Network Layer

- This note is based on Chapters 4 and 5 of the textbook. You are recommended to read the textbook for more details.

- **Outline of this note**
  - Overview of network layer
  - Data plane (Chapter 4)
    - Router architecture and functions
    - Internet Protocol (IP)
    - Generalized forwarding and software defined networks (SDN)
    - Broadcast, multicast and anycast
  - Control plane (Chapter 5)
    - Routing algorithms (how to create routing tables automatically at routers)
    - Routing protocols, (creating routing tables in TCP/IP internet)
    - SDN control plane
    - Internet Control Message Protocol (ICMP) and Simple Network management Protocol (SNMP)

- **Overview of Network Layer**
  The Internet is a collection of networks connected by routers. In a TCP/IP internet, special computers called IP routers or IP gateways provide the interconnections among physical networks. Routers operate at network layer. Internet Protocol (IP) transfers datagrams over networks. At a source, IP encapsulates data (e.g., segments from transport layer) into datagrams and transfers datagrams to the next hop router or destination. A datagram has an IP header and a data portion. The IP header contains a source IP address and a destination IP address. At a destination, IP receives datagrams and forwards the data in the received datagram to appropriate receivers (e.g., forwards segments to transport layer protocols). A router receives datagrams from an input link, checks IP destination address in the header of the received datagrams and transfers them to an output link connected to a next hop router or destination.

- **Two Key Network-Layer Parts and Functions**
  The network layer can be partitioned into two parts, a data plane and a control plane (see Figures 1 and 2). There are two key functions at network layer. One is **forwarding**, realized at data plane, moving datagrams from a (router’s) input to an output based on a pre-defined route (path) from source to destination. Forwarding is a process to get datagrams passing through a single router. IP is the protocol for forwarding in TCP/IP internet. The other key function is **routing**, realized at control plane, deciding a route for datagrams from source to destination. Routing is a process to compute a route from source to destination. In a TCP/IP internet, the information of a route to reach a destination is kept in routing tables at routers. A router uses its routing table and the destination address in the header of a datagram to find
the route to reach the destination. The computation of routes for forwarding is realized by computing routing tables. Routing tables can be computed by routing algorithms at routers (traditional approach, see Figure 1) or by a remote controller (software defined network (SDN) approach, see Figure 2).

- **Network Layer Service**

To meet the application requirements, there are many network layer service models. Typical models include: guaranteed delivery, in-order delivery and best-effort delivery. The guaranteed delivery assures the delivery of data packets to the destination. The in-order delivery guarantees data packets arriving at the destination in the order they are sent. The best-effort tries its best to deliver data packets to destination in a correct order but does not guarantee either of them.

Based on how datagrams are transferred, a TCP/IP internet provides two types of data delivery services, one is virtual circuit (VC) connection service and the other is connectionless data delivery service.

In the VC connection service, there are three phases: (1) A source-to-destination route (which is a VC) is created before data transfer. In the route creation, link and router resources (bandwidth, buffers etc) are allocated to the route to guarantee the QoS. (2) Datagrams are transferred on the route. Each datagram is attached an additional header which carries a route label. A router forwards datagrams based on the route label. Datagrams with a same route label are transferred on a same route. (3) After the communication session is over, the route is terminated and the resources assigned to the route are released. The VC connection service is commonly used in the Internet backbones for a small number of connections with a large traffic in each connection.

In the connection-less data delivery service, there is no route set-up at network layer and a
router forwards each datagram independently by the destination IP address in the header of the datagram. The connection-less data delivery service is commonly used in the Internet edges and intranets.

- **Data Plane**
  Data packet forwarding is performed at data plane of network layer. The forwarding is realized by IP running on routers, devices connecting networks together to form an internet.

- **Router Architecture and Functions**
  A typical router performs two key functions: one is use IP to forward datagrams from inputs to outputs based on the routing table and destination addresses of datagrams, and the other is to use routing protocols (RIP, OSPF, BGP) to create a router usually consists of multiple input ports, multiple output ports, a switch to connect the inputs and outputs, and a control unit.

