

Distance Vector Algorithm

Bellman-Ford Equation (dynamic programming)

Define

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

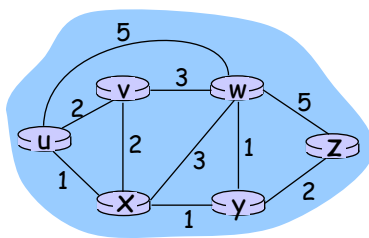
Then

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

where min is taken over all neighbors v of x

Network Layer 4-1

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 \rightarrow forwarding table

Network Layer 4-2

Distance Vector Algorithm

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

Network Layer 4-3

Distance vector algorithm (4)

Basic idea:

- Each node periodically sends its own distance vector estimate to neighbors
- When a node 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 $d_x(y)$

Network Layer 4-4

Distance Vector Algorithm (5)

Iterative, asynchronous:

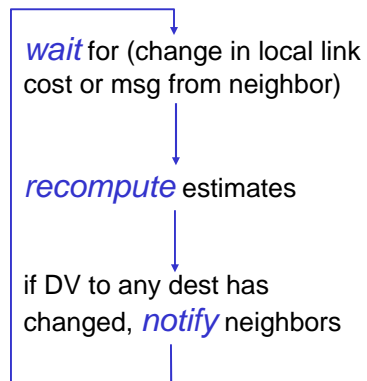
each local iteration caused by:

- local link cost change
- DV update message from neighbor

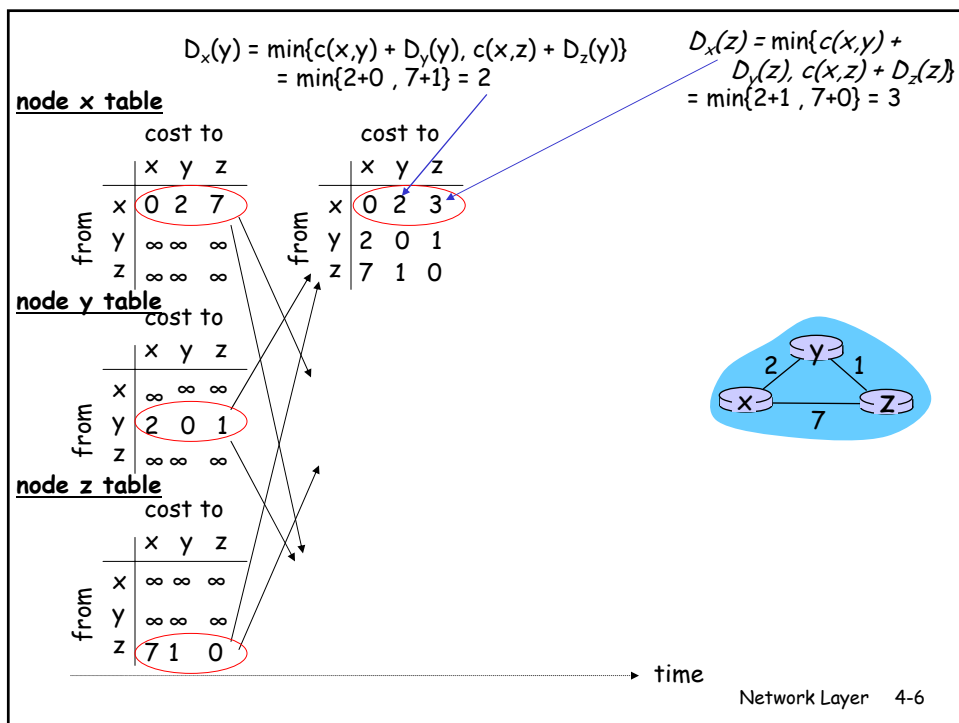
Distributed:

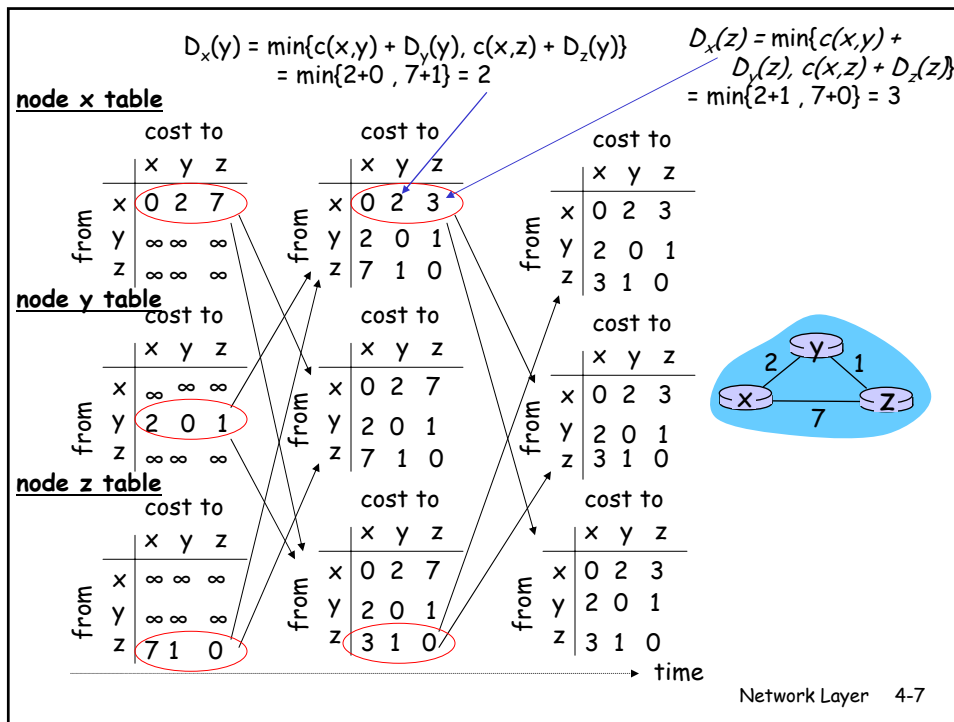
- each node notifies neighbors *only* when its DV changes
 - neighbors then notify their neighbors if necessary

