Computer Networks

Chapter 5: The Network Layer

(Version April 10, 2023)

Xiao Mingzhong

CIST, Beijing Normal University

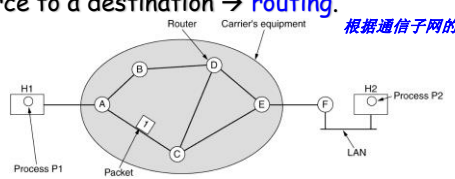
Outline

- **Network layer design issue**
- **Routing algorithms**
  - 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
- **Congestion Control Algorithms**
  - Congestion Prevention Policies, Hop-by-Hop Choke Packets ,RED, Jitter Control
- **Quality of Service**
  - Flow Requirements
  - Techniques for Achieving Good Quality of Service
  - 服务类型(Integrated Services、Differentiated Services)
  - Label Switching and MPLS
- **Internetworking**
- **The Network Layer in the Internet**

Network layer (1/3)

- Main service: Provide facilities for getting data from a source to a destination → routing.

根据通信子网的拓扑结构选路

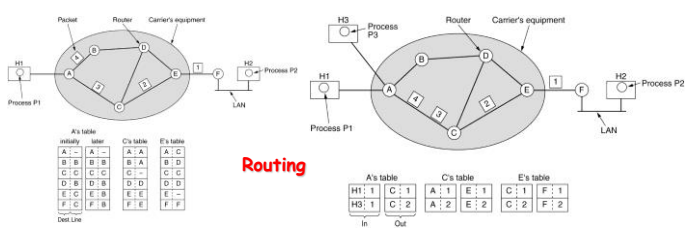


Note: data link layer moves frames only from one end of the wire to the other

- Again, distinguish between connectionless and connection-oriented services.
  - 无连接服务：基本原语send packet和receive packet及少量其他原语；分组的排序和流控制由主机负责。保持网络尽可能简单。
  - 有连接服务：先建立连接，做好资源预留，才收发分组。

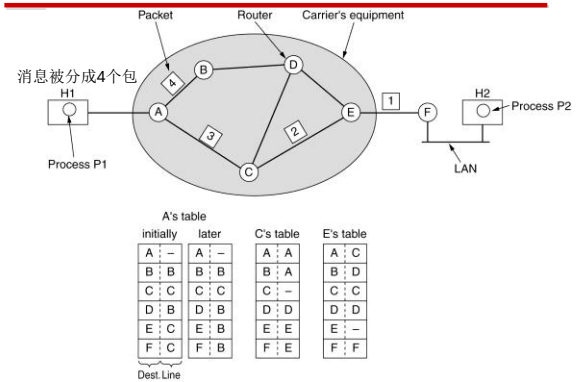
Network layer (2/3)

- There are two implementation techniques:
  - Virtual circuits are complete routes that are set up in advance.
  - Datagrams comprise individual packets of which the route is determined on the fly: they hop from router to router



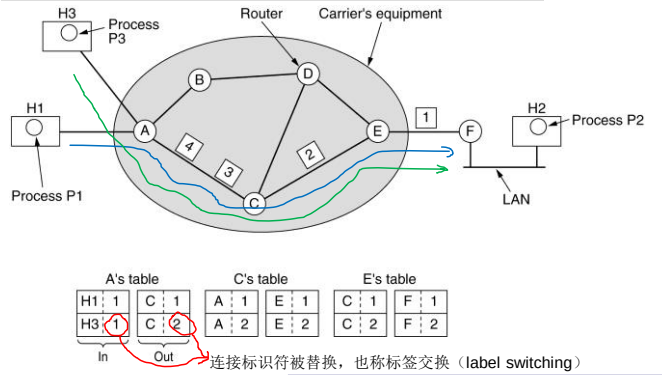
Routing

Implementation of Connectionless Service



Routing (forwarding) within a diagram subnet.