  At the physical layer, an input port handles the line termination, bit-level reception, etc. At the link layer, an input port deals with data frame receiving. At the network layer, a received datagram is queued at input port and IP checks the destination address of the received datagram and routing table look-up to finds the output port for the datagram. As the number of IP addresses is huge, each entry of a routing table usually contains a range of destination addresses rather than a single destination address (see Figure 3).

  At an output port, datagrams are put in a queue waiting for transmission. When the output link is available, datagrams are transmitted to the link in the order they are put in the queue.

  A switch in a router transfers datagrams from an input to an output. The performance of a switch is measured by the switching rate, the number of datagrams transferred from inputs to outputs. There are three types of switches.
The first type is to transfer datagrams via memory. When a conventional computer is used as a router, a datagram from an input is first kept in the main memory. Next IP performs routing table look up and decides the output for the datagram. Finally, IP reads the datagram from the memory and sends it to the output. Many routers used in the Internet edges and local networks use this type of switches. Switching rate depends on the memory access speed.

The second type of switch is to connect every input and every output to a data bus (a shared medium). At a specific time, one input-output pair uses the bus for datagram transfer. The switching rate depends on the bandwidth of the data bus. An example of routers using this type of switch is Cisco 5600 which has a 32 Gbps data bus.

The third type of switch uses a circuit-switched network to connect an input and an output by a dedicated physical path. This type of switch has very high switching rate and usually used in core routers.

- **Datagram Forwarding Table**

  IP uses a table driving forwarding. IP employs a forwarding (routing) table on each end system. The routing table keeps the information on destinations and how to reach them, one entry of the table for one group of destinations. To make the forwarding efficient and the routing table small, IP usually considers a network as the destination for all end systems connected to the network and keep the network prefix (the IP address for the network, which defines a block of continuous IP addresses used for the end systems in the network) in one entry of the routing table. Also, the routing table usually does not keep the info of an entire route to reach a destination but only contains the info of the next hop router and the network interface (output port) to reach the destination. Below is an example of a conceptual routing table.

<table>
<thead>
<tr>
<th>Destination Address Range</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00000*** ******</td>
<td>0</td>
</tr>
<tr>
<td>000 00000000</td>
<td></td>
</tr>
</tbody>
</table>
There are four entries in the table. Based on this routing table,
a received datagram with a destination address between 11001000 00010111 00000000 00000000 and 11001000 00010111 00000111 11111111 will be forwarded to interface 0;
a received datagram with a destination address between 11001000 00010111 00010000 00000000 and 11001000 00010111 00010111 11111111 will be forwarded to interface 1;
a received datagram with a destination address between 11001000 00010111 00011000 00000000 and 11001000 00010111 00011111 11111111 will be forwarded to interface 2;
and a received datagram with a destination address not in any of the ranges above will be forwarded to interface 3.

- **Routing Table Look Up**

The header of an IP datagram contains a destination address. When a router $R$ receives a datagram, $R$ checks the destination address and looks up the routing table to find the entry which matches the destination address. To save space, a routing table usually does not include a distinct entry for every destination address in an internet but uses an entry for a block of addresses defined by a sub-string of an IP address (usually called a network address). We say an entry in a routing table matches a destination address in a datagram if the network address of the entry matches a prefix of the destination address (the destination address is included in the block of addresses defined by the network address in the entry). For example, a router has the routing table above. The destination address 11001000 00010111 00010*** ***week* of a datagram is matched by the entry 11001000 00010111 00010*** ***week*. The destination address 11001000 00010111 00011*** ***week* of a datagram is matched by the entry 11001000 00010111 00011*** ***week*.

In the case that multiple entries in a routing table match a prefix of the destination address of a datagram, the datagram should be forwarded to the network defined by the entry which matches the **longest prefix** of the destination address of the datagram. For example, a router has the following routing table:

<table>
<thead>
<tr>
<th>Destination Address Range</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00000*** **<em>week</em></td>
<td>0</td>
</tr>
</tbody>
</table>
Assume a datagram $D$ has destination address (DA) $11001000\ 00010111\ 00010010\ 10100001$. Then this DA is matched by two entries $11001000\ 00010111\ 0001001\ 00010***\ ********$ and $11001000\ 00010111\ 0001001\ 0001001*\ ********$. Because entry $11001000\ 00010111\ 0001001\ 0001001*\ ********$ matches the longest prefix of the DA, datagram should be forwarded to the network specified by $11001000\ 00010111\ 0001001*\ ********$.

