

# FCM 742 - Network Security

*Network Layer*

*John Jay D4CS Program  
Spring 2015*

slides provided by Prof. Jim Kurose

1

## Chapter 4: Network Layer

### Chapter goals:

- ❖ understand principles behind network layer services:
  - network layer service models
  - forwarding versus routing
  - how a router works
  - routing (path selection)
  - broadcast, multicast
- ❖ instantiation, implementation in the Internet

4-2

## 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
- IPv4 addressing
- ICMP
- IPv6

#### 4.5 Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

#### 4.6 Routing in the Internet

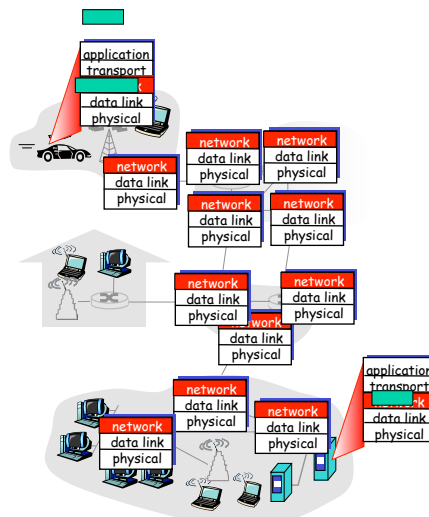
- RIP
- OSPF
- BGP

#### 4.7 Broadcast and multicast routing

4-3

## Network layer

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



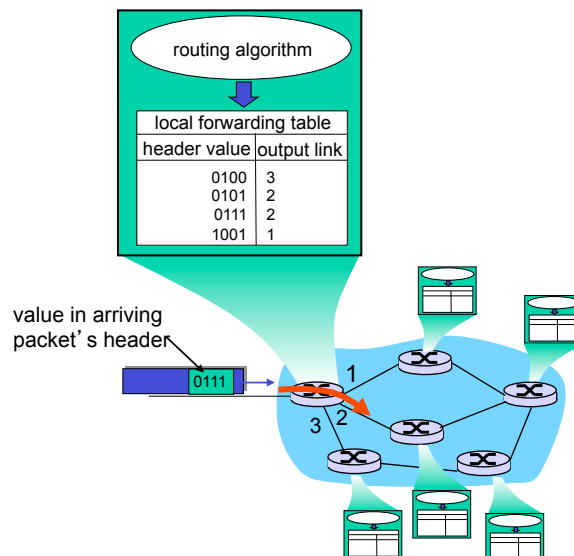
4-4

## 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

4-5

## Interplay between routing and forwarding



4-6

## 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
- 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

4-7

## 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
- IPv4 addressing
- ICMP
- IPv6

### 4.5 Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

### 4.6 Routing in the Internet

- RIP
- OSPF
- BGP

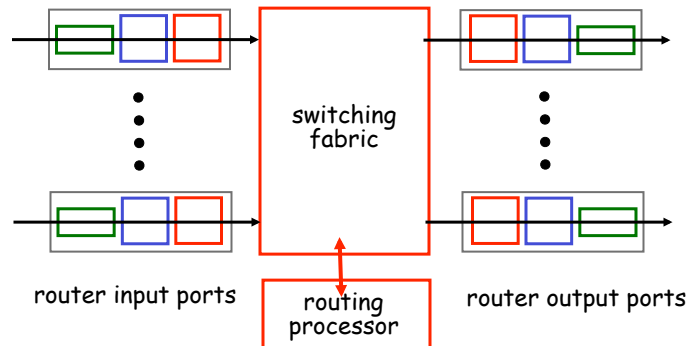
### 4.7 Broadcast and multicast routing

4-8

## Router Architecture Overview

two key router functions:

- ❖ run routing algorithms/protocol (RIP, OSPF, BGP)
- ❖ *forwarding* datagrams from incoming to outgoing link



4-9

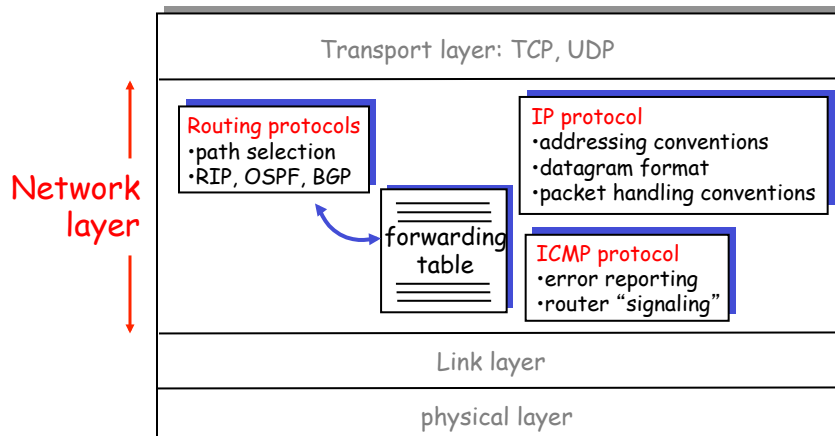
## Chapter 4: Network Layer

- ❖ 4.1 Introduction
- ❖ 4.2 Virtual circuit and datagram networks
- ❖ 4.3 What's inside a router
- ❖ 4.4 IP: Internet Protocol
  - Datagram format
  - 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

4-10

## The Internet Network layer

Host, router network layer functions:



