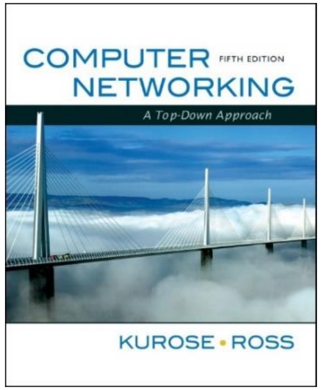


Computer Networks: Network Layer Protocols

Rajesh Palit, Ph.D.

North South University, Dhaka

Chapter 4: Network Layer

4.1 Introduction

4.2 Packet forwarding

4.3 What’s inside a router

4.4 IP: Internet Protocol

- Datagram format
- DHCP/NAT
- ICMP
- IPv6

4.5 Routing algorithms

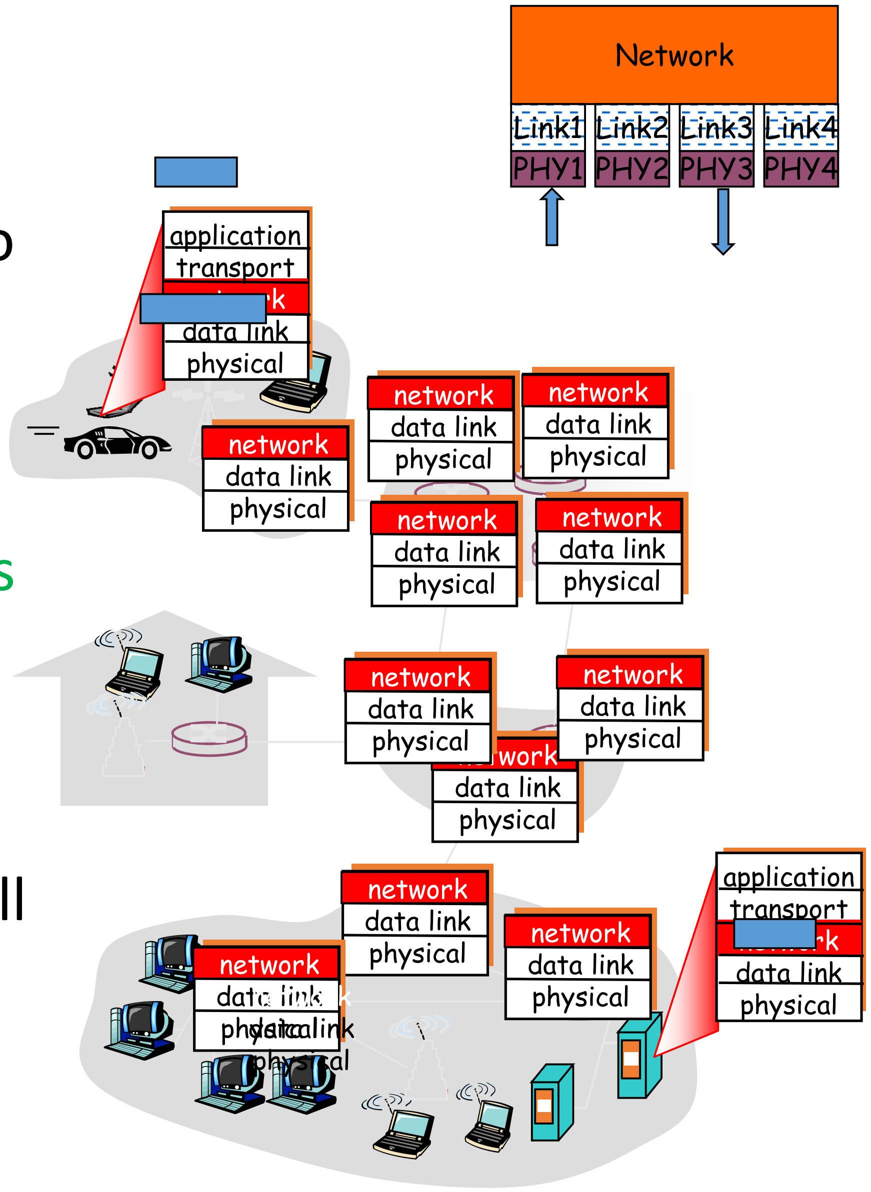
- Distance Vector
- Link state
- Hierarchical routing

4.6 Routing in the Internet

- RIP
- OSPF
- BGP

Network layer

- transport segment from sending to receiving host
- on sending side **encapsulate** segments into datagrams
- on receiving side, deliver segments to transport layer
- network layer protocols in *every* host, router
- router **examines** header fields in all IP datagrams passing through it



Two key network-layer functions

- *forwarding:* move packets from router's input to appropriate router output

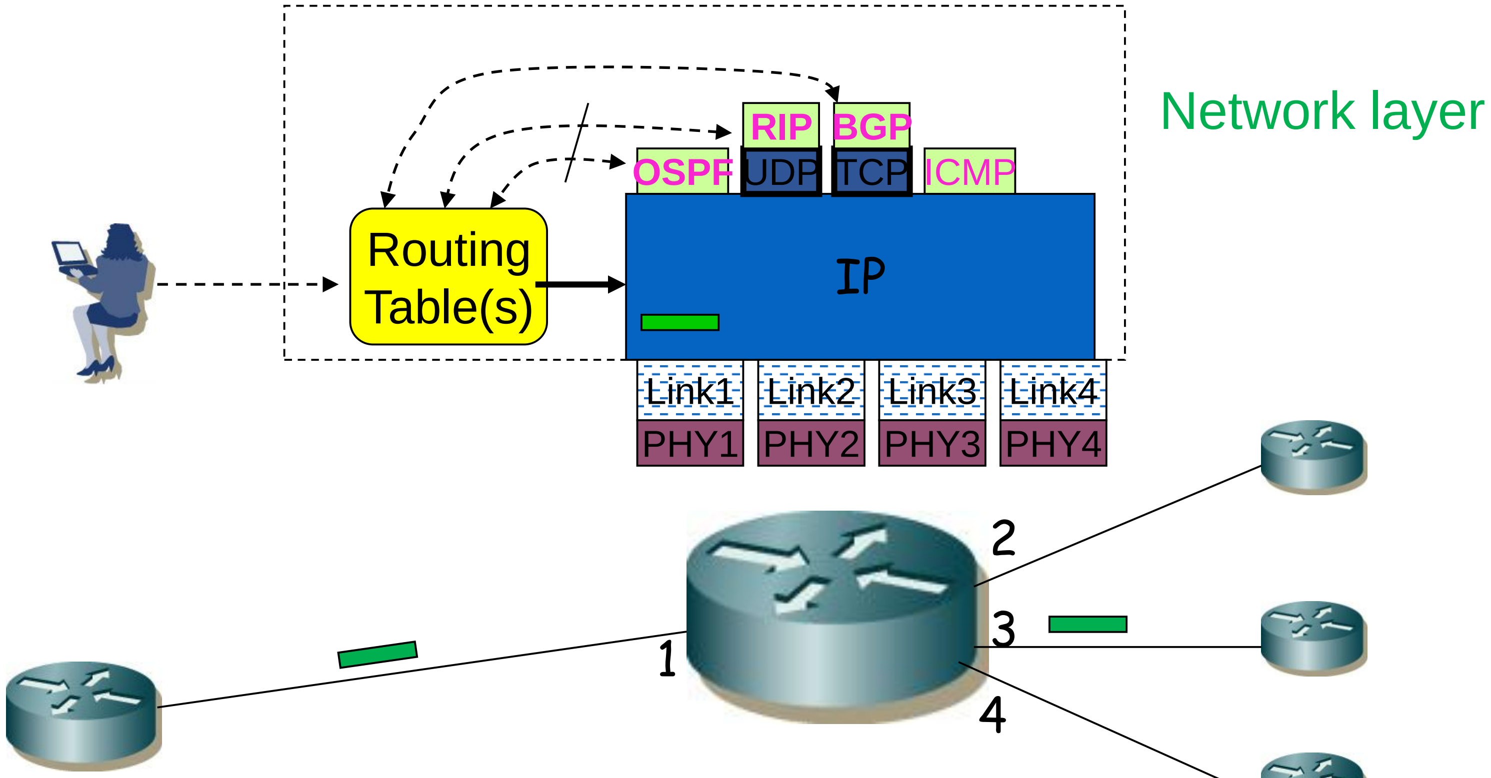
- *routing:* determine route taken by packets from source to dest.
  - *routing algorithms*

analogy:

- *routing:* process of planning trip from source to dest

- *forwarding:* process of getting through single interchange

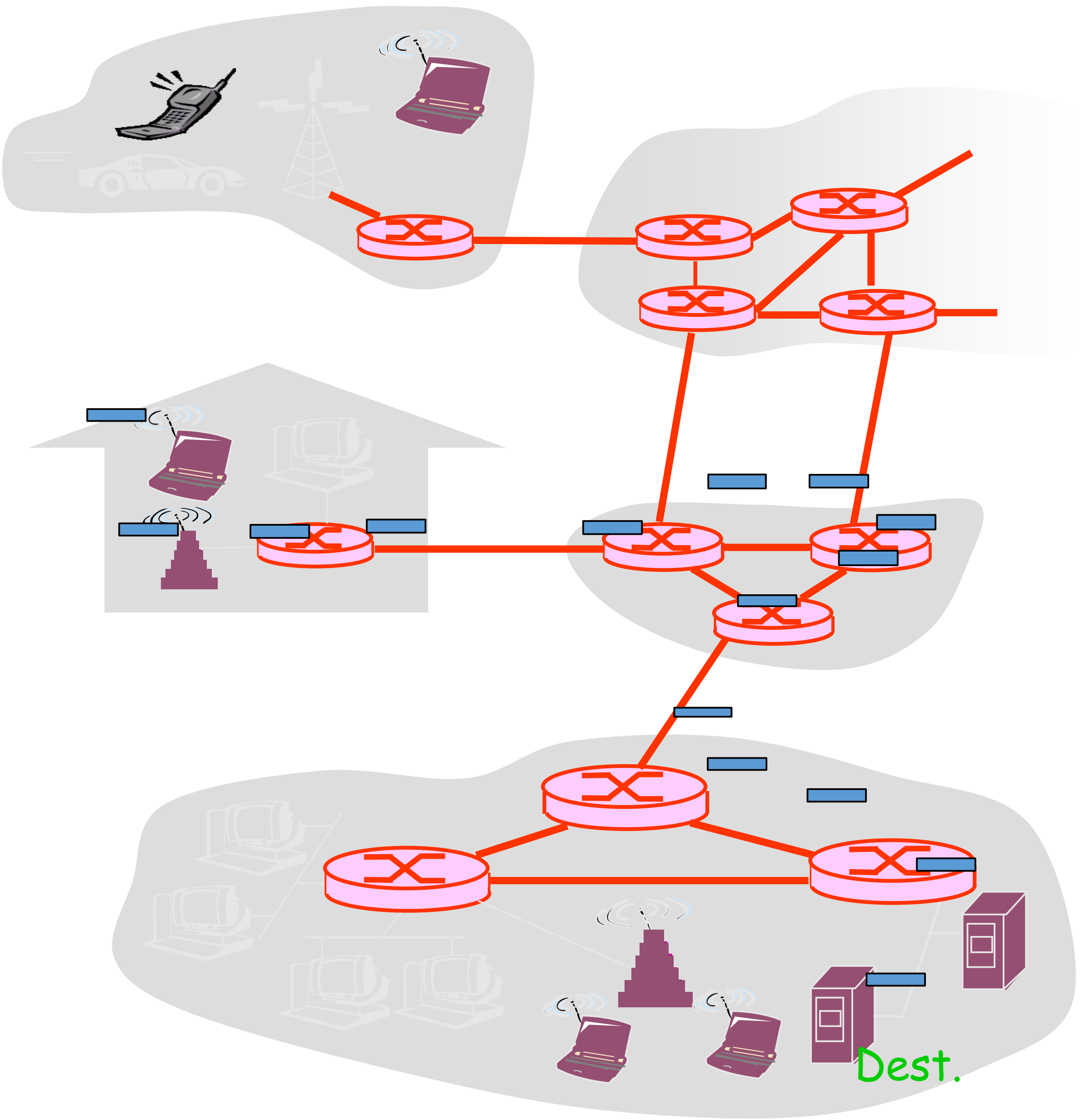
Forwarding and Routing



OSPF: Open Shortest Path First
RIP: Routing Information Protocol
BGP: Border Gateway Protocol
ICMP: Internet Control Message Protocol

TCP: Transmission Control Protocol
UDP: User Datagram Protocol

Hop-by-hop routing



IP address

In IPv4, an IP address is 32-bit long

Example: 10000001 01100001 01011100 00100101

This is also written as: 129.97.92.37

$(10000001)_2 = 129$
$(01100001)_2 = 97$
$(01011100)_2 = 92$
$(00100101)_2 = 37$

Dotted decimal notation
(easy to enter …..)

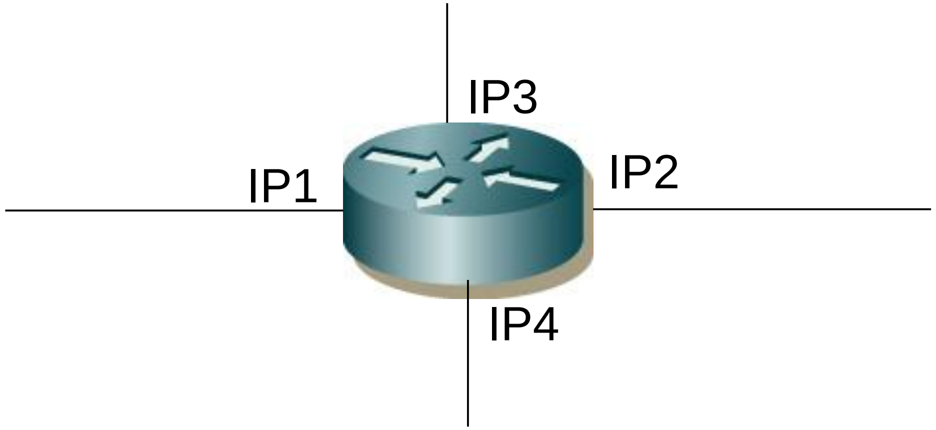
IPv6 uses a 128-bit address, allowing 2^{128}, or approximately 3.4×10^{38} addresses, or more than 7.9×10^{28} times as many as IPv4, which uses 32-bit addresses and provides approximately 4.3 billion addresses.

IP address

Switches and hubs do not have IP addresses

End-devices generally have one IP address each …..
(Laptops, desktops, servers, … have one IP address each)

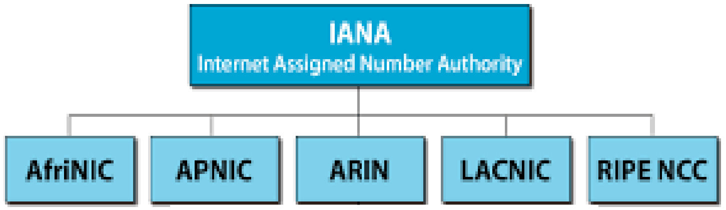
Routers have multiple IP addresses … one+ for each physical interface …



Who gives ISPs IP address blocks?

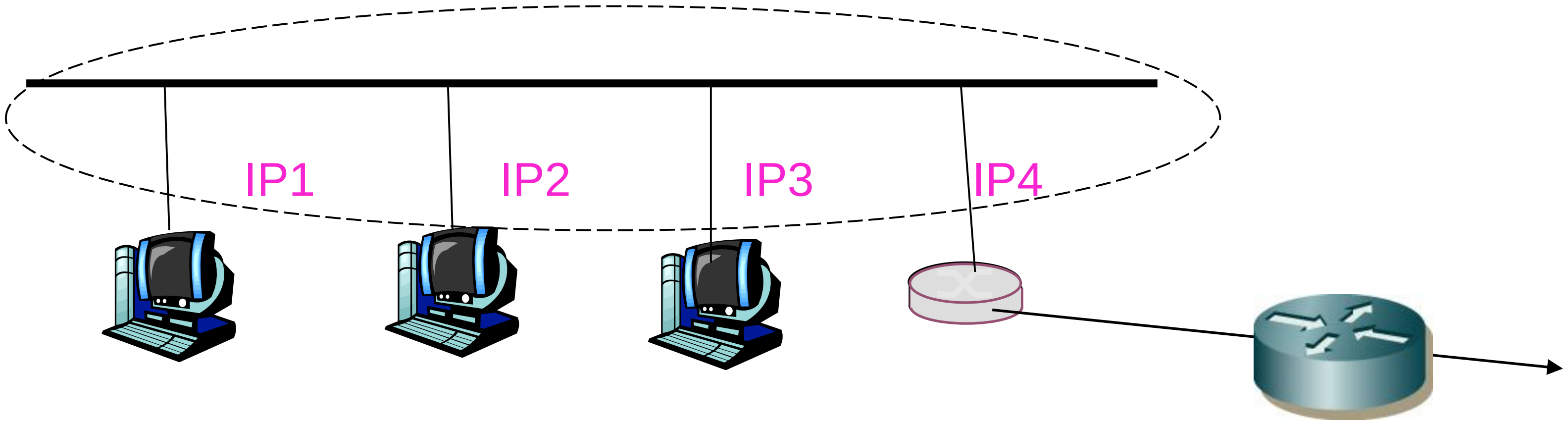
ICANN allocates IP address blocks to regional internet registries

- AFRINIC: African Registry for Network Info Centre (Mauritius)
- APNIC: Asia-Pacific Network Information Centre (South Brisbane)
- ARIN: American Registry for Internet Numbers (Virginia)
- LACNIC: Latin American and Caribbean Network Info. Centre (Uruguay)
- RIPE: Réseaux IP Européens Network Coordination Centre (Amsterdam)



ICANN: Internet Corp. for Assigned Names and Numbers

The concept of network prefix



All the IP addresses on the same link have a common portion in their most significant bits

IP1: x.y.z.1 0 1 0 1 0 **0 0**
IP2: x.y.z. 1 0 1 0 1 0 **0 1**
IP3: x.y.z. 1 0 1 0 1 0 **1 0**
IP4: x.y.z. 1 0 1 0 1 0 **1 1**

common portion

Network prefix　　host ID

The concept of network prefix

IP1: x.y.z.1 0 1 0 1 0 **0 0**
IP2: x.y.z. 1 0 1 0 1 0 **0 1**
IP3: x.y.z. 1 0 1 0 1 0 **1 0**
IP4: x.y.z. 1 0 1 0 1 0 **1 1**

Network prefix host ID

Remember:
Routers do **not** store routing info. for individual destination IPs…

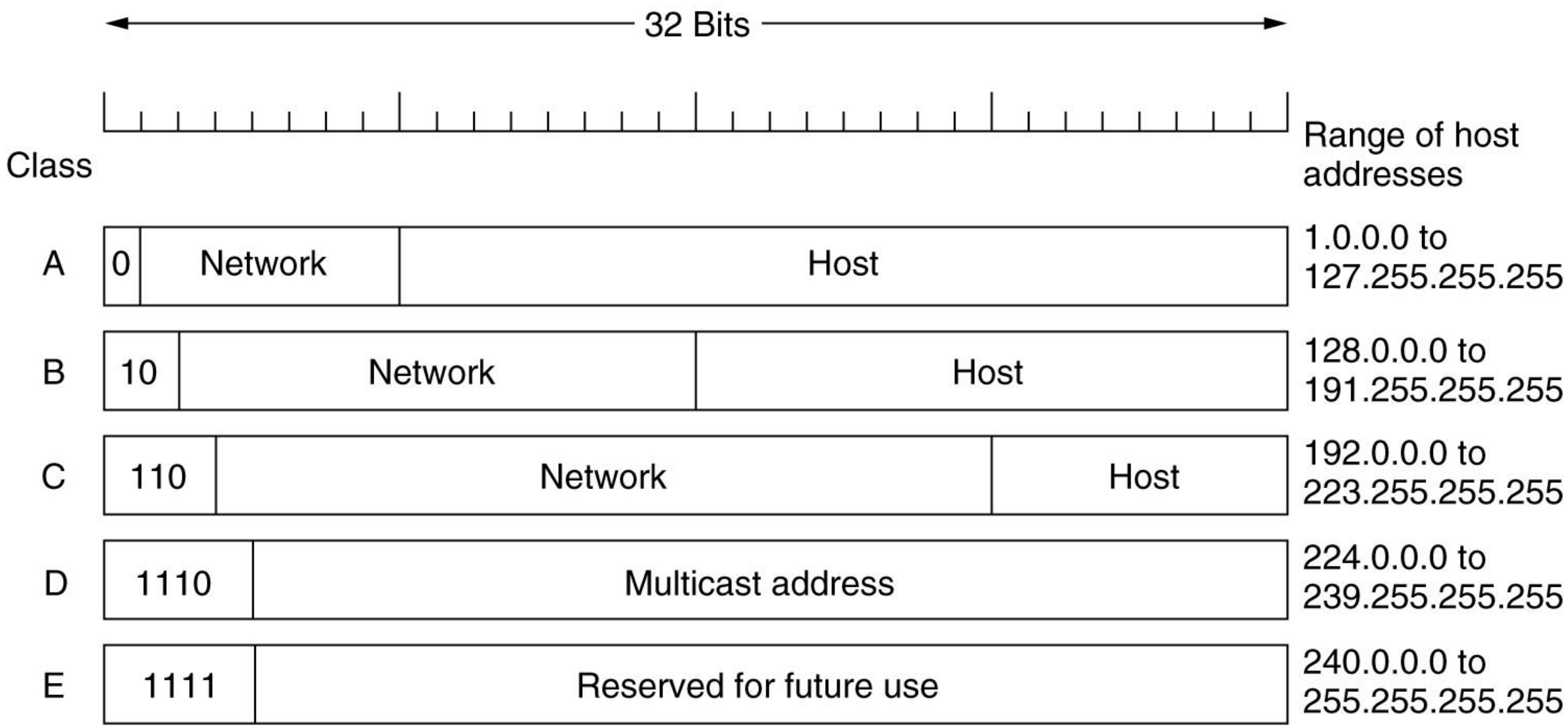
(there are billions of IP addresses….. storing and searching all those IP addresses will need much more CPU power and memory…..

Instead, routers store aggregated IP addresses in their routing tables……

Example: x.y.z.168/30

$(10101000)_2 = 168$
30 = length of the network prefix in bits

IP Addresses (Classful)



IP Addresses (Classful)

Special IP addresses

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	This host
0 0 . . . 0 0 | Host	A host on this network
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Broadcast on the local network
Network | 1 1 1 1 . . . 1 1 1 1	Broadcast on a distant network
127 | (Anything)	Loopback
Network | 0 ….. 0 0 0 0 0 0 0	

Private IP addresses (RFC 1918)

10.0.0.0	– 10.255.255.255/8	(16,777,216 hosts)
172.16.0.0	– 172.31.255.255/12	(1,048,576 hosts)
192.168.0.0	– 192.168.255.255/16	(65,536 hosts)

10.0.0.0/8: Valid IP addresses are 10.0.0.1 -- 10.255.255.254.

172.16.0.0/12: Valid IP addresses are 172.16.0.1 -- 172.31.255.254.

192.168.0.0/16: Valid IP addresses are 192.168.0.1 -- 192.168.255.254.

Note

UW uses the **172.16.0.0/12** block

Public IP addr vs. Private IP addr

Public IP addresses

- These addresses are globally unique.
- Routers everywhere recognize these.
- Any host can open a TCP connection with a machine with a global IP address

Private IP addresses

- These are not globally unique. (Unique within an org.)
- Routers outside the org. do not recognize these.
- If a host has a private IP address, a host outside the org. **cannot** open a TCP conn with it.

