4.3 Routing protocols

We first look at Routing Tables and routing mechanisms.

A routing table has columns for the destination network (or hosts) with the corresponding cost and the next router address to reach a destination. Additional information may vary depending on the routing protocol being used.

Distance vector (DV) routing

In the distance vector-based routing algorithm, (Bellman-Ford algorithm), routers pass periodic copies of a routing table to other routers to indicate the changes in the topology. Each router receives a routing table from its direct neighbor.

One basic idea for DV algorithm is as follows. For two hosts v and y, let $d(v, y)$ denote the least cost path from v to y. For a host x and its neighbor v the cost from x to v is $c(x, v)$. Then

$$d(x, y) = \min\{c(x, v) + d(v, y) : v \text{ is a neighbor of } x\}. \qquad (1)$$

Initialization:

```
for all destination y in N:
 D(x, y) = c(x, y)
  // if y is not a neighbor, c(x,y) = infinite.
for each neighbor w
 D(w, y) = ?
for each neighbor w
 send vector D(x) = [D(x, y) : y in N] to w
```

loop

```
wait (until a link cost change to some neighbor w,
       or received a distance vector from neighbor w)
for each y in N:
D(x, y) = min{c(x, v) + D(v, y): v is a neighbor of x}
if D(x,y) changed for any destination y
 send distance vector D(x) = [D(x, y): y in N] to all neighbors
```

To understand the algorithm, let us look at a simple example of network in Figure 1. In this network, there are 4 nodes and the weights (cost) for the links are different.

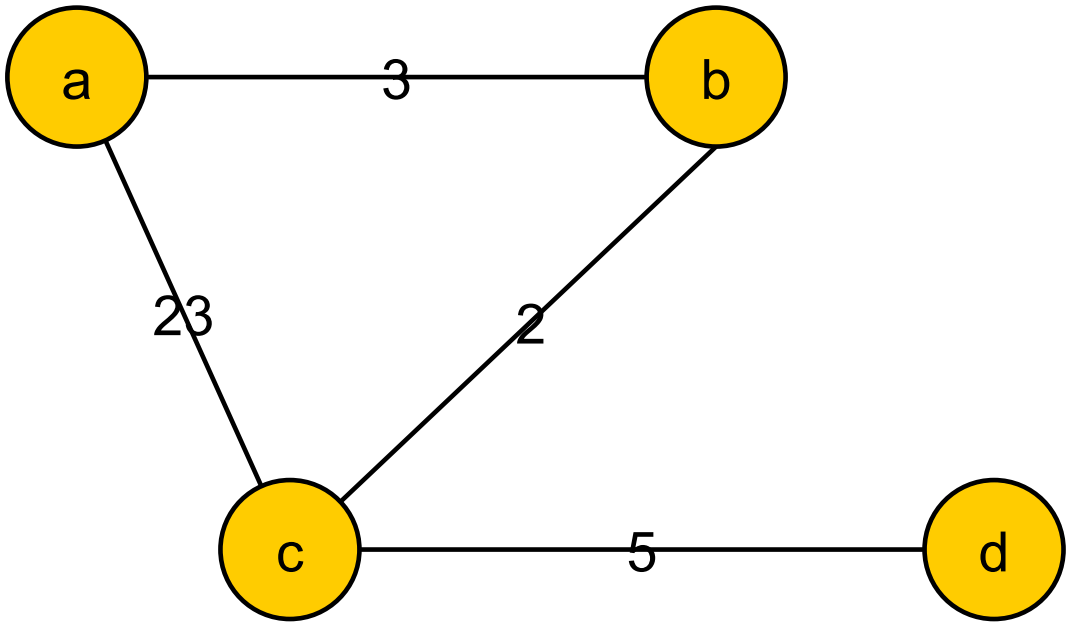

Figure 1: An example for DV algorithm

After initialization, the vectors in each node are as in Table 1.

D(a)	(a,b):3	(a,c):23	(a,d):∞	(b,c):?	(b,d):?	(c,d):?
D(b)	(a,b):3	(b,c):2	(b,d):∞	(a,d):?	(c,d):?	(a,c):?
D(c)	(a,c):23	(b,c):2	(c,d):5	(a,b):?	(b,d):?	(a,d):?
D(d)	(c,d):5	(b,d):∞	(a,d):∞	(c,d):?	(a,b):?	(c,b):?

Table 1: Initialized vectors

After node a received the vectors from its neighbors b and c, node a will update its vector as follows. The vector D(b) tells that D(b, c) = 2 and the vector D(c) tells that D(c, d) = 5. So node a can compute an update as:

D(a)	(a,b):3	(a,c):5	(a,d):10	(b,c):2	(b,d):?	(c,d):5

The DV routing algorithm is decentralized algorithm. In this algorithm, each node does not need to know the topology of the network. All the updates are performed by communicating to the neighbors. It can be proved that the distance vectors will be convergent to the least costs.

After the router has a complete vector, then it will be easy to find which way to forward a packet by using the equation (1). In fact, the equation tells which neighbor v should the node forward to.

The above discussion assumes that the cost for each link is stable. If the cost changes, then the above algorithm will have problems. If some cost is reduced, then the above algorithm can easily get the updated vectors. However, if some link cost increases, then the above algorithm will take a lot of loops to update. Some technique called poisoned reverse has been used to solve this problem. We omitted the details here.

Link-state (LS) routing

This routing algorithm is also know as shortest-path-first (SPF) algorithm.

Link sate routing uses link state advertisements(LSAs), a topological data base, the SPF algorithm, the resulting SFP tree, and finally, a routing table of paths and ports to each network.

