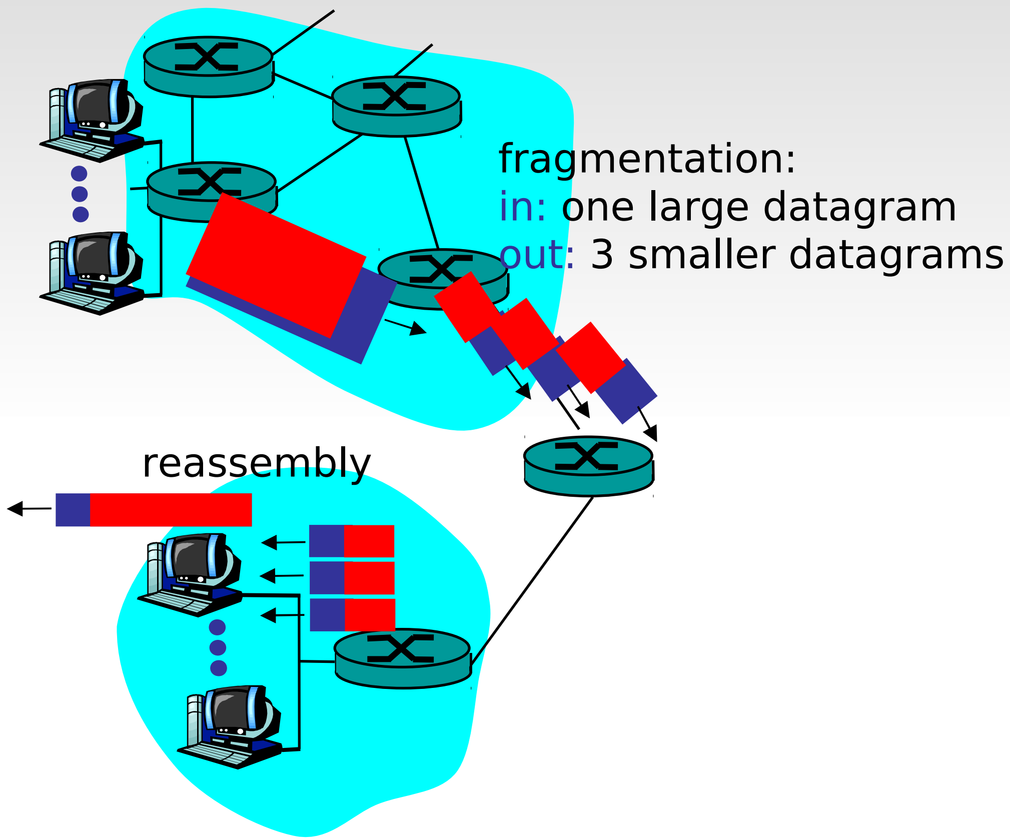
Network Layer

IP Fragmentation & Reassembly

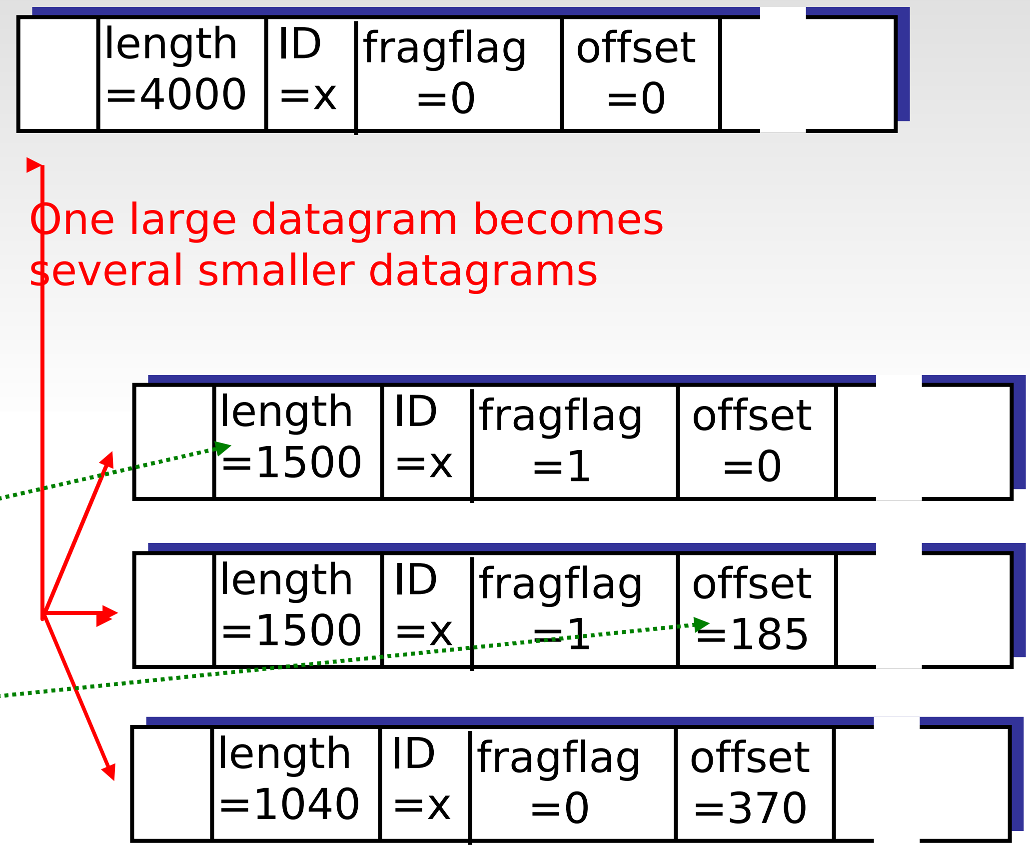
- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify,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



IP Fragmentation & Reassembly (Contd...)

Example

- 4000 byte datagram
- MTU = 1500 bytes

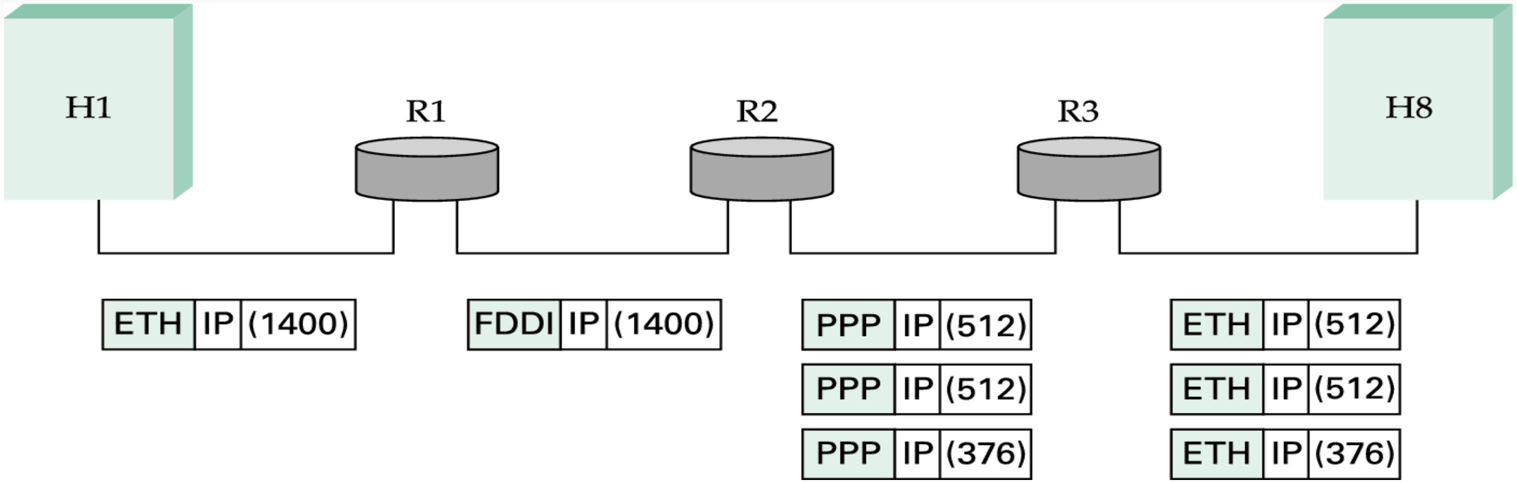


1480 bytes in data field

offset = 1480/8

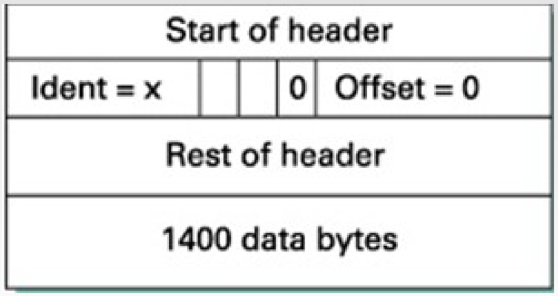
IP Fragmentation & Reassembly (Contd...)