Each node:



Network Layer 4-5

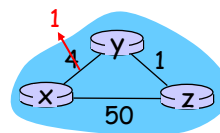




Distance Vector: link cost changes

Link cost changes:

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



"good news travels fast"

At time t_0 , y detects the link-cost change, updates its DV, and informs its neighbors.

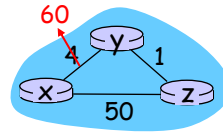
At time t_1 , z receives the update from y and updates its table. It computes a new least cost to x and sends its neighbors its DV

At time t_2 , y receives z 's update and updates its distance table. y 's least costs do not change and hence y does *not* send any message to z .

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?

Network Layer 4-9

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

Network Layer 4-10

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

Network Layer 4-11

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

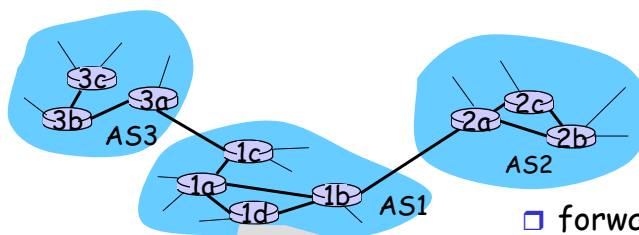
Network Layer 4-12

Hierarchical Routing

- aggregate routers into regions, "autonomous systems" (AS)
 - routers in same AS run same routing protocol
 - "intra-AS" routing protocol
 - routers in different AS can run different intra-AS routing protocol
- Gateway router
- Direct link to router in another AS

Network Layer 4-13

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

Network Layer 4-14

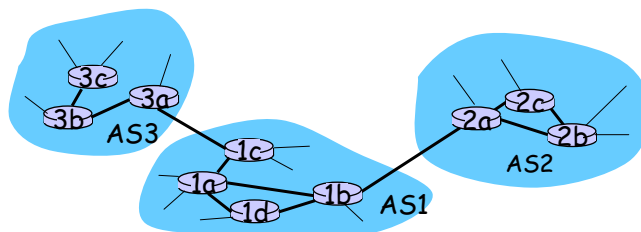
Inter-AS tasks

- suppose router in AS1 receives datagram dest outside of AS1
 - router should forward packet to gateway router, but which one?

AS1 must:

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

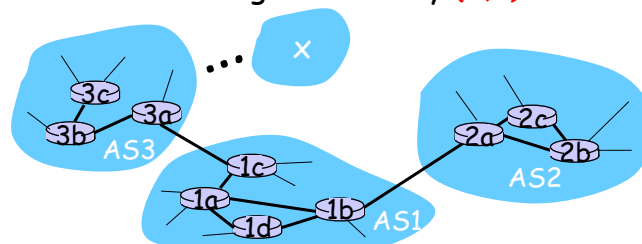
Job of inter-AS routing!



Network Layer 4-15

Example: Setting forwarding table in router 1d

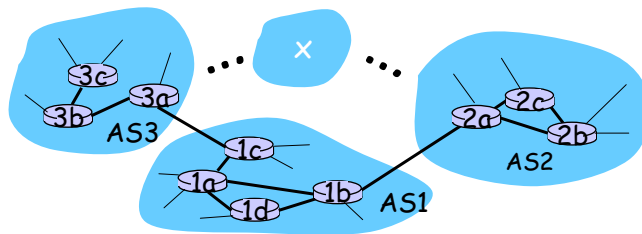
- suppose AS1 learns (via inter-AS protocol) that subnet x reachable via AS3 (gateway 1c) but not via AS2.
- inter-AS protocol propagates reachability info to all internal routers.
- router 1d determines from intra-AS routing info that its interface I is on the least cost path to 1c.
 - installs forwarding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



Network Layer 4-16

Example: Choosing among multiple ASes

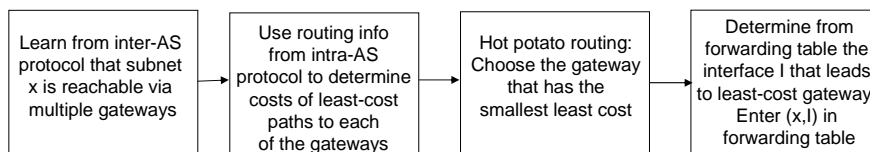
- now suppose AS1 learns from inter-AS protocol that subnet x is reachable from AS3 *and* from AS2.
- to configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest x .
 - this is also job of inter-AS routing protocol!