Each router does the following

- Keeps track of its neighbors: the neighbor's name, it is up or down, the the cost of the link to the neighbor.
- Construct an LSA packet that lists its neighbor router names and link costs. This includes new neighbors, changes in link costs, and links to neighbors that have gone down.
- Send out this LSA packet so that all other routers receive it.
- Records each LSA packet it receives in its database immediately, to ensure that it has the most recently generated LSA packet from every other router.
- Using accumulated LSA packet data to construct a complete map of the internetwork topology, proceeds from this common starting point to return the SPF algorithm and compute routes to every network destination.

The routing table updates in the link state routing method use the following process.

- Routers exchange LSAs with each other, Each router begins with directly connected networks for which it has direct information.
- Each router, in parallel with the others, constructs a topological database consisting of all the LSAs from the internetwork.
- The SPF algorithm computes network reachability, determining the shortest path first to each other network in the link state protocol internetwork. Each router constructs this logical topology of shortest paths as SPF tree. With itself as root, this tree expresses paths from the router to all destinations.
- Each router lists its best paths and the ports to these destination networks in the routing table. It also maintains other database of topology elements and status details.

Routing policies

There are mainly two policies to form the routing tables, manual and automatic.

In the manual policy, the system administrator sets a router's table at start-up. This is called static routing. When the router receives a packet with a particular destination address, it looks at the table for the next hop. If the address is not found in the table, the router forwards the packet to a default router connected to it.

In the other case, the routers accept information from other router to periodically update their entries. This type of routing policy is called dynamic routing.

Static router are explicitly configured and entered into the routing table. When the router uses a combination of static and dynamic policies, static routes take precedence.

LAN-to-LAN and LAN-to-WAN routing

One of the main problem for internet routing is the size of the internet. The number of hosts in the internet is huge. Therefore it is not possible for a router to have a table for all nodes on the internet. For example, we cannot use LS routing to find the path for every node in the internet because we cannot form a graph which includes all nodes in the internet.

In internet routers are organized into *autonomous systems* (ASs). Each AS consists of a group of routers that are typically under the same administrative control and running the same routing algorithm.

Intra-AS routing

The routing algorithm running within an autonomous system is called an intra-autonomous system routing protocol.

We will investigate two Intra-AS routing:

- Routing Informatin Protocol (RIP).
- Open Shortest Path First (OSPF).

RIP is specified in RFC 1058. In RIP, costs are actually from source router to a destination subnet. RIP uses the term hop, which is the number of subnets traversed along the shortest path from the source router to destination subnet, including the destination subnet. The maximum cost is limited to 15, thus limiting the use of RIP to AS that are fewer than 15 hops in diameter.

In RIP, routing updates are exchanged between neighbors approximately every 30 seconds using a RIP response message (or RIP advertisement). The response message contains a list of up to 25 destination subnets within the AS, as well as the sender's distance to each of those subnets.

If a router does not hear from its neighbor at least every 180 seconds, that neighbor is considered to be no longer reachable.

Routers send RIP request and response messages to each other over UDP using port number 520.

OSPF is defined in RFC 2328. It is based on link-state information and a Dijkstra's least-cost path algorithm.

With OSPF, a router constructs a complete topological map of the entire AS. Then the router uses the Dijkstra algorithm to determine the shortest paths tree with itself as the root node. Individual link costs are configured by the network administrator.

With OSPF, a router broadcasts routing information to all other routers in the AS. A router broadcasts link-state information whenever there is a change in a link's state. It also broadcasts a link's state periodically, even if the link's state has not changed. OSPF advertisements are contained in OSPF messages that are carried directly by IP. The OSPF protocol also checks that links are operational (via HELLO message that is sent to an attached neighbor) and allows an OSPF router to obtain a neighboring router's database of network-wide link state.

An OSPF autonomous system can be configured hierarchically into areas. Each area runs its won OSPF routing algorithm, with each router in an area broadcasts its link stake to all other routers in the area.

Within the area, one or more area border routers are responsible for routing packets outside the area. One OSPF area in the AS is configured to be the backbone. The main function of the backbone area is to route traffic between the other areas in the AS.

The backbone area consists of all the border routers in the AS and may contain some other routers.

OSPF also defines some advanced features. It considered the security of the routing, so passwords among the routers are used.

Inter-AS routing means routing among different ASs.

Internet uses the Border Gateway Protocol (BGP) (RFC4271, 4274) for inter-AS routing.

BGP is one of the most important protocol for Internet. Without that protocol, the ASs cannot be glue together and the Internet cannot be formed. On the other hand, BGP is a very complicated protocol and there are entire books for BGP.

In BGP, pairs of routers exchange routing information over semi-permanent TCP connections using port 179.

There is typically one BGP TCP connection for each link that directly connects two routers in two different ASs.

There are also semi-permanent BGP TCP connections between routers within an AS. A common configuration is one TCP connection for each pair of routers internal to an AS.

For each TCP connection, the two routers at the end of the connection are called BGP peers, and the TCP connection along with all the BGP messages sent over the connection is called a BGP session.

A BGP session that spans two ASs is called an external BGP (eBGP) session and a BGP session between routers in the same AS is called an internal BGP (iBGP) session.

BGP allows each AS to learn which destinations are reachable via its neighboring ASs. In BGP, destinations are not hosts but instead are netids (prefix). We will use Figure 2 as an example to explain the BGP session.