Typical Scenario

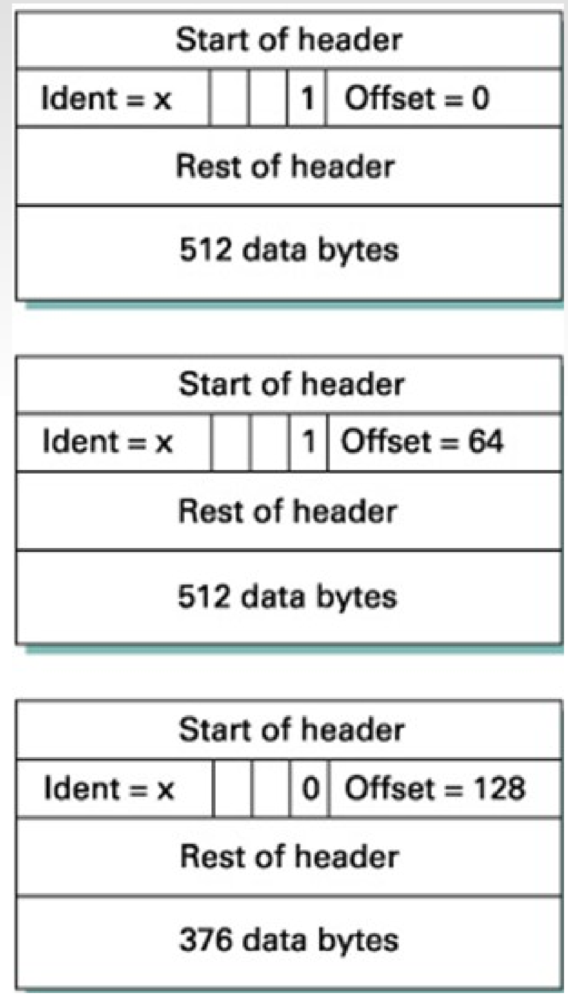


IP Fragmentation & Reassembly (Contd...)

From ETH -> FDDI



From FDDI-> PPP



Fragmentation and reassembly

- Fields related to fragmentation

1. Identification : 16 bit field
   - used to identify the datagram
   - uniqueness achieved by using counter
   - all fragments have the same identification number as the original datagram

1. Flags :3 bit field
   - first bit is reserved
   - Second bit D → 1, do not fragment. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP message to the source host
   - D → 0, fragment

Fragmentation and reassembly

- Third bit M: more fragments
  - M→ 0, no more fragments
  - M→ 1, More fragments

3. Fragmentation offset: 13 bit field

gives the relative position of the fragment with respect to the whole datagram

It is the offset of the data in the original datagram measured in units of 8 bytes

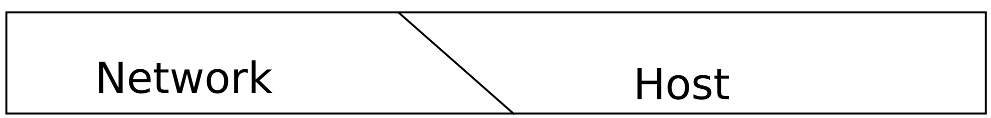
Global Addresses

Difference between Ethernet address and IP Address.

Why we are not using Ethernet address to address a node(Globally)?

IP Addresses are hierarchical… …means they are Made up of several part that corresponds to some sort of hierarchy in the internetwork

IP address (IPV4)

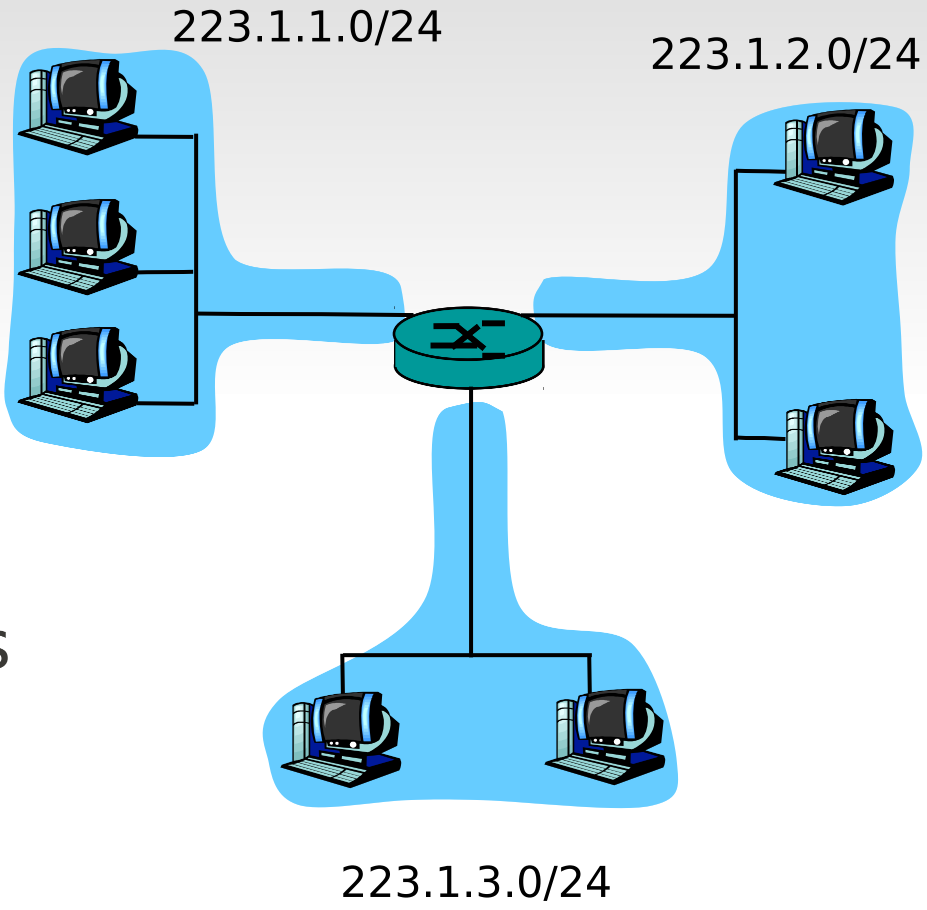


IP Addressing

- IP address: 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one interface
  - IP addresses associated with each interface

Subnetting

- subnetting refers to the partitioning of a network address space into separate autonomous subnetworks.
- Through subnetting, efficient use of an IPv4 address can be made, as it reduces wasted address space



Supernetting

- Combine several class C blocks to create a super network or supernet
- Supernetting decreases the number of 1’s in the mask where as subnet increases number of 1’s in the mask

Mask

- 32 bit number
- In IPv4 addressing, a block of addresses can be defined as

    x.y.z.t/n

where x.y.z.t defines one of the addresses and the /n defines the mask

Contd......