The routing table look up for a received datagram is to find the entry which matches the longest prefix of the destination address of the datagram.

If no entry matches the destination address, $R$ may discard the datagram and forward it to a default route.

- **Network (IP) Layer Protocols**

  Major protocols of TCP/IP work for network layer include IP (Internet Protocol), ICMP (Internet Control Message Protocol), RIP (Routing Information Protocol), OSPF (Open Shortest Path First) and BGP (Border Gateway Protocol) (see Figure 4). IP is the foundation of the internet and provides the best-effort and connectionless data packet delivery service. IP defines the rules for data delivery (internet addressing scheme, destination address based forwarding by default etc.), datagram format, and handles datagram forwarding. ICMP is used to allow routers and end systems to report errors in communication and conditions of networks. RIP, OSPF and BGP are protocols to create and update routing tables. Currently, there are two versions of IP in using, IPv4 and IPv6.
IPv4 Datagram Format

The basic data unit transferred in the Internet is called *IP datagram*. A datagram consists of a header and a data area. In IPv4, the datagram header has a size of multiple of 4 bytes (at least 20 bytes) and the data field has a size of multiple of bytes. The datagram size is at most \(2^{16} - 1\) bytes.

The fields in the IPv4 datagram header (see Figure 5):

- Version, 4 for IPv4
- Header length, in 32-bit words.
- Type of service, give hints to forwarding algorithm for selecting forwarding paths. An internet does not guarantees any particular type of service.
- Total length, the size of the whole datagram (header and data) in bytes.
- Identification, a unique id for the datagram, that is used for fragmentation.
- Flags, \(df\) bit and \(mf\) bit, used for fragmentation
- Fragment offset, specify the offset in the original datagram, in 8 bytes, used for fragmentation.
- Time to live, specify the number of hops a datagram can live, default 64.
- Protocol, specify the protocol which creates the data in data area. Examples include 1 for ICMP, 2 for IGMP, 6 for TCP, 17 for UDP. The values in the examples are decimals. RFC1700 gives a list of the values used in this field.
- Checksum, used in error check for the header.
- Source address, 32 bits.
- Destination address, 32 bits.
- Options and padding
- Data

IPv4 Addressing

IPv4 provides a universal communication service which allows any host to communicate with any other host in the same system. To achieve this, a global identifier for each host (network interface) is needed. An end system usually has only one network interface and a router has multiple network interfaces. IPv4 uses 32-bits global internet addresses to identify network interfaces. Each network interface is assigned a unique 32-bit internet address. The address is expressed in dotted decimal notation. For example, the IP address 11000000 10101000 00000000 01100100 is expressed by 192.168.0.100.

There are \(2^{32}\) different IPv4 addresses. Most of them are used for identifying network interfaces, known as unicast (one-to-one) addresses. A unicast address is partitioned into two parts: **netid** part (most significant bits part) which identifies a network (defines a block of IP addresses assigned to the network) and **hostid** part (least significant bits part) which identifies a host in the network. For example, assume that an IP address w.x.y.z has netid w.x.y and hostid z. Then the network is assigned a block of 256 addresses from w.x.y.0 to w.x.y.255.
CIDR, Classless InterDomain Routing

In CIDR, netid can have an arbitrary number of bits but the size of a block addresses for a network must be a power of two. CIDR uses a bit mask to identify the size of the block. The most significant $m$ bits of 1 in the mask means that the block has size of $2^{32-m}$. A short notation $w.x.y.z/m$, where $m$ is the length of netid, known as CIDR notation or slash notation, is used for expressing an IP address. For example, $128.211.168.0/21$ denotes the IP address $128.211.168.0$ with a 21 bits netid (the block for the network has $2^{32-21} = 2048$ addresses).

Special IPv4 Addresses

- All 0 and all 1 addresses
  
  An address with a valid netid and all 0 in hostid is not used for any specific host in the network specified by the netid but is reserved to refer to the network. It is not a valid source address.

