LABORATORY WORK NO. 5
DYNAMIC ROUTING

1. Objectives

The aims of this laboratory are: understanding dynamic routing and routing metrics, differentiating routed and routing protocols, distance-vector and link-state routing protocols, understanding the concept of autonomous systems, studying the RIP protocol, learning to configure RIP as the network routing protocol.

2. Theoretical considerations

2.1 Using dynamic routing

An important element of routing strategy is the decision to use static or dynamic routing on your network. With static routing, network administrators must manually create and modify the routing table entries. Dynamic routing uses specialized protocols that enable routers to communicate with each other and share their routing table information. For a router to effectively forward traffic to a distant network, it must have information in the form of a routing table entries it has obtained from a router connected to that network. When you configure a router to use dynamic routing, it transmits the contents of its routing table to others routers at various intervals.

Dynamic routing eliminates the need for network administrators to manually create static routes on each router. More importantly, dynamic routing enables routers to compensate for changes in the network. For example, network designers often create redundant routes between networks, so that if a router or a connection fails, traffic can still reach any destination. For this type of failover system to work, routing tables entries must be changed when a failure occurs. It is possible for administrators to make changes, if they are on duty when the failure occurs, and if they are aware of the failure. However, dynamic routing enables the routers to make these changes automatically.

When a router fails to transmit its routing table entries on schedule, the other routers detect the absence of incoming messages and remove the failed router from their routing table. This prevents the routers from forwarding traffic to that failed router; instead, they use other paths through the network. When the failed router is back in operation, it resumes transmitting its dynamic routing message and the other routers on the network begin to use it again by modifying their routing tables accordingly.

2.2 Understanding routing metrics

One of the most important functions of dynamic routing protocols is to evaluate the relative efficiency of routes to a specific destination. On a network with redundant routers, there might be several paths that packets can take from a particular source to a particular destination. When this is the case, a router might have multiple entries for the same destination in its routing table, and it is up to the router to forward packets using the most efficient route available. Routing table entries all include a numeric qualifier called a metric, which the router uses to evaluate routes to the same destination. The lower the metric value the more efficient the route.

Although IP routers all use the metric the same way, there is no standardized definition for what the metric actually represents, if anything. On a network that uses static routing, network administrators can arbitrarily assign metrics to the routing table entries they create. In dynamic routing, the metric values must represent a specific attribute for routing protocols to compute them. However, different routing protocols use different algorithms to compute the metric for each routing table entry; this is one of the main characteristics that differentiate between routing protocols. For some routing protocols these metrics are static and may not be changed. For other routing protocols these values may be assigned by a network administrator. The most common metric values are hop, bandwidth, delay, reliability, load, and cost.

A hop is a metric value used to measure distance based on the number of networks a datagram traverses. Each time a router forwards a datagram onto a segment this is count as a single hop. Routing protocols that observe hops as their primary metric value consider the best or preferred path (when multiple paths exist) to a destination to be the one with the least number of network hops.

Bandwidth, protocols that consider the capacity of a link use this metric. Bandwidth is measured in terms of bits per second. Links that support higher transfer rates like gigabit are preferred over lower capacity links like 56Kb. These protocols determine and consider the bandwidth capacity of each link along the path end to end. The path with the overall higher bandwidth is chosen as the best route.

Delay represents the amount of time it takes for a router to process, queue, and transmit a datagram out an interface. Protocols that use this metric must determine the delay values for all links along the path end to end, considering the path with the lowest (cumulative) delay to be a better route.

Reliability, although this metric may be configured as a fixed value by an administrator, it is generally measured dynamically over a specific time frame, such as five seconds. Routers observe attached links, reporting problems, such as link failures, interface errors, lost datagrams and so on. Links experiencing more problems would be considered less reliable than others making them less desirable paths - the higher the reliability the better the path. Because network conditions are constantly changing, link reliability will change. This value is generally measured as a percentage of 255, with 255 being the most reliable and 1 being least reliable.

