

Announcement

- Project 3 out, due 3/10
- Homework 3 out last week
 - Due next Mon. 3/1

Review

- Hierarchical Routing
- The Internet (IP) Protocol
 - IPv4 addressing
 - Moving a datagram from source to destination

Overview

- The Internet (IP) Protocol
 - Datagram format
 - IP fragmentation
 - ICMP: Internet Control Message Protocol
 - NAT: Network Address Translation
- Routing in the Internet
 - Intra-AS routing: RIP and OSPF
 - Inter-AS routing: BGP
- Multicast Routing

Getting a datagram from source to dest.

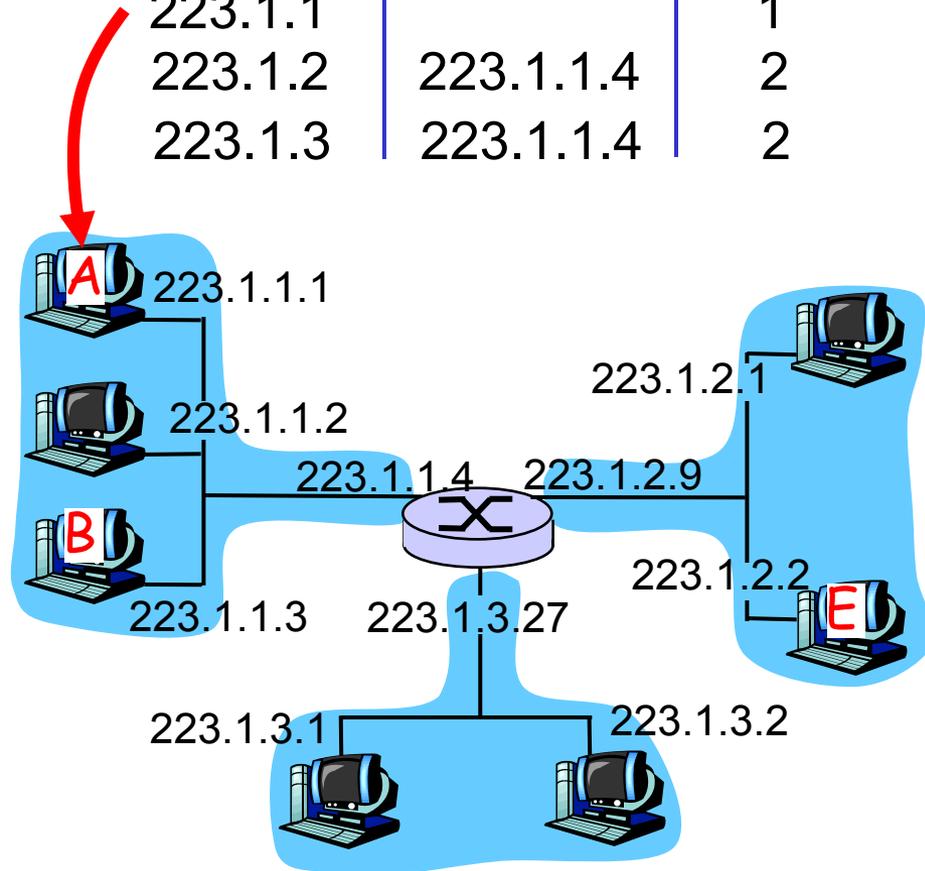
IP datagram:

| | | | |
|-------------|----------------|--------------|------|
| misc fields | source IP addr | dest IP addr | data |
|-------------|----------------|--------------|------|

- ❑ datagram remains **unchanged**, as it travels source to destination
- ❑ addr fields of interest here

forwarding table in A

| Dest. Net. | next router | Nhops |
|------------|-------------|-------|
| 223.1.1 | | 1 |
| 223.1.2 | 223.1.1.4 | 2 |
| 223.1.3 | 223.1.1.4 | 2 |



Getting a datagram from source to dest.

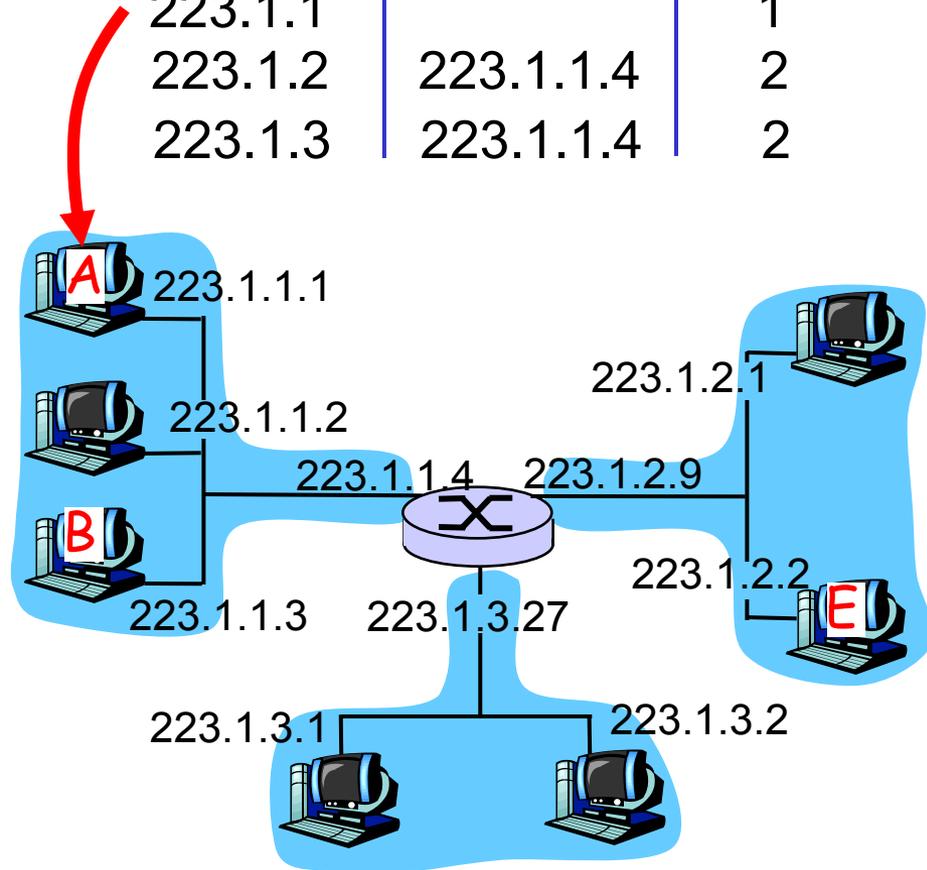
| | | | |
|-------------|-----------|-----------|------|
| misc fields | 223.1.1.1 | 223.1.1.3 | data |
|-------------|-----------|-----------|------|

