Chapter 5

The Network Layer

2019 Edition

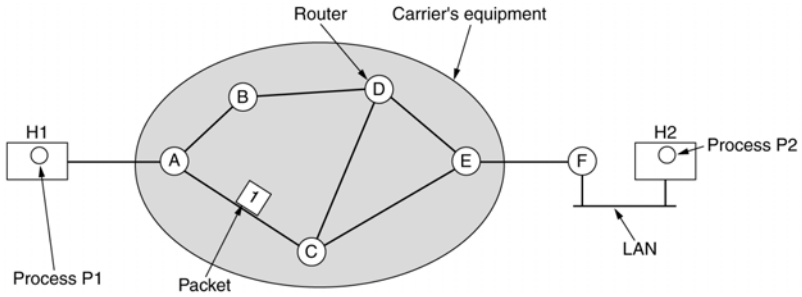
Network Layer Overview

- network layer design issues,functions and services
- routing principle: path selection
- Routing algorithms
- Congestion control
- Service quality
- Internet working
- The network layer in the internet

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

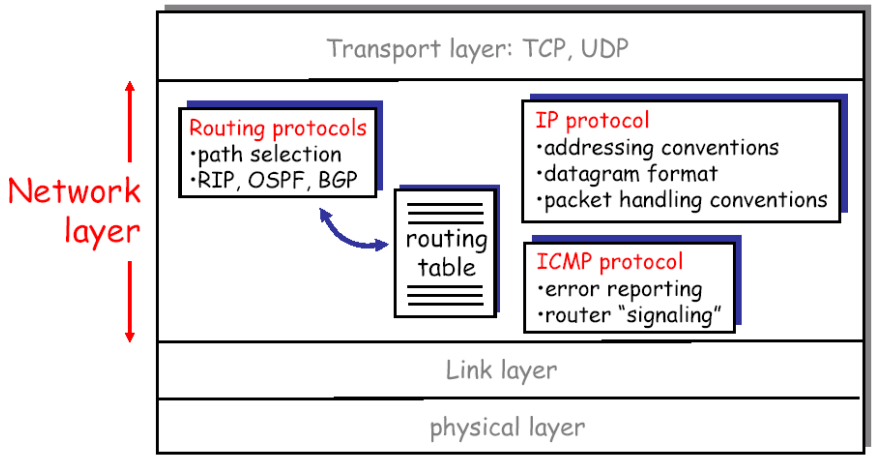
Store-and-Forward Packet Switching



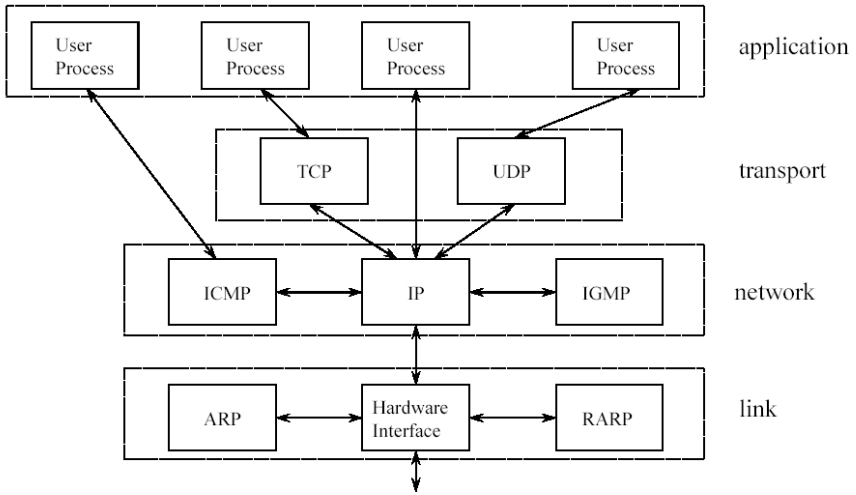
The environment of the network layer protocols.

The Internet Network layer

Host, router network layer functions:



Internet Protocol structure

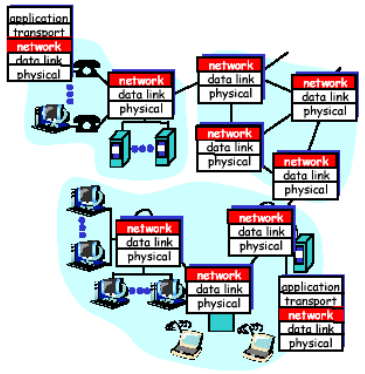


Network layer functions

- transport packet from sending to receiving hosts
- network layer protocols in *every* host, router

three important functions:

- *path determination:* route taken by packets from source to dest. *Routing algorithms*
- *switching:* move packets from router's input to appropriate router output
- *call setup:* some network architectures require router call setup along path before data flows



Other functions?

Network layer functions (cont.)

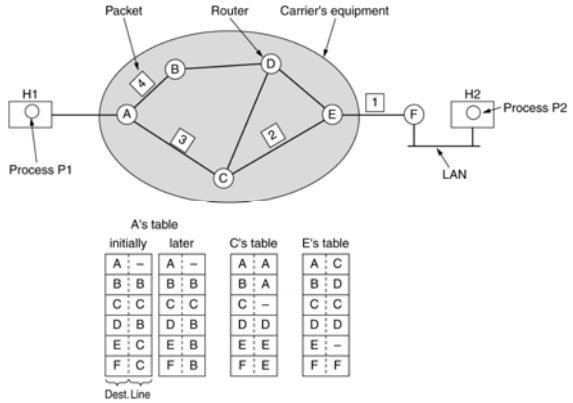
- Addressing
- Congestion control
- Impartiality …

Services acquired from DL layer

Network layer services

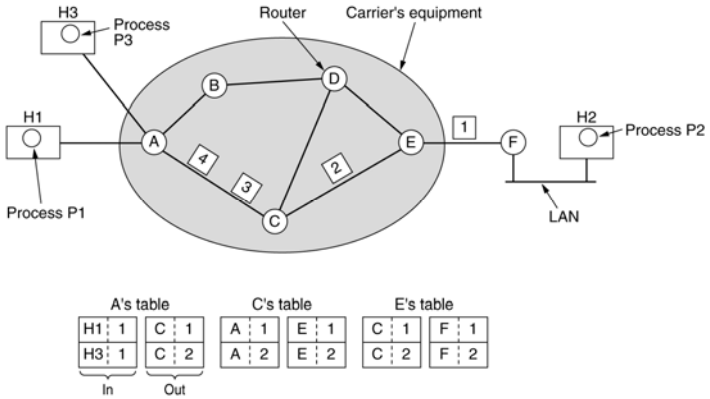
- Design principles
  - Services should be independent to network layer technology
  - Hide network type, number and topology details from upped layers
  - Provide uniform interface to upper layer
- Two kinds of services and why
  - Connect-oriented services
  - Connectionless services

Implementation of Connectionless Service



Routing within a diagram subnet.

Implementation of Connection-Oriented Service



Routing within a virtual-circuit subnet.

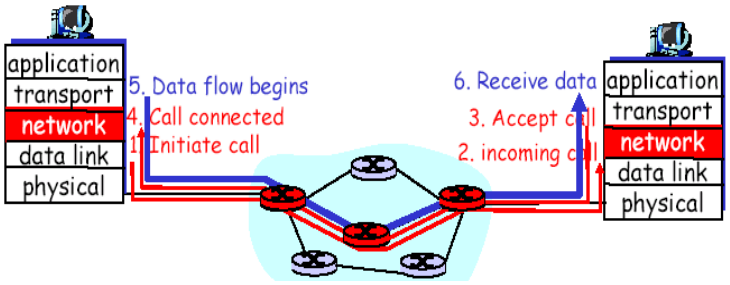
Virtual circuits

"source-to-destination path behaves much like telephone circuit"
- performance-wise
- network actions along source-to-destination path

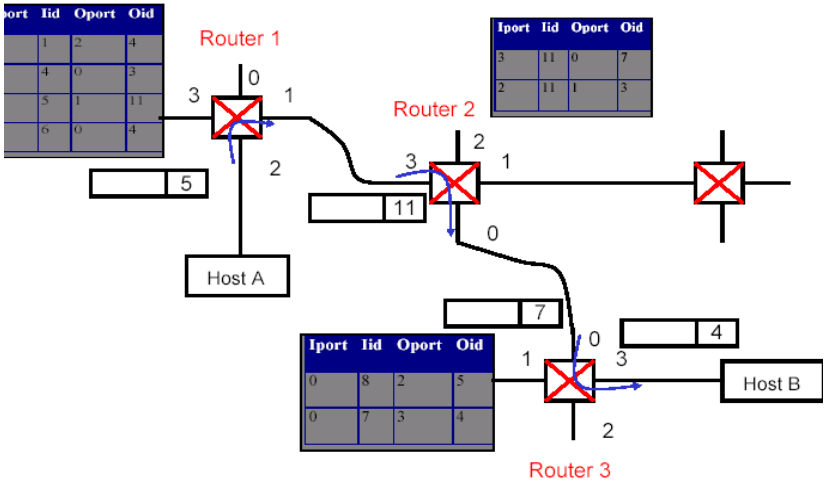
- call setup, teardown for each call *before/after* data flow
- each packet carries VC identifier (not destination host ID)
- *every* router on source-destination path s maintain "state" for each passing connection
  - transport-layer connection only involved two end systems
- link, router resources (bandwidth, buffers) may be *allocated* to VC
  - to get circuit-like performance.

Virtual circuits: signaling protocols

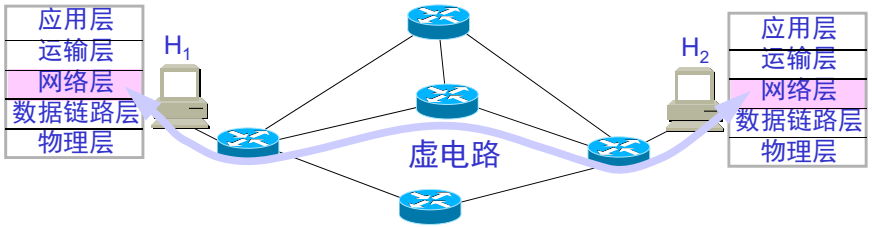
- used to setup, maintain teardown VC
- used in ATM, frame-relay, X.25
- not used in today's Internet



Virtual Circuit Routing



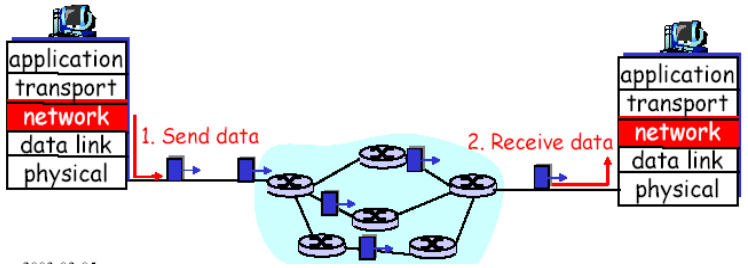
虚电路服务



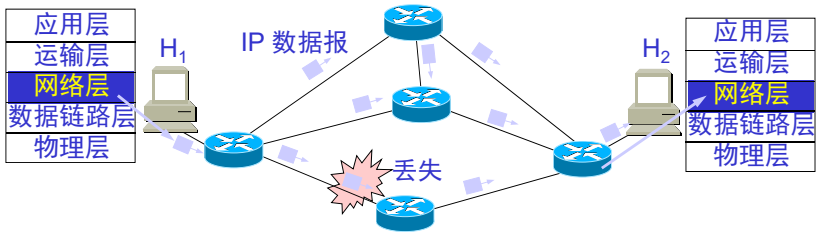
H_1 发送给 H_2 的所有分组都沿着同一条虚电路传送

Datagram networks: the Internet model

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets typically routed using destination host ID
  - packets between same source-dest pair may take different paths

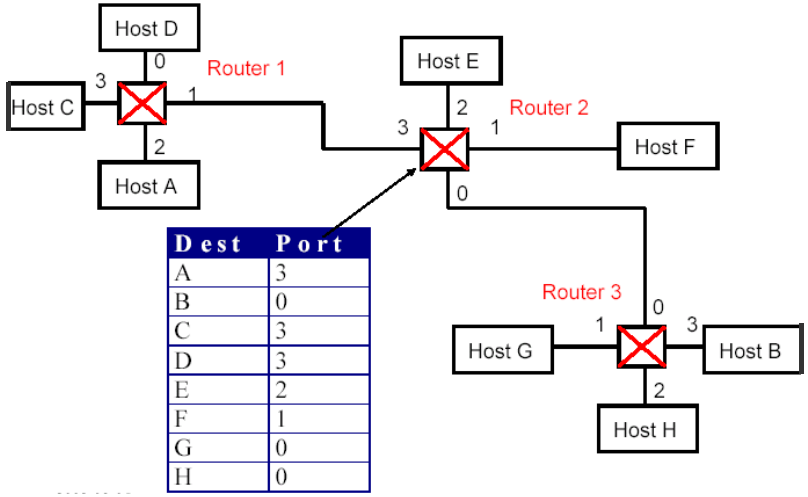


数据报服务



H_1 发送给 H_2 的分组可能沿着不同路径传送

Datagrams Routing



Dest	Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0

Comparison of Virtual-Circuit and Datagram Subnets

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

虚电路服务与数据报服务的对比

对比的方面	虚电路服务	数据报服务
思路	可靠通信应当由网络来保证	可靠通信应当由用户主机来保证
连接的建立	必须有	不需要
终点地址	仅在连接建立阶段使用，每个分组使用短的虚电路号	每个分组都有终点的完整地址
分组的转发	属于同一条虚电路的分组均按照同一路由进行转发	每个分组独立选择路由进行转发
当结点出故障时	所有通过出故障的结点的虚电路均不能工作	出故障的结点可能会丢失分组，一些路由可能会发生变化
分组的顺序	总是按发送顺序到达终点	到达终点时不一定按发送顺序
端到端的差错处理和流量控制	可以由网络负责，也可以由用户主机负责	由用户主机负责

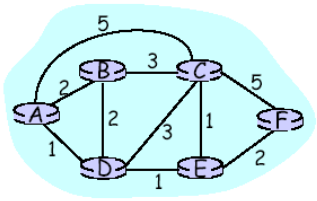
Routing Algorithms

- The Optimality Principle
- Shortest Path Routing
- Flooding
- Distance Vector Routing
- Link State Routing
- Hierarchical Routing
- Broadcast Routing
- Multicast Routing
- Routing for Mobile Hosts
- Routing in Ad Hoc Networks

Routing

Routing protocol

Goal: determine "good" path (sequence of routers) thru network from source to dest.



Graph abstraction for routing algorithms:

- graph nodes are routers
- graph edges are physical links
  - link cost: delay, $ cost, or congestion level

- "good" path:
  - typically means minimum cost path
  - other def's possible
- Routing differences in different services

Routing Algorithm classification

Global or decentralized information?

Global (Centralized):

- all routers have complete topology, link cost info
- Link-state algorithm

Decentralized: (Distributed)

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

Static or dynamic?