Implementation of Connection-Oriented Service



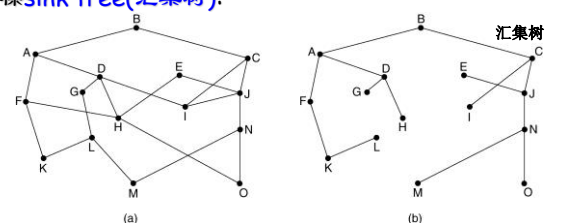
Routing(forwarding) within a virtual-circuit subnet.

Network layer（3/3） --- Comparison

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

Routing

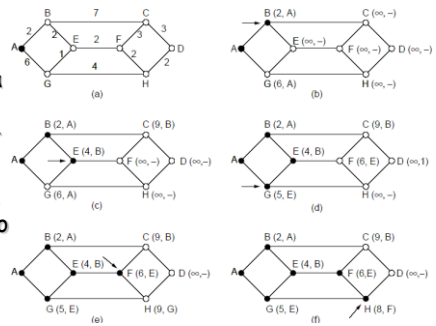
- Main issue: 网络中的路由器应该彼此协同，to find the best routes between all pairs of stations.
- Observation: 从其他站点到达B的最优路径"加在一起"应该是一棵sink tree(汇集树):



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

Shortest Path Routing(最短路径路由算法)

- 节点间距离定义
- Basic idea: during each step, select a newly reachable node at the lowest cost, and add the edge to that node, to the tree built so far.
- Dijkstra算法P300



Flooding(泛洪)

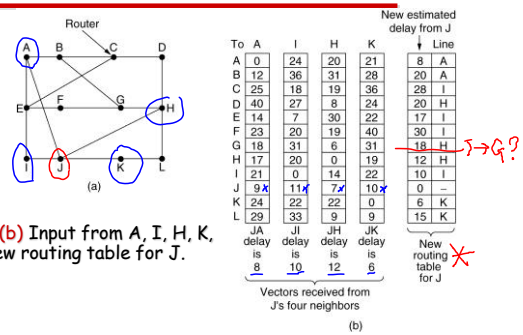
- Basic idea: Forward an incoming packet across every outgoing line, except the one it came in through.
- Basic problem: how to avoid "drowning by packets"?
  - Use a hop counter: after a packet has been forwarded across N routers, it is discarded. hop count=?
  - Be sure to forward a packet only once (i.e. avoid directed cycles). Requires sequence numbers per source router. Each router keeps track of the last sequence number per source router.
  - Flood selectively: only in the direction that makes sense.
- In general: flooding make sense only when robustness is needed:同时更新所有分布式数据库；无线网络消息传输；用来比较其他的路由算法（延迟最短）……

Distance Vector Routing*

- **DVR工作原理：** 每个路由器维护一张表(矢量),表中列出了当前已知的到每个目标的最佳距离以及所使用的线路。通过在邻居之间相互交换信息，路由器不断地更新它们内部的表。
- Basic idea: Take a look at the costs your direct neighbors are advertising to get a packet to the destination; Select the neighbor whose advertised cost, added with the cost to get to that neighbor, is the lowest; Advertise that new cost to the other neighbors.
  - **Example:**

Neighbor:	R_1	R_2	R_3
Link cost:	12	8	5
Advertised:	28	25	39
Total:	40	33	44

DVR - Example



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

DVR —— Count-to-infinity

- Problem: In DVR, it is possible to never find the route in the presence of node crashes:

初始A处于停机状态，
BCDE到A距离无穷大

A	B	C	D	E	
•	•	•	•	•	Initially
1	•	•	•	•	After 1 exchange
1	2	•	•	•	After 2 exchanges
1	2	3	•	•	After 3 exchanges
1	2	3	4	•	After 4 exchanges

(a)

好消息(A上线)反应快，
经过N次消息交换完成.