Load is a variable value, generally measured over a five-second window indicating the traffic load over a specific link. Load measures the amount of traffic occupying the link over this time frame as a percentage of the link's total capacity. The value 255 is equivalent to 100% utilization or load - the higher the value the higher the traffic load (bandwidth utilization) across this link. As traffic increases, this value increases. Values approaching 255 indicate congestion, while lower values indicate moderate traffic loads - the lower the value, the less congested the path, the more preferred. This value may be manually configured as a static value by an administrator or it may be dynamically tracked allowing it to adjust as traffic patterns within the network change.

Network administrators can affect the way routers make path decisions by setting arbitrary metric values on links along the path end to end. These arbitrary values are typically single integers with lower values indicating better paths.

2.3 Routed and routing protocols

Confusion often exists between the terms routed protocol and routing protocol.

A routed protocol defines any network protocol that provides enough information in its network layer address to allow a packet to be forwarded from host to host based on the addressing scheme. Routed protocols define the format and use of fields within a packet. Packets generally are conveyed from one end system to end system. IP (Internet Protocol) is an example of routed protocol.

A routing protocol defines a protocol that supports a routed protocol by providing mechanism for sharing routing information. Routing protocol messages move between the routers. A routing protocol allows the routers to communicate with other routers to update and maintain tables. TCP/IP examples of routing protocols are Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), Enhanced Interior Gateway Routing Protocol (EIGRP) and Open shortest Path First (OSPF).

2.4 Distance-vector and link-state routing

Most routing protocols fall into one of two classes: distance vector or link state.

2.4.1 Distance vector routing

The name distance vector is derived from the fact that routes are advertised as vectors of (distance, direction), where distance is defined in terms of a metric and direction is defined in terms of the next-hop router. For example, "Destination A is a distance of 5 hops away, in the direction of next-hop router X". As that statement implies, each router learns routes from its neighboring routers' perspectives and then advertises the routes from its own perspective. Because each router depends on its neighbors for information, which the neighbors in turn may have learned from their neighbors, and so on, distance vector routing is sometimes facetiously referred to as "routing by rumor."

Distance vector routing protocols include the following: Routing Information Protocol (RIP) for IP, Xerox Networking System's XNS RIP, Novell's IPX RIP , Cisco's Internet Gateway Routing Protocol (IGRP), DEC's DNA Phase IV, AppleTalk's Routing Table Maintenance Protocol (RTMP) .

A typical distance vector routing protocol uses a routing algorithm in which routers periodically send routing updates to all neighbors by broadcasting their entire route tables. Periodic updates means that at the end of a certain time period, updates will be transmitted. This period typically ranges from 10 seconds for AppleTalk's RTMP to 90 seconds for Cisco's IGRP. At issue here is the fact that if updates are sent too frequently, congestion may occur; if updates are sent too infrequently, convergence time may be unacceptably high.

In the context of routers, neighbors always mean routers sharing a common data link. A distance vector routing protocol sends its updates to neighboring routers and depends on them to pass the update information along to their neighbors. For this reason, distance vector routing is said to use hop-by-hop updates.

When a router first becomes active on a network, how does it find other routers and how does it announce its own presence? Several methods are available. The simplest is to send the updates to the broadcast address (in the case of IP, 255.255.255.255). Neighboring routers speaking the same routing protocol will hear the broadcasts and take appropriate action. Hosts and other devices uninterested in the routing updates will simply drop the packets.

Most distance vector routing protocols take the very simple approach of telling their neighbors everything they know by broadcasting their entire route table. Neighbors receiving these updates glean the information they need and discard everything else.

2.4.2 Link-state routing

Link state protocols, sometimes called shortest path first or distributed database protocols, are built around a well-known algorithm from graph theory, Dijkstra's shortest path algorithm. Examples of link state routing protocols are: Open Shortest Path First (OSPF) for IP, The ISO's

Intermediate System to Intermediate System (IS-IS) for CLNS and IP, DEC's DNA Phase V, Novell's NetWare Link Services Protocol (NLSP).
Although link state protocols are rightly considered more complex than distance vector protocols, the basic functionality is not complex at all:

1. each router establishes a relationship—an adjacency—with each of its neighbors;
2. each router sends link state advertisements (LSAs);
3. each router stores a copy of all the LSAs it has seen in a database. If all works well, the databases in all routers should be identical;
4. the completed topological database, also called the link state database, describes a graph of the internetwork. Using the Dijkstra algorithm, each router calculates the shortest path to each network and enters this information into the route table.