- The address and the /n notation completely define the whole block : the first address, last address and the number of addresses
- First address: found by setting the rightmost 32-n bits to 0s
- Last address: found by setting the rightmost 32-n bits to 1s
- Number of addresses: found by using the formula 2^{32-n}

Classless addressing

- Overcome address depletion
- Address is granted in blocks
- The size of the block(the number of addresses) varies based on the nature and size of the entity.
- The internet authorities impose three restrictions on classless address blocks
  - The address must be contiguous
  - The number of addresses must be in power of 2
  - The first address must be divisible by the number of addresses

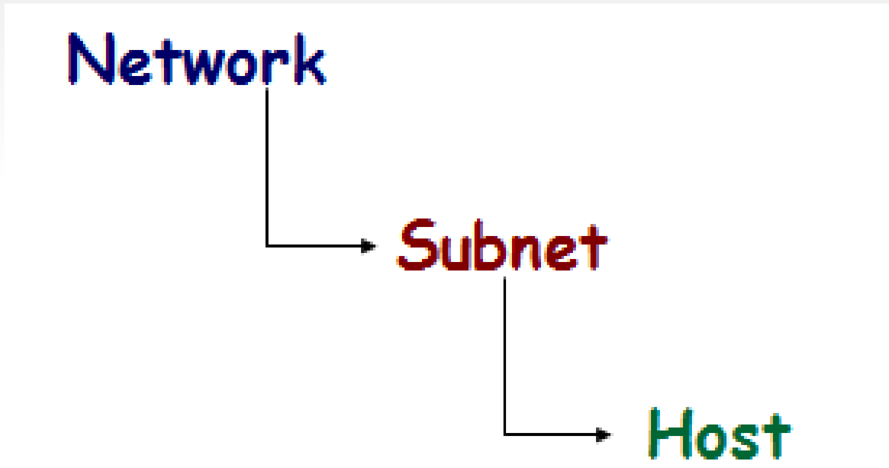
Hierarchy

- Two level hierarchy: No Subnetting
  - Leftmost n bits (prefix) define the network
  - Rightmost 32-n bits(suffix) define the host
- The prefix is common to all addresses in the network; the suffix changes from one device to another

Ntk	Host

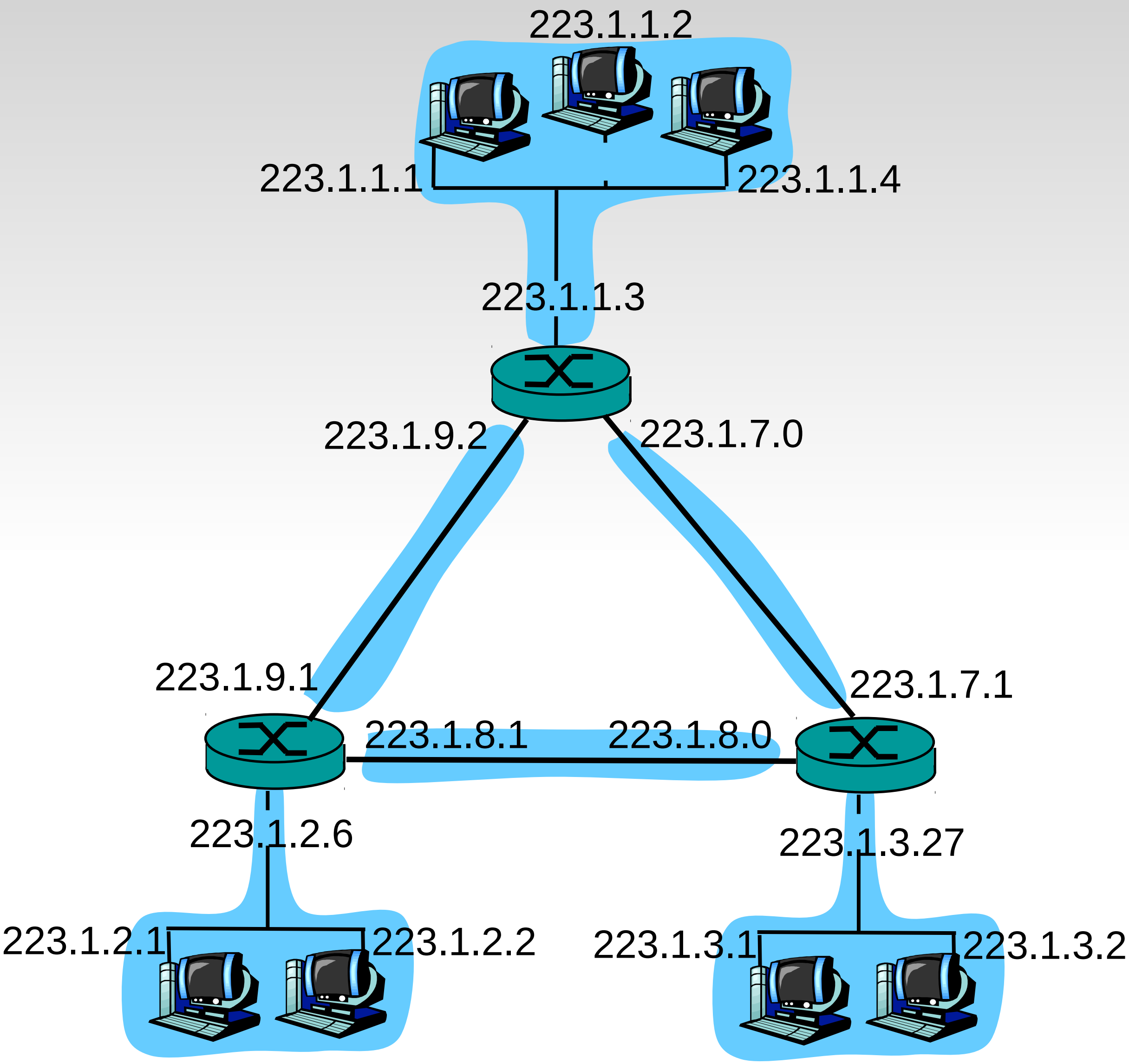
Hierarchy

- Three levels of hierarchy: Subnetting



Subnets

How many?



Special Addresses

00000000	00000000	00000000	00000000

This Host

00000000	Host

Host on this Network

11111111	11111111	11111111	11111111

Broadcast on local Network

Ntk	11111111 11111111 11111111

Broadcast on distant Network

127	11111111 11111111 11111111

LoopBack

Address allocation

ICANN

↓

National ISP

↓

Regional ISP

↓

Local ISP

↓

Organizations

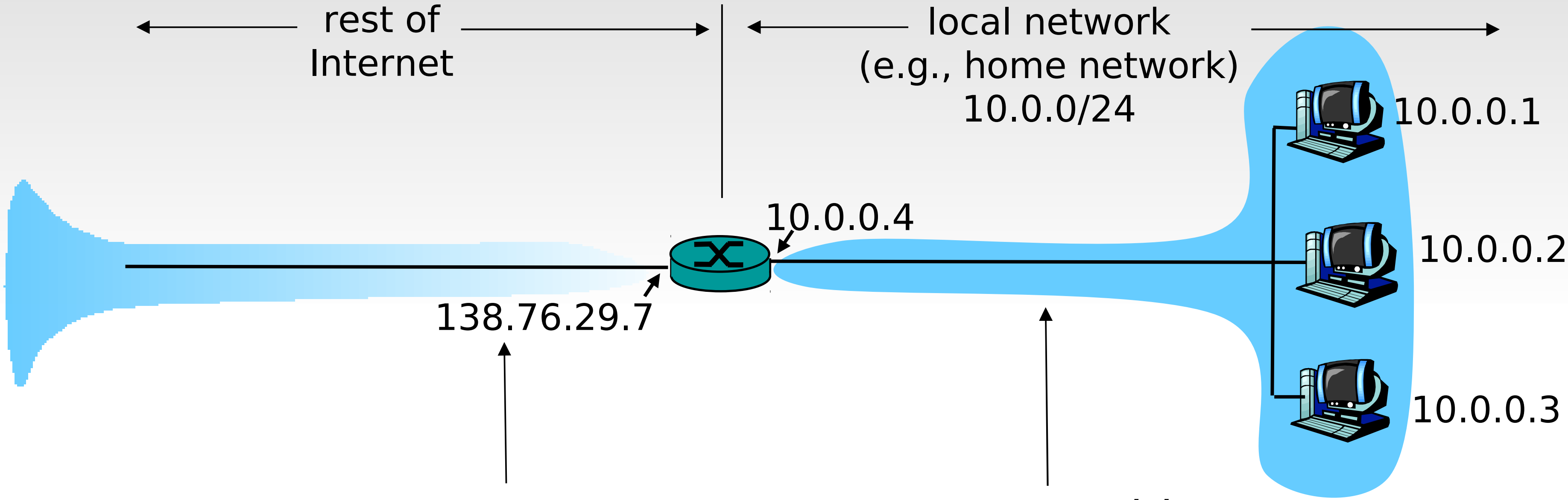
IP addressing...

How does an ISP get block of addresses?

