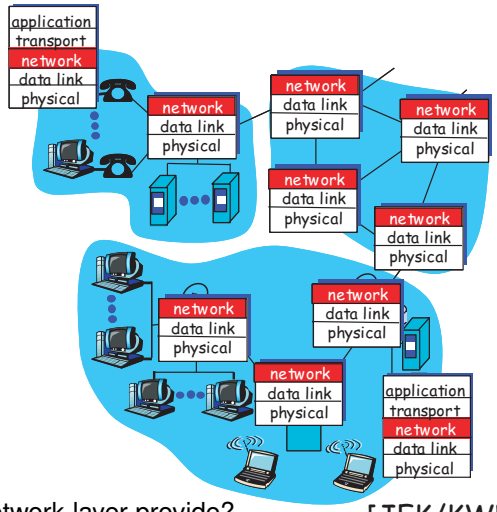
Slide 1

The network Layer



[JFK/KWR 2/E]

Issues:

- What services does the network layer provide?
- How do end-to-end routing algorithms work?
- How does the Internet IP protocol work?

Slide 2

What Services Does the Network Layer Provide?

- Host-to-host connectivity: two basic service models:
  - connectionless service (datagram service)
  - connection oriented service (aka virtual circuit service).

  Note: unlike the UDP/TCP situation, users can't choose which service model to use; each network provider uses one model, or the other.

Slide 3

Network Layer Concept 1

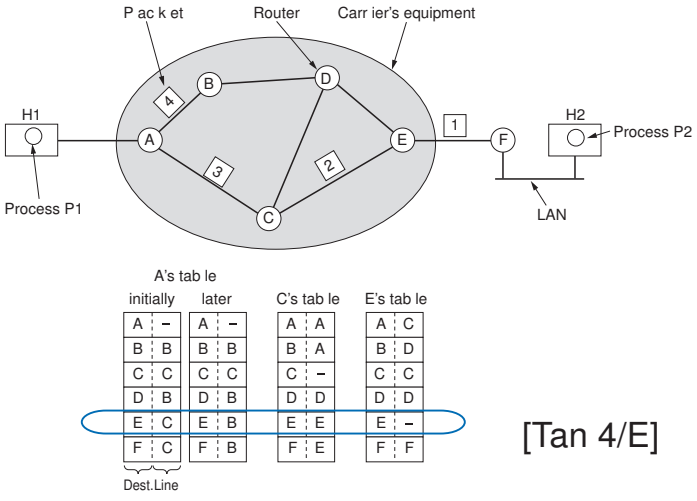
(next few slides)

Datagram networks *differ from* **virtual circuit** *networks with respect to*

- *structure of the routing tables, and*
- *ability to control routes and allocate bandwidth.*

Slide 4

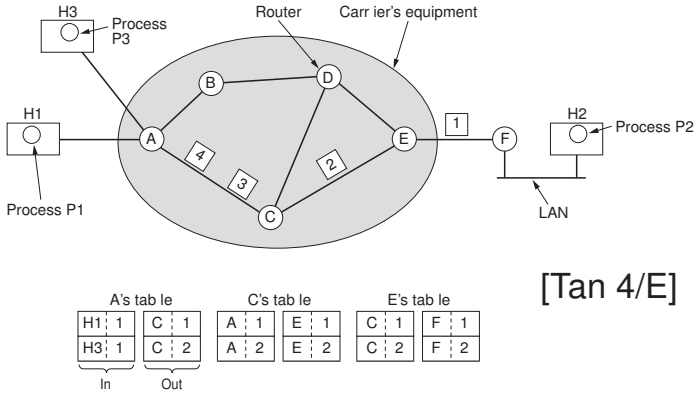
- Network layer connectionless service
  - hop-by-hop routing, pkts may travel along different routes
  - no bandwidth or delay guarantees (i.e., best effort service)
  - Example: note A's routing table changed after forwarding pkt 3.