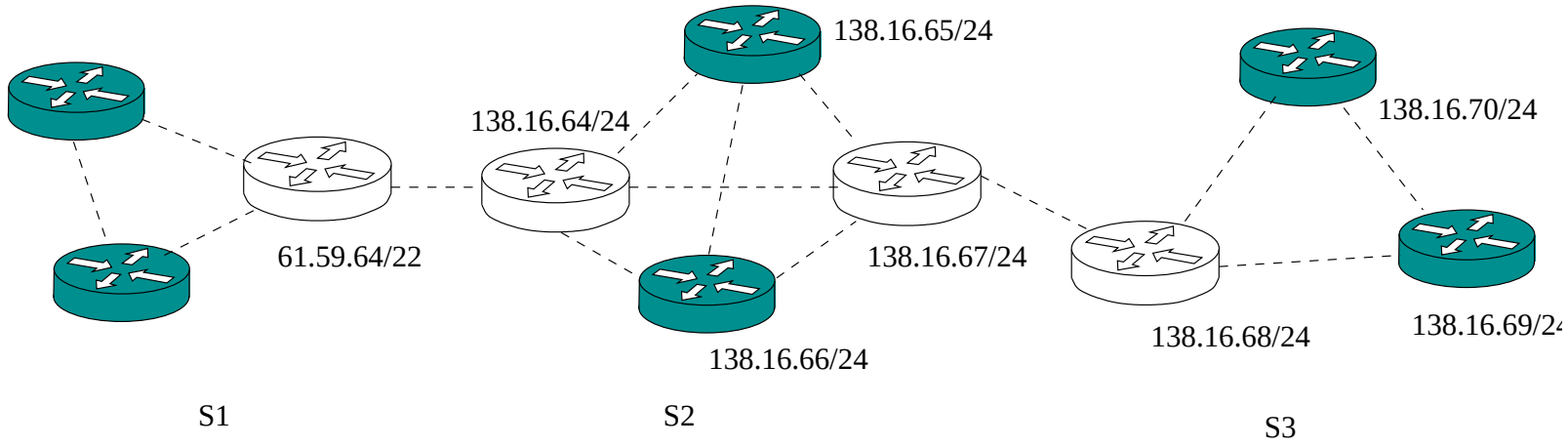

Figure 2: BGP sessions

In this example, there are 3 ASs, each of AS contains several routers. The dashed lines in the picture are not physical links, but BGP TCP semi-permanent connections.

The white routers are BGP peers. There are four subsets attached to AS2: 138.16.64/24, 138.16.65/24, 138.16.66/24, and 138.16.67/24. AS2 will aggregate the netid for these four subnets and use BGP to advertise the single netid to 138.16.64/22.

Similarly, the AS3 can advertise the netid 138.16.68/22. In this case, if the subnet 138.16.71/24 is at some AS other than AS3, it can still advertise 138.16.71/24, because routers use longest-prefix matching for forwarding datagrams.

eBGP sessions between connected BGP peers will send reachable netid information each other. On the other hand, when a BGP peer receives eBGP-learned prefixes, the peer router uses its iBGP sessions to distribute the prefixes to the other routers in the AS. When a router learns about a new prefix, it creates an entry for the prefix in its forwarding table.

In BGP, an AS is identified by its globally unique autonomous system number (ASN) (RFC 1930). ASNs, like IP addresses, are assigned by ICANN (Intenet Corporation for Assigned Names and Numbers) regional registries.

When a router advertises a prefix across a BGP session, it includes with the prefix a number of BGP attributes. These information are called a route.

- The attribute AS-PATH contains the ASs through which the advertisement for the prefix has passed. When a prefix is passed into an AS, the AS adds its ASN to the AS-PATH. For example, if AS3 advertises its prefix to AS2 and then AS2 advertises the prefix to AS1, then the AS-PATH will be AS3 AS2. Routers use the AS-PATH attribute to detect and prevent looping advertisements. If a router sees that its AS is contained in the path list, then it will reject the advertisement. Routers also use the AS-PATH attribute in choosing among multiple paths to the same prefix.

- The NEXT-HOP is the IP address of the router interface that begins the AS-PATH. For example, when the peer in AS1 advertises some AS-PATH to AS2 using eBGP, it contains the IP address of its interface as the NEXT-HOP attribute. After routers inside AS2 learned the advertisement (route) by iBGP, it can use that IP address as forwarding address if it wants to use that path. So the router can put it into its forwarding table.
- It is possible that two ASs are connected by two peering links. The two routers could have the same AS-PATH bu different NEXT-HOP values. In this situation, the router can use the NEXT-HOP values and the intra-AS routing algorithm to determine the cost of the path to each peering link and determine the appropriate interface to use.

By eBGP and iBGP, a router may learn about more than one route to a prefix. In this case, the route must select one of the possible route. The input of the route selection processor is the set of all routes that have been learned and accepted by the router (the router usually has its import policy to decide whether accept or filter the route).

- Routes are assigned a local preference value as one of their attributes. The local preference of a route could be set by the router or by another router in the same AS. This is a policy decision that is left up to the AS's network administrator. The routes with the highest local preference values are selected.
- From the remaining routes (all with the same local preference value), the route with the shortest AS-PATH is selected. If this rule were the only rule for route selection, then BGP would be using a DV algorithm for path determination, where the distance metric uses the number of AS hops rather than the number of router hops.

- From the remaining routers (all with the same local preference values and the same AS-PATH length), the route with the closest NEXT-HOP router is selected.
- If more than one route still remains, the router uses BGP identifiers to select the router.

Broadcast and multicast routing

Broadcasting information for a host in the network means sending the information to every host on the network. It is not realistic to use the unicast (point-to-point) routing methods discussed previously for the broadcast routing. This is not just because using unicast routing for broadcast is not efficient, but also because a host usually does not know addresses of all the hosts on the network before broadcasting.

We will view the network as a graph $G = (N, E)$, where the vertices N contains all the hosts on the network and the edges contains all the links of the network.

Controlled flooding

To flood a packet means a host sends the packet to all of its neighboring hosts and the neighboring host forwards the packet to its neighbors except the neighbor from which it received the packet. If the network is acyclic (i.e., the network is a tree), then the flooding method is an efficient way.