A停机或AB间线路断了

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
	⋮				
	•	•	•	•	

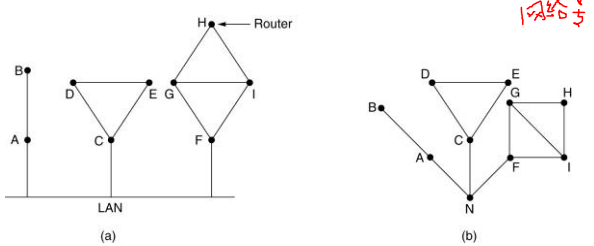
无穷计算! (b) 思路：设置距离上限。

Link State Routing

- Better idea: One way or the other, broadcast info on the entire network topology to all routers, and let each of them calculate a sink tree to the other routers. (next to just forwarding)
- What a router needs to do:
  - Find out who its neighbors are and get their network addresses.
  - Measure the cost for getting a packet to a neighbor.
  - Construct a link state packet telling all it has just learned.
  - Send that packet to all other router (hop-by-hop).
  - Do a Dijkstra.//计算出到每一个其他路由器的最短路径

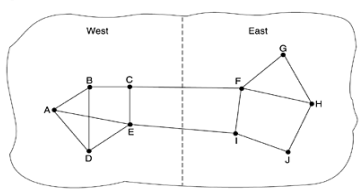
发现邻居节点

- 路由器启动，第一个任务就是找出哪些路由器是它的邻居
  - 首先，在每条点到点线路上发送一个特殊的Hello分组
  - 线路另一端的路由器回送应答说明它是谁（地址）
  - 另外，路由器是连接到的LAN的话，将LAN看作是一个节点。

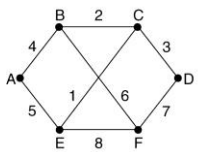


测量线路开销

- Simple: Just send an ECHO packet through each interface, and measure the round-trip delay. That'll give you a reasonable estimate of the actual delay.
- Problem: do we take the local load into account, or not (i.e., measure from the moment you insert packet into the network): //反映在队列延迟上



创建链路状态分组



(a)

	Link	State		Packets	
	A	B	C	D	E	F
1	Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
2	Age	Age	Age	Age	Age	Age
3	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

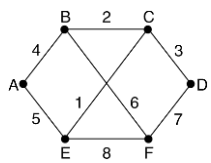
(b)

(a) A subnet. (b) 6个路由器的link state packets for this subnet.

- 创建链路状态分组的时机
  - 定期创建
  - 某些重要的事件发生,如:邻居节点离线,上线,…

发布链路状态分组

- 问题
  - 首先获得分组的路由器将改变它们的路由路径。不同路由器有可能使用不同版本拓扑结构,从而导致不一致、环、不可达等问题。
- 如何可靠地发布？扩散法。
  - 用递增的序列号，标识分组。接收方新分组转发，旧分组丢弃。
  - 序列号应足够大，避免序列号回转。
  - 年龄字段每秒减1，为0时该分组被丢弃，避免传输差错和路由器崩溃的情况。
  - 优化：延迟扩散，用于比较后来分组;分组需可靠传输（确认）;线路空闲，发送分组或确认。



			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	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	//来自E的分组到达两次
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

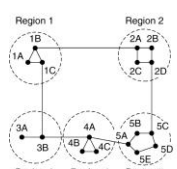
B的分组缓冲区

计算新的路由路径

- "有向"子网拓扑结构
  - 运行Dijkstra算法，构建到所有可能目标的最短路径。
- 例子
  - OSPF

Hierarchical Routing(层次路由)

- Problem: No routing algorithm discussed so far can scale: all of them require each router to know about all others →too demanding with respect to memory capacity and processing power.
- Solution: Go for suboptimal routes by introducing regions, and separate algorithms for intra-region and inter-region routing. Two or three levels will generally do.