ICANN: Internet Corporation for Assigned Names and Numbers

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

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)

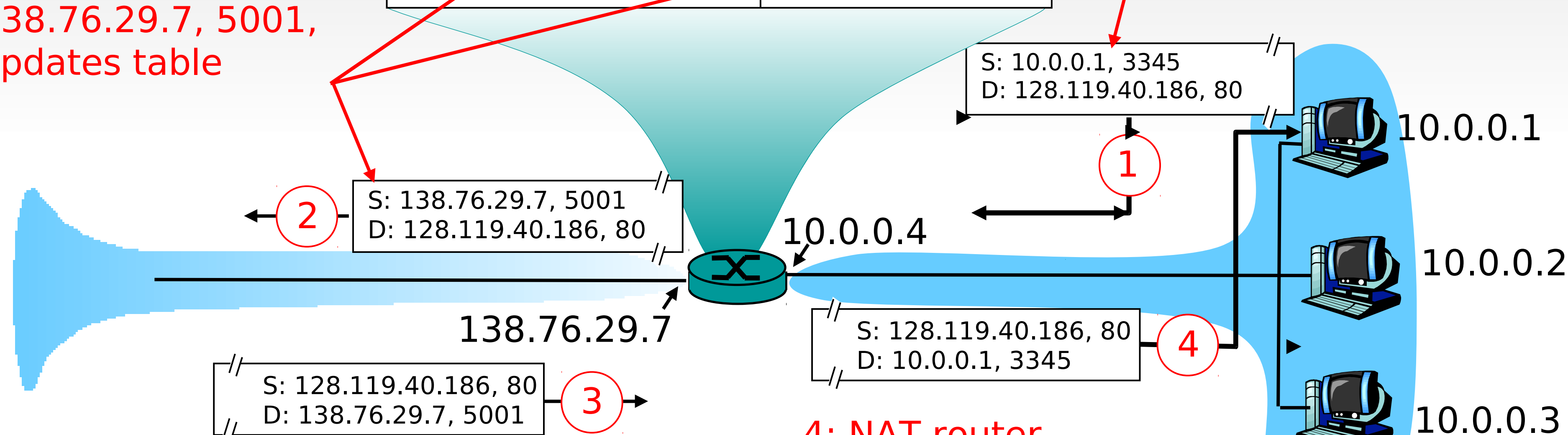
NAT: Network Address Translation

NAT translation table

WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80



3: Reply arrives dest. address: 138.76.29.7, 5001

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

ICMP: Internet Control Message Protocol

- used by hosts & routers to communicate 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 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

Trace-route and ICMP

- Source sends series of UDP segments to dest
  - First has TTL =1
  - Second has TTL=2, etc.
  - Unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram
  - And sends to source an ICMP message (type 11, code 0)
  - Message includes name of router& IP address

- When ICMP message arrives, source calculates RTT
- Traceroute does this 3 times

Stopping criterion

- UDP segment eventually arrives at destination host
- Destination returns ICMP "host unreachable" packet (type 3, code 3)
- When source gets this ICMP, stops.

IP version 6 (IPv6)

Introduction

- Deficiencies:
  - Address depletion
  - The internet must accommodate real time audio and video transmission which requires minimum delay strategies and reservation of resources not provided in the IPv4 design
  - The internet must accommodate encryption and authentication of data for some applications. No encryption or authentication is provided by IPv4
- Solution : IPv6 - IPng

Introduction (contd...)

- Initial motivation: 32-bit address space soon to be completely allocated.
- Objectives:
  - Support billions of hosts
  - Simplify the protocol, to allow routers to process packets faster
  - Pay more attention to type of service, particularly for real time data
  - Aid multicasting by allowing scopes to be specified
  - Provide better security (authentication and privacy) than current IP
  - Reduce the size of the routing tables

IPv6

- resolves current addressing problems
- maintains most IPv4 function
- IPv4 vs IPv6
  - IPv4 header length is variable, IPv6 header length is fixed to 320 bits (40 bytes).
    - In IPv6, options are separated from the base header and inserted, when needed, between the base header and the upper layer data
  - IPv4 has 14 fields, IPv6 has only 9 fields
  - IPv4 uses 32 bit address where as IPv6 uses 128 bits address – larger address space

IPv6 vs. IPv4

- IPv4 allows fragmentation where as IPv6 no fragmentation allowed
- New options: to allow additional functionalities
- Allowance for extension: extension of the protocol if needed by the technology or application
- Support for resource allocation:
  - In IPv6, no TOS field
  - flow label has been added to enable source to request special handling of packet – real time application

– Support for more security in IPv6

IPv6 format

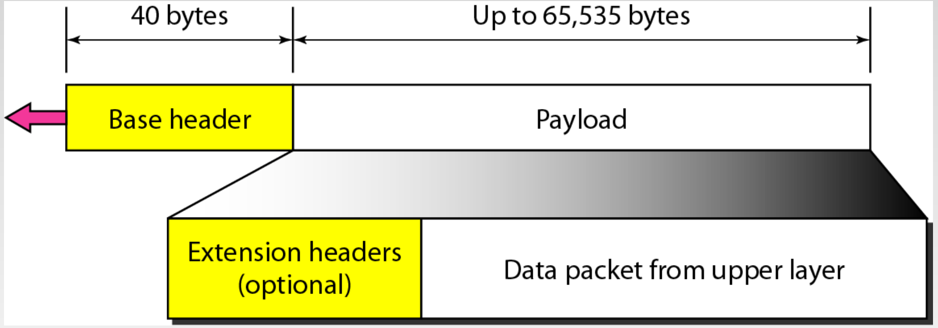
Priority: identify priority among datagrams in flow

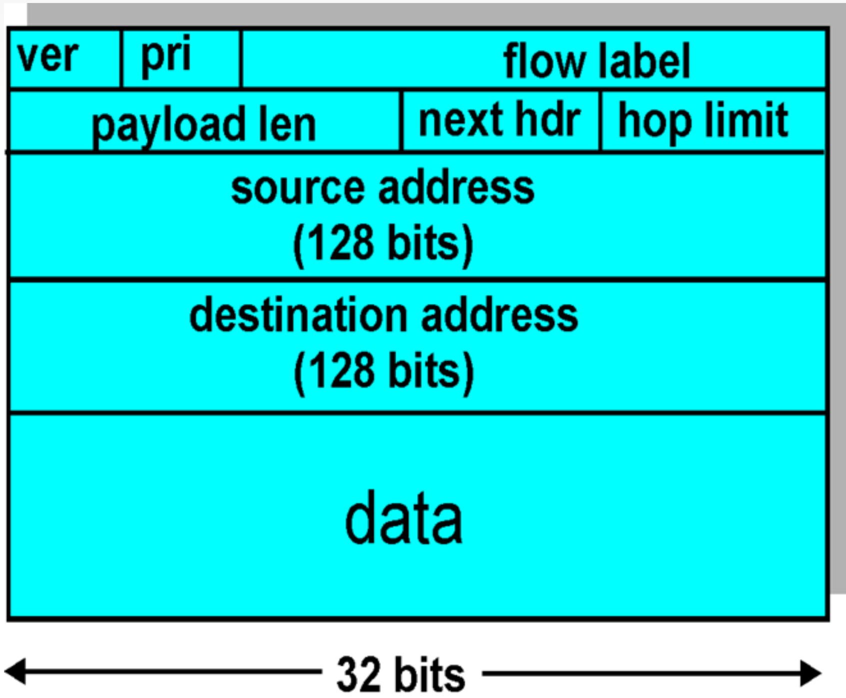
Flow Label: identify datagrams in same "flow."

(concept of "flow" not well defined).

