Creating Routing Tables

Recommended reading: This note is based on Chapters 12,13,14,16,28 of the text book.

Routing Architecture

- **Routing table**
  The Internet is a collection of networks connected by routers. Major functions for delivering IP datagrams from one network to another can be partitioned into two groups: (1) IP forwards datagrams based on the routing tables (data plane) and (2) routing protocols create routing tables (control plane). The routing table in a router is a key in IP routing. Two issues are important on routing tables: what data should be kept in the table and how to get those data. The answer depends on the architectural complexity, the size of the Internet, and administrative policies. A routing table is usually set-up by initialization and maintained by updating when there is any change in the network. In large, rapidly changing networks, automated updating methods are needed.

- **Routing with partial information**
  To send a datagram to the destination, the routers select the routes for the next hop based on the routing tables. Each routing table, like a road sign, may only have partial routing information, but a series of routers will finally deliver the datagrams to the destination. A key issue of the road signs is the correctness: if the road signs in the routing tables finally point to the destination. Further, we may want the road signs provide a shortest path to the destination.

- **Core router and peer backbone internets**
  When the Internet used ARPANET as the backbone, routers can be partitioned into core routers and noncore routers. Core routers keep complete information about all possible destinations and noncore routers only keep partial information (the routes to a core router and local destinations). To avoid the inefficiency, all core routers exchange routing information so that each would have complete information about optimal routes to all possible destinations. The core system assumes a centralized set of routers keep all routing information. It works well for the internet which has a single and centrally managed backbone.
  
  When the Internet used multiple backbones (called peer backbone networks) the routing becomes more complex. The simple partition of the core systems to each peer backbone may have routing loop problem due to the default routes setting.

- **Automatic route propagation and computation**
  In the core system, core routers exchange the routing information by propagating the route information each router holds. To deal with the dynamic changing of the Internet, automatic route propagation is needed for updating the routing information. One approach for automatic route propagation and computation is called traditional approach: routers connecting a collection of networks work collectively to create the routing tables; every router uses some routing algorithms to compute and propagate some routing paths. Another approach is called software defined networking (SDN) approach: routing tables are computed and propagated by a remote central controller.
• **Distance-vector routing**

There is a class of algorithms, known as *distance-vector* (or Bellman-Ford) routing, used by routers for propagating routing information.

The idea in distance-vector routing is as follows. Initially, a router has an entry for each directly connected network in its routing table. Each entry in the table gives a destination network and the distance to the network (usually in hops). Periodically, each router sends a copy of its routing table to any other router it can reach directly. On receiving the report from a neighbor, a router updates its own routing table if the following happens: the report contains a shorter route for some entry in the current table; the report contains a route to a destination that has no entry in the current table; or the route in an entry to a destination with this neighbor as the next hop router and the distance from the neighbor to the destination has changed.

The message exchanged periodically consists of a list of pairs \((V, D)\), where \(V\) gives a destination and \(D\) is the distance to the destination.

The original core routers used a distance-vector protocol called *gateway-to-gateway protocol (GGP)*. Now GGP is no longer part of TCP/IP standards.

• **Link-state (SPF) routing**

The *link-state* or *shortest path first (SPF)* routing refers to another class of algorithms which is the primary alternative to distance-vector algorithms. The SPF algorithm requires each participating router to have complete topology information. Two routers are neighbors if they are connected by the same network, and conceptually connected by a link. Each router tests the status of all neighbor routers and periodically propagates the link status to all other routers. Each router uses the link status information and the topology of network to update its routing table by applying a shortest path algorithm (e.g., Dijkstra’s algorithm).

• **Routing in large internet**

Propagating route information among all routers in a large internet may not be practical, because there could be too many routers and the networks, and routers may not be managed by a single authority. A solution for the above problem is to limit router interaction by partitioning a large internet into groups. All routers within a group are involved in exchanging the route information of the group. The route information exchange among groups are handled by representative routers of groups.

The *delay* and *overhead* are important architecture issues for the size of groups. To decide the partition of groups, administrative issues should be considered as well.

• **Autonomous system (AS)**

An autonomous system is a group of networks and routers controlled by a single administrative authority. The Internet is partitioned into autonomous systems. The routing within an autonomous system is called *interior routing* and the routing between autonomous systems is called *exterior routing*. Each autonomous system is free to choose an *interior gateway protocol (IGP)* to handle the routing inside the autonomous system but only one *exterior gateway protocol (EGP)* is usually used for routing between autonomous systems.