[Tan 4/E]

Slide 5

- Network layer connection oriented service
  - Basic characteristics:
    - ∗ single path per connection
    - ∗ network guarantees in-order pkt delivery, and minimal bandwidth amount at connection setup time
    - ∗ if network is overly utilized, future calls are blocked



[Tan 4/E]

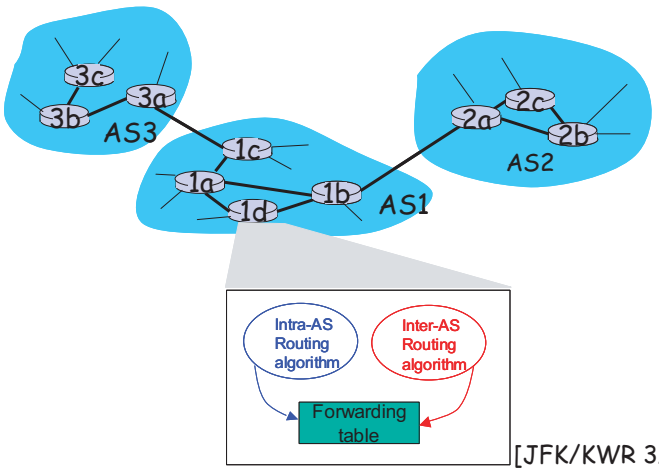
Slide 6

- ∗ Each pkt carries VC identifier (not the destination host address)
- ∗ Each router maintains "state" for each passing connection
- ∗ Router resources (e.g., buffers and bandwidth) may be allocated differently to different VCs.

- Example architectures: ATM (Asynchronous Transfer Mode), Frame Relay, and X.25 networks

- Of course, for large networks, effective routing requires hierarchical organization:

Slide 7



[JFK/KWR 3/E]

- Autonomous Systems (AS): routers in each AS are under the same administrative authority; they can run the same routing algorithm, and "trust each other"
- So, it makes sense to distinguish between:
  - Intra-domain routing protocol (aka intra-AS protocol, or interior-gateway protocol): e.g., RIP, IS-IS, OSPF, Ships-in-Night, Integrated Routing
  - Inter-domain routing protocol (aka inter-AS protocol, or exterior-gateway protocol): e.g., EGP, BGP, IDRP

Slide 8

- So, routing (end-to-end path determination) is the most important service provided by the network layer.
- Network-assisted congestion control mechanisms: yet another service
- Provisioning quality-of-service (QoS): a major step beyond best-effort services: a third service
- Supporting broadcasting and multicasting: a fourth service

Slide 9

Network Layer Concept 2

(next few slides)

Many **distributed** *network algorithms are variants of the basic* **Distance Vector Routing** *and* **Link State Routing** *algorithms.*

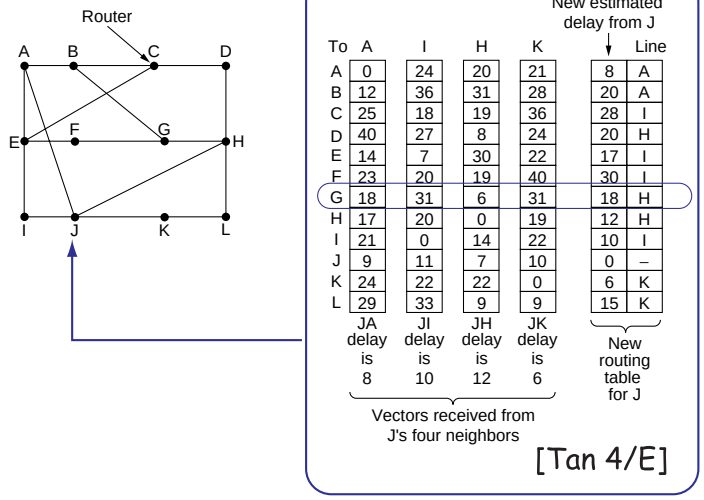
Slide 10

Distance Vector Routing (aka Distributed Bellman-Ford)

- See section 4.5.2.
- One of two important distributed least-cost routing algorithms.
- Used in some routing protocols: e.g.,
  - the original ARPAnet, RIP (Routing Information Protocol), Novel IPX,
  - BGP (Border Gateway Protocol), and ISO IDRP (Inter-Domain Routing Protocol)
- Basic properties:
  - Each node begins discovering costs of its own directly attached links
  - Gradually (through an iterative process of information exchange with neighbours), each node calculates a vector of least-cost path to each destination.
  - At any time, no node has a complete information about the costs of all network links.