4-11

## 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**
- IPv4 addressing
- ICMP
- IPv6

### 4.5 Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

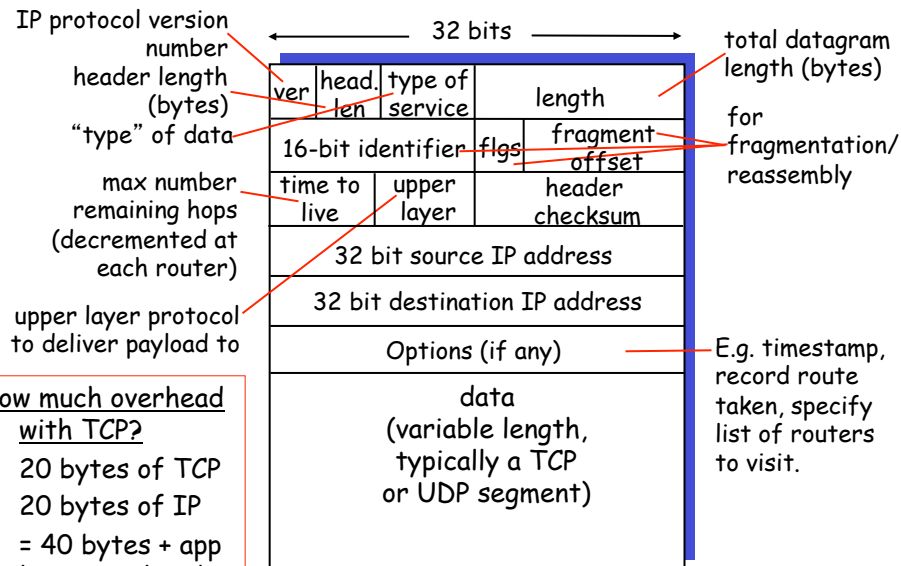
### 4.6 Routing in the Internet

- RIP
- OSPF
- BGP

### 4.7 Broadcast and multicast routing

4-12

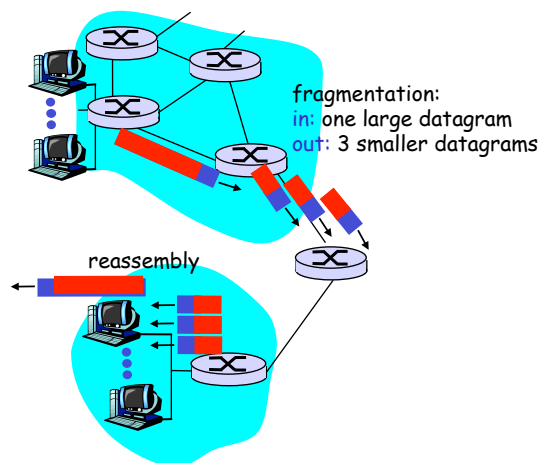
## IP datagram format



4-13

## 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



4-14

## IP Fragmentation and Reassembly

### Example

- ❖ 4000 byte datagram
- ❖ MTU = 1500 bytes

1480 bytes in data field

offset =  
 $1480/8$

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

4-15

## 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
- IPv4 addressing
- ICMP
- IPv6

### 4.5 Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

### 4.6 Routing in the Internet

- RIP
- OSPF
- BGP

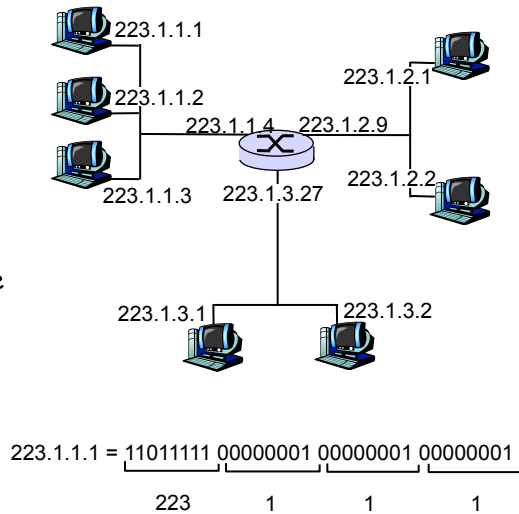
### 4.7 Broadcast and multicast routing

4-16



## IP Addressing: introduction

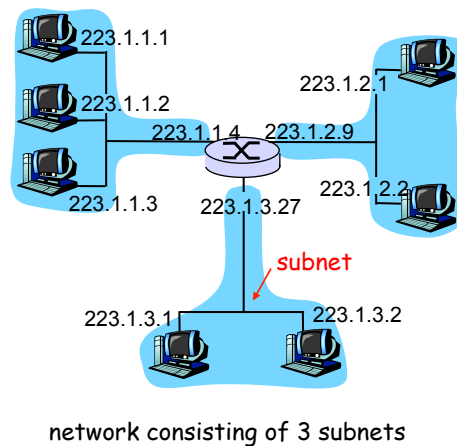
- ❖ **IP address:** 32-bit identifier for host, router *interface*
- ❖ **interface:** connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one interface
  - IP addresses associated with each interface



4-17

## Subnets

- ❖ **IP address:**
  - subnet part (high order bits)
  - 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

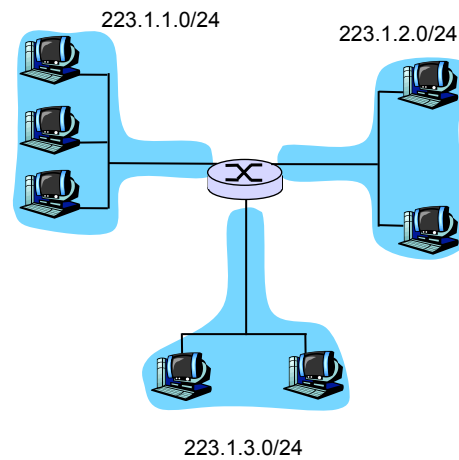


4-18

## Subnets

### Recipe

- ❖ to determine the subnets, detach each interface from its host or router, creating islands of isolated networks
- ❖ each isolated network is called a **subnet**.

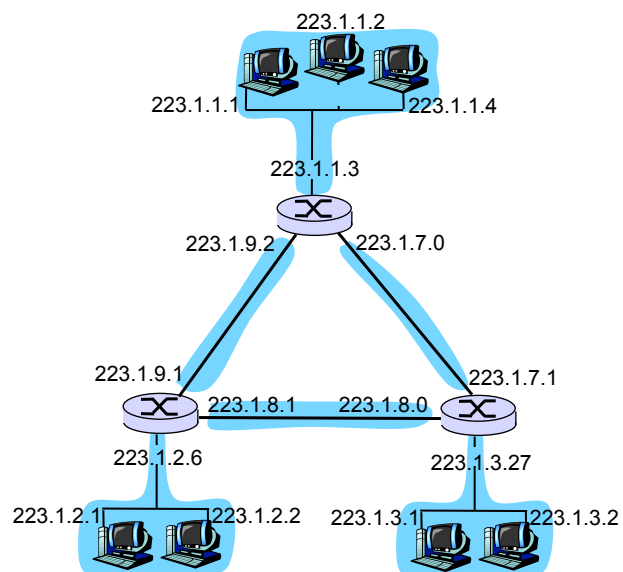


Subnet mask: /24

4-19

## Subnets

How many?

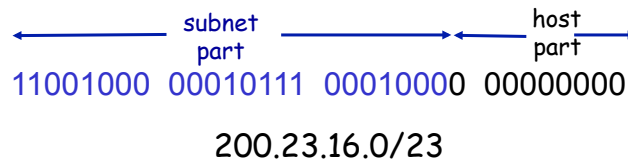


4-20

## IP addressing: CIDR

### **CIDR: Classless InterDomain Routing**

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



4-21

## IP addresses: how to get one?

**Q:** How does a *host* get IP address?

- ❖ hard-coded by system admin in a file
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- ❖ **DHCP: Dynamic Host Configuration Protocol:** dynamically get address from AS server
  - “plug-and-play”

4-22

## 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 an "on")

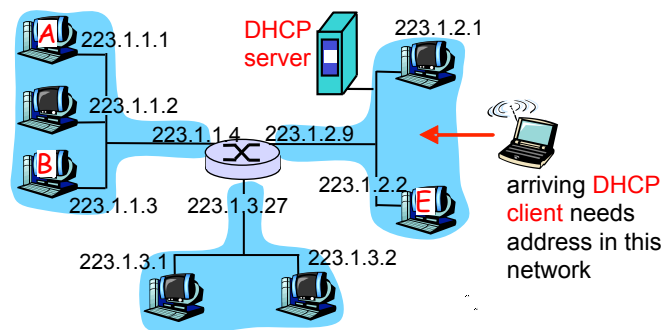
Support for mobile users who want to join network (more shortly)

DHCP overview:

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

4-23

## DHCP client-server scenario



4-24

## DHCP: more than IP address

DHCP can return more than just allocated IP address on subnet:

- address of first-hop router for client
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address)

