage of this?
  - Loose source routing: include landmark routers to visit on the path
    - What would be the utility of this?
- Router processing
  - No need for address lookup – why?
  - The first entry in the address list is the next hop router
  - Take it off from the list and forward the packet to that router
- Advantages
  - Forwarding is fast – no need for address lookup
- Disadvantages
  - Variable and long header size
  - Source need to keep track of an accurate picture of the topology

Virtual Circuit/Tag switching

- Connection setup/teardown for each call *before/after* data flow
- Each packet carries VC identifier (not destination host ID)
- *Every* router on source-dest path
  - Maintains "state" for each passing connection
  - Router processing
    - Lookup flow ID – simple table lookup
      - Faster than IP lookup – why?
    - Replace flow ID with outgoing flow ID
    - Forward to output port
- Advantages
  - More efficient lookup
  - More flexible – can have different paths for each flow
  - Can do resource reservation
  - Easier to implement in hardware
- Disadvantages
  - Still need to route connection setup request
  - More complex failure recovery – need to re-establish

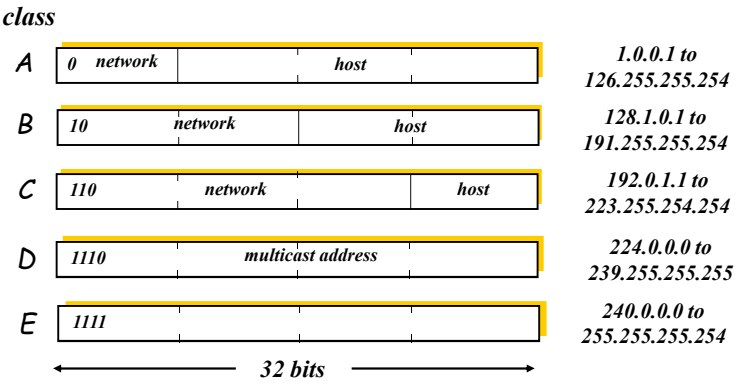
IP Addresses

IP addresses are structured/hierarchical:
(network, (*subnet*,) host) parts

Number of networks = $2^{\#network-bits}$

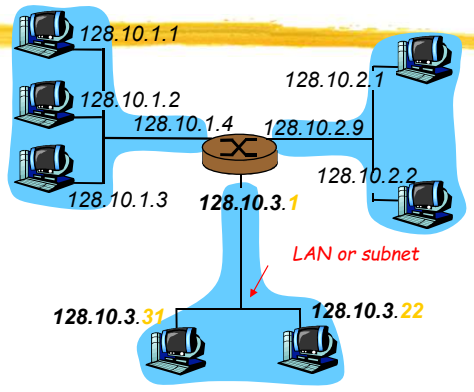
Number of hosts = $2^{\#host\_bits} - 2$

- The initial addressing scheme:
  - Classful addressing:



IP Addressing

- IP addresses are hierarchical
  - network part (high order bits)
  - host part (low order bits)
- *What's a network ?* (from IP address perspective)
  - device interfaces with same network part of IP address
  - can physically reach each other without intervening router



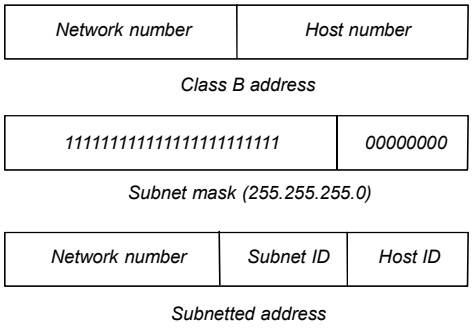
network consisting of 3 IP networks
(For IP addresses starting with 128.10.
In each, first 24 bits are network address)

Address prefix matching

- How to speed up packet forwarding in routers?
- Forwarding database in IP routers include
  - 32-bit address + 32-bit mask
  - Address prefix: initial part of the address whose length is defined by the number of "1" bits in the mask
- An IP address may match against multiple routing entries
- Example:
  - IP destination address: **11001111 01011100 00000000 10000111**
  - Entry 1:
    - Value: **11001111 01011100 00000000 10000111**
    - Mask: **11111111 11111111 11111111 11111111**
    - Prefix: **11001111 01011100 00000000 10000111**
  - Entry 2:
    - Value: **11001111 01011100 00000000 00000000**
    - Mask: **11111111 11111111 00000000 00000000**
    - Prefix: **11001111 01011100**
  - Entry 3:
    - Value: **11001111 01011100 00000000 00000000**
    - Mask: **11111111 11111111 11100000 00000000**
    - Prefix: **11001111 01011100 000**