Adv. — Reuse of IP addresses
Added security

IP addressing: CIDR

CIDR: Classless Inter-Domain Routing

- subnet portion (prefix portion) of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address

subnet part | host part

11001000 00010111 00010000 00000000

200.23.16.0/23

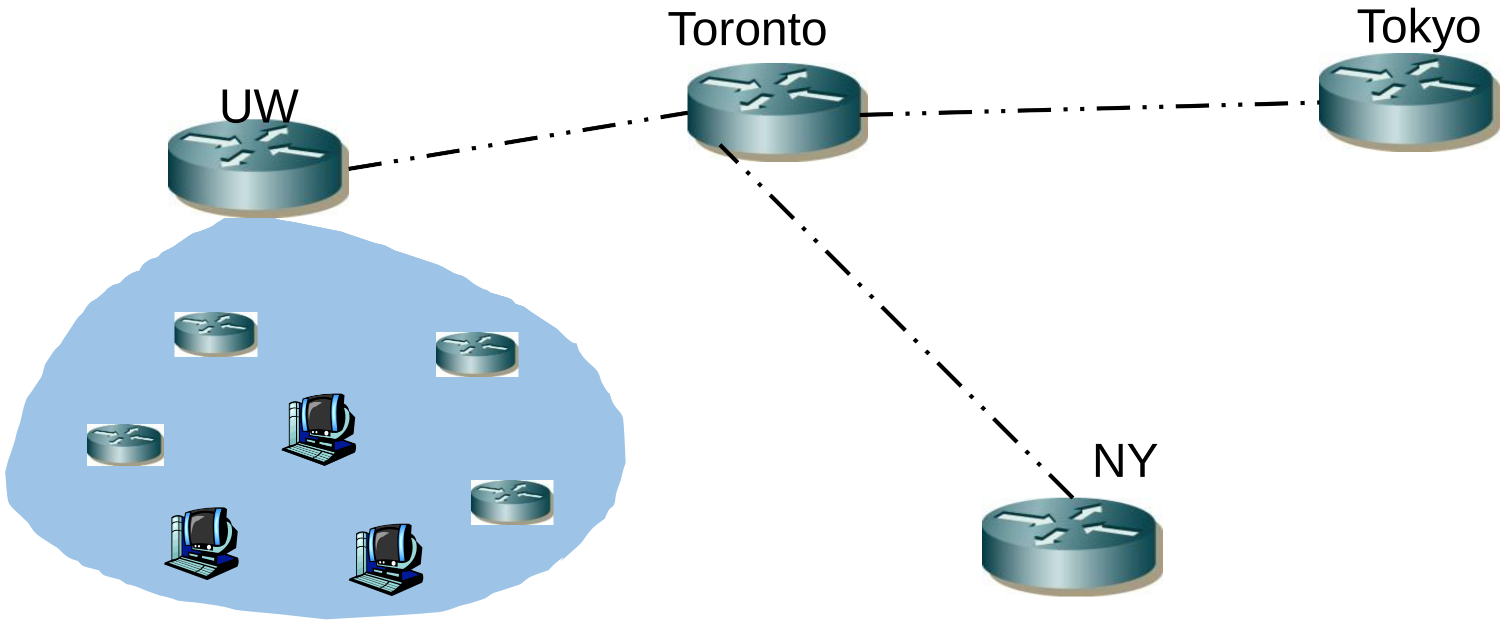
The concept of network prefix

#1: Net Prefix

#2: Autonomous Systems

The concept of network prefix is very simple,
yet it is a powerful concept that makes the Internet scalable…..

Routers in Toronto, Tokyo, New York, … know all UWaterloo hosts by 129.97.0.0/16. The whole UW is one dest.



The concept of network prefix

For discussion purpose, we use the notation 129.97.0.0/16.

However, routers use the following notation:

Dest. Address: 129.97.0.0

Network Mask: 11111111.11111111.00000000.00000000

: 255.255.0.0

Destination address: 129.97.0.0/ 255.255.0.0

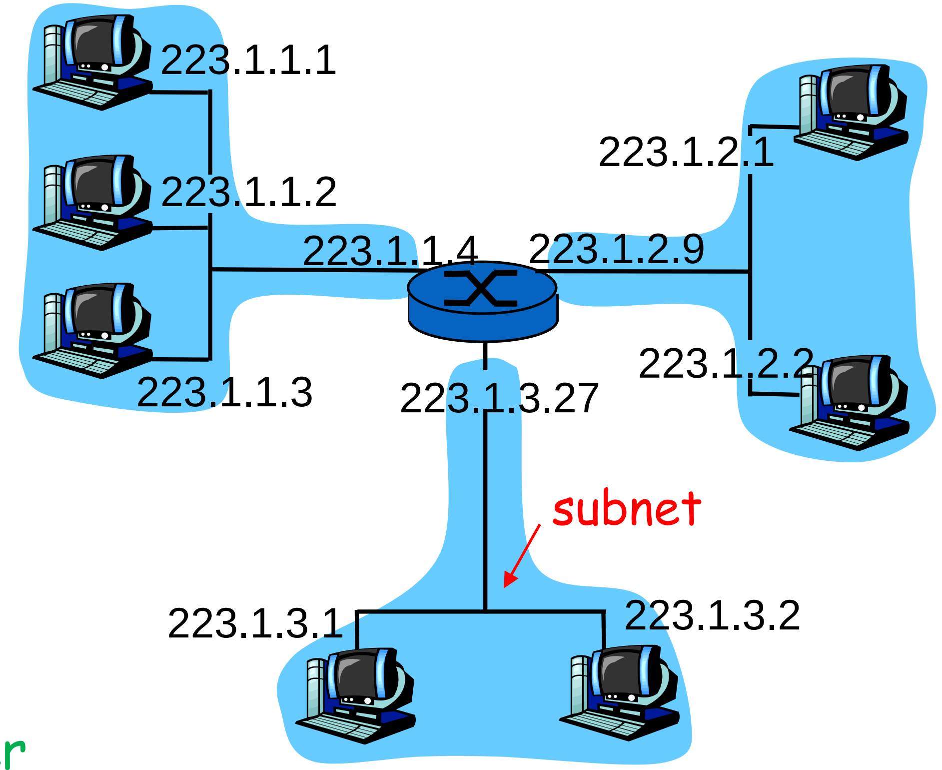
Subnets

- IP address:
  - subnet part (high order bits)
  (Recall: Network Prefix)
  - host part (low order bits)

What's a subnet ?

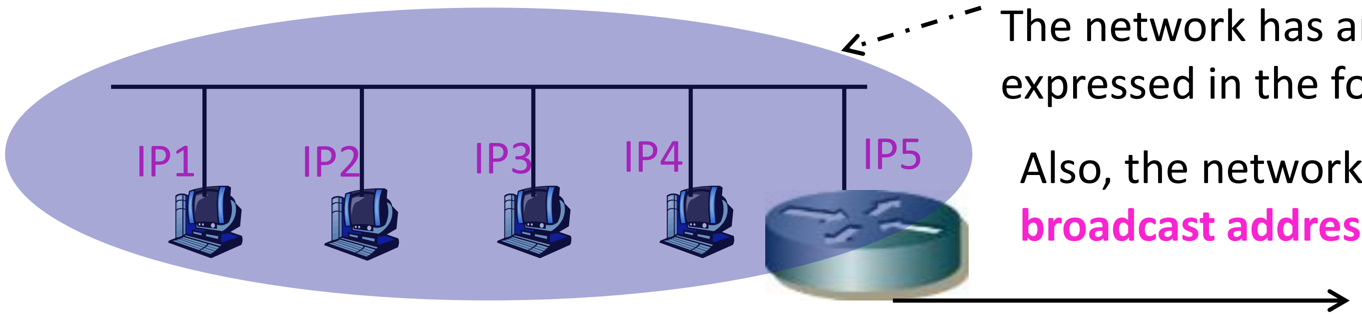
device interfaces with same
subnet part of IP address

can physically reach each other
without intervening router



network consisting of 3 subnets

The concepts of network ID and broadcast address



The network has an ID called **Network ID**, expressed in the form of an IP address.

Also, the network has a **broadcast address**.

Consider the network 10.2.5.16/28

/28 = 11111111 . 11111111 . 11111111 . 11110000 = 255.255.255.240

The **FIRST** addr in the net 10.2.5.16/28 is 10.2.5.00010000 = 10.2.5.16 — **Net ID**

The next addr in the net 10.2.5.16/28 is 10.2.5.00010001 = 10.2.5.17 — **Assign to Router (con.)**

The next addr in the net 10.2.5.16/28 is 10.2.5.00010010 = 10.2.5.18

: : : : : :

The next addr in the net 10.2.5.16/28 is 10.2.5.00011110 = 10.2.5.30 — **Assign to hosts**

The **LAST** addr in the net 10.2.5.16/28 is 10.2.5.00011111 = 10.2.5.31 — **Broadcast addr**

The concepts of network ID and broadcast address

- **Network ID** appears in routing tables.
  - Individual host IP addresses do NOT.

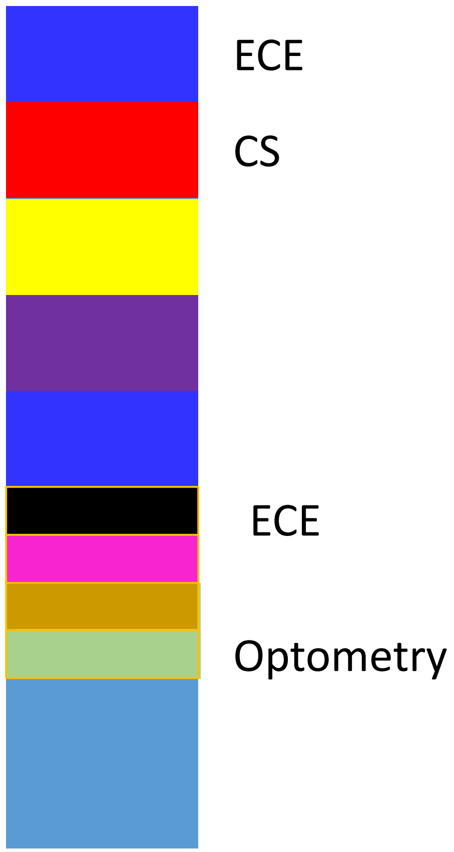
- Broadcast Address is used to perform IP-level broadcast.
  - You send an IP packet with a Broadcast Addr as the Destination, the IP packet is delivered to ALL the nodes on the network.

Net ID and Broadcast address are NOT assigned to any host/router.

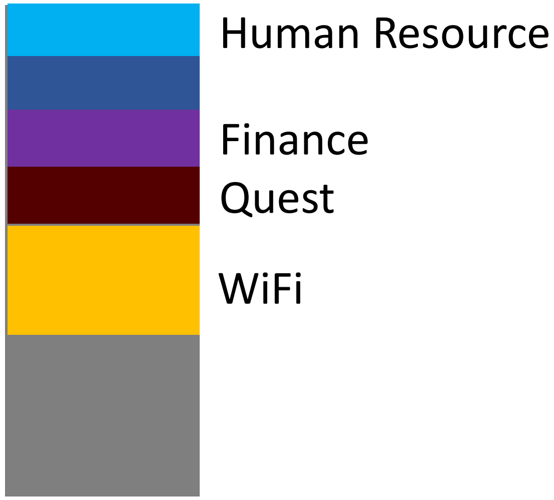
Partitioning an IP address block into different networks

An ISP (UW) gets a block of **public** IP addresses (129.97.0.0/16) from IANA/ARIN

Public IP address space



Private IP address space

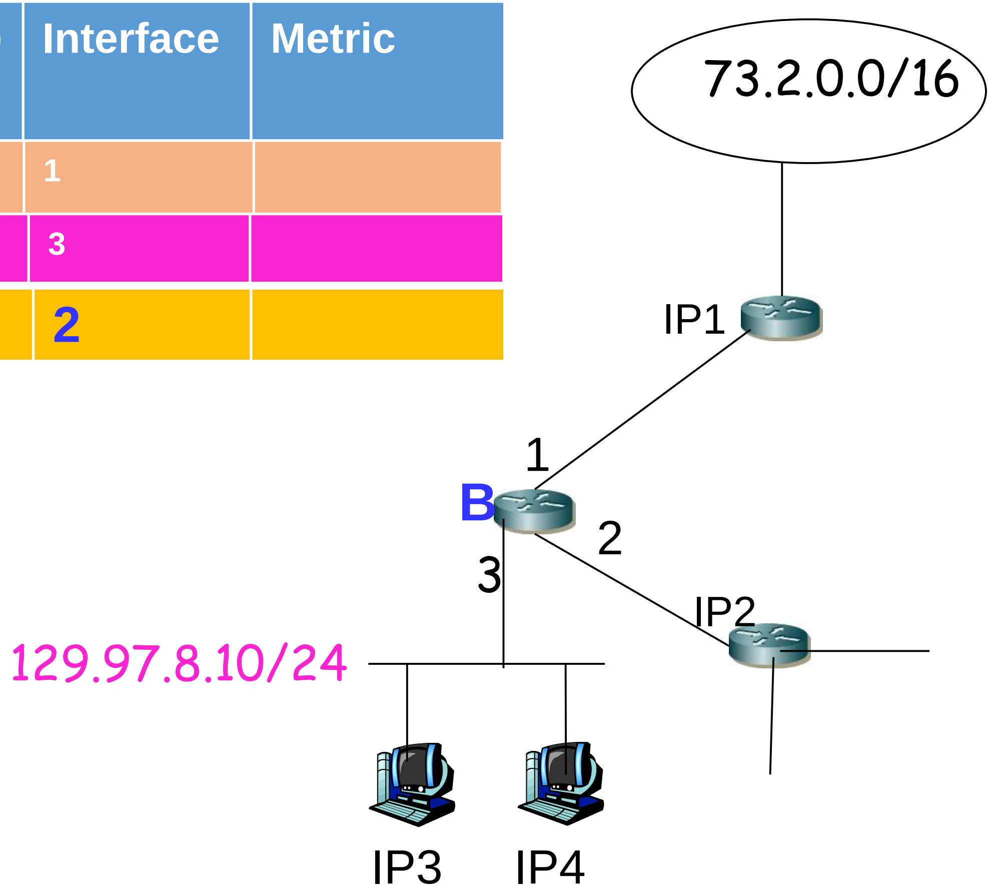


Simple structure of a routing table (at B): Example

	Dest. Address	Mask	Next hop	Interface	Metric
*	73.2.0.0	255.255.0.0	IP1	1	
	129.97.8.10	255.255.255.0	"connected"	3	
	0.0.0.0	0.0.0.0	IP2	2	

Default entry (configure it)

A network connected to the router (configure it)



* **Learn** this entry by running routing protocols.

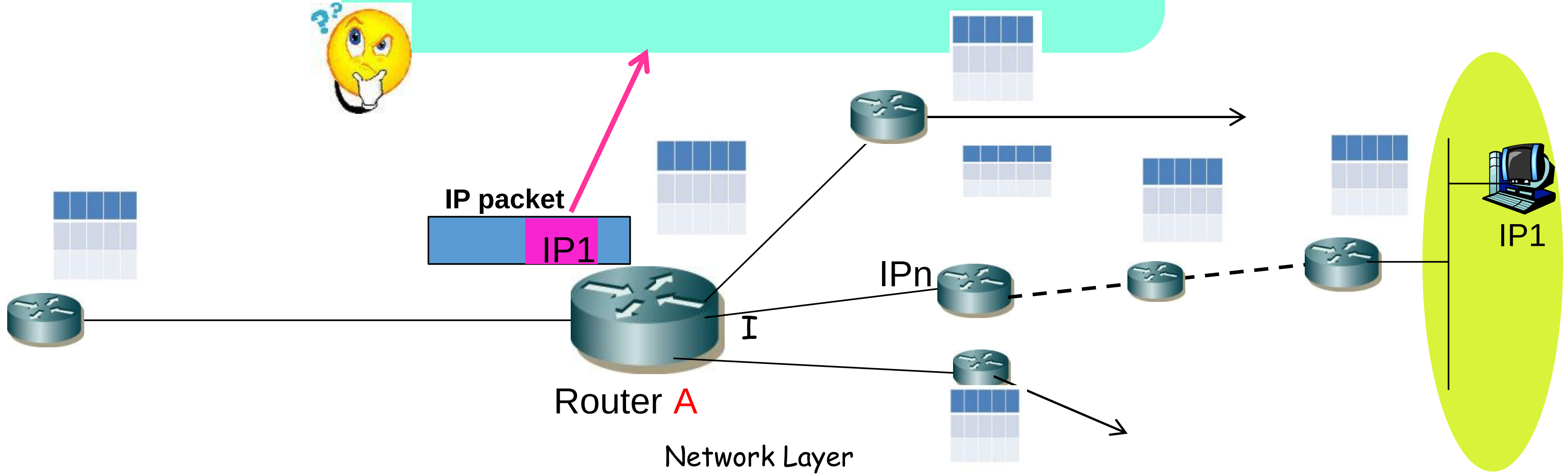
How does a router choose the next hop for a packet….

Address	Mask	Next hop	Interface	Metric
x.y.z.w	a.b.c.d	IPn	I	m

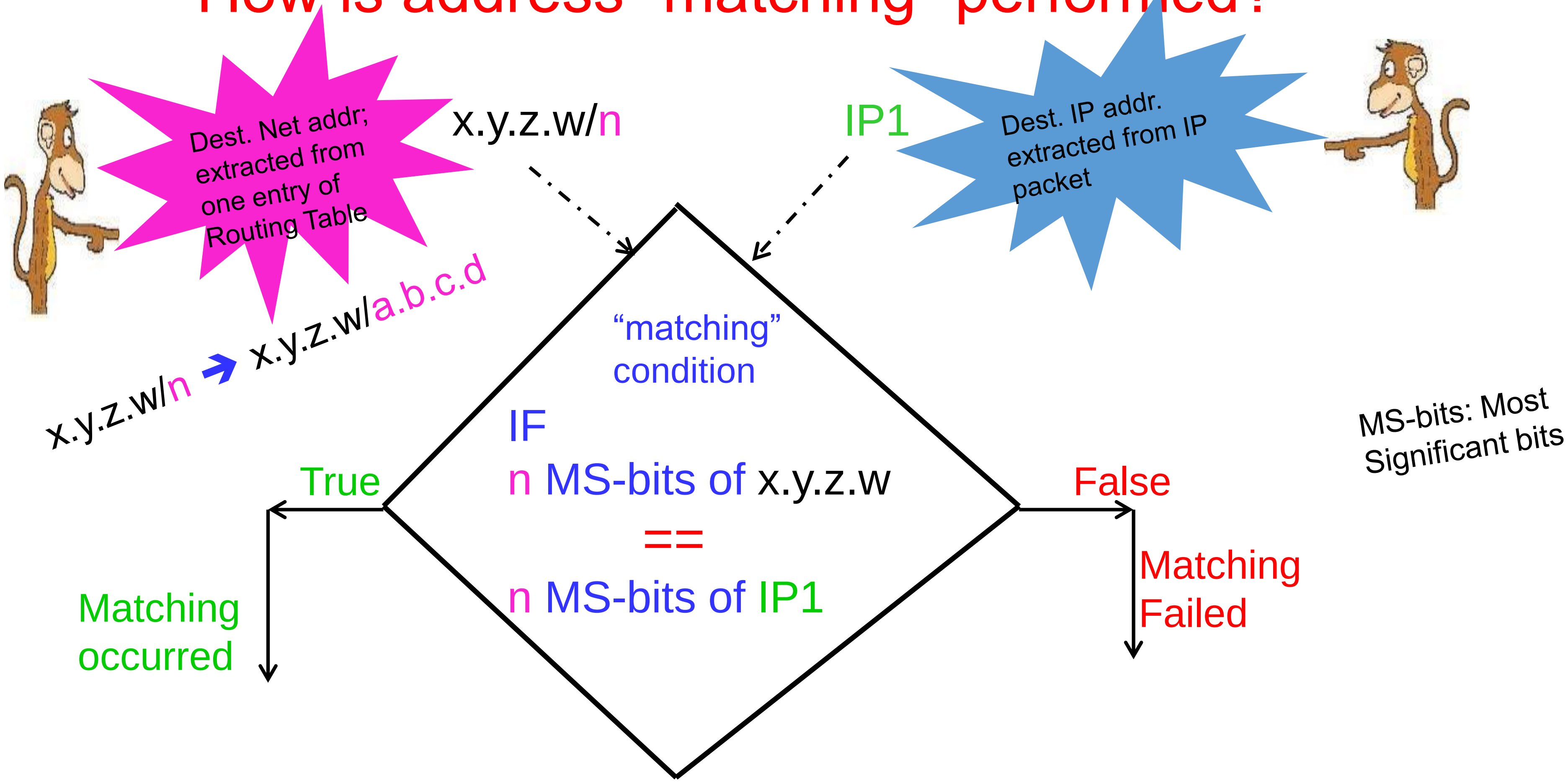
Routing Table of A

Next hop?

If dest. addr (IP1) “matches” with an entry in the RT, choose interface I.



How is address "matching" performed?



If ((x.y.z.w **AND** a.b.c.d) == (IP1 **AND** a.b.c.d)), **matching occurs**

For an RT and an IP address, many entries may match

Destination Address	Interface #
11001000 00010111 00010/21	0
11001000 00010111 00011000/24	1
11001000 00010111 00011/21	2
Otherwise (default): **0.0.0.0/0**	3

Examples:

IP1: 11001000 00010111 00010 110 10100001 matches with the **1st one.**

IP2: 11001000 00010111 00011 000 10101010

matches with the 2nd entry

matches with the 3rd entry

Note: If matching occurs, there is a "longest" prefix matching …..