Slide 11

- Idea of a basic DV algorithm: each T msec:
  1. each router measures the cost metric (e.g., average delay) to each of its neighbours: e.g., J's measured delays are: 8 msec to A, 10 to I, 12 to H, and 6 to K.



Slide 12

  2. each router sends to all of its neighbours a vector of estimated delays to each destination:
     e.g., J receives four vectors from A, I, H, and K.
  3. each router computes a new distance vector from the measured and received information.

- How Fast Does DV Converge to Least-Cost Routes?
  - Property: Good news travels fast! Suppose
    - the cost metric is the number of hops, and
    - A is initially down and then goes up
    - the good news spread after $O(n)$ exchanges (where n is the length of the longest path)

Slide 13

Slide 14

Internet routing black hole

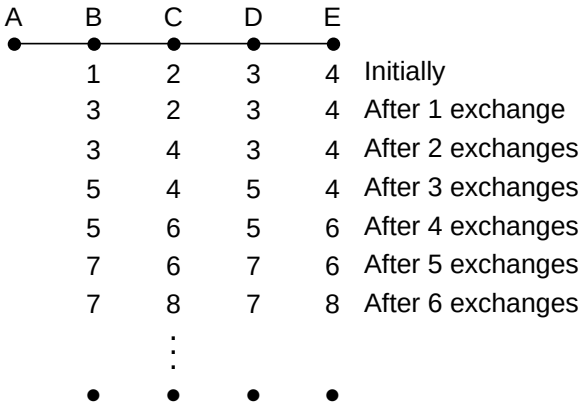
"Peter G. Neumann" neumann@csl.sri.com

Thu, 1 May 97 18:30:34 PDT

On 23 Apr 1997 at 11:14a.m. EDT, Internet service providers lost contact with nearly all of the U.S. Internet backbone operators. As a result, much of the Internet was disconnected, some parts for 20 minutes, some for up to 3 hours. The problem was attributed to MAI Network Services in McLean, Virginia (www.mai.net), which provided Sprint and other backbone providers with incorrect routing tables, the result of which was that MAI was flooded with traffic. In addition, the InterNIC directory incorrectly listed FL Internet Exchange as the owner of the routing tables. . . .

- Property: Bad news travels slow (the count-to-infinity problem)
  - Suppose that A goes down – theoretically, routers count to infinity

Slide 15



- Note: the scenario introduces routing loops : in order to get to A, C routes through B, then B routes through C, etc.
- Routing loops are like *black holes*: the TTL field decrements to zero and pkts are dropped!

- How to fix?

  Split Horizon with Poisoned Reverse

Slide 16

- Many proposals for "fixing" the slow-convergence problem in DV routing have been made:
  - some are extremely expensive (e.g., reporting the entire path to each destination in the distance vector),
  - others are based on timers, or
  - involve complex protocols for coordinating all routers
- The "split horizon with poisoned reverse" (RFC 1058), however, is particularly popular (simple and works in many cases):

  If C routes through B to get to A then
  - C tells B its (C's) distance to A is infinite (so, B won't use C to route to A

- Exercise. Consider a subnetwork with 4 routers: A to D, and links $\{AB, BC, AC, CD\}$. Suppose D goes down. Show a timing scenario under which the split horizon technique converges slowly.