Unfortunately, usually the network contains a lot of cycles, that makes the flooding method problematical. In fact a packet may be forwarded along the cycle forever. To avoid dead loops for flooding, several methods are used to control the flooding.

Sequence-number-controlled flooding: A source node puts its address (or other unique identifier) as well as a broadcast sequence number into a broadcast packet. Each node maintains a list of the source address and sequence number of each broadcast packet it has already received, duplicated and forwarded. When a node receives a broadcast packet, it checks whether the packet is in this list. If so, the packet is dropped, otherwise the packet is duplicated and forwarded to all the node's neighbors (except the node from which the packet has just been received).

Reverse path forwarding (RPF): When a router receives a broadcast packet with a given address, it transmits the packet on all of its outgoing links (neighbors except the one on which it was received) only if the packet arrived on the link that is on its own shortest unicast path back to the source. Otherwise, the router simply discards the incoming packet. Such a packet can be dropped because the router knows it either will receive or has already received a copy of this packet on the link that is on its own shortest path back to the sender.

The RPF does not use unicast routing to deliver a packet to a destination, nor does it require that a router know the complete shortest path from itself to the source. RPF need only know the next neighbor on its unicast shortest path from itself to the source and determine whether or not to flood a received broadcast packet.

Spanning-tree broadcast

Using controlled flooding still cannot completely avoid transmission of redundant broadcast packets. Some nodes will receive duplicated packets from different paths and discards all except one.

Ideally, every node should receive only one copy of the broadcast packet.

If the network is a tree, then the simple flooding method can achieve the broadcast. So the concept of spanning tree of a connected graph can be used here.

A spanning tree of a connected graph $G = (N, E)$ is a graph $G' = (N, E')$ such that E' is a subset of E, G' is connected and contains no cycles. If each link has an associated cost (the graph is then called weighted graph) and the cost of a tree is the sum of the link costs, then a spanning tree whose cost is the minimum of all the graph's spanning tree is called a minimum spanning tree.

One approach to the broadcast routing is first constructing a spanning tree of the network (minimum will be better).

There are many algorithms for finding a spanning tree for a graph. One simple method is called center-based approach (but it will not create the minimum spanning tree in general).

In this method, a center node (also know as a rendezvous point or a core) is defined. Nodes then unicast tree-join messages addressed to the center node. A tree-join message is forwarded using unicast routing toward the center until it either arrives at a node that already belongs to the spanning tree or arrives at the center. In either case, the path that the tree-join message has followed defines the branch of the spanning tree between the edge node that initiated the tree-join message and the center.

Multicast

Multicast service means that some message is delivered to a subset of network nodes. In this case, both unicast and broadcast are not efficient.

A number of emerging network applications require the delivery of packets from one or more senders to a group of receivers. Examples include bulk data transfer (software upgrade), streaming continuous media (a live lecture to lecture participants), shared data applications (teleconference), data feeds (stock quotes), interactive gaming (multiplayer games), etc.

In multicast communication, we first need to decide two things: how to identify the receivers of a multicast packet and how to address a packet sent to these receivers.

It is not practical to include all the IP addresses of the receivers in the packet, if the size of the receivers is large.

In the Internet architecture, a multicast packet is addressed using address indirection. That is, a single identifier is used for the group of receivers, and a copy of the packet that is addressed to the group using this single identifier is delivered to all of the multicast receivers associated with that group.

In the Internet, the single identifier that represents a group of receivers is a class D (first octect 224-239) multicast IP address. To manage the multicast addresses, an Internet Group Management Protocol (IGMP) is developed.

Internet Group Management Protocol (IGMP) defined in RFC 3376 operates between a host and its directly attached router (one-hop router).

IGMP provides the means for a host to inform its attached router that an application running on the host wants to join a specific multicast group.

Given that the scope of IGMP interaction is linited to a host and its attached router, another protocol is clearly required to coordinate the multicast routers throughout the Internet.

Therefore, the multicast consists of two components: IGMP and mlticast routing protocols.

IGMP has three message types.

- `membership_query` message is sent by a router to all hosts on an attached interface to determine the set of all multicast groups that have been joined by the hosts on that interface.
- `membership_report` message is used to response the `membership_query` message. This message can also used by a host when an application first joins a multicast group without the query message from the router.
- `leave_group` message is used to indicate the host's leaving of the group. This message is optional. If a host does not use `membership_report` message to reply the `membership_query` message, then the router will change the state of the host as leaving.

Several multicast routing protocols have been proposed.

Basic idea is to find a tree of links that connects all of the routers that have attached hosts belonging to the multicast group. Multicast packets will then be routed along this tree from the sender to all of the hosts belonging to the multicast group.

Some methods are used for forming the tree.

- Multicast routing using a group-shared tree: based on building a spanning tree that includes all edge routers with attached hosts belonging to multicast group. Usually, a center-based approach is used to construct the multicast routing tree, with edge routers with attached hosts belonging to the multicast group sending join messages addressed to the center node. A join message is forwarded using unicast routing toward the center until it either arrived at a router that already belongs to the tree or arrived at the center.
- Multicast routing using a source-based tree: each source in the muticast group constructs a routing tree within the multicst group. In practice, an RPF (reverse path forwarding) algorithm is used to construct a multicast forwarding tree for multicast datagrams originating at that source.

4.4 User datagram protocol (UDP)

UDP is a simple connectless protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols.

UDP does not perform handshake and mainly just indicate the port numbers for the process.

Each UDP message contains the port information for source and destination machine. Port numbers are used to keep track of different conversations crossing the network at the same time.