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)

表空间减小

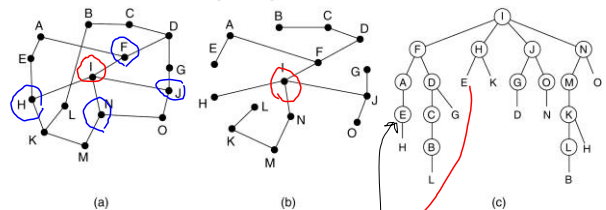
定理:
对于一个包含N个路由器的子网，最优的 (a)
级数是lnN，每个路由器要求elnN个表项

Broadcast Routing

- **Problem:** we want to send a message to (almost) every host (not router! but routing by routers) on the network. In practice, this means we're talking about (interconnected) LANs, or (relatively small) WANs.
  - Just send the message to each host individually. Not really good.
  - Use flooding. Acceptable provided that we can dam the flood.
  - Use multidestination routing, by means of a bitmap or list that is sent along with the packet. A router checks the destinations, and splits the list when forwarding it across different output lines.
  - Build a sink tree at the source and use that as your broadcast route. The sink tree has to be a spanning tree (as in Dijkstra). The routers need to know the trees.

Broadcasting: Reverse Path Forwarding

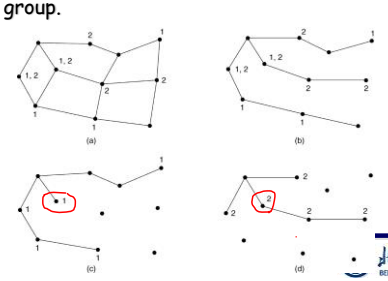
- Problem: suppose we don't know any spanning tree. How can we construct one at low cost?
- Solution: 路由器具备确认广播分组的能力;知道从自己到广播源的转发方向(线路);逆向转发或丢弃



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

Multicast Routing（1/2）

- Problem: we want to send a message to only a subset of all the nodes in a network. In that case, we need to know when a host enters or leaves the multicast group(**组管理:边缘路由器必须知道某刻它的哪些主机属于哪个组**)
- (Routing) Solution: Construct a spanning tree at each router. Use the group-id to prune paths to needs that do not contain members for that group.



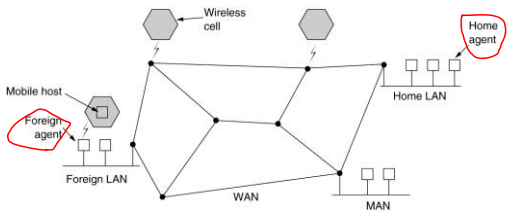
(a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

Multicast Routing（2/2）

- 多播分组只沿着正确的生成树被转发
  - 采用链路状态路由算法的网络，每个路由器有全局拓扑结构，能为源构造生成树，并从路径末端开始向根路由器前进“修剪”路由器，最后形成多播树。
  - 采用距离矢量路由算法的网络，修剪策略的基本算法是逆向路径转发。
- 可扩展性问题
  - m个成员的组有m棵组播树，n个组则有nm棵组播树。
- 基于核心的树(core-based tree,CBT)
  - 组播消息首先到达组根，然后均沿同一组播树被转发
  - 开销：n个组n棵树；问题：组播树并非最优。

Routing for Mobile Hosts(1/2)

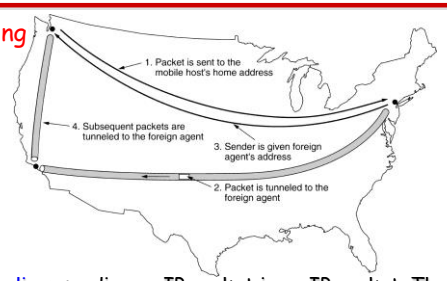
- 移动主机指迁移主机和漫游主机,即离开了原始站点(home)还想继续连接网络的主机。
- Problem: How can we forward messages to the mobile computer?



移动主机首先要向外部代理注册：1）联系上外部代理；2）提供其IP地址、Mac地址、安全信息等；3）外部代理请求家乡代理证实及处理；4）通过证实，记录在案，通知移动主机注册完成。

Routing for Mobile Hosts(2/2)

Routing



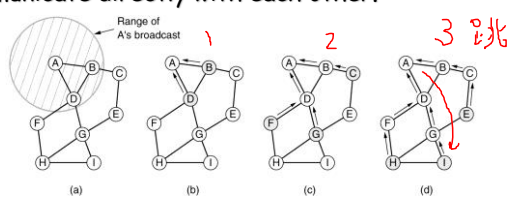
- Tunneling: sending an IP packet in an IP packet. That's the way to keep the routers ignorant of the fact that they're routing something else.
- Adapt routers: when you send the new address back to the source, intermediate routers can adapt their tables.

