I. Introduction to Routing

Week 14

- **routing** is the process of moving data across an internetwork from a source to a destination

- why routing/routers, why not just use switches?

    - router is a layer 3 device, uses ? (IP) address
    - switch is a layer 2 device, uses ? (MAC) address
    - if we only had switches, each switch would need to know/store the location of every MAC address it was connected to

- routers identify all hosts on a network as a single entry, routers don't care where a node is, just the next hop to get the data there

- what do we need to route? routing tables

- who needs them? everybody on the network, e.g. each host and each router

- each **host** needs to know:

    1. how do I reach myself?

        - loopback address (127.0.0.1)
        - allows client & server to reside on same machine and not spew data across a NW
        - allows testing of client/server apps as well as self testing of NIC

    2. how do I reach my network?

        - how does a node know if an address is from/to their NW? by the subnet mask

    3. how do I reach everything else?

        - need to know the IP address of the default/gateway router

    Note all 3 of these are manually configured!

- each **router** needs to know:

  1. how do I reach myself?

     - loopback address (127.0.0.1)

  2. how do I reach my network?

     - via the ports of the router

  3. how do I reach everything else?

     - via routing tables

       a) static routing

          - manually configured via data files
          - must be configured for every router on NW
          - if NW changes, must be re-configured

       b) dynamic routing

          - allows router to configure itself

     - via default route (0.0.0.0)

       if not in table (no match found), behave like this route

II. General Routing Goals

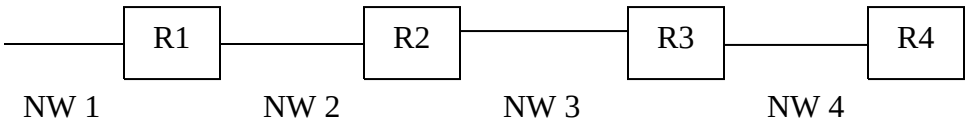
- **optimality** – selecting the best route based upon routing metrics, i.e. # hops, delay, congestion, etc.
- **simple & efficient** – minimum overhead
- **robust & stable** – good performance in adverse conditions
- **rapid convergence** – agreement of all routers on optimal routes
- **flexible** – adaptable to network changes

III. Routing Algorithm Types and Techniques

- static vs. dynamic
- single-path vs. multi-path
- host intelligent vs. router intelligent
  - host intelligent (also called source routing) is where the host specifies the route
  - most routers drop
- intra-domain (IGP) vs. inter-domain (BGP, was EGP)
- distance vector vs. link state

IV. Distance Vector Routing

- RFC 1058
- DVP share information about how far away NW are in hops, ignoring distance
- lower hop counts are route of choice
- used by RIP, RIP-2, IGRP (Cisco)
- broadcast routing tables to all neighbors periodically (e.g. RIP – 30 sec)
- DVP "route by rumor"



R1 sends to R2: I have a route to NW 1
R2 sends to R3: I have a route to NW 1

- require very little in overhead and processor power
- routing loops avoided by setting an infinity value, e.g. RIP = 16

- employs split horizon: routing information cannot be sent back the way it was received
- see also poison reverse
- see Bellman-Ford algorithm (below)

V. Link State Routing

- routers send each other info about the links they have established with other routers
- information sent as Link State Advertisements (LSAs)
- as the LSAs are propagated throughout the NW, each router builds its own picture of the NW
- link state routers use this to build a forwarding table based upon cost, i.e. status and connection type (and thus speed)
- require more processing power than D-V
- used by OSPF (open shortest path first) which uses Dijkstra's shortest path first algorithm

VI. Other Routing Protocols

- Hybrid
  - features properties from both DVP and L-S routing
  - example: EIGRP – Enhanced Interior Gateway Routing Protocol (from Cisco)
- Path Vector
  - subset of DVP
  - uses path-vectors to make metric decisions
  - example: BGP – Border Gateway Protocol (from Cisco)

TABLE 7.1 Sample Routing Table Information

Destination	Gateway	Flags[a]	Ref[b]	Use[c]	Interface
localhost	localhost	UH	0	33106	lo0
215.103.16.227	187.96.25.13	UGHD	29	102	le0
215.103.16.141	187.96.25.35	UGHD	116	16128	le1
default	187.96.25.1	UG	0	2888304	
187.96.25.0	187.96.25.2	U	210	29024	le0

[a] U = Route is up and operational; G = Packet must pass through at least one router; H = Route is to a specific host and not a network; D = Route was created dynamically
[b] Current number of routes that share the same link layer address
[c] Number of packets sent using this route

Routing Algorithms

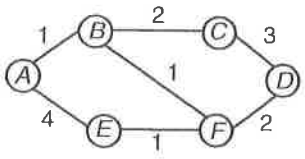
Two general algorithms are available for computing metric information: *distance-vector* and *link-state*. The goal of both types of algorithms is to route a packet from one point in the network to another point in the network through some set of intermediate routers without "looping," a situation in which a packet is forwarded across the same link several times. The primary difference between distance-vector and link-state algorithms is the manner in which they collect and propagate routing information throughout the network. Let's examine these two algorithms separately.