Neighbor discovery is the first step in getting a link state environment up and running. In keeping with the friendly neighbor terminology, a Hello protocol is used for this step. The protocol will define a Hello packet format and a procedure for exchanging the packets and processing the information the packets contain.

At a minimum, the Hello packet will contain a router ID and the instance, an IP address from one of the router's interfaces. Other fields of the packet may carry a subnet mask, Hello intervals, specified maximum period the router will wait to hear a Hello before declaring the neighbor "dead," a descriptor of the circuit type, and flags to help in bringing up adjacencies.

Beyond building adjacencies, Hello packets serve as keepalives to monitor the adjacency. If Hello is not heard from an adjacent neighbor within a certain established time, the neighbor is considered unreachable and the adjacency is broken. A typical interval for the exchange of hello packets is 10 seconds, and a typical dead period is four times that.

After the adjacencies are established, the routers may begin sending out LSAs. The advertisements are sent to every neighbor. In turn, each received LSA is copied and forwarded to every neighbor except the one that sent the LSA. This process is the source of one of link state's advantages over distance vector. LSAs are forwarded almost immediately, whereas distance vector must run its algorithm and update its route table before routing updates, even the triggered ones, can be forwarded. As a result, link state protocols converge much faster than distance vector protocols converge

when the topology changes. The flooding process is the most complex piece of a link state protocol. There are several ways to make flooding more efficient and more reliable, such as using unicast and multicast addresses, checksums, and positive acknowledgments.

2.4.3 Comparison of routing philosophies

Table 5.1 *Distance-vector versus link-state routing*

Distance-Vector Routing	Link-State Routing
Each router sends routing information to its neighbors	Each router send routing information to all other routers
The information send is an estimate of its path cost to all networks	The information send is the exact value of its link cost to adjacent networks.
Information is send on a regular periodic basis	Information is send when changes occurs
A router determines next-hop information by using the distributed Bellman-Ford algorithm on the received estimated path costs	A router first builds up a description of the topology of the internet and then may use any routing algorithm to determine next-hop information

2.5 Autonomous Systems

To proceed in our discussion of routing protocols, we need to introduce the concept of an autonomous system. An autonomous system (AS) exhibits the following characteristics:

- an AS consists of a group of routers exchanging information via a common routing protocol;
- an AS is a set of routers and network managed by a single organization;
- except times of failures, an AS is connected (in a graph-theoretic sense).

A common routing protocol, referred as an interior routing protocol (IRP), passes routing information between routers within an AS. The protocol used within the AS does not need to be implemented outside of the system. This flexibility allows IRPs to be custom tailored to specific applications and requirements.

It may happen, however, that a network will be constructed of more that one AS. For example, all of the LANs at a site, such as an office complex or campus, could be linked by routers to form an AS. This might be linked through a wide area network to other ASs. In this case, the routing algorithms and information in routing tables used by routers in different ASs may differ. Nevertheless, the routers in one AS need at least a minimal level of information concerning networks outside the system that can be reached. The protocol used to pass routing information between routers in different ASs is referred to as an exterior routing protocol (ERP).

We can expect that an ERP will need to pass less information that an IRP, for the following reason. If a datagram is to be transferred from a host in one AS to a host in another AS, a router in the first system need only determine the target AS and devise a route to get into the target system. Once the datagram enters the target AS, the routers within that system can cooperate to deliver the datagram; the ERP is not concerned with, and does not know about, the details of the route followed within the target AS.

Whereas IGPs discover paths between networks, EGPs discover paths between autonomous systems. Examples of EGPs include the following: Border Gateway Protocol (BGP) for IP, Exterior Gateway Protocol (EGP) for IP, the ISO's InterDomain Routing Protocol (IDRP).