Matching and forwarding algorithm

Inputs: IP address from packet header (call it IP1)

Routing Table (call it RT)

Processing:

```
if (matching occurs between IP1 and RT), {
        find the matching entry with the longest prefix
        forward the packet via the appropriate interface
}
else if (default entry exists in RT) { // default: 0.0.0.0/0
        forward the packet via the appropriate interface
}

else {
        send error message (ICMP message) to the source
        of the IP packet
}
```

ICMP: Internet Control Message Protocol

Chapter 4: Network Layer

4.1 Introduction

4.2 Packet forwarding

4.3 What's inside a router?

4.4 IP: Internet Protocol

- Datagram format
- DHCP/NAT
- ICMP
- IPv6

4.5 Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

4.6 Routing in the Internet

- RIP
- OSPF
- BGP

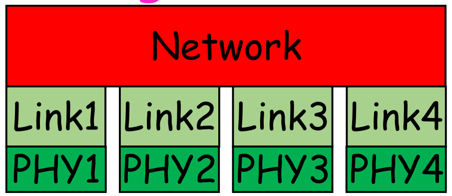
4.7 Broadcast and multicast routing

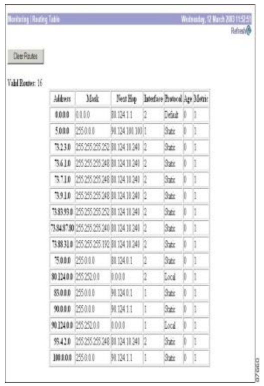
Router Architecture Overview

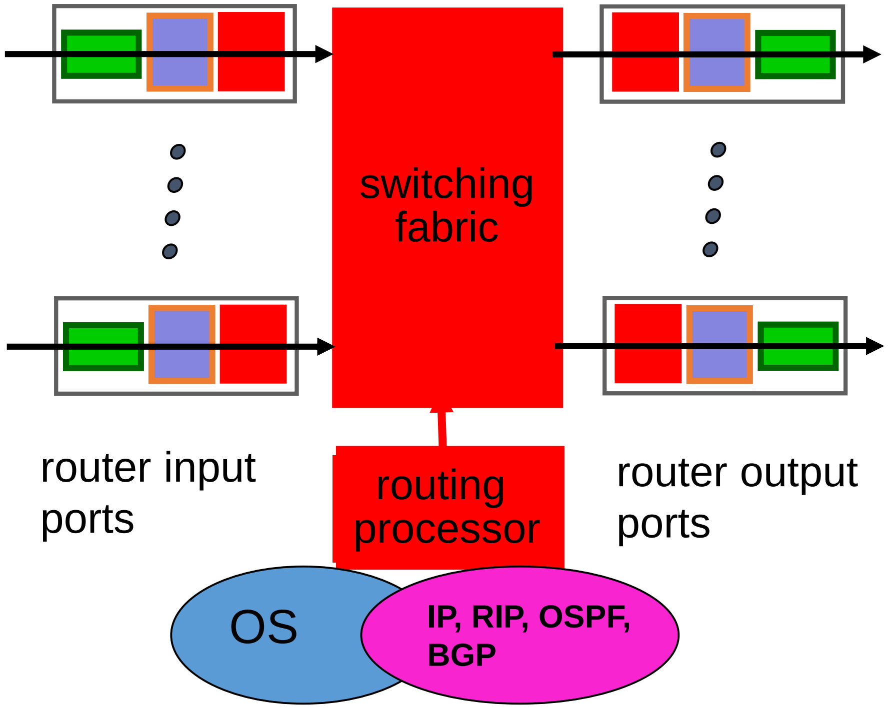
Two key router functions:

- run routing algorithms/protocols (RIP, OSPF, BGP)
- *forward* datagrams from incoming to outgoing links

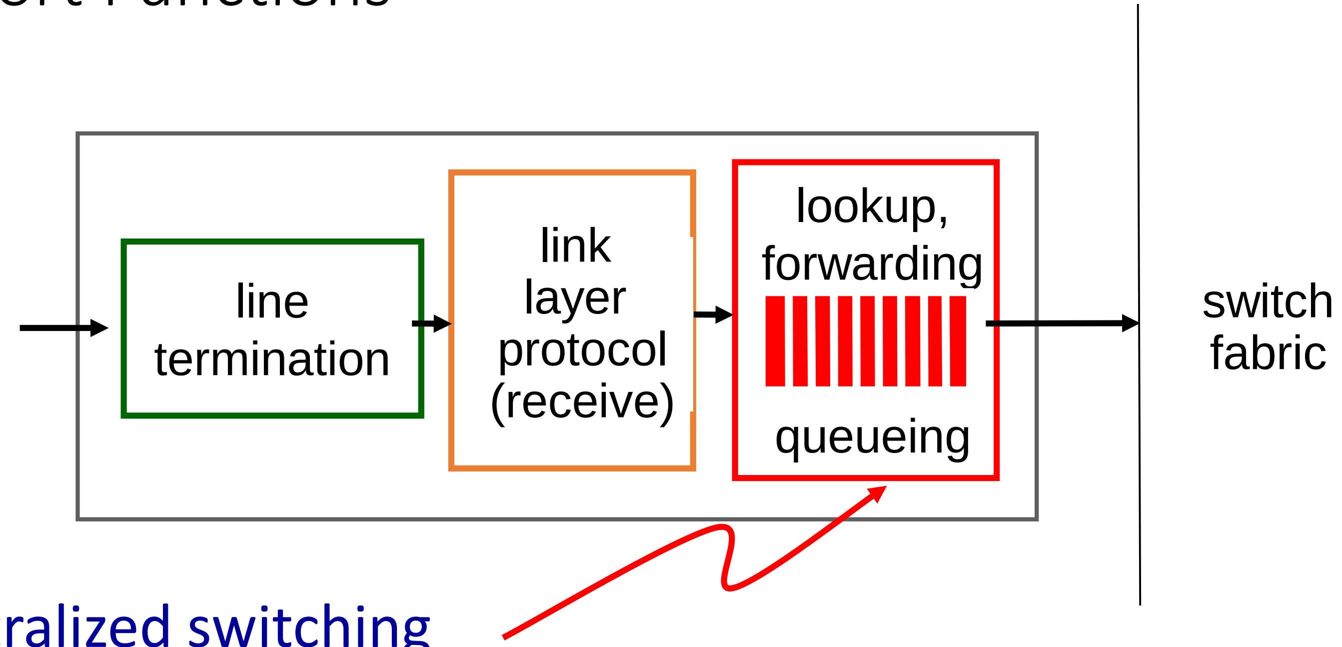
Logical view







Input Port Functions



Decentralized switching

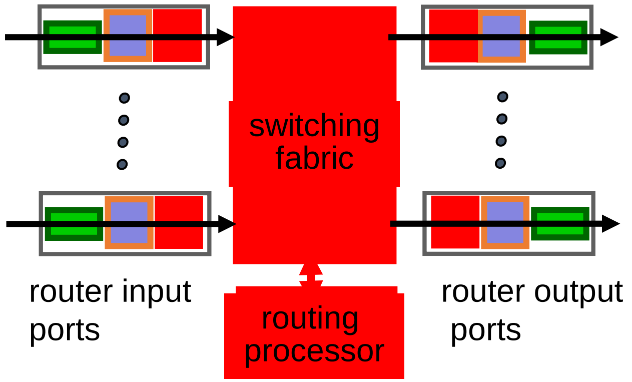
Given datagram dest. IP addr, lookup output port using RT

- Queuing occurs at both input ports and output ports

If fabric is slower than input ports combined, queueing may occur at input queues:

queueing delay and loss due to input buffer overflow!

Switching fabrics



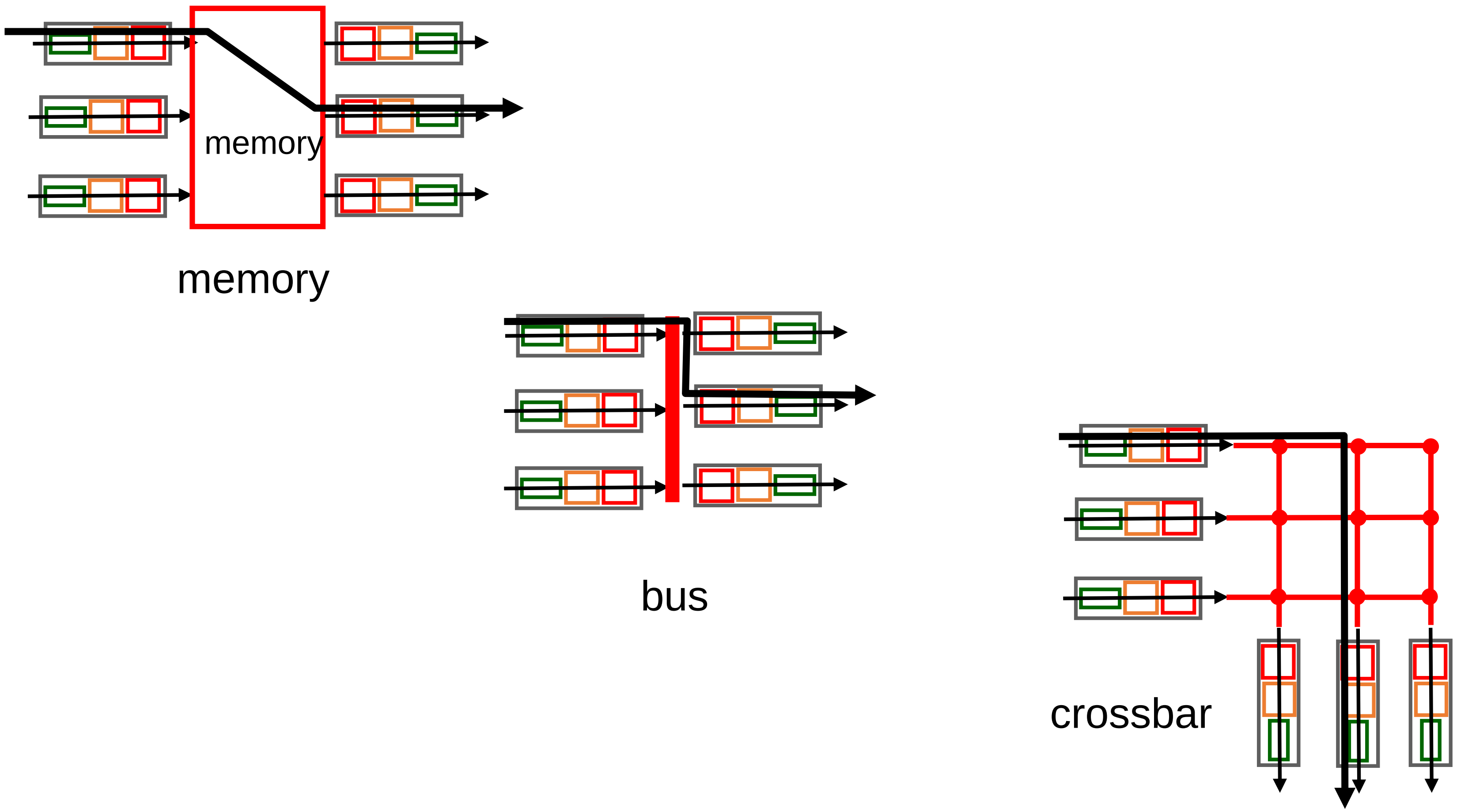
- transfer packet from input buffer to appropriate output buffer

❖ (Performance of switching fabric) switching rate: rate at which packets can be transferred from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: Ideally, switching rate is N times line rate

    Example: 4 input lines with 10 Gbps per line
    Desired switching rate: 4 x 10 Gbps = 40 Gbps

Switching fabrics

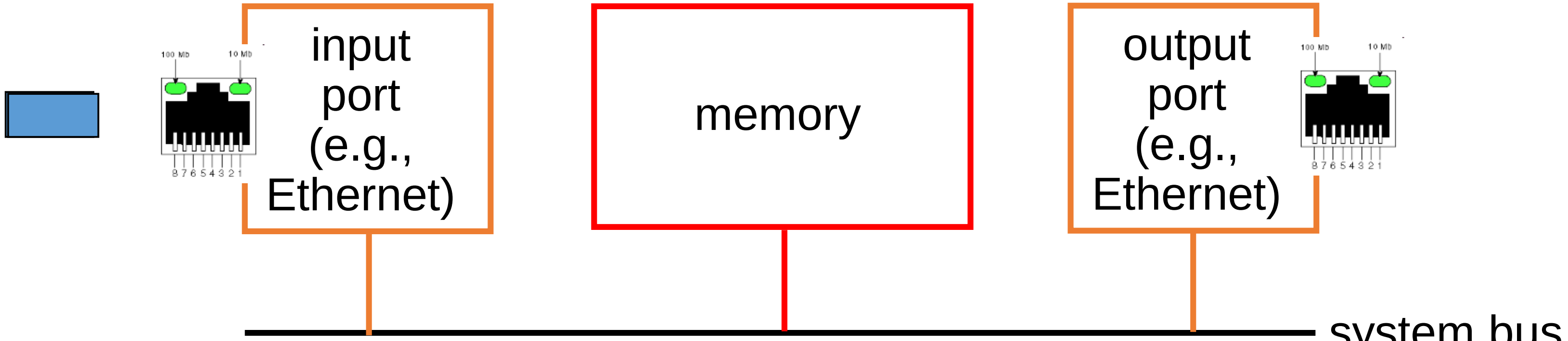
- Three types of switching fabrics



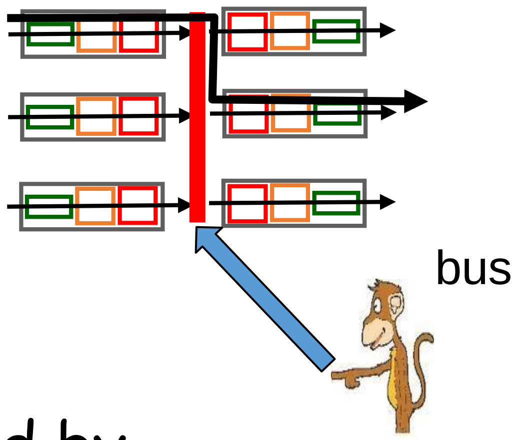
Switching Via Memory

First generation routers:

- traditional computers with switching under direct control of CPU
- packets are **copied** to system's memory
- **speed is limited** by memory bandwidth (2 bus crossings per datagram)



Switching Via a Bus



- datagram from input port memory
  to output port memory via a shared bus

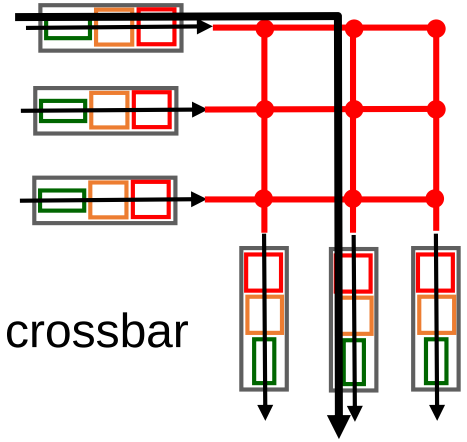
bus contention: switching speed limited by bus bandwidth

❖ Example:
  - Cisco 5600 router: 32 Gbps bus

Switching Via An Interconnection Network

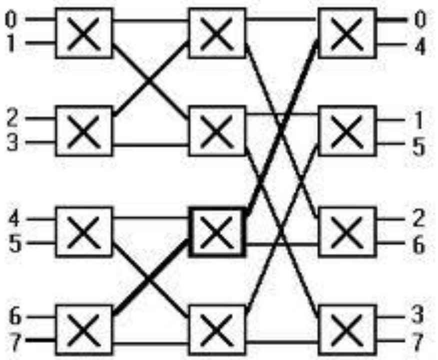
- Overcome bus bandwidth limitations

  ❖ Banyan networks and other interconnection nets initially developed for parallel processing



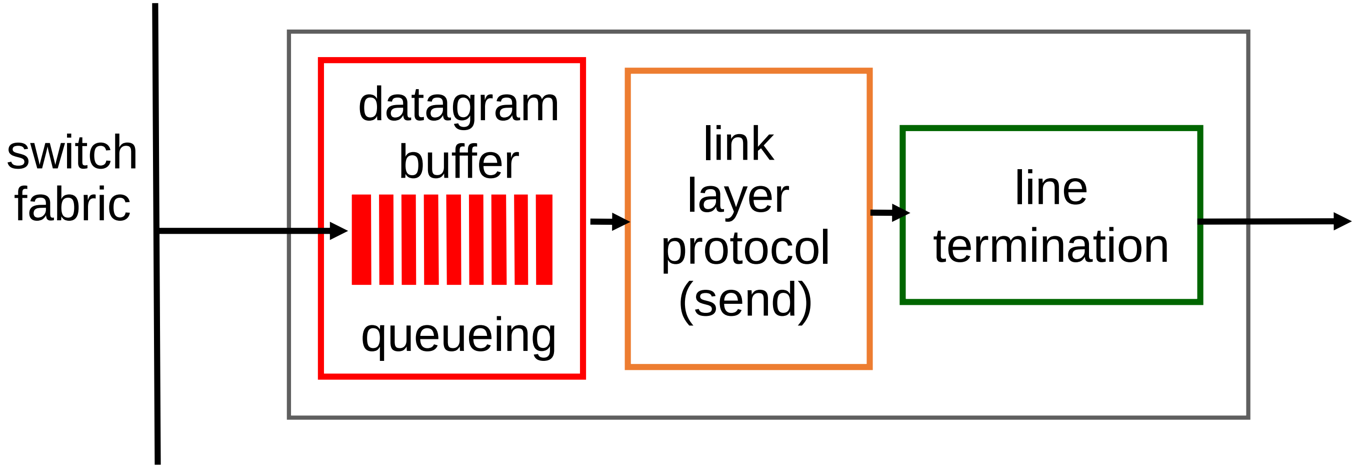
Example:

Cisco 12000 router: switches 60 Gbps through the interconnection network



An 8x8
banyan network

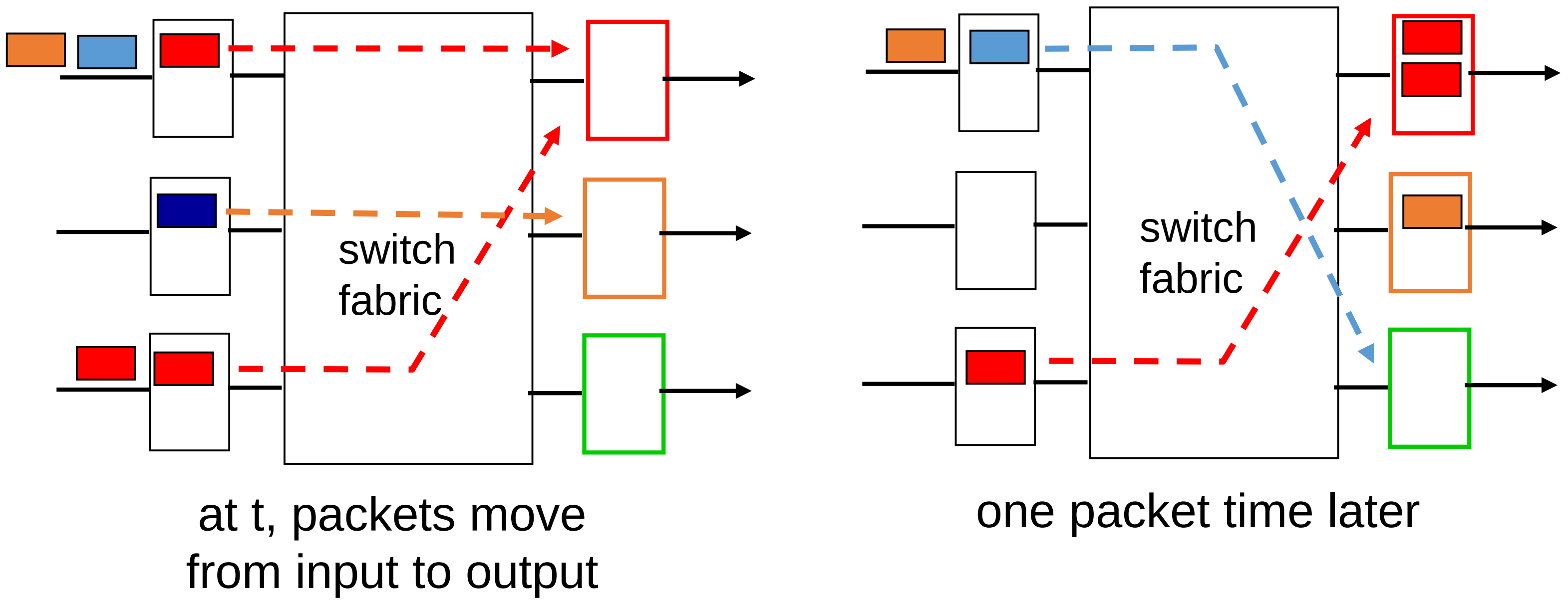
Output Ports Functions



- *buffering* required when datagrams arrive from fabric faster than the transmission rate

- *scheduling discipline* chooses among queued datagrams for transmission

Output port queueing



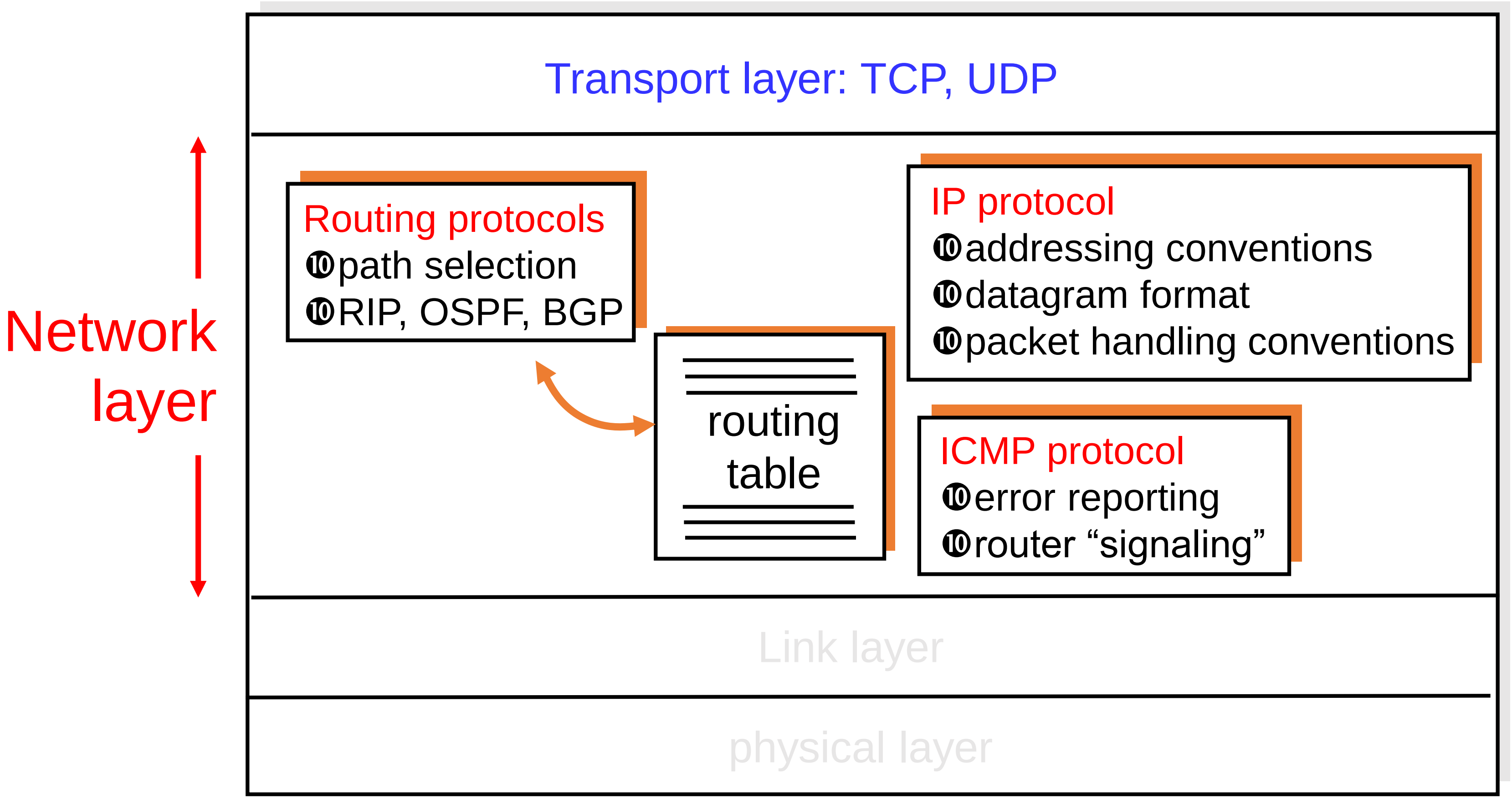
- buffering when arrival rate via switch exceeds output line speed
- queueing (delay) and loss due to output port buffer overflow!

Chapter 4: Network Layer

- 4.1 Introduction
- 4.2 Packet Forwarding
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - DHCP/NAT
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

The Internet Network layer

Host, router network layer functions:



Chapter 4: Network Layer

4.1 Introduction

4.2 Packet Forwarding

4.3 What’s inside a router

4.4 IP: Internet Protocol

- Datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 Routing algorithms

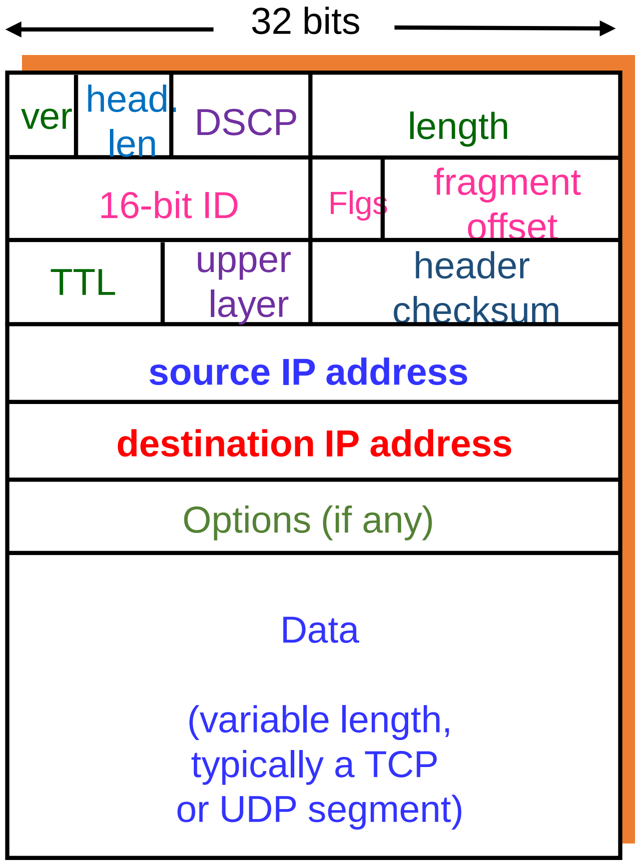
- Link state
- Distance Vector
- Hierarchical routing

4.6 Routing in the Internet

- RIP
- OSPF
- BGP

4.7 Broadcast and multicast routing

IP Packet Format



Version: 4

Header length: unit is 4-bytes
(A 20-byte header is rep. by 5.)