Static:

- routes change slowly over time

Dynamic:

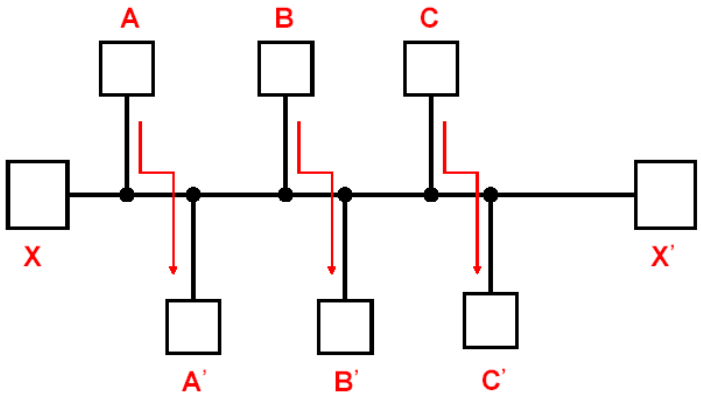
- routes change more quickly
  - periodic update
  - in response to link cost changes

Adaptive or nonadaptive

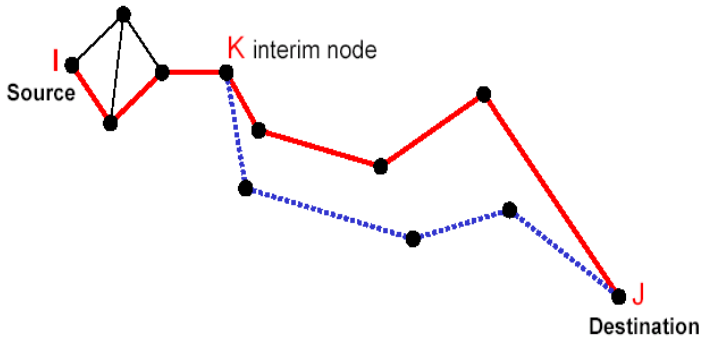
Desirable properties in routing algorithms

- Correctness
- Simplicity
- Robustness
- Stability
- Fairness
- Optimality

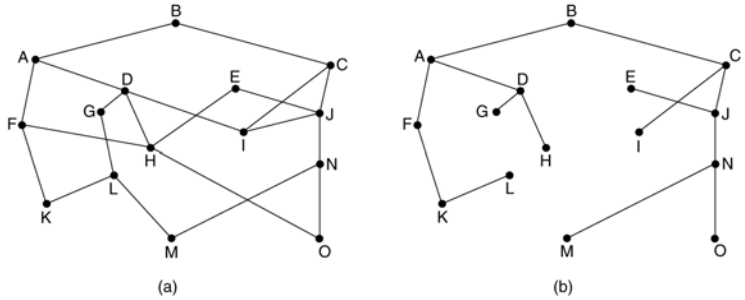
Fairness vs. optimality



The optimality principle

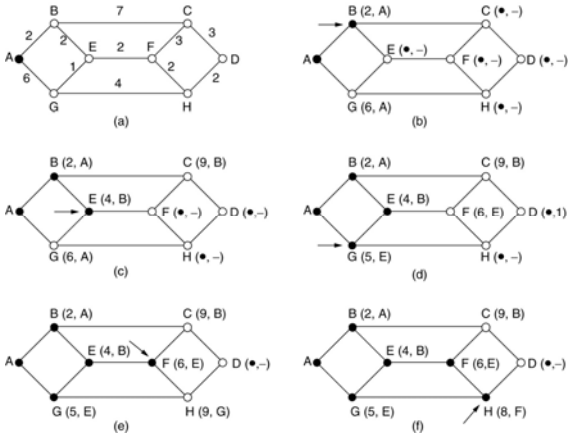


The Optimality Principle



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

Shortest Path Routing



The first 5 steps used in computing the shortest path from A to D. The arrows indicate the working node.

Flooding

```
#define MAX_NODES 1024                  /* maximum number of nodes */
#define INFINITY 1000000000             /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES];/* dist[i][j] is the distance from i to j */

void shortest_path(int s, int t, int path[])
{ struct state {                        /* the path being worked on */
      int predecessor;                  /* previous node */
      int length;                       /* length from source to this node */
      enum {permanent, tentative} label; /* label state */
  } state[MAX_NODES];

  int i, k, min;
  struct state *p;

  for (p = &state[0]; p < &state[n]; p++) { /* initialize state */
      p->predecessor = -1;
      p->length = INFINITY;
      p->label = tentative;
  }
  state[t].length = 0;  state[t].label = permanent;
  k = t;                                /* k is the initial working node */
```

Dijkstra's algorithm to compute the shortest path through a graph.

Flooding (2)

```
  do {                                  /* Is there a better path from k? */
      for (i = 0; i < n; i++)           /* this graph has n nodes */
          if (dist[k][i] != 0 && state[i].label == tentative) {
              if (state[k].length + dist[k][i] < state[i].length) {
                  state[i].predecessor = k;
                  state[i].length = state[k].length + dist[k][i];
              }
          }

      /* Find the tentatively labeled node with the smallest label. */
      k = 0; min = INFINITY;
      for (i = 0; i < n; i++)
          if (state[i].label == tentative && state[i].length < min) {
              min = state[i].length;
              k = i;
          }
      state[k].label = permanent;
  } while (k != s);

  /* Copy the path into the output array. */
  i = 0;  k = s;
  do {path[i++] = k; k = state[k].predecessor; } while (k >= 0);
}
```

Dijkstra's algorithm to compute the shortest path through a graph.

Distance-Vector Algorithms

“距离”的定义

- 从一路由器到直接连接的网络的距离定义为 1。
- 从一个路由器到非直接连接的网络的距离定义为所经过的路由器数加 1。
- RIP 协议中的“距离”也称为“跳数”(hop count)，因为每经过一个路由器，跳数就加 1。
- 这里的“距离”实际上指的是“最短距离”，

“距离”的定义

- RIP 认为一个好的路由就是它通过的路由器的数目少，即“距离短”。
- RIP 允许一条路径最多只能包含 15 个路由器。
- “距离”的最大值为16 时即相当于不可达。可见 RIP 只适用于小型互联网。
- RIP 不能在两个网络之间同时使用多条路由。RIP 选择一个具有最少路由器的路由（即最短路由），哪怕还存在另一条高速(低时延)但路由器较多的路由。

距离向量算法步骤

收到相邻路由器（其地址为 X）的一个 RIP 报文：
(1) 先修改此 RIP 报文中的所有项目：把“下一跳”字段中的地址都改为 X，并把所有的“距离”字段的值加 1。
(2) 对修改后的 RIP 报文中的每一个项目，重复以下步骤：
若项目中的目的网络不在路由表中，则把该项目加到路由表中。
　否则
　　若下一跳字段给出的路由器地址是同样的，则把收到的项目替换原路由表中的项目。
　　否则
　　若收到项目中的距离小于路由表中的距离，则进行更新，否则，什么也不做。
(3) 若 3 分钟还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达路由器，即将距离置为16（距离为16表示不可达）。
(4) 返回。

Distance Vector Routing Algorithm

iterative:
- continues until no nodes exchange info.
- *self-terminating*: no "signal" to stop

asynchronous:
- nodes need *not* exchange info/iterate in lock step!

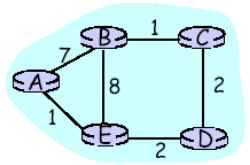
distributed:
- each node communicates *only* with directly-attached neighbors

Distance Table data structure
- each node has its own
- row for each possible destination
- column for each directly-attached neighbor to node
- example: in node X, for dest. Y via neighbor Z:

$$D^X(Y,Z) = \text{distance } \textit{from } X \textit{ to } Y, \textit{ via } Z \text{ as next hop} = c(X,Z) + \min_w\{D^Z(Y,w)\}$$

is sometimes called as Bellman-Ford, or Ford-Fulkerson routing algorithm

Distance Table: example



$$D^E(C,D) = c(E,D) + \min_w\{D^D(C,w)\} = 2+2 = 4$$

$$D^E(A,D) = c(E,D) + \min_w\{D^D(A,w)\} = 2+3 = 5 \text{ loop!}$$

$$D^E(A,B) = c(E,B) + \min_w\{D^B(A,w)\} = 8+6 = 14 \text{ loop!}$$

$D^E()$ | cost to destination via A | B | D

destination	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

Distance table gives routing table

$D^E()$ — cost to destination via

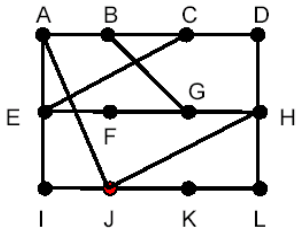
destination	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

Outgoing link to use, cost

destination	
A	A,1
B	D,5
C	D,4
D	D,4

Distance table → Routing table

Distance Vector algorithm



New estimated delay from J

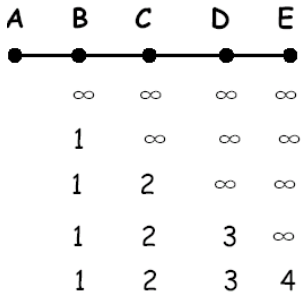
TO	A	I	H	K	line	
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K
Cost of	JA=8	JI=10	JH=12	JK=6		

Vector received from J's neighbor

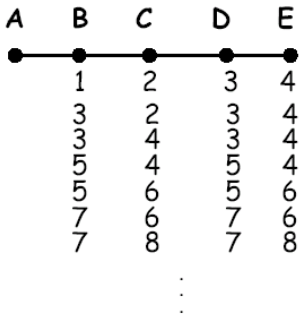
New routing table for J

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

Distant Vector algorithm: the count-to-infinity problem



When A power up later…



When A shutdown or the link between A and B shutdown

The count-to-infinity problem.

内部网关协议 RIP

(Routing Information Protocol)

工作原理

- 路由信息协议 RIP 是内部网关协议 IGP 中最先得到广泛使用的协议。
- RIP 是一种分布式的基于距离向量的路由选择协议。
- RIP 协议要求网络中的每一个路由器都要维护从它自己到其他每一个目的网络的距离记录。

RIP 协议的三个要点

- 仅和相邻路由器交换信息。
- 交换的信息是当前本路由器所知道的全部信息，即自己的路由表。
- 按固定的时间间隔交换路由信息，例如，每隔 30 秒。

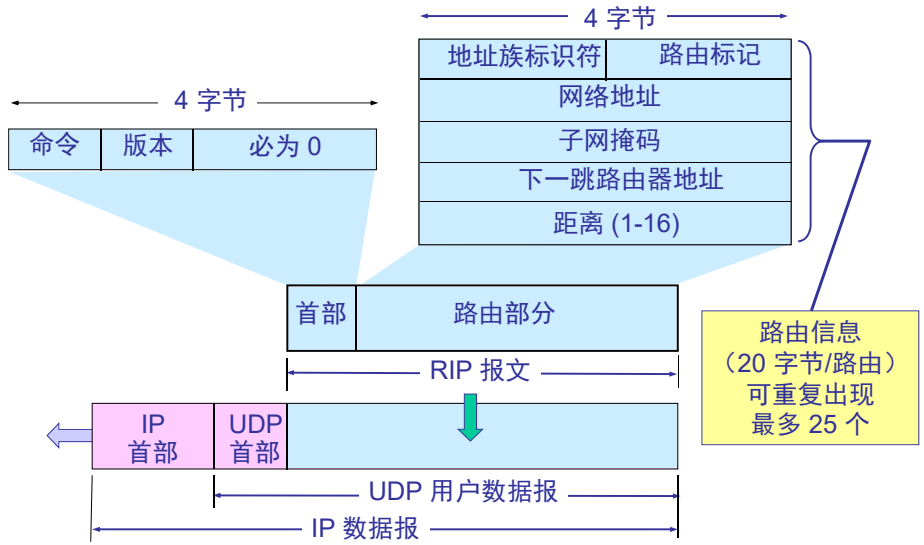
路由表的建立

- 路由器在刚刚开始工作时，只知道到直接连接的网络的距离（此距离定义为1）。
- 以后，每一个路由器也只和数目非常有限的相邻路由器交换并更新路由信息。
- 经过若干次更新后，所有的路由器最终都会知道到达本自治系统中任何一个网络的最短距离和下一跳路由器的地址。
- RIP 协议的收敛(convergence)过程较快，即在自治系统中所有的结点都得到正确的路由选择信息的过程。

路由器之间交换信息

- RIP协议让互联网中的所有路由器都和自己的相邻路由器不断交换路由信息，并不断更新其路由表，使得从每一个路由器到每一个目的网络的路由都是最短的（即跳数最少）。
- 虽然所有的路由器最终都拥有了整个自治系统的全局路由信息，但由于每一个路由器的位置不同，它们的路由表当然也应当是不同的。

RIP2 协议的报文格式

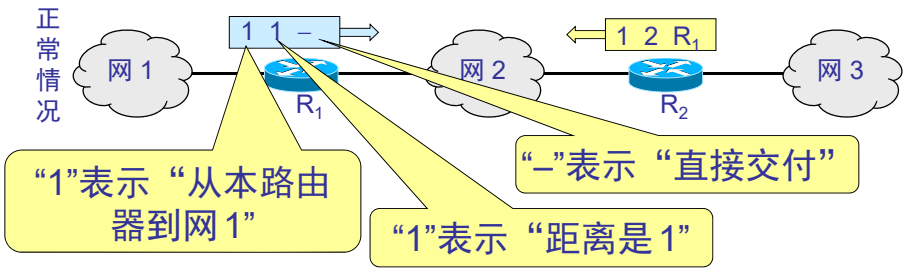


RIP2 的报文
由首部和路由部分组成。

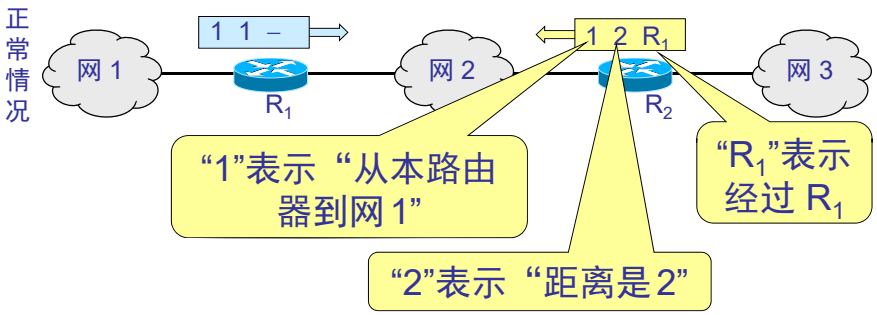
- RIP2 报文中的路由部分由若干个路由信息组成。每个路由信息需要用 20 个字节。地址族标识符（又称为地址类别）字段用来标志所使用的地址协议。
- 路由标记填入自治系统的号码，这是考虑使 RIP 有可能收到本自治系统以外的路由选择信息。再后面指出某个网络地址、该网络的子网掩码、下一跳路由器地址以及到此网络的距离。

RIP 协议的优缺点

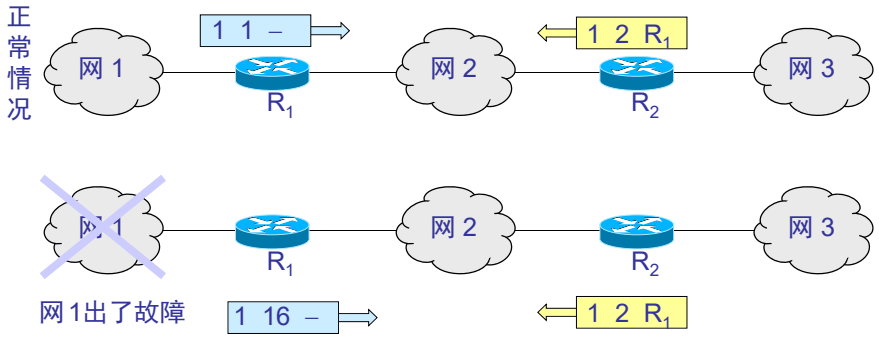
- RIP 存在的一个问题是当网络出现故障时，要经过比较长的时间才能将此信息传送到所有的路由器。
- RIP 协议最大的优点就是实现简单，开销较小。
- RIP 限制了网络的规模，它能使用的最大距离为 15（16 表示不可达）。
- 路由器之间交换的路由信息是路由器中的完整路由表，因而随着网络规模的扩大，开销也就增加。