4-25

## IP addresses: how to get one?

**Q:** How does *network* get subnet part of IP addr?

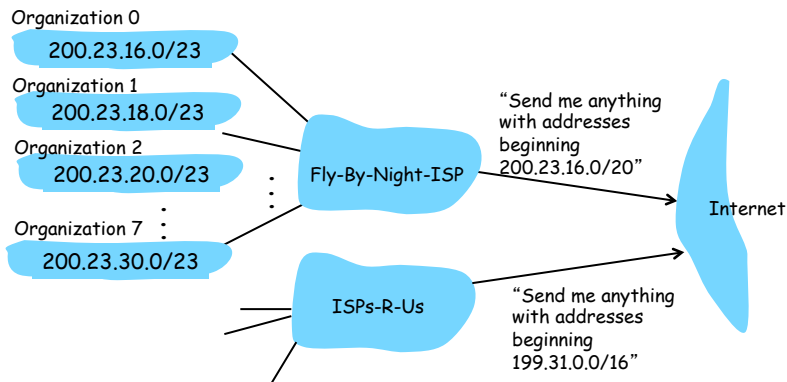
**A:** gets allocated portion of its provider ISP's address space

ISP's block	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/20
Organization 0	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/23
Organization 1	<u>11001000</u>	<u>00010111</u>	<u>00010010</u>	00000000	200.23.18.0/23
Organization 2	<u>11001000</u>	<u>00010111</u>	<u>00010100</u>	00000000	200.23.20.0/23
...	....	....	....	....	....
Organization 7	<u>11001000</u>	<u>00010111</u>	<u>00011110</u>	00000000	200.23.30.0/23

4-26

## Hierarchical addressing: route aggregation

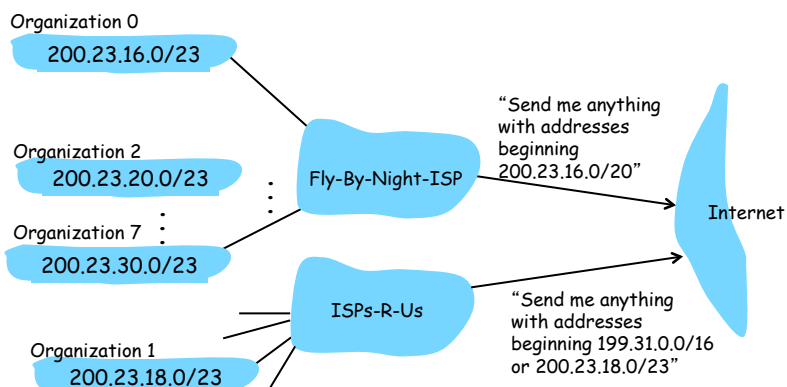
Hierarchical addressing allows efficient advertisement of routing information:



4-27

## Hierarchical addressing: more specific routes

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



4-28

## IP addressing: the last word...

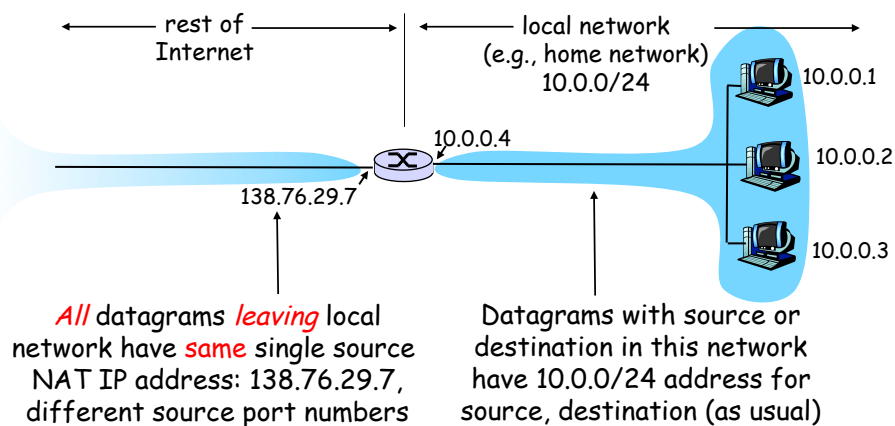
Q: How does an ISP get block of addresses?

A: **ICANN**: **I**nternet **C**orporation for **A**ssigned  
**N**ames and **N**umbers

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

4-29

## NAT: Network Address Translation



4-30

## NAT: Network Address Translation

- ❖ **Motivation:** local network uses just one IP address as far as outside world is concerned:
  - range of addresses not needed from ISP: just one IP address 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).

4-31

## NAT: Network Address Translation

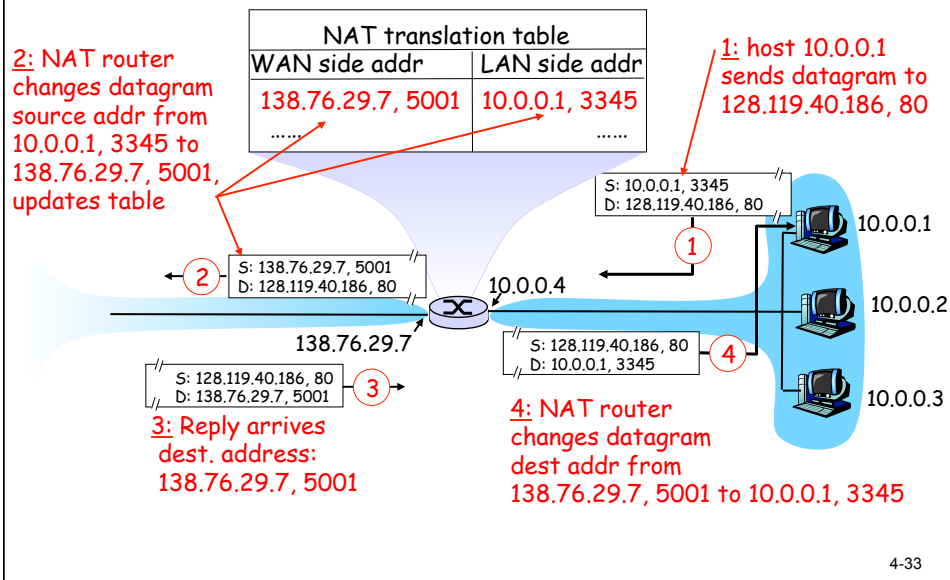
**Implementation:** NAT router must:

- *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

4-32



## NAT: Network Address Translation



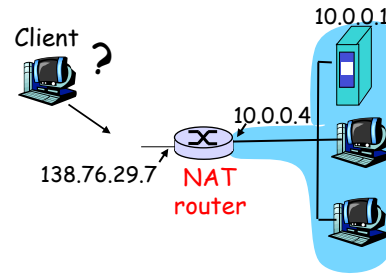
## NAT: Network Address Translation

- ❖ 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, e.g., P2P applications
  - address shortage should instead be solved by IPv6

4-34

## 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
- ❖ solution 1: statically configure NAT to forward incoming connection requests at given port to server
  - e.g., (138.76.29.7, port 2500) always forwarded to 10.0.0.1 port 2500