DSCP (Differentiated Services Code Point):
Type of data carried

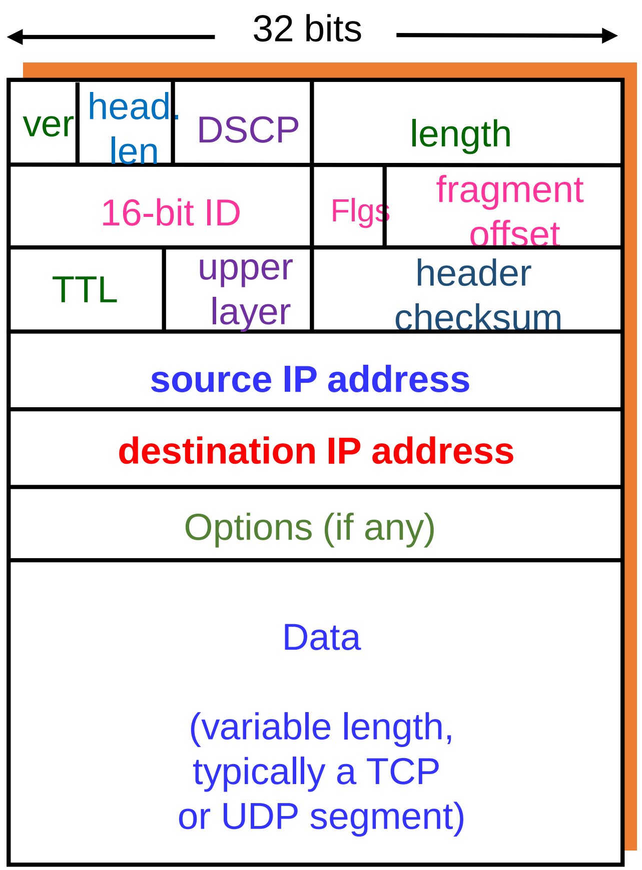
Length: Packet length in bytes

16-bit ID: A long IP packet is fragmented into smaller packets. All those small packets carry the same 16-bit-ID.

3-bit flags:
<Not used, Don’t frag., More frags. to follow>

Fragment offset x 2^3 : gives the position of the fragment in the original packet.

IP Packet Format



TTL: Time To Live
Max # of remaining hops.
TTL is decremented by 1 at each router.
TTL = 0 ➔ Router discards the packet

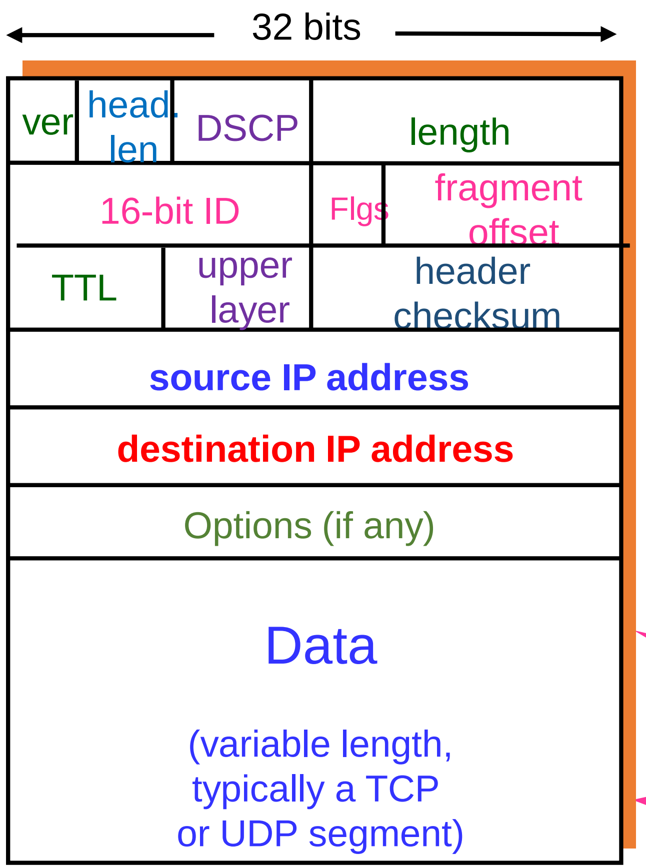
Upper layer: Upper layer protocol to deliver payload to (TCP = 6)

Header Checksum: to detect bit errors in packet header
(errors in “data” are ignored.)

Source IP address: 32-bit IP address of the node that originally created the packet.

Destination IP address: 32-bit IP address of the destination node of the packet.

IP Packet Format



Options:

timestamp, record route taken, specify list of routers to visit, …

Data: from the upper layer.
Usually one TCP (Transport Control Protocol) or one UDP (User Datagram Protocol) segment

IPv4 header length without options is 20 bytes……….

Header Checksum Calculation (at the Sender)

Example : IP header (in Hex) with checksum set to **0000**

4500 0073 0000 4000 4011 **0000** **c0a8** **0001** **c0a8** **00c7**

Step 1: Add all 16-bit blocks of the header

Result = **0010** **0100 0111 1001 1100**

Add the carry (0010) to the rest to get

Temp = 0100 0111 1001 1110

Step 2: Take 1's complement of Temp to get the checksum

Checksum = 1's complement(Temp)
= 1011 1000 0110 0001
=b861

Header with checksum =

4500 0073 0000 4000 4011 **b861** **c0a8** **0001** **c0a8** **00c7**

Send an IP packet with this header ….

Header Checksum Re-calculation (at the Receiver)

Assume that the header is received without bit error.

4500 0073 0000 4000 4011 **b861** **c0a8** **0001** **c0a8** **00c7**

Step 1: Add all the 16-bit blocks of the header

Result = 2 fffd

Add carry (2) to fffd to get ffff

Step 2: Take 1's complement of the final result from step 1,

1's complement of ffff = 0000.

Step 3: Decision

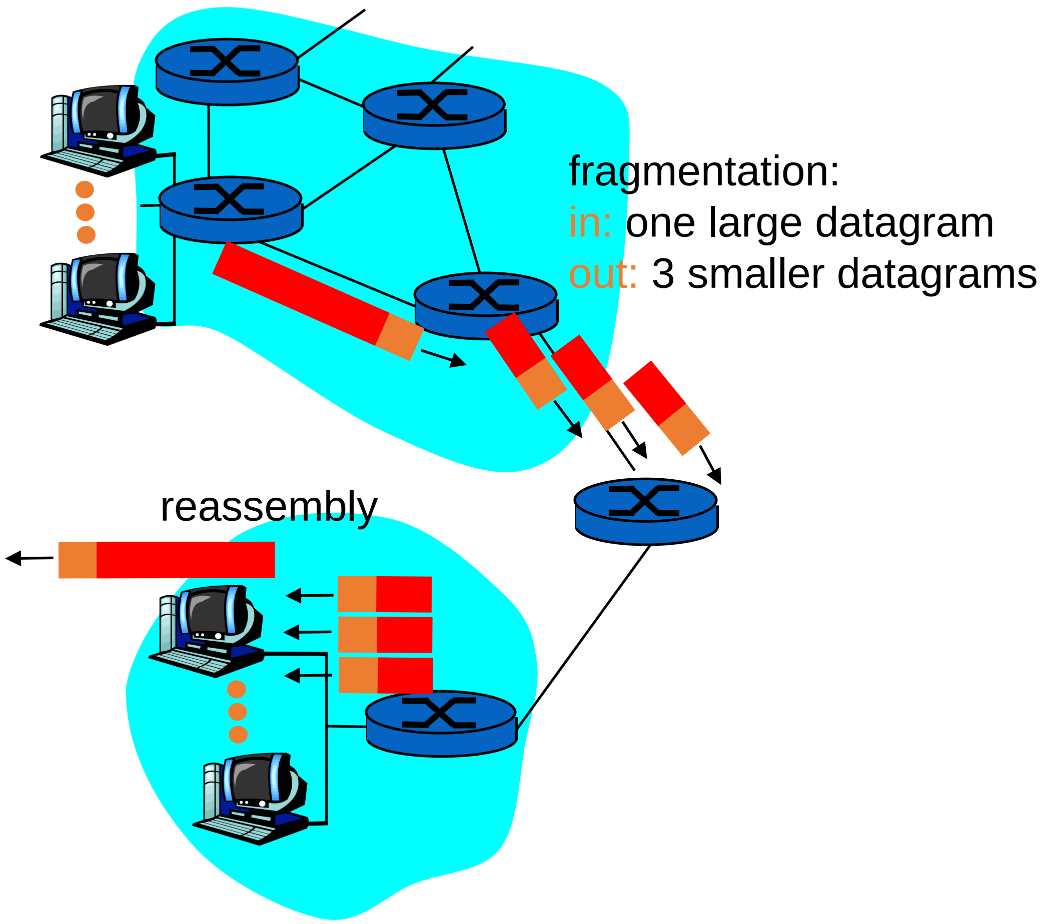
If the result from step 2 is **0000**: No error

Else: bit-error; drop packet

IP Fragmentation & Reassembly

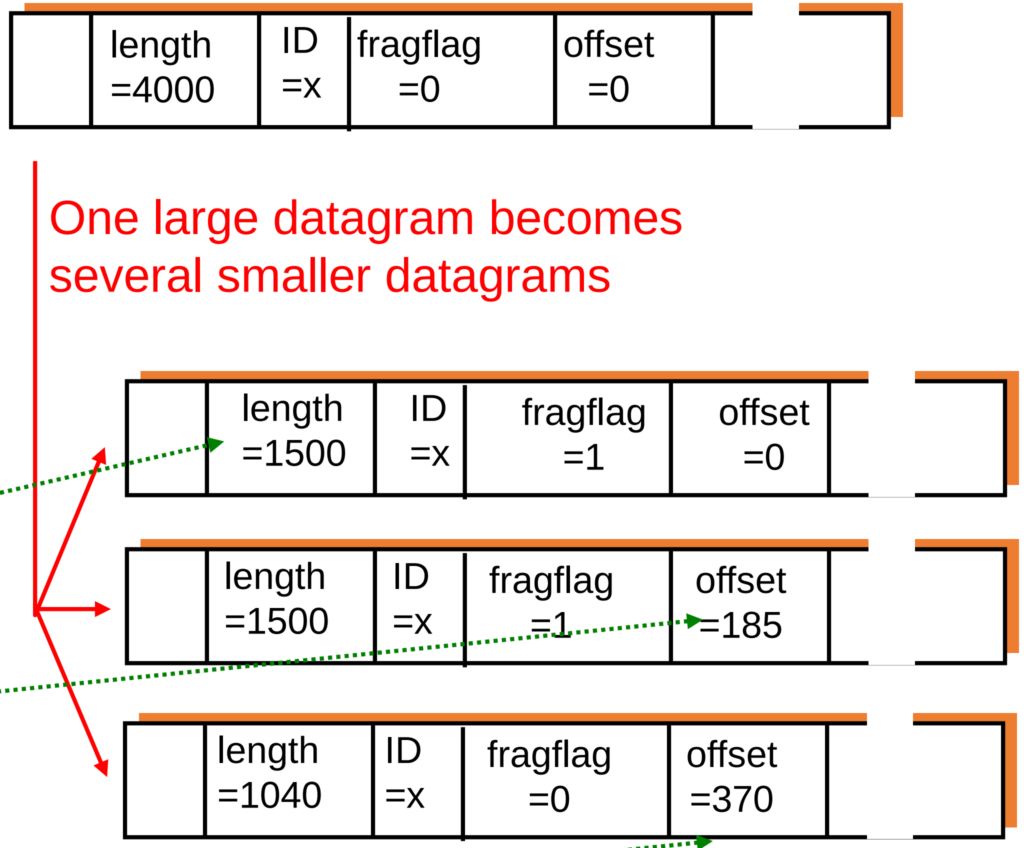
- network links have MTU (max. transfer unit)

- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify and order related fragments



IP Fragmentation and Reassembly

Data size = 4000-20 = 3980 Bytes



Example

- 4000 byte datagram
- MTU = 1500 bytes

1480 bytes in data field

offset = 1480/8

offset = (1480 +1480)/8

Verification: Data size = 1500 + 1500 +1040 – (20 + 20 + 20) = 3980 Bytes

Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What's inside a router

4.4 IP: Internet Protocol

- Datagram format
- DHCP/NAT
- ICMP
- IPv6

4.5 Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

4.6 Routing in the Internet

- RIP
- OSPF
- BGP

4.7 Broadcast and multicast routing

IP addresses: how to get one?

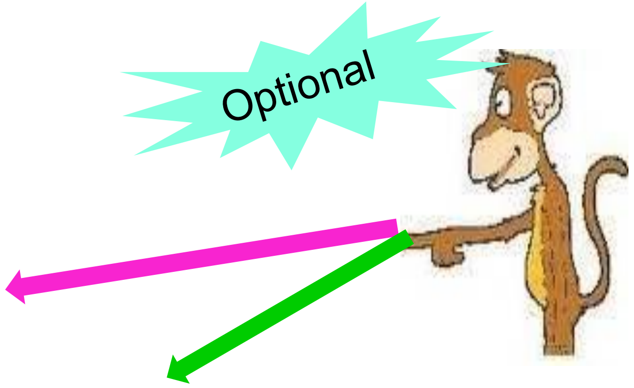
- hard-coded by system admin in a file

- DHCP: Dynamic Host Configuration Protocol: dynamically get address from a server

DHCP: Dynamic Host Configuration Protocol

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

- Allows reuse of addresses (only hold addr while connected)
- Support for mobile users …..



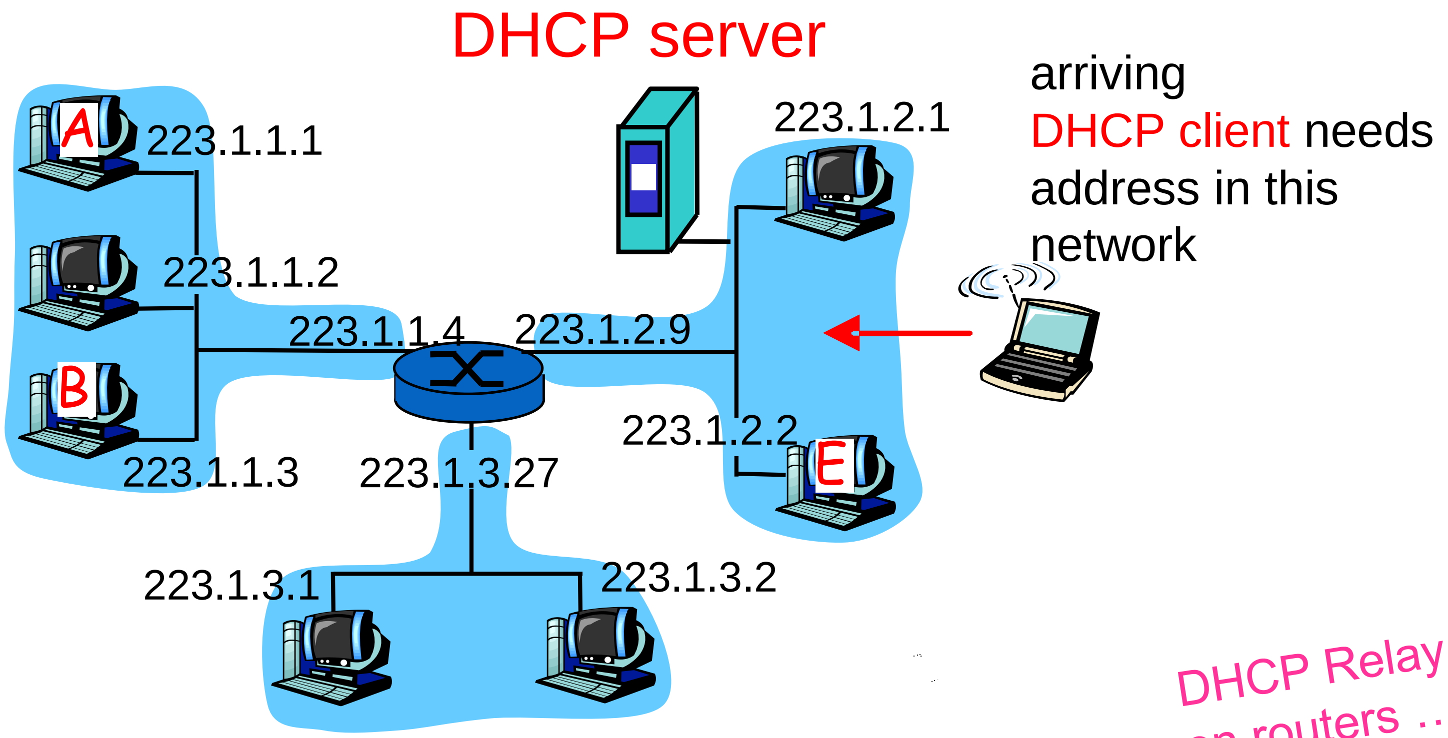
DHCP overview:

host broadcasts "DHCP discover" msg

DHCP server responds with "DHCP offer" msg

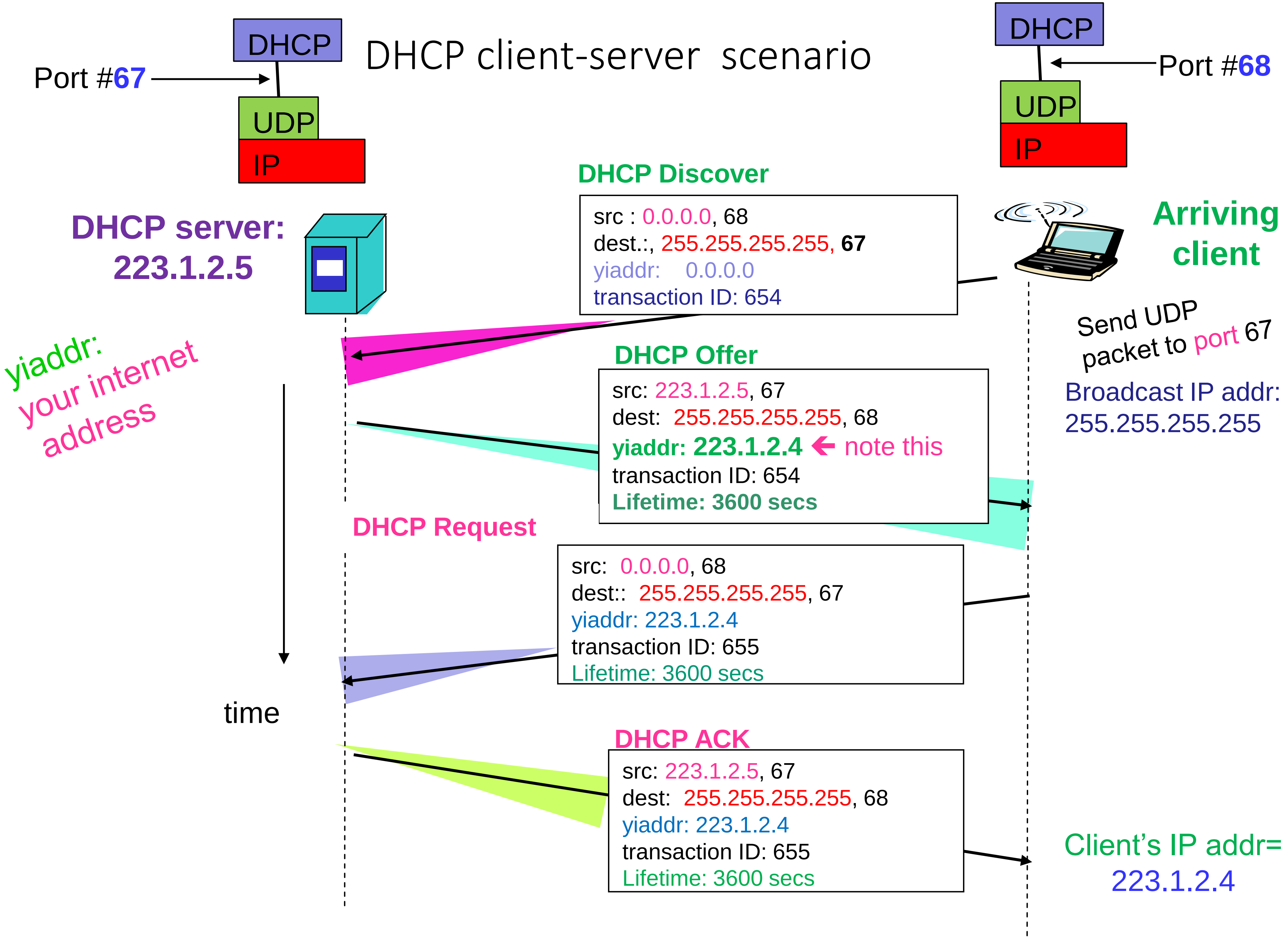
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

DHCP client-server scenario



DHCP Relay Agent
on routers …

DHCP client-server scenario



DHCP: returns more than an IP address

It returns:

- address of first-hop router for client
- name and IP address of DNS sever

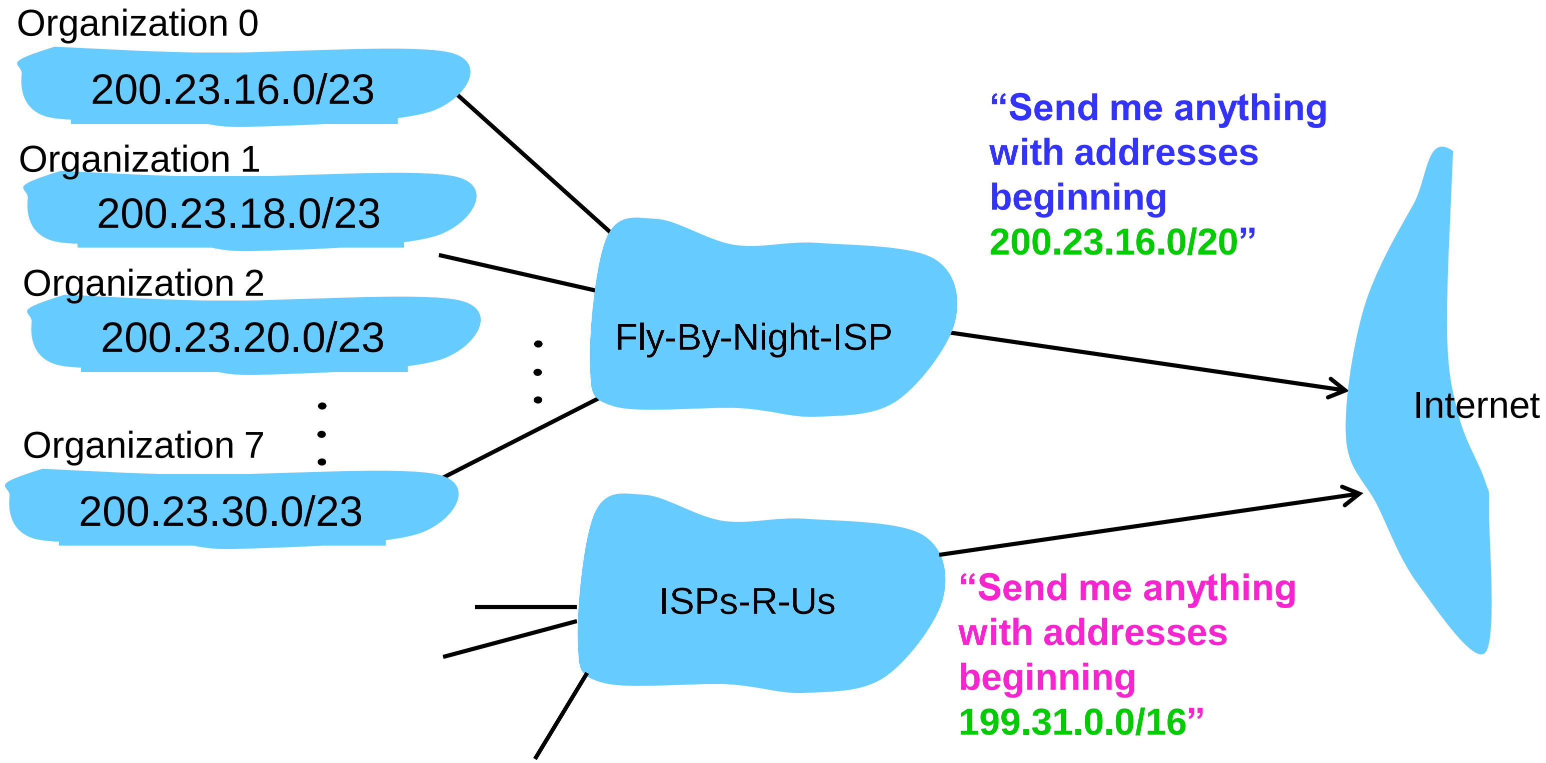
  DNS: Domain Name System
  Function: Machine name ←→ IP address
  Example: naik3.uwaterloo.ca ←→ 129.97.10.192