R_1 说：“我到网 1 的距离是 1，是直接交付。”

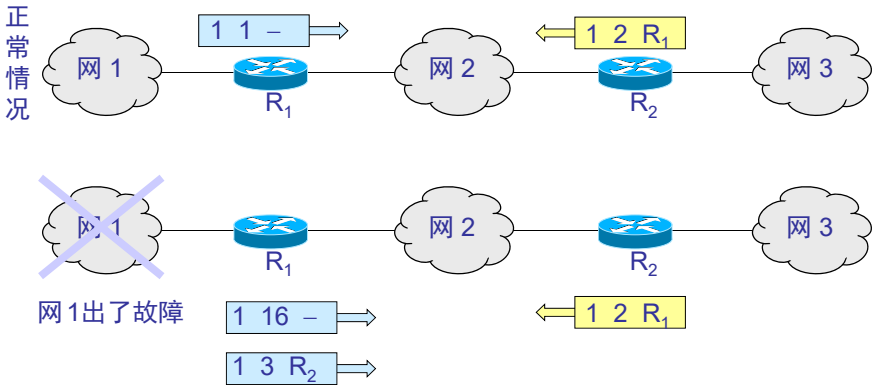


R_2 说：“我到网 1 的距离是 2，是经过 R_1。”

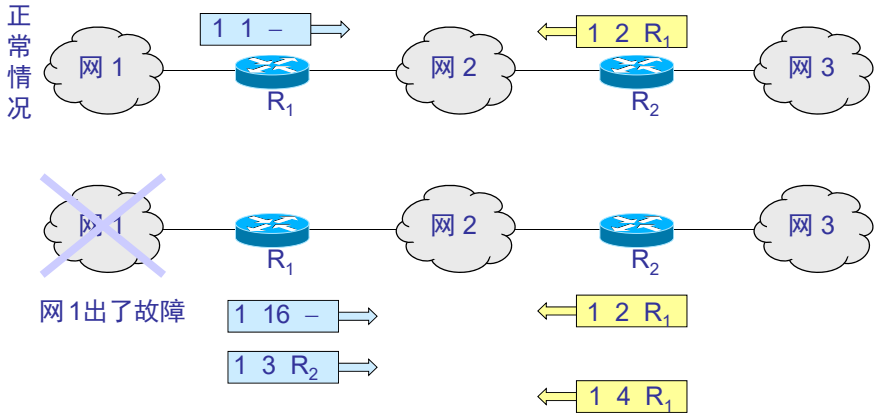


R_1 说：“我到网 1 的距离是 16 （表示无法到达），是直接交付。”

但 R_2 在收到 R_1 的更新报文之前，还发送原来的报文，因为这时 R_2 并不知道 R_1 出了故障。

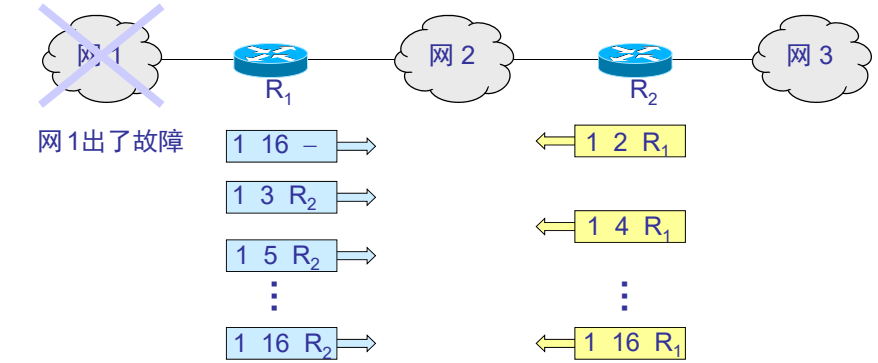


R_1 收到 R_2 的更新报文后，误认为可经过 R_2 到达网1，于是更新自己的路由表，说：“我到网 1 的距离是 3，下一跳经过 R_2”。然后将此更新信息发送给 R_2。



R_2 以后又更新自己的路由表为“1，4，R_1”，表明“我到网 1 距离是 4，下一跳经过 R_1”。

这就是好消息传播得快，而坏消息传播得慢。网络出故障的传播时间往往需要较长的时间(例如数分钟)。这是 RIP 的一个主要缺点。



这样不断更新下去，直到 R_1 和 R_2 到网 1 的距离都增大到 16 时，R_1 和 R_2 才知道网 1 是不可达的。

Link-State Algorithms

Link-State Algorithm (LSA)

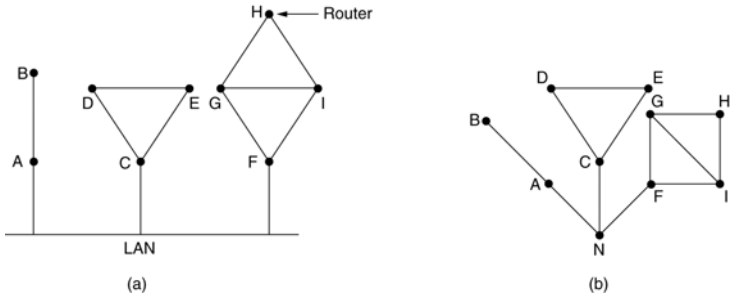
- Developed as a result of DV's looping and non-termination problems.
- Two components:
  - Topology map distribution
  - Local shortest path computation
- Each router runs a local shortest-path algorithm (Dijkstra's) using the topology stored locally.
- Flooding is used to replicate the topology map at every router.
- Each router is responsible for reporting the state of outgoing links to the rest of the network.

Link State Routing

Each router must do the following:

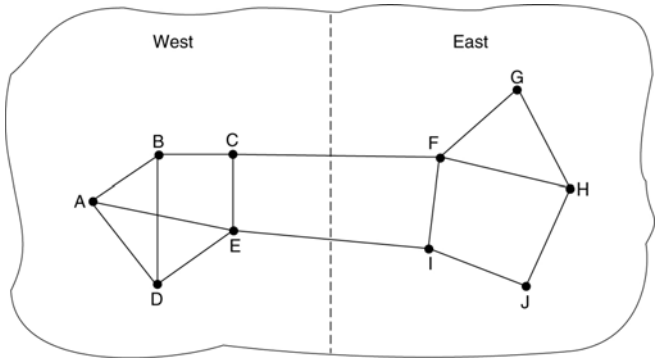
- Discover its neighbors, learn their network address.
- Measure the delay or cost to each of its neighbors.
- Construct a packet telling all it has just learned.
- Send this packet to all other routers.
- Compute the shortest path to every other router.

Learning about the Neighbors



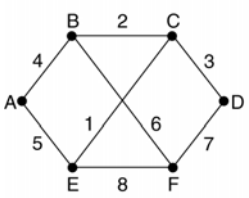
(a) Nine routers and a LAN. (b) A graph model of (a).

Measuring Line Cost



A subnet in which the East and West parts are connected by two lines.

Building Link State Packets



(a)

Link	State			Packets	
A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

(b)

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

Distributing the Link State Packets

Source	Seq.	Age	Send flags A	Send flags C	Send flags F	ACK flags A	ACK flags C	ACK flags F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

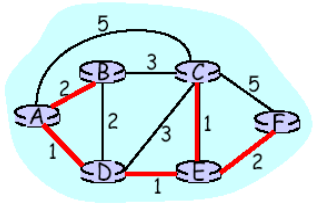
The packet buffer for router B in the previous slide (Fig. 5-13).

Dijkstra's Algorithm

```
1  Initialization:
2    N = {A}
3    for all nodes v
4      if v adjacent to A
5        then D(v) = c(A,v)
6        else D(v) = infinity
7
8  Loop
9    find w not in N such that D(w) is a minimum
10   add w to N
11   update D(v) for all v adjacent to w and not in N:
12      D(v) = min( D(v), D(w) + c(w,v) )
13   /* new cost to v is either old cost to v or known
14    shortest path cost to w plus cost from w to v */
15 until all nodes in N
```

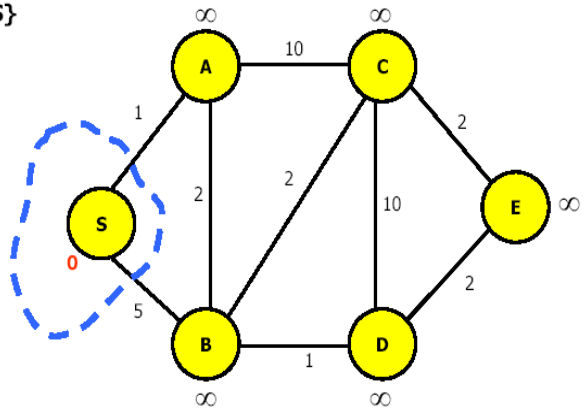
Dijkstra's algorithm: example

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
→ 0	A	2,A	5,A	1,A	infinity	infinity
→ 1	AD	2,A	4,D		2,D	infinity
→ 2	ADE	2,A	3,E			4,E
→ 3	ADEB		3,E			4,E
→ 4	ADEBC					4,E
5	ADEBCF					



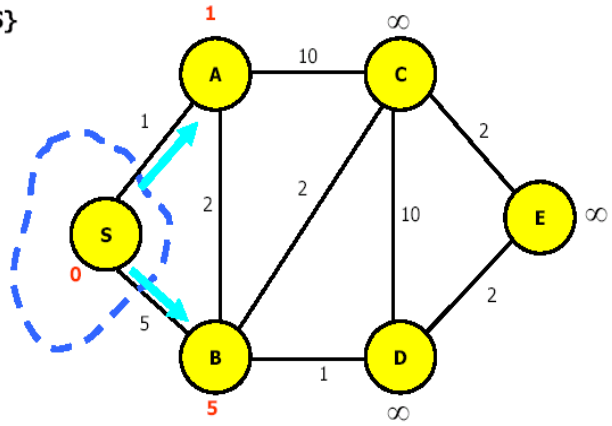
Dijkstra's algorithm: graphics example

N ={S}



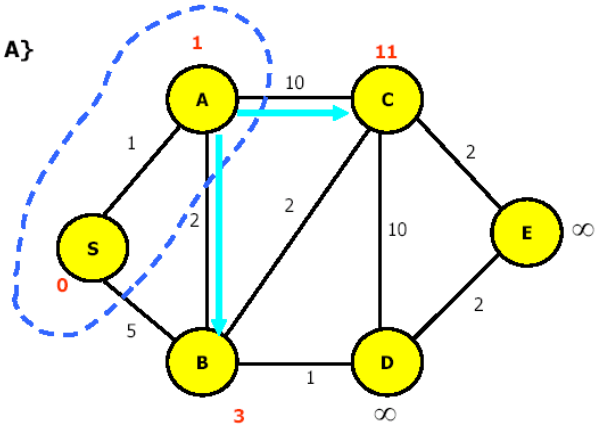
Dijkstra's algorithm: graphics example

N ={S}



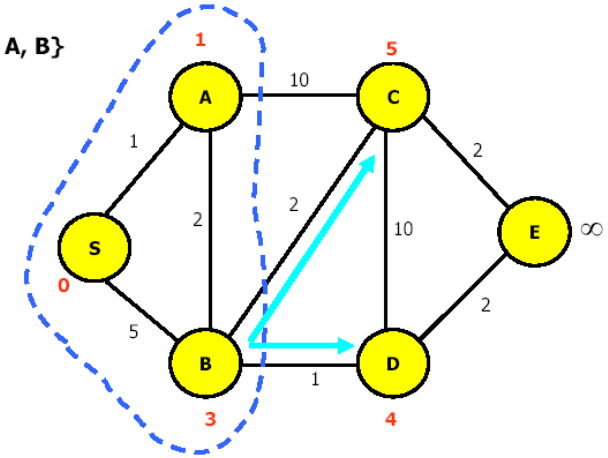
Dijkstra's algorithm: graphics example

N ={S, A}



Dijkstra's algorithm: graphics example

N ={S, A, B}



Dijkstra's algorithm: graphics example

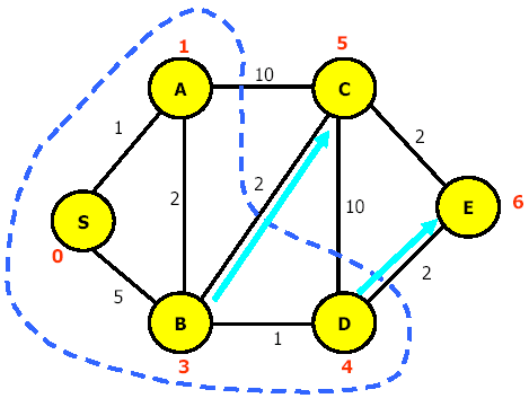
N ={S, A, B, D}

Labels do not change as we continue to expand set N

N ={S, A, B, D, C}

N ={S, A, B, D, C, E}

Stop after covering E since all nodes are covered by set N.



Note that iteration is on the **next node that can be covered with the next shortest path**; hence **complete topology must be known by router.**

Flooding of Link States

Information Stored at Routers:

- Each router maintains all the nodes and all the links in the network in a topology graph.
- Each link in the graph has a cost, a sequence number, and an age.

Flooding of Link States

Information Exchanged:

- Each router is responsible for broadcasting the latest state of each adjacent outgoing link periodically.
- The router sends a link state packet (LSP) update to report changes on an adjacent outgoing link.
- A sequence number is used to identify the latest LSP.
- An LSP also specified the age of the LSP, and the age of an LSP is decremented each time it is forwarded and while it is in storage.

We assume that LSPs are exchanged reliably between any two routers and that a router knows who its neighbors are!

Flooding of LSPs

Flooding Mechanism consists of three rules:

- **Rule 1: Deleting old LSPs**
  - **Discard old LSPs locally:**
    A router discards an LSP in its topology graph when it reaches a maximum age.
  - **Refresh own LSPs:**
    A router transmits periodically an LSP for each of its outgoing links, and assigns an age and the highest sequence number to the LSP.
  - **Handle of sequence numbers in LSPs:**
    The router originating an LSP is the only one that can change the sequence number of the LSP.

Flooding of LSPs

- **Rule 2: Propagating LSPs**
  - **Forward valid LSPs:**
    A router that receives a more recent LSP with a valid age from a given neighbor propagates it to all its *other* neighbors.
  - **Correct neighbor that reported old data:**
    A router that receives an outdated LSP from a neighbor *discards* the LSP received and sends its more recent LSP to the neighbor.
  - **Propagate "resets" (of link)**
    A router that receives a *more recent LSP with a zero age* propagates the LSP to all its other neighbors if the link is in its topology graph and deletes the link from its topology graph; else, it ignores the LSP.

Flooding of LSPs

- **Rule 3: Handling Topology Changes**
  - **Make sure that neighbors have the same topology map:**
    A router that detects a new neighbor sends its topology graph to that neighbor.
  - **Cope with partitions and reboots:**
    A router that hears a more recent LSP from a neighbor for one of its outgoing links creates a new LSP with a higher sequence number and sends it to all its neighbors.

Examples of LSA

- New routing algorithm of ARPANET
- OSPF (open shortest path first)
- IS-IS (intermediate system to intermediate system)

- The limitation with LSA is that it incurs substantial communication and processing overhead (to flood link states and compute shortest paths after that).

Comparison of LS and DV algorithms

Robustness: what happens if router malfunctions?

