Chapter 6

The Network Layer
Network Layer Design Issues

- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets
Store-and-Forward Packet Switching

The environment of the network layer protocols.
The Network Layer

Network Layer Design Issues

Services Provided to the Transport Layer

The network layer services have been designed with the following goals in mind.

1. The services should be independent of the subnet technology.

2. The transport layer should be shielded from the number, type, and topology of the subnets present.

3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.
Internal Organization of the network Layer

- connection - Virtual circuit
- connectionless - datagram

Virtual circuits are generally used in subnets whose primary service is connection-oriented, so we will describe them in that context. The idea behind virtual circuits is to avoid having to choose a new route for every packet or cell sent. Instead, when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and remembered. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated.

In contrast, with a datagram subnet no routes are worked out in advance, even if the service is connection-oriented. Each packet sent is routed independently of its predecessors. Successive packets may follow different routes. While datagram subnets have to do more work, they are also generally more robust and adapt to failures and congestion more easily than virtual circuit subnets. We will discuss the pros and cons of the two approaches later.
Implementation of Connectionless Service

Routing within a datagram network
Implementation of Connection-Oriented Service Routing within a virtual-circuit network

A’s table

<table>
<thead>
<tr>
<th>In</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>1</td>
</tr>
<tr>
<td>H3</td>
<td>1</td>
</tr>
</tbody>
</table>

C’s Table

<table>
<thead>
<tr>
<th>In</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>1</td>
</tr>
</tbody>
</table>

A’s Table

<table>
<thead>
<tr>
<th>In</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
</tr>
</tbody>
</table>

E’s Table

<table>
<thead>
<tr>
<th>In</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

F’s Table

<table>
<thead>
<tr>
<th>In</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>2</td>
</tr>
</tbody>
</table>

ISP’s equipment

Routing within a virtual-circuit network
Connections should have the following properties:

1. Before sending data, a network layer process on the sending side must set up a connection to its peer on the receiving side. This connection, which is given a special identifier, is then used until all the data have been sent, at which time it is explicitly released.

2. When a connection is set up, the two processes can enter into a negotiation about the parameters, quality, and cost of the service to be provided.

3. Communication is in both directions, and packets are delivered in sequence.

4. Flow control is provided automatically to prevent a fast sender from dumping packets into the pipe at a higher rate than the receiver can take them out, thus leading to overflow.
# Comparison of Virtual-Circuit and Datagram Subnets

<table>
<thead>
<tr>
<th>Issue</th>
<th>Datagram subnet</th>
<th>Virtual-circuit subnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit setup</td>
<td>Not needed</td>
<td>Required</td>
</tr>
<tr>
<td>Addressing</td>
<td>Each packet contains the full source and destination address</td>
<td>Each packet contains a short VC number</td>
</tr>
<tr>
<td>State information</td>
<td>Routers do not hold state information about connections</td>
<td>Each VC requires router table space per connection</td>
</tr>
<tr>
<td>Routing</td>
<td>Each packet is routed independently</td>
<td>Route chosen when VC is set up; all packets follow it</td>
</tr>
<tr>
<td>Effect of router failures</td>
<td>None, except for packets lost during the crash</td>
<td>All VCs that passed through the failed router are terminated</td>
</tr>
<tr>
<td>Quality of service</td>
<td>Difficult</td>
<td>Easy if enough resources can be allocated in advance for each VC</td>
</tr>
<tr>
<td>Congestion control</td>
<td>Difficult</td>
<td>Easy if enough resources can be allocated in advance for each VC</td>
</tr>
</tbody>
</table>
The desirable properties of a routing algorithm:

a. correctness
b. simplicity
c. robustness
d. stability
e. fairness
f. optimality

Routing algorithms can be grouped into two classes: nonadaptive and adaptive
Routing Algorithms(1)

- Optimality principle
- Shortest path algorithm
- Flooding
- Distance vector routing
- Link state routing
Routing Algorithms (2)

- Broadcast routing
- Multicast routing
- Anycast routing
- Routing for mobile hosts
Routing Algorithms (2)