Special routers called *boundary* or *border routers* handle the routing between autonomous systems. *Interior routers* handle the routing within the autonomous system. An interior router runs an
IGP to find the routes within the autonomous system. A border router runs both an IGP and an EGP. The EGP gets the information about the routes within the autonomous system from the IGP and propagates this information to the border routers in other autonomous systems. The EGP also relay the information from the border routers of other autonomous systems to the IGP which propagate the information to the interior routers in the autonomous system.

Routing within an AS

- **Interior routing**
  The routing within an autonomous system is called interior routing. Static interior routing, where the routing tables are managed manually, can be used for small slowly changing internets. For large, rapidly changing internets, dynamic interior routing (automated approach for managing routing tables) is used.

  Unlike exterior router communication, for which BGP provides a widely accepted standard, there is no single standard for interior routing. Interior gateway protocol (IGP) is used to refer any protocol used for routing in an autonomous system.

- **Routing information protocol (RIP)**
  RIP is a widely used IGP. It is simple, easy to implement, and generally adequate for small and stable autonomous systems.

  RIP operation: RIP is a distance-vector routing protocol and uses hop count metric to measure distances. It partitions the machines in the autonomous system into active ones and passive ones. Active machines advertise their routes to others periodically; passive machines do not advertise but listen to RIP messages and use them to update routing tables. Only a router can run RIP in active mode; a host must use passive mode.

  More specifically, RIP has following operations:

  1. A router initializes its routing table with an entry for each network directly connected to the router. The distance to each network is one (1 hop).
  2. A router advertises (broadcasts in RIP1, multicasts or broadcasts in RIP2) its routing table to all directly connected networks.
  3. A router $A$ updates its own routing table $T_A$ when it receives an advertisement $T_B$ from a neighbor router $B$ as follows:
     - For a destination $N$ in $T_A$, if $T_B$ contains a shorter route to $N$ then $A$ updates $T_A$ to have $B$ as the next hop router to $N$ and changes the distance to $N$ accordingly.
     - If $T_B$ contains a route to a destination $N$ which does not have an entry in $T_A$ then $A$ creates an entry for $N$ in $T_A$ and fills in the routing information for $N$ in the entry.
     - For a destination $N$ in $T_A$ with the next hop router $B$, if $T_B$ shows that the distance from $B$ to $N$ has changed then $A$ updates the distance to $N$ in $T_A$ accordingly.
  4. On receiving a new route, a router advertises the new route information immediately (called triggered update).
  5. A router advertises its routing table periodically (periodic update).

  The key points for the above operations are
Routers exchange information on the same autonomous system they are connected.
Routers exchange information only with neighbors.
Routers exchange information periodically.

An example, router $A$ has routing table and receives an advertisement from neighbor router $B$ as below:

<table>
<thead>
<tr>
<th>Routing Table of $A$</th>
<th>Advertisement from $B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dest</td>
<td>Next-hop</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>N1</td>
<td>B</td>
</tr>
<tr>
<td>N2</td>
<td>C</td>
</tr>
<tr>
<td>N3</td>
<td>D</td>
</tr>
</tbody>
</table>

After the updating, $A$’s routing table:

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next-hop</th>
<th>Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>B</td>
<td>8    /* Changed, B was the next hop router for N1 in A’s table */</td>
</tr>
<tr>
<td>N2</td>
<td>C</td>
<td>2    /* No change, B does not advertise a better route */</td>
</tr>
<tr>
<td>N3</td>
<td>B</td>
<td>4    /* Changed, B advertises a better route */</td>
</tr>
<tr>
<td>N4</td>
<td>B</td>
<td>6    /* New entry */</td>
</tr>
</tbody>
</table>

RIP uses three timers to support its operations. The periodic timer defines the route advertisement period. The value of this timer is usually 30 seconds.

Each route is given an expiration timer. It is set to 180 seconds once an update information for the route is received. If no update information is received for a route in 180 seconds, the route is considered as expired and the hop count for that route is set to unreachable.

When a route becomes expired, the route is not deleted from the routing table. Instead, RIP continues advertise the route with hop count unreachable. At the time of the expiration timer expires, a garbage collection timer is set to 120 seconds to the expired route. When the garbage collection timer expires, the route is deleted from the routing table.

One problem with RIP is slow convergence, which means a change in a network is propagated very slowly through the rest of the internet. One method to deal with this problem is to limit the maximum hops (diameter) of an autonomous system to 15 (16 for unreachable).