Starting at A, send IP datagram addressed to B:

- look up net. address of B in forwarding table
- find B is on same net. as A
- link layer will send datagram directly to B inside link-layer frame
 - B and A are directly connected

forwarding table in A

| Dest. Net. | next router | Nhops |
|------------|-------------|-------|
| 223.1.1 | | 1 |
| 223.1.2 | 223.1.1.4 | 2 |
| 223.1.3 | 223.1.1.4 | 2 |



Getting a datagram from source to dest.

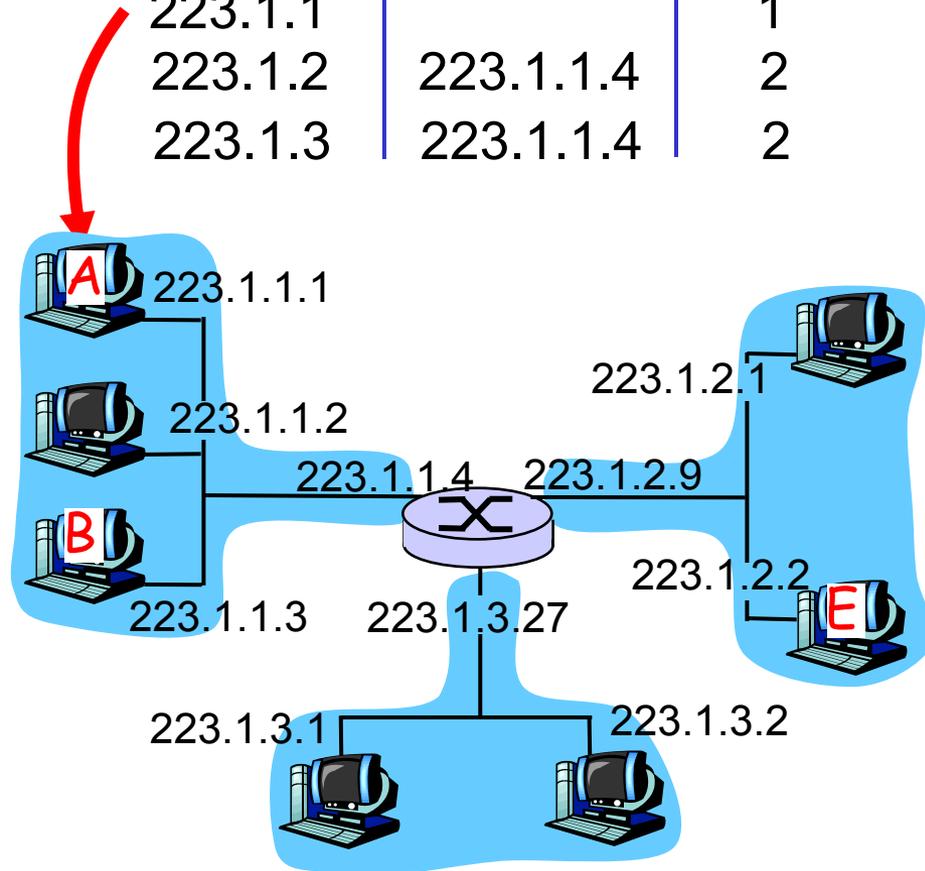
| | | | |
|-------------|-----------|-----------|------|
| misc fields | 223.1.1.1 | 223.1.2.3 | data |
|-------------|-----------|-----------|------|

Starting at A, dest. E:

- ❑ look up network address of E in forwarding table
- ❑ E on *different* network
 - A, E not directly attached
- ❑ routing table: next hop router to E is 223.1.1.4
- ❑ link layer sends datagram to router 223.1.1.4 inside link-layer frame
- ❑ datagram arrives at 223.1.1.4
- ❑ continued.....

forwarding table in A

| Dest. Net. | next router | Nhops |
|------------|-------------|-------|
| 223.1.1 | | 1 |
| 223.1.2 | 223.1.1.4 | 2 |
| 223.1.3 | 223.1.1.4 | 2 |



Getting a datagram from source to dest.