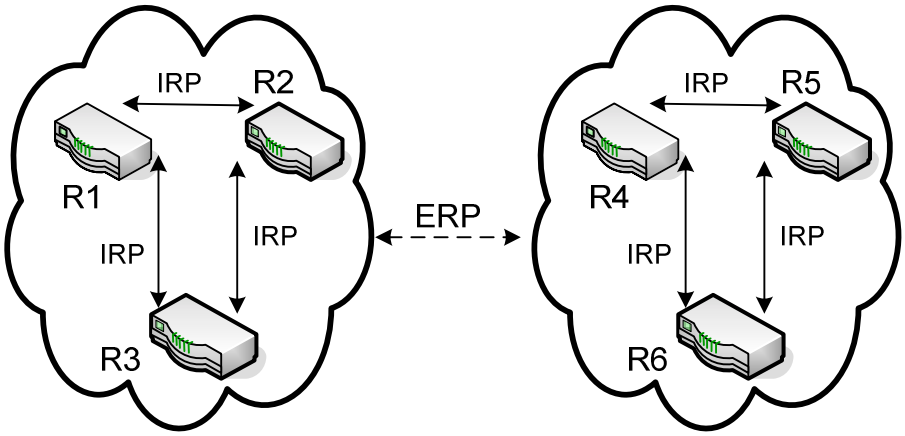

Figure 5.1 *Autonomous systems with interior and exterior routing protocols*

2.6 Routing Information Protocol

RFC 1058 defines RIPv1 (Routing Information Protocol). RIP is a distance-vector routing protocol that uses router hop count as the metric, it uses a maximum hop count equal to 15. For RIP, infinity is 16 hops. It is a classful routing protocol (no support for VLMS – Variable Length subnet Mask and CIDR – Classless Inter-Domain Routing). Despite its simplicity, it remains suitable for smaller internets and is one of the most widely used routing protocols.

RIPv2 was first described in RFC 1388 and RFC 1723 (1994); the current RFC is 2453. Although current environments use advanced routing protocols such as OSPF and EIGRP, there still are networks using RIP. The need to use VLSMs and other requirements prompted the definition of RIPv2.

RIPv2 improves upon RIPv1 with the ability to use VLSM, with support for route authentication, and with multicasting of route updates. RIPv2 supports CIDR. It still sends updates every 30 seconds and retains the 15-hop limit; it also uses triggered updates. RIPv2 still uses UDP port 520; the RIP process is responsible for checking the version number. It retains the loop-prevention strategies of poison reverse and counting to infinity. You can use RIPv2 in small networks where VLSM is required. It also works at the edge of larger networks.

RIP sends routing-update messages at regular intervals and when the network topology changes. When a router receives a routing update that includes changes to an entry, it updates its routing table to reflect the new route. The metric value for the path is increased by 1, and the sender is indicated as the next hop. RIP routers maintain only the best route (the route with the lowest metric value) to a destination. After updating its routing table, the router immediately begins transmitting routing updates to inform other network routers of the change. These updates are sent independently of the regularly scheduled updates that RIP routers send.
RIP uses a single routing metric (hop count) to measure the distance between the source and a destination network. Each hop in a path from source to destination is assigned a hop count value, which is typically 1. When a router receives a routing update that contains a new or changed destination network entry, the router adds 1 to the metric value indicated in the update and enters the network in the routing table. The IP address of the sender is used as the next hop.

RIP prevents routing loops from continuing indefinitely by implementing a limit on the number of hops allowed in a path from the source to a destination. The maximum number of hops in a path is 15. If a router receives a routing update that contains a new or changed entry, and if increasing the metric value by 1 causes the metric to be infinity (that is, 16), the network destination is considered unreachable. The downside of this stability feature is that it limits the maximum diameter of a RIP network to less than 16 hops.

RIP uses numerous timers to regulate its performance. These include a routing-update timer, a route-timeout timer, and a route-flush timer. The routing-update timer clocks the interval between periodic routing updates. Generally, it is set to 30 seconds, with a small random amount of time added whenever the timer is reset. This is done to help prevent congestion, which could result from all routers simultaneously attempting to update their neighbors. Each routing table entry has a route-timeout timer associated with it. When the route-timeout timer expires, the route is marked invalid but is retained in the table until the route-flush timer expires.

RIP includes a number of other stability features that are common to many routing protocols. These features are designed to provide stability despite potentially rapid changes in a network's topology. For example, RIP implements the split horizon and holddown mechanisms to prevent incorrect routing information from being propagated.