Subnetting

- Add another level to address/routing hierarchy: *subnet*
- *Subnet masks* define variable partition of host part
- Subnets visible only within the site



IP Address Assignment Problem

- Classful addressing is bad!
  - In class A & B, address blocks are not utilized well
    - Running out of IP addresses (class A & B addresses)
  - In class C, so few addresses (only 256) per network
    - Not enough for most organization
    - Having multiple class C addresses increase the routing table sizes

Solution 1 – CIDR Classless Inter-Domain Routing (supernetting)

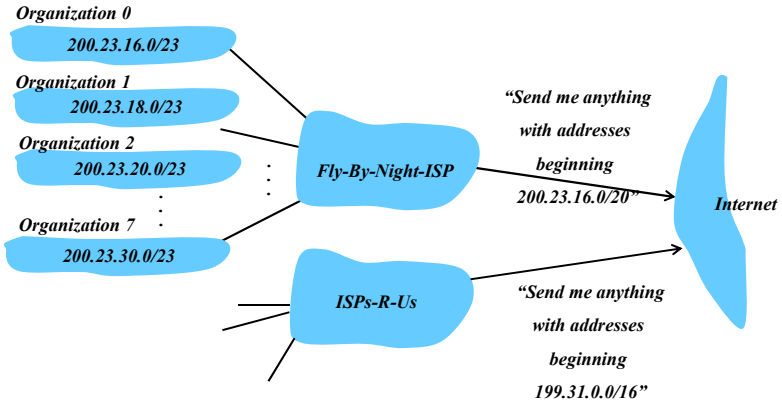
- Assign multiple class C addresses to a network
- Assign them from a consecutive blocks as to enable aggregation
- Network portion of the address is of arbitrary length
  - Do not use the classes to determine the network number
  - Use the common part of the address as network number
    - Indicate the length of this common part as network/CIDR mask
    - Address format: a.b.c.d/x where x is the #bits in network portion of the address
- Requirement: All routers must understand CIDR addressing
- CIDR tries to balance the desire to minimize the number of routes that a router needs to know against the need to hand out addresses efficiently.

Hierarchical addressing

- How does an ISP get its IP addresses?
  - Get it from IANA (Internet Assigned Numbers Authority)
- How does a customer network get its IP addresses
  - From its ISP
- How does a host get its IP addresses
  - Manual configuration, DHCP

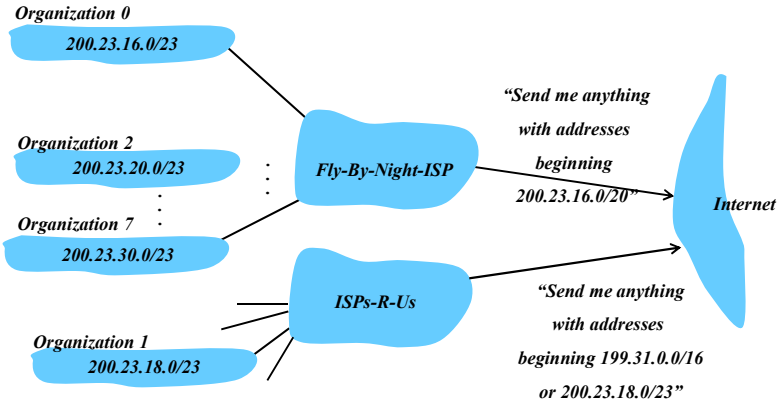
Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:

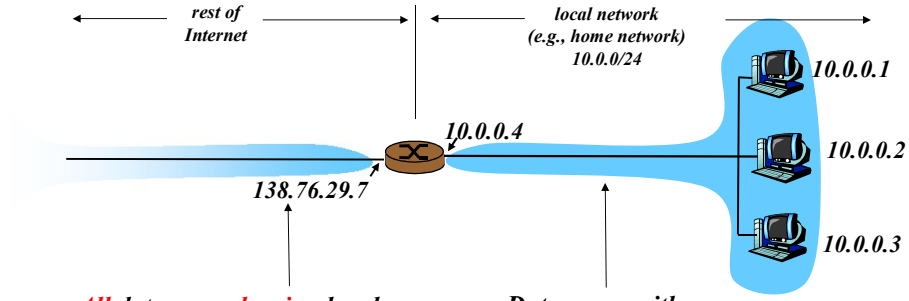


Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1



Solution 2 - NAT (Network Address Translation)



All datagrams leaving local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT

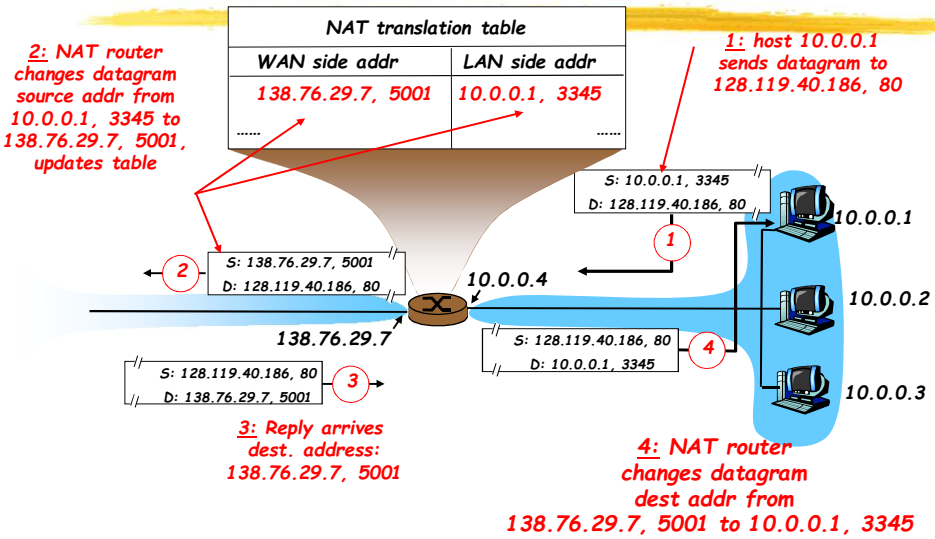
- Motivation: local network could use as few as one IP address that is globally unique in the public Internet:
  - no need to be allocated range of addresses from ISP: - just one IP address is used for all devices
  - can change addresses of devices in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - devices inside local net not explicitly addressable, visible by outside world (a security plus).

NAT

Implementation: NAT router must:

- ***outgoing datagrams: replace*** **(source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)**
  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr.
- ***remember (in NAT translation table)*** **every (source IP address, port #) to (NAT IP address, new port #) translation pair**
- ***incoming datagrams: replace*** **(NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table**

NAT

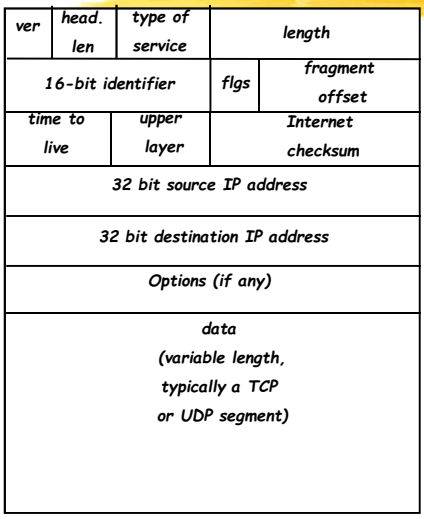


NAT

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, eg, P2P applications
  - address shortage should instead be solved by IPv6