Network Layer 4-17

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.



Network Layer 4-18

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

Network Layer 4-19

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)

Network Layer 4-20

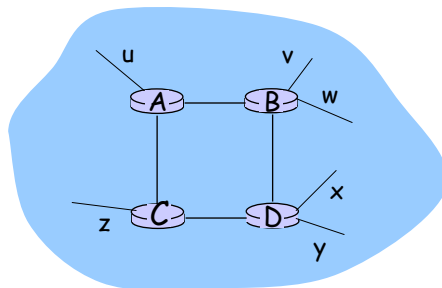
Chapter 4: Network Layer

- ❑ 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

Network Layer 4-21

RIP (Routing Information Protocol)

- ❑ distance vector algorithm
- ❑ included in BSD-UNIX Distribution in 1982
- ❑ distance metric: # of hops (max = 15 hops)



From router A to subsets:

<u>destination</u>	<u>hops</u>
u	1
v	2
w	2
x	3
y	3
z	2

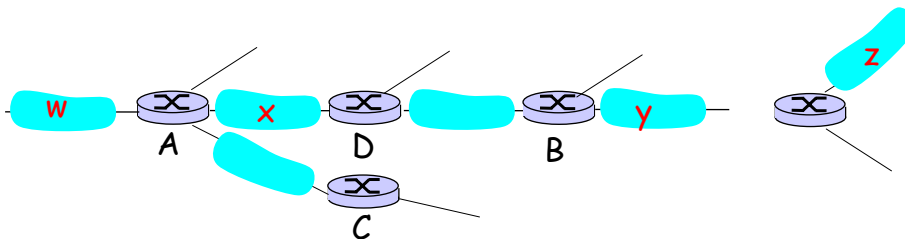
Network Layer 4-22

RIP advertisements

- distance vectors: exchanged among neighbors every 30 sec via Response Message (also called **advertisement**)
- each advertisement: list of up to 25 destination nets within AS

Network Layer 4-23

RIP: Example



Destination Network	Next Router	Num. of hops to dest.
W	A	2
Y	B	2
Z	B	7
X	--	1
...

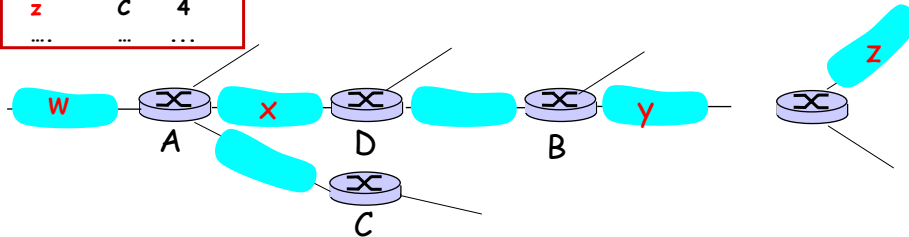
Routing table in D

Network Layer 4-24

RIP: Example

Dest	Next	hops
w	-	1
x	-	1
z	C	4
...

Advertisement from A to D



Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B A	7 5
x	--	1
...

Routing table in D

Network Layer 4-25

RIP: Link Failure and Recovery

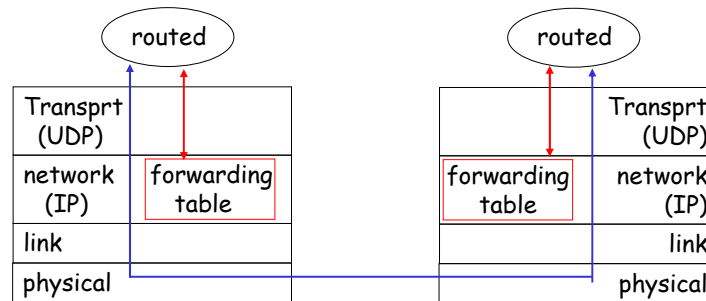
If no advertisement heard after 180 sec --> neighbor/link declared dead

- 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
- *poison reverse* used to prevent ping-pong loops (infinite distance = 16 hops)

Network Layer 4-26

RIP Table processing

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



Network Layer 4-27

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

Network Layer 4-28

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)

Network Layer 4-29

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)
- ❑ 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.