A more important problem with RIP is instability. The following approaches can be used to deal with the instability problem, but none of them is perfect.

Triggered update: If there is no change in the network, updates are sent in regular period. However, the router reports the change of a network immediately.

Split horizons: If a router has received a route update information from an interface, the router does not send this same updated information back through the same interface.

Poison reverse: It is a variation of split horizon.
• **RIP1 (version 1) message**
  The message has the following fields.
  Command: defines the type of the message, 1 for request, 2 for response.
  Version: 1 for RIP1.
  Family: gives the family of the protocol used, 2 for TCP/IP.
  Network address: IP address of the destination network. RIP1 uses IP broadcasting to advertise routing tables.
  Distance: the hop count to the destination network, the value is between 1 and 16, where 16 denotes infinity (no route exists).
  The family, network, and distance fields can be repeated.
  RIP1 can not be used to propagate either variable-length subnet addresses or classless addresses used in CIDR.

• **RIP2 extension**
  RIP2 extends RIP1 by including a subnet mask field and a next hop field for each network address. By this extension, RIP2 can deal with variable-length subnet addresses and classless addresses, and offers improved resistance to errors. Also RIP2 may use multicasting to advertise routing tables.
  Compared with RIP1 message, RIP2 message has the following additional fields.
  Route tag: it usually gives the autonomous system number of the network.
  Subnet mask: it gives the subnet mask used in the network.
  Next hop address: it gives the IP address of the next hop network.

• **Encapsulation**
  RIP uses UDP with port 520. The length of an RIP message is determined in UDP.

• **Hello protocol**
  Hello protocol is a distance-vector routing protocol but uses link delay as the metric for calculating routes. It is now obsolete but was important in the history of the Internet.

• **Open SPF protocol (OSPF)**
  OSPF is an open standard link-state routing protocol. It is more appropriate to use OSPF in large autonomous systems with more complex topology and less stability.
  In link-state routing algorithm, a router tests the status of the links connected to it and propagates the link status periodically to all the other routers. Each router keeps the topology of the network and calculates the routes based on the network topology and the link status by a shortest path algorithm.
  In OSPF, an autonomous system is modelled by a weighted directed graph. The nodes of the graph denote routers and networks. There is a directed edge from a node $u$ to node $v$ if there is a connection from $u$ to $v$ in the autonomous system. A network is called a *stub network* if there is only one router to connect the network to the other networks. A stub network is represented in the directed graph by two nodes, one for the router and one for the network, and one direct
edge from the router to the network. To reflect the reality in network routing, each edge from a router to a network is given a cost and no cost is given to an edge from a network to a router. The information a router propagated is the destinations reachable by a single link and the cost of the link.

Major operations of OSPF include:

- A router uses a **hello process** to test the reachability of neighbors and to set-up the neighborhood relationships.
- A router periodically sends the states of the links directly connected to itself to all the other routers in the same autonomous system (or the same area of an AS).
- A router uses the link-state from other routers to build the link-state database (the weighted graph) for the AS.
- A router finds the routes for its routing table by computing the shortest paths in the graph.
- A router can request the information on specific links.

In OSPF, a router exchanges the states of directly connected links with every router in the AS (or area if the AS is large).

If the autonomous system is large, traffic load and delay may be problems for the propagation of link status. To solve these problems, OSPF partitions an autonomous system into **areas**. Routers within an area flood the link status. Special routers, **area border routers**, summarize the information of the area and send it to other areas. There must be a special area, called **backbone area**, in the partition, and all other areas must be connected to the backbone.

Each area is given an area identification number. OSPF allows the administrator to assign a metric to each route. The metric can be based on a type of service like minimum delay, maximum throughput, etc. A router can have multiple routing tables, each based on a different type of service.

**OSPF message**

OSPF uses five different types of messages: hello message, database description (topology) message, link state request message, link state update message, and link state acknowledgment message. All OSPF messages use a common message header which contains the following fields.

**Version:** specify the version of OSPF, currently 2.

**Type:** specify message type, 1 for hello, 2 for database description, 3 for link state request, 4 for link state update, and 5 for link state acknowledgement.

**Message length:** total length of the message including header.

**Source router IP address:** IP address of the router who sends the message.

**Area ID:** specify the area where the routing takes place.

**Checksum:** for error detection.

**Authentication type:** types of authentication are defined currently, 0 for none and 1 for password.

**Authentication:** give authentication data, currently all 0 for type 0 and an eight characters password for type 1.

Hello message is used to create and test neighbor reachability.
Database description (topology) message is used to initialize the network topology database in routers.