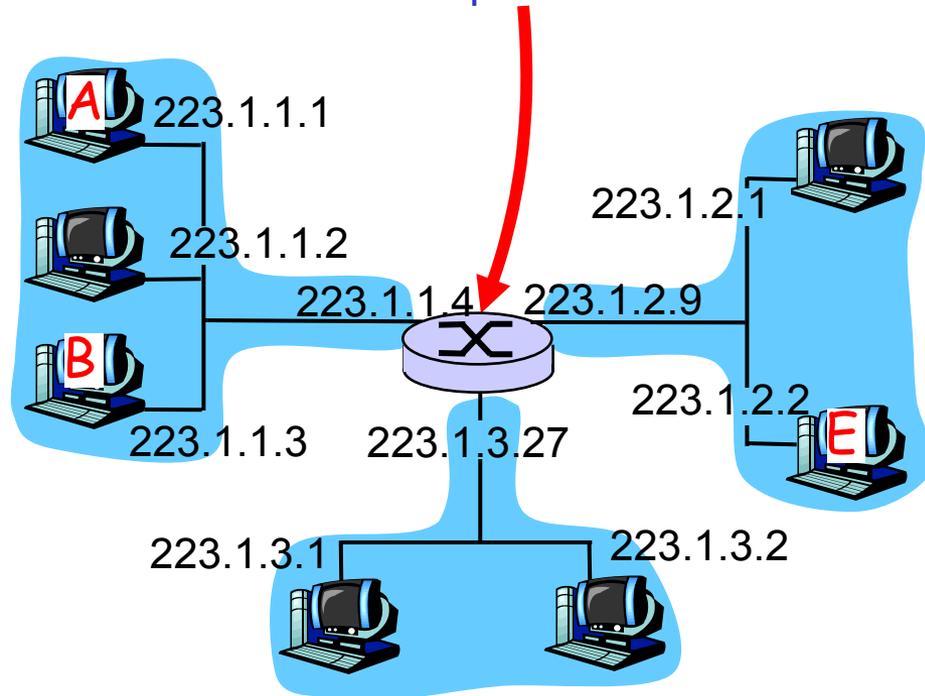
| | | | |
|-------------|-----------|-----------|------|
| misc fields | 223.1.1.1 | 223.1.2.3 | data |
|-------------|-----------|-----------|------|

Arriving at 223.1.4,
destined for 223.1.2.2

- ❑ look up network address of E in router's forwarding table
- ❑ E on *same* network as router's interface 223.1.2.9
 - router, E directly attached
- ❑ link layer sends datagram to 223.1.2.2 inside link-layer frame via interface 223.1.2.9
- ❑ datagram arrives at 223.1.2.2!!! (hooray!)

forwarding table in router

| Dest. Net | router | Nhops | interface |
|-----------|--------|-------|------------|
| 223.1.1 | - | 1 | 223.1.1.4 |
| 223.1.2 | - | 1 | 223.1.2.9 |
| 223.1.3 | - | 1 | 223.1.3.27 |



IP datagram format

IP protocol version number

header length (bytes)

"type" of data

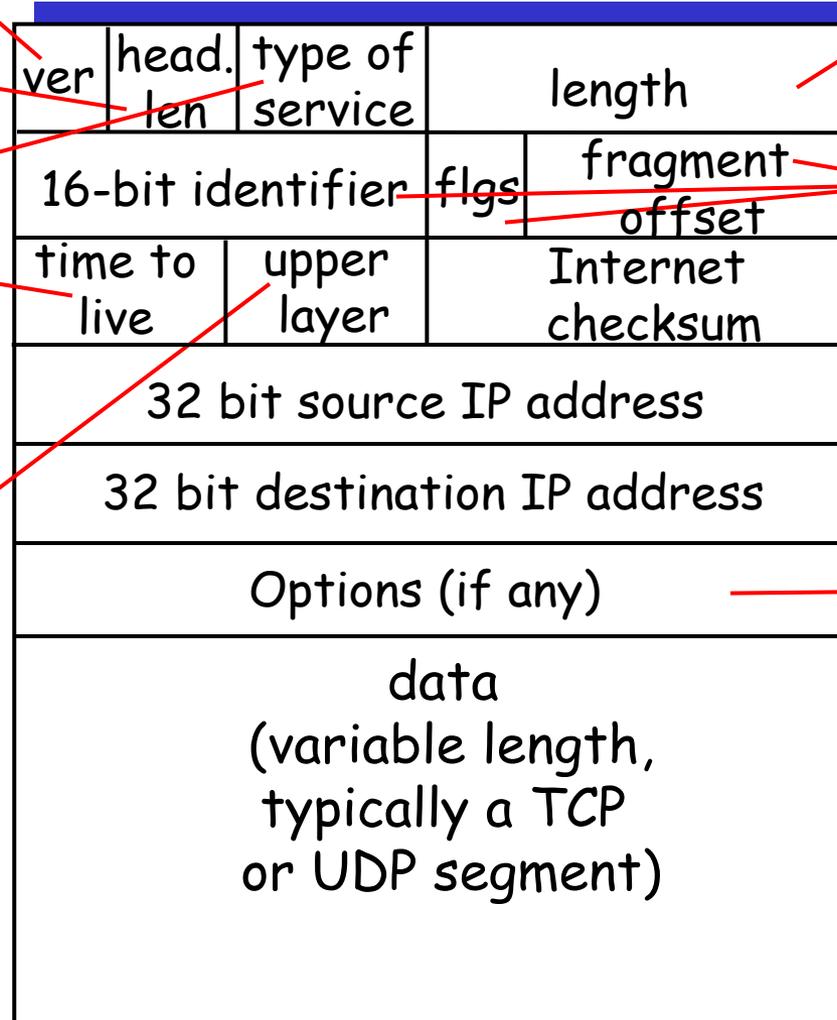
max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to