Conflict between fairness and optimality.
The Optimality Principle

Before getting into specific algorithms, it may be helpful to note that one can make a general statement about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle. It states that if router $J$ is on the optimal path from router $I$ to router $K$, then the optimal path from $J$ to $K$ also falls along the same route. To see this, call the part of the route from $I$ to $J$ $r_1$ and the rest of the route $r_2$. If a route better than $r_2$ existed from $J$ to $K$, it could be concatenated with $r_1$ to improve the route from $I$ to $K$, contradicting our statement that $r_1r_2$ is optimal.
The Optimality Principle

(a) A subnet. (b) A sink tree for router B.
Shortest Path Routing

Labelling Algorithm for Shortest Path Routing

1. Mark the source node as permanent and the working node.

2. Examine each of the nodes adjacent to the working node, label or relabel each one with the shorter distance to the source and the nearest node along the path.

3. Examine all the tentatively labeled nodes in the whole graph and make the one with smallest label permanent take this as the new working node.

4. If the entire graph is searched and made permanent, then trace back from the destination to find the shortest path otherwise go to step 2.
Shortest Path Routing

The first 5 steps used in computing the shortest path from A to D. The arrows indicate the working node.
Flooding
Flooding Algorithm: Every incoming packet is sent out on every outgoing line except the one it's arrived on

Methods for damming the flood
(a) hop count
(b) keeping track of which packets have been flooded (list of sequence numbers)

Modified flooding algorithm: Selective flooding

Application of the flooding algorithm
(a) military
(b) distribution of database
(c) wireless networks
(d) benchmark for other algorithm (shortest path)
Distance Vector Routing

1. Each router maintains a routing table, whose entry contains two parts: the preferred outgoing line to use for that destination and an estimate of the time or distance to that destination.

2. The router is assumed to know the distance to each of its neighbors.

3. Each router sends the routing table (destination, distance) to each neighbor.

4. Based on the collected routing tables from the neighbors, the router computes the shortest distance for each destination and update the routing table.
**Distance Vector Routing**

(a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.
Distance Vector Routing (2)

- Good news propagates fast

- Routers may take infinite number of exchanges to know bad news (a router was down)

\[\begin{array}{cccc}
A & B & C & D & E \\
\bullet & \bullet & \bullet & \bullet & \bullet & \text{Initially} \\
1 & \bullet & \bullet & \bullet & \text{After 1 exchange} \\
1 & 2 & \bullet & \bullet & \text{After 2 exchanges} \\
1 & 2 & 3 & \bullet & \text{After 3 exchanges} \\
1 & 2 & 3 & 4 & \text{After 4 exchanges} \\
\end{array}\]

\[\begin{array}{cccc}
A & B & C & D & E \\
1 & 2 & 3 & 4 & \text{Initially} \\
3 & 2 & 3 & 4 & \text{After 1 exchange} \\
3 & 4 & 3 & 4 & \text{After 2 exchanges} \\
5 & 4 & 5 & 4 & \text{After 3 exchanges} \\
5 & 6 & 5 & 6 & \text{After 4 exchanges} \\
7 & 6 & 7 & 6 & \text{After 5 exchanges} \\
7 & 8 & 7 & 8 & \text{After 6 exchanges} \\
\vdots \\
\bullet & \bullet & \bullet & \bullet & \bullet \\
\end{array}\]

The count-to-infinity problem.
Link State Routing

1. Discover neighbors, learn network addresses.
2. Set distance/cost metric to each neighbor.
3. Construct packet telling all learned.
4. Send packet to, receive packets from other routers.
5. Compute shortest path to every other router.
Learning about the Neighbors

- The LAN is modeled as a node (artificial)

(a) Nine routers and a LAN. (b) A graph model of (a).
Setting Link Costs

• Have a distance or cost metric for finding shortest paths.
• Make the cost inversely proportional to the bandwidth (high priority for high-capacity path).
• Consider delay as part of cost.
Building Link State Packets

Once the information needed for the exchange has been collected, the step is for each router to build a packet containing all the data. The packet start with the identity of the sender, followed by a sequence number and age, and a list of neighbors. For each neighbor, the delay to that neighbor is given.

![Diagram of a subnet and link state packets]

Fig. (a) A subnet. (b) The link state packets for this subnet.