- network mask

IP addr block: 11001000 00010111 00010000 00000000 200.23.16.0/**20**
Net. mask: 11111111 11111111 11110000 00000000 255.255.240.0

Hierarchical addressing: address aggregation

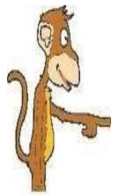
Hierarchical addressing allows efficient advertisement of routing information:



Address aggregation

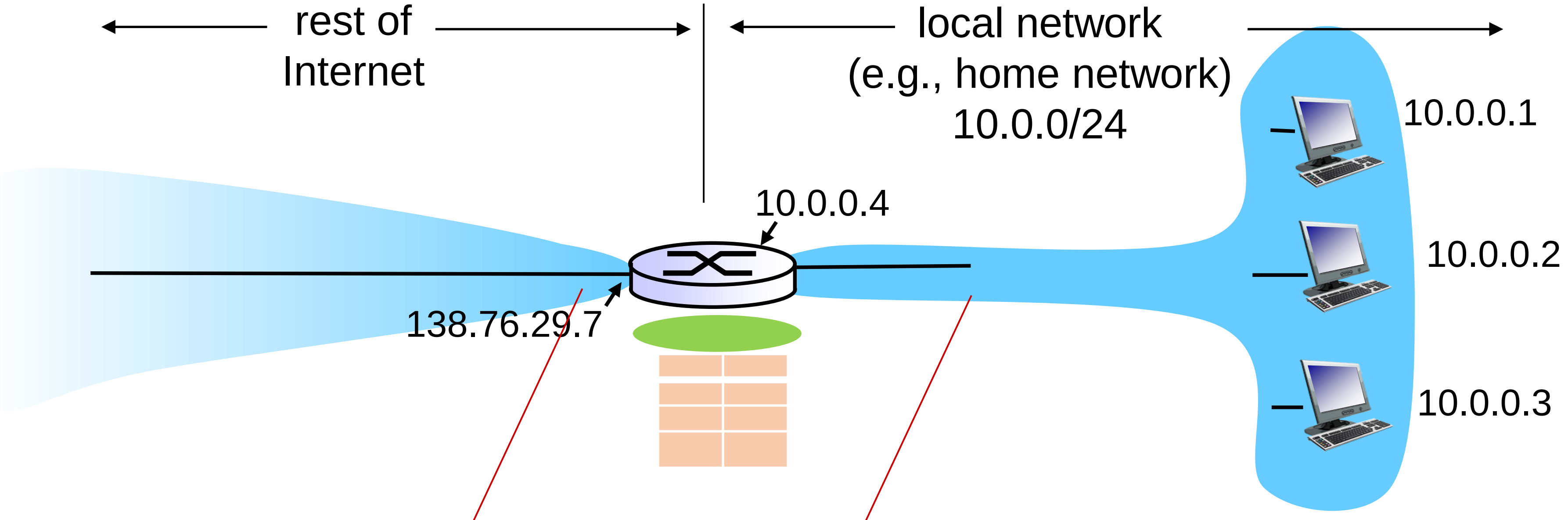
Organization 0	11001000 00010111 0001000	0	00000000	200.23.16.0/23
Organization 1	11001000 00010111 0001001	0	00000000	200.23.18.0/23
Organization 2	11001000 00010111 0001010	0	00000000	200.23.20.0/23
...	…..		….	….
Organization 7	11001000 00010111 0001111	0	00000000	200.23.30.0/23
ISP's **Address block**	11001000 00010111 0001000	0	00000000	**200.23.16.0/20**

NAT: Network Address Translation



Motivation: One way to solve the address shortage problem …..

The 2nd way is to use IPv6 …



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

(with unchanged dest. IP addr)

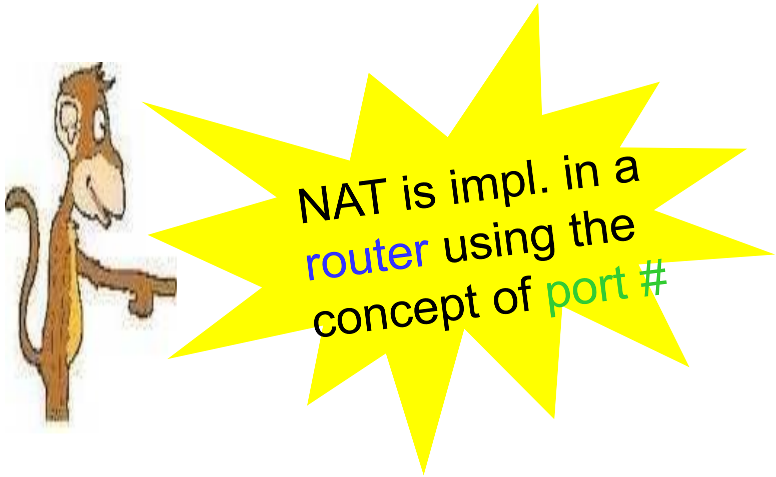
datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

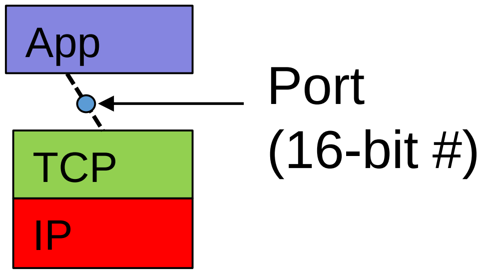
NAT: Network Address Translation

- Local network uses just one IP address as far as outside world is concerned. 🡐 more devices are supported.

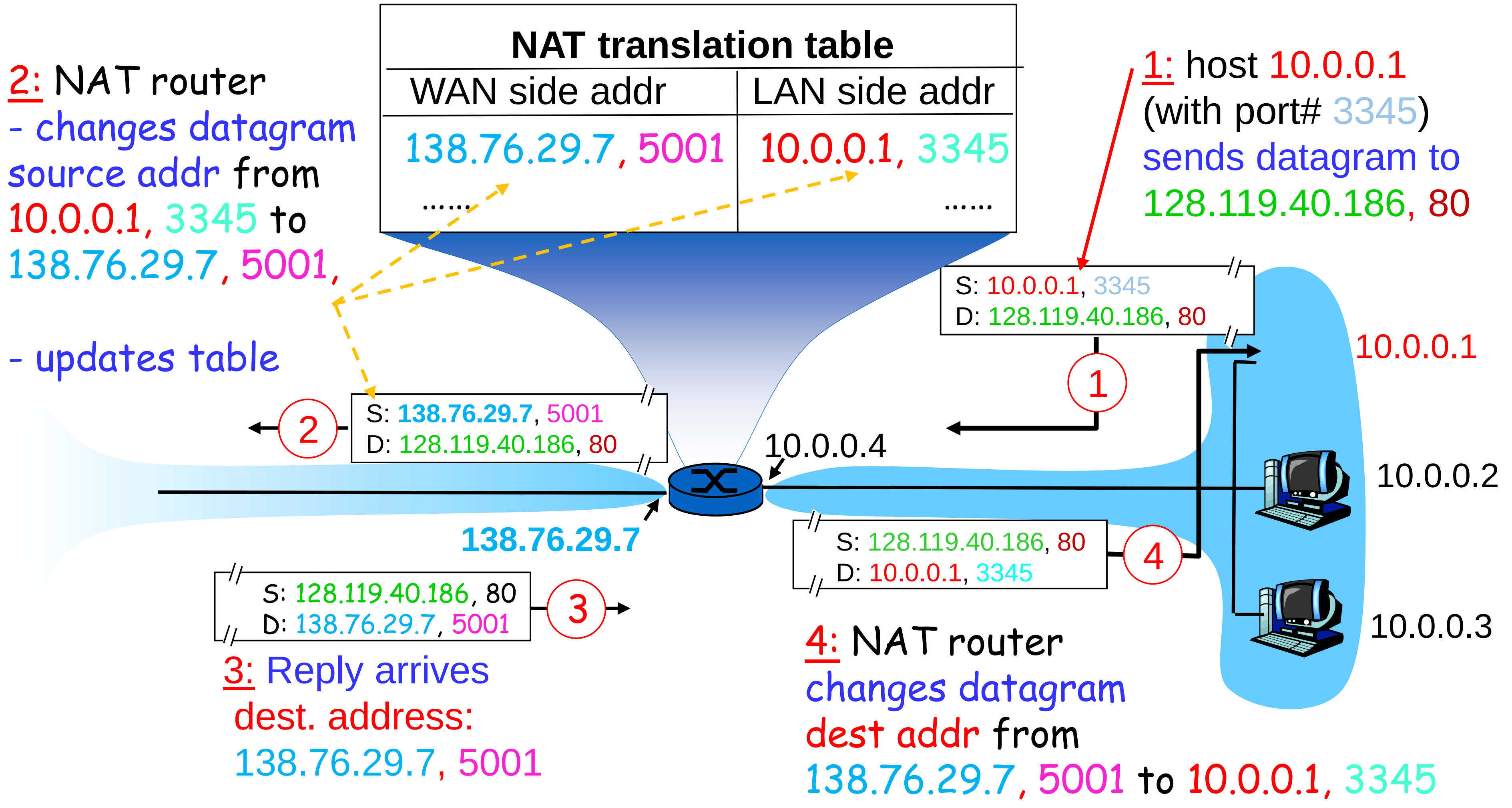
❖ A range of address is not needed from ISP: just one IP addr for all.

- Devices inside local net are **not** explicitly addressable (i.e. visible) by outside world (a security plus).
  - This constraint is seen as a security plus point.





NAT: Network Address Translation



Implementation of NAT

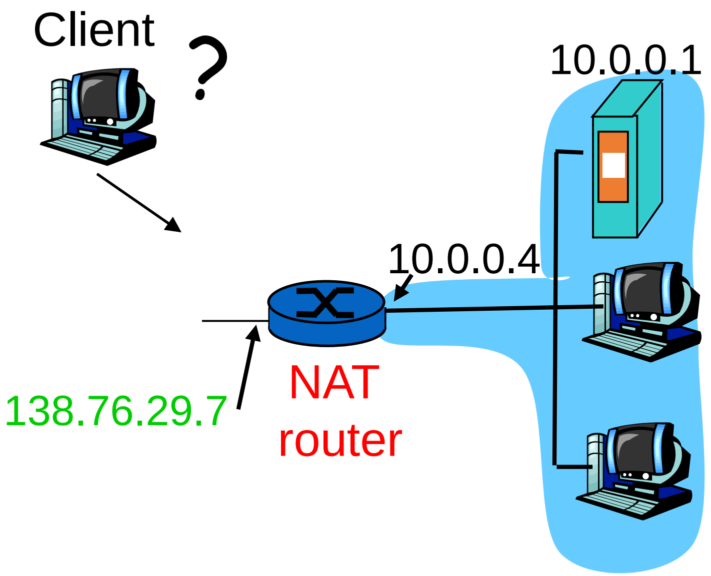
- *outgoing datagrams*
  - *replace* (src IP addr, port #) of every outgoing datagram with (NAT IP addr, new port #)
  - [remote clients/servers will respond using (NAT IP addr, new port #) as dest. addr.]
  - *remember* (in NAT translation table) every mapping (src IP addr, port #) ←→ (NAT IP addr, new port #)

incoming datagrams

- *replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (src IP addr, port #) stored in NAT table

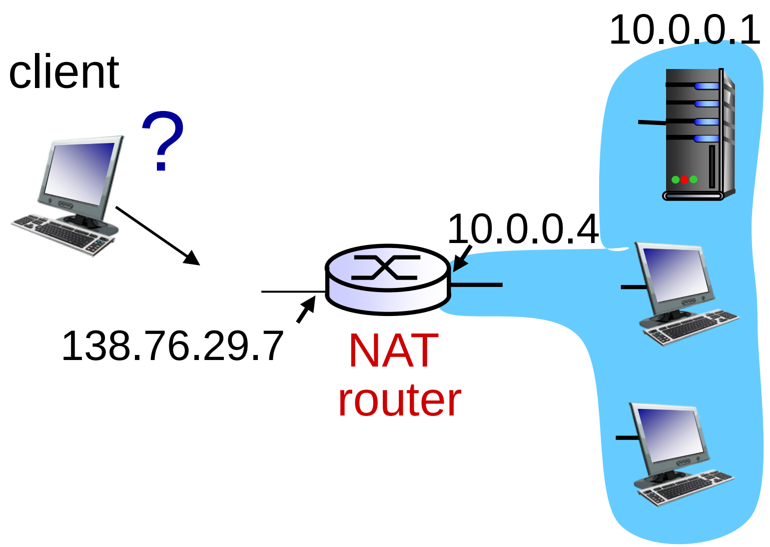
NAT traversal problem

- Client wants to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client can't use it as destination addr)
  - only one externally visible NATed address: 138.76.29.7



NAT traversal problem

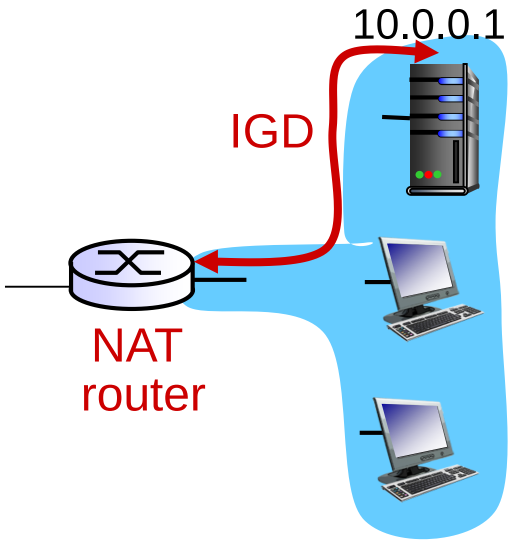
- *solution1:* statically configure NAT to forward incoming connection requests at given port to server
  - e.g., (123.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000



NAT traversal problem

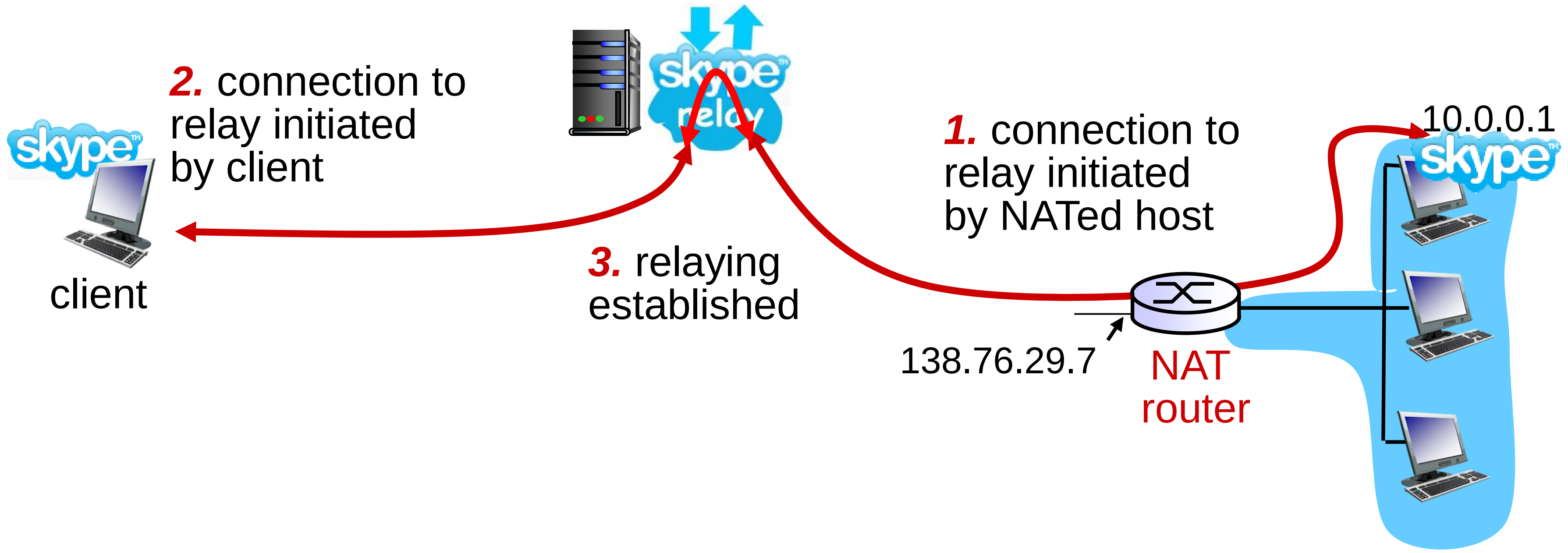
- *solution 2:* Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATed host to:
  - ❖ learn public IP address (138.76.29.7)
  - ❖ add/remove port mappings (with lease times)

  i.e., automate static NAT port map configuration



NAT traversal problem

- *solution 3:* relaying (used in Skype)
  - NATed client establishes connection to relay
  - external client connects to relay
  - relay bridges packets between to connections



Chapter 4: Network Layer

4.1 Introduction

4.2 Virtual circuit and datagram networks

4.3 What's inside a router

4.4 IP: Internet Protocol

- Datagram format
- DHCP/NAT
- ICMP
- IPv6

4.5 Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

4.6 Routing in the Internet

- RIP
- OSPF
- BGP

4.7 Broadcast and multicast 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
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)
  - ICMP 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 "port unreachable" packet (type 3, code 3)
- when source gets this ICMP, stops.

Chapter 4: Network Layer

4.1 Introduction

4.2 Packet Forwarding

4.3 What's inside a router

4.4 IP: Internet Protocol

- Datagram format
- DHCP/NAT
- ICMP
- IPv6

4.5 Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

4.6 Routing in the Internet

- RIP
- OSPF
- BGP

4.7 Broadcast and multicast routing

IPv6

- Initial motivation: 32-bit address space soon to be completely allocated.

- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS (Quality of Service)

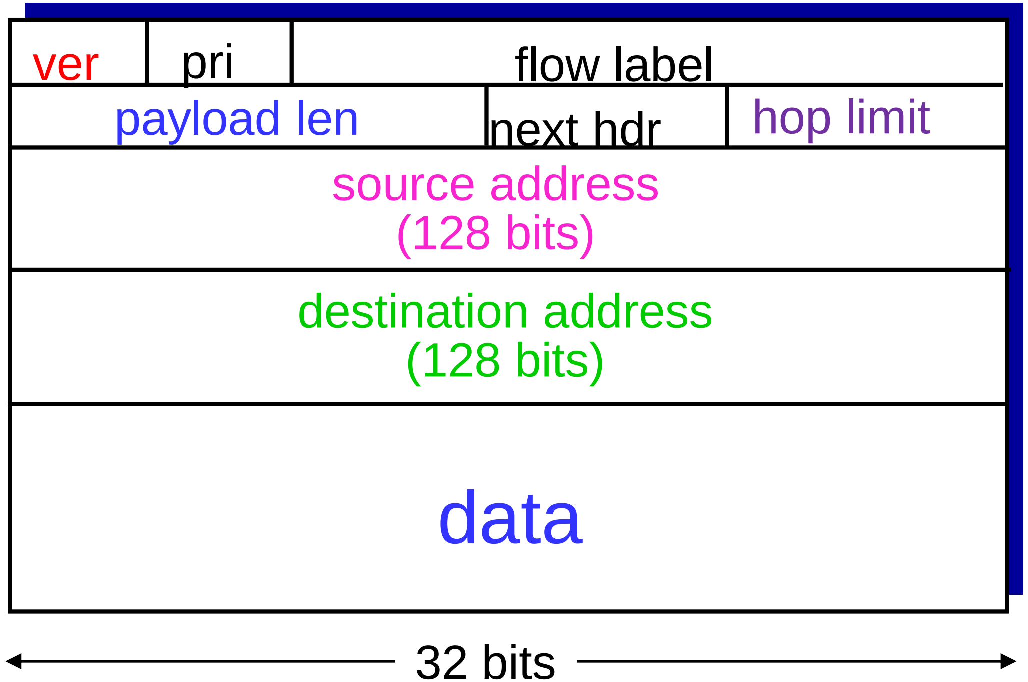
- IPv6 datagram format:
  - fixed-length 40 byte header
  - no fragmentation allowed

IPv6 Header (Cont)

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



Other Changes from IPv4

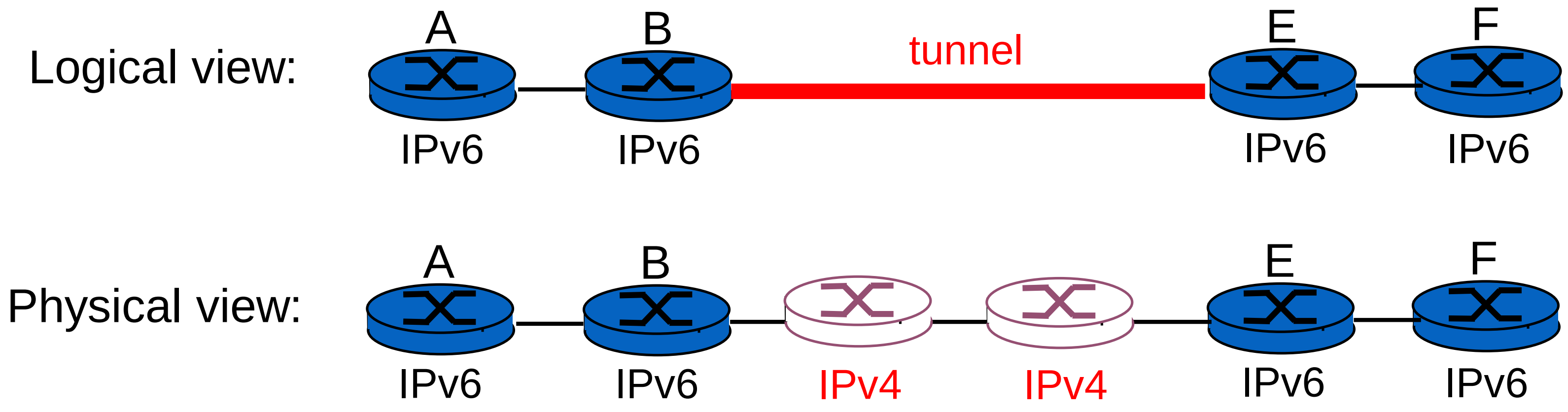
- *Checksum*: removed entirely to reduce processing time at each hop
- *Options:* allowed, but outside of header, indicated by "Next Header" field
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions

Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneously
  - How will the network operate with mixed IPv4 and IPv6 routers?