Routing in Ad Hoc Networks

- Essence: 不仅主机是移动的，路由器也是移动的。
  Example networks:
  - A fleet of ships, military vehicles in a battlefield, etc.
  - Spontaneous networks of PDAs, notebooks, digital phones and other mobile devices
  - Wireless mesh networks created as an alternative to wired networks //通常每个网络节点，既是路由器又是主机
- Problem: 网络拓扑动态改变; nothing is fixed anymore.
  - AODV (Ad hoc On-demand Distance Vector)路由算法,考虑了带宽有限、电源寿命短的限制（通过不周期广播整个路由表达成）
  - 当要给目标发送分组时，才计算到目标的路由。//按需特性

AODV – Route Discovery (1/3)

- Starting point: represent the network as a graph in which any two nodes are connected only if they can communicate directly with each other.



- Issue: when A wants to send a message to I, it needs to know where to forward the message to (B or D) →broadcast a route request (which will reach only B and D) if A does not have a route to I.

AODV – Route Discovery (2/3)

- Assume route request arrives at an intermediate node that doesn't know how to get to I: increment the request's hop counter, and broadcast again. Eventually, request will arrive at I.
- Several route requests for A → I may have arrived. The one with the lowest hop counter indicates the shortest path.
- I Send a route reply back to the neighbor that had the lowest hop count, which will forward it towards A.

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

Source address	Destination address	Destination sequence #	Hop count	Lifetime

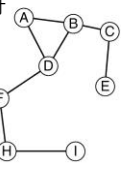
AODV – Route Maintenance（3/3）

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)

F出来的包刚才有前往ABC的

AB出来的包刚才有前往FGHI的



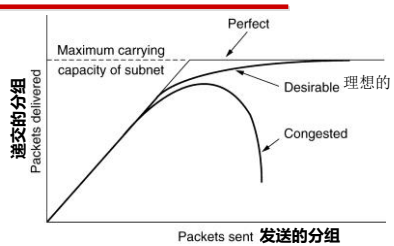
(b)

(a) D's routing table before G goes down. (b) The graph after G has gone down.

- Assume G leaves the network. In the case, D will discover that the routes it had registered for E,G, and I are no longer valid（定期广播hello消息）, and will inform A and B that they need to update their tables.

Congestion

- Problem: when too many packets have to be transmitted through the network, we can get into a serious performance problem---拥塞:



- Congestion can be caused by lack of bandwidth, but also by ill-configured or slow routers.

分析：都会导致在路由器的排队。队列满，分组会被丢弃。增大缓存不能根本解决，因为分组到达队列前端的时候，已经超时被重发。又有大量的包拥入网络，拥塞更为严重。

拥塞控制 vs 流量控制

- **拥塞控制**
  - 任务是确保子网能够承载其能力范围内的流量；
  - 涉及全网主机和路由器可能削弱子网承载容量的因素，如：路由器的存储转发过程等。
- **流量控制**
  - 任务是确保一个快速的发送方不会持续地以超过接收方接收能力的速率传输数据；
  - 通常做法是接收方向发送方提供反馈，告诉发送方自己的当前情况。
- **混淆的原因**
  - 一台主机既可能由于接收方不能处理负载的原因，也可能因为网络不能处理负载的原因而接收到“减慢发送速度”的消息。

拥塞控制的通用原则

- 开环方案
  - 一开始就保证问题不会发生，系统启动运行后不作修正；
  - 手段有：确定何时接收新的流量、何时丢弃分组及丢弃哪些分组，以及在网络的不同点上执行调度决策。共同点都是做出决定时，不考虑网络的当前状态。
- 闭环方案
  - 建立在反馈思想之上。流程大致为：