The data base can be updated periodically or when some significant event occurs.
Distributing the Link State Packets

The fundamental idea is to use flooding. Each packet contains a sequence number that is incremented for each new packet sent. Routers keep track of all the (source router, sequence) pairs they see. When a new link state packet comes in, it is checked against the existing list. If it is new, it is forwarded. Otherwise, it is discarded. If a packet with a sequence number lower than the highest one seen so far ever arrives, it is rejected.
Computing the New Routes

Once a router has accumulated a full set of link state packets, the shortest Path Routing (5.2.2) (Dijkstra's algorithm) can be run locally. The results can be installed in the routing table.

1. For a subnet with n routers, each of which has k neighbors, the memory required is proportional to kn.
2. The computation time can be an issue.
3. Hardware and software failure may cause problems (e.g., A router may claim a non-existing link).
4. Run out of memory or calculation error may be an issue.

Link state routing is widely used (e.g., OSPF).
This algorithm has a few problems, but they are manageable. First, if the sequence numbers wrap around, confusion will reign. The solution here is to use a 32-bit sequence number. With one link state packet per second, it would take 137 years to wrap around, so this possibility can be ignored.

Second, if a router ever crashes, it will lose track of its sequence number. If it starts again at 0, the next packet will be rejected as a duplicate.

Third, if a sequence number is ever corrupted and 65,540 is received instead of 4 (a 1-bit error), packets 5 through 65,540 will be rejected as obsolete, since the current sequence number is thought to be 65,540.

The solution to all these problems is to include the age of each packet after the sequence number and decrement it once per second. When the age hits zero, the information from that router is discarded. Normally, a new packet comes in, say, every 10 seconds, so router information only times out when a router is down (or six consecutive packets have been lost, an unlikely event). The age field is also decremented by each router during the initial flooding process, to make sure no packet can get lost and live for an indefinite period of time (a packet whose age is zero is discarded).
Distributing the Link State Packets

• Flooding is used to distribute the link state packet

<table>
<thead>
<tr>
<th>Source</th>
<th>Seq.</th>
<th>Age</th>
<th>Send flags</th>
<th>ACK flags</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>A  C  F</td>
<td>A  C  F</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>21</td>
<td>60</td>
<td>0  1  1</td>
<td>1  0  0</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>21</td>
<td>60</td>
<td>1  1  0</td>
<td>0  0  1</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>21</td>
<td>59</td>
<td>0  1  0</td>
<td>1  0  1</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>20</td>
<td>60</td>
<td>1  0  1</td>
<td>0  1  0</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>21</td>
<td>59</td>
<td>1  0  0</td>
<td>0  1  1</td>
<td></td>
</tr>
</tbody>
</table>

Over a router has accumulates a full set of link state packet, it can construct the entire subnet graph and compute the shortest path.

The packet buffer for router B in the previous power point.
Hierarchical Routing

Hierarchical routing.
Broadcast Routing

a. the source simply sends a distinct packet to each destination
b. flooding
c. multidestination routing
d. spanning tree
e. reverse path forwarding

Fig. Reverse path forwarding.
Broadcast Routing

Reverse path forwarding. (a) A subnet. (b) a Sink tree. (c) The tree built by reverse path forwarding.
Multicast Routing (1)

(a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.
(a) Core-based tree for group 1.
(b) Sending to group 1.
(a) Anycast routes to group 1.
(b) Topology seen by the routing protocol.
Routing for Mobile Hosts

1. Every host has a permanent home location(+886-2-3366-3366) and a home agent. The mobile host must acquire a local network address before it can use the network (a care of address) when it is away from home. The mobile host tells the care of address to its home agent where it is now (send a registration message to the home agent).

2. The sender sends a packet to the mobile host using its permanent address.

3. The home agent intercepts this packet. It then encapsulates the packet with a new header and sends it to the care of address (tunneling).