Next header: identify upper layer protocol for data

Hop limit: equivalent to TTL





Next header codes for IPv6

Code	*Next Header*
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

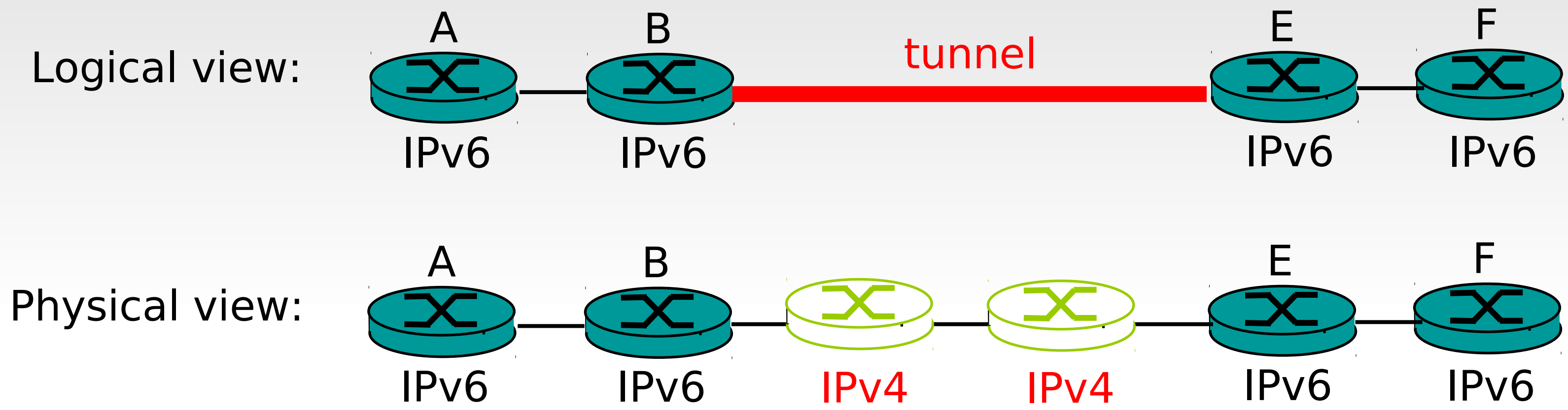
Other Changes from IPv4

- *Checksum:* removed entirely to reduce processing time at each hop
- *Options:* allowed, but outside of header, indicated by “Next Header” field
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. “Packet Too Big”
  - multicast group management functions

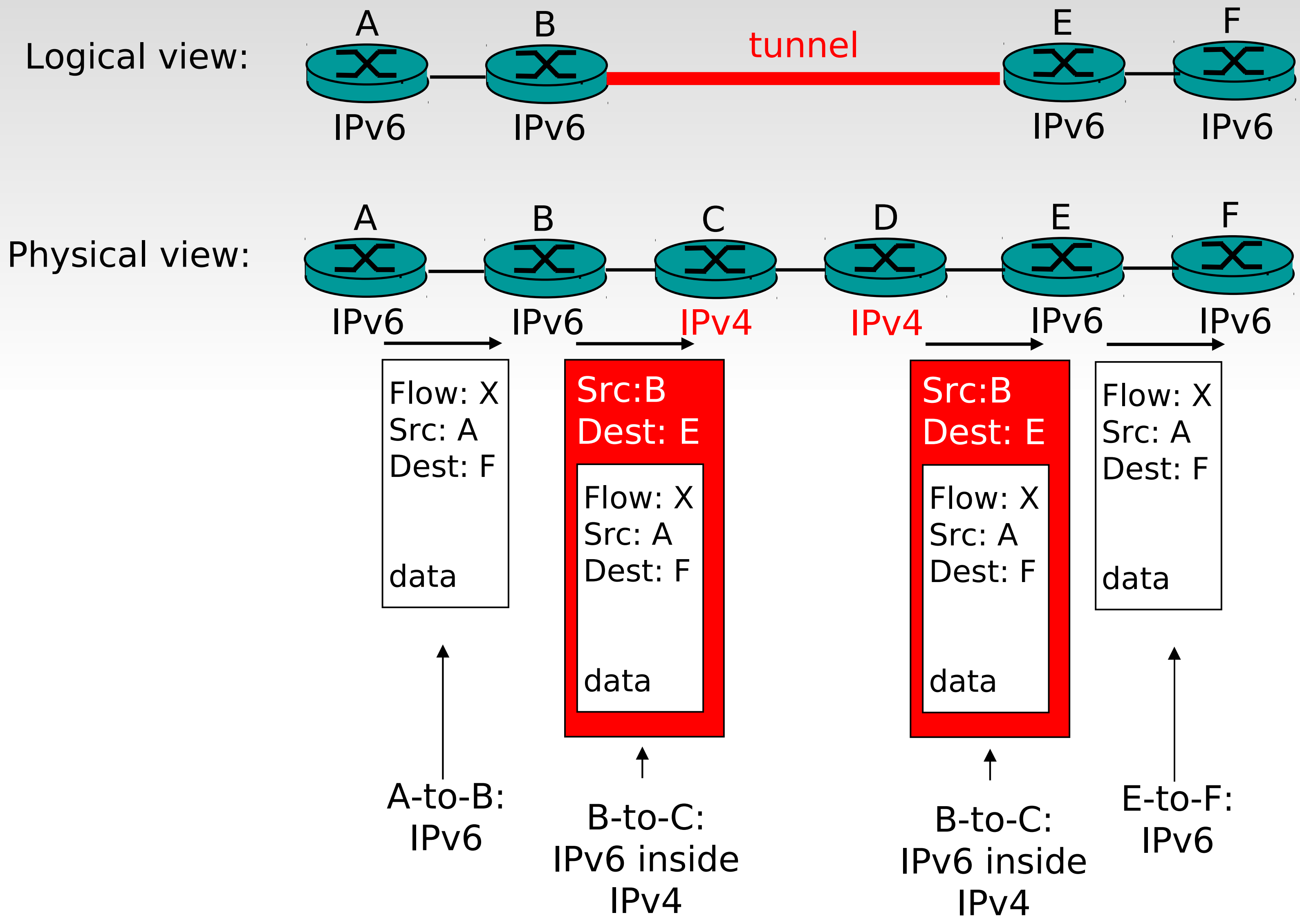
Transition From IPv4 To IPv6

- 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

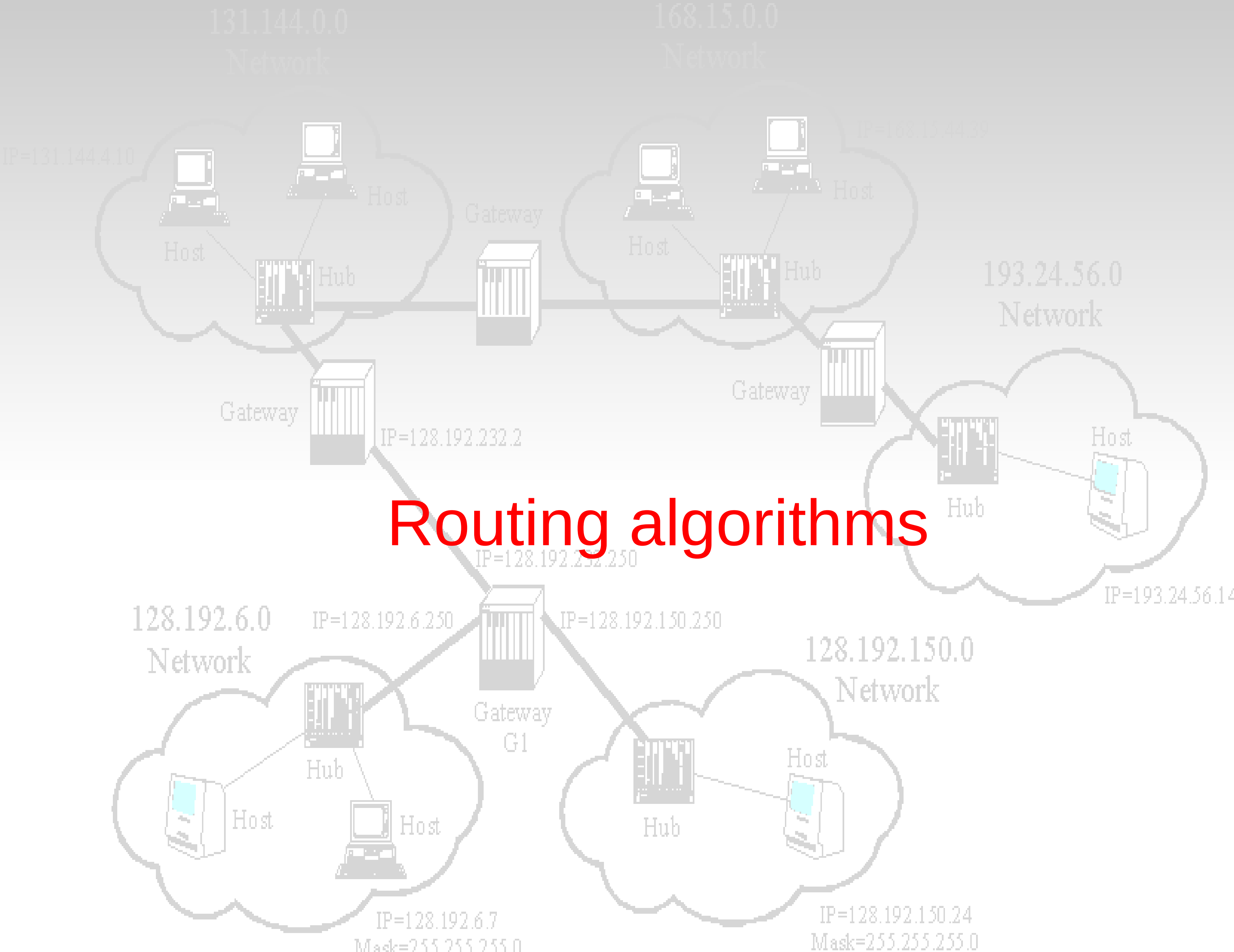
Tunneling


Logical view:
A
B
tunnel
E
F
IPv6
IPv6
IPv6
IPv6
Physical view:
A
B
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6