  An address with a valid netid and all 1 in hostid is the broadcast address (for all hosts) in the network specified by the hostid. The address is called directed broadcast address. It is not a valid source address. Directed broadcast provide a powerful mechanism which allows a host to send message to all hosts in a specific network. This mechanism is considered somehow dangerous and many sites configure routers to reject directed broadcast packets from outside.

  An address of 32 1’s is called limited broadcast address which provides a broadcast address for the local network independent of the assigned IP address of the network. It is not a valid source address. A host can broadcast within the same network to which it is directly connected by this address when the host does not know the netid of the network (e.g., at start-up stage). Once the netid is known, directed broadcast is preferred.
An address of 32 0’s is used for the start-up of a host. When a host wants to communicate over a network but does not know its IP address, it can send a packet to the limited broadcast address and uses the all 0 address to identify itself. It is not a valid destination address.

- Loopback address
  127.x.y.z (usually 127.0.0.1) is reserved for loopback.

- Address for private networks
  The IETF reserved a set of netids as private network addresses. The reserved netids will not be assigned to networks in the global Internet. They are used in private networks. The reserved network addresses include 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, and 169.254.0.0/16. By definition, no one owns these addresses and they can be used by anyone. Thus, these addresses are not used on the Internet. Together with the network address translation (NAT) technology, the private network addresses can be used to extend the IP address space within an institution: using a private network address for an internal network and connect the internal network to the Internet via a proxy server which uses the NAT to mapping the private IP addresses to the public IP addresses.

- Address for multicast, 224.0.0.0~239.255.255.255.

- Address reserved, 240.0.0.0~255.255.255.255.

**Get IPv4 Address for Host**

Usually, the IP address of a computer is manually assigned and kept in its secondary storage where the OS finds it at start up. However, some computer may need to find its IP address from a remote file server. In TCP/IP with IPv4, a number of protocols including RARP, BOOTP and DHCP (Dynamic Host Configuration Protocol) can be used for this purpose.

DHCP is a client-server application. A host (client) sends a DHCP request to server. Server responds to client with DHCP reply which contains an IP address assigned to the client and other info. DHCP allows automatic and dynamic address assignment. In dynamic address assignment, server leases IP address to client for a limited time. Figure 6 explains how to get an IP address via DHCP.

**Get IPv4 Address for Network**

An ISP gets a block of IP addresses from ICANN and partitions the block of addresses into sub-blocks. An organization gets a sub-block for a network from the ISP. An example, ISP partitions a block into eight sub-blocks of equal size:

ISP’s block 11001000 00010111 00010000 00000000 200.23.16.0/20
Sub-block 0 11001000 00010111 00010000 00000000 200.23.16.0/23
Sub-block 1 11001000 00010111 00010010 00000000 200.23.18.0/23
Sub-block 2 11001000 00010111 00010100 00000000 200.23.20.0/23
........
Sub-block 7 11001000 00010111 00011110 00000000 200.23.30.0/23
• **Network Address Translation (NAT)**

NAT is a technique to connect an intranet with private network addresses with the global internet. An organization uses private network address for its intranet and a gateway to connect the intranet and the global internet. The gateway performs private address and global address translation. Usually, the number of global addresses available to the organization is smaller than the number of private addresses. In this case, port numbers are used to identify hosts for global communication.

• **Why NAT**

NAT is a powerful approach to extend IPv4 address space and allows an organization to use a single IP address (not a block of global addresses) to connect a network to the Internet. NAT provides easier management as an organization can change addresses of its intranet without notifying its ISP and can change ISP without changing addresses of the intranet. NAT provides better security, hosts in the intranet are invisible to outside.

Problems with NAT include: The gateway may become a bottleneck for the connection between the intranet and Internet. The gateway has to use protocols at layers higher than 3 to perform address translation. NAT is a major factor to save IPv4, on the other hand a major factor to postpone IPv6.

• **IPv6**

A major motivation to change IPv4 is that IPv4 addresses will be exhausted soon. Other motivations include: make processing/forwarding faster by a new header format, header changes to facilitate QoS. The new version of IP to replace IPv4 is IPv6. Features of IPv6 are basically the same as IPv4, but IPv6 has many new features:

- Larger address space, IPv6 uses 128 bits for address (IPv4 uses 32 bits).
- Extended address hierarchy, IPv6 supports additional levels of addressing hierarchy and can define a hierarchy of ISPs and a hierarchy within a given site.