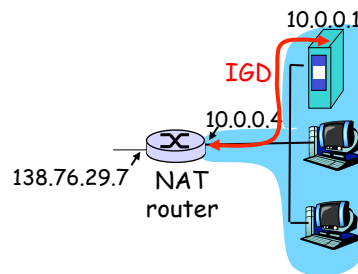


4-35

## 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

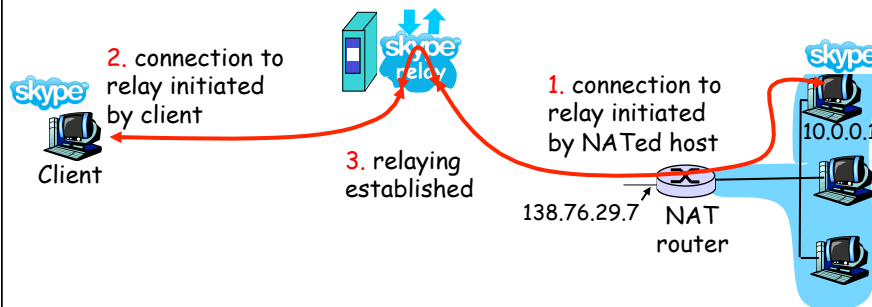


4-36

## 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



4-37

## 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
- 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

4-38

## 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                                 |

4-39

## 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.

4-40

## 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
- 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

4-41

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

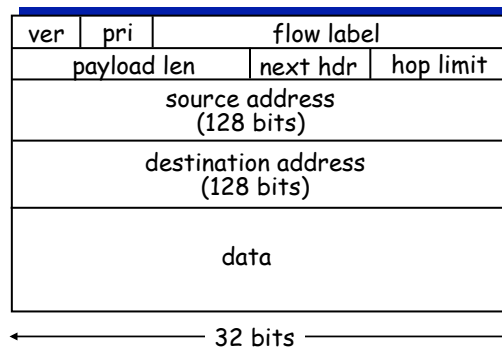
4-42

## 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



4-43

## 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

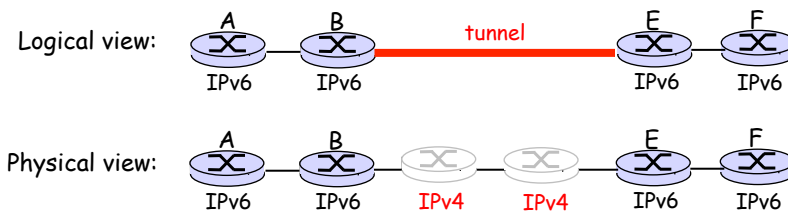
4-44

## 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

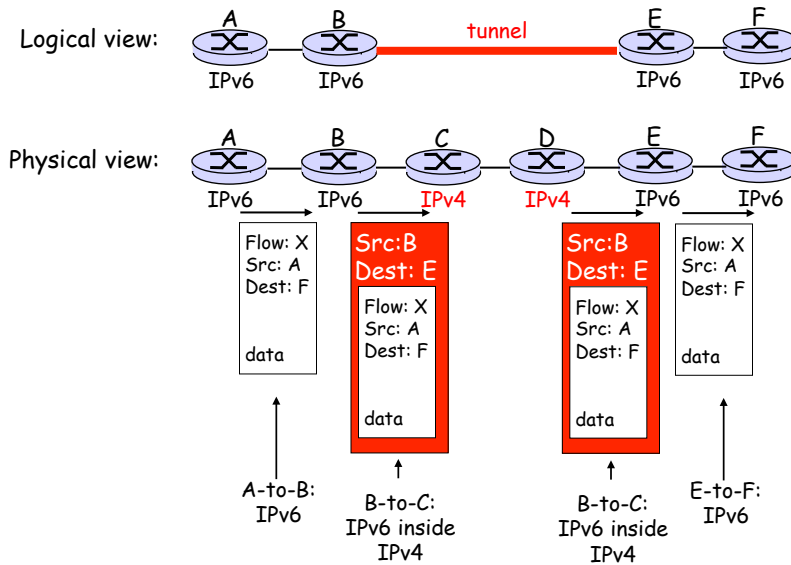
4-45

## Tunneling



4-46

## Tunneling



4-47

## 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
- 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

4-48



## 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
- IPv4 addressing
- ICMP
- IPv6

### 4.5 Routing algorithms

- Link state
- Distance Vector
- Hierarchical routing

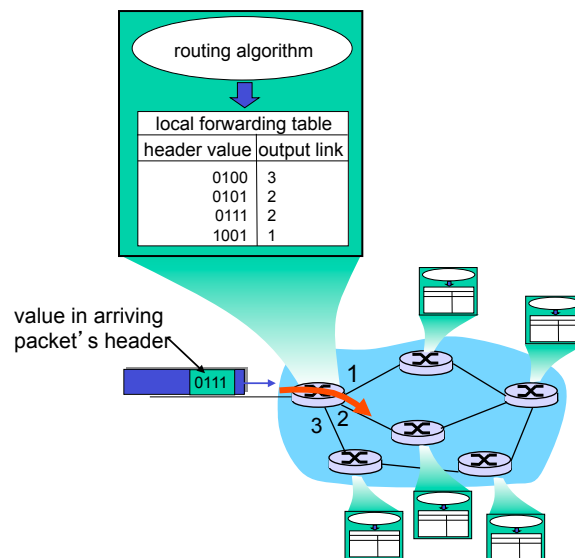
### 4.6 Routing in the Internet

- RIP
- OSPF
- BGP

### 4.7 Broadcast and multicast routing

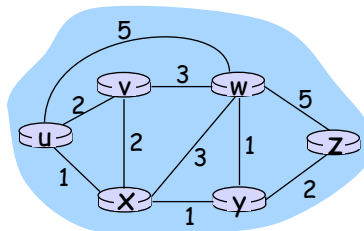
4-49

## Interplay between routing, forwarding



4-50

## Graph abstraction



Graph:  $G = (N, E)$

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

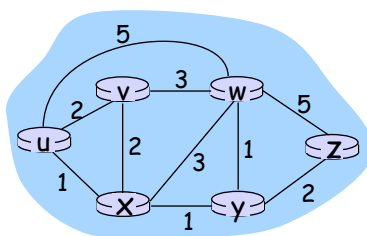
$E$  = set of links =  $\{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

Remark: Graph abstraction is useful in other network contexts

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

4-51

## Graph abstraction: costs



•  $c(x, x') = \text{cost of link } (x, x')$

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

• cost could always be 1, or  
inversely related to bandwidth,  
or inversely related to  
congestion

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

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

Routing algorithm: algorithm that finds least-cost path

4-52

## 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

4-53

## 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
- 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

4-54

## 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
- 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

4-55

## Distance Vector Algorithm

### Bellman-Ford Equation (dynamic programming)

Define

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

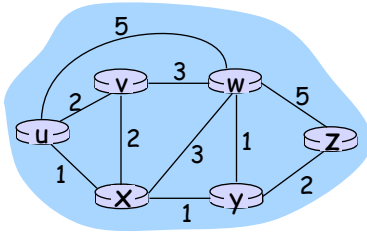
Then

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

where min is taken over all neighbors  $v$  of  $x$

4-56

## Bellman-Ford example



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

B-F equation says:

$$\begin{aligned}
 d_u(z) &= \min \{ c(u,v) + d_v(z), \\
 &\quad c(u,x) + d_x(z), \\
 &\quad c(u,w) + d_w(z) \} \\
 &= \min \{ 2 + 5, \\
 &\quad 1 + 3, \\
 &\quad 5 + 3 \} = 4
 \end{aligned}$$

Node that achieves minimum is next hop in shortest path → forwarding table

4-57

## Distance Vector Algorithm

- ❖  $D_x(y)$  = estimate of least cost from x to y
  - x maintains distance vector  $D_x = [D_x(y): y \in N]$
- ❖ node x:
  - knows cost to each neighbor v:  $c(x,v)$
  - maintains its neighbors' distance vectors. For each neighbor v, x maintains  $D_v = [D_v(y): y \in N]$

4-58

## Distance vector algorithm (cont.)

### Basic idea:

- ❖ from time-to-time, each node sends its own distance vector estimate to neighbors
- ❖ when  $x$  receives new DV estimate from neighbor, it updates its own DV using B-F equation:

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

- ❖ under minor, natural conditions, the estimate  $D_x(y)$  converge to the actual least cost  $d_x(y)$

4-59

## Distance Vector Algorithm (cont.)

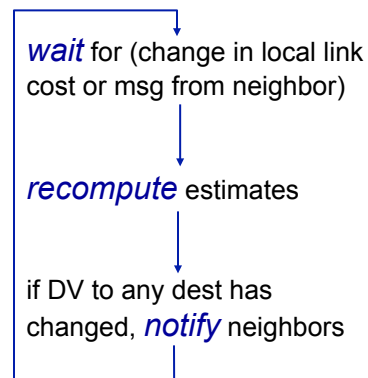
**Iterative, asynchronous:**  
each local iteration caused by:

- ❖ local link cost change
- ❖ DV update message from neighbor

**Distributed:**

- ❖ each node notifies neighbors *only* when its DV changes
  - neighbors then notify their neighbors if necessary

**Each node:**



4-60

## 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
- 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

4-61

## 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

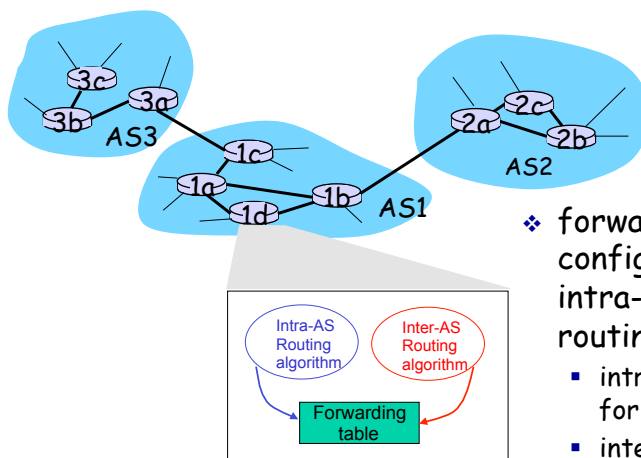
4-62

## 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
- ❖ at “edge” of its own AS
  - ❖ has link to router in another AS

4-63

## Interconnected ASes



- ❖ forwarding table 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

4-64



## Inter-AS tasks

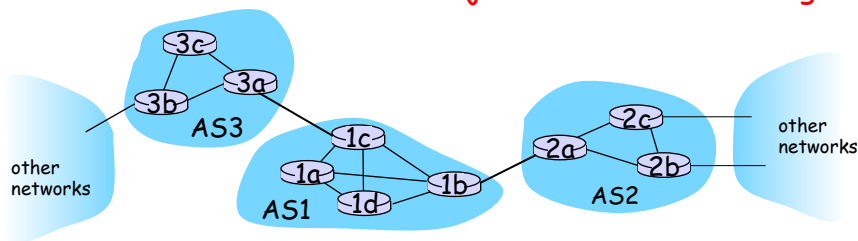
- ❖ suppose router in AS1 receives datagram destined outside of AS1:

- router should forward packet to gateway router, but which one?

### AS1 must:

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

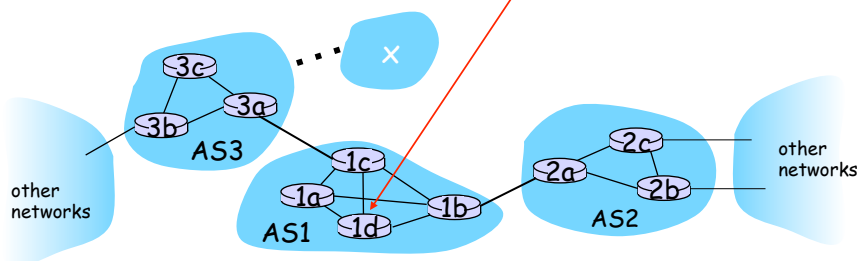
job of inter-AS routing!



4-65

## Example: Setting forwarding table in router 1d

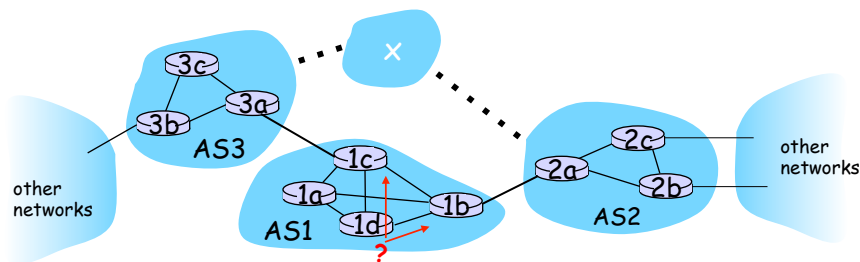
- ❖ suppose AS1 learns (via inter-AS protocol) that subnet  $x$  reachable via AS3 (gateway 1c) but not via AS2.
  - inter-AS protocol propagates reachability info to all internal routers
- ❖ router 1d determines from intra-AS routing info that its interface  $I$  is on the least cost path to 1c.
  - installs forwarding table entry  $(x, I)$



4-66

## Example: Choosing among multiple ASes

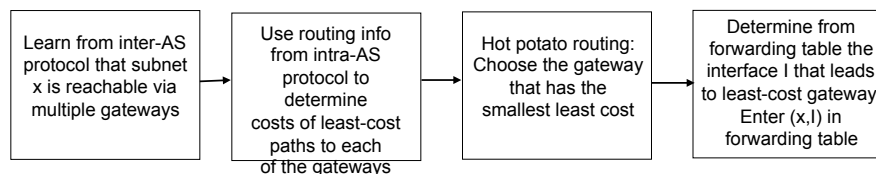
- ❖ now suppose AS1 learns from inter-AS protocol that subnet **x** is reachable from AS3 and from AS2.
- ❖ to configure forwarding table, router 1d must determine which gateway it should forward packets towards for dest **x**
  - this is also job of inter-AS routing protocol!



4-67

## Example: Choosing among multiple ASes

- ❖ now suppose AS1 learns from inter-AS protocol that subnet **x** is reachable from AS3 and from AS2.
- ❖ to configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest **x**.
  - this is also job of inter-AS routing protocol!
- ❖ **hot potato routing**: send packet towards closest of two routers.



4-68

## 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
- 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

4-69

## 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)

4-70

## Internet inter-AS routing: BGP

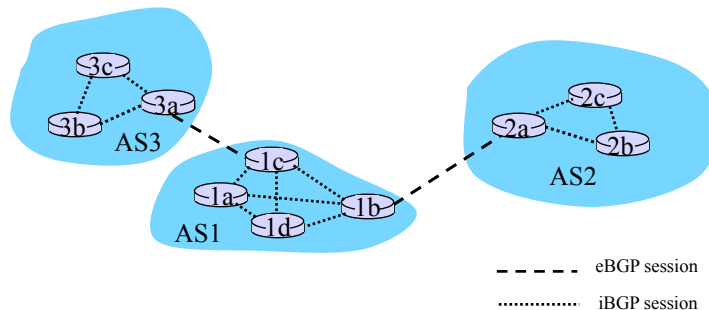
- ❖ **BGP (Border Gateway Protocol):** *the de facto standard*
- ❖ BGP provides each AS a means to:
  1. Obtain subnet reachability information from neighboring ASs.
  2. Propagate the reachability information to all routers internal to the AS.
  3. Determine “good” routes to subnets based on reachability information and policy.
- ❖ Allows a subnet to advertise its existence to rest of the Internet: *“I am here”*