Link-State:

- node can advertise incorrect *link* cost
- each node computes only its *own* table

Distance-Vector:

- DV node can advertise incorrect *path* cost
- each node's table used by others
  - error propagate thru network

内部网关协议 OSPF

(Open Shortest Path First)

OSPF 协议的基本特点

- “开放”表明 OSPF 协议不是受某一家厂商控制，而是公开发表的。
- “最短路径优先”是因为使用了 Dijkstra 提出的最短路径算法SPF
- OSPF 只是一个协议的名字，它并不表示其他的路由选择协议不是“最短路径优先”。
- 是分布式的链路状态协议。

三个要点

- 向本自治系统中所有路由器发送信息，这里使用的方法是洪泛法。
- 发送的信息就是与本路由器相邻的所有路由器的链路状态，但这只是路由器所知道的部分信息。
  - “链路状态”就是说明本路由器都和哪些路由器相邻，以及该链路的“度量”(metric)。
- 只有当链路状态发生变化时，路由器才用洪泛法向所有路由器发送此信息。

链路状态数据库

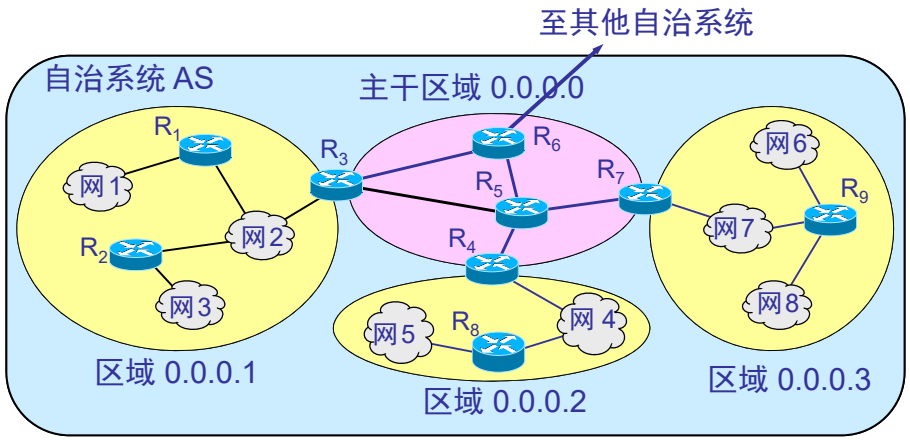
(link-state database)

- 由于各路由器之间频繁地交换链路状态信息，因此所有的路由器最终都能建立一个链路状态数据库。
- 这个数据库实际上就是全网的拓扑结构图，它在全网范围内是一致的（这称为链路状态数据库的同步）。
- OSPF 的链路状态数据库能较快地进行更新，使各个路由器能及时更新其路由表。OSPF 的更新过程收敛得快是其重要优点。

OSPF 的区域(area)

- 为了使 OSPF 能够用于规模很大的网络，OSPF 将一个自治系统再划分为若干个更小的范围，叫作区域。
- 每一个区域都有一个 32 位的区域标识符（用点分十进制表示）。
- 区域也不能太大，在一个区域内的路由器最好不超过 200 个。

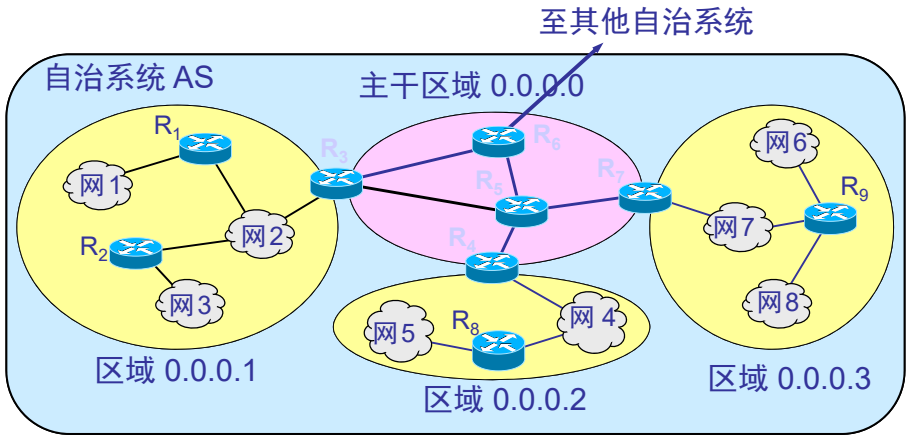
OSPF 划分为两种不同的区域



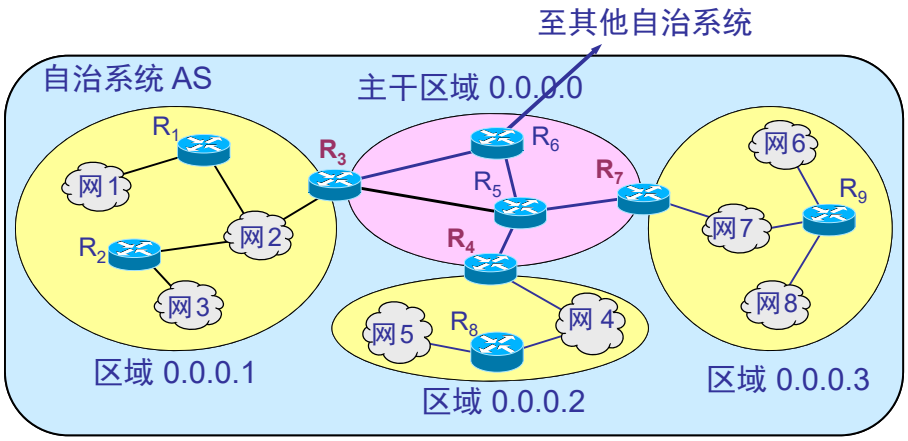
划分区域

- 划分区域的好处就是将利用洪泛法交换链路状态信息的范围局限于每一个区域而不是整个的自治系统，这就减少了整个网络上的通信量。
- 在一个区域内部的路由器只知道本区域的完整网络拓扑，而不知道其他区域的网络拓扑的情况。
- OSPF 使用层次结构的区域划分。在上层的区域叫作主干区域(backbone area)。主干区域的标识符规定为0.0.0.0。主干区域的作用是用来连通其他在下层的区域。

主干路由器



区域边界路由器



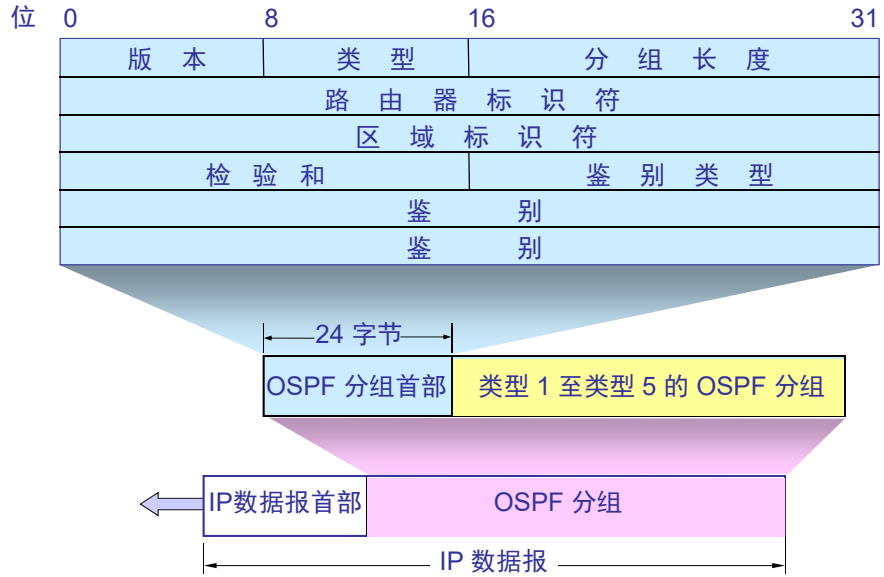
OSPF 直接用 IP 数据报传送

- OSPF 不用 UDP 而是直接用 IP 数据报传送。
- OSPF 构成的数据报很短。这样做可减少路由信息的通信量。
- 数据报很短的另一好处是可以不必将长的数据报分片传送。分片传送的数据报只要丢失一个，就无法组装成原来的数据报，而整个数据报就必须重传。

OSPF 的其他特点

- OSPF 对不同的链路可根据 IP 分组的不同服务类型 TOS 而设置成不同的代价。因此，OSPF 对于不同类型的业务可计算出不同的路由。
- 如果到同一个目的网络有多条相同代价的路径，那么可以将通信量分配给这几条路径。这叫作多路径间的负载平衡。
- 所有在 OSPF 路由器之间交换的分组都具有鉴别的功能。
- 支持可变长度的子网划分和无分类编址 CIDR。
- 每一个链路状态都带上一个 32 位的序号，序号越大状态就越新。

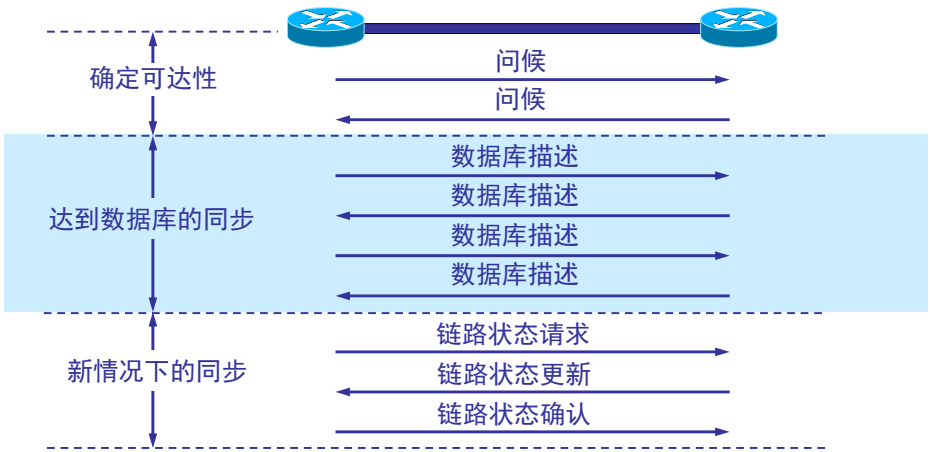
OSPF 分组



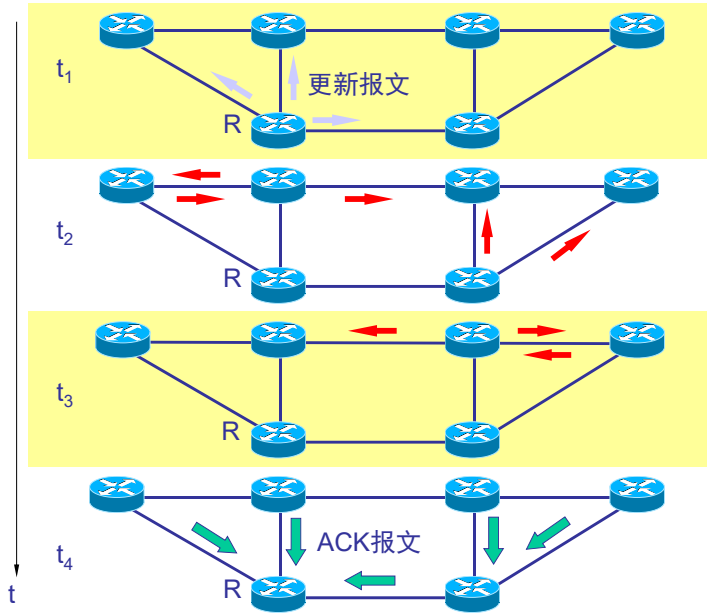
2. OSPF 的五种分组类型

- 类型1，问候(Hello)分组。
- 类型2，数据库描述(Database Description)分组。
- 类型3，链路状态请求(Link State Request)分组。
- 类型4，链路状态更新(Link State Update)分组，用洪泛法对全网更新链路状态。
- 类型5，链路状态确认(Link State Acknowledgment)分组。

OSPF的基本操作



OSPF 使用的是可靠的洪泛法



OSPF 的其他特点

- OSPF 还规定每隔一段时间，如 30 分钟，要刷新一次数据库中的链路状态。
- 由于一个路由器的链路状态只涉及到与相邻路由器的连通状态，因而与整个互联网的规模并无直接关系。因此当互联网规模很大时，OSPF 协议要比距离向量协议 RIP 好得多。
- OSPF 没有“坏消息传播得慢”的问题，据统计，其响应网络变化的时间小于 100 ms。

指定的路由器

(designated router)

- 多点接入的局域网采用了指定的路由器的方法，使广播的信息量大大减少。
- 指定的路由器代表该局域网上所有的链路向连接到该网络上的各路由器发送状态信息。

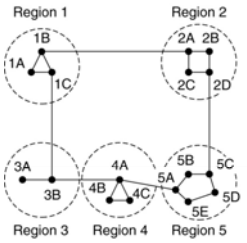
Hierarchical Routing

Our routing study thus far - idealization
- all routers identical
- network "flat"
- … *not* true in practice

scale: with 50 million destinations:
- can't store all dest's in routing tables!
- routing table exchange would swamp links!

administrative autonomy
- internet = network of networks
- each network admin may want to control routing in its own network

Hierarchical Routing



Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

(a)

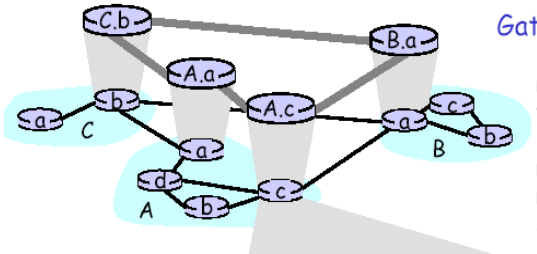
Hierarchical routing.