Application software developers agree to use well-known port numbers to do specific tasks that are defined in RFC 1700. Conversations that do not involve an application with a well-known port number are assigned numbers randomly chosen from within a specific range:

- Numbers below 255 are for public applications.
- Numbers from 255 to 1023 are assigned to companies for salable applications.
- Numbers above 1023 are unregulated.

Protocols that use UDP include Trivial File Transfer Protocol (TFTP), Simple Network Management Protocol (SNMP), Network File System (NFS), and the domain name system (DNS), Routing Information Protocol (RIP), etc.

The UDP frame consists of 16-bit source and destination ports numbers. The other fields include the checksum (16 bits) and the data length (16 bits), allowing a maximum of 64 KB of data (including the header). The minimum value of length field is 8. The frame format is shown in Figure 3.

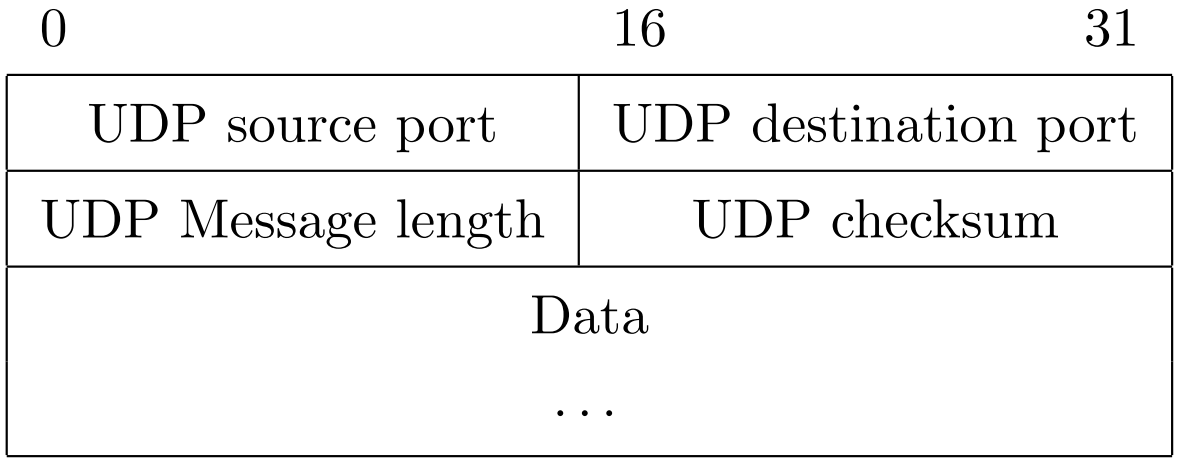

Figure 3: UDP frame format

In IPv6 Jumbograms, it is possible to have UDP packets of size greater than 64KB. This allows for a maximum length value of 4,294,967,295 bytes ($2^{32} - 1$) with 8 bytes representing the header and 4,294,967,287 bytes for data.

UDP is an efficient transport protocol, but it is not reliable transport protocol and not including congestion control. That may cause some problem.

The checksum field is used for error-checking of the header and data. If no checksum is generated by the transmitter, the field uses the value all-zeros. However, this field is not optional for IPv6.

As an example, suppose that we have three 16-bit words:

`0110011001100000`
`0101010101010101`
`1000111100001100`

Adding these three words we have the sum

`0100101011000010`

Note that the addition had overflow, which was wrapped around. The 1s complement of the sum is `1011010100111101` which is the checksum. When the receiver checks the correctness of the message, he add all the 16-bits words together with the checksum. If the result is `1111111111111111`, then the message is correct.

4.5 Transmission control protocol (TCP)

TCP is one of the most widely used transport protocols on the Internet. It provides a reliable and in-order bytes stream between two ends as it frees applications from worrying about missing or reordered data.

It also provides a flow control mechanism for the byte stream. The flow control mechanism allows the receiver to limit the number of bytes it accepts from the sender. TCP is a connection-oriented protocol meaning that the data transfer is preceded by a connection establishment phase, and a connection termination phase follows the data transfer.

TCP is a protocol at the transport layer. So it is implemented at the end systems, but not in the intermediate device such as switches and routers. The intermediate devices do not maintain TCP connection state. It is not an end-to-end TDM or FDM circuit, or a virtual circuit.

A TCP connection provides a full-duplex service. With a TCP connection, either end system can send datagram to the other end system. So each end system maintains TCP send buffer and receive buffer.

TCP is a reliable transmission protocol based on IP. It allows multiple application programs on one machine to communicate to one another through a demultiplexing operation.

The demultiplexing operation makes it possible for two or more application programs running on same or different hosts to simultaneously carry out a data transfer. For this purpose, it uses port numbers like UDP.

The connection is considered to be an abstraction consisting of a virtual circuit between two applications running on usually two different machines. The host machine address and port number for each machine serve as the end points of such a virtual circuit.

The frame format of a TCP message is shown in Figure 4.