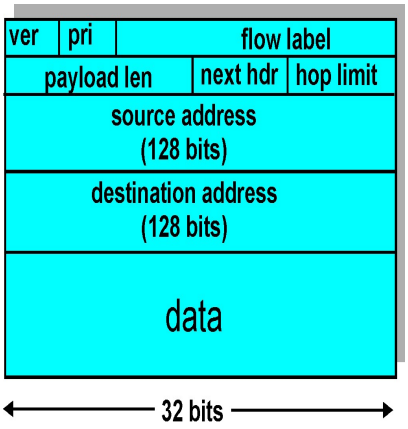
Solution 3 – IPv6

IPv4



Header is variable length: 20+ bytes

IPv6



Header is fixed length: 40 bytes

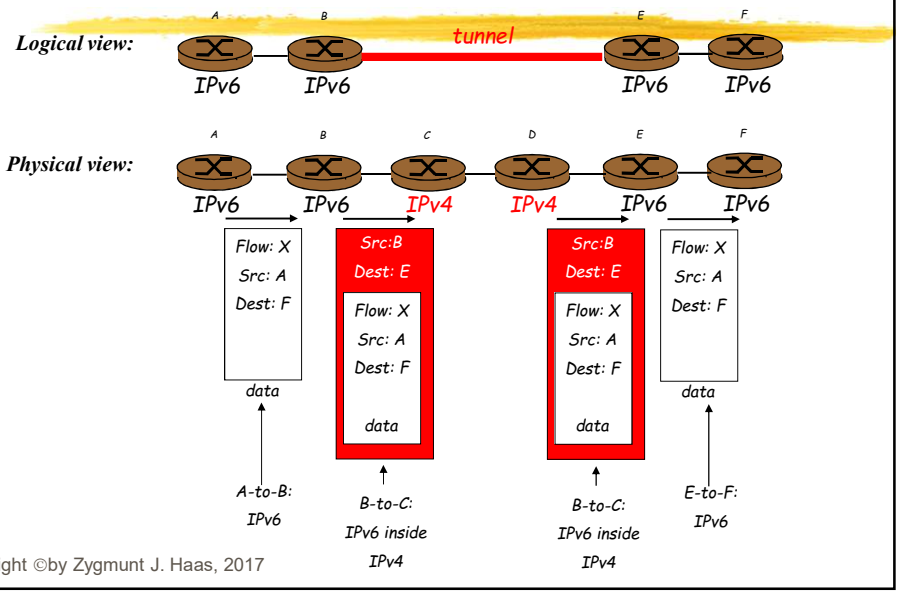
IPv6 – Changes from IPv4

- *Priority:* identify priority among datagrams in flow
- *Flow Label:* identify datagrams in same "flow."
- (concept of "flow" not well defined).
- *Next header:* identify upper layer protocol for data
- *Checksum*: removed entirely to reduce processing time at each hop
- *Options:* allowed, but outside of header, indicated by "Next Header" field
- No in-network fragmentation is allowed in IPv6
  - Need to do MTU discovery
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions

IPv6 – Transition from IPv4

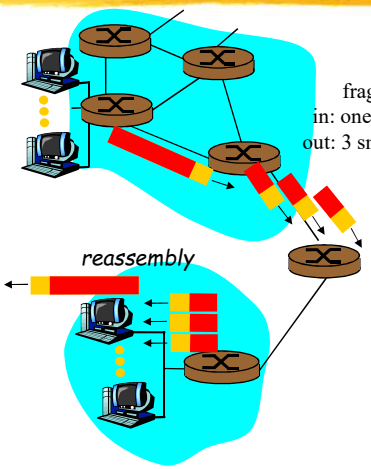
- Not all routers can be upgraded simultaneous
  - no "flag days"
  - How will the network operate with mixed IPv4 and IPv6 routers?
    - *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers

Transition – Tunneling



IP Fragmentation & Reassembly

- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments



fragmentation:
in: one large datagram
out: 3 smaller datagrams

IP Fragmentation and Reassembly

Example

- *4000 byte datagram*
- *MTU = 1500 bytes*

Offset field is 13-bits long!

need to make fragments such that the payload size is a multiple of 8

the value in offset field equals to payloadsize/8

length	ID	fragflag	offset
=4000	=x	=0	=0

One large datagram becomes several smaller datagrams

length	ID	fragflag	offset
=1500	=x	=1	=0
=1500	=x	=1	=185
=1040	=x	=0	=370

ICMP: Internet Control Message Protocol

- used by hosts, routers, gateways to communication network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing the error

Type	*Code*	*description*
0	*0*	*echo reply (ping)*
3	*0*	*dest. network unreachable*
3	*1*	*dest host unreachable*
3	*2*	*dest protocol unreachable*
3	*3*	*dest port unreachable*
3	*6*	*dest network unknown*
3	*7*	*dest host unknown*
4	*0*	*source quench (congestion control - not used)*
8	*0*	*echo request (ping)*
9	*0*	*route advertisement*
10	*0*	*router discovery*
11	*0*	*TTL expired*
12	*0*	*bad IP header*

Efficient use of available IP addresses
DHCP: Dynamic Host Configuration Protocol

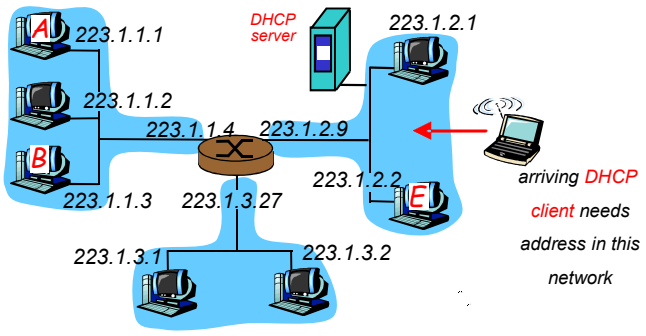
Goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- Can renew its lease on address in use
- Allows reuse of addresses (only hold address while connected and "on")
- Support for mobile users who want to join network

DHCP overview:

- host broadcasts "DHCP discover" msg
- DHCP server responds with "DHCP offer" msg
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

DHCP client-server scenario



DHCP client-server scenario