4. The mobile host sends its reply packet directly to the sender.(Triangle routing)

5. Subsequent packets can be routed directly to the mobile host by tunneling them to the care of address.
Routing for Mobile Hosts

Packet routing for mobile hosts
Congestion Control Algorithms (1)

- Approaches to congestion control
- Traffic-aware routing
- Admission control
- Traffic throttling
- Load shedding
When too much traffic is offered, congestion sets in and performance degrades sharply.
General Principles of Congestion Control

1. Monitor the system.
   – detect when and where congestion occurs.
2. Pass information to where action can be taken.
3. Adjust system operation to correct the problem.
Approaches to Congestion Control

Network provisioning  Traffic-aware routing  Admission control  Traffic throttling  Load shedding

Slower (Preventative)  Faster (Reactive)

Timescales of approaches to congestion control
Traffic-Aware Routing

A network in which the East and West parts are connected by two links.
Traffic Throttling (1)

(a) A congested network. (b) The portion of the network that is not congested. A virtual circuit from A to B is also shown.
Explicit congestion notification
Hop-by-Hop Choke Packets

(a) A choke packet that affects only the source.

(b) A choke packet that affects each hop it passes through.
Load Shedding

discarding packets

• Wine policy

• Milk policy

• Discarding low priority packets

• Random Early Detection
Quality of Service

- Application requirements
- Traffic shaping
- Packet scheduling
- Admission control
- Integrated services
- Differentiated services
Application Requirements (1)

<table>
<thead>
<tr>
<th>Application</th>
<th>Bandwidth</th>
<th>Delay</th>
<th>Jitter</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>File sharing</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Web access</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Remote login</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Audio on demand</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Video on demand</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Telephony</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Videoconferencing</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>

How stringent the quality-of-service requirements are.
ATM networks classify flows into four categories:

1. Constant bit rate (telephony)
2. Real-time variable bit rate (compressed video)
3. Non-real-time variable bit rate (movie over Internet)
4. Available bit rate (file transfer)
Traffic Shaping (Regulating the average rate and burstiness of a flow of data)

The Leaky Bucket Algorithm

Each host is connected to the network by an interface containing a leaky bucket, a finite internal queue. If a packet arrives at the queue when it is full, the packet is discarded. In fact it is nothing other than a single-server queuing system with constant service time.

Fig. (a) A leaky bucket with water. (b) A leaky bucket with packets.
The Token Bucket Algorithm

The leaky bucket holds tokens, generated by a clock at the rate of one token every $\Delta T$ second. For a packet to be transmitted, it must capture and destroy one token. The token bucket algorithm allow idle hosts to save up permission to the maximum size of bucket $n$ for burst traffic latter.

Fig. 5-26. The token bucket algorithm. (a) Before. (b) After.
The Leaky Bucket Algorithm

(a) Input to a leaky bucket.  
(b) Output from a leaky bucket.  Output from a token bucket with capacities of (c) 250 KB, (d) 500 KB, (e) 750 KB,  
(f) Output from a 500KB token bucket feeding a 10-MB/sec leaky bucket.
Calculating the length of the maximum rate burst is slightly tricky. It is not just 1 MB divided by 25 MB/sec because while the burst is being output, more tokens arrive. If we call the burst length $S$ sec, the token bucket capacity $C$ bytes, the token arrival rate $\rho$ bytes/sec, and the maximum output rate $M$ bytes/sec, we see that an output burst contains a maximum of $C + \rho S$ bytes. We also know that the number of bytes in a maximum-speed burst of length $S$ seconds is $MS$. Hence we have

$$C + \rho S = MS$$

We can solve this equation to get $S = C/(M - \rho)$. For our parameters of $C = 250$ KB, $M = 25$ MB/sec, and $\rho = 2$ MB/sec, we get a burst time of about 11 msec. Figure 5-25(d) and Fig. 5-25(e) show the token bucket for capacities of 500-KB and 750 KB, respectively.
Kinds of resources can potentially be reserved for different flows:

1. Bandwidth
2. Buffer space
3. CPU cycles
Because of random arrivals, and random service time distribution, queues can build up and delays can occur.

Consider that packets arrive with Poisson distribution with mean $\lambda$ packet/sec, the service time is exponential with $\mu$ packet/sec. The CPU can be modeled as an M/M/1 queue.

The delay is given as

$$T = \frac{\rho}{\lambda(1 - \rho)} \quad \rho = \frac{\lambda}{\mu}$$

$$= \frac{1}{\mu} \frac{1}{1-\rho}$$

For example =  $\lambda = 950000 \text{ packets/ sec}$

$\mu = 1000000 \text{ packets/ sec}$

$\rho = 0.95$

$$T = 20 \mu \text{ seconds}$$
Packet Scheduling

1. Fair queueing algorithm

The router has separate queues for each output line, one for each flow. When a line becomes idle, the router scans the queues round robin, taking the first packet on the next queue.