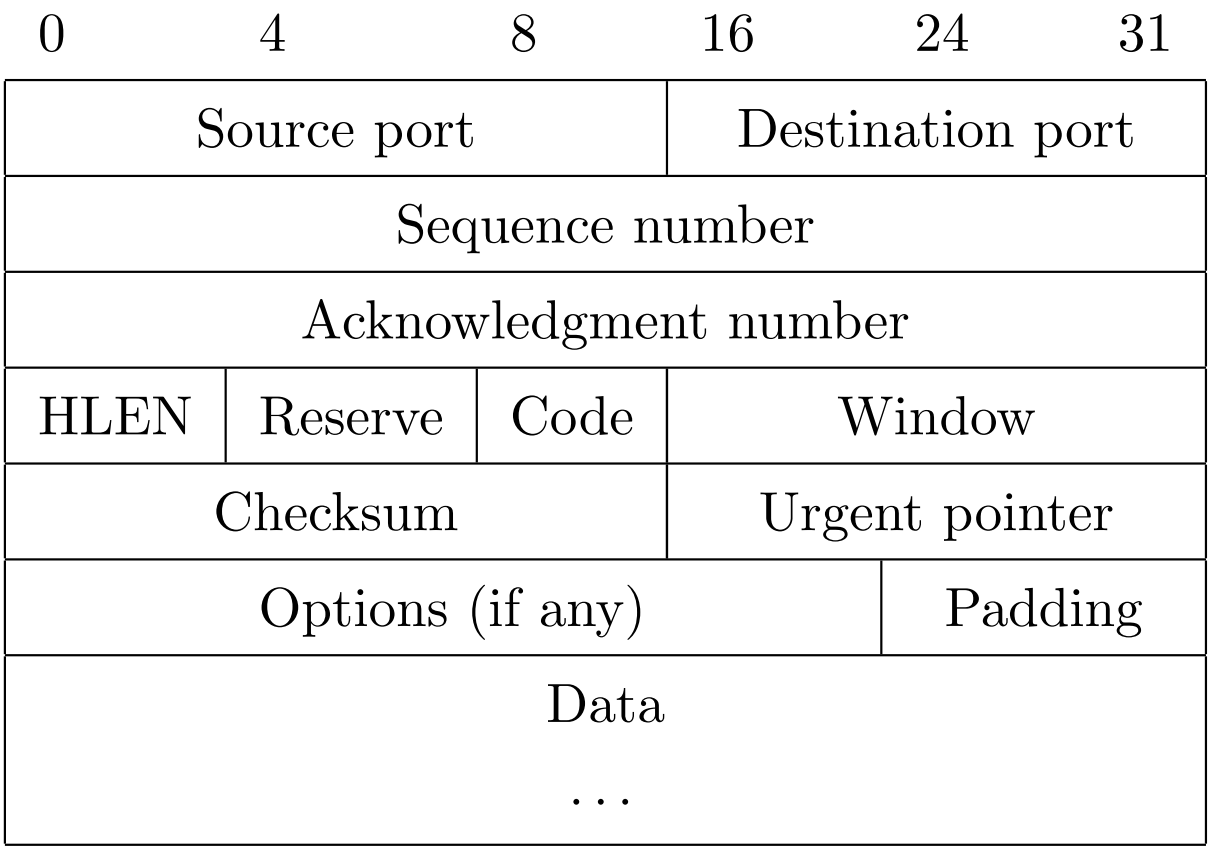

Figure 4: TCP frame format

- The sequence number identifies the position in the sender's byte stream of the data in the segment. If the SYN flag (see code below) is set (1), then this is the initial sequence number. The sequence number of the actual first data byte and the acknowledged number in the corresponding ACK are then this sequence number plus 1. TCP views data as an unstructured, but ordered, stream of bytes. So the sequence number for a segment is the byte-stream number of the first byte in the segment. If the SYN flag is clear (0), then this is the accumulated sequence number of the first data byte of this segment for the current session.

The maximum amount of data that can be placed in a segment is limited by the maximum segment size (MSS). The MSS is typically determined by the length of the largest link-layer frame that can be sent by the local sending host (maximum transmission unit, MTU). The MSS is usually set to ensure that a tCP segment will fit into a single link-layer frame. Both Ethernet and PPP link layer protocols have an MTU of 1500 bytes. Thus a typical value of MSS is 1460 bytes.

For example, suppose Host A wants to send some data which consisting of $500,000$bytes, and the MSS is $1,000$ bytes. Suppose the first byte of the data stream is numbered 0. then the second segment gets assigned sequence number 1,000, the third segment gets assigned sequence number 2,000 and so on.

- The acknowledgment number identifies the number the source expects to receive next. If the ACK flag is set then the value of this field is the next sequence number that the receiver is expecting. This acknowledges receipt of all prior bytes (if any). The first ACK sent by each end acknowledges the other end's initial sequence number itself, but no data.

  Suppose Host A received one segment from Host B containing bytes 0 through 535 and another segment containing bytes 900 through 1000. In this case, A is still waiting for byte 536 and beyond. So A's next segment to B will contain 536 in the acknowledgment number field.

- HLEN specifies the size of the TCP header in 32-bit words. The minimum size header is 5 words and the maximum is 15 words thus giving the minimum size of 20 bytes and maximum of 60 bytes, allowing for up to 40 bytes of options in the header. This field gets its name from the fact that it is also the offset from the start of the TCP segment to the actual data.
- The reserve field is for future use and should be set to zero.

- The code field consists 8 bits in original TCP header. From the left-most to right, flags are:
  - CWR: Congestion Window Reduced field used from the TCP sender to inform the receiver that the sender has slowed down so that the receiver can stop sending the ECE-Echo.
  - ECE: Explicit Congestion Notification is used to signal congestion.
  - URG: Urgent pointer field is valid. It is seldom used.
  - ACK: Acknowledgment field is valid. All packets after the initial SYN packet sent by the client should have this flag set.

- PSH: This segment requests a push. The receiver is requested to deliver the data to the application upon arrival and not buffer it until a full buffer has been received.
- RST: Reset the connection. It is also used to reject an invalid segment or refuse an attempt to open a connection.
- SYN: Synchronize sequence numbers. Only the first packet sent from each end should have this flag set.
- FIN: Sender has reached the end of its byte stream.

Original TCP header contains a code field of 6 bits. More bits from the reserved field are used for additional flags by later RFC documents.

- The Window indicates the size of the receive window, which specifies the number of window size units (by default, bytes) (beyond the sequence number in the acknowledgment field) that the sender of this segment is currently willing to receive (see Flow control and Window size below).
- The checksum field is used for error-checking of the header and data.
- Urgent pointer: if the URG flag is set, then this field is an offset from the sequence number indicating the last urgent data byte.