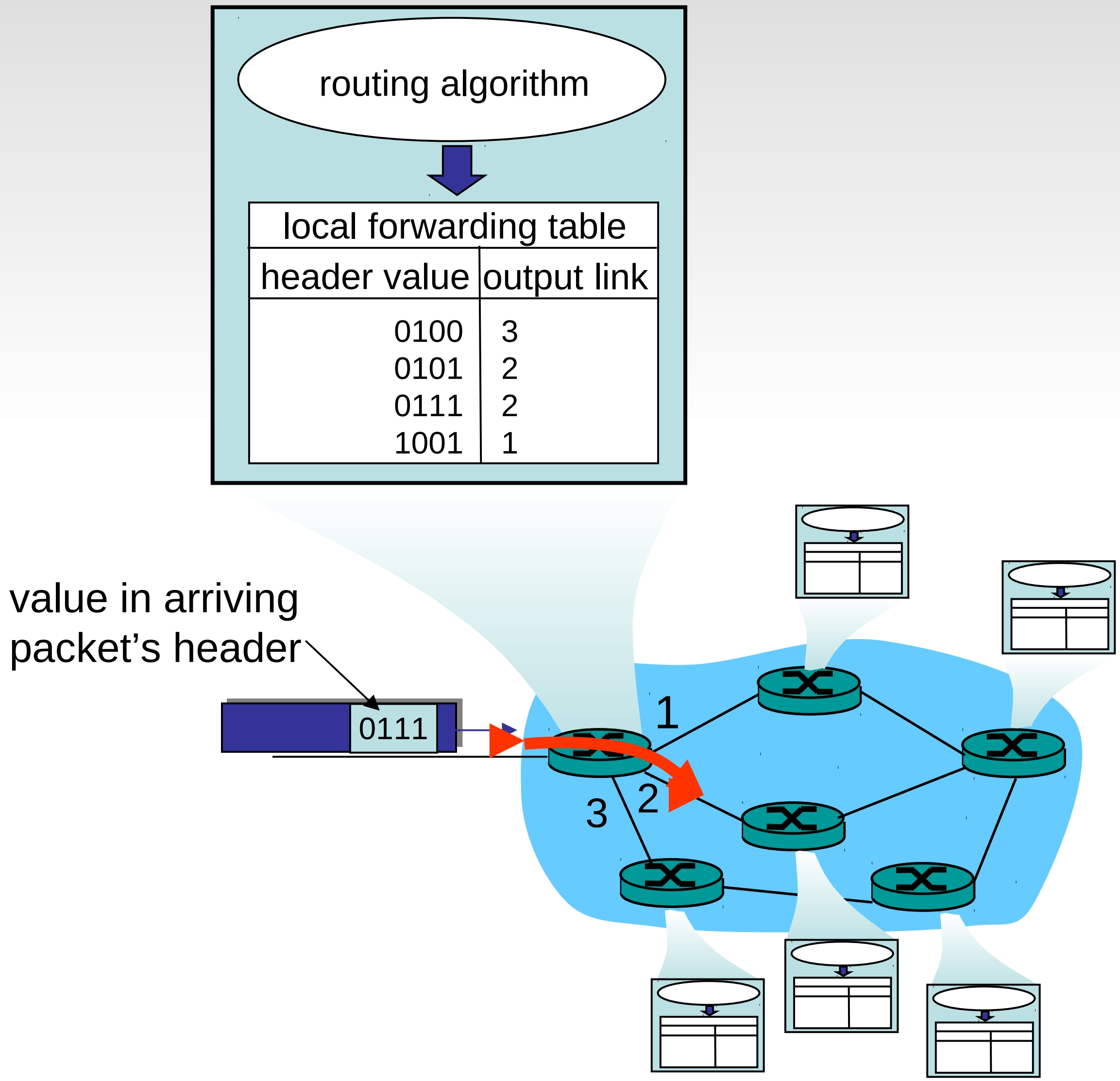
Tunneling (contd...)



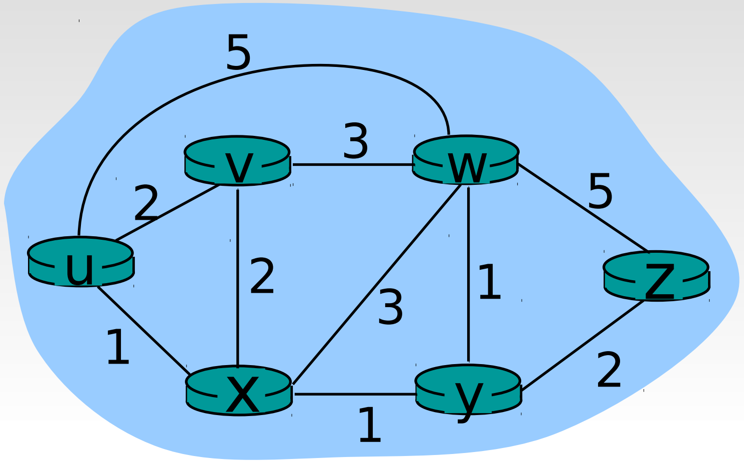
Routing algorithms



Interplay between routing and forwarding



Graph abstraction



Graph: G = (N,E)

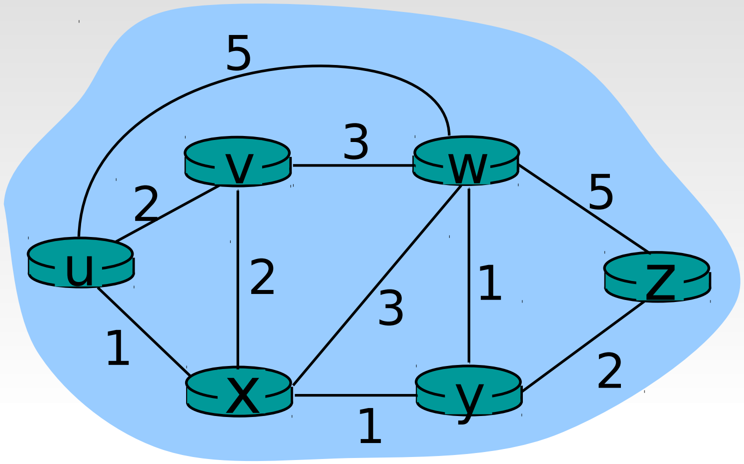
N = set of routers = { u, v, w, x, y, z }

E = set of links ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z), (u,w)}

Remark: Graph abstraction is useful in other network contexts

Example: P2P, where N is set of peers and E is set of TCP connections

Graph abstraction: costs



- c(x,x') = cost of link (x,x')

  - e.g., c(w,z) = 5

- cost could always be 1, or inversely related to bandwidth,

Cost of path $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

Question: What's the least-cost path between u and z ?

Routing algorithm: algorithm that finds least-cost path

Routing Algorithm classification

Global or decentralized information?

Global:

- all routers have complete topology, link cost info

- "link state" algorithms

Decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors

- "distance vector" algorithms

Static or dynamic?