2. Weighted fair queueing algorithm

* for priority service (e.g. giving some servers more bandwidth)

(a) A router with five packets queued for line O.
(b) Finishing times for the five packets.
Admission Control

When the incoming traffic from some flow is well shaped, the capacity can be reserved in advance on the routers along the path. When a flow with flow specification (produced by the sender) is offered, the router along the route examines the flow specification and modifies the parameters as need be. Finally the acceptable flow specification (acceptable parameters) can be established.
An example of flow specification

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token bucket rate</td>
<td>Bytes/sec</td>
<td>maximum sustained rate average over a long time interval</td>
</tr>
<tr>
<td>Token bucket size</td>
<td>Bytes</td>
<td>maximum capacity of the bucket</td>
</tr>
<tr>
<td>Peak data rate</td>
<td>Bytes/sec</td>
<td>maximum tolerated transmission rate</td>
</tr>
<tr>
<td>Minimum packet size</td>
<td>Bytes</td>
<td>related to protocols and processing time</td>
</tr>
<tr>
<td>Maximum packet size</td>
<td>Bytes</td>
<td></td>
</tr>
</tbody>
</table>
Integrated Services

It was aimed at both unicast (streaming video) and multicast (digital television broadcasting) applications.

In many multicast applications groups can change membership dynamically. Under these conditions, the approach of having the senders reserve bandwidth in advance does not work well. If the number of receivers are large, the dynamical grouping does not work at all.
Integrated Services (flow-based algorithms)

RSVP-The ReSerVation Protocol

(a) A network,  (b) The multicast spanning tree for host 1.  
(c) The multicast spanning tree for host 2.
RSVP-The ReSerVation Protocol (2)

(a) Host 3 requests a channel to host 1. (b) Host 3 then requests a second channel, to host 2. (c) Host 5 requests a channel to host 1.
Differentiated services (class-based)

Differentiated Service (DS) can be offered by a set of routers forming an administrative domain (e.g. an ISP or a Telco). The administration defined a set of service classes with corresponding forwarding rules. Some classes (e.g. premium service) with higher requirements may have better service than others. DS scheme does not require advance setup, resource reservation or time consuming end-to-end negotiation for each flow. This makes DS relatively easy to implement.

For example, the operator reserves resources for all IP-phone users (the IP-phone class), on contrary, a flow-based scheme, each phone call gets its own resources and guarantees.
Differentiated Services (class-based service)  

Expedited Forwarding

- Two queues for each output line, one for expedited packets and one for regular packets.

Expedited packets experience a traffic-free network.
Assured Forwarding

The assured forwarding scheme which manages the service classes, specifies three packet discard probabilities and four priority classes, each class having its own resources. Taken together, three two factors define 12 service classes.

Implementation procedure

1. Classify the packets into one of the four classes (on the sending host or the ingress router)
2. Mark the packets (type A service in IP, which is specified in the type of service field)
3. Pass the packets through a shaper/dropper filter (e.g. leaky buckets or token buckets)
Differentiated Services (2)

A possible implementation of assured forwarding
Internetworking

• How networks differ
• How networks can be connected
• Tunneling
• Internetwork routing
• Packet fragmentation
### How Networks Differ

Some of the many ways networks can differ

<table>
<thead>
<tr>
<th>Item</th>
<th>Some Possibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service offered</td>
<td>Connectionless versus connection oriented</td>
</tr>
<tr>
<td>Addressing</td>
<td>Different sizes, flat or hierarchical</td>
</tr>
<tr>
<td>Broadcasting</td>
<td>Present or absent (also multicast)</td>
</tr>
<tr>
<td>Packet size</td>
<td>Every network has its own maximum</td>
</tr>
<tr>
<td>Ordering</td>
<td>Ordered and unordered delivery</td>
</tr>
<tr>
<td>Quality of service</td>
<td>Present or absent; many different kinds</td>
</tr>
<tr>
<td>Reliability</td>
<td>Different levels of loss</td>
</tr>
<tr>
<td>Security</td>
<td>Privacy rules, encryption, etc.</td>
</tr>
<tr>
<td>Parameters</td>
<td>Different timeouts, flow specifications, etc.</td>
</tr>
<tr>
<td>Accounting</td>
<td>By connect time, packet, byte, or not at all</td>
</tr>
</tbody>
</table>
How Networks Can Be Connected