Link status request message is used to request information on specific route or routes (for updating the topology database). It is replied by a link status updating message.

Link status update message is used by a router to advertise the states of its links. This message is the heart of the OSPF operation. There are five different link state advertisements (LSAs) in OSPF.

- Router link LSA is used to advertise the links from to a router.
- Network link LSA is used advertise the links from a network.
- Summary link to network LSA is used by an area border router to advertise the existence of other networks outside the area. One packet is used for one network.
- Summary link to autonomous system boundary router LSA is used to announce the route to an autonomous system border router.
- External link LSA is used to announce the networks outside the autonomous system.

Link status acknowledgement message is used to acknowledge the receipt of every link state update message (make the routing more reliable).

- **Encapsulation**
  
  OSPF messages are encapsulated in IP datagrams. They have the acknowledgment mechanism for flow and error control.

- **Routing among ASs**

  - **Exterior gateway protocol (EGP)**

    An EGP is a protocol used for passing routing info between autonomous systems. Currently, an EGP called *border gate-way protocol (BGP)* is used in most TCP/IP internets. The exchange of information between autonomous systems are handled by the border routers or peers of the systems.

  - **BGP characteristics**

    BGP is not a distance-vector protocol nor a link-state protocol. It is a *path vector routing* protocol. In path vector routing, each entry in the routing table contains the destination network, the next router, and the path to reach the destination. The path is usually defined by an ordered list of autonomous systems that a packet should travel through to reach the destination. Because the metrics used in different autonomous systems may not be comparable, the efficiency of routing in BGP is simply the connectivity, a destination is reachable using a given route or it is not reachable. On the other hand, BGP provides more administrative controls as shown below.

    BGP’s primary role is to provide inter-autonomous system communication. If there are multiple BGP speakers in an autonomous system, BGP provides the coordination among the multiple BGP speakers to guarantee the consistency of information given by the BGP speaker.

    BGP provides the propagation of reachability information to allow an autonomous system to advertise the reachable destinations through the system and learn such information from other systems.
BGP provides next-hop information for each destination.
BGP provides the policy support to allow a more flexible format for the next-hop information.
BGP uses TCP to provide reliable transport.

In addition to the reachable destinations and the next-hop information for them, BGP provides path information that allows a receiver to know the autonomous systems along a path to the destination.

BGP uses incremental update to minimize the bandwidth used.
BGP supports classless (CIDR) addressing.

BGP allows route aggregation for minimizing the bandwidth used.
BGP allows a receiver to verify the identity of a sender (authentication).

• **BGP functionality**

BGP peers perform three basic functions: (1) initial peer acquisition/authentication, peers set-up a TCP connection for this; (2) send reachability information; and (3) verify connections between peers/networks are functioning correctly.

• **BGP messages**

All BGP packets have the same common header with a 16 bytes *marker field* (used for synchronization and authentication), a two bytes *length field* to specify the whole message length, and a one byte *type field* to define the message type. There are four types of messages.

• **Open message (type 1)**

The message used to open a TCP connection with a peer. Following the common header, there are following fields in the message.

Version, one byte, gives the version of BGP, the current version is 4.
Autonomous system number, autonomous system number of the sender.
Hold time, the maximum time that the receiver should wait for a message (keepalive or update) from the sender.
BGP identifier, unique identifier for the sender, usually an IP address.
Option parameter length, the length of total option parameters.
Option parameters, the only option defined so far is authentication.

• **Update message (type 2)**

Update message is used by a peer to withdraw destinations that have been advertised previously, advertise a route to a new destination, or both. One update message can withdraw multiple routes but can only announce one new route. The message has the following fields.
Unfeasible routes length, defines the length of the withdraw destination field.
Withdraw destinations, gives all routes (in compressed mask-IP-address pair) that should be deleted.
Path attributes length, defines the length of the path attributes field.
Path attributes, gives the attributes of the announced new route. Destination networks, gives the compressed mask-IP-address pairs of the route to the destination. The compressed mask-IP-address pair is a data structure which provides the mask and IP address in a compressed form.

- **Notification message (type 3)**
  The message is used for error report and has the following fields.
  - Error code, defines the category of errors
  - Error subcode, further defines the types of error in each category.
  - Error data, gives more information about errors.

- **Keepalive message (type 4)**
  Two BGP peers periodically exchange keepalive message to test network connectivity and to verify both peers continue to function.