- *BGP Routing Policies in ISP networks, Matthew Caesar and Jennifer Rexford, IEEE Network, Vol 19, Issue 6, 2005*

71

## BGP basics

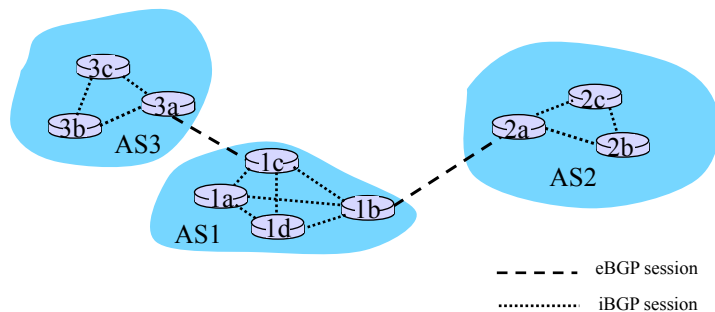
- ❖ Pairs of routers (BGP peers) exchange routing info over semi-permanent TCP connections: **BGP sessions**
- ❖ Note that BGP sessions do not correspond to physical links.
- ❖ When AS2 advertises a prefix to AS1, AS2 is *promising* it will forward any datagrams destined to that prefix towards the prefix.
  - AS2 can aggregate prefixes in its advertisement



72

## Distributing reachability info

- ❖ With eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.
- ❖ 1c can then use iBGP to distribute this new prefix reach info to all routers in AS1
- ❖ 1b can then re-advertise the new reach info to AS2 over the 1b-to-2a eBGP session
- ❖ When router learns about a new prefix, it creates an entry for the prefix in its forwarding table.



73

## Path attributes & BGP routes

- ❖ When advertising a prefix, advert includes BGP attributes.
  - prefix + attributes = "route"
- ❖ Two important attributes:
  - **AS-PATH:** contains the ASs through which the advert for the prefix passed: AS 67 AS 17
  - **NEXT-HOP:** Indicates the specific internal-AS router to next-hop AS. (There may be multiple links from current AS to next-hop-AS.)
- ❖ When gateway router receives route advert, uses **import policy** to accept/decline.

74

## BGP route selection

- ❖ Router may learn about more than 1 route to some prefix. Router must select route.
- ❖ **Elimination rules:**
  1. Local preference value attribute: policy decision
  2. Shortest AS-PATH
  3. Closest NEXT-HOP router: hot potato routing
  4. Additional criteria

75

## BGP messages

- ❖ BGP messages exchanged using TCP.
- ❖ 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; also ACKs OPEN request
  - **NOTIFICATION:** reports errors in previous msg; also used to close connection

76

## Different Intra- and Inter-AS routing

### Policy:

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

### Scale:

- ❖ hierarchical routing saves table size, reduced update traffic

### Performance:

- ❖ Intra-AS: can focus on performance
- ❖ Inter-AS: policy may dominate over performance

### Security:

- ❖ Intra-AS: tight control
- ❖ Inter-AS: very challenging, need to ensure integrity of entire path

77

## AS Relations - Business Relation

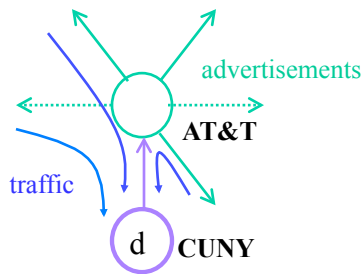
- ❖ Common relationships
  - Customer-provider
  - Peer-peer
  - Backup, sibling, ...
- ❖ Implementing in BGP
  - Import policy
    - Ranking customer routes over peer routes
  - Export policy
    - Export only customer routes to peers and providers

78

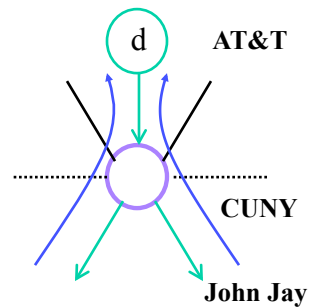
## AS Relation: Customer-Provider

- ❖ Customer pays provider for access to Internet
  - Provider exports customer's routes to everybody
  - Customer exports provider's routes to customers

Traffic **to** the customer



Traffic **from** the customer

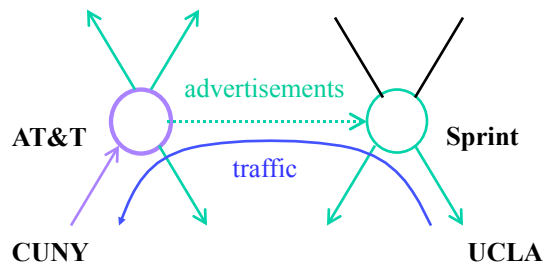


79

## AS Relation: Peer-Peer

- ❖ Peers exchange traffic between customers
  - AS exports *only* customer routes to a peer
  - AS exports a peer's routes *only* to its customers

Traffic to/from the peer and its customers



80



## How Peering Decisions are Made?

### Peer

- ❖ Reduces upstream transit costs
- ❖ Can increase end-to-end performance

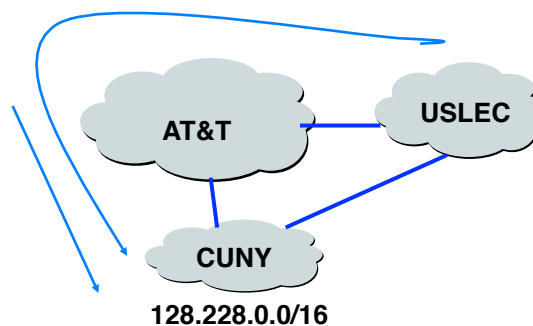
### Don't Peer

- ❖ You would rather have customers
- ❖ Peers are usually your competition
- ❖ Peering relationships may require periodic renegotiation

81

## AS Relation: Backup

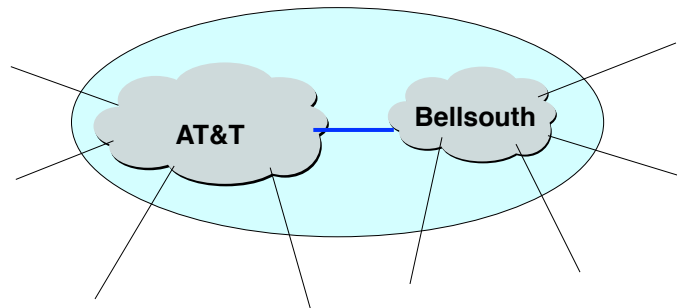
- ❖ Backup provider
  - Only used if the primary link fails
  - Routes through other paths