MPLS: Multi Protocol Label Switching
(a) A packet crossing different networks.
(b) Network and link layer protocol processing.
Tunneling (1)

Tunneling a packet from Paris to London.
Tunneling a car from France to England
Fragmentation

Each network imposes some maximum size on its packets. These limits have various causes, among them:

1. Hardware (e.g., the width of a TDM transmission slot).

2. Operating system (e.g., all buffers are 512 bytes).

3. Protocols (e.g., the number of bits in the packet length field).

4. Compliance with some (inter)national standard.

5. Desire to reduce error induced retransmissions to some level.

6. Desire to prevent one packet from occupying the channel too long.
Fragmentation

(a) Transparent fragmentation.  
(b) Nontransparent fragmentation.

(a) G₁ fragments a large packet. G₂ reassembles the fragments. 
(b) G₁ fragments a large packet. The fragments are not reassembled until the final destination (a host) is reached.
Fragmentation (2)

Fragmentation when the elementary data size is 1 byte.
(a) Original packet, containing 10 data bytes.
(b) Fragments after passing through a network with maximum packet size of 8 payload bytes plus header.
(c) Fragments after passing through a size 5 gateway.
Packet Fragmentation (6)

Path Maximum Transmission Unit (MTU) Discovery
The Network Layer in the Internet (1)

- The IP Version 4 Protocol
- IP Addresses
- IP Version 6
- Internet Control Protocols
- Label Switching and MPLS
- OSPF—An Interior Gateway Routing Protocol
- BGP—The Exterior Gateway Routing Protocol
- Internet Multicasting
- Mobile IP
The network layer in the Internet

A machine is on the Internet if it runs the TCP/IP protocol stack, has an IP address and has the ability to send IP packets to all the other machines on the Internet.

Communication in the Internet works as follows. The transport layer takes data streams and breaks them up into datagrams. In theory, datagrams can be up to 64 Kbytes each, but in practice they are usually around 1500 bytes. Each datagram is transmitted through the Internet, possibly being fragmented into smaller units as it goes. When all the pieces finally get to the destination machine, they are reassembled by the transport layer, which inserts it into the receiving process' input stream.
Design Principles for Internet

1. Make sure it works.
2. Keep it simple.
3. Make clear choices.
4. Exploit modularity.
5. Expect heterogeneity.
6. Avoid static options and parameters. (negotiation is preferred)
7. Look for a good design; it need not be perfect.
8. Be strict when sending and tolerant when receiving.
9. Think about scalability.
The Internet is an interconnected collection of many networks.
The IP Protocol

<table>
<thead>
<tr>
<th>Version</th>
<th>IHL</th>
<th>Differentiated Services</th>
<th>Total length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Identification**: Fragment offset
- **Time to live**: Packet lifetime
- **Protocol**: The IPv4 (Internet Protocol) header.
- **Source address**
- **Destination address**
- **Options (0 or more words)**

**version**: version of the protocol

**IHL**: length of the header in 32-bit words ≥ 5

**Type of service**: e.g. reliability, speed

**Total length**: length of header and data, ≤ 65,535 bytes

**Identification**: determine which datagram a fragment belongs to

**DF**: Do not Fragment

**MF**: More Fragment

**Fragment offset**: Tell where in the current datagram this fragment belongs (A maximum of $2^{13} = 8192$ fragments per datagram)

**Time to line**: Packet lifetime ≤ 255 seconds
### The IP Protocol (2)

Some of the IP options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Specifies how secret the datagram is</td>
</tr>
<tr>
<td>Strict source routing</td>
<td>Gives the complete path to be followed</td>
</tr>
<tr>
<td>Loose source routing</td>
<td>Gives a list of routers not to be missed</td>
</tr>
<tr>
<td>Record route</td>
<td>Makes each router append its IP address</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Makes each router append its address and timestamp</td>
</tr>
</tbody>
</table>
An IP prefix and a subnet mask.
Here, the vertical bar (|) shows the boundary between the subnet number and the host portion.