Network Layer 4-30

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

Network Layer 4-31

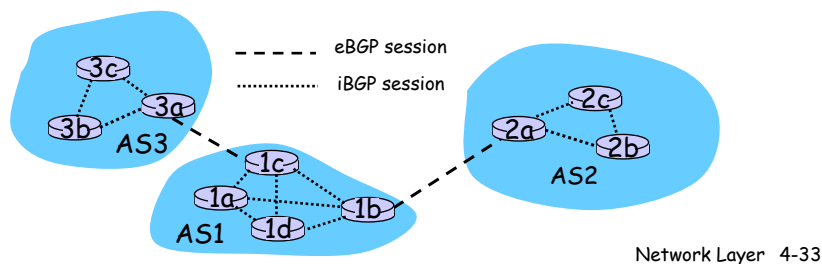
Internet inter-AS routing: BGP

- ❑ **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 reachability information to all AS-internal routers.
 3. Determine "good" routes to subnets based on reachability information and policy.
- ❑ allows subnet to advertise its existence to rest of Internet: *"I am here"*

Network Layer 4-32

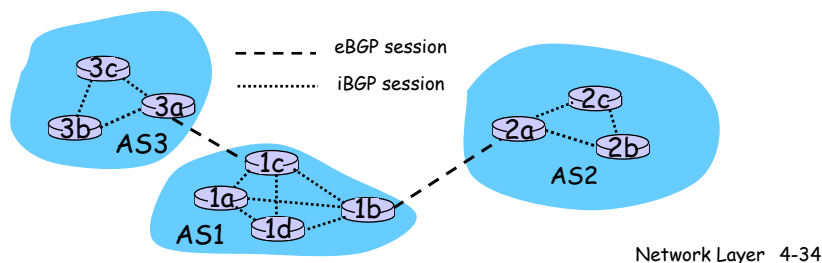
BGP basics

- pairs of routers (BGP peers) exchange routing info over semi-permanent TCP connections: **BGP sessions**
 - BGP sessions need not correspond to physical links.
- when AS2 advertises prefix to AS1:
 - AS2 *promises* it will forward any addresses datagrams towards that prefix.
 - AS2 can aggregate prefixes in its advertisement



Distributing reachability info

- 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 routers in AS1
 - 1b can then re-advertise new reachability info to AS2 over 1b-to-2a eBGP session
- when router learns of new prefix, creates entry for prefix in its forwarding table.



Path attributes & BGP routes

- ❑ advertised prefix includes BGP attributes.
 - prefix + attributes = "route"
- ❑ two important attributes:
 - **AS-PATH**: contains ASs through which prefix advertisement has passed: e.g, AS 67, AS 17
 - **NEXT-HOP**: indicates specific internal-AS router to next-hop AS. (may be multiple links from current AS to next-hop-AS)
- ❑ when gateway router receives route advertisement, uses **import policy** to accept/decline.

Network Layer 4-35

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

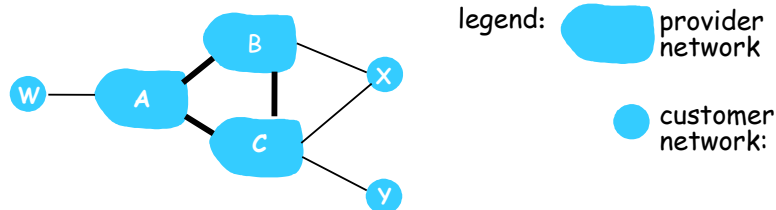
Network Layer 4-36

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

Network Layer 4-37

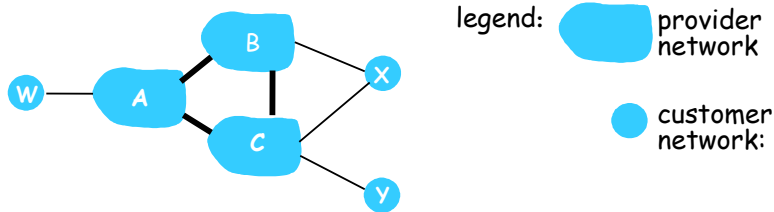
BGP routing policy



- ❑ 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

Network Layer 4-38

BGP routing policy (2)



- ❑ A advertises path AW to B
- ❑ B advertises path BAW to X
- ❑ Should B advertise path BAW to C?
 - 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!

Network Layer 4-39

Why 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

Network Layer 4-40

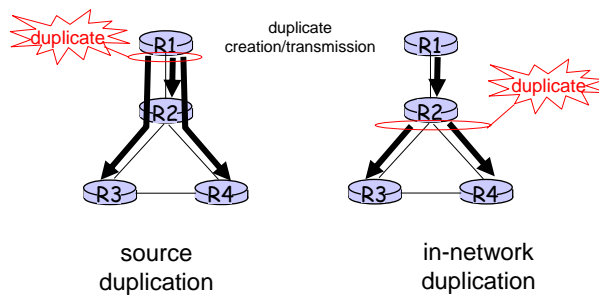
Chapter 4: Network Layer

- ❑ 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

Network Layer 4-41

Broadcast Routing

- ❑ deliver packets from source to all other nodes
- ❑ source duplication is inefficient:



- ❑ source duplication: how does source determine recipient addresses?

