Network Layer of Router

- The network layer of a router consists of two parts: control plane and data plane.

- Control plane of router has routing protocol that computes routing (or forwarding) table.

- Data plane of router uses routing table to route packets through a switch from an input interface to output interface.
Need to discuss:
- input
- output
- switch
- routing table
- **Link Layer:**
  - verify checksum in link header of frame
  - decapsulate link header from frame: pkt

- **Network Layer:**
  - verify checksum in IP header of pkt

- **Route + Queue:**
  - lookup routing table to determine appropriate output interface for pkt
  - add output interface to pkt
  - queue pkt after RED (Random Early Detection) processing. See next
RED (Random Early Detection) Queues

- A REQ is specified using 3 parameters:
  - a probability $P$
  - min Bytes
  - max Bytes

- Admit pkt (into queue), if current queue length is $\leq \text{min}$

- Admit pkt (into queue), with probability $P$, if current queue length is in interval $[\text{min}, \text{max}]$

- Drop pkt (from queue), if current queue length is $> \text{max}$
Output

Switch → Queue → Network Layer → Link Layer → Frame

- **Queue**: queue after RED processing of pkt

- **Network Layer**: modify TTL, checksum, ... in IP header of pkt

- **Link Layer**: encapsulate pkt in a newlink header forming a frame
Switch

- let "int" denote "an output interface"
- a switch can be designed as: memory, bus, or a crossbar

**Memory:**

```
frame → input → pkt + int → M0 → M1 → M2 → output
```

**Bus:**

```
frame → input → pkt + int → output
```
crossbar:

frame → input → pkt + int → go horizontally or go vertically → output
IP Addresses

- interfaces:

  - each interface has an IP address that consists of 32 Bits and can be represented by 4 integers (each is bet. 0..255) separated by dots: a.b.c.d

  - a block of consecutive IP addresses can be represented by (a.b.c.d)/x, where (a.b.c.d) is an IP address and x is an integer bet. 0 and 32 called a subnet mask

  - a block (a.b.c.d)/x has an IP address (a'.b'.c'.d') iff x left-most bits in (a.b.c.d) equal x left-most bits in (a'.b'.c'.d')
**Routing Table**

- **network example:**

  ![Diagram of network example]

  - Router
  - 223.1.1.255
  - 223.1.1.129
  - add block of Net 1 = 223.1.1.128/25
  - 223.1.1.128
  - 223.1.1.0
  - add block of Network 2 = 223.1.1.0/25
  - Rest of Internet

- **routing table of router:**

<table>
<thead>
<tr>
<th>if dst of pkt is in block</th>
<th>then forward pkt to interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>223.1.1.0/25</td>
<td>1</td>
</tr>
<tr>
<td>223.1.1.128/25</td>
<td>0</td>
</tr>
<tr>
<td>other</td>
<td>2</td>
</tr>
</tbody>
</table>

- if dst of pkt is in two or more blocks in a routing table, choose block with the largest subnetwork mask
Allocating Fixed IP Addresses

- ICANN (Internet Corp. for Assigned Names and Numbers) assigns a block of IP addresses to an ISP which divides the block into smaller blocks and assigns them to its client nets.

**Example:**

```
ISP
  └── net7
      └── net8
      └── net1
       └── ...
```

- (200.23.16.0/20)
- (200.23.16.0/23)
- (200.23.28.0/23)
- (200.23.30.0/23)
Allocating Temporary IP Addresses

• When a client host ch becomes in a new network and ch is not assigned any IP address that belongs to this network, then the DHCP-client c in ch communicates with some DHCP-server s in the network.

• The result of this communication is for c to obtain for 1 hour an IP address, that belongs to the network, from s.

• DHCP stands for Dynamic Host Configuration Protocol.
DHCP has four messages:

C → S: Discover (DHCP)
C → S: Offer (a temp. IP address)
C → S: Request (an offered IP address)
C ← S: Ack (agree to the request)

The reason for the request and ack messages is that many DHCP servers can make different offers to the client and the client ends up requesting only one of these offers.

DHCP runs on top of UDP. The server runs on top of port 67 and the client runs on top of port 68.
DHCP Messages

- **Discover:**
  - src = 0.0.0.0*  
  - dst = 255.255.255.255+  
  - port 68 UDP  
  - port 67 UDP  
  - offered IP address = none  
  - by server = none  
  - for period = none  
  - seq number = 516

- **Offer:**
  - src = 223.1.2.5  
  - dst = 255.255.255.255  
  - port 67 UDP  
  - port 68 UDP  
  - offered IP address = 223.1.2.159  
  - by server = 223.1.2.5  
  - for period = 1 hour  
  - seq number = 516

* "do not reply to this IP address"
+ "this msg is destined to all hosts"
"in same subnet"
DHCP Messages (Continues) 66

- Request:
  src = 0.0.0.0
  dst = 255.255.255.255
  offered IP address = 223.1.2.159
  by server = 223.1.2.5
  for period = 1 hour
  seq number = 517