- Flexible header format, IPv6 uses new datagram format and defines a set of optional headers.

- Improved options, IPv6 includes new options that provide additional facilities not available in IPv4.

- Provision for protocol extension, IPv6 does not specify every detail of the protocol and can permit additional features.

- Support for autoconfiguration and renumbering, IPv6 provides facilities that allow computers on an isolated network to assign themselves addresses and start communication without depending on a router or manual configuration, and permit a manager to renumber networks dynamically.

- Support for resource allocation, IPv6 has flow abstraction which allows preallocation of network resources and differentiated service specification for differentiated services.

- **IPv6 Datagram**

IPv6 has a fixed-size base header, followed by zero or more extension headers, followed by data (see Figure 7).

An IPv6 base header has the following fields.

- Version (4 bits), specifies the version of the protocol (6 for IPv6).

- Traffic class (8 bits), specifies service type (equivalent to service class in IPv4).

- Flow label (20 bits), specifies a flow abstraction that routers use to associate a datagram with a specific flow and priority.

- Payload length (16 bits), specifies the number of bytes in the datagram excluding the header itself.

- Next header (8 bits), specifies if the header is followed by an extension header or not.

- Hop limit (8 bits), specifies the maximum number of hops the datagram can travel in the Internet.

- Source address (128 bit)

- Destination address (128 bit)

IPv6 extension headers are similar to IPv4 options. Each datagram includes extension headers for only those facilities the datagram uses. Routers use the next header field to parse the datagram and extract all header information.

- **IPv6 Address**

IPv6 uses 128 bits for an IP address (the address space is $2^{128}$, more than $10^{24}$ addresses per square meter of the earth’s surface). To represent an IP address, IPv6 uses colon hexadecimal notation (colon hex for short) in which the value of each 16 bits is represented in hexadecimal separated by colons. The colon hex for the address
IPv6 Base Header

<table>
<thead>
<tr>
<th>Version</th>
<th>traffic class</th>
<th>flow label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Payload length</th>
<th>next header</th>
<th>hop limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination address</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7: IPv6 datagram format.

0100 1000 1110 0110 1000 1011 0100 0100
1111 1111 1111 1111 1111 1111 1111 1111
0000 0000 0000 0000 0001 0001 1000 0000
1001 0110 0000 1010 1111 1111 1111 1111

is represented as


The colon hex notation allows zero compression. For example, the address FF05:0:0:0:0:0:0:B3 can be rewritten as FF05::B3. It incorporates dotted decimal suffixes, 0:0:0:0:0:128.10.2.1 (or ::128.10.2.1) is a valid colon hex notation. IPv6 extends CIDR-like (classless inter-domain routing) notation by allowing an address to be followed by a slash and an integer that specifies a number of bits. For example, 12AB::CD20/60 specifies the first 60 bits of the address.

- Transition From IPv4 to IPv6
  A number of technologies have been developed for transiting IPv4 to IPv6. Major technologies include:
  - Dual-stack, this technique allows IPv4 and IPv6 co-exist in a system/device. Both IPv4 and IPv6 are bundled with new OS releases. When IPv4 addresses are involved, a system/device uses IPv4, otherwise uses IPv6.
  - Tunnel, this technique allows IPv6 datagrams go through IPv4 only routers. The idea is to encapsulate an IPv6 datagram in an IPv4 datagram.
  - Translation, this technique translates the headers between IPv4 and IPv6 datagrams.

- Generalized Forwarding and Software Defined Networks (SDN)
  In a simple scenario of IP forwarding, each entry of a routing table is identified by a network IP address and the entry specifies the action for the destination IP address which matches
the entry. The action usually is to forward the datagram to an output link or drop the datagram. When a datagram is received, IP looks up the routing table to find an entry to which destination addresses of the datagram matches and follows the action specified by the entry. This is known as simple match-plus-forwarding.

At network layer, more complex actions than forwarding and drop can be performed as we have discussed in the network address translation and so on. In a more complex case, each entry of a (routing) table may be identified by information in the headers of multiple layers (e.g., addresses and other parameters in the TCP, IP and data link layer protocols headers). Actions specified in an entry of the routing table may include forward, drop, modify and analyze the datagram. This is known as generalized match-plus-action. A match-plus-action table is usually computed by a remote controller which is a key component in SDN.