Network Layer 4-42

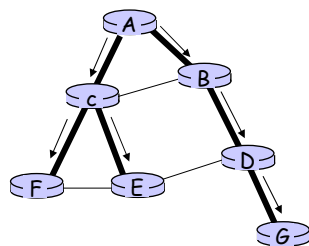
In-network duplication

- ❑ flooding: when node receives brdcst pkt, sends copy to all neighbors
 - Problems: cycles & broadcast storm
- ❑ controlled flooding: node only brdcsts pkt if it hasn't brdcst same packet before
 - Node keeps track of pkt ids already brdcsted
 - Or reverse path forwarding (RPF): only forward pkt if it arrived on shortest path between node and source
- ❑ spanning tree
 - No redundant packets received by any node

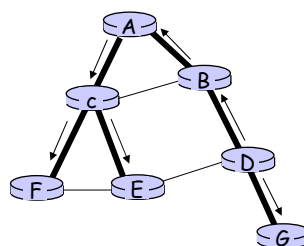
Network Layer 4-43

Spanning Tree

- ❑ First construct a spanning tree
- ❑ Nodes forward copies only along spanning tree



(a) Broadcast initiated at A

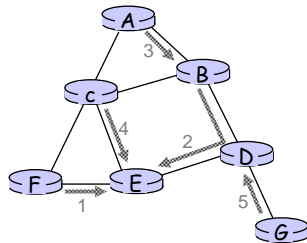


(b) Broadcast initiated at D

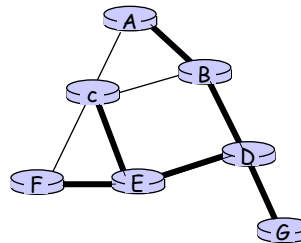
Network Layer 4-44

Spanning Tree: Creation

- Center node
- Each node sends unicast join message to center node
 - Message forwarded until it arrives at a node already belonging to spanning tree



(a) Stepwise construction of spanning tree

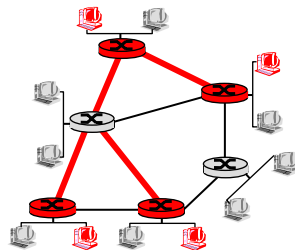


(b) Constructed spanning tree

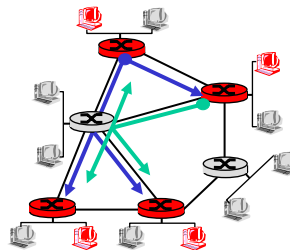
Network Layer 4-45

Multicast Routing: Problem Statement

- **Goal:** find a tree (or trees) connecting routers having local mcast group members
 - **tree:** not all paths between routers used
 - **source-based:** different tree from each sender to rcvrs
 - **shared-tree:** same tree used by all group members



Shared tree



Source-based trees

Approaches for building mcast trees

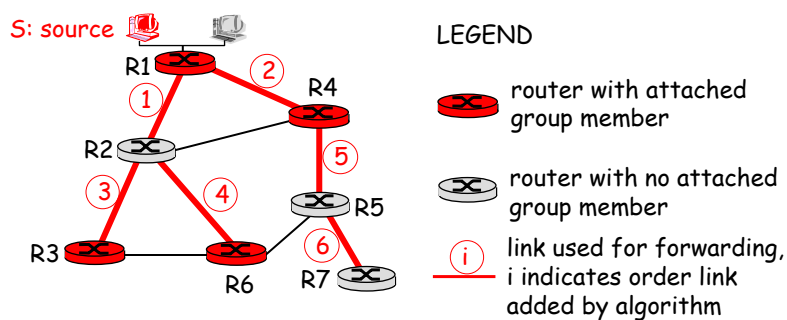
Approaches:

- **source-based tree:** one tree per source
 - shortest path trees
 - reverse path forwarding
- **group-shared tree:** group uses one tree
 - minimal spanning (Steiner)
 - center-based trees

...we first look at basic approaches, then specific protocols adopting these approaches

Shortest Path Tree

- mcast forwarding tree: tree of shortest path routes from source to all receivers
 - Dijkstra's algorithm

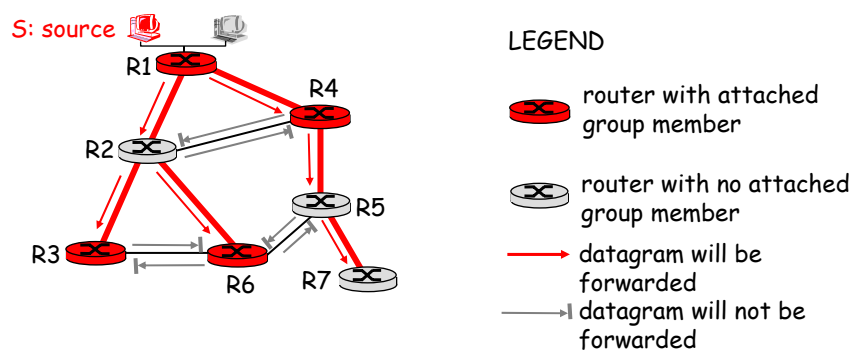


Reverse Path Forwarding

- rely on router's knowledge of unicast shortest path from it to sender
- each router has simple forwarding behavior:

if (mcast datagram received on incoming link on shortest path back to center)
then flood datagram onto all outgoing links
else ignore datagram