Distance-Vector Algorithms A *distance-vector routing algorithm* determines the distance (hence the name) between source and destination nodes by calculating the number of router hops a packet traverses en route from the source network to the destination network. An example of a distance-vector algorithm is the Bellman-Ford algorithm, which is described in Box 7.1. Two distance-vector-based routing protocols are RIP and RIP-2, which exchange routing tables with their neighbors every 30 seconds. RIP and RIP-2 also support a maximum of 15 hops. Thus, if the number of router-to-router hops between source and destination nodes is greater than 15, then the network to which the destination node is connected is considered "unreachable." This limitation restricts the size of an internetwork to 15 consecutively connected networks.

Link-State Algorithms In a *link-state routing algorithm* every router of a network does not send every other router its routing table. Instead, routers send each other information about the links they have established to other routers. This information is sent via a *link-state advertisement* (LSA), which contains the names and various cost metrics of a router's neighbors. LSAs are flooded throughout an entire router's domain. (An example of how this is done is described later in our discussion of OSPF.) Routers also store the most recent LSA they receive, and destination routes are calculated using LSA information. Thus, rather than storing actual paths, which is the case with distance-vector algorithms, link-state algorithms store the information needed to generate such paths. An example of a link-state algorithm is Dijkstra's shortest path algorithm, which iterates on length of path to determine a shortest route. Link-state-based routing protocols include OSPF, OSI's IS-IS, and Netware's Link Services Protocol (NLSP). Box 7.2 illustrates Dijkstra's shortest path algorithm.

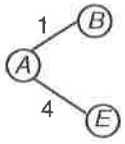
BOX 7.1 Bellman-Ford Algorithm

The Bellman-Ford routing algorithm is distance-vector-based and iterates on the number of hops a source node is from a destination node. To illustrate this algorithm, consider the following undirected graph, which depicts a sample network. The vertices *A, B, C, D, E,* and *F* may be thought of as routers, and the edges connecting the vertices are communication links. Edge labels represent an arbitrary cost. Our goal is to find the shortest path from *A* to *D* using the number of hops as the basis for our path selection.



We examine the costs of all paths leading from *A* to each node on a hop-by-hop basis.

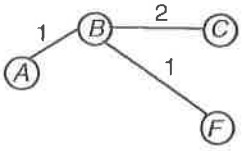
One Hop



Path *AB* = 1
Path *AE* = 4

Choose path *AB*

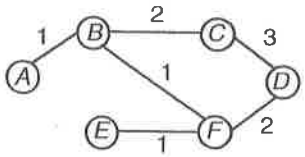
Two Hops



Path *ABC* = 3
Path *ABF* = 2

Choose path *ABF*

Three Hops

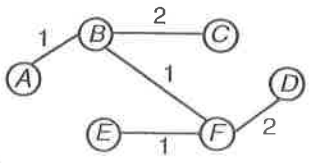


Path *ABCD* = 6
Path *ABFD* = 4
Path *ABFE* = 3

Choose path *ABFD*
Choose path *ABFE*

In the last step (three hops), two paths are selected. The first path, *ABFD,* represents the least-cost path from *A* to *D* based on the hops metric. The second path, *ABFE,* is selected because it represents the least-cost path from *A* to *E.*

The final result of the Bellman-Ford algorithm yields a tree that represents the least cost incurred from the source node to every node of the network. Similar trees can be generated for every node of the network. Node *A*'s least-cost tree for our example is as follows:



From Node *A*:

- the least-cost path to *B* is *AB* = 1
- the least-cost path to *C* is *ABC* = 3
- the least-cost path to *D* is *ABFD* = 4
- the least-cost path to *E* is *ABFE* = 3
- the least-cost path to *F* is *ABF* = 2

BOX 7.2 Dijkstra's Short

Dijkstra's SPF routin
uses a "closest nodes

Given a source noa
directly connects n
ate closest nodes pl

Consider the followi
and *F* may be thoug
Edge labels represen
on distance.