Hierarchical Routing

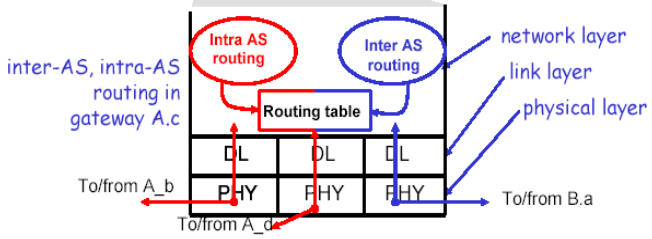
- aggregate routers into regions, "autonomous systems" (AS)
- routers in same AS run same routing protocol
  - "intra-AS" routing protocol
  - routers in different AS can run different intra-AS routing protocol

gateway routers
- special routers in AS
- run intra-AS routing protocol with all other routers in AS
- *also* responsible for routing to destinations outside AS
  - run *inter-AS routing* protocol with other gateway routers

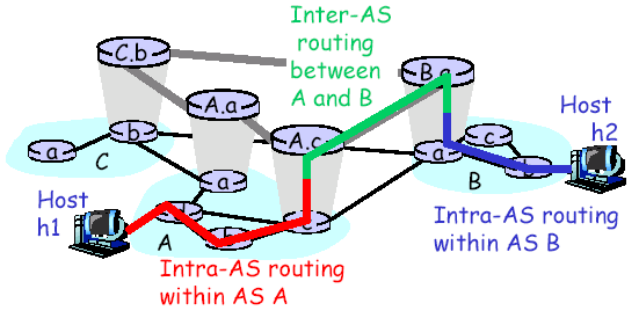
Intra-AS and Inter-AS routing



Gateways:
- perform inter-AS routing amongst themselves
- perform intra-AS routers with other routers in their AS



Intra-AS and Inter-AS routing

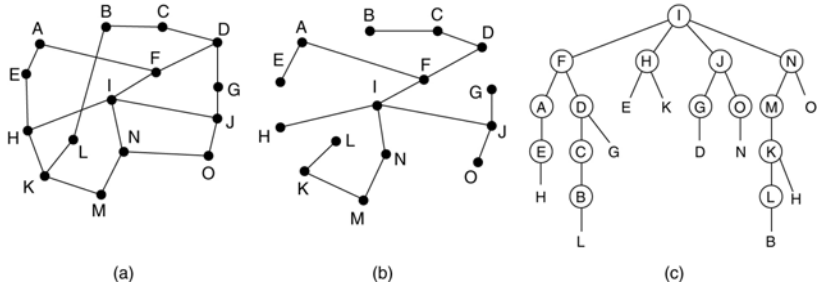


- We'll examine specific inter-AS and intra-AS Internet routing protocols shortly

Other routing algorithms…

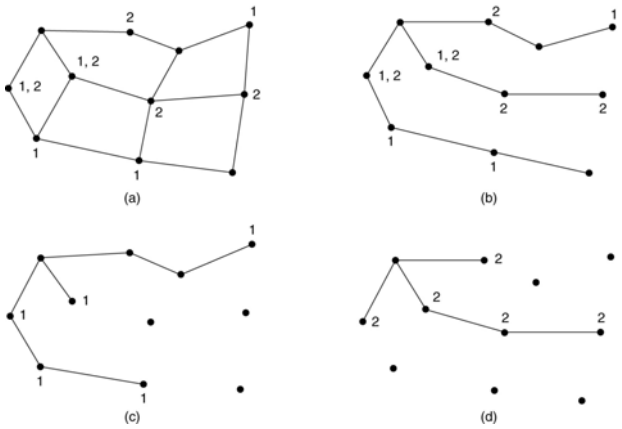
- Flooding
- Flow-based Routing
- Broadcast Routing
- Multicast Routing
- Ad hoc routing algorithms
- Etc…

Broadcast Routing



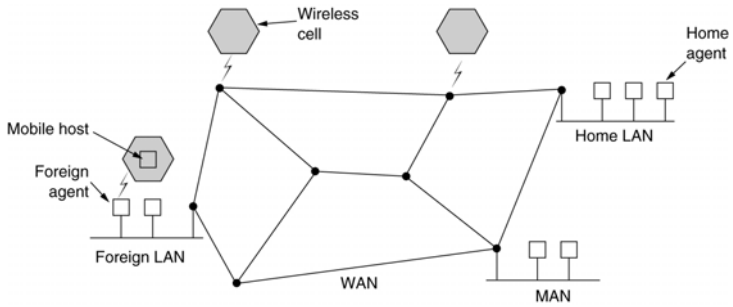
Reverse path forwarding. (a) A subnet. (b) a Sink tree. (c) The tree built by reverse path forwarding.

Multicast Routing



(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.

Routing for Mobile Hosts



A WAN to which LANs, MANs, and wireless cells are attached.

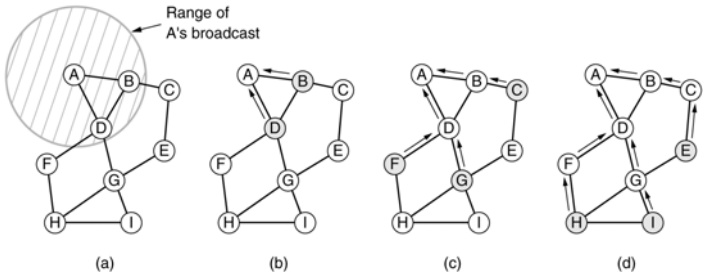
Routing for Mobile Hosts (2)

Packet routing for mobile users.

Routing in Ad Hoc Networks

- Possibilities when the routers are mobile:
- Military vehicles on battlefield.
  - No infrastructure.
- A fleet of ships at sea.
  - All moving all the time
- Emergency works at earthquake .
  - The infrastructure destroyed.
- A gathering of people with notebook computers.
  - In an area lacking 802.11.

Route Discovery



- Range of A's broadcast.
- After B and D have received A's broadcast.
- After C, F, and G have received A's broadcast.
- After E, H, and I have received A's broadcast.
- Shaded nodes are new recipients. Arrows show possible reverse routes.

Route Discovery (2)

Source address	Request ID	Destination address	Source sequence #	Dest. sequence #	Hop count

Format of a ROUTE REQUEST packet.

Route Discovery (3)

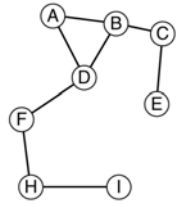
Source address	Destination address	Destination sequence #	Hop count	Lifetime

Format of a ROUTE REPLY packet.

Route Maintenance

Dest.	Next hop	Distance	Active neighbors	Other fields
A	A	1	F, G	
B	B	1	F, G	
C	B	2	F	
E	G	2		
F	F	1	A, B	
G	G	1	A, B	
H	F	2	A, B	
I	G	2	A, B	

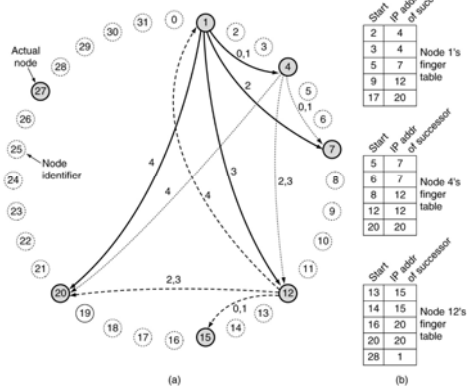
(a)



(b)

- D's routing table before G goes down.
- The graph after G has gone down.

Node Lookup in Peer-to-Peer Networks



- (a) A set of 32 node identifiers arranged in a circle. The shaded ones correspond to actual machines. The arcs show the fingers from nodes 1, 4, and 12. The labels on the arcs are the table indices.
- (b) Examples of the finger tables.

Congestion Control

Why Congestion Control is necessary

- Point to Point Congest control is applied through the whole network?
- Point to Point Congest control is enough?
- Every link has the same capacity?
  - ➔ why congestion?
- Even if all the above are satisfied, suppose multi senders (source) send data to the same receiver (destination)…

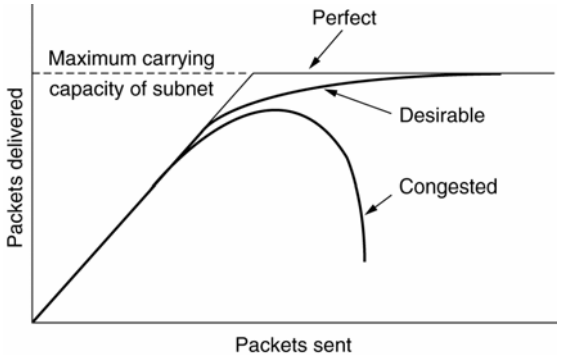
Open loop and close loop control

- Open loop control
- Closed loop control

Congestion Control Algorithms

- General Principles of Congestion Control
- Congestion Prevention Policies
- Congestion Control in Virtual-Circuit Subnets
- Congestion Control in Datagram Subnets
- Load Shedding
- Jitter Control

Congestion



When too much traffic is offered, congestion sets in and performance degrades sharply.

General Principles of Congestion Control

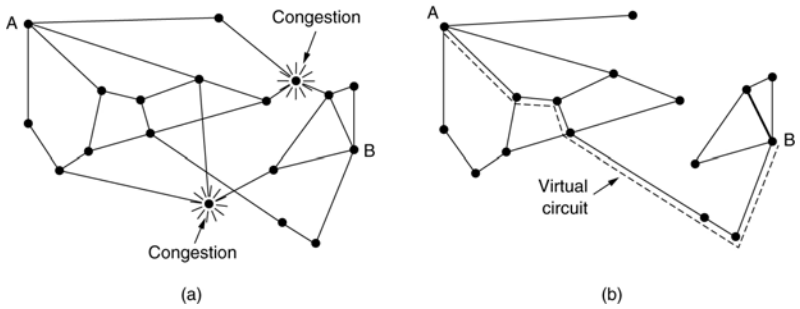
- Monitor the system .
  - detect when and where congestion occurs.
- Pass information to where action can be taken.
- Adjust system operation to correct the problem.

Congestion Prevention Policies

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

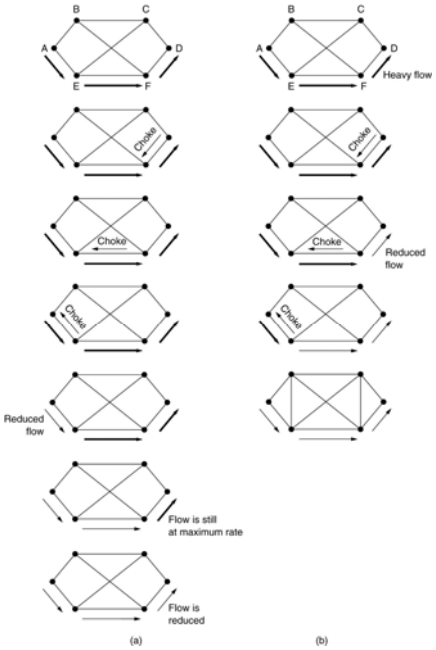
Policies that affect congestion.

Congestion Control in Virtual-Circuit Subnets



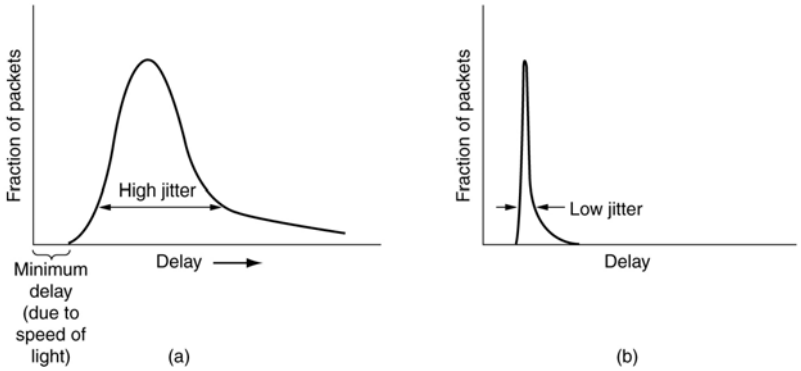
(a) A congested subnet. (b) A redrawn subnet, eliminates congestion and a virtual circuit from A to B.

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.

Jitter Control



(a) High jitter. (b) Low jitter.

Quality of Service

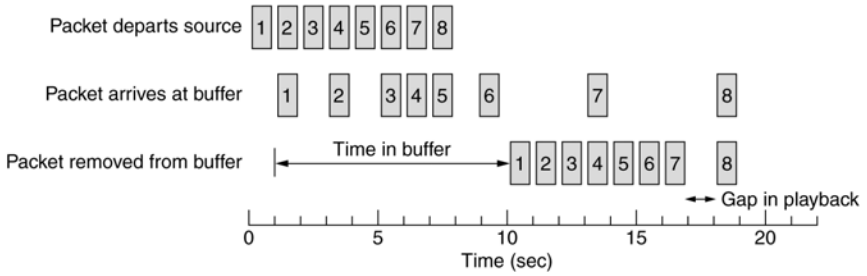
- Requirements
- Techniques for Achieving Good Quality of Service
- Integrated Services
- Differentiated Services
- Label Switching and MPLS

Requirements

Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High	High

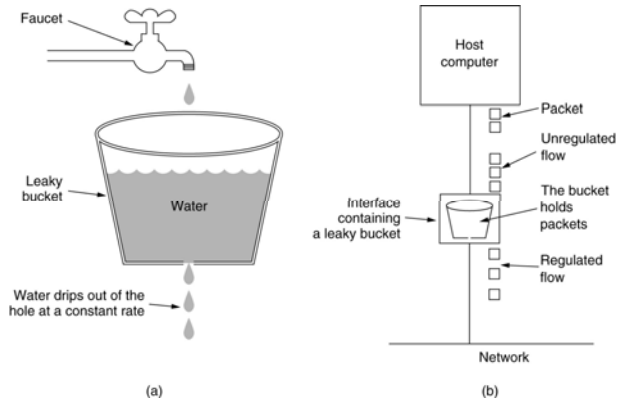
How stringent the quality-of-service requirements are.

Buffering



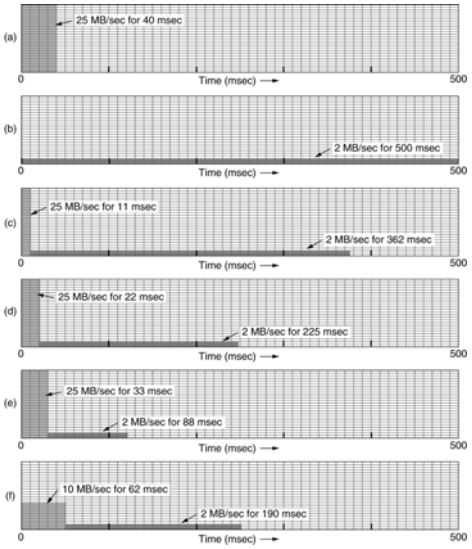
Smoothing the output stream by buffering packets.

The Leaky Bucket Algorithm



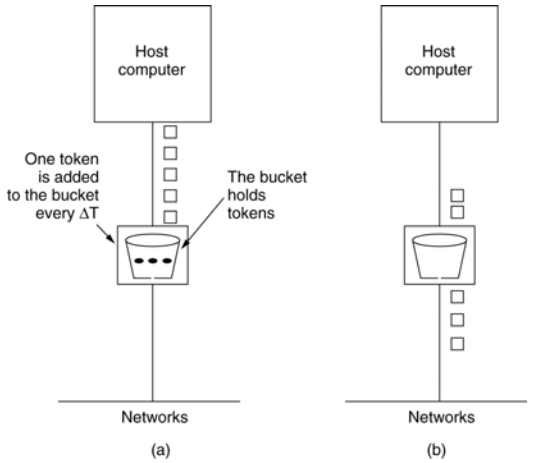
(a) A leaky bucket with water. (b) a leaky bucket with packets.

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.

The Token Bucket Algorithm



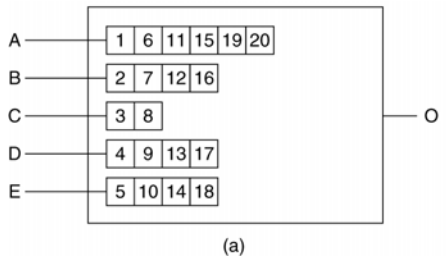
(a) Before. (b) After.

Admission Control

Parameter	Unit
Token bucket rate	Bytes/sec
Token bucket size	Bytes
Peak data rate	Bytes/sec
Minimum packet size	Bytes
Maximum packet size	Bytes

An example of flow specification.