how much overhead with TCP?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

← 32 bits →



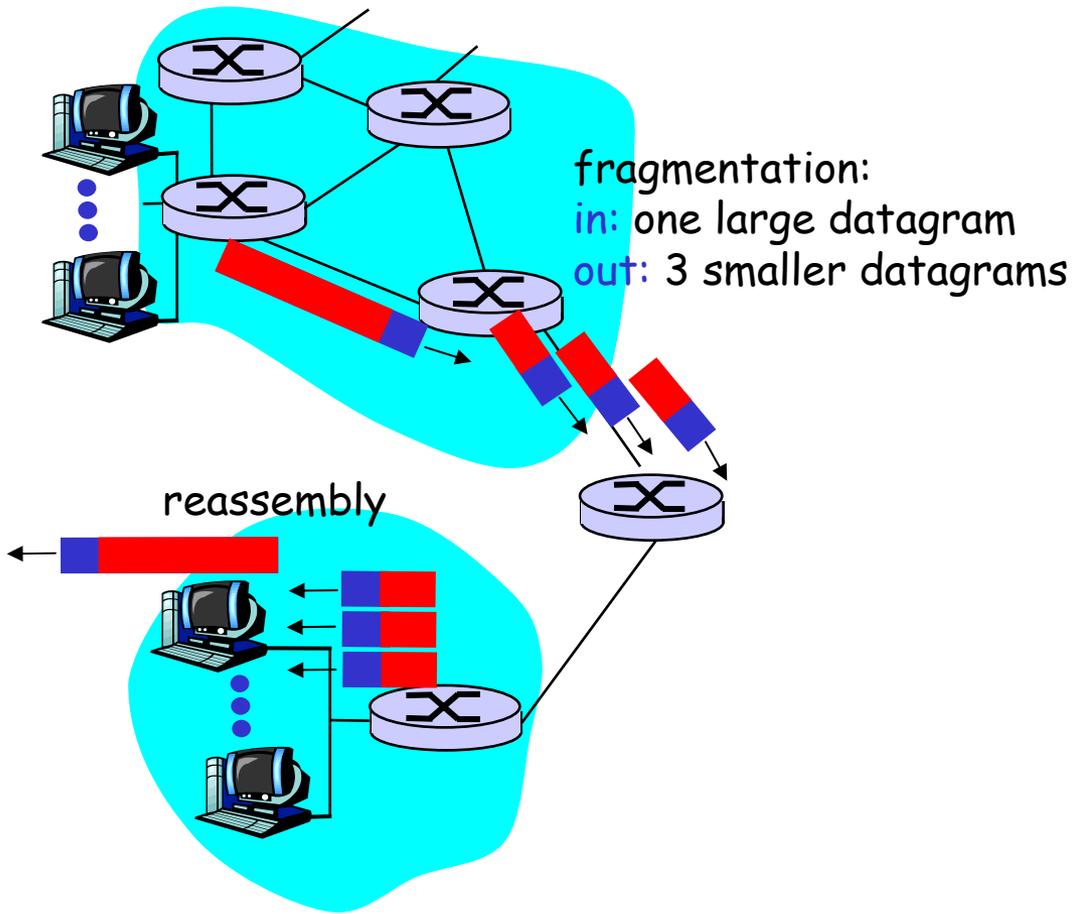
total datagram length (bytes)

for fragmentation/reassembly

E.g. timestamp, record route taken, specify list of routers to visit.

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 and Reassembly

Example

- ❑ 4000 byte datagram
- ❑ MTU = 1500 bytes

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

One large datagram becomes several smaller datagrams

| | | | | | |
|--|--------|----|----------|--------|--|
| | length | ID | fragflag | offset | |
| | =1500 | =x | =1 | =0 | |

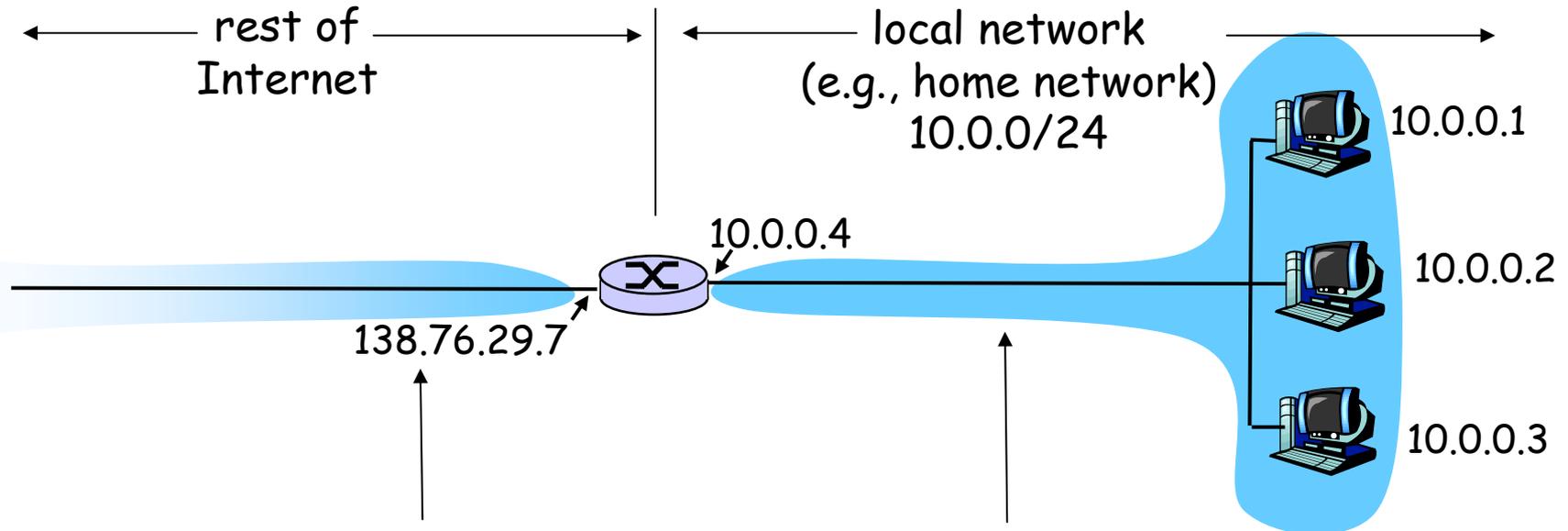
| | | | | | |
|--|--------|----|----------|--------|--|
| | length | ID | fragflag | offset | |
| | =1500 | =x | =1 | =1480 | |

| | | | | | |
|--|--------|----|----------|--------|--|
| | length | ID | fragflag | offset | |
| | =1040 | =x | =0 | =2960 | |