- Ack:
  src = 223.1.2.5
  dst = 255.255.255.255
  offered IP address = 223.1.2.159
  by server = 223.1.2.5
  for period = 1 hour
  seq number = 517
Internet Control Msg Protocol (ICMP)

- If pkt p is dropped before reaching its dst, then pkt q is sent back to src of p to inform it that p has been dropped

- IP header of q has:
  - src = IP address of computer where p is dropped
  - dst = IP address of src of p

- Data of q has:
  - ICMP header (type, code) describing why p is dropped
  - IP header of p
  - 8 bytes of data of p

- Examples of ICMP header:

<table>
<thead>
<tr>
<th>type</th>
<th>code</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>dst net unreachable</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>dst host unreachable</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>TTL expired</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>IP header bad</td>
</tr>
</tbody>
</table>
IP Headers

- **IPv4 header:**
  includes following fields:
  - version of IP (4)
  - TTL (at most 64)
  - Upper layer protocol (UDP, TCP, ICMP)
  - IP checksum* (2 Bytes)
  - IP address of src (4 Bytes)
  - IP address of dst (4 Bytes)

- **IPv6 header:**
  includes following fields:
  - version of IP (6)
  - traffic class
  - flow
  - hop limit (at most 64)
  - next header (UDP, TCP, ICMP)
  - IP address of src (16 Bytes)
  - IP address of dst (16 Bytes)

* IPv6 header has no checksum
Internet Transition from IP4 to IP6

- This transition will take decades to complete.
- During this transition, many computers will be using IP4 only while others will be using IP4 and IP6.
- To transmit pkts bet. computers that use IP4 only and those that use IP4 and IP6, employ a technique "Pkt tunneling."
- An example of pkt tunneling is discussed next.
Consider a pkt generated at host A, then transmitted through routers B and C, and finally reach host D. A, B, D use both IP4 and IP6 and C uses IP4 only.

First, the pkt is transmitted as Pkt1 from A to B. Then it is transmitted as Pkt2 from B to C and from C to D:

- Pkt1 = (IP6, next header, src A, dst D, ..)

- Pkt2 = (IP4, protocol IP6, src A, dst D, (IP6, next header, src A, dst D))

- Pkt1 is "tunneled" inside Pkt2
Private Networks

- IP addresses used in one private network are also used in all private networks

- example: assume that the IP addresses in all private networks are taken from the IP address block (10.0.0.0/24)

- assume also that a private net N has a web client c running on port 33450 in host ch whose IP address is (10.0.0.5)

- also assume that client c needs to send pkt 1 to a web server s running on port 80 in host sh whose IP address is (138.1.4.7) and c needs to rcv later a reply pkt 4 from s.
Problem of Private Networks

- the two pkts pkt1 and pkt4 are defined as follows:

  pkt1: (src=10.0.0.5, src port=33450, 
dst=138.1.4.7, dst port=80)

  pkt4: (src=138.1.4.7, src port=80, 
dst=10.0.0.5, dst port=33450)

- there is a problem concerning routing of pkt4. To solve this problem, use the technique of Network Address Translation (NAT)
Network Address Translation (NAT)

- After pkt1 is generated by c, it is routed inside private net N until it reaches router R that connects N with rest of Internet.

- R translates pkt1 to pkt2 as follows:
  - pkt2: (src = 99.3.7.7, src port = 50001, 
    dst = 138.1.4.7, dst port = 80)
  - where src = 99.3.7.7 is the IP address of the interface that connects R with rest of Internet and src port = 50001 is selected randomly by R.

- Finally, R forwards pkt2 to its dst and adds the following entry to its NAT table:

  \[
  \begin{array}{c}
  \text{dst} = 99.3.7.7 \\
  \text{dst port} = 50001
  \end{array}
  \rightarrow
  \begin{array}{c}
  \text{dst} = 10.0.0.5 \\
  \text{dst port} = 33450
  \end{array}
  \]

- When src vs pkt2, it computes pkt3 as follows:
  - pkt3: (src = 138.1.4.7, src port = 80, 
    dst = 99.3.7.7, dst port = 50001)
  - and forwards pkt3 to its dst.
NAT Continues

* when R rcvs pkt3, it uses its NAT table to translate pkt3 into pkt4 and forwards pkt4 to its dst (over the private net N)
Internet consists of networks. Each is either an access network or an ISP network. Each network is called an Autonomous System (or AS).

Two types of routing protocols in Internet:

- **Intra-AS Routing Protocols:**
  - route pkts within one AS
  1. Routing Information Protocol (RIP)
  2. Open Shortest Path First (OSPF)

- **Inter-AS Routing Protocols:**
  - route pkts across multiple ASes
  3. Border Gateway Protocol (BGP)
Routing Information Protocol (RIP) 76

each router in AS, that uses RIP, has a routing table with 3 columns.