1）监视系统，检测到何时何地发生了拥塞，如：平均队列长度、丢包率、分组延迟等度量指标；

2）将该信息传递到能够采取行动的地方，如：告诉流量源；

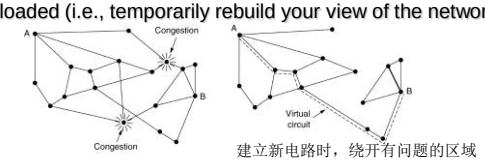
3）调整系统的行为，以改正问题，如：流量源慢发或择路；基本思路是增加资源或者降低负载。

影响拥塞的网络技术

Layer	会导致或影响拥塞的Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination 太短，丢弃重传；太长，丢弃话，响应时间大
Network	• Virtual circuits versus datagram inside the subnet 方法特有 • Packet queueing and service policy 队列设置及处理顺序 • Packet discard policy 丢弃哪个分组 • Routing algorithm 选好路 • Packet lifetime management 生存期太短，未到目标就丢弃引发重传
Data link	• Retransmission policy 选择性重传和回退n帧 • Out-of-order caching policy 丢弃会引发重传 • Acknowledgement policy 捎带确认和立即确认 • Flow control policy 窗口大小，越小发送速率越低

Congestion Control in Virtual-Circuit Subnets

- Principle: when you set up a circuit, be sure that congestion can be avoided.
  - Admission control: if it's too busy, just refuse to set up a virtual circuit. This is the same as refusing new users at an FTP site.
  - Select alternative routes when a part in the network is getting overloaded (i.e., temporarily rebuild your view of the network):



建立新电路时，绕开有问题的区域

  - Negotiate the quality of the circuit in advance, so that the network provider can reserve buffers and the like. Resources are guaranteed to be there.

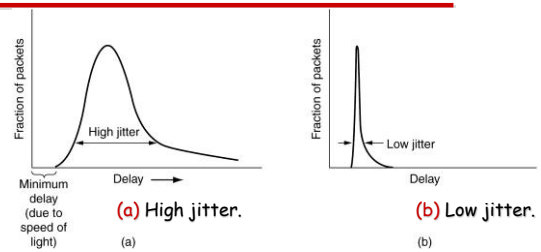
Congestion Control in Datagram Subnets

- 警告位
  - 路由器对经过的分组设置警告位，在接收方确认中告诉发送方
- 抑制分组
  - 路由器给源主机发送抑制分组，通常减速方法是收到第i个抑制分组，发送速率降为$1/2^i$。
- 逐跳抑制分组
  - 给源主机发送抑制分组，反应太慢。比较在P332。
  - 要求上游更多的缓冲空间。可以将拥塞消灭在萌芽状态，同时又不会丢失任何分组。See book for detail。

负载丢弃

- 当路由器感觉来不及处理而要被淹没的时候，就将分组丢弃。
  - 随机丢弃、按优先级丢弃
  - 丢弃应用新分组，保留老分组，葡萄酒策略
  - 丢弃应用老分组，保留新分组，牛奶策略
- 随机的早期检测（RED算法）*
  - 计算平均队列长度，在实际耗尽所有的缓存之前就开始丢弃分组。丢弃谁？概率接收新分组的进入。
  - 要告诉发送方吗？可以发送给源抑制分组；隐式反馈：不发送抑制报文，接收方没有收到确认就认为网络拥塞，包被丢弃，于是减缓发送速度。
  - 不适用于无线网络，因为丢包多由于干扰引起。

抖动控制 (done by subnet)



- 分组到达时间的变化量，称为抖动。
- 计算出沿途每一跳的传输时间，来对抖动实施控制。
  - 发生线路竞争时: 晚了，尽快转发；早了，可呆会。
- 区分:流媒体点播系统也可利用“缓存”来消除抖动，但会议或电话系统这样的延迟不可接受。

应用（包流）对（网络）QoS的需求

- Definition: A stream of packets from source to destination is called a flow.
- Quality of Service (QoS): the needs of each flow are determined by reliability, delay, jitter, and bandwidth.