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
- ❑ Ping, traceroute uses ICMP

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

- **Motivation:** local network uses just one IP address as far as outside world is concerned:
 - 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: Network Address Translation

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: Network Address Translation

| NAT translation table | |
|-----------------------|----------------|
| WAN side addr | LAN side addr |
| 138.76.29.7, 5001 | 10.0.0.1, 3345 |
| | |

1: host 10.0.0.1 sends datagram to 128.119.40, 80

S: 10.0.0.1, 3345
D: 128.119.40.186, 80

1



10.0.0.4

S: 128.119.40.186, 80
D: 10.0.0.1, 3345

4

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

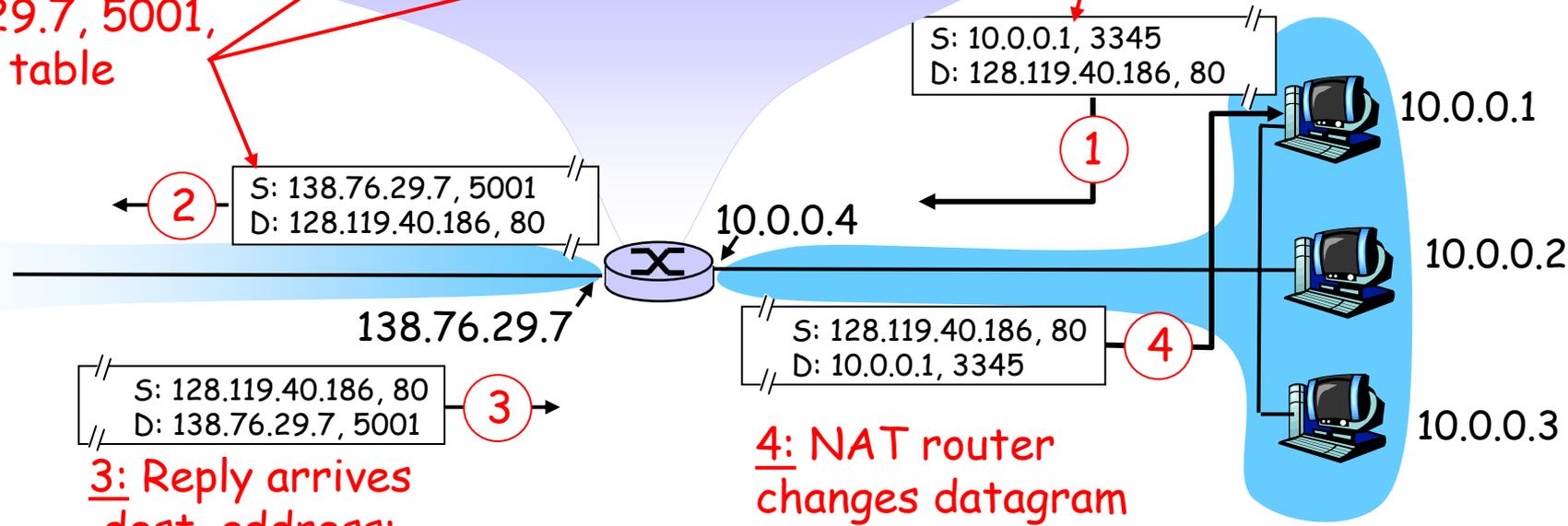
S: 138.76.29.7, 5001
D: 128.119.40.186, 80

2

S: 128.119.40.186, 80
D: 138.76.29.7, 5001

3

3: Reply arrives dest. address: 138.76.29.7, 5001



NAT: Network Address Translation

- ❑ 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

Overview

- The Internet (IP) Protocol
 - Datagram format
 - IP fragmentation
 - ICMP: Internet Control Message Protocol
 - NAT: Network Address Translation
- Routing in the Internet
 - Intra-AS routing: RIP and OSPF
 - Inter-AS routing: BGP
- Multicast Routing