Routing table of A

<table>
<thead>
<tr>
<th>dst subnet</th>
<th># hops reach dst</th>
<th>best ngh reach dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>v</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>w</td>
<td>2</td>
<td>B</td>
</tr>
<tr>
<td>x</td>
<td>2</td>
<td>B</td>
</tr>
<tr>
<td>y</td>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td>z</td>
<td>2</td>
<td>C</td>
</tr>
</tbody>
</table>

RIP is appl. running on top of UDP port 520
Operation of RIP

- each router in AS, that uses RIP, sends its routing table (in a RIP advertisement msg) to each adjacent router in AS every 30 seconds.

- when a router rcvs a routing table from an adjacent router, it uses the rcvd table to update its own table

- if a router does not rcv a routing table from an adjacent router for 180 seconds, it considers the adjacent router dead and updates its routing table accordingly

- eventually the routing tables of all routers in AS assume their correct entries
Example of RIP Operation

- Network that uses RIP

- Initial RT of A:

- Initial RT of B:

- RT of B after rcving initial RT of C:

- RT of A after rcving RT of B:

- RT of A after B becomes "dead"

- When # hops to reach dst from A is 15 (max value), then this means dst is not reachable from A

RT stands for routing table
Open Shortest Path First (OSPF) 79

- every 30 minutes each router in AS, that uses OSPF, broadcasts its local state to every other router in AS.

  local state of A:
  
  local state of B:
  
  local state of C:

- each router, say A, puts all rcvd local states together to construct global state of AS and compute its routing table:

  global state of AS:
  
  routing table of A

<table>
<thead>
<tr>
<th>dst</th>
<th>best ngh to reach dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>-</td>
</tr>
<tr>
<td>v</td>
<td>B</td>
</tr>
<tr>
<td>w</td>
<td>-</td>
</tr>
<tr>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>y</td>
<td>B</td>
</tr>
<tr>
<td>z</td>
<td></td>
</tr>
</tbody>
</table>
Border Gateway Protocol (BGP)

- A router that is connected to computers in two or more ASes is called gateway.

- BGP informs each router R how to route packets to an IP prefix pf (i.e., block of IP addresses) that is used in ASi different from ASj, where R is located:

- BGP consists of two parts:
  i. external BGP (eBGP):
      informs each gateway

  ii. internal BGP (iBGP):
      informs each router that is not a gateway

- Each router has a BGP routing table:
External BGP

- there is a TCP connection between each two gateways in same AS, and each two "adjacent" gateways in different ASes.

- these gateway pairs are called peers. They send route advertisements as follows:
  (prefix, AS path, next-hop)

```
AS1    1   AS2    2   AS3
B1  x  A2  y  B2  A3
```

- each gateway Ai or Bj adds an entry to its BGP routing table:

  A2: (pf, B1)  B2: (pf, best nght to x)
  A3: (pf, B2)
Internal BGP

- There is a TCP connection between each two routers in the same AS, provided one of them is a gateway.

\[
\text{adv} : (pf, (AS_i, ..., AS_j), x) \\
\text{adv}^- : (pf, x)
\]

- Each router that is not a gateway adds an entry to its BGP routing table:
  \[ C_k : (pf, \text{best ngh to reach } x) \]
Broadcast Protocols

- a broadcast network is an undirected connected graph \((N, E)\), where each node in \(N\) represents a routed and its attached hosts, and each edge in \(E\) represents a subnetwork.

- periodically, each node generates a msg that needs to be broadcasted to every node in the network

- one protocol for broadcasting generated msgs to every node in the network is called the broadcast flooding protocol
Broadcast Flooding Protocol

- Each (broadcast) msg is uniquely identified by \((u, sq)\), where \(u\) is the node that generated the msg, and \(sq\) is seq# generated by \(u\) for the msg.

- If latest msg generated by \(u\) is \((u, sq)\), then next msg generated by \(u\) is \((u, sq+1)\), and \(u\) forwards a copy of msg to every neighbor of \(u\).

- Each node \(v\) in the network keeps track of seq# of latest msg that a node \(u\) generated and node \(v\) rcvd.

- When a node \(v\) rcvs a msg \((u, sq)\) from a neighbor \(w\), and \(v\) observes that \(sq \leq \text{seq# of the latest msg that } u \text{ generated and } v \text{ rcvd}\), then \(v\) discards the msg. Otherwise, \(v\) forwards a copy of msg to each of \(v\)'s neighbors other than \(w\).
Multicast Protocols

- A multicast network is a broadcast network \((N,E)\), where some of the nodes in \(N\) are called mg-nodes to signify that these nodes belong to the same multicast group.

- The mg-nodes in a multicast network satisfy three conditions:
  1. For every pair of mg-nodes \(u\) and \(v\), there is a path of mg-nodes that connects \(u\) and \(v\).
  2. The network has no cycle whose nodes are all mg-nodes.
  3. Each mg-node knows every neighboring mg-node.

- Periodically, each mg-node generates a msg that needs to be multicast to every mg-node in the network. The multicast tree protocol (next) can be used.
Multicast Tree Protocol

- periodically, each mg-node in the network generates a msg then forwards a copy of the msg to every neighboring mg-node

- when an mg-node v receives a msg from a neighboring mg-node w, then v forwards a copy of the msg to every neighboring mg-node other than w