Figure 5-49. Splitting an IP prefix into separate networks with subnetting.
A set of IP address assignments.

If a site needs 2000 addresses, it is given a block of 2048 addresses. For example, a set of IP addresses assignments is given as follows.

<table>
<thead>
<tr>
<th>University</th>
<th>First address</th>
<th>Last address</th>
<th>How many</th>
<th>Written as</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cambridge</td>
<td>194.24.0.0</td>
<td>194.24.7.255</td>
<td>2048</td>
<td>194.24.0.0/21</td>
</tr>
<tr>
<td>Edinburgh</td>
<td>194.24.8.0</td>
<td>194.24.11.255</td>
<td>1024</td>
<td>194.24.8.0/22</td>
</tr>
<tr>
<td>Oxford</td>
<td>194.24.16.0</td>
<td>194.24.31.255</td>
<td>4096</td>
<td>194.24.16.0/20</td>
</tr>
</tbody>
</table>

A set of IP address assignments.
IP Addresses (4)

Aggregation of IP prefixes
IP Addresses (5)

Longest matching prefix routing at the New York router.
IP Addresses (6)

IP address formats
### Special IP addresses

<table>
<thead>
<tr>
<th>Network</th>
<th>1111</th>
<th>...</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>00</td>
<td>...</td>
<td>00</td>
</tr>
<tr>
<td>127</td>
<td>(Anything)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This host</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A host on this network</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Broadcast on the local network</td>
<td>111111111111111111111111111111111111111111111111111111111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Broadcast on a distant network</td>
<td>111111111111111111111111111111111111111111111111111111111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loopback</td>
<td>127</td>
<td>(Anything)</td>
<td></td>
</tr>
</tbody>
</table>
Placement and operation of a NAT box.
NAT – Network Address Translation

• IP addresses are scarce. NAT is a quick fix. Each company is assigned a few IP addresses for Internet traffic. Within the company, every computer gets a unique IP address for intramural traffic. No packets containing these addresses may appear on the Internet.

• There are three reserved ranges. Most IP packets carry TCP or UDP payload, which have source port and destination port. NAT box will establish translation table based on the TCP or UDP source port and destination port.

Placement and operation of a NAT box.
NAT can quickly solve the IP address shortage problem (even for ADSL, cable modem application)

Here are some of the objections

1. NAT violates the architectural model of IP, i.e. every IP address identifies a single machine worldwide

2. NAT establish the translation table, which changes the Internet from connectionless to a kind of connection-oriented network

3. NAT violates the layer structure

4. NAT can only handle IP packets with TCP or UDP payload, not others

5. NAT will not work when IP addresses are inserted in the body of the text (e.g. FTP, H.323)

6. Each IP address can only map up to 65536-4096=61440 machines

7. Delay the implantation of IPV6
IP Version 6 Goals

- Support billions of hosts
- Reduce routing table size
- Simplify protocol
- Better security
- Attention to type of service
- Aid multicasting
- Roaming host without changing address
- Allow future protocol evolution
- Permit coexistence of old, new protocols.
IP Version 6 (1)

The IPv6 fixed header (required).

<table>
<thead>
<tr>
<th>Version</th>
<th>Diff. Serv.</th>
<th>Flow label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Payload length**
- **Next header**
- **Hop limit**

- **Source address (16 bytes)**
- **Destination address (16 bytes)**
### IPv6 extension headers

<table>
<thead>
<tr>
<th>Extension header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hop-by-hop options</td>
<td>Miscellaneous information for routers</td>
</tr>
<tr>
<td>Destination options</td>
<td>Additional information for the destination</td>
</tr>
<tr>
<td>Routing</td>
<td>Loose list of routers to visit</td>
</tr>
<tr>
<td>Fragmentation</td>
<td>Management of datagram fragments</td>
</tr>
<tr>
<td>Authentication</td>
<td>Verification of the sender’s identity</td>
</tr>
<tr>
<td>Encrypted security payload</td>
<td>Information about the encrypted contents</td>
</tr>
</tbody>
</table>
The hop-by-hop extension header for large datagrams (jumbograms).
The extension header for routing.
# Internet Control Protocols (1)

- Internet Control Message Protocol (ICMP)
- Address Resolution Protocol (ARP)  
  (Neighbor Discovery Protocol, NDP for IP v.6)
- Dynamic Host Configuration Protocol (DHCP)