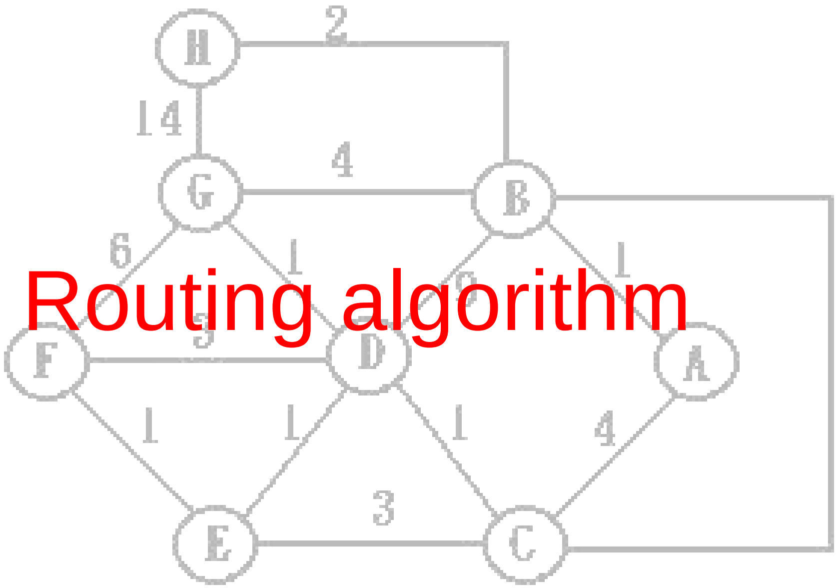
Static:

- routes change slowly over time

Dynamic:

- routes change more quickly
  - periodic update
  - in response to link cost changes

Link state Routing algorithm



A Link-State Routing Algorithm

Dijkstra's algorithm

- net topology, link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ('source") to all other nodes
  - gives forwarding table for that node
- iterative: after k iterations, know least cost path to k dest.'s

Notation:

- c(x,y): link cost from node x to y; $=\infty$ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

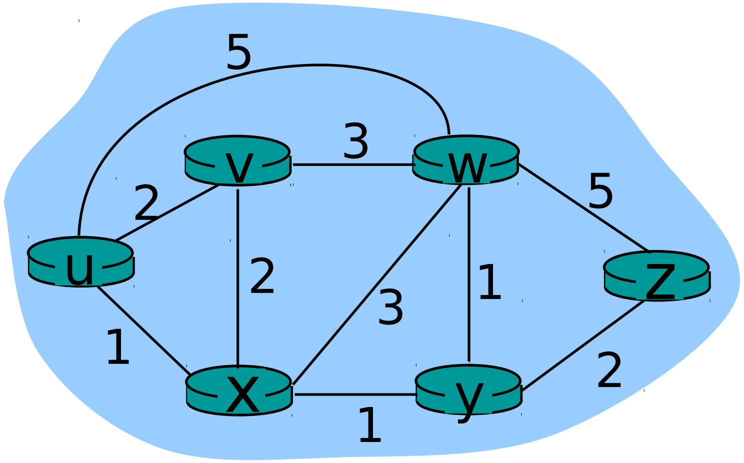
Dijsktra’s Algorithm

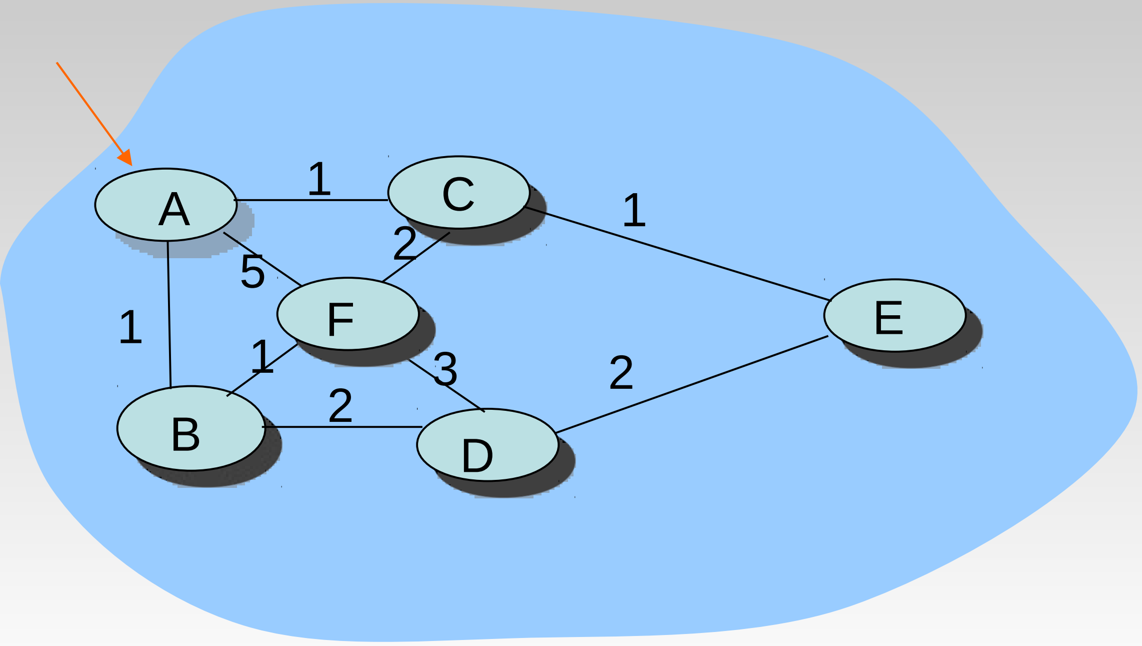
```
Initialization:
  N' = {u}
  for all nodes v
    if v adjacent to u
        then D(v) = c(u,v)
    else D(v) = ∞

Loop
  find w not in N' such that D(w) is a minimum
  add w to N'
  update D(v) for all v adjacent to w and not in N' :
      D(v) = min( D(v), D(w) + c(w,v) )
  /* new cost to v is either old cost to v or known
   shortest path cost to w plus cost from w to v */
until all nodes in N'
```

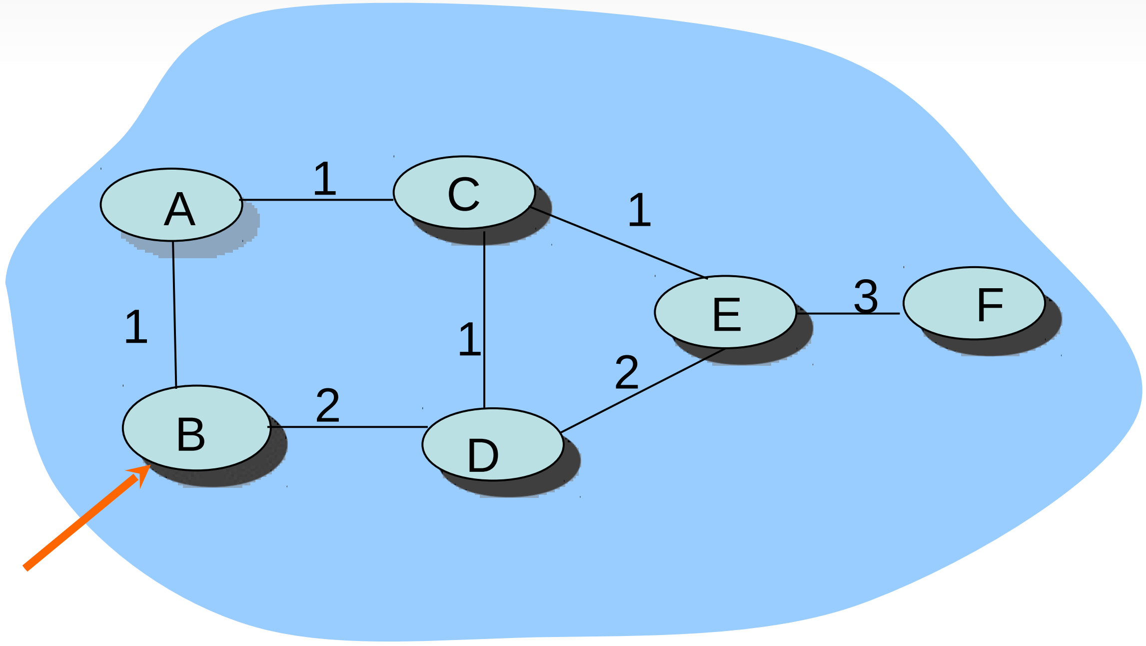
Dijkstra’s algorithm: example

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					





A
C
F
E
B
D
1
1
2
5
1
1
3
2
2



A
C
E
F
B
D
1
1
3
1
1
2
2

Hierarchical Routing

Our routing study thus far - idealization

- all routers identical
- network “flat”