- Yet another implementation of the DV algorithm:

  Each router x:

Slide 17

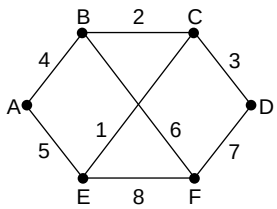
- Maintains a distance matrix $D_x(.,.)$ (not just one vector)
- After initialization, sends its DV to its neighbours
- Updates $D_x(.,.)$ if the cost to one its neighbours change, or a new minimum cost (from some other router to some destination) is received.
- Updates its neighbours if a new min. cost from itself to some other router is being computed
- Iterations continue until no update messages are sent (the algorithm enters a quiescent state)

Link State Routing

- See section 4.5.1.
- Used in: ARPAnet, OSPF (Open Shortest Path First), and the ISO CLNP (Connectionless Network Protocol)

Slide 18

- Basic idea:
  - Each router constructs a link state pkt (LSP) containing names and link costs to each of its neighbours (either periodically, or when some incident link changes cost.)
  - LSP Dissemination: The LSP is transmitted to all other routers; each router stores the most recently generated LSP from each other router in an LSP database .
  - Each router computes routes to each destination.



(a)

Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

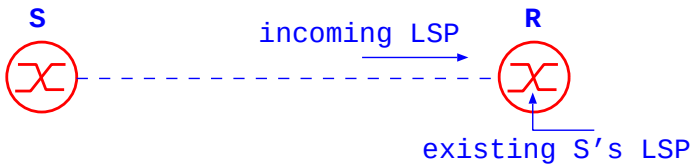
(b)

Slide 19

LSP Distribution: How To?

- The method should not depend on the routing table (since the routing table relies on the LSPs).
- Some sort of controlled flooding is preferred: e.g.,
  - Each LSP is generated with a *max. hop count*.
  - Each router forwards a received LSP on every outgoing link, except the link the LSP arrived from, and decrements the hop count by one
  - Pkts with zero hop count are ignored.

Managing LSPs



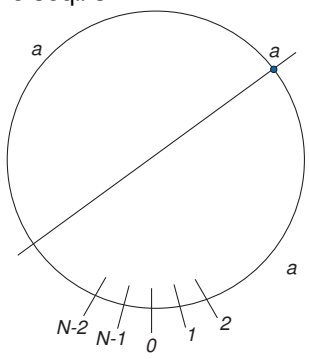
- How can R decide whether the received LSP is more recent than the stored one?

Slide 20

- Option 1: Timestamps
  - S time stamps every LSP it generates
  - R compares the time stamps, and ignores the older LSP
  - Caveat: if S accidentally generates LSP with a wrong time-stamp, say one year in the future, then R will ignore lots of valid LSPs.
  - Fix: Since time has a global meaning, R can check that the time stamp is not too far in the future.
  - The solution requires synchronized routing clocks (difficult to achieve). This option is not used in practice.

Slide 21

- Option 2: Wrapped seq#'s
  - Use sequence numbers $[0, \ldots, N-1]$ (i.e., $\bmod N$)
  - But, how to compare two seq#'s?



  - Let's define: a is less than b if
    - $a < b$ and $|b - a| < N/2$, or
    - $a > b$ and $|b - a| > N/2$.
  - Now, R overwrites the stored LSP if it receives an LSP with a larger seq#. We'll use this scheme for a while.

Slide 22

- Initial Seq #: What happens if S is rebooted?
  - S must choose an initial seq# so that its most newly generated LSPs are not ignored by other routers.
  - So, how to choose an initial number to avoid the risk of S flooding the network with a new LSP that is always ignored?
  - Fix: add a new age field.
- The age field
  - age is initialized to some constant when the LSP is first generated; it is decremented every T sec while the LSP is stored in a router's memory.
  - Example: sometime in ARPAnet: age is a 3-bit field, initialized to 56 sec, and decremented by one every 8 sec. Every router is required to generate a new LSP within 60 sec.
  - LSP with zero age is not propagated further.
  - A received LSP overwrites a stored one if the stored LSP is either
    - aged out (regardless of the seq #'s), or
    - not aged out yet, but the received LSP has a larger seq#