<table>
<thead>
<tr>
<th>Message type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination unreachable</td>
<td>Packet could not be delivered</td>
</tr>
<tr>
<td>Time exceeded</td>
<td>Time to live field hit 0</td>
</tr>
<tr>
<td>Parameter problem</td>
<td>Invalid header field</td>
</tr>
<tr>
<td>Source quench</td>
<td>Choke packet</td>
</tr>
<tr>
<td>Redirect</td>
<td>Teach a router about geography</td>
</tr>
<tr>
<td>Echo and Echo reply</td>
<td>Check if a machine is alive</td>
</tr>
<tr>
<td>Timestamp request/reply</td>
<td>Same as Echo, but with timestamp</td>
</tr>
<tr>
<td>Router advertisement/solicitation</td>
<td>Find a nearby router</td>
</tr>
</tbody>
</table>

The principal ICMP message types.
Internet Control Protocols (2)

Two switched Ethernet LANs joined by a router
Dynamic Host Configuration Protocol (DHCP)

DHCP allows both manual IP assignment and automatic assignment.

Operation of DHCP.
Label Switching and MPLS (1)

Transmitting a TCP segment using IP, MPLS, and PPP.
Label Switching and MPLS (2)

Forwarding an IP packet through an MPLS network
OSPF-The Interior Gateway Routing Protocol

Design principles

1. published algorithm
2. support variety of distance metrics, including physical distance, delay, etc.
3. dynamic, adapting to changes
4. support routing based on type of service
5. Do load balancing (load splitting)
6. support for hierarchical systems
7. security
The Interior Gateway Routing Protocol:

Open Shortest Path First (OSPF)
OSPF supports three kinds of connections and networks:

1. Point-to-point lines between exactly two routers.

2. Multiaccess networks with broadcasting (e.g., most LANs).

3. Multiaccess networks without broadcasting (e.g., most packet-switched WANs).
An autonomous system
A graph representation of the previous slide.

OSPF—An Interior Gateway Routing Protocol (2)
The relation between ASes, backbones, and areas in OSPF.
### The five types of OSPF messages

<table>
<thead>
<tr>
<th>Message type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello</td>
<td>Used to discover who the neighbors are</td>
</tr>
<tr>
<td>Link state update</td>
<td>Provides the sender’s costs to its neighbors</td>
</tr>
<tr>
<td>Link state ack</td>
<td>Acknowledges link state update</td>
</tr>
<tr>
<td>Database description</td>
<td>Announces which updates the sender has</td>
</tr>
<tr>
<td>Link state request</td>
<td>Requests information from the partner</td>
</tr>
</tbody>
</table>

OSPF—An Interior Gateway Routing Protocol (4)
BGP—The Exterior Gateway Routing Protocol (1)

Examples of routing constraints:

1. No commercial traffic for educat. network
2. Never put Iraq on route starting at Pentagon
3. Choose cheaper network
4. Choose better performing network
5. Don’t go from Apple to Google to Apple
Routing policies between four Autonomous Systems

Routing policy:
TR = Transit
CU = Customer
PE = Peer
BGP—The Exterior Gateway Routing Protocol (3)

Propagation of BGP route advertisements
Mobile IP

Goals

1. Mobile host use home IP address anywhere.
2. No software changes to fixed hosts
3. No changes to router software, tables
4. Packets for mobile hosts – restrict detours
5. No overhead for mobile host at home.

The Routing for Mobile Hosts described in 5.2.10 can be applied for Mobile IP.
1. Describe the registration procedure of routing for mobile hosts.

2. If a mobile host’s home LAN is in Taipei and currently the mobile host visits New York. A caller is in Tokyo. Describe the packet routing for the mobile host when the caller sends packets.

3. Give reasons why the fragmentation is needed.

4. Who reassembles the fragments for a nontransparent fragmentation internetwork?

5. Describe the five steps Link State routing (hand writing).

6. Compare the datagram and virtual-circuit networks (hand writing. No cut and paste)

7. Are there any circumstances when connection-oriented service will deliver packets out of order? Explain.

8. While IP addresses are tried to specific networks. Ether addresses are not. Can you think of a good reason why they are not?