- **Restriction of BGP**
  BGP only provides reachability information. It does not advise a better route.

- **Encapsulation**
  BGP uses TCP service on port 179.

- **BGP extensions**
  BGP was designed for IPv4. Multiprotocol extensions for BGP allows BGP to advertise destinations with address families other than IPv4. These address families include IPv6 addresses and multi-protocol label switching (MPLS, will be discussed later). The idea in multiprotocol extensions is simple, replacing the IPv4 addresses by the addresses of the advertised destinations in the Path Attributes. For this purpose, new path attribute types (called *Network Layer Reachable Information (NLRI)*) are created. There are two new types: Multiprotocol Reachable NLRI (type 14) and Multiprotocol Unreachable NLRI (type 15). Figure 13.14 (page 283) of the textbook gives the data format for Multiprotocol Reachable NLRI.

**Label Switching**

- **IP switching and MPLS**
  The IP forwarding algorithm described in previous chapters finds a longest-prefix matching between the destination IP address of a datagram and the IP address of a destination network in a routing table to determine the next hop router for forwarding the datagram. The longest-prefix matching may have a big overhead in routers with large routing tables. IP switching is an alternative approach for datagram forwarding to avoid the overhead.

  The basic ideas for reducing the look-up overhead is follows: It takes $\log N$ time to search a set of $N$ items (as in the routing table look-up) while it takes only constant time to search an array element if the index of the array element is given. To realize the high speed indexing, the IP switching uses the connection-oriented forwarding. For the datagrams in the same *flow*, each datagram is assigned a unique label (an integer). A router keeps the next hop router for each
active flow in an array which are indexed by the labels of active flows. To keep the array small, a router may change the labels (called label switching) for the flows it forwards.

**Multi-Protocol Label Switching (MPLS)** is a current standard for IP switching. MPLS works as follows: Routers which connect end users use conventional (routing table look-up) forwarding, while routers at center of a network understand MPLS and use switching. For example, data from a host $A$ to host $B$ may be transmitted like this: The datagrams from $A$ are forwarded by a conventional router to a router $R_0$ which uses MPLS. Router $R_0$ establishes a virtual path consisting of a sequence of routers $R_0, R_1, ..., R_k$ using MPLS and assigns the datagrams in a flow on the path a label $L$ (this is known as MPLS ingress). Datagrams with label $L$ are forwarded by routers $R_i$ ($0 \leq i \leq k - 1$). Router $R_k$ removes the label and forwards datagrams of $A$ to a conventional router which further forwards the datagrams to $B$. The operation of $R_k$ is known as MPLS egress.

MPLS does not require the connection-oriented networks. MPLS appends an MPLS header to an IP datagram for the switching. The MPLS header has a label field which contains the label for a flow.

A router that implements MPLS is called a **Label Switching Router** (LSR). Usually, an LSR can handle both conventional forwarding and IP switching, and can serve as an interface between a non-MPLS internet and an MPLS core (a contiguous set of routers using MPLS). The array (table) for switching in an LSR is called Next Hop Label Forwarding Table and each entry in the table is called Next Hop Label Forwarding Entry (NHLFE).

### Generalized Forwarding and SDN

- **Simple match-forwarding approach**
  
  **Match:** on receiving an IP datagram $D$, the Internet Protocol uses the destination IP address in $D$’s header and IP addresses in a routing table to make a forwarding decision for $D$. **Forwarding:** the Internet Protocol may forward $D$ to an output link (interface) or discard $D$.

- **Generalized match-action approach**
  
  **Match:** Forwarding decisions are made based information in the headers of data packets at different layers and a match-action table. **Action:** data packets can be forwarded to an output link, modified, filtered, or handled in other ways. A match-action table is usually computed by a remote central controller.

- **Software defined networking (SDN)**
  
  To meet many applications requirements, connection oriented networking technologies (e.g., MPLS) were developed to classify communication traffic into data flows and set-up a dedicated routing path for each flow. Routing overlay technologies allow the routing paths created based on a virtual network topology and data streams forwarded based on the existing protocols in the nodes (routers) of the virtual topology. SDN combines the connection oriented networking technologies and the routing overlay technologies. SDN provides flow-based forwarding scheme (using match-action approach), and separates data plane and control plane. Network control functions are distributed in remote controllers (which compute the match-action tables and propagate the tables to routers) and data plane routers (which perform data forwarding based the match-action tables). SDN provides a platform for programmable networks.