Slide 23

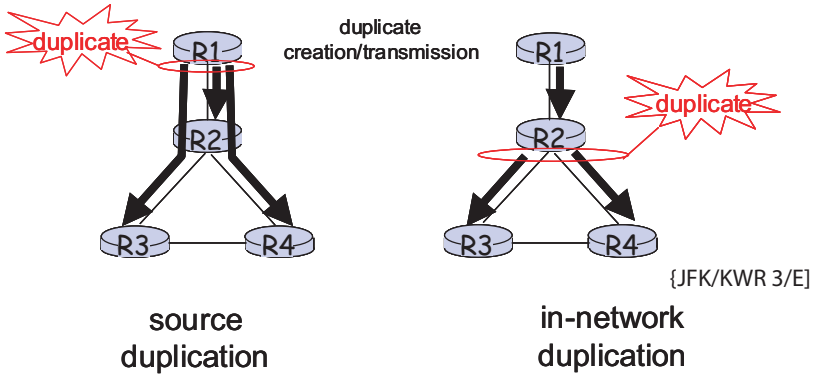
- Choosing initial LSP age:
  - Too short: LSP will time out in routers quickly (and will not propagate further)
  - Too long: a rebooted router must wait idle for a long time

Slide 24

Multicast Routing

- See section 4.7
- unicast: one to one, multicast: one to many, and broadcast: one to all.
- Applications requiring multicasting: transfer of software upgrades, multimedia streaming, video conferencing, etc.
- Why network-assisted multicast?
  - Efficiency: multicasting from the source introduces lots of duplicate messages. e.g.,

Slide 25



- A sender application need not keep track of the recipients joining and leaving a multicast group. (So, application-level multicast is not the way to go.)

Slide 26

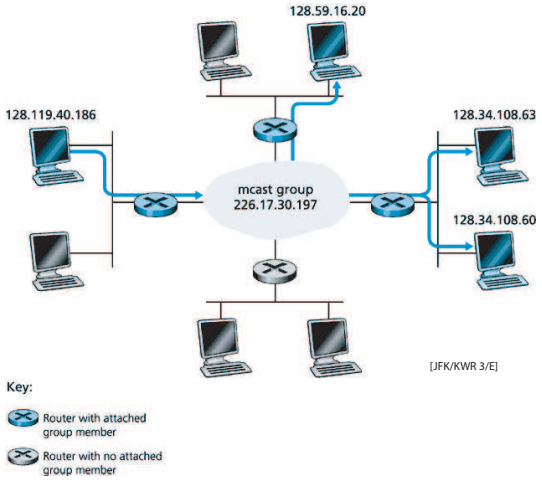
Network Layer Concept 3 *(next few slides)*

IP multicasting*: introduced in RFC 1112 [Steve Deering 1989]:*

1. Defined as the transmission of an IP datagram to a **"host group"**, a set of zero or more hosts identified by a **single IP destination address**.
2. A multicast datagram is delivered to all members of its destination host group
3. The membership of a host group is **dynamic**; that is, hosts may join and leave groups at any time.
4. There is no restriction on the location or number of members in a host group.
5. A host may be a member of more than one group at a time.
6. A host need not be a member of a group to send datagrams to it.

Slide 27

- The Internet Community Approach to the Problem:
  - Designate special IP addresses for multicasting (class D)



  - Design "host-to-attached router" protocols (e.g., IGMP) for a host to inform an attached router that an application running on the host wants to join a specific multicast group.

Slide 28

  - Identify good "multicast routing algorithms": e.g., *source-based trees*, and *core-based trees*
  - Design "multicast routing protocols": e.g., DVMRP (Distance Vector Multicast Routing Protocol), MOSPF (Multicast OSPF), PIM (Protocol Independent Multicast).