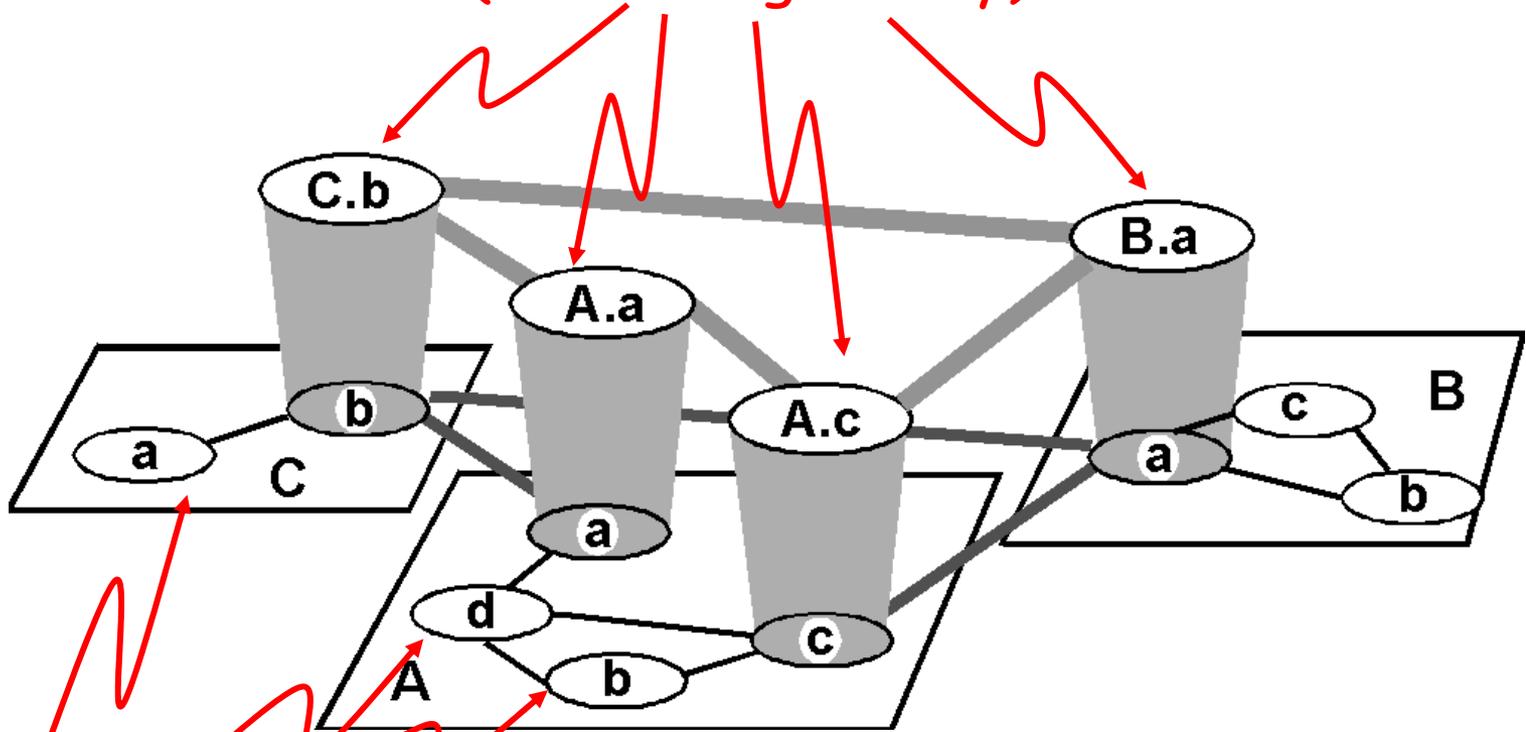
Routing in the Internet

- 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 border (exterior gateway) routers



Inter-AS interior (gateway) routers

Intra-AS Routing

- ❑ Also known as **Interior Gateway Protocols (IGP)**
- ❑ Most common Intra-AS routing protocols:
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First
 - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)

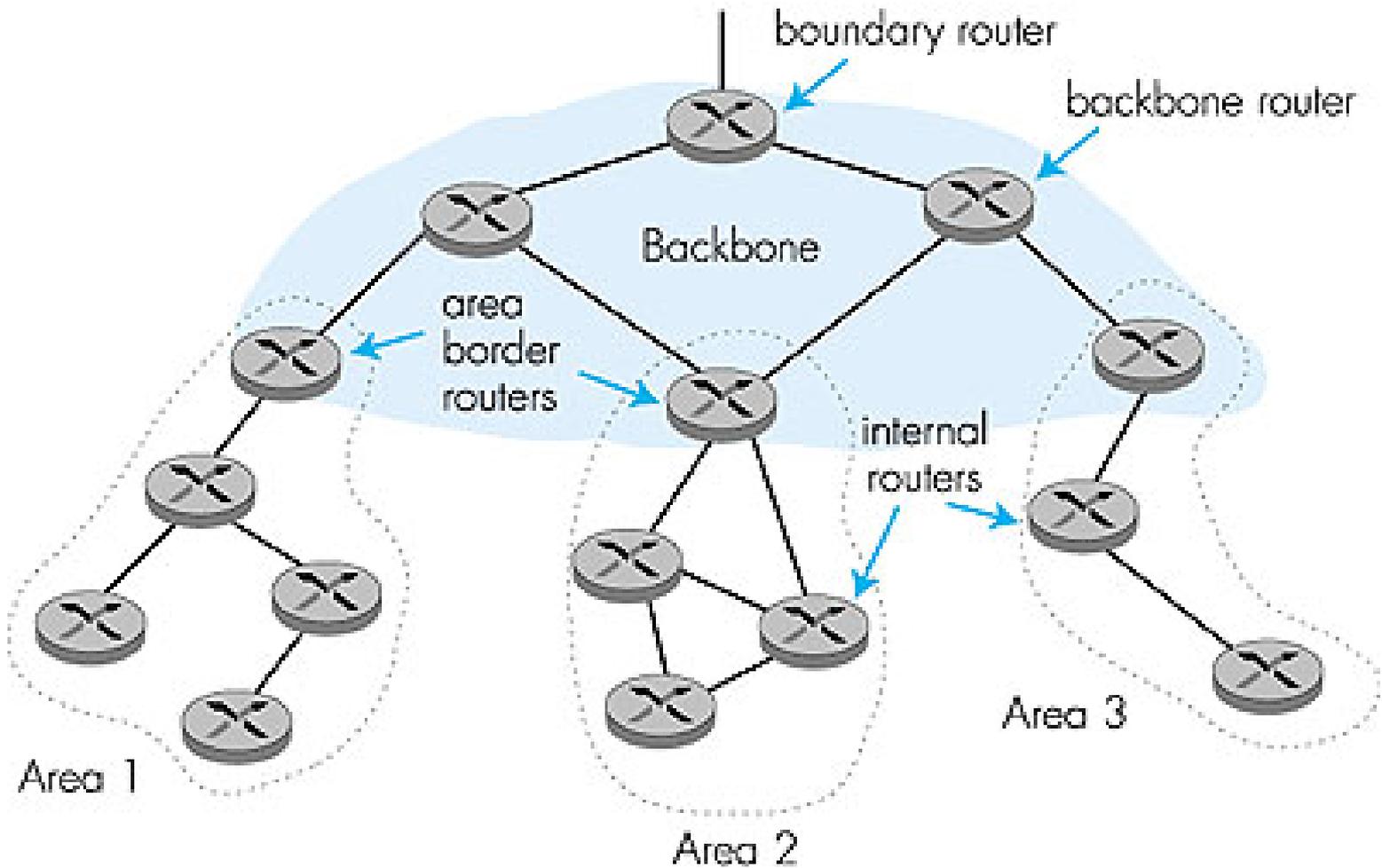
OSPF (Open Shortest Path First)