Packet Scheduling

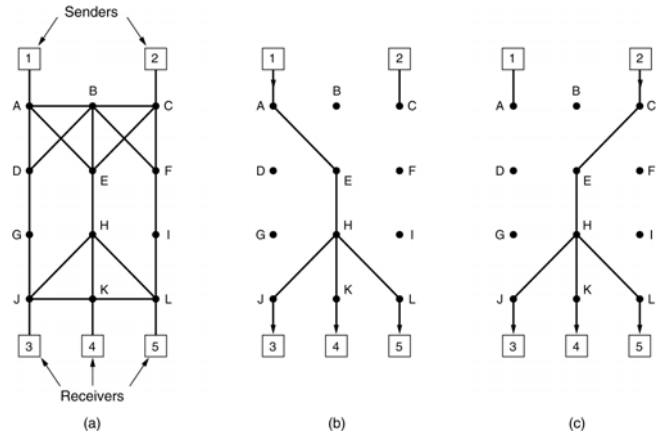


Packet	Finishing time
C	8
B	16
D	17
E	18
A	20

(b)

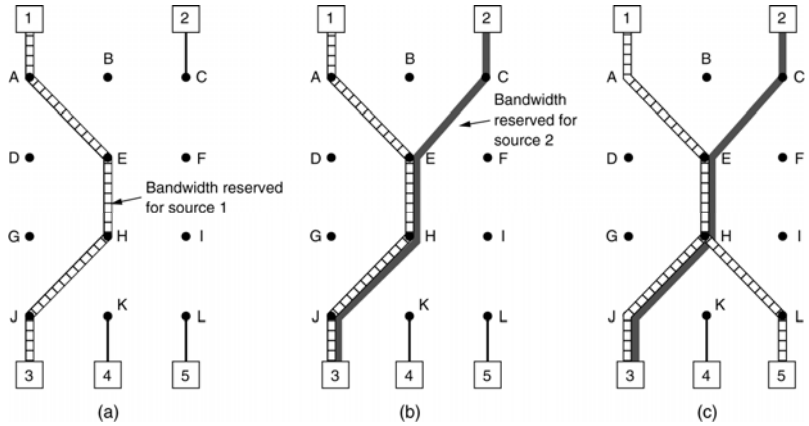
- (a) A router with five packets queued for line O.
- (b) Finishing times for the five packets.

RSVP-The ReSerVation Protocol



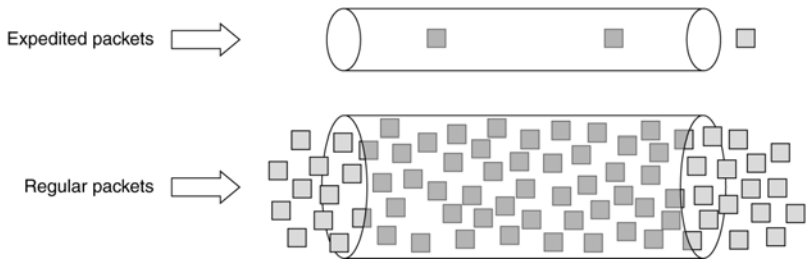
(a) A network, (b) The multicast spanning tree for host 1.
(c) The multicast spanning tree for host 2.

RSVP-The ReSerVation Protocol (2)



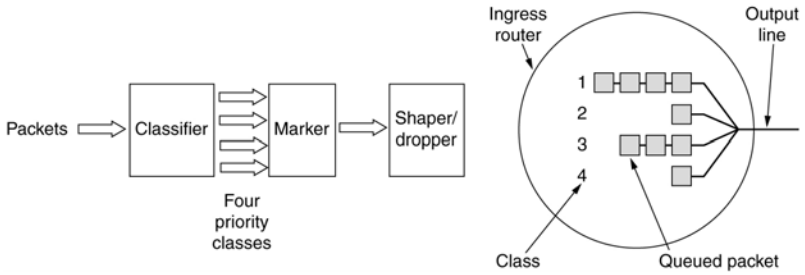
(a) Host 3 requests a channel to host 1. (b) Host 3 then requests a second channel, to host 2. (c) Host 5 requests a channel to host 1.

Expedited Forwarding



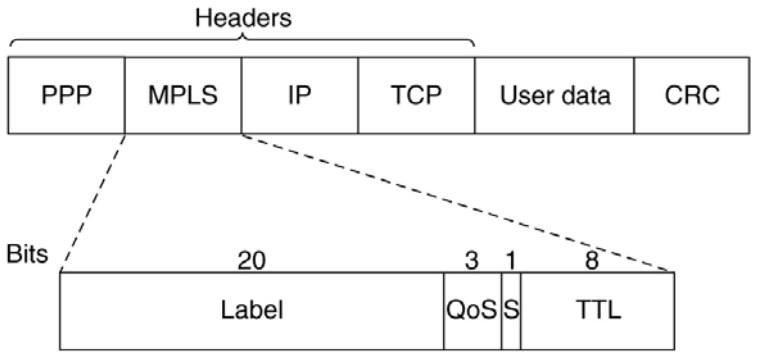
Expedited packets experience a traffic-free network.

Assured Forwarding



A possible implementation of the data flow for assured forwarding.

Label Switching and MPLS

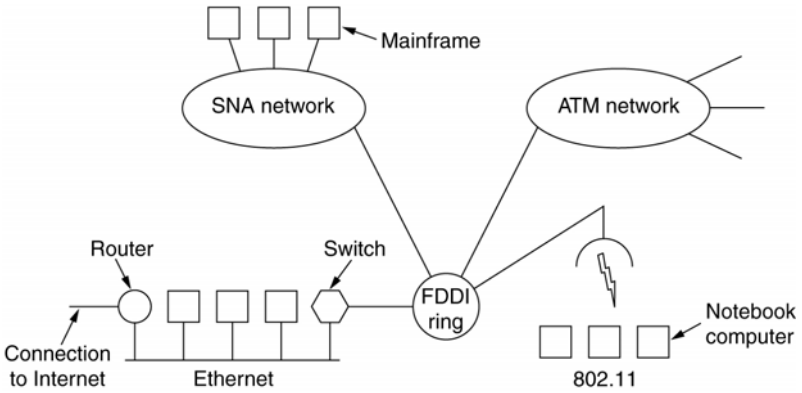


Transmitting a TCP segment using IP, MPLS, and PPP.

Internetworking

- How Networks Differ
- How Networks Can Be Connected
- Concatenated Virtual Circuits
- Connectionless Internetworking
- Tunneling
- Internetwork Routing
- Fragmentation

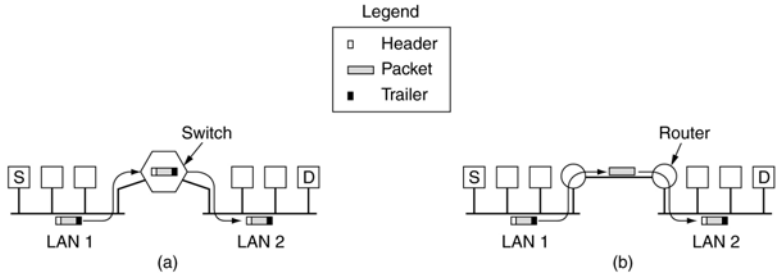
Connecting Networks



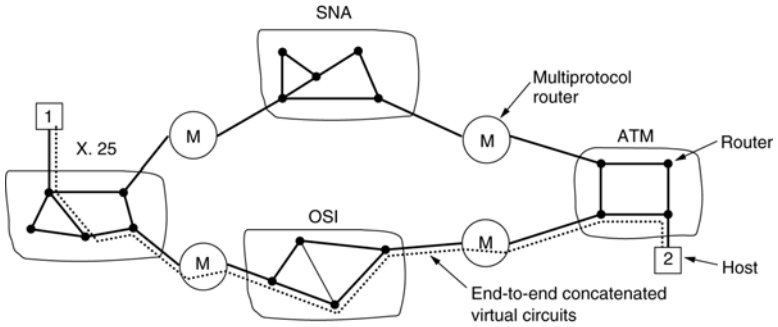
How Networks Differ

Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered, and unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

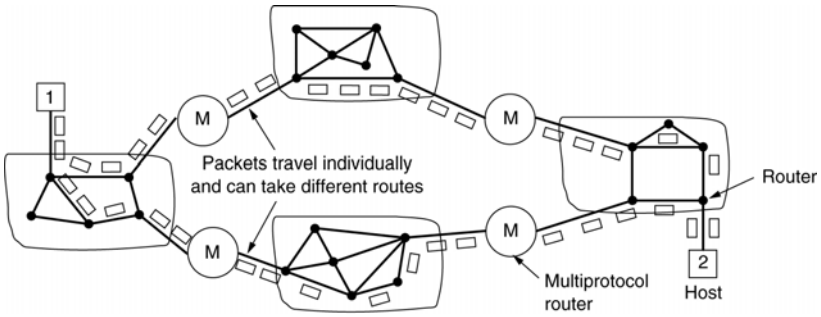
How Networks Can Be Connected



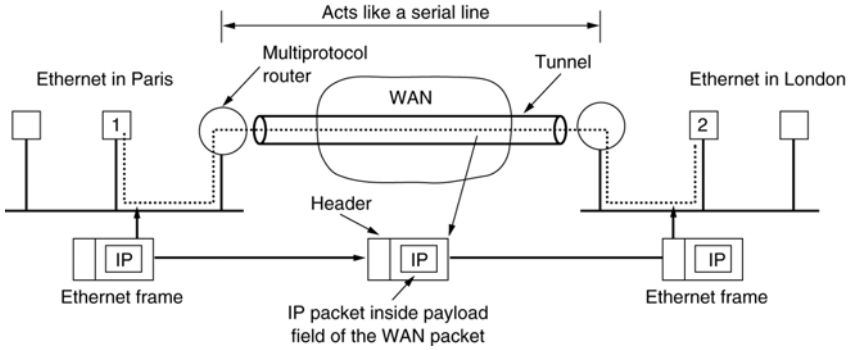
Concatenated Virtual Circuits



Connectionless Internetworking

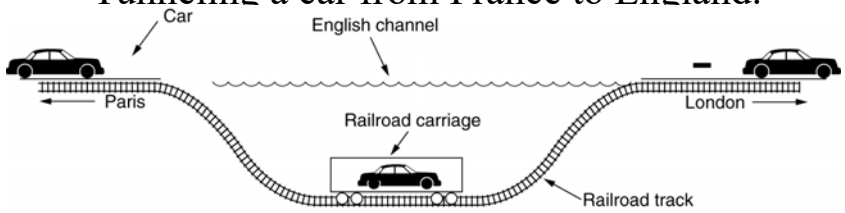


Tunneling

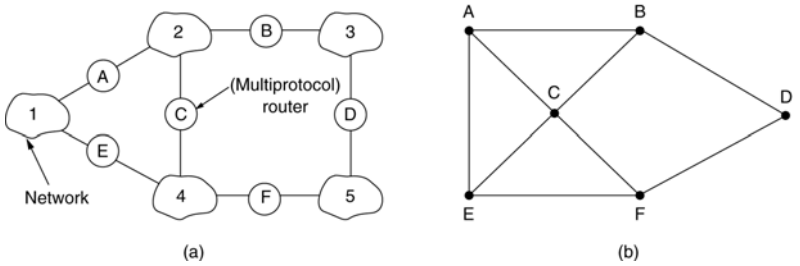


Tunneling (2)

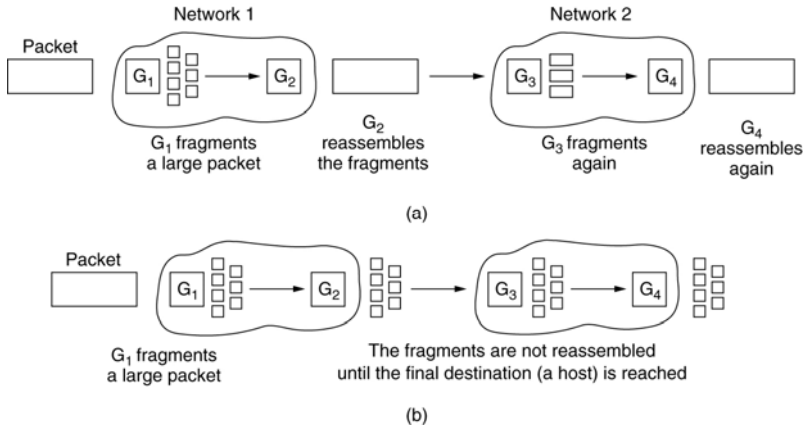
Tunneling a car from France to England.



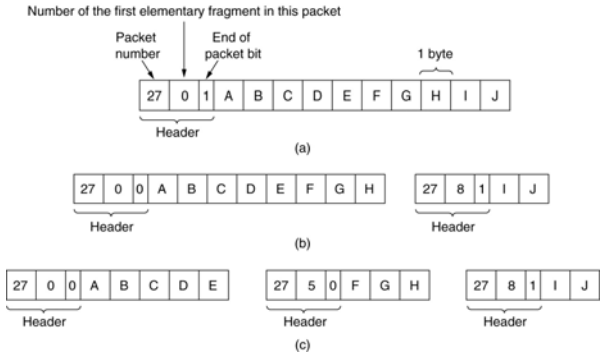
Internetwork Routing



Fragmentation



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.

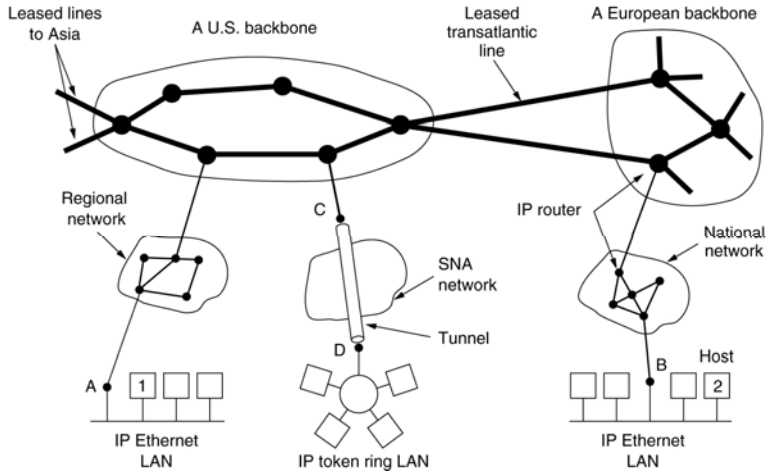
The Network Layer in the Internet

- The IP Protocol
- IP Addresses
- Internet Control Protocols
- OSPF – The Interior Gateway Routing Protocol
- BGP – The Exterior Gateway Routing Protocol
- Internet Multicasting
- Mobile IP
- IPv6

Design Principles for Internet

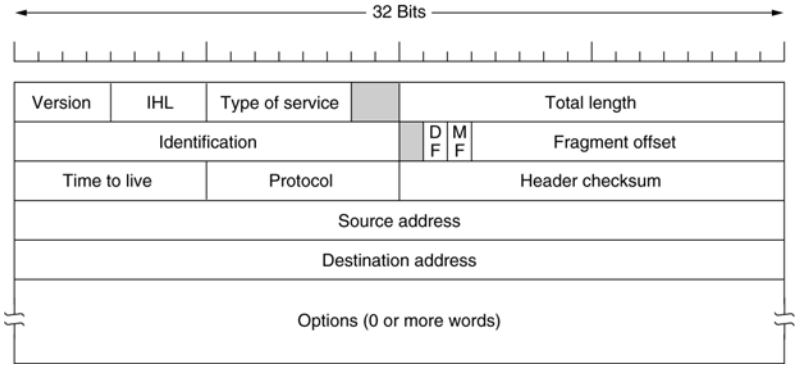
- Make sure it works.
- Keep it simple.
- Make clear choices.
- Exploit modularity.
- Expect heterogeneity.
- Avoid static options and parameters.
- Look for a good design; it need not be perfect.
- Be strict when sending and tolerant when receiving.
- Think about scalability.
- Consider performance and cost.