- Options: The length of this field is determined by the HLEN field. Options have up to three fields: Option-Kind (1 byte), Option-Length (1 byte), Option-Data (variable). The Option-Kind field indicates the type of option, and is the only field that is not optional. Depending on what kind of option we are dealing with, the next two fields may be set: the Option-Length field indicates the total length of the option, and the Option-Data field contains the value of the option, if applicable. Some options may only be sent when SYN is set.
- Padding: The TCP header padding is used to ensure that the TCP header ends and data begins on a 32 bit boundary. The padding is composed of zeros

Connection establishment and termination

Connection establishment in TCP is done via a three-way handshake between the sending and receiving hosts.

To initial the handshake, the sending host A sends a SYN signal to the receiver host B indicating that it is ready to transmit and would like to synchronize the byte sequence number x. After B has received the SYN signal, B sends a SYN-ACK signal back to A with a sequence number, say y, and an ACK field equal to $x + 1$. The sequence number y in the SYN signal indicates the starting sequence number of backward traffic from B to A. When A receives this SYN signal from B, it sends an ACK signal to B indicating its expectation of sequence $y + 1$ onward. After that, a full-duplex communication is established.

When an endpoint wishes to stop its half of the connection, it transmits a FIN packet, which the other end acknowledges with an ACK. Therefore, a typical tear-down requires a pair of FIN and ACK segments from each TCP endpoint. After both FIN/ACK exchanges are concluded, the side which sent the first FIN before receiving one waits for a timeout before finally closing the connection, during which time the local port is unavailable for new connections; this prevents confusion due to delayed packets being delivered during subsequent connections. Usually the timeout is set for twice the time a packet might live on the Internet (120 seconds).

A connection can be "half-open", in which case one side has terminated its end, but the other has not. The side that has terminated can no longer send any data into the connection, but the other side can. The terminating side should continue reading the data until the other side terminates as well.

It is also possible to terminate the connection by a 3-way signals, when host A sends a FIN and host B replies with a FIN-ACK (merely combines 2 steps into one) and host A replies with an ACK. This is perhaps the most common method. It is possible for both hosts to send FINs simultaneously then both just have to ACK. This could possibly be considered a 2-way signals since the FIN/ACK sequence is done in parallel for both directions.

Flow control and window size advertising

TCP uses a sliding window flow control protocol. In each TCP segment, the receiver specifies in the receive window field the amount of additionally received data (in bytes) that it is willing to buffer for the connection. The sending host can send only up to that amount of data before it must wait for an acknowledgment and window update from the receiving host.

When a receiver advertises a window size of 0, the sender stops sending data and starts the persist timer. The persist timer is used to protect TCP from a deadlock situation that could arise if a subsequent window size update from the receiver is lost, and the sender cannot send more data until receiving a new window size update from the receiver.

When the persist timer expires, the TCP sender attempts recovery by sending a small packet so that the receiver responds by sending another acknowledgment containing the new window size.

If a receiver is processing incoming data in small increments, it may repeatedly advertise a small receive window. This is referred to as the silly window syndrome, since it is inefficient to send only a few bytes of data in a TCP segment, given the relatively large overhead of the TCP header.

TCP senders and receivers typically employ flow control logic to specifically avoid repeatedly sending small segments.

TCP time-out and retransmission

TCP guarantees a reliable delivery of data even in the case of transmission losses and delays. This is possible because each segment of data is retransmitted if an ACK is not received within a certain period of time. The time-out period is set as a function of the round-trip time (RTT). However, because of the variation of the RTTs between hosts on the Internet, we need some method to decide the RTT. One simple technique uses a recursive estimation of the RTT based on the current estimate and the sample RTT as below.

estimated RTT $= a\times$ estimated RTT $+b\times$ sample RTT,

where $a + b = 1$, and a and b are shosen to smooth the estimated RTT. If b is large, the changes in RTT are tracked. A large a is more stable but not quick enough to adapt to real changes in the RTT. When an estimate for the RTT is obtained, the time-out period is simply taken to be twice as long.

There are some algorithms that modify the foregoing simple algorithm for the computation of the RTT. Enhancing TCP to reliably handle loss, minimize errors, manage congestion and go fast in very high-speed environments are ongoing areas of research and standards development. As a result, there are a number of TCP congestion avoidance algorithm variations.

4.6 Domain name system (DNS)

The Domain Name System (RFC 1034, 1035) is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network.

It associates various information with domain names assigned to each of the participating entities.

It translates domain names meaningful for users to the numerical IP addresses needed for the purpose of locating computer services and devices worldwide.

By providing a worldwide, distributed keyword-based redirection service, the Domain Name System is an essential component of the functionality of the Internet.

The Domain Name System distributes the responsibility of assigning domain names and mapping those names to IP addresses by designating authoritative name servers for each domain.

Authoritative name servers are assigned to be responsible for their particular domains, and in turn can assign other authoritative name servers for their sub-domains.

DNS provides a few other services in addition to translating host names to IP addresses.

- Host aliasing: a host with a complicated host name can have one or more alias names. For example, a host name `wei-office-ccc.cs.lakeheadu.ca` may have alias such as `ccc.cs.lakeheadu.ca` and `www.ccc.cs.lakeheadu.ca`.

  The host name is called the canonical hostname. DNS can be invoked by an application to obtain the canonical hostname for a supplied alias host name as well as the IP address of the host.

- Mail server aliasing: DNS can be invoked by a mail application to obtain the canonical hostname for a supplied alias hostname as well as IP addresses of the host. In fact, the MX record (we will discuss the record type later) permits an institute's server and web server to have identical aliased hostname.