- ❑ “open”: publicly available
- ❑ Uses Link State algorithm
 - LS packet dissemination
 - Topology map at each node
 - Route computation using Dijkstra’s algorithm
- ❑ OSPF advertisement carries one entry per neighbor router
- ❑ Advertisements disseminated to **entire** AS (via flooding)
 - Carried in OSPF messages directly over IP (rather than TCP or UDP)

OSPF "advanced" features (not in RIP)

- ❑ **Security:** all OSPF messages authenticated (to prevent malicious intrusion)
- ❑ For each link, multiple cost metrics for different **TOS** (e.g., satellite link cost set "low" for best effort; high for real time)
- ❑ Integrated uni- and **multicast** support:
 - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- ❑ **Hierarchical** OSPF in large domains.

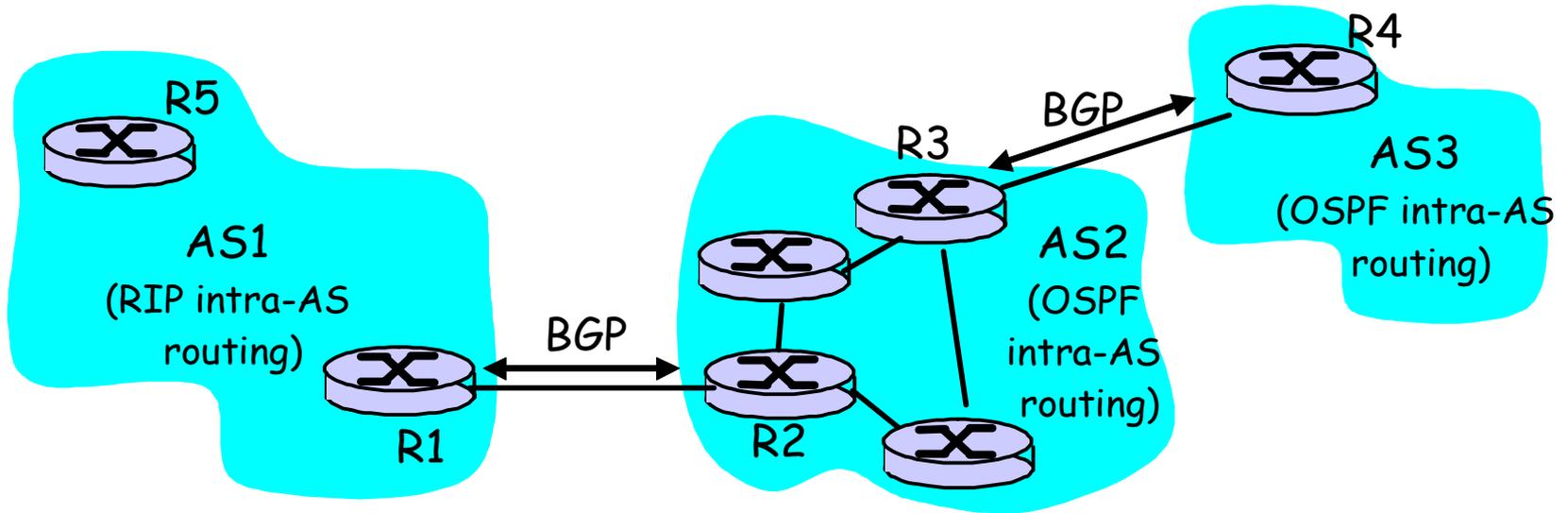
Hierarchical OSPF



Hierarchical OSPF

- ❑ **Two-level hierarchy:** local area, backbone.
 - Link-state advertisements only in area
 - each nodes has detailed area topology; only know direction (shortest path) to nets in other areas.
- ❑ **Area border routers:** "summarize" distances to nets in own area, advertise to other Area Border routers.
- ❑ **Backbone routers:** run OSPF routing limited to backbone.
- ❑ **Boundary routers:** connect to other AS's.