To implement this alg
the source node. We
the zero closest node
search for the success

First Closest Node (*k*

The first closest node
the *AB* path has a sma

Second Closest Node

The second closest n
includes the first clos
8, *ABE* = 7, or *AE* =
closest node to *A*.

Third Closest Node (

The third closest nod
more direct links to *A*
= 11, or *AEF* = 11. T

Fourth Closest Node

The fourth closest n
paths and related cos
the fourth closest no
of the algorithm bec

Fifth Closest Node (

The fifth closest no
paths and related co
shortest path is *ABF*

Since *D* is the destin

I. Intro to Dynamic Routing

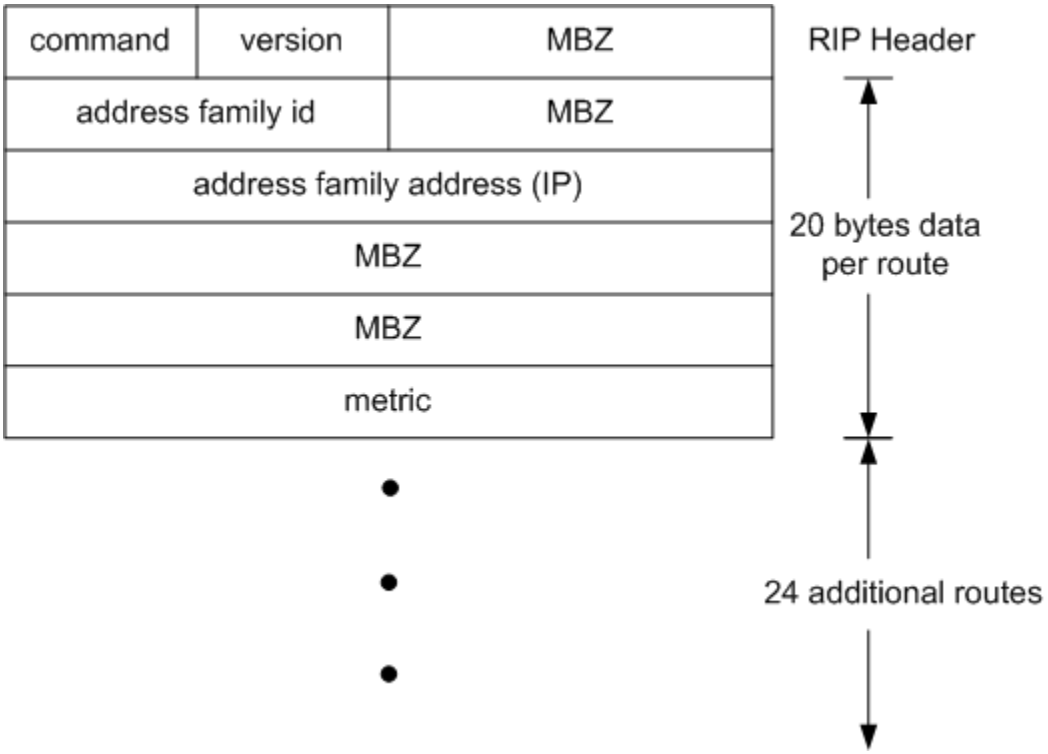
Routed vs. Routing Protocols

routed protocol - responsible for moving data across an internetwork, e.g. IP
routing protocol – responsible for exchanging routing information, e.g. RIP

RIP – Routing Information Protocol

- RFC 1058
- distance vector protocol (DVP), based upon hop count
- 2 versions, RIP & RIP-2
- originally from Xerox used in Xerox Network System – XNS (see http://www.tcpipguide.com/free/t_RIPOverviewHistoryStandardsandVersions.htm
- IGP – thus does not scale well, mostly due to limit of 16 hops
- RIP uses UDP (specifically port 520)
- relatively slow convergence
- uses timers to trigger updates
    - broadcast (30 sec) – sends routing table to all neighboring routers
    - aging (180 sec) – if no message received from neighbor, drops from table
- other issues
    - cannot measure/calculate delay, throughput or reliability (like a L-S P)
    - single-path protocol
    - does not support variable length subnet masks (only natural)
    - not a secure protocol

II. RIP PDU



- max size: 4 (header) + 20 * 25 (routes) +8 (UDP) + 20 (IP)
- **command:** 1 – request (for other routers' tables)
  2 – reply (of this routers' table)
- **version:** 1 – RIPv1, 2 – RIPv2
- **AFI:** 2 for IP
- **IP addr:** addr of NW
- **metric:** # hops

IP dest addr = ? (broadcast)