• **Broadcast and Multicast**

A communication application can be classified into unicast, broadcast, multicast and anycast based on how many end systems are involved in a communication session. In a unicast, a source sends a message to a single destination. In a broadcast, a source sends a message to all hosts within a scope. In a multicast, a source sends a message to a subset of hosts within a scope. A multicast can be considered a generalization of unicast and broadcast: when the subset contains only one host, it becomes unicast; and when the subset contains all hosts, it becomes broadcast.

A broadcast or multicast can be realized by multiple unicasts, sending multiple copies of the message, one copy for each destination. However, this is not an efficient approach, redundant traffic and difficulty in finding address for each receiver. Many physical networks (e.g., Ethernet) has functions to support efficient broadcast/multicast. For efficient broadcast/multicast within a same physical network, the broadcast/multicast functions supported by the network are used. For efficient broadcast/multicast over an internet (multiple networks), key issues include avoid duplication and routing loop. A general approach is to create a broadcast/multicast tree before sending the message.

Anycast is a communication pattern that a source sends a message to a set of receivers in a scope and the message is delivered to one (any) member of the receivers.

• **Control Plane**

Routers use routing (or generalized match-plus-action) tables and addresses of data packets to decide a route for forwarding (action for handling) received datagrams. Routing tables can be created by routing algorithms at routers. This is known as per-route control or traditional approach. Routing tables can also be created by remote controllers. This is known as logically centralized control or software defined network (SDN) approach.

• **Routing Algorithms**

Routers in an internet forward datagrams based on routing tables. Routing algorithms create and update routing tables in routers (see Figure 1). An internet is usually expressed by a weighted graph $G(V, E, W)$, $V$ set of routers and networks, $E$ set of links (a link is a connection between a router and a network or a connection between two routers), $W$ function assigns a cost to each link. Routing algorithms can be classified into global information algorithms and
local information based algorithms. A global info algorithm at a router computes the routing table based on the entire graph. Link-state algorithm is an example. A local info algorithm at a router computes the routing table based on the information of links adjacent to the router. Distance-vector algorithm is an example.

- **Link State Routing Algorithm**
  
  An internet is considered as a graph $G$ and link state routing is shortest path first routing. A link state routing algorithm requires each router to have the entire graph $G$. Two routers are neighbours if they are connected by a same network. Each router tests the state of all links incident to it and periodically propagates the link states to all other routers. Each router uses the link states to get graph $G$ and uses a shortest path algorithm (e.g., Dijkstra’s algorithm) to compute the routes from the router to all destinations. The routes info are kept in the routing table.

- **Distance Vector Routing Algorithm**
  
  Distance vector routing algorithm is based on the Bellman-Ford shortest path algorithm. Each router has a routing table and each entry of the routing table has a destination, a next hop router and a distance (hops) to the destination. Initially, the routing table has an entry for each directly connected network. Periodically, each router sends a copy of its routing table to every neighbour routers. On receiving a report from a neighbour, a router updates its own routing table if one of 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.
  - The route in an entry to a destination with this neighbour as the next hop router and the distance from the neighbour to the destination has changed.

- **Hierarchical Routing**
  
  It is not practical to consider the Internet as a single graph because it is too large. Also, an organization may want to control the routing in its own networks. 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 relays the information from the border routers of other autonomous systems to the IGP which propagates the information to the interior routers in the autonomous system.
• Internet Routing Protocols

There are two interior gateway protocols used in the Internet. One is Routing Information Protocol (RIP) based on the distance vector routing. The other is Open Shortest Path First Protocol (OSPF) based in the link state routing. There is only one exterior gateway protocol, Border Gateway Protocol (BGP) which uses the path vector routing.

• RIP

RIP is a simple IGP and often used for small and stable autonomous systems. RIP is a distance-vector routing protocol and uses hop count metric to measure distances. It partitions the machines in an autonomous system into active ones (routers) and passive ones (hosts). Routers advertise their routing tables to neighbours periodically; hosts 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 its routing table to all directly connected networks periodically (in every 30 seconds).