Collection of Subnetworks



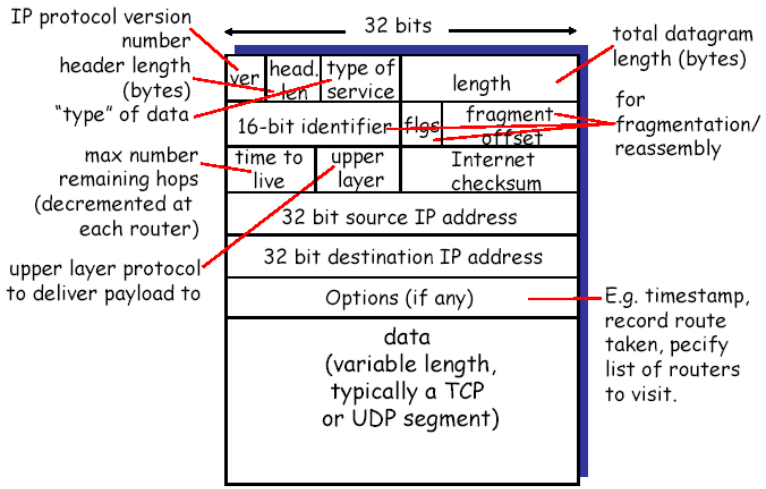
The Internet is an interconnected collection of many networks.

The IP Protocol



- IP packet Format etc...

IP datagram format



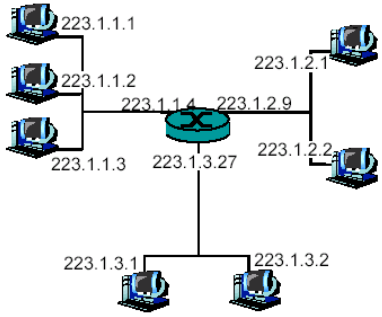
The IP Protocol (2)

Option	Description
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

Some of the IP options.

IP Addressing: introduction

- IP address: 32-bit identifier for host, router *interface*
- *interface:* connection between host, router and physical link
  - router's typically have multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with interface, not host, router

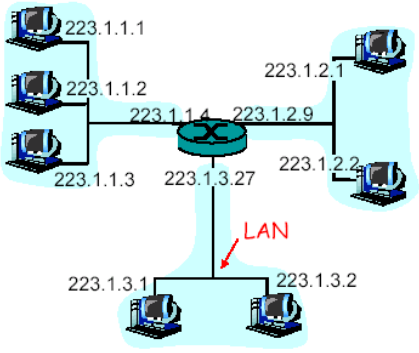


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

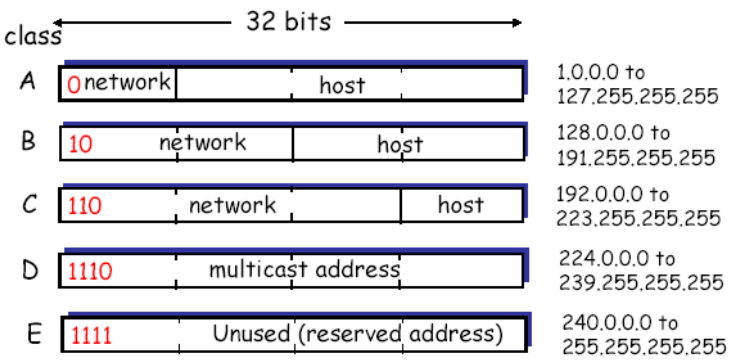
IP Addressing

- IP address:
  - network part (high order bits)
  - host part (low order bits)
- *What's a network ?* (from IP address perspective)
  - device interfaces with same network part of IP address
  - can physically reach each other without intervening router

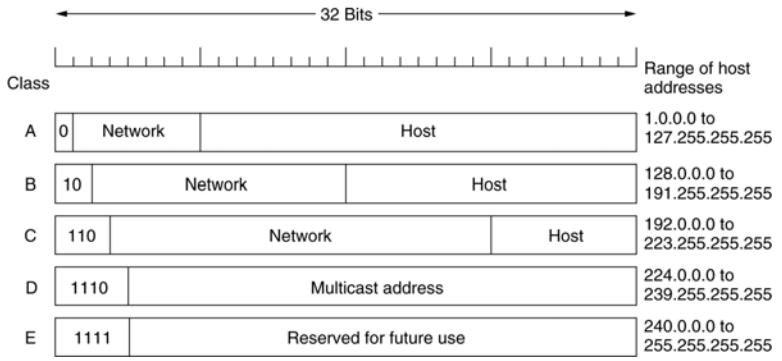


network consisting of 3 IP networks (for IP addresses starting with 223, first 24 bits are network address)

IP Addresses



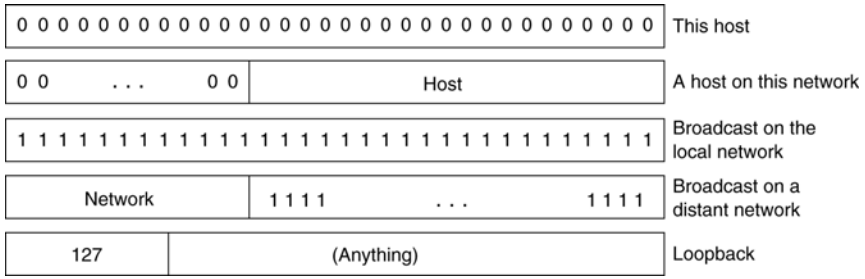
IP Addresses



IP Address Class Ranges

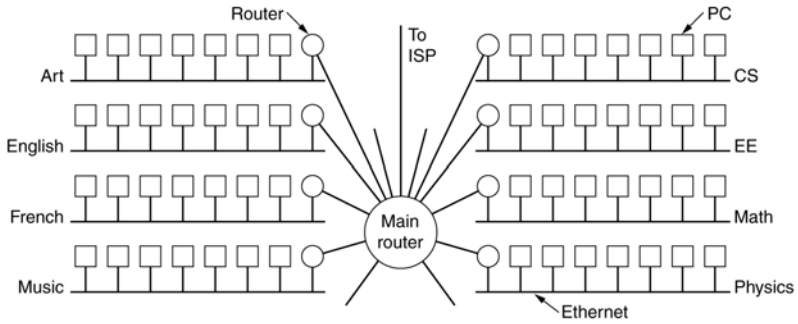
Class	Range
A	0.0.0.0 to 127.255.255.255
B	128.0.0.0 to 191.255.255.255
C	192.0.0.0 to 223.255.255.255
D	224.0.0.0 to 239.255.255.255
E	240.0.0.0 to 255.255.255.255

IP Addresses (2)



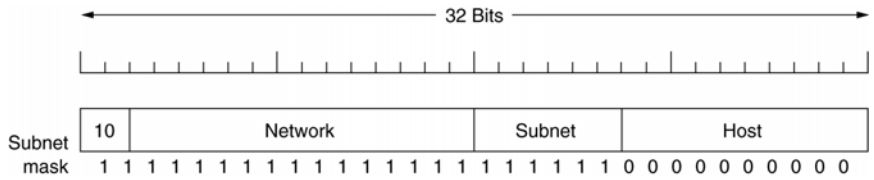
Special IP addresses.

Subnets



A campus network consisting of LANs for various departments.

Subnets (2)



A class B network subnetted into 64 subnets.

IP addresses: how to get one?

Network (network portion):

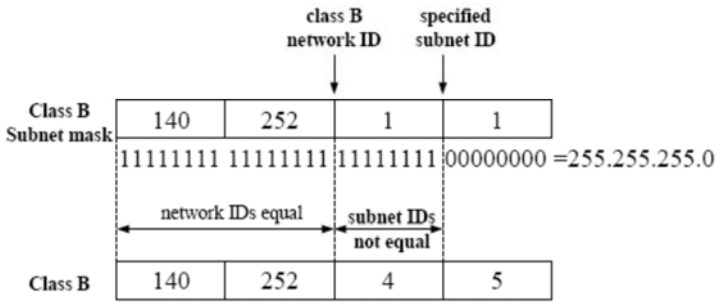
- get allocated portion of ISP's address space:

ISP's block	11001000 00010111 0001 0000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 0001000 0	00000000	200.23.16.0/23
Organization 1	11001000 00010111 0001001 0	00000000	200.23.18.0/23
Organization 2	11001000 00010111 0001010 0	00000000	200.23.20.0/23
...	.....	....	....
Organization 7	11001000 00010111 0001111 0	00000000	200.23.30.0/23

IP Address ---- Mask

- 202.119.22.5 mask 255.255.255.0
  first 3 bytes correspond to network ID
  the left to host ID
  11001010 01110111 00010110 00000101
  & 111 11 111 1 111 1111 1111 1111 00000000
  110 01 010 0111 0111 0001 0110 00000000
  (202) (119) (22) (0)
- 192.168.0.0 mask 255.255.255.0
- 221.2.2.0 mask 255.255.255.128

Using Subnet Masks_Example



Special IP Addresses

- **Host id "all 0s" is reserved to refer to a network Number**

  166.111.0.0 (mask 255.255.0.0),
  202.119.22.0 (mask 255.255.255.0)

- **Host id "all 1s" is reserved to broadcast to all hosts on a specific network**

  166.111.255.255 (mask 255.255.0.0),
  202.119.22.255 (mask 255.255.255.0)

Special IP Addresses (cont)

- **0.0.0.0** is reserved and means "unknown address". Normally use to boot diskless workstation
- **255.255.255.255** is reserved to broadcast to every host on the local network
- **127.x.x.x** means "this node" (local loopback).

  Messages sent to this address will never leave the
  local host. It's purpose is testing network software

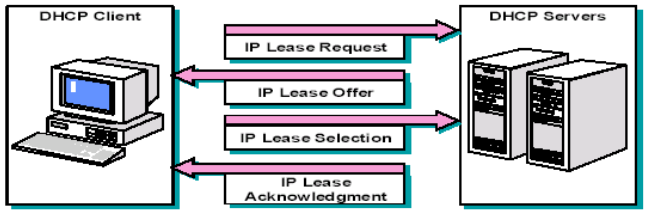
IP addresses: how to get one?

Hosts (host portion):

- hard-coded in system
- DHCP: Dynamic Host Configuration Protocol: dynamically get address: "plug-and-play"
  - host broadcasts "DHCP discover" msg
  - DHCP server responds with "DHCP offer" msg
  - host requests IP address: "DHCP request" msg
  - DHCP server sends address: "DHCP ack" msg

DHCP: Dynamic Host Configuration Protocol

Automatic assignment of IP numbers to network clients on boot-up using DHCP eliminates some of IP's administrative complexity



DHCP概述

- DHCP：Dynamic Host Configuration Protocol(RFC2131)，动态主机配置协议。动态配置TCP/IP协议栈参数(例如：子网掩码、缺省路由器、打印机地址、系统时间等等）。
- 动态配置协议的必要性：实现IP地址的合理分配和充分利用。为移动数据通信提供便利（如mobile IP)。

DHCP实例

- 经统计，有400个人在一个可容纳50个人的机房用笔记本电脑上机，这个学校分配有一个C类地址，或是有254个可用地址的B类地址的一个子网。显然，静态分配IP地址是不够用的，动态分配则最多使用了20%的地址空间。

协议评价

- DHCP适应了当前便携机的发展对动态配置地址机相关参数的要求，同时通过客户机、服务器的通用模型达到了网络地址等参数的集中式控制，兼顾了经济性和可扩展性，是当前流行的网络地址配置协议。

协议评价(续)

- DHCP是建立在UDP协议上的，由于UDP数据报可以通过路由器转发，可以实现多个物理网断共享DHCP服务器。
- 由于UDP数据包存在潜在的安全性问题，DHCP协议存在安全问题。例如，未授权的DHCP服务器可能会发送错误的配置信息。恶意的使用者可以利用这些漏洞。

CIDR – Classless InterDomain Routing

University	First address	Last address	How many	Written as
Cambridge	194.24.0.0	194.24.7.255	2048	194.24.0.0/21
Edinburgh	194.24.8.0	194.24.11.255	1024	194.24.8.0/22
(Available)	194.24.12.0	194.24.15.255	1024	194.24.12/22
Oxford	194.24.16.0	194.24.31.255	4096	194.24.16.0/20

A set of IP address assignments.

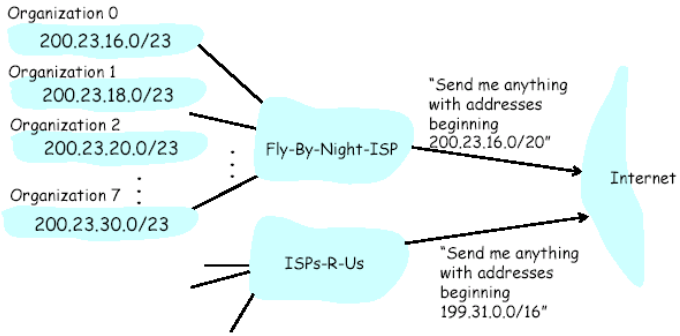
CIDR (Classless Inter-Domain Routing)

- classful addressing:
  - inefficient use of address space, address space exhaustion
  - e.g., class B net allocated enough addresses for 65K hosts, even if only 2K hosts in that network
- CIDR: Classless InterDomain Routing
  - network portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is # bits in network portion of address

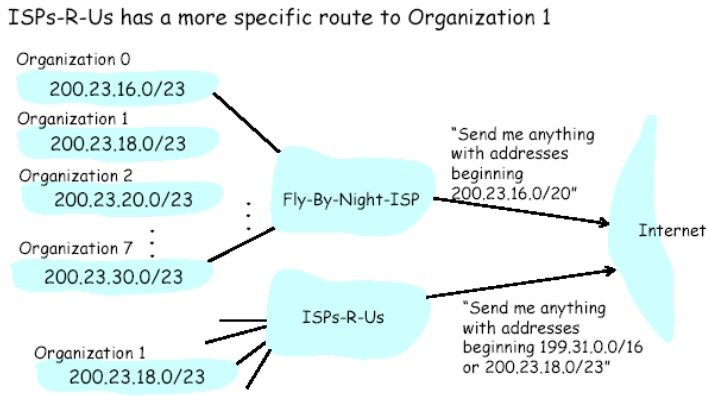
200.23.16.0/23

Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes



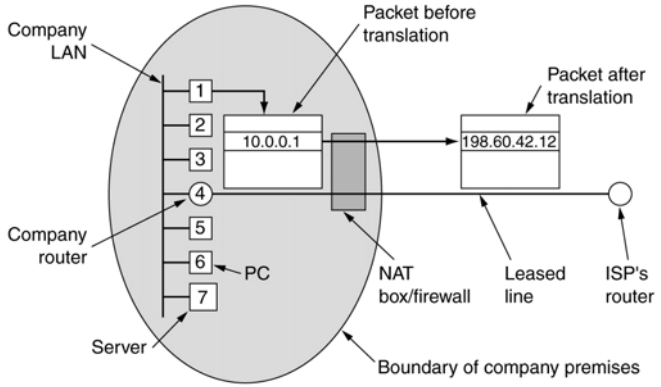
IP addressing: the last word...

Q: How does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes
- Website: www.icann.org

NAT – Network Address Translation



Placement and operation of a NAT box.