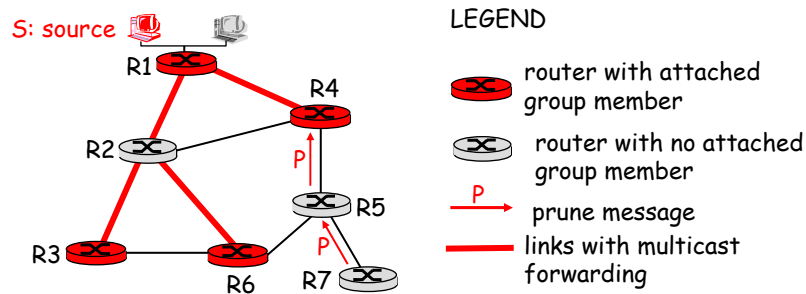
Reverse Path Forwarding: example



- result is a source-specific *reverse* SPT
 - may be a bad choice with asymmetric links

Reverse Path Forwarding: pruning

- forwarding tree contains subtrees with no mcast group members
 - no need to forward datagrams down subtree
 - "prune" msgs sent upstream by router with no downstream group members



Shared-Tree: Steiner Tree

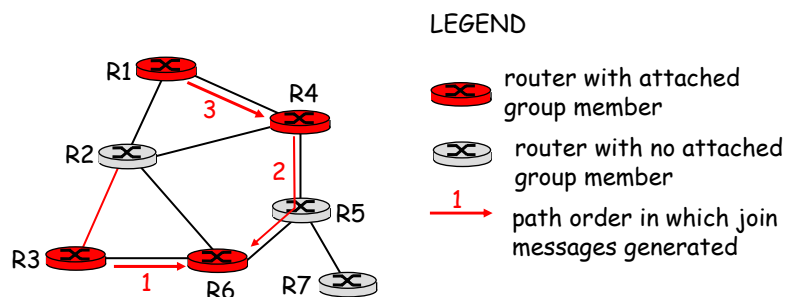
- **Steiner Tree:** minimum cost tree connecting all routers with attached group members
- problem is NP-complete
- excellent heuristics exists
- not used in practice:
 - computational complexity
 - information about entire network needed
 - monolithic: rerun whenever a router needs to join/leave

Center-based trees

- ❑ single delivery tree shared by all
- ❑ one router identified as "*center*" of tree
- ❑ to join:
 - edge router sends unicast *join-msg* addressed to center router
 - *join-msg* "processed" by intermediate routers and forwarded towards center
 - *join-msg* either hits existing tree branch for this center, or arrives at center
 - path taken by *join-msg* becomes new branch of tree for this router

Center-based trees: an example

Suppose R6 chosen as center:



Internet Multicasting Routing: DVMRP

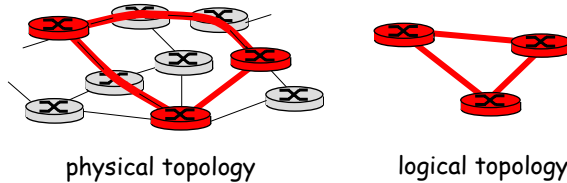
- **DVMRP**: distance vector multicast routing protocol, RFC1075
- ***flood and prune***: reverse path forwarding, source-based tree
 - RPF tree based on DVMRP's own routing tables constructed by communicating DVMRP routers
 - no assumptions about underlying unicast
 - initial datagram to mcast group flooded everywhere via RPF
 - routers not wanting group: send upstream prune msgs

DVMRP: continued...

- ***soft state***: DVMRP router periodically (1 min.) "forgets" branches are pruned:
 - mcast data again flows down unpruned branch
 - downstream router: re prune or else continue to receive data
- routers can quickly regraft to tree
 - following IGMP join at leaf
- odds and ends
 - commonly implemented in commercial routers
 - Mbone routing done using DVMRP

Tunneling

Q: How to connect "islands" of multicast routers in a "sea" of unicast routers?