- Load distribution: DNS is also used to perform load distribution among replicated servers, such as replicated web servers. Some site, such as `cnn.com`, are replicated over multiple servers, with each server running on a different end system and each having a different IP address.

  For replicated web servers, a set of IP addresses is associated with one canonical hostname. The DNS database contains this set of IP addresses. When client make a DNS query for a name mapped to a set of addresses, the server responds with the entire set of IP addresses, but rotates the ordering of the addresses within each reply.

  Because a client usually uses the first IP address in the set, DNS rotation distributes the traffic among the replicated servers. Similar method is used for multiple mail servers, content distribution companies.

Mapping domain names to IP addresses

The DNS implementation is just like a tree in which each node represents one possible label. The right-most label corresponds to the node closes to the root, whereas the left-most label corresponds to the host name and it is farthest node from the root.

Roughly to say, there are three classes of DNS servers:

- Root DNS servers: In the Internet, there are 13 root DNS servers (labeled A through M), most of which are located in North America. Each of the 13 root servers actually is a network of replicated server, for both security and reliability purpose. There are 247 root servers as of fall 2011.
- Top-level domain (TLD) servers: There servers are responsible for top-level domains such as com, org, edu, ca, jp, cn etc.
- Authoritative DNS server: Every organization with publicly accessible hosts on the Internet must provide publicly accessible DNS records that map the name of those hosts to IP addresses. An organization's authoritative DNS server houses theses DNS records.

There is another important type of DNS server called the local DNS server. Each ISP has a local DNS server (or several servers). When a host connects to an ISP, the ISP provides the host with the IP address of one or more of its local DNS servers.

The organization domain consists of labels describing the types of client organization. Some examples of labels are:

.com	Commercial organization
.edu	Educational institution
.gov	Government organization
.int	International organization
.mil	Military
.net	Network-related support center
.org	Other organization

It is noticed that the number of levels in the hierarchy is not limited to three. The country name is used as suffix after the organization type. For examples, .us for United States, .fr for France, .cn for China, etc.

The DNS servers store resource records (RRs), that provide hostname-to-IP address mappings. Each DNS reply message carries one or more RRs.

A resource record is a four-tuple that contains the fields: `(Name, Value, Type, TTL)`.

The meaning of `Name` and `Value` depends on `Type`:

- If `Type=A`, then `Name` is a hostname and `Value` is the IP address for the hostname.
- If `Type=NS`, the `Name` is a domain and `Value` is the hostname of an authoritative DNS server that knows how to obtain the IP addresses for hosts in the domain. This record is used to route DNS queries further along in the query chain.
- If `Type=CNAME`, then `Value` is a canonical hostname for the alias hostname `Name`. This record can provide querying hosts the canonical name for an alias hostname.

- If `Type=MX`, then `Value` is the canonical name of a mail server that has an alias hostname `Name`. MX records allow the hostnames of mail servers to have simple aliase. By using the MX record, a company can have the same aliased name for its mail server and for one of its other servers. To obtain the canonical name for the mail server, a DNS client would query for an MX record; to obtain the canonical name for the other server, the DNS client would query for the CNAME record.

Name servers

The Domain Name System is maintained by a distributed database system, which uses the client-server model. The nodes of this database are the name servers.

Each domain has at least one authoritative DNS server that publishes information about that domain and the name servers of any domains subordinate to it.

When domain names are registered with a domain name registrar, their installation at the domain registry of a top level domain requires the assignment of a primary name server and at least one secondary name server.

The requirement of multiple name servers aims to make the domain still functional even if one name server becomes inaccessible or inoperable.

The designation of a primary name server is solely determined by the priority given to the domain name registrar.

Only the fully qualified domain name of the name server is required, unless the servers are contained in the registered domain, in which case the corresponding IP address is needed as well.

Primary name servers are often master name servers, while secondary name server may be implemented as slave servers.

To insert records into the DNS database, the domain names need to register at a registrar.

A registrar is a commercial entity that verifies the uniqueness of the domain name, enters the domain mane into the DNS database, and collects a small fee for its services. A list and some other information about the registrars can be found at
`www.internic.net`

DNS resolvers

The client-side of the DNS is called a DNS resolver. It is responsible for initiating and sequencing the queries that ultimately lead to a full resolution (translation) of the resource sought, e.g., translation of a domain name into an IP address.

A DNS query may be either a non-recursive query or a recursive query:

- A non-recursive query is one in which the DNS server provides a record for a domain for which it is authoritative itself, or it provides a partial result without querying other servers.
- A recursive query is one for which the DNS server will fully answer the query (or give an error) by querying other name servers as needed. DNS servers are not required to support recursive queries.

The resolver, or another DNS server acting recursively on behalf of the resolver, negotiates use of recursive service using bits in the query headers.

Resolving usually entails iterating through several name servers to find the needed information. However, some resolvers function more simply by communicating only with a single name server. These simple resolvers (called "stub resolvers") rely on a recursive name server to perform the work of finding information for them.

A reverse lookup is a query of the DNS for domain names when the IP address is known. Multiple domain names may be associated with an IP address.

Users generally do not communicate directly with a DNS resolver. Instead DNS resolution takes place transparently in applications such as web browsers, e-mail clients, and other Internet applications. When an application makes a request that requires a domain name lookup, such programs send a resolution request to the DNS resolver in the local operating system, which in turn handles the communications required.

DNS primarily uses User Datagram Protocol (UDP) on port number 53 to serve requests. DNS queries consist of a single UDP request from the client followed by a single UDP reply from the server. The Transmission Control Protocol (TCP) is used when the response data size exceeds 512 bytes, or for tasks such as zone transfers.

Some resolver implementations use TCP for all queries.