One of the most significantly problems with RIP is its slow convergence to a change in topology. Consider the configuration of Figure 5.2 with all link costs equal to one. B maintains a distance to network 5 of 2, with a next hop of D, while A and C both maintain a distance to network 5 of 3, with next hp to B. Now suppose that router D fails. Then the following scenario could occur:

1. B determines that the network 5 is no longer reachable via D and sets its distance count to 4 based on a report from A and C. This happens because B has recently received a report from both A and C that D is reachable with a distance of 3. At the next reporting interval, B advertises this information in distance vectors to A and C;
2. A and C receive this increased distance information from D and increment their reachability information to network 5 to a distance of 5 (4 plus 1 to reach B);

3. B receives the distance count of 5 and assumes that network 5 is now a distance 6 away.

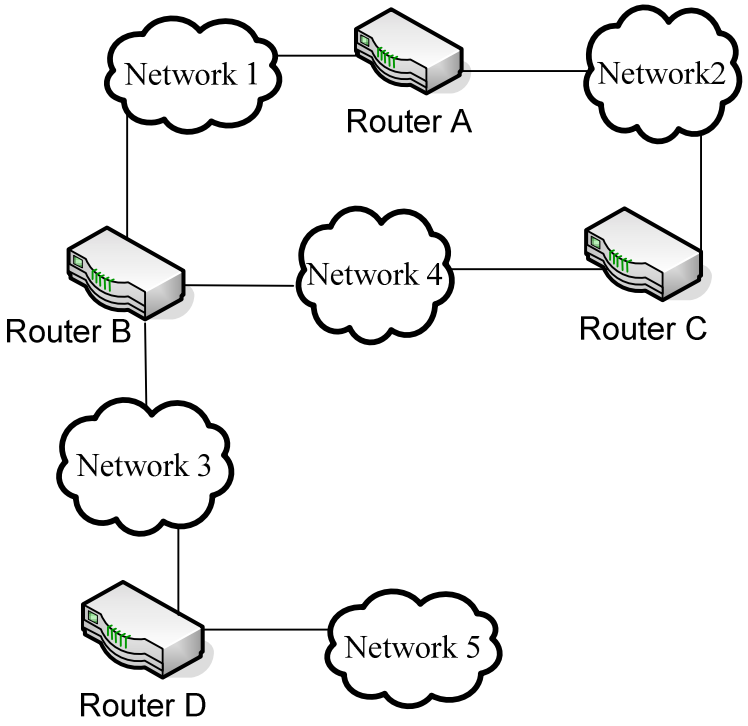

Figure 5.2 *Scenario*

This pattern continues until the distance value reaches infinity, which is only 16 in RIP. Once this value is obtained, a node determines that the target network is no longer reachable.

The counting-to-infinity problem is caused by a mutual misunderstanding between B and A (and between B and C). Each thinks that it can reach network 5 via the other. The split horizon rule in RIP states that it is never useful to send information about a route back in the direction from which it came, as the router sending your information is nearer to the destination than you are. The split horizon rule does speed things up; an erroneous route will be eliminated within the interval of the 180-second timeout.

At a small increase in message size, RIP provides even faster response by the using poisoned reverse. This rule differs from simple split horizon by sending updates to neighbors with a hop count of 16 for routes learned from

those neighbors. If two routers have routes pointing to each other, advertising reverse routes with a metric of 16 breaks the loop immediately. RIP continues to be a popular routing protocol because it is simple and because it is well suited to small internets. However, it does have a number of limitations, including the following: as internets grow, destinations that require a metric more than 15 become unreachable, making RIP unsuitable for large configurations; the overlay simplistic metric leads to suboptimal routing tables, resulting in packets being sent over slow (or otherwise costly) links when better paths are available; RIP-enabled devices will accept RIP updates from any device – this enables a misconfigured device easily to disrupt an entire configuration.

3. Lab activity

You will learn how to configure RIP as the network routing protocol. For this, we will use the following configuration, see Figure 5.3.