- ❑ mcast datagram encapsulated inside "normal" (non-multicast-addressed) datagram
- ❑ normal IP datagram sent thru "tunnel" via regular IP unicast to receiving mcast router
- ❑ receiving mcast router unencapsulates to get mcast datagram

PIM: Protocol Independent Multicast

- ❑ not dependent on any specific underlying unicast routing algorithm (works with all)
- ❑ two different multicast distribution scenarios :

Dense:

- ❑ group members densely packed, in "close" proximity.
- ❑ bandwidth more plentiful

Sparse:

- ❑ # networks with group members small wrt # interconnected networks
- ❑ group members "widely dispersed"
- ❑ bandwidth not plentiful

Consequences of Sparse-Dense Dichotomy:

Dense

- ❑ group membership by routers *assumed* until routers explicitly prune
- ❑ *data-driven* construction on mcast tree (e.g., RPF)
- ❑ bandwidth and non-group-router processing *profligate*

Sparse:

- ❑ no membership until routers explicitly join
- ❑ *receiver-driven* construction of mcast tree (e.g., center-based)
- ❑ bandwidth and non-group-router processing *conservative*

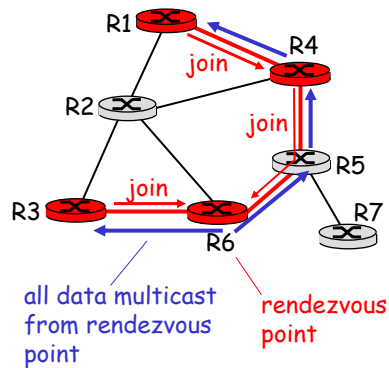
PIM- Dense Mode

flood-and-prune RPF, similar to DVMRP but

- ❑ underlying unicast protocol provides RPF info for incoming datagram
- ❑ less complicated (less efficient) downstream flood than DVMRP reduces reliance on underlying routing algorithm
- ❑ has protocol mechanism for router to detect it is a leaf-node router

PIM - Sparse Mode

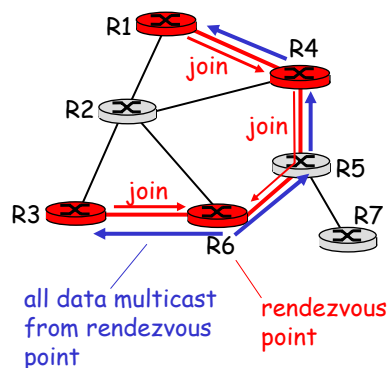
- ❑ center-based approach
- ❑ router sends *join* msg to rendezvous point (RP)
 - intermediate routers update state and forward *join*
- ❑ after joining via RP, router can switch to source-specific tree
 - increased performance: less concentration, shorter paths



PIM - Sparse Mode

sender(s):

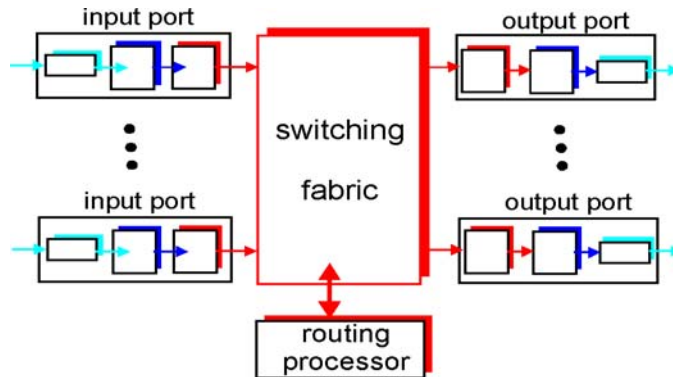
- ❑ unicast data to RP, which distributes down RP-rooted tree
- ❑ RP can extend mcast tree upstream to source
- ❑ RP can send *stop* msg if no attached receivers
 - "no one is listening!"



Router Architecture Overview

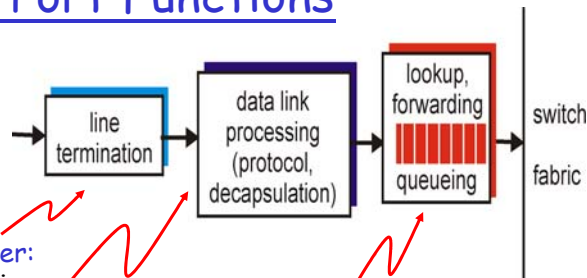
Two key router functions:

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



Network Layer 4-63

Input Port Functions



Physical layer:
bit-level reception

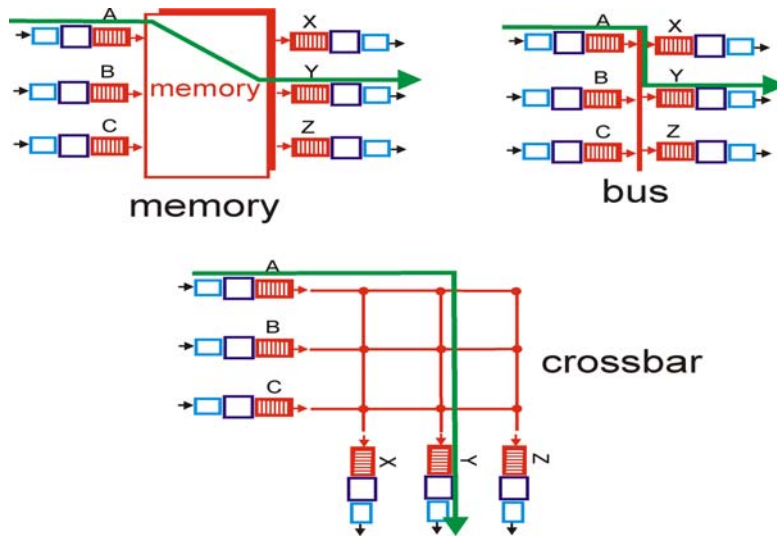
Data link layer:
e.g., Ethernet
see chapter 5