- More on class D IP addresses: in the range 224.0.0.0 through 239.255.255.255 (the low-order 28 bits is the *mcast group ID*).
  A few special cases/remarks:
  - The *all hosts* group (244.0.0.0): all multicast nodes (hosts, routers, printers, etc.) on a subnet must join this group
  - The *all-routers* group (244.0.0.2): all multicast-capable routers on a subnet must join
  - In general, the range 244.0.0.0 through 244.0.0.255 is called the *link local* group (reserved for low-level topology discovery or maintenance)
  - A mcast *session* (e.g., audio stream of conference) is defined by a mcast

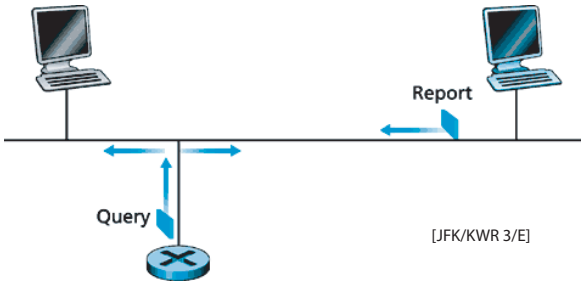
address + a port #

- All well-known mcast addresses appear in the DNS under the mcast.net hierarchy.
- See, also www.iana.org/assignments/multicast-addresses
- The Session Announcement Protocol (SAP) (224.2.127.254, sap.mcast.net), and the Session Description Protocol (SDP) are used to announce multicast addresses (and UDP port numbers) of mcast activities

Slide 29

IGMP (Internet Group Management Protocol) [RFC 2236]

- Works between a host and an attached router.



- Messages:
  - membership_query: sent by router to all attached hosts to determine all (or a specific) multicast groups joined by the hosts.
  - membership_report: sent by a host in reply to a query message, or when an application running on the host first joins a multicast group.
  - membership_leave: sent by a host to leave a group (optional). Because it is optional, the router keeps a *soft state* of the host.
- From network programming point of view:
  - UDP is used for sending and receiving mcast datagrams
  - Joining a mcast group is done by setting an *IP_ADD_MEMBERSHIP* socket option
  - Sending data to an mcast group (using UDP) does not require special operations.

Slide 30

Routing Topologies

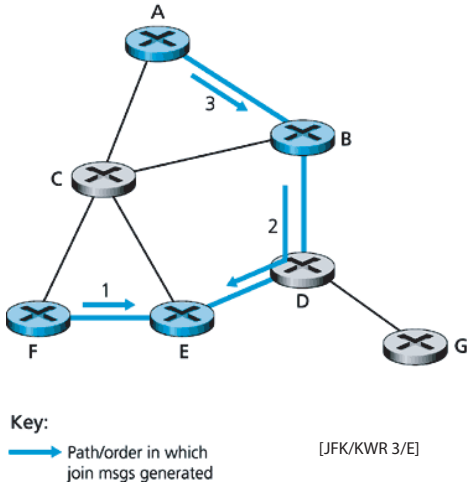
- Routing using Group-Shared Tree
  - Group-shared tree: all senders use the same multicast tree.

Slide 31

* Core-Based Tree (CBT)

  Creation:

  - First, a center (core, or rendezvous point) node is selected (how? different center-selection algorithms)
  - Each node sends a unicast *join* message to center.



[JFK/KWR 3/E]

Slide 32

  - Message forwarded until it arrives at a node already belonging to the CBT.