82

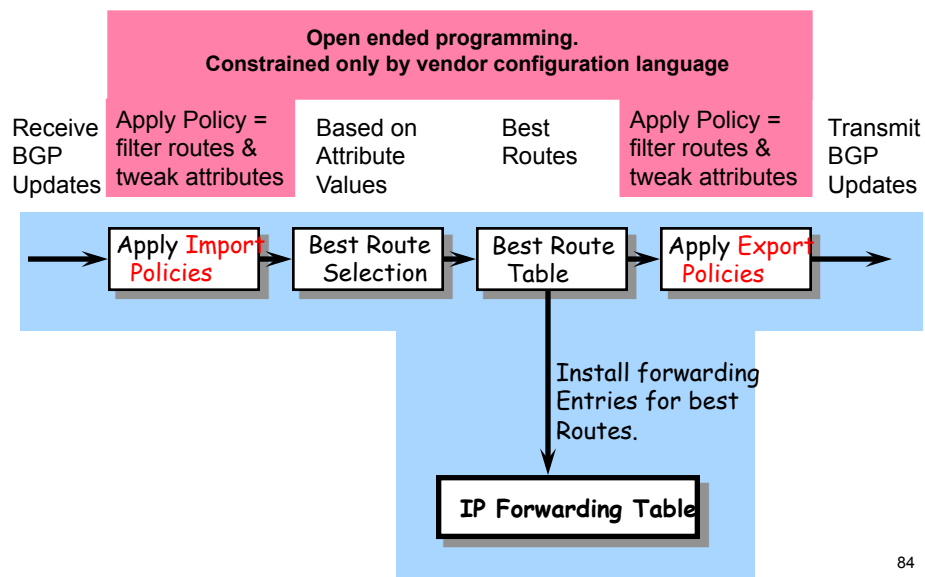
## AS Relation: Sibling

- ❖ Two ASes owned by the same institution
  - E.g., two ASes that have merged
  - E.g., two ASes simply for scaling reasons
  - Essentially act as a single AS



83

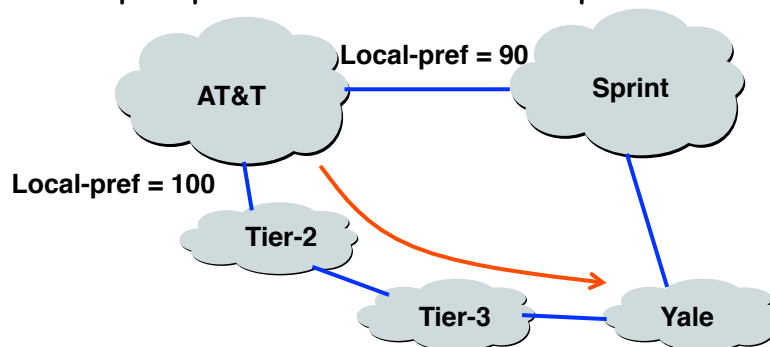
## BGP Policy: Influencing Decisions



84

## Import Policy: Local Preference

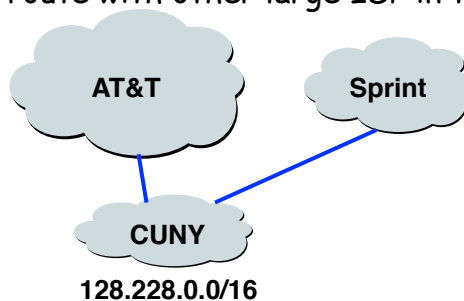
- ❖ Favor one path over another
  - Override the influence of AS path length
  - Apply local policies to prefer a path
- ❖ Example: prefer customer over peer



85

## Import Policy: Filtering

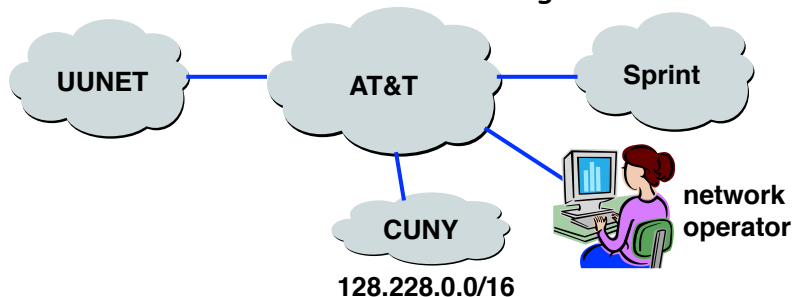
- ❖ Discard some route announcements
  - Detect configuration mistakes and attacks
- ❖ Examples on session to a customer
  - Discard route if prefix not owned by the customer
  - Discard route with other large ISP in the AS path



86

## Export Policy: Filtering

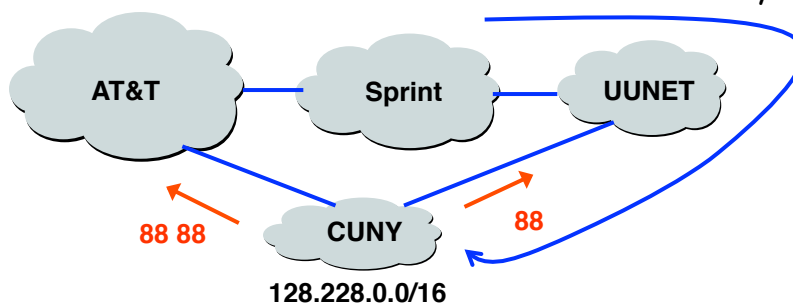
- ❖ Discard some route announcements
  - Limit propagation of routing information
- ❖ Examples
  - Don't announce routes from one peer to another
  - Don't announce routes for management hosts



87

## Export Policy: Attribute Manipulation

- ❖ Modify attributes of the active route
  - To influence the way other ASes behave
- ❖ Example: AS prepending
  - Artificially inflate AS path length seen by others
  - Convince some ASes to send traffic another way



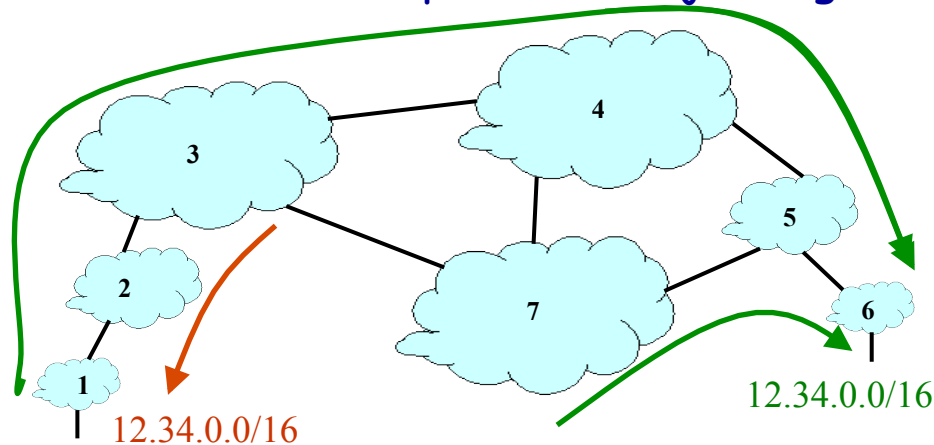
88

## Security Goals for BGP

- ❖ Secure message exchange between neighbors
  - Confidential BGP message exchange
    - Can ASes exchange messages w/o someone watching?
  - No denial of service
    - Prevent overload, session reset, tampered messages?
- ❖ Validity of the routing information
  - Origin authentication
    - Is the prefix owned by the AS announcing it?
  - AS path authentication
    - Is AS path the sequence of ASes the update traversed?
  - AS path policy
    - Does AS path adhere to the routing policies of each AS?