Inter-AS routing in the Internet: BGP



Internet inter-AS routing: BGP

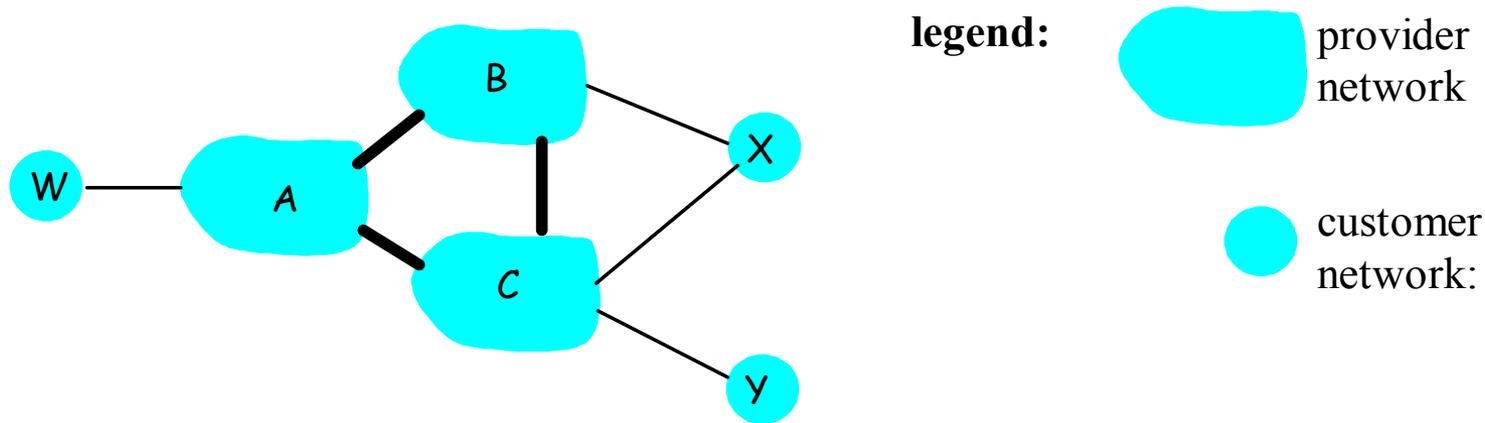
- **BGP (Border Gateway Protocol):** *the de facto standard*
- **Path Vector** protocol:
 - similar to Distance Vector protocol
 - each Border Gateway broadcast to neighbors (peers) *entire path* (i.e., sequence of AS's) to destination
 - BGP routes to networks (ASs), not individual hosts
 - E.g., Gateway X may send its path to dest. Z:

Path (X,Z) = X,Y1,Y2,Y3,...,Z

Internet inter-AS routing: BGP

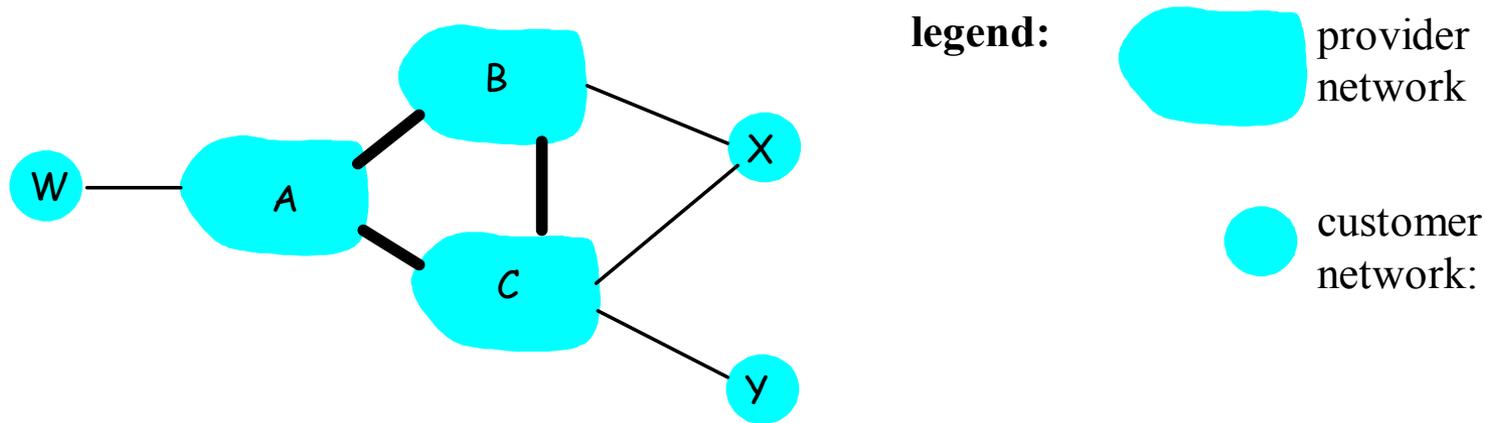
- Suppose:* gateway X send its path to peer gateway W
- W may or may not select path offered by X
 - cost, policy (don't route via competitors AS), loop prevention reasons.
 - If W selects path advertised by X, then:
$$\text{Path}(W,Z) = w, \text{Path}(X,Z)$$
 - Note: X can control incoming traffic by controlling its route advertisements to peers:
 - e.g., don't want to route traffic to Z -> don't advertise any routes to Z

BGP: controlling who routes to you



- A,B,C are **provider networks**
- X,W,Y are customer (of provider networks)
- X is **dual-homed**: attached to two networks
 - X does not want to route from B via X to C
 - .. so X will not advertise to B a route to C

BGP: controlling who routes to you



- ❑ A advertises to B the path AW
- ❑ B advertises to X the path BAW
- ❑ Should B advertise to C the path BAW?
 - No way! B gets no "revenue" for routing CBAW since neither W nor C are B's customers
 - B wants to force C to route to w via A
 - B wants to route *only* to/from its customers!

BGP operation

Q: What does a BGP router do?

- ❑ Receiving and filtering route advertisements from directly attached neighbor(s).
- ❑ Route selection.
 - To route to destination X, which path (of several advertised) will be taken?
- ❑ Sending route advertisements to neighbors.