ICMP: Internet Control Message Protocol

- used by hosts, routers, gateways to communication network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

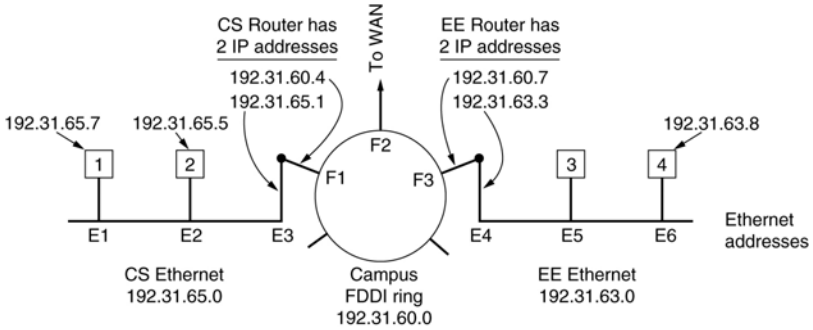
Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Internet Control Message Protocol

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

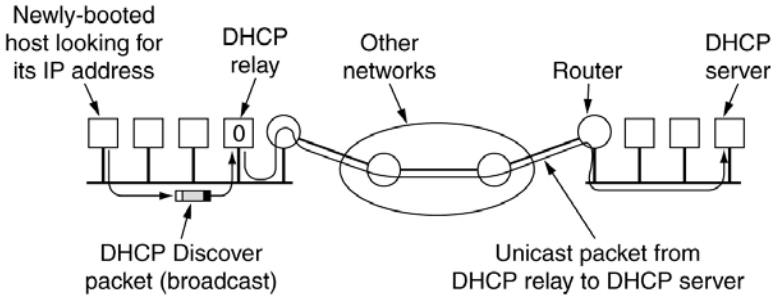
The principal ICMP message types.

ARP– The Address Resolution Protocol



Three interconnected /24 networks: two Ethernets and an FDDI ring.

Dynamic Host Configuration Protocol

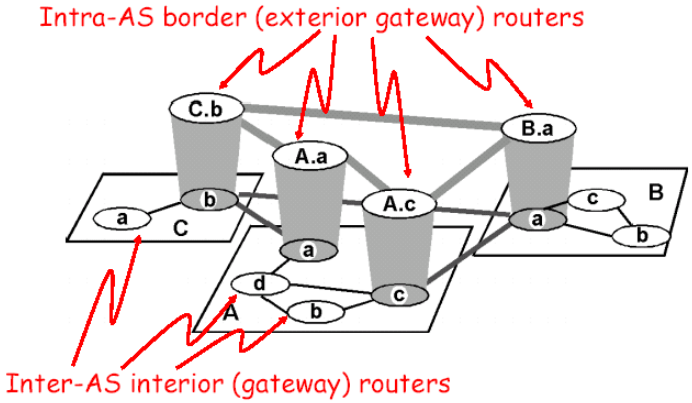


Operation of DHCP.

Routing in the Internet

- The Global Internet consists of Autonomous Systems (AS) interconnected with each other:
- Two-level routing:
  - **Intra-AS:** administrator is responsible for choice
  - **Inter-AS:** unique standard
- Network mask
  => Determine the same network
  route by network
- Routing Table
  - Each node maintains a set of triples
    `< SubnetNum, SubnetMask, NextHop>`

Internet AS Hierarchy



Basic: Getting a datagram from source to dest.

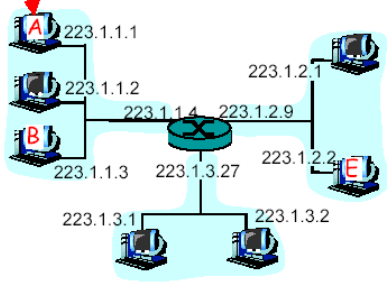
IP datagram:

misc fields	source IP addr	dest IP addr	data

- datagram remains unchanged, as it travels source to destination
- addr fields of interest here

A's View

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



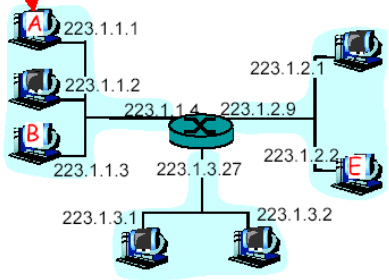
Basic: Getting a datagram from source to dest.

misc fields	223.1.1.1	223.1.1.3	data

Starting at A, given IP datagram addressed to B:

- look up net. address of B
- find B is on same net. as A
- link layer will send datagram directly to B inside link-layer frame
  - B and A are directly connected

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



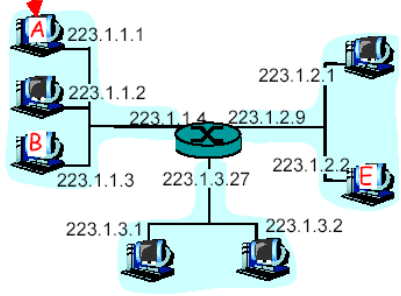
Basic: Getting a datagram from source to dest.

misc fields	223.1.1.1	223.1.2.3	data

Starting at A, dest. E:

- look up network address of E
- E on *different* network
  - A, E not directly attached
- routing table: next hop router to E is 223.1.1.4
- link layer sends datagram to router 223.1.1.4 inside link-layer frame
- datagram arrives at 223.1.1.4
- continued.....

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



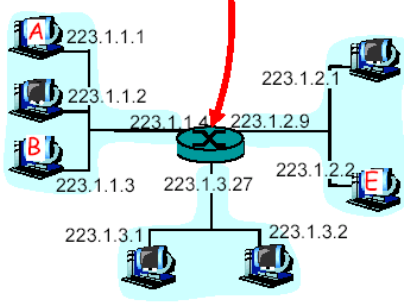
Basic: Getting a datagram from source to dest.

misc fields	223.1.1.1	223.1.2.3	data

Arriving at 223.1.4, destined for 223.1.2.2

- look up network address of E
- E on *same* network as router's interface 223.1.2.9
  - router, E directly attached
- link layer sends datagram to 223.1.2.2 inside link-layer frame via interface 223.1.2.9
- datagram arrives at 223.1.2.2!!! (hooray!)

Dest. network	next router	Nhops	interface
223.1.1	-	1	223.1.1.4
223.1.2	-	1	223.1.2.9
223.1.3	-	1	223.1.3.27



Forwarding Algorithm

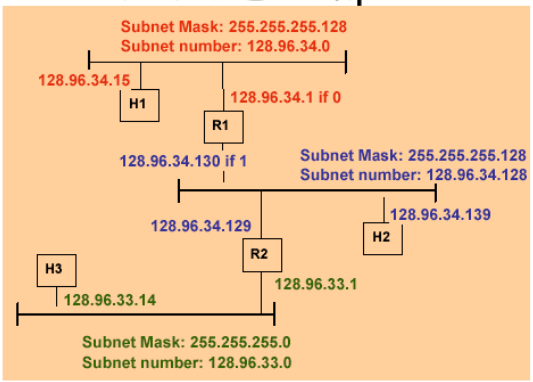
```
D = destination IP address
Routed = false

for each entry < SubnetNum, SubnetMask, NextHop>
{
   D1 = SubnetMask & D

   if D1 = SubnetNum
   {
      routed = true
      if NextHop is an interface
         deliver datagram directly to destination
      else
         deliver datagram to NextHop (a router)
      break /*exit for.. Loop */

   }
}
If routed = false
  SendICMPMessage("No route to host");
```

Subnet Example



Forwarding table at router R1

Subnet Number	Subnet Mask	Next Hop
128.96.34.0	255.255.255.128	interface 0
128.96.34.128	255.255.255.128	interface 1
0.0.0.0	0.0.0.0	R2

Intra-AS Routing

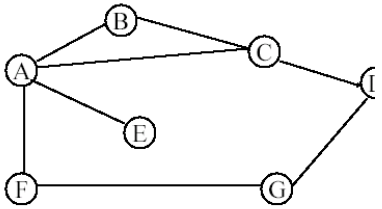
- Also known as Interior Gateway Protocols (IGP)
- Most common IGPs:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco propr.)

RIP (Routing Information Protocol)

- Distance vector algorithm
- Each node maintains a set of triples:
  - (Destination, Cost, NextHop)
- Each node sends updates to (and receives updates from) its directly connected neighbors via Response Message (also called **advertisement**)
  - periodically (on the order of several seconds)
  - whenever its table changes (called triggered update)
- Distance metric: # of hops (max = 15 hops)
- Each advertisement: route to up to 25 destination nets
- Details in Distant-Vector algorithm

  ... Update routing table when short path or came from next hop

Example



Routing table at node B

Destination	Cost	Next Hop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	3	A

B receives (E,1)(F,1) from A
B receives (D,1) from C
B receives (G,2) from A

OSPF (Link State)

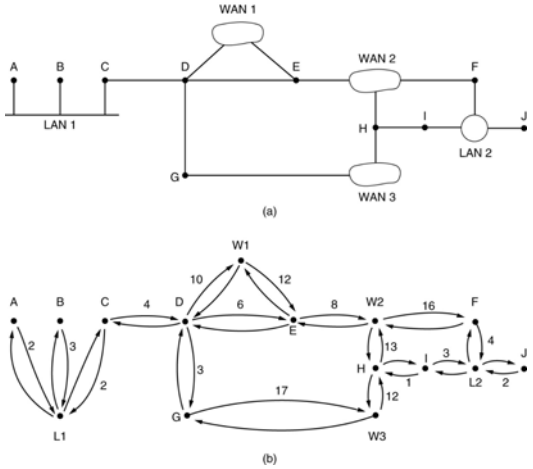
"advanced" features (not in RIP)

- Security: all OSPF messages authenticated (to prevent malicious intrusion); TCP connections used
- Multiple same-cost paths allowed (only one path in RIP)
- For each link, multiple cost metrics for different TOS (type of service) (eg, satellite link cost set "low" for best effort; high for real time)
- Integrated uni- and multicast support:
  - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- Hierarchical OSPF in large domains.

About RFCs

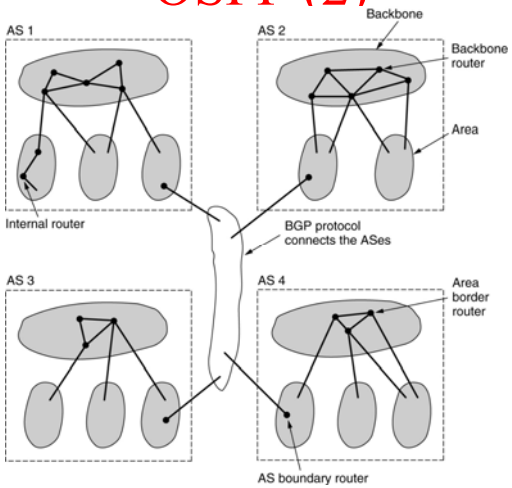
- OSPF V2 : RFC 2328 (Obsoletes: 2178)
- RIP V2 : RFC 2453 (Obsoletes: 1723,1388)
- How to find the "correct" RFCs
  http://www.apps.ietf.org/rfc/stdlist.html
- The newest and drafts can be found at working group pages
  http://www.ietf.org/html.charters/wg-dir.html

OSPF – The Interior Gateway Routing Protocol



(a) An autonomous system. (b) A graph representation of (a).

OSPF (2)



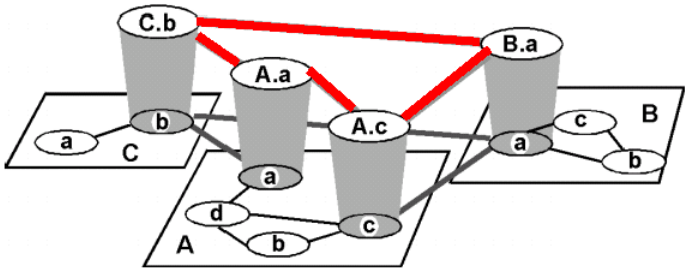
The relation between ASes, backbones, and areas in OSPF.

OSPF (3)

Message type	Description
Hello	Used to discover who the neighbors are
Link state update	Provides the sender's costs to its neighbors
Link state ack	Acknowledges link state update
Database description	Announces which updates the sender has
Link state request	Requests information from the partner

The five types of OSPF messeges.

Inter-AS routing



Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): *the* de facto standard,RFC1771
- **Path Vector** protocol:
  - similar to Distance Vector protocol
  - each Border Gateway broadcast to neighbors (peers) *entire path* (I.e, sequence of ASs) to destination
  - E.g., Gateway X may send its path to dest. Z:

Path (X,Z) = X,Y1,Y2,Y3,...,Z

Internet inter-AS routing: BGP

Suppose: gateway X send its path to peer gateway W

- W may or may not select path offered by X
  - cost, policy (don't route via competitors AS), loop prevention reasons.
- If W selects path advertised by X, then:

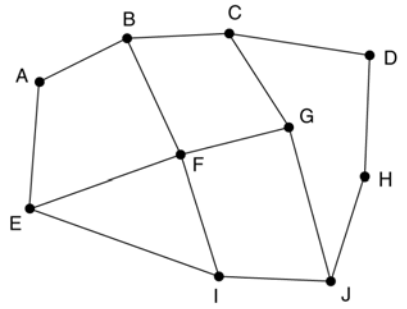
Path (W,Z) = w, Path (X,Z)

- Note: X can control incoming traffic by controlling its route advertisements to peers:
  - e.g., don't want to route traffic to Z -> don't advertise any routes to Z

Internet inter-AS routing: BGP

- BGP messages exchanged using TCP.
- BGP messages:
  - OPEN: opens TCP connection to peer and authenticates sender
  - UPDATE: advertises new path, withdraws old
  - KEEPALIVE keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - NOTIFICATION: reports errors in previous msg; also used to close connection

BGP – The Exterior Gateway Routing Protocol



(a)

Information F receives from its neighbors about D

From B: "I use BCD"
From G: "I use GCD"
From I: "I use IFGCD"
From E: "I use EFGCD"

(b)

(a) A set of BGP routers. (b) Information sent to F.

Why different Intra- and Inter-AS routing ?

Policy:

- Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

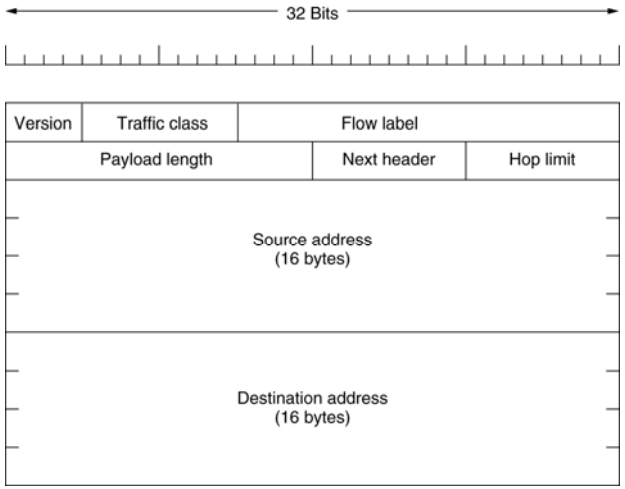
Scale:

- hierarchical routing saves table size, reduced update traffic

Performance:

- Intra-AS: can focus on performance
- Inter-AS: policy may dominate over performance

The Main IPv6 Header



The IPv6 fixed header (required).

Extension Headers

Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents

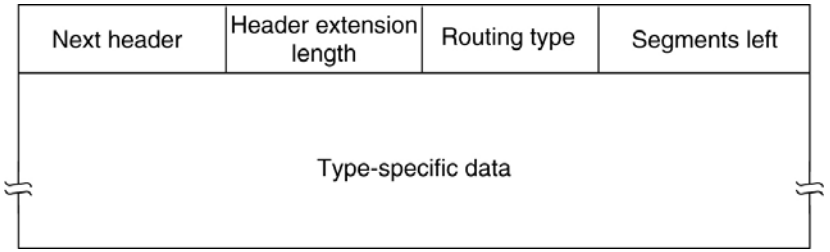
IPv6 extension headers.

Extension Headers (2)

Next header	0	194	4
Jumbo payload length			

The hop-by-hop extension header for large datagrams (jumbograms).

Extension Headers (3)



The extension header for routing.

Assignment P474 7 9 27 40