89

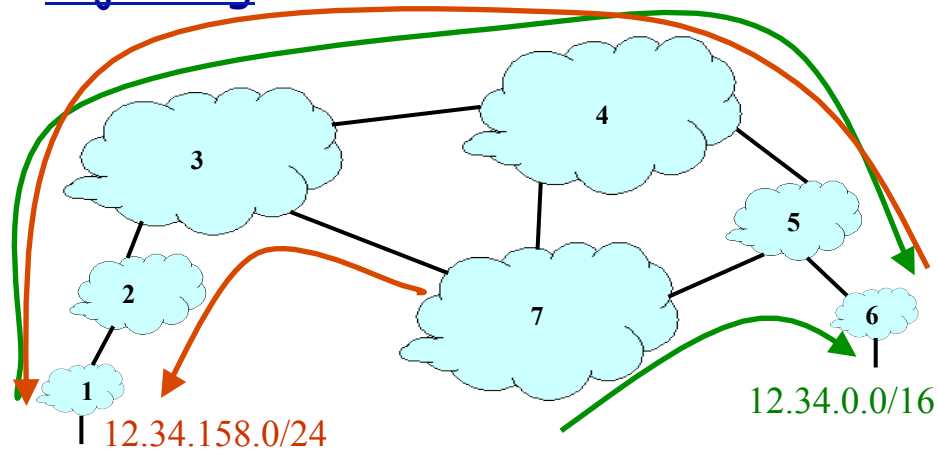
## Address Ownership: Prefix Hijacking



- ❖ Consequences for the affected ASes
  - Blackhole: data traffic is discarded
  - Snooping: data traffic is inspected, and then redirected
  - Impersonation: data traffic is sent to bogus destinations

90

## Address Ownership: Subprefix Hijacking



- ❖ Originating a more-specific prefix
  - Every AS picks the bogus route for that prefix
  - Traffic follows the longest matching prefix

91

## 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
- 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

4-92

## Chapter 4: summary

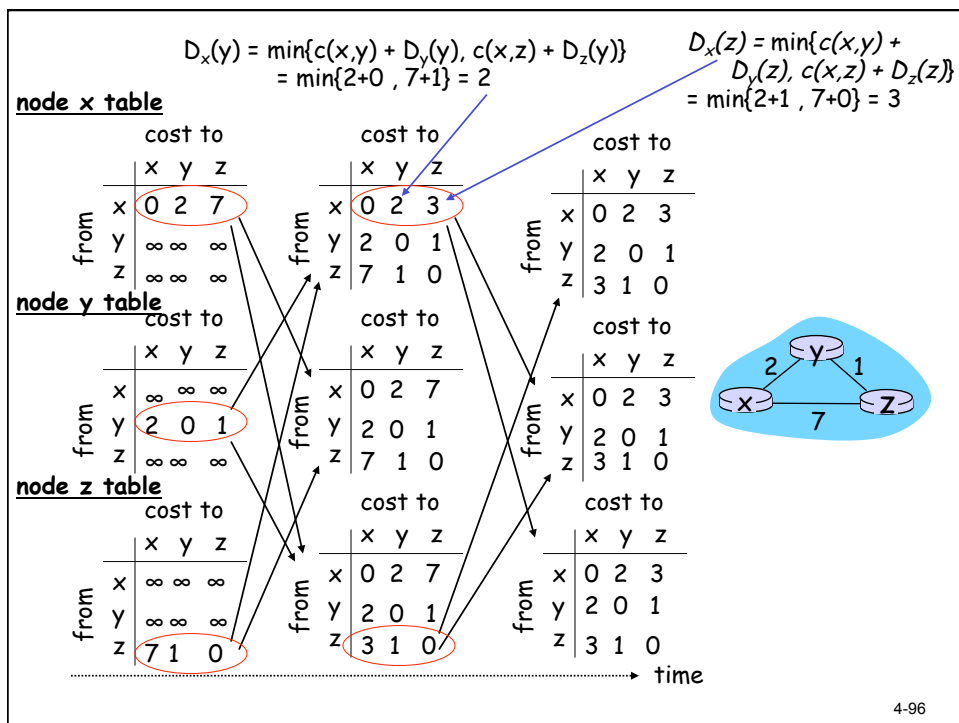
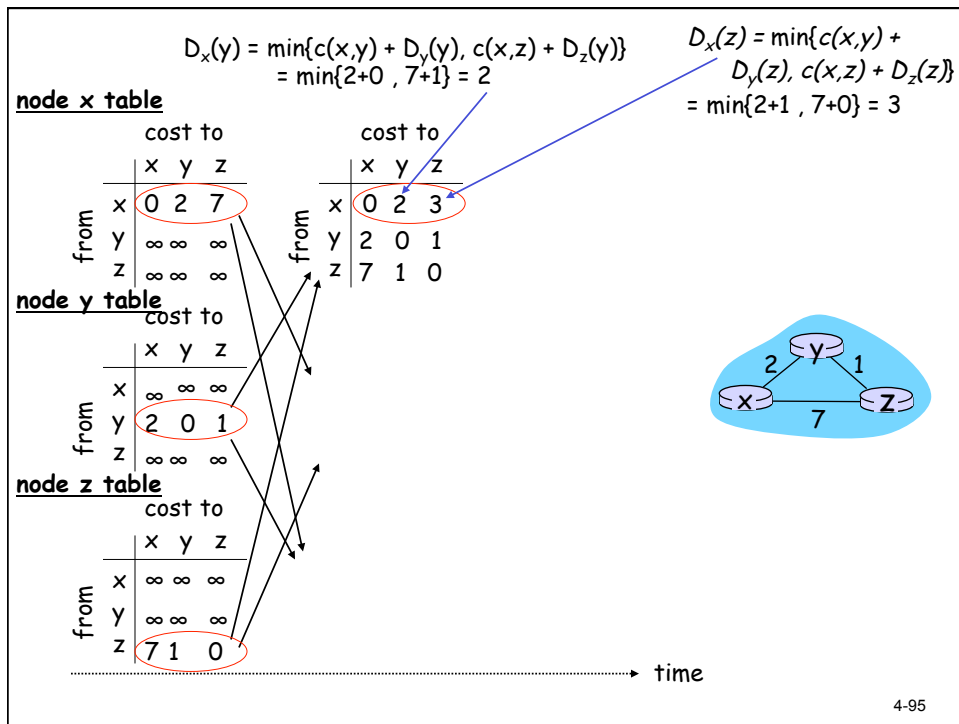
- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

4-93

## Distance Vector Algorithm

- ❖  $D_x(y)$  = estimate of least cost from  $x$  to  $y$ 
  - $x$  maintains distance vector  $D_x = [D_x(y): y \in N]$
- ❖ node  $x$ :
  - knows cost to each neighbor  $v$ :  $c(x,v)$
  - maintains its neighbors' distance vectors.  
For each neighbor  $v$ ,  $x$  maintains  $D_v = [D_v(y): y \in N]$

4-94

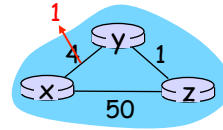




## Distance Vector: link cost changes

### Link cost changes:

- ❖ node detects local link cost change
- ❖ updates routing info, recalculates distance vector
- ❖ if DV changes, notify neighbors



“good  
news  
travels  
fast”

$t_0$ : y detects link-cost change, updates its DV, informs its neighbors.

$t_1$ : z receives update from y, updates its table, computes new least cost to x, sends its neighbors its DV.

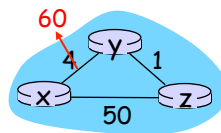
$t_2$ : y receives z's update, updates its distance table. y's least costs do *not* change, so y does *not* send a message to z.

4-97

## Distance Vector: link cost changes

### Link cost changes:

- ❖ good news travels fast
- ❖ bad news travels slow - “count to infinity” problem!
- ❖ 44 iterations before algorithm stabilizes: see text



### Poisoned reverse:

- ❖ If Z routes through Y to get to X :
  - Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
- ❖ will this completely solve count to infinity problem?

4-98

## 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

4-99