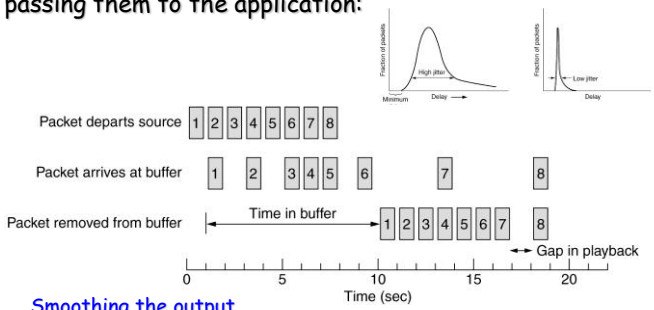
Application	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Videoconferencing	Low	High	High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

Techniques for Good QoS

- Overprovisioning
  - 提供足够的路由器容量、缓冲区空间和带宽，以保证分组能顺利地通过。
- Buffering to reduce jitter*(done by target)
- Traffic shaping and policing to avoid bursts*
  - 流量整形是调节数据传输的速率及突发性
  - 流量监管是按照服务等级协定考察流量形状
- Resource reservation
- Admission control
- Packet scheduling*

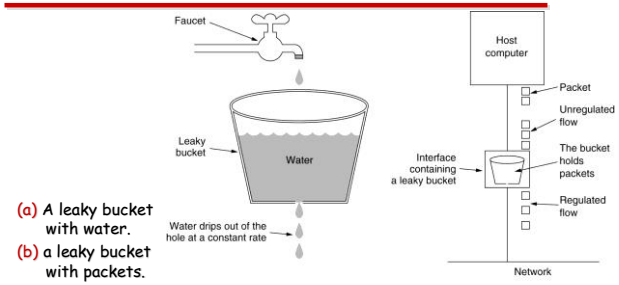
QoS: Buffering

- Basics: just try to reduce jitter as much as possible by buffering incoming packets at the receiver before passing them to the application:



Smoothing the output stream by buffering packets.

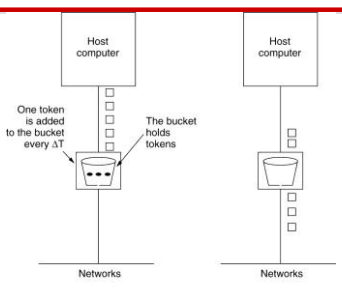
QoS: Traffic Shaping – Leaky Bucket



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

- Note: we've eliminated bursts completely: packets are passed to the network when available, and all at the same rate. This may be a bit overdone. Also, packets can get lost.

QoS: Traffic Shaping – Token Bucket



(a) Before. (b) After.

- Tokens are added at a constant rate; as soon as enough tokens have been saved, one or more packets can be sent. //允许突发
- To avoid still too much burstness, put a leaky bucket behind a token bucket (with a larger rate).

QoS: Effects of Buckets

(a) Input to a leaky bucket.

(b) Output from a leaky bucket.

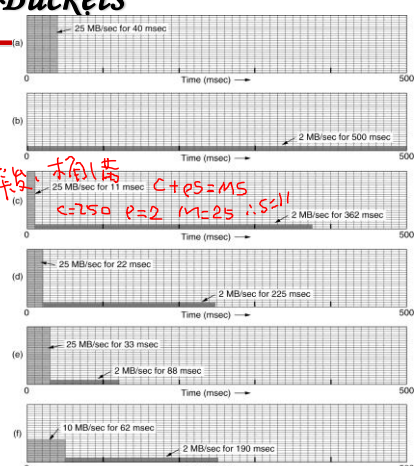
Output from a token bucket with capacities of

(c) 250 KB

(d) 500 KB,

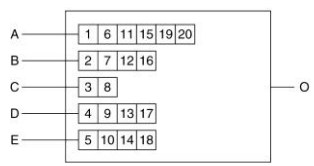
(e) 750 KB,

(f) Output from a 500KB token bucket feeding a 10-MB/sec leaky bucket.



QoS: Packet Scheduling

- Problem: what happens when you need to support multiple flows and one of them is too resource-consuming. To enforce cooperation, use fair queuing or weighted fair queuing:



(a)

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