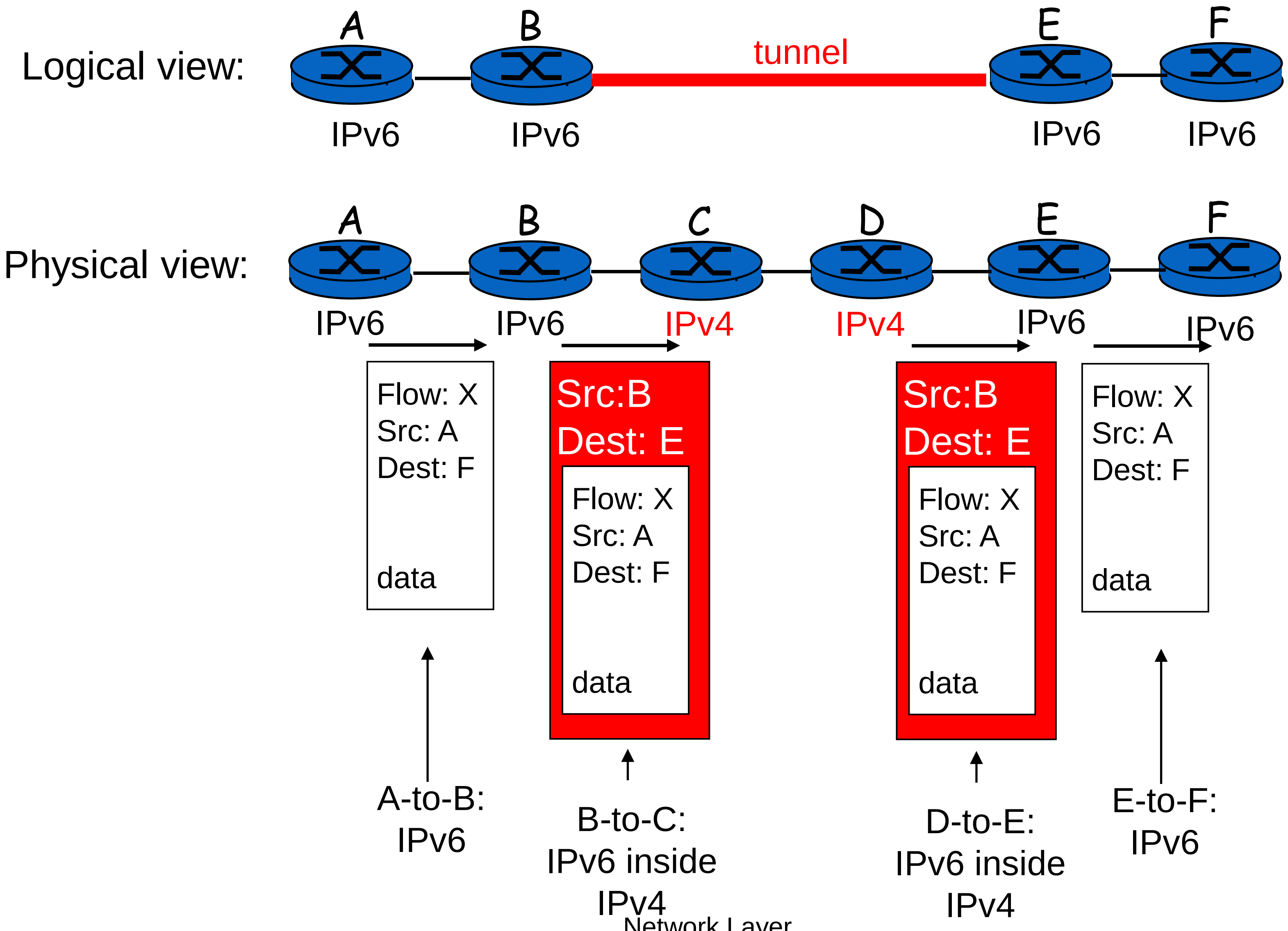
- *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers

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



Logical view:
A
B
tunnel
E
F
IPv6
IPv6
IPv6
IPv6
Physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Flow: X
Src: A
Dest: F
data
A-to-B:
IPv6
B-to-C:
IPv6 inside
IPv4
D-to-E:
IPv6 inside
IPv4
E-to-F:
IPv6

Chapter 4: Network Layer

4.1 Introduction

4.2 Packet Forwarding

4.3 What's inside a router

4.4 IP: Internet Protocol

- Datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5/4.6 Routing algorithms

- Distance vector + RIP
- Link state + OSPF
- Hierarchical routing
- BGP

Hierarchical Routing

Scale: with 200 million destination networks

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

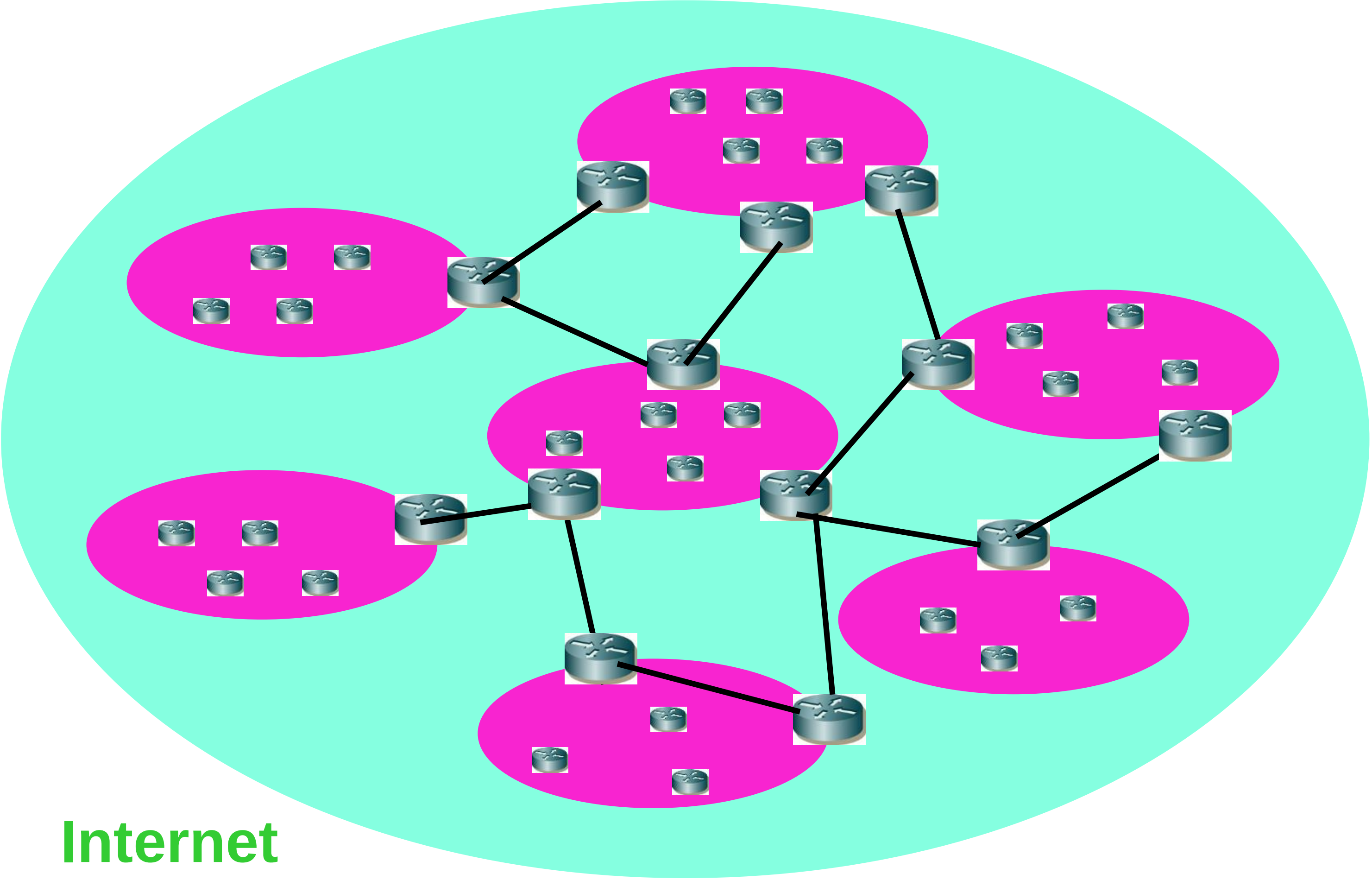
•*Address aggregation* alleviates the problem, but not enough…..

Hierarchical organization of the Internet

Autonomous System

BGP (Border Gateway Protocol) Routers

(Ordinary) routers



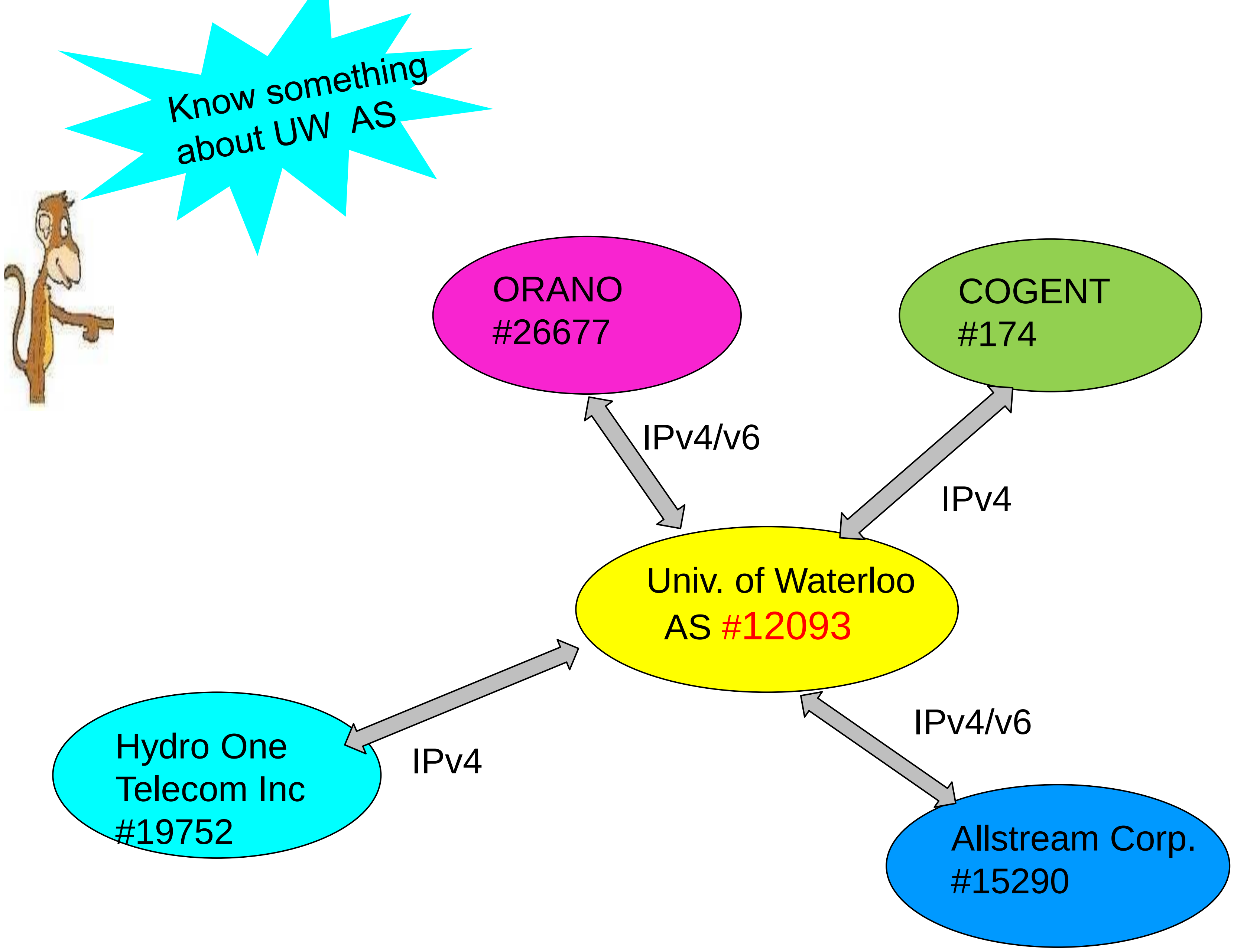
Autonomous System (AS)

An Autonomous System is **a set of routers** under a single technical administration, using an interior gateway protocol and common metrics to route packets **within the AS**, and using an exterior gateway protocol to route packets to other AS's.

AS's are identified by a 16-bit ID, called AS number.

Example AS numbers and names

Num of Customers	AS number	Network Name
159	174	COGENT Cogent/PSI
113	577	BACOM – Bell Canada
105	15290	ALLST-15290 – Allstream Corp.
102	852	ASN852 – Telus Advanced Communications
88	3356	LEVEL3 Level 3 Communications
84	6539	GT-BELL – Bell Canada
84	6327	SHAW – Shaw Comm. Inc.
64	701	UUNET – MCI Comm. Services, Inc. d/b/a Verizon Business
63	3257	TINET-BACKBONE Tinet SpA
57	6453	GLOBEINTERNET TATA Communications
50	3549	GBLX Global Crossing Ltd.
49	13768	PEER1 – Peer 1 Network Inc.


Know something
about UW AS
ORANO
#26677
COGENT
#174
IPv4/v6
IPv4
Univ. of Waterloo
AS #12093
IPv4/v6
Hydro One
Telecom Inc
#19752
IPv4
Allstream Corp.
#15290

Routing Protocols

Intra-AS routing
(within an AS)

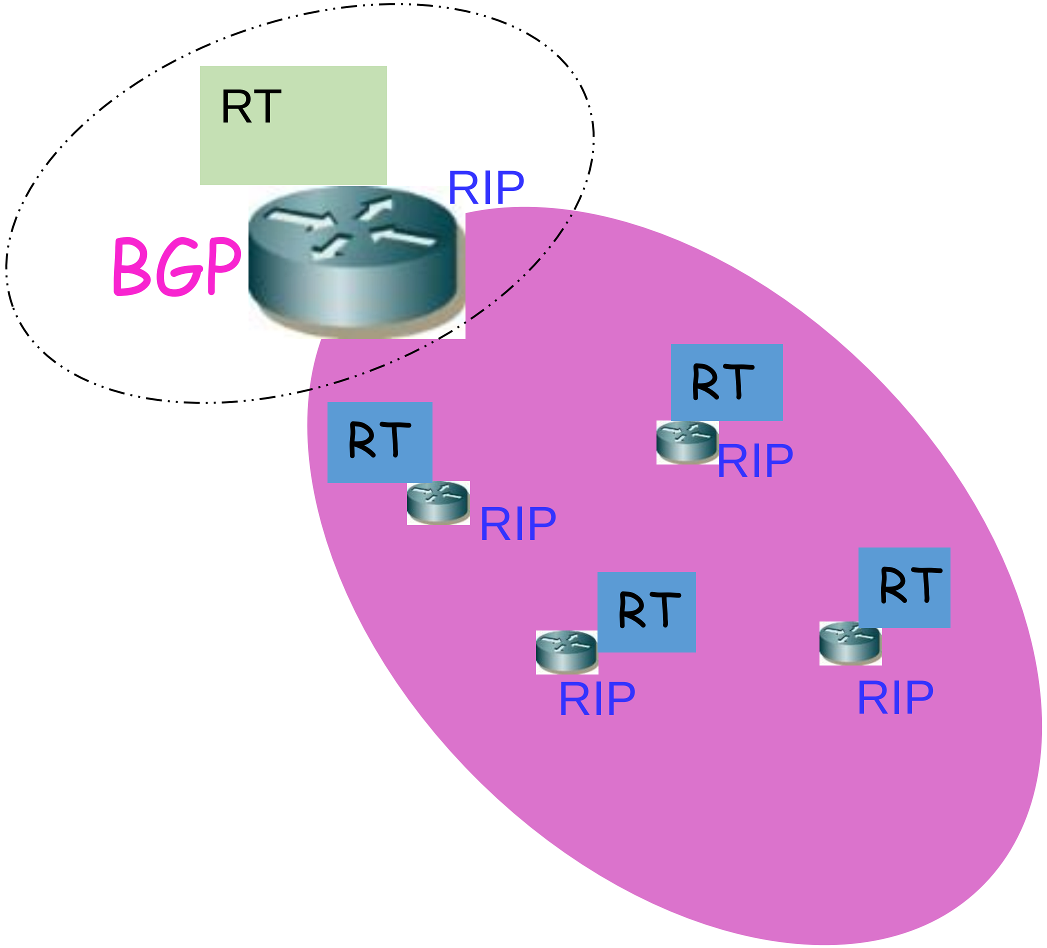
RIP: Routing Information Protocol
or
OSPF: Open Shortest Part First
or
your own proprietary protocol

Inter-AS routing
(among AS)

BGP: Border Gateway Protocol

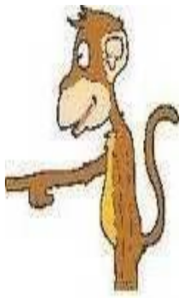
Choose one intra-AS routing protocol in a given AS.
Two different intra-AS routing protocols do NOT run in the same AS.

NOTE



AS (running RIP for intra-AS routing)

A few things to remember ….



RIP

- Based on the idea of distance vector
- Routers use “local” info. of the AS
- Applies Bellman-Ford algorithm
- Uses “hop count” as cost metric

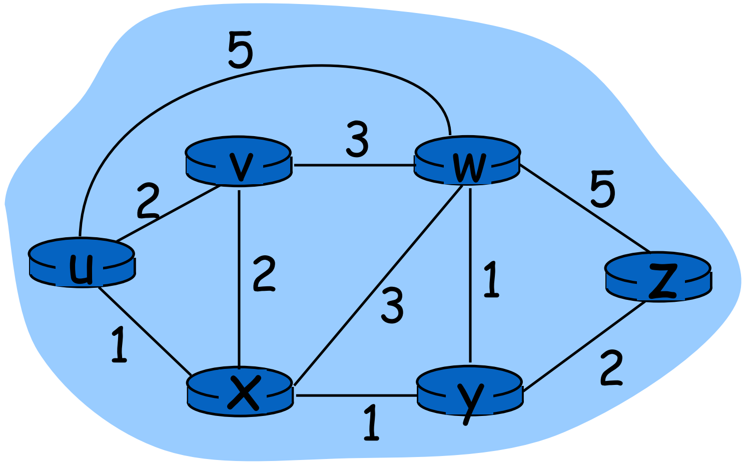
OSPF

- Based on the idea of link state
- Routers use “global” info of the AS
- Applies Dijkstra’s algorithm
- Uses a variety of cost metrics: hop count, delay, …..

BGP

- Based on the idea of “policy”
- Uses “global” info (AS) of the Internet

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) }

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

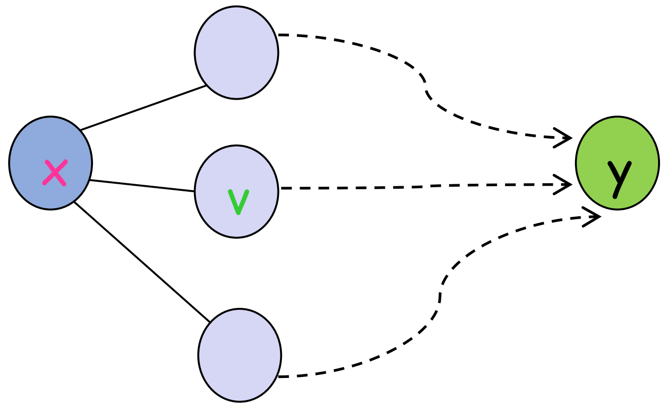
Example: c(w,z) = 5

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

Cost: hop count, delay, ….

RIP
Routing Information Protocol

Distance Vector Algorithm



Let

$d_x(y)$:= cost of least-cost path from x to y

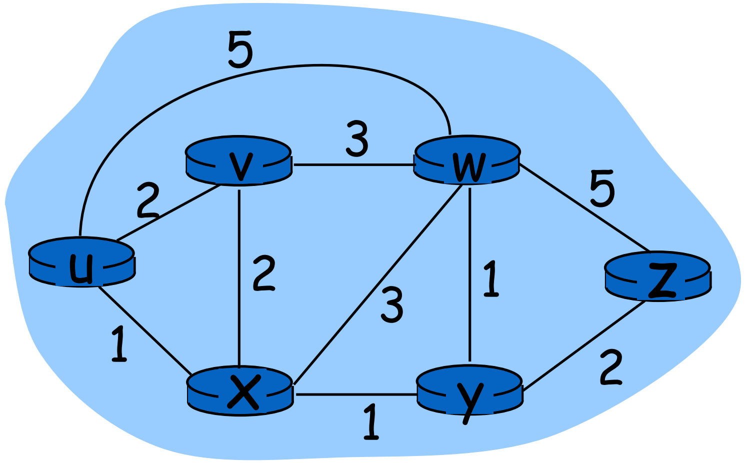
Bellman-Ford Equation

$$d_x(y) = \min \{c(x, v) + d_v(y)\}$$

min is taken over all neighbors v of x

The neighbor that leads to the minimum cost is the next hop on shortest path from x to y.

Bellman-Ford example



$$d_u(z) = \min \{ c(u,v) + d_v(z),\; c(u,x) + d_x(z),\; c(u,w) + d_w(z) \}$$

$$= \min \{2 + 5,\; 1 + 3,\; 5 + 3\}$$

$$= 4$$

Clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

Distance Vector Algorithm

- $D_x(y)$ = estimate of least cost from x to y
  - x maintains distance vector $\mathbf{D}_x = \{D_x(y): y \in N\}$

node x:

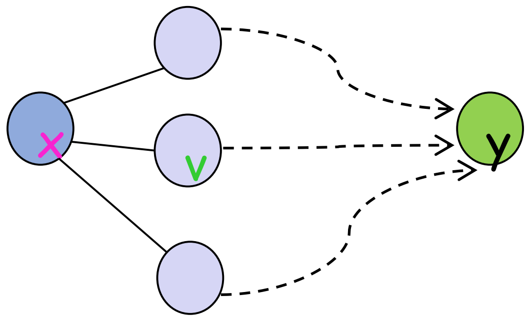
knows cost to each neighbor v: c(x, v)

maintains its neighbors' distance vectors.

For each neighbor v, x maintains

$\mathbf{D}_v = \{D_v(y): y \in N\}$

Distance vector algorithm



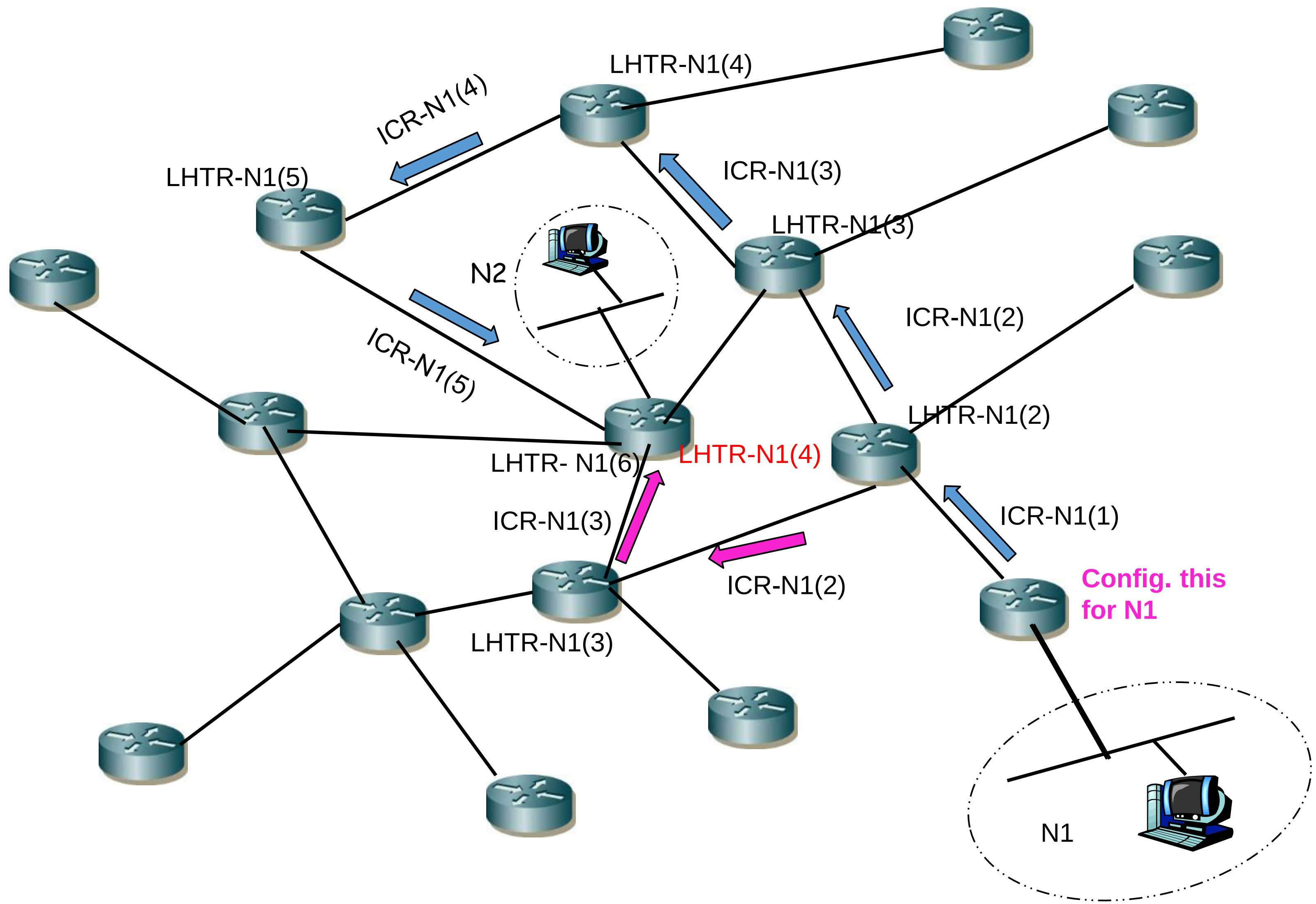
Basic idea:

- from time-to-time, each node sends its own distance vector estimate to neighbors

- when x receives new DV estimate from a neighbor, it updates its own DV using B-F equation:

$$D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\} \quad \text{for each node } y \in N$$

- In steady state
  the estimate $D_x(y)$ ***converges*** *to the actual least cost* $d_x(y)$



I can reach N1 (Hopcount cost): ICR-N1(cost)

Learns how to reach N1 (Hopcount cost): LHTR-N1(cost)

Distance Vector Algorithm: General Idea

Each node:

wait for (change in local link cost or msg from neighbor)

↓

recompute estimates

↓

if DV to any dest has changed, *notify* neighbors

Distance Vector Algorithm

- In <u>steady state</u>, shortest paths are established.