3.1 Configure the computers with the IPs provided.
3.2 Configure the routers' interfaces using the computers as consoles. Do not forget the clock rate.
3.3 Configure a static route between Router1 and Router2.
3.4 Verify that the ping works between Computer1 and Computer2. Also, execute `traceroute –d` *`ip_adress`* to visualize the route.
3.5 Configure RIP protocol on routers: dynamic routes between Router1 and Router3 and Router2 and Router3.
3.6 Verify that ping works between Computer1 and Computer2 and tell which route will be chosen: the static or the dynamic one (use `traceroute` command).
3.7 Kill the static route between Router1 and Router2.
3.8 Verify that ping works between Computer 1 and Computer 2. Also, execute `traceroute –d` *`ip_address`* to visualize the route.
3.9 Configure RIP protocol on routers: dynamic route between Router1 and Router2.
3.10 Verify that ping works between Computer1 and Computer2. Also, execute `traceroute –d` *`ip_address`* to visualize the route. Which route was chosen? Why?

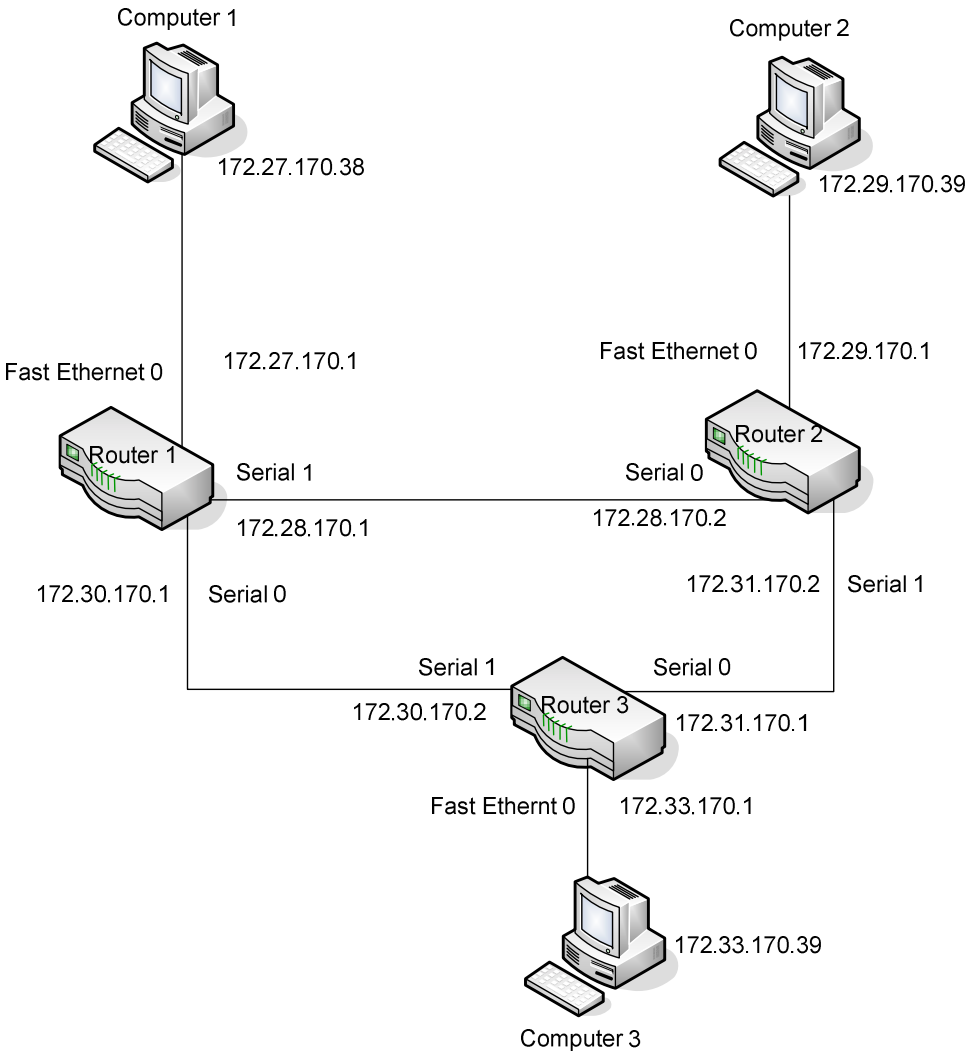

Figure 5.3 *Lab configuration*

Notes