Decentralized switching:

- ❑ given datagram dest., lookup output port using forwarding table in input port memory
- ❑ goal: complete input port processing at 'line speed'
- ❑ queuing: if datagrams arrive faster than forwarding rate into switch fabric

Network Layer 4-64

Three types of switching fabrics

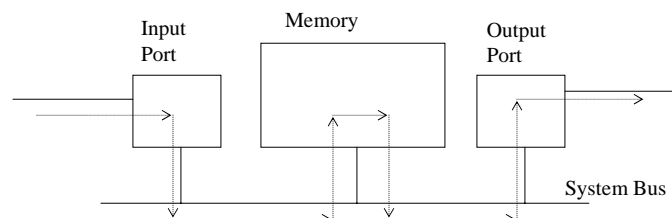


Network Layer 4-65

Switching Via Memory

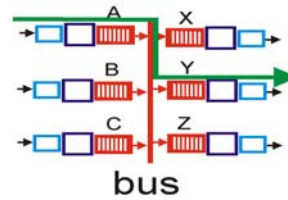
First generation routers:

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



Network Layer 4-66

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
- ❑ 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers

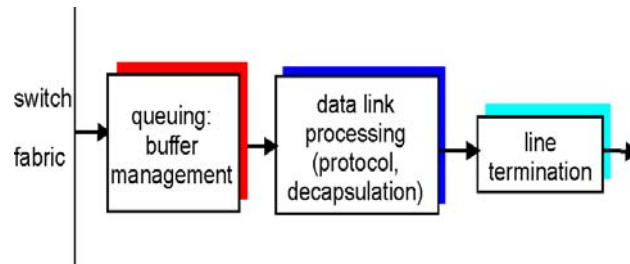
Network Layer 4-67

Switching Via An Interconnection Network

- ❑ overcome bus bandwidth limitations
- ❑ Banyan networks, other interconnection nets initially developed to connect processors in multiprocessor
- ❑ advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- ❑ Cisco 12000: switches 60 Gbps through the interconnection network

Network Layer 4-68

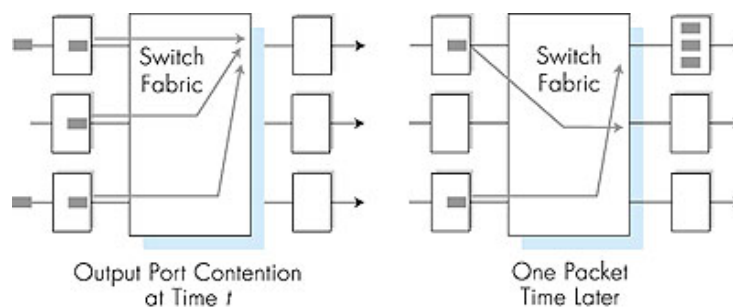
Output Ports



- ❑ *Buffering* required when datagrams arrive from fabric faster than the transmission rate
- ❑ *Scheduling discipline* chooses among queued datagrams for transmission

Network Layer 4-69

Output port queueing



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

Network Layer 4-70

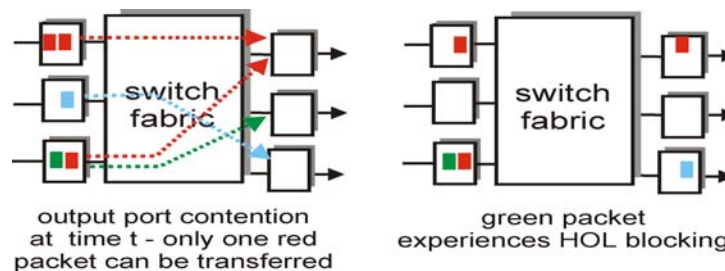
How much buffering?

- RFC 3439 rule of thumb: average buffering equal to "typical" RTT (say 250 msec) times link capacity C
 - e.g., $C = 10$ Gps link: 2.5 Gbit buffer
- Recent recommendation: with N flows, buffering equal to $\frac{RTT \cdot C}{\sqrt{N}}$

Network Layer 4-71

Input Port Queuing

- Fabric slower than input ports combined \rightarrow queuing may occur at input queues
- **Head-of-the-Line (HOL) blocking**: queued datagram at front of queue prevents others in queue from moving forward
- *queuing delay and loss due to input buffer overflow!*



Network Layer 4-72