3. On receiving a routing table advertised from another router, a router updates its own routing table if one of the followings happens:
   the received routing table contains a shorter route for some entry in its own table;
   the received routing table contains a route to a destination which does not have an entry in its own table; and
   the route in some entry pass through this neighbour router and the received routing table shows the distance from the neighbour to the destination has changed.

RIP is used for small autonomous system and restricts the maximum cost of a route to 15. RIP is an application level process called routed (daemon) and uses UDP for message exchange.

• OSPF

OSPF is an open standard link-state routing protocol. It is usually used in large autonomous systems with more complex topology and less stability.

In link-state routing algorithm, a router tests the states of the links connected to it and propagates (by flooding) the link states 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 states 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. 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. OSPF uses IP to carry data.
• **OSPF Advanced Features**

OSPF has the following features that RIP does not have:

- Security, OSPF messages authenticated, prevent malicious intrusion
- Multiple same-cost paths to a destination allowed (only one path in RIP)
- Different costs for a link allowed for different QoS
- Integrated unicast and multicast support
- Hierarchical OSPF in large AS. A large AS is partitioned into a core area and multiple other areas, each other area is connected to the core by area border routers. Each router advertises link states to all routers within the same area. Area border router sends the graph of one area to another area it connects.

• **Border Gateway Protocol (BGP)**

BGP is used in the Internet for passing routing info between autonomous systems. The exchange of information between autonomous systems are handled by the border routers or peers of the systems. Major tasks of BGP include:

- eBGP, obtain destination reachability info from neighboring ASs.
- iBGP, propagate reachability info to routers within the AS.
- Decide paths to destinations based on reachability info and policy.

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 router to reach the next autonomous system, and the path (sequence of autonomous systems) 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. BGP uses TCP (port 179) for communication with peers.

• **Software Defined Network (SDN) Control Plane**

Main characteristics of an SDN architecture includes the following: (1) A flow-based forwarding (match-plus-action) paradigm is used by routers to handle the received data packets. (2) The data plane and control plane are clearly separated. The data plane consists of switches which perform as defined by the match-plus-action rules. The control plane consists of remote controllers (servers and software) which create the match-plus-action rules. (3) There are network control functions for remote controllers to get the information of networks and send control information to switches. (4) The network is programmable through the control plane. Figure 2 gives a conceptual explanation for an SDN architecture.

An SDN control plane consists of controllers and network control applications. The protocols used by a controllers can be partitioned into three layers: (1) A communication layer supports the communication between a controller and network devices at data plane. OpenFlow Protocol is a commonly used protocol in this layer. (2) A network-wide state-management layer where the information for making control rules is kept. (3) An interface to network-control
application layer allows a controller to interact with network-control applications where the control rules are made.

- **Internet Control Message Protocol (ICMP)**

ICMP is used by host/router to transfer control/error messages. ICMP is included in IP implementation and provides communication between different computers at layer 3. ICMP only reports the error/problems but does not correct the error. ICMP is a user of IP. Its message has two levels of encapsulation (ICMP header and IP header) for transmission.

An ICMP message has the following fields. Type: specify the message type. Code: give further information on message type. Checksum: for error check of the entire message. Information: has variable length and structure depending on message type. ICMP messages include: Echo/reply messages (type=8/0) (used by ping, traceroute); destination unreachable messages (type=3, code=0-12); source quench message (type=0); time exceeded message (type=11, code=1/2); and others.

- **Network Management and Simple Network Management Protocol (SNMP)**

Network management includes several operations to maintain a network to meet certain service requirements (please refer to the textbook for a more formal definition). Key components in a network management framework include: (1) A managing server which is an application running on a centre device. (2) Managed devices, network equipments in a managed network. Each device may have multiple managed objects. (3) Managed Information Base (MIB) which keeps the information associated with each managed object. (4) Management agent which is a process on a managed device communicating with a managing server. (5) Network management protocol which is used to realize communications between managing servers and managed devices. Notice that a network management protocol itself does not manage the network but provides functions that network managers can use to manage networks. Simple Network Management Protocol (SNMP) is the standard network management protocol in TCP/IP internets. The most recent version for SNMP is SNMPv3.