Routers learn from *neighbors only*. ← simplicity

*Good news travels fast (and **explicitly**).*

When a router finds a shorter path to a dest., the news is broadcasted to its neighbors in the next round of comm.

Bad news travels slowly

When a router does not hear from a neighbor, it does not tell its other neighbors about the failure.

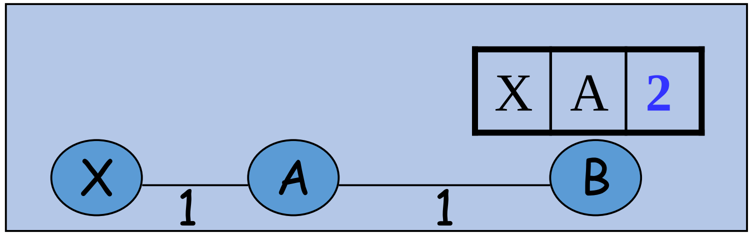
Routing loops are formed.

DV: Two-node instability problem

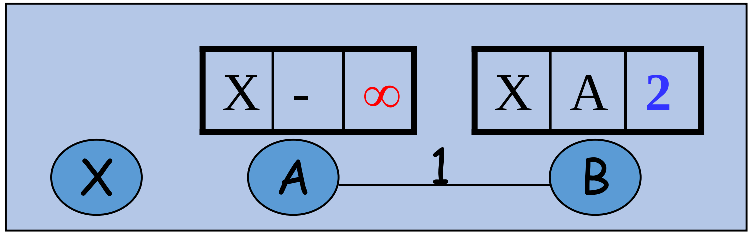
Note: RT @ each node

Dest.	NH	Cost

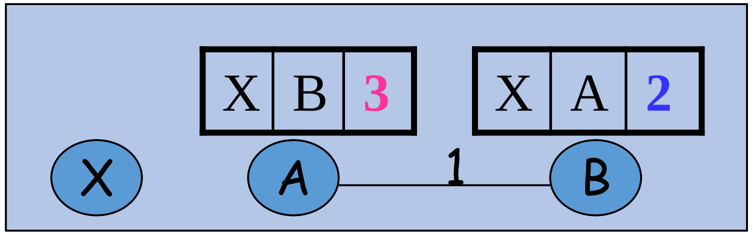
Before failure [Dest, NH, cost]



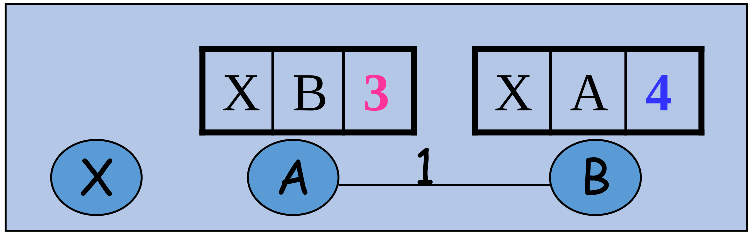
After failure



After **A** receives update from **B**



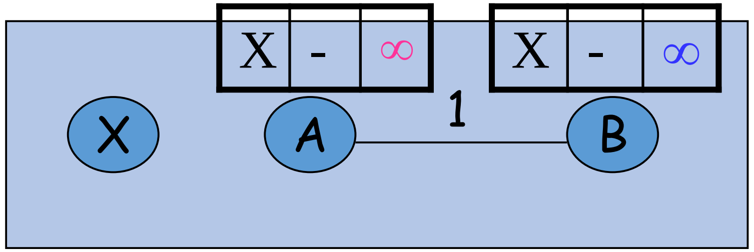
After **B** receives update from **A**



⋮

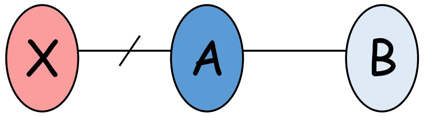
⋮

Finally



DV: Solutions to the loop problem

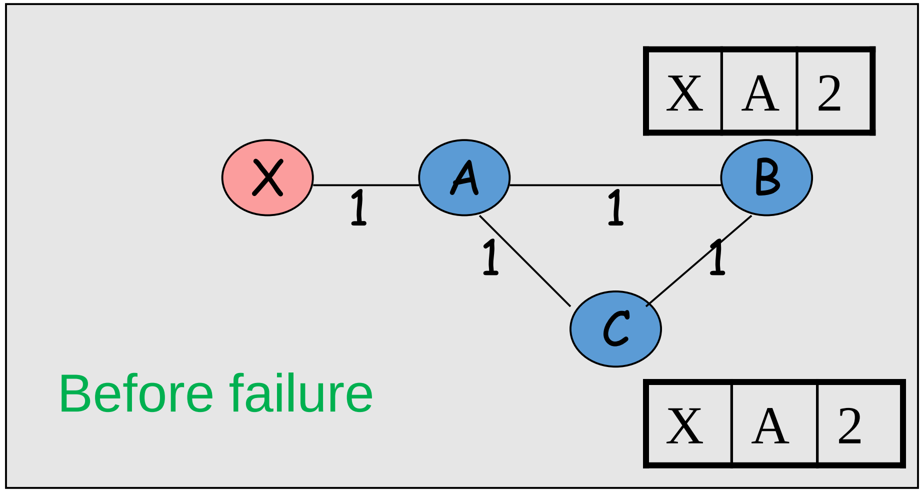
- Redefine "infinity" to a smaller number (say, 16).



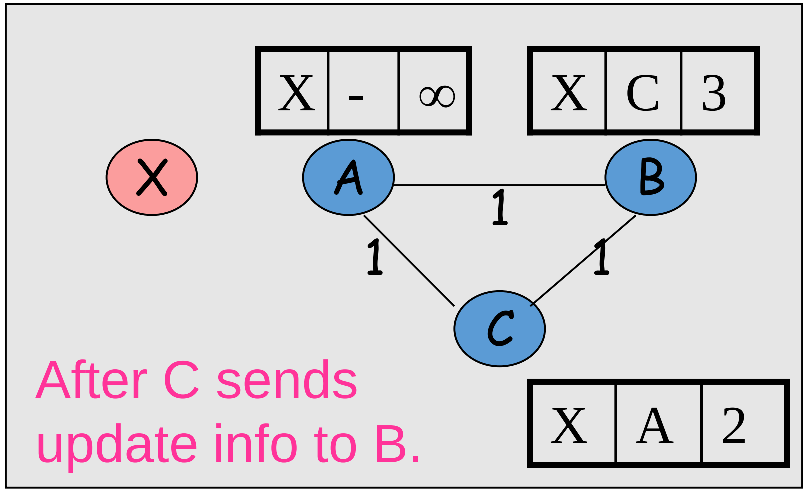
- Split horizon
  - If node B learns about a **path** to X from A, subsequently, B does not tell A about this.
    - Taking information from A, modifying it, and sending it back to A creates confusion.

- Split horizon with poisoned reverse
  - If A routes to X via B, A tells B that it has an infinity-cost path to X.

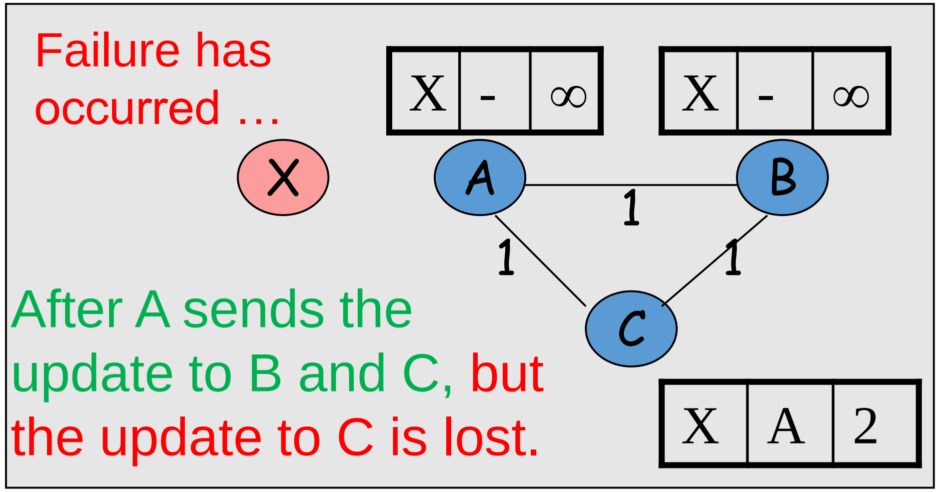
DV: Three-node instability problem



Before failure

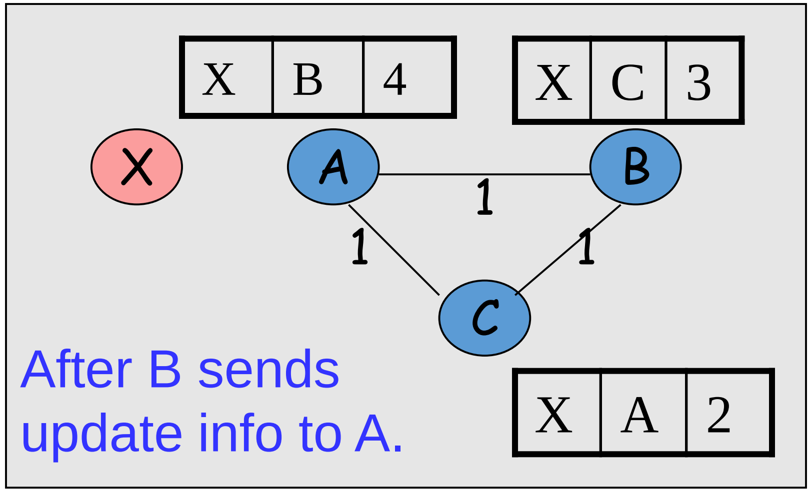


After C sends update info to B.



Failure has occurred …

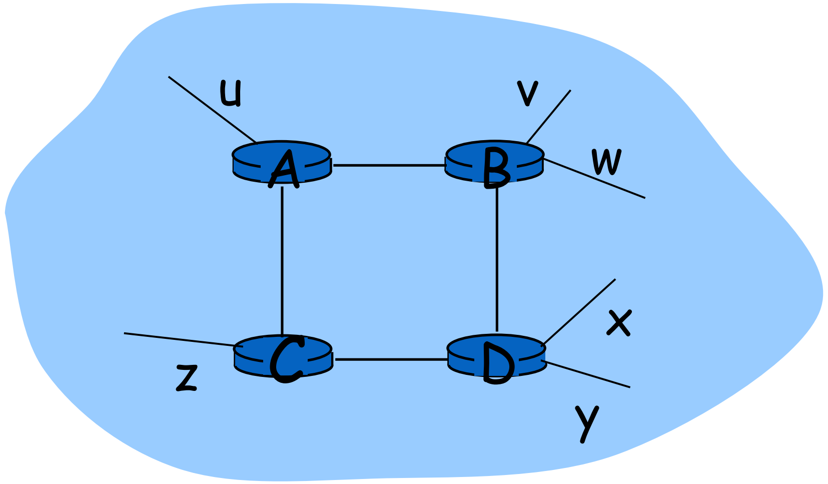
After A sends the update to B and C, but the update to C is lost.



After B sends update info to A.

RIP (Routing Information Protocol)

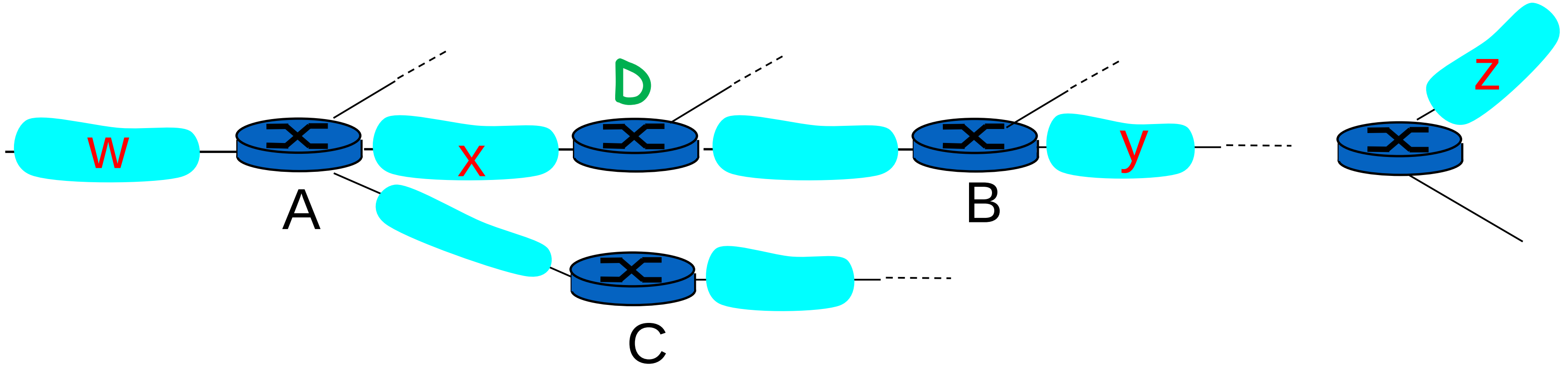
- included in BSD-UNIX distribution in 1982
- uses Distance Vector algorithm
  - cost metric: # hops (max = 15 hops), each link has cost 1
  - DVs exchanged with neighbors every 30 sec in response message (aka advertisement)
  - each advertisement: list of up to 25 destination *subnets (in IP addressing sense)*



from router A to destination subnets:

subnet	hops
u	1
v	2
w	2
x	3
y	3
z	2

RIP: Example



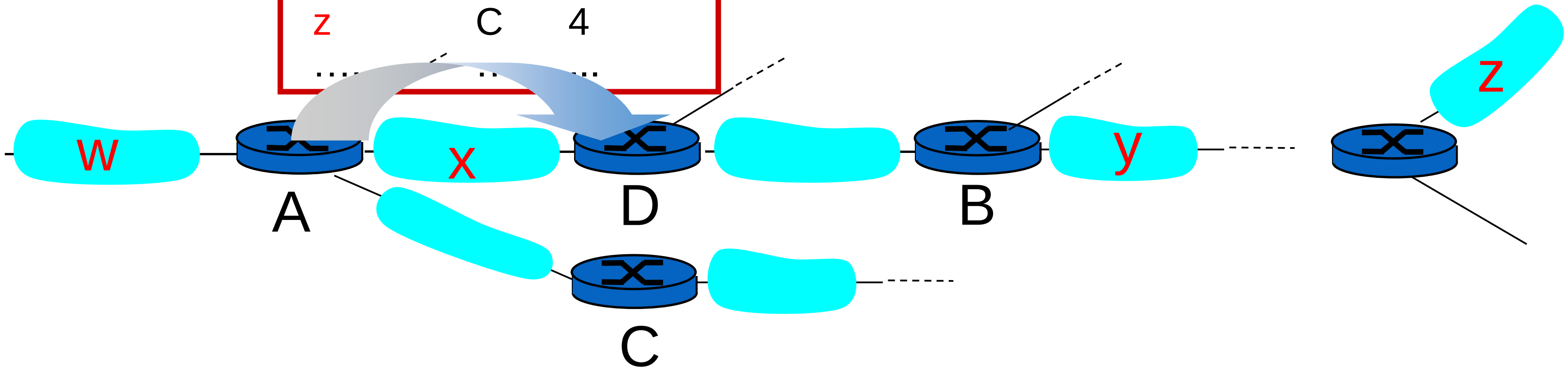
routing table in router D

destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	B	7
x	--	1
….	….	….

RIP: Example

A-to-D advertisement

dest	next	hops
w	-	1
x	-	1
z	C	4
….	…	...



routing table in router D

destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
….	….	....

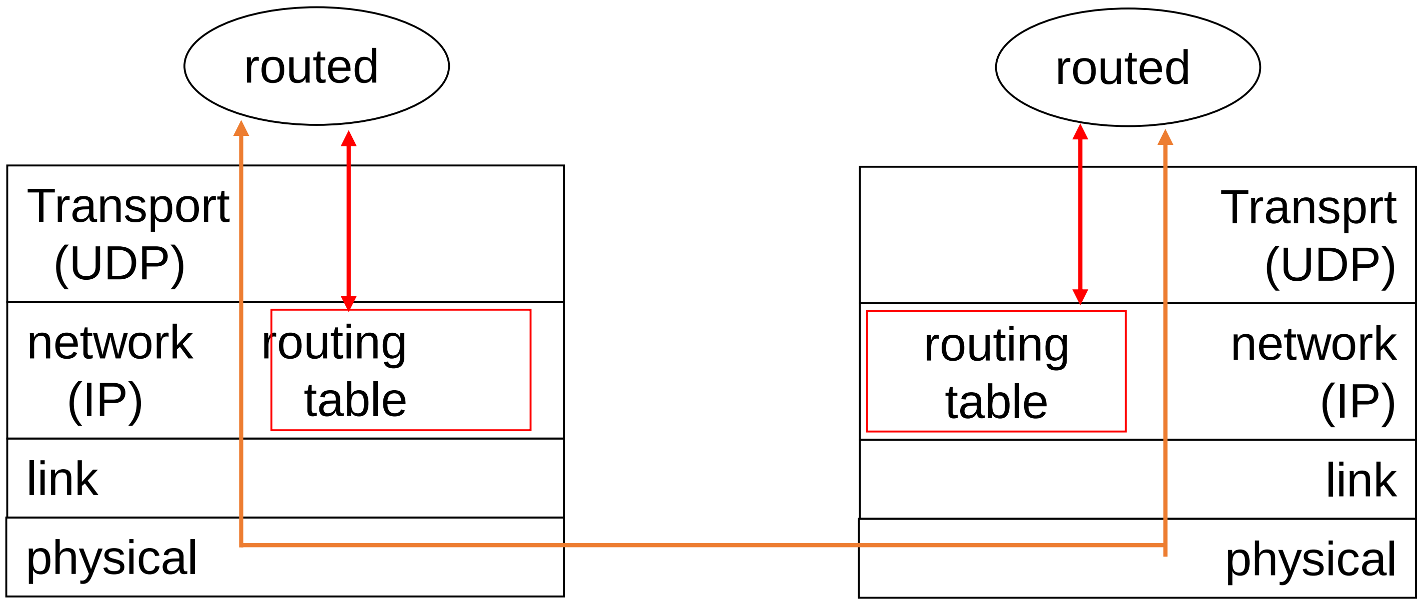
RIP: Link Failure and Recovery

If no advertisement heard from a neighbor after 180 sec --> neighbor/link declared dead

- routes via neighbor invalidated
- new advertisements sent to neighbors
- neighbors in turn send out new advertisements (if tables changed)
- link failure info quickly (?) propagates to entire net
- *poisoned reverse* used to prevent ping-pong loops (infinite distance = 16 hops)

RIP Table processing

- RIP routing tables managed by **application-level** process called route-d (daemon)
- advertisements sent in UDP packets, periodically repeated



OSPF
Open Shortest Path First

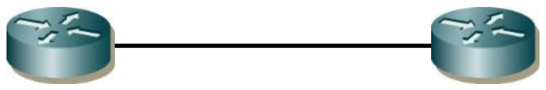
OSPF (Open Shortest Path First)

- "open": publicly available
- Uses Link State algorithm

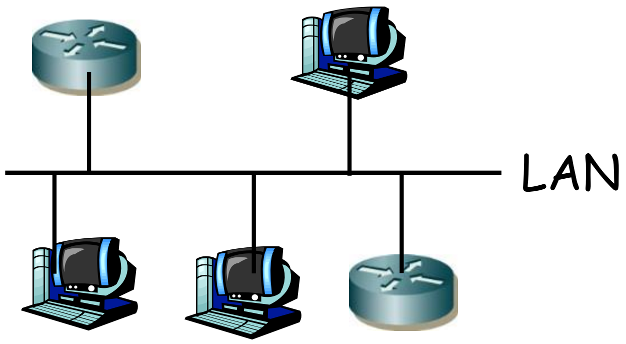
  - Each router learns the complete topology of the network (within AS) via flooding of OSPF-advertisement messages
  - Use Dijkstra's algorithm to compute shortest paths between all pairs of nodes
  - For a given source/dest. pair, find "next hop" from the shortest path.

OSPF: Types of links

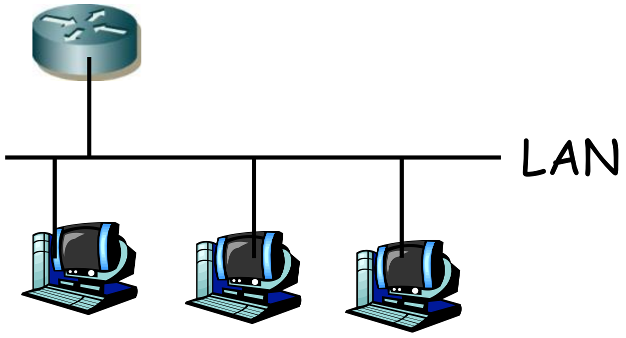
1. Point-to-Point Link



2. Transient Link



3. Stub Link



By means of LS-advertisement, other routers know what destinations are connected in the AS.