… *not* true in practice

scale: with 200 million destinations:

- can’t store all dest’s in routing tables!
- routing table exchange would swamp links!

administrative autonomy

- internet = network of networks
- each network admin may want to control routing in its own network

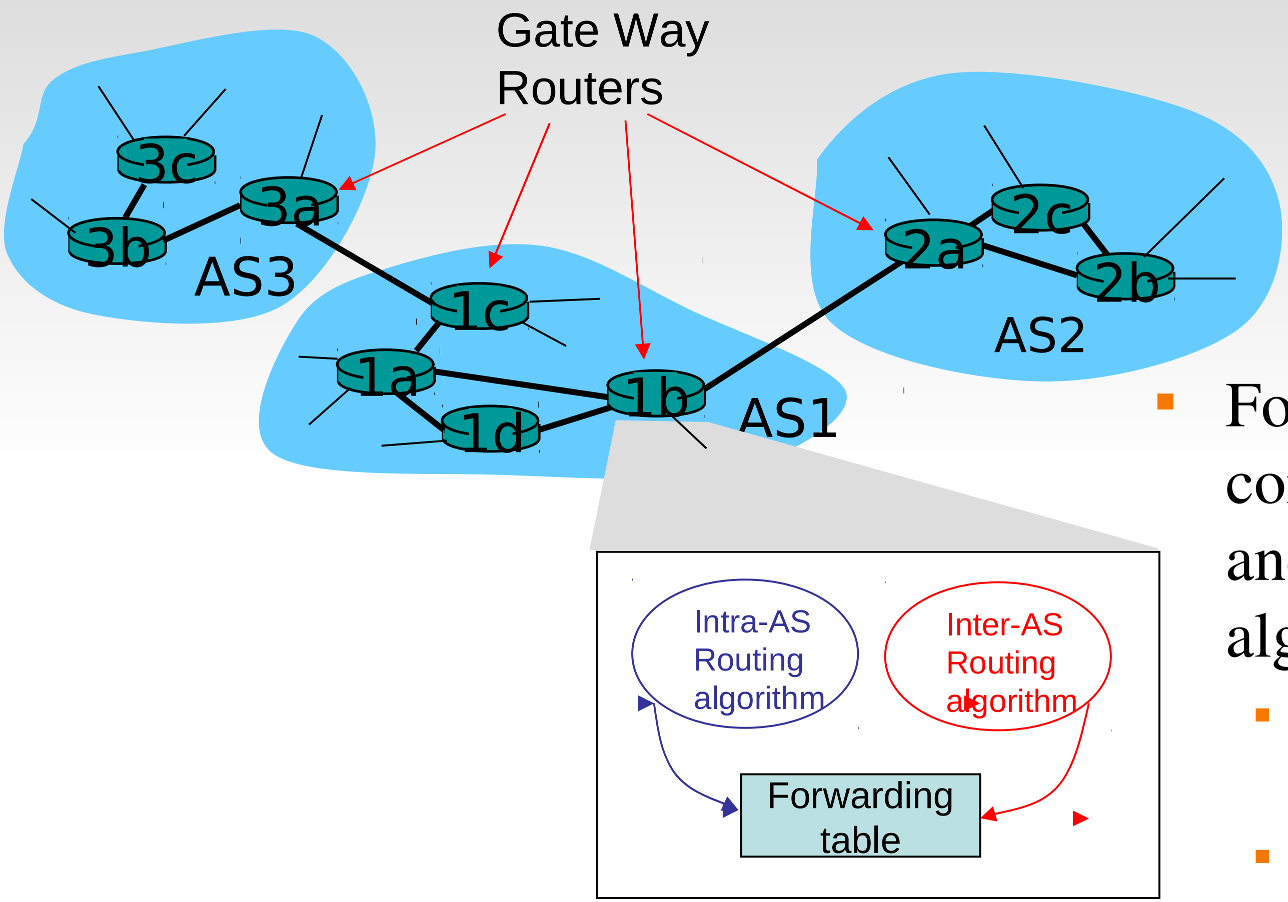
Hierarchical Routing

- aggregate routers into regions, "autonomous systems" (AS)
- routers in same AS run same routing protocol
  - "intra-AS" routing protocol
    - routers in different AS can run different intra-AS routing protocol

Gateway router

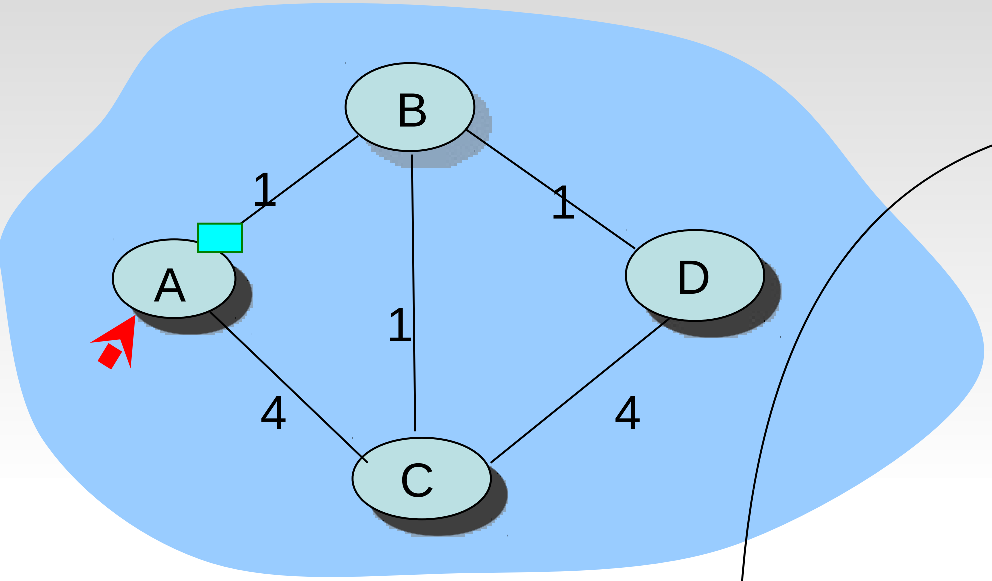
- Direct link to router in another AS

Interconnected AS's



- Forwarding table is configured by both intra- and inter-AS routing algorithm
  - Intra -AS sets entries for internal dests
  - Inter-AS & Intra-As sets entries for external dests

Distance Vector Algorithm



	A	B	C	D
A	0	1	4	∞
B	1	0	1	1
C	4	1	0	4
D	∞	1	4	0

$\boxed{AB = 1}$

+

When DV from B arrives 1 0 1 1

2 1 2 2

0 1 4 ∞

0 1 2 2

Count to Infinity Problem

Refer the class note.

Comparison of LS and DV algorithms

Message complexity

- LS: with n nodes, E links, O(nE) msgs sent
- DV: exchange between neighbors only
  - convergence time varies

Speed of Convergence

- LS: $O(n^2)$ algorithm requires O(nE) msgs
  - may have oscillations
- DV: convergence time varies
  - may be routing loops
  - count-to-infinity problem

Robustness: what happens if router malfunctions?

LS:

- node can advertise incorrect *link* cost
- each node computes only its *own* table

DV:

- DV node can advertise incorrect *path* cost
- each node's table used by others
  - error propagate thru network

Other Routing Algorithms

- Flooding
- Hot Potato Routing
- Source path Routing

ICMP: Internet Control Message Protocol

- **used by hosts & routers to communicate 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 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	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Traceroute and ICMP

- Source sends series of UDP segments to dest
  - First has TTL =1
  - Second has TTL=2, etc.
  - Unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram
  - And sends to source an ICMP message (type 11, code 0)
  - Message includes name of router& IP address
- When ICMP message arrives, source calculates RTT

Stopping criterion

- UDP segment eventually arrives at destination host
- Destination returns ICMP
- When source gets this ICMP, stops.