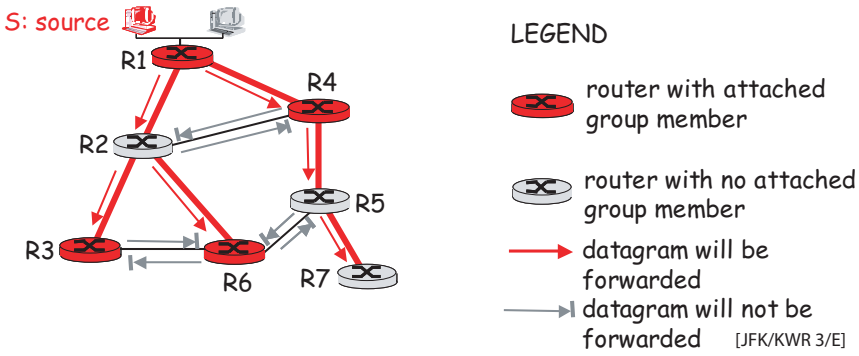
- Routing using Source-based Tree
  * Each source (sender) uses a certain tree (constructed using controlled flooding).
  * The tree construction algorithm (explained below) is called *reverse path forwarding (RPF)* .
  * RPF is used in the DVMRP (Distance vector Multicast Routing protocol) [RFC 1075] (1988) as an *easy* to implement mcast routing alg. that:
    - does not require a priori knowledge of the subnet (e.g., diameter),
    - relies on router's knowledge of the next hop on a unicast shortest path *from it to sender* (as in DV routing),
    - does not require checking of duplicate messages, and
    - does not require a special mechanism to stop the flooding operation (e.g. a hop counter)
  * Forwarding rule (controlled flooding):

Slide 33

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

∗ Why choose that particular link?

There is a good chance that the link is on the best path from the source, and therefore the received copy is the first copy to arrive to the router.



∗ In the example above:

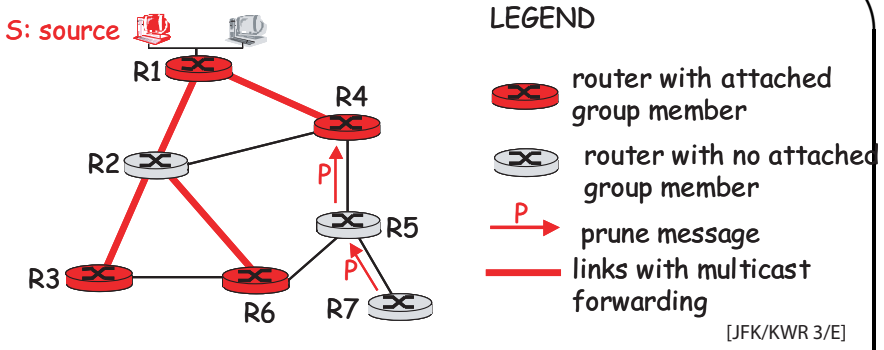
Slide 34

- $R1$ is a source
- thick lines represent least cost paths to $R1$
- $R2$ and $R4$ accept pkts from $R1$ (but not from each other)
- Similarly, $(R3, R6)$, and $(R5, R6)$ ignore pkts from each other.

∗ Result is a source-specific *reverse* tree (may be a bad choice with asymmetric links).

∗ Constructed tree may contain routers with no mcast group members (e.g., $R5$ and $R7$), *prune* msgs sent upstream by such routers.

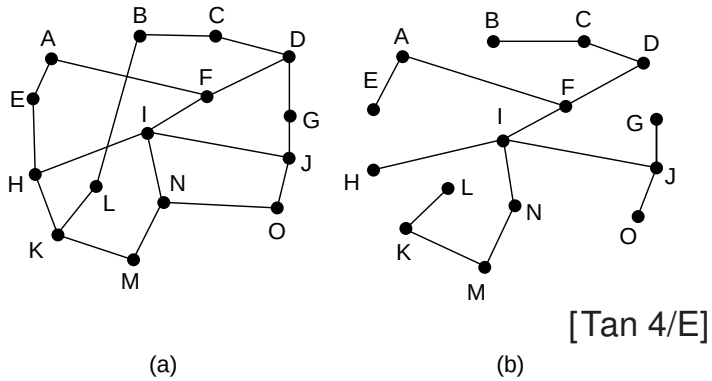
Slide 35



[JFK/KWR 3/E]

Slide 36

- Exercise. For network (a) below, suppose that (b) is the sink tree for router I (i.e., the path for each node in the tree to I is a least-cost path). Using the reverse-path forwarding (RPF) algorithm, how many pkts are required to broadcast a pkt from I?



[Tan 4/E]