OSPF Link-State

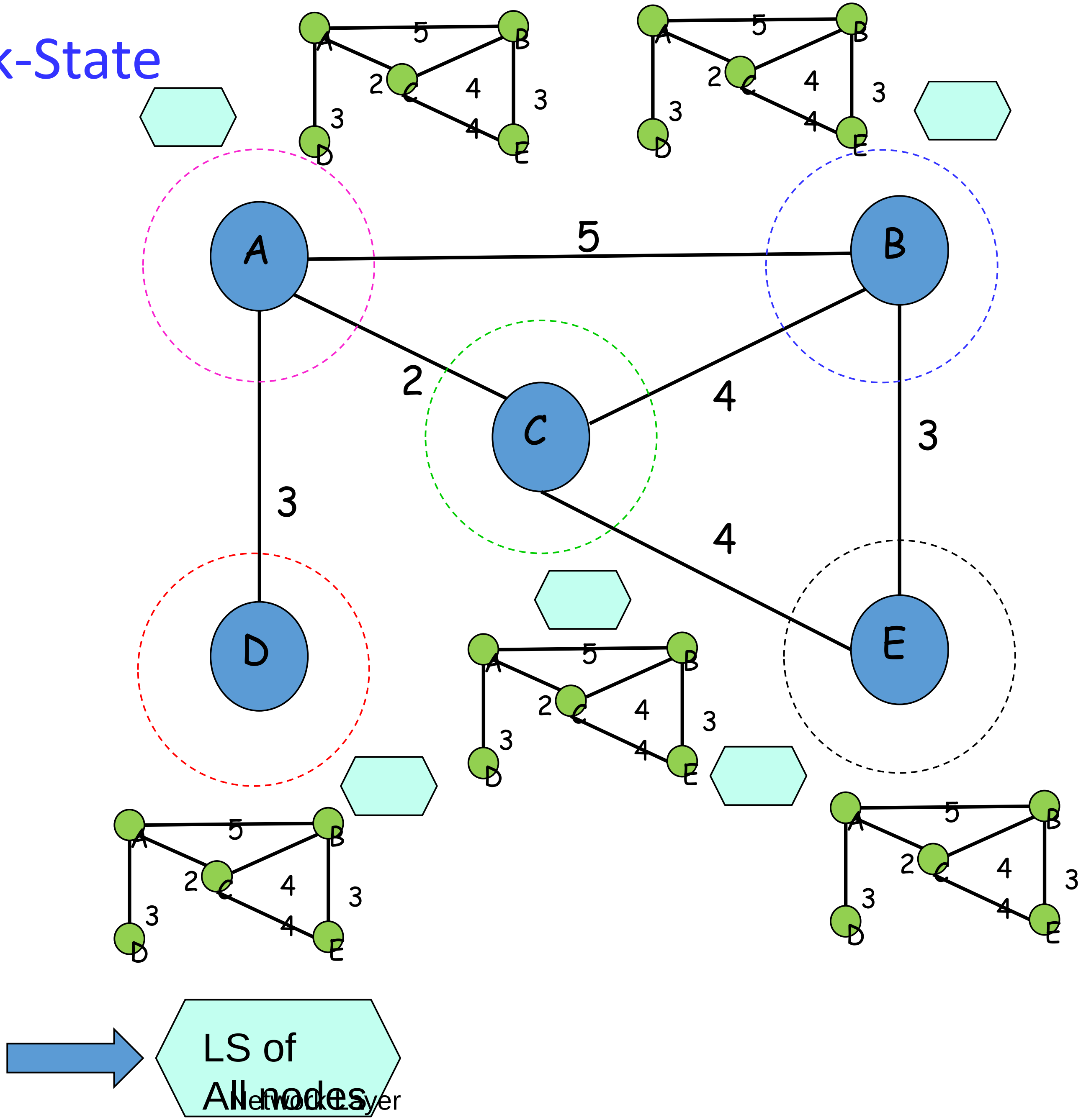
Link-State of A:
(A, B, 5)
(A, C, 2)
(A, D, 3)

Link-State of B:
(B, A, 5)
(B, C, 4)
(B, E, 3)

Link-State of C:
(C, A, 2)
(C, B, 4)
(C, E, 4)

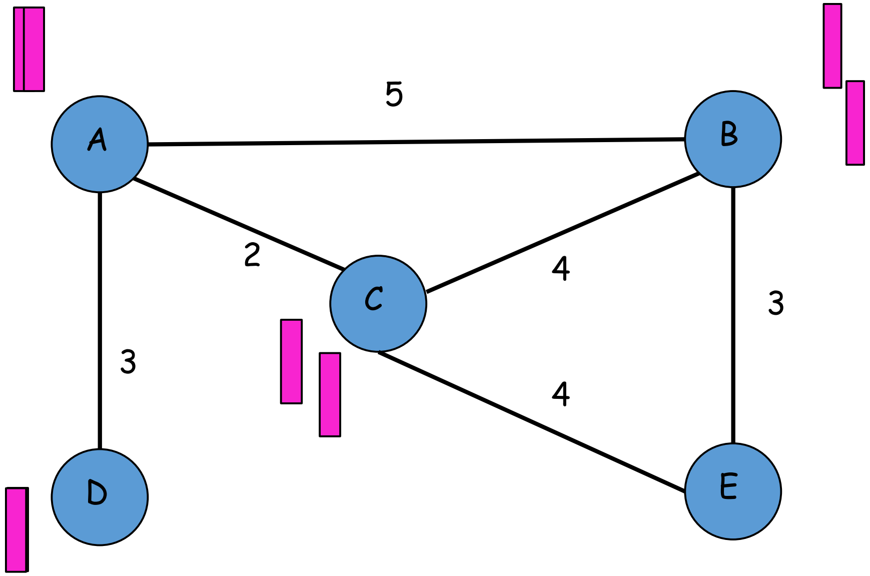
Link-State of D:
(D, A, 3)

Link-State of E
(E, B, 3)
(E, C, 4)



OSPF Flooding of LS-advertisemnt

Flooding example of LS of D: (D, A, 3) =



Similarly, LS of other nodes are flooded ……

A Link-State Routing Algorithm

Cost?
Hop count, delay, loss, …

Notation:

- c(x,y): link cost from node x to y;

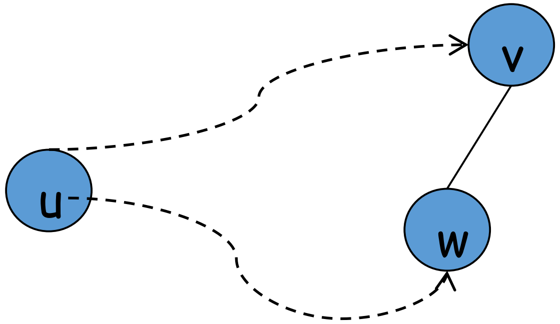
  = ∞ if not direct neighbors

- 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

Dijkstra's Algorithm

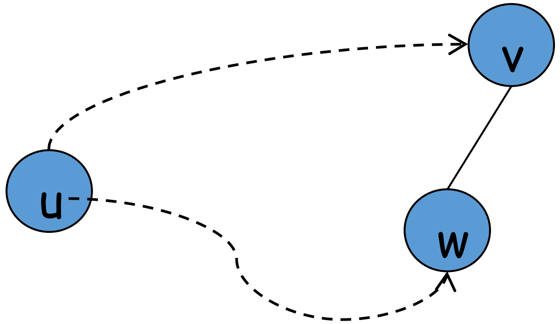
All nodes run this algorithm.

```
Initialization:  u is assumed to be "self."
  N' = {u}
  for all nodes v
    if v adjacent to u
        then D(v) = c(u,v) and p(v) = u
    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) ); p(v) = w /* if w is chosen */

  /* 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'
```



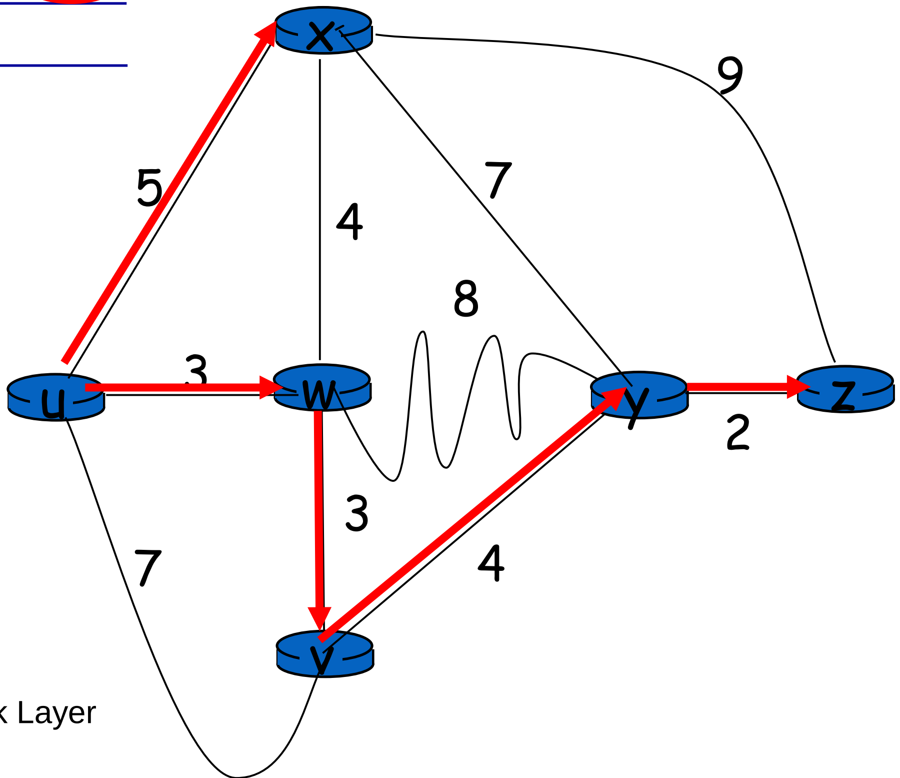
Dijkstra’s algorithm: example

Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					



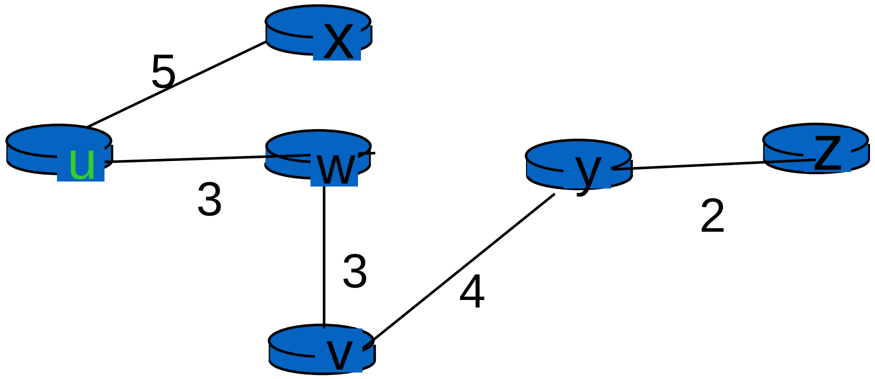
Notes:

- construct shortest path tree by tracing predecessor nodes
- ties can exist (broken arbitrarily)



Dijkstra's algorithm: example

Resulting shortest-path tree from u:



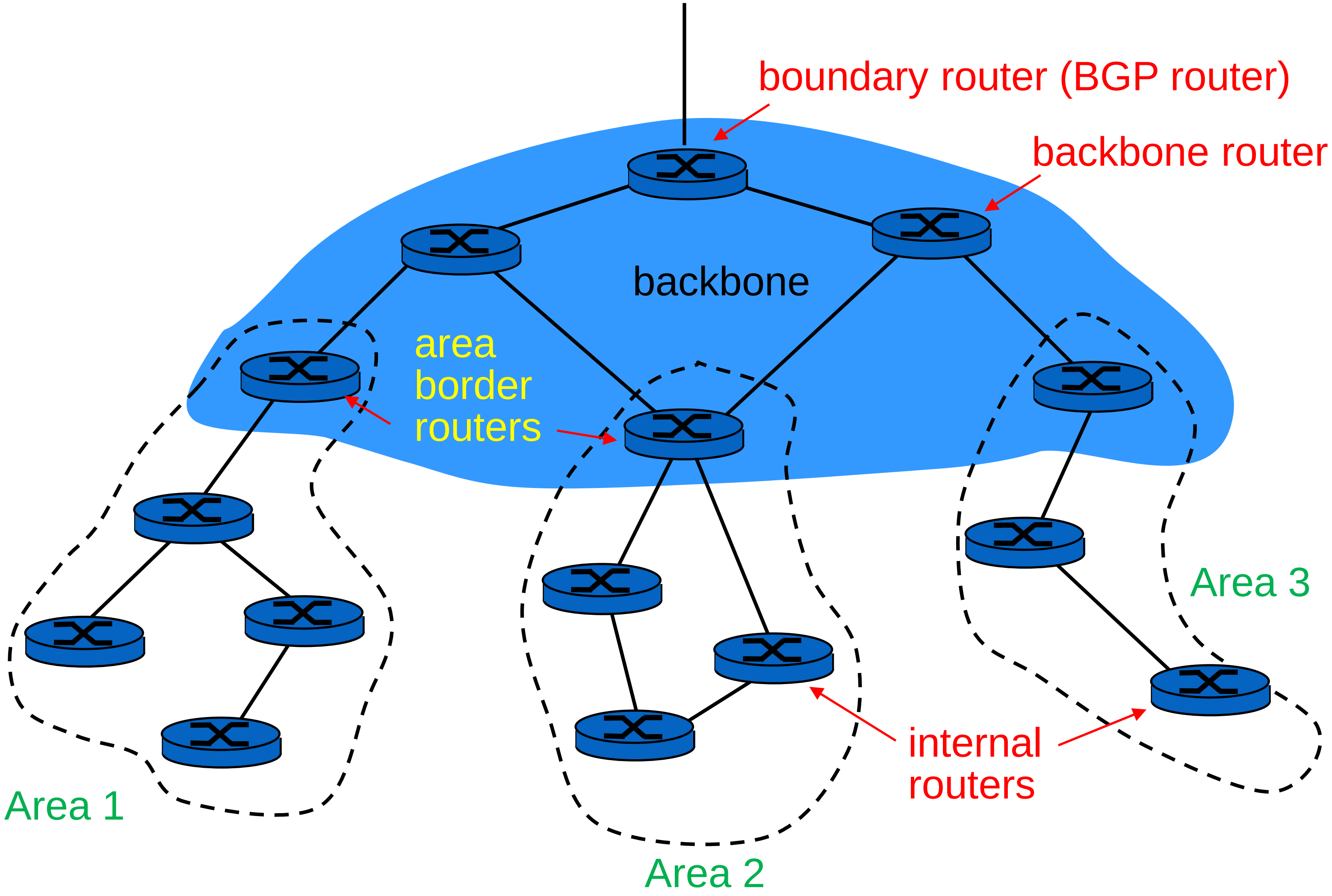
Resulting routing table in u:

Destination	Next hop
v	w
x	x
y	w
w	w
z	w

OSPF "advanced" features (not in RIP)

- security: all OSPF messages authenticated (to prevent malicious intrusion)
- multiple same-cost paths allowed (only one path in RIP)
- hierarchical OSPF in large domains (next slide ....)

Hierarchical OSPF



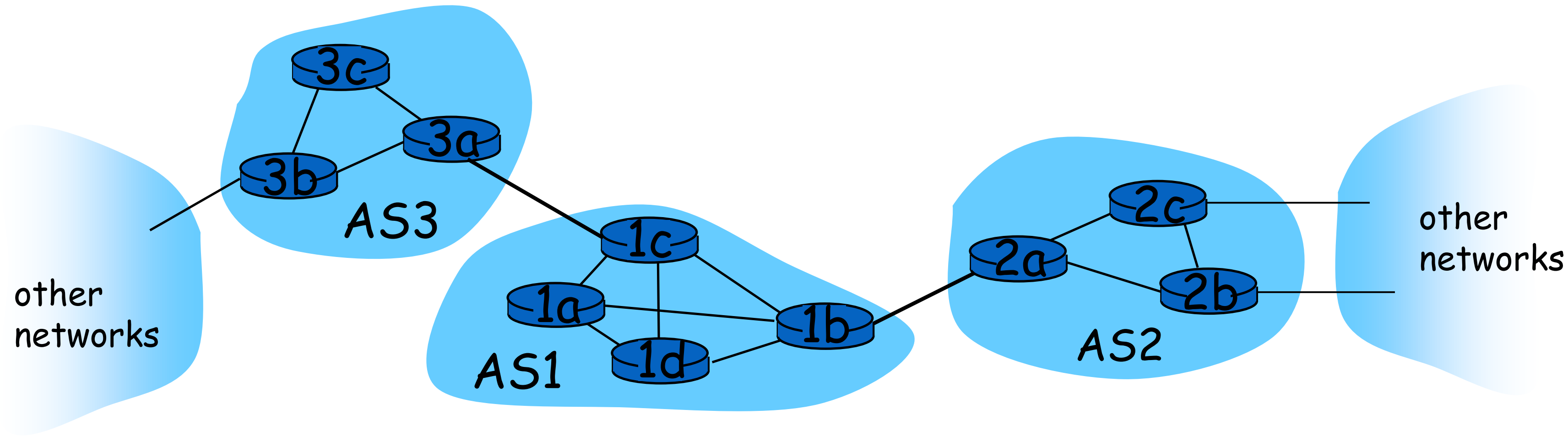
Border Gateway Protocol

Inter-AS tasks

- suppose a router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to edge router, but which one?

<u>AS1 must:</u>

1. learn which dests are reachable through AS2, which through AS3
2. propagate this reachability info to all edge routers in AS1



Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): *the* de facto inter-domain routing protocol
  - "glue that holds the Internet together"

allows subnet to advertise its existence to rest of Internet: *"I am here"*

BGP provides each AS a means to:

(eBGP) obtain subnet reachability information from neighboring ASs.

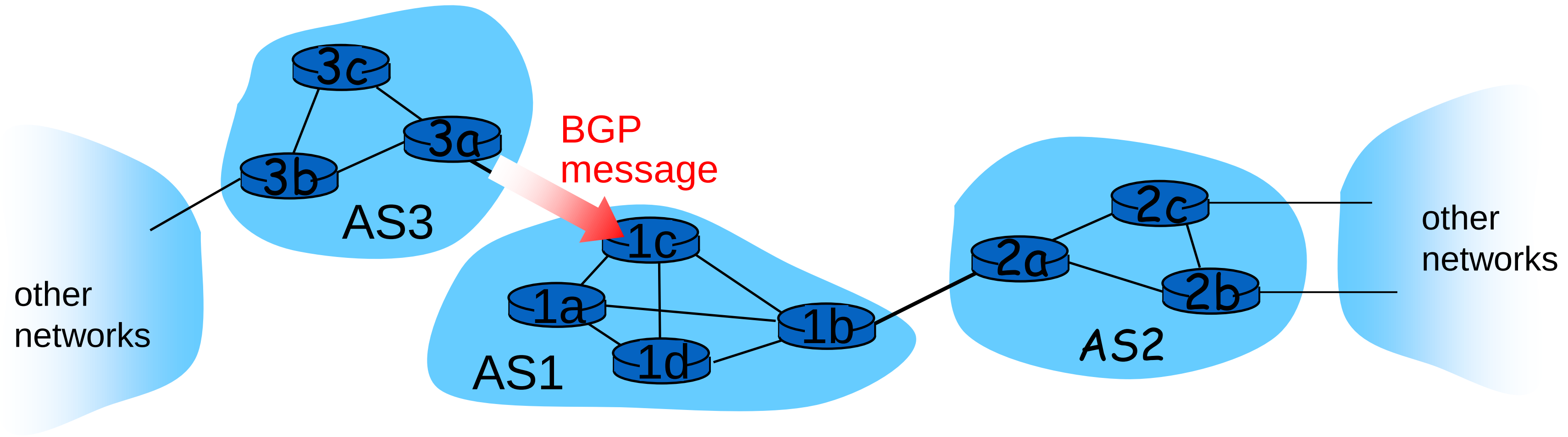
(iBGP) propagate reachability information to other AS-internal BGP routers.

determine "good" routes to other networks based on reachability information and policy

BGP basics

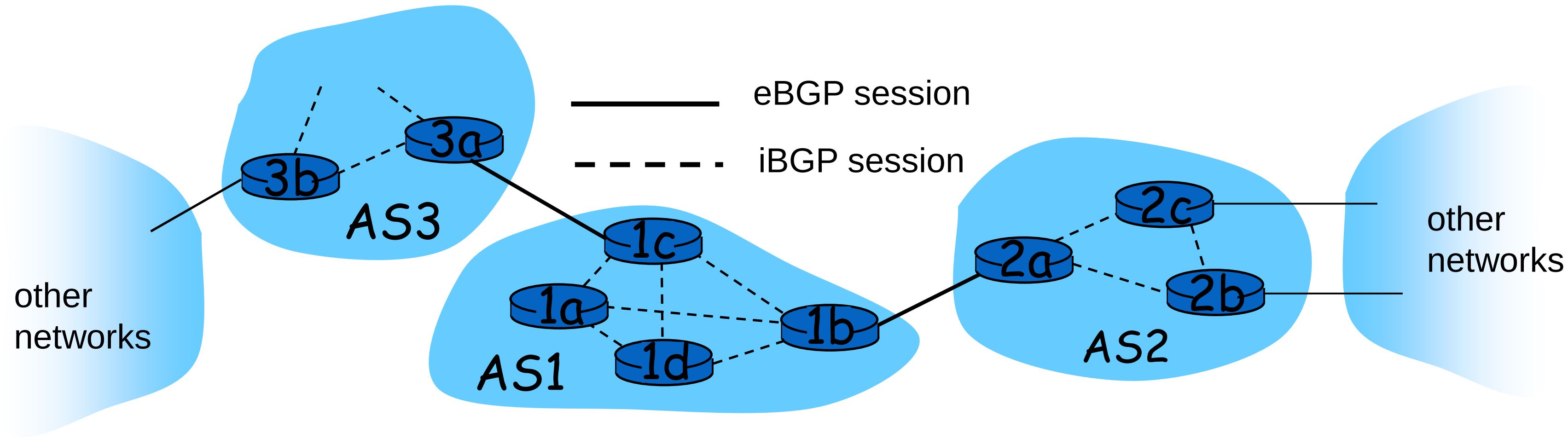
- **BGP session:** two BGP routers ("peers") exchange BGP msg:
  - advertising *paths* to different destination network prefixes ("path vector" protocol)
  - exchanged over semi-permanent TCP connections

- when AS3 advertises a prefix to AS1:
  - AS3 *promises* that it will forward datagrams towards that prefix
  - AS3 can aggregate prefixes in its advertisement



BGP basics: distributing path information

- using eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.
  - 1c can then use iBGP to distribute new prefix info to all other BGP routers in AS1 (Note: with all-to-all TCP conns.)
  - 1b can then re-advertise new reachability info to AS2 over 1b-to-2a eBGP session
- when a router learns of a new prefix, it creates an entry for the prefix in its routing table.



Advertisement of "paths to destination nets" between neighboring AS's

- Recall: Dest. Nets are represented by prefixes.

Route (Path) = prefix + attributes

- Two important attributes:
  - AS-PATH: a sequence of ASs through which prefix advertisement has passed: e.g., AS 67, AS 17, AS 205, …
  - NEXT-HOP: indicates specific router IP addr in next-hop AS.

Advertisement of "paths to destination nets" between neighboring AS's

- gateway (BGP) router receiving route advt. uses import policy to accept/decline
  - Example: never route through AS 75

You receive an AS-PATH to dest. X: AS 12, AS 215, AS 75, AS 99 (X)

You do not trust AS 75…….

So you decline the above path to X

Remember

BGP prefers policy-based routing
as opposed to
cost-based routing in OSPF and RIP

BGP route selection

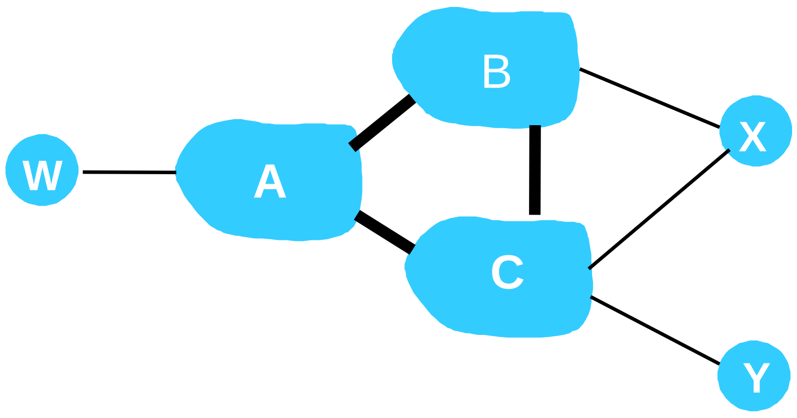
- router may learn about more than 1 route to destination AS; it selects a route based on:

1. **policy decision**
2. shortest AS-PATH

BGP messages: exchanged between peers over TCP conn.

- BGP messages:
  - OPEN
    - opens TCP connection to peer and authenticates sender
  - UPDATE: advertises new path (or withdraws old)
  - KEEPALIVE: keeps connection alive in absence of UPDATES
  - NOTIFICATION: reports errors in previous msg; also used to close connection

BGP routing policy



- X is dual-homed: attached to two networks (B and C)
  - if X does **not** want to route from B to C

    X will **not** advertise to B a route to C

Essentially, X does not tell B that it (X) has a path to C…..

Why different Intra- and Inter-AS routing ?

Policy:

- Inter-AS: admin wants control over how its traffic is routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

Scale:

- hierarchical routing saves table size, reduces update traffic

Performance:

- Intra-AS: focuses on performance
- Inter-AS: policy may dominate performance

Multi-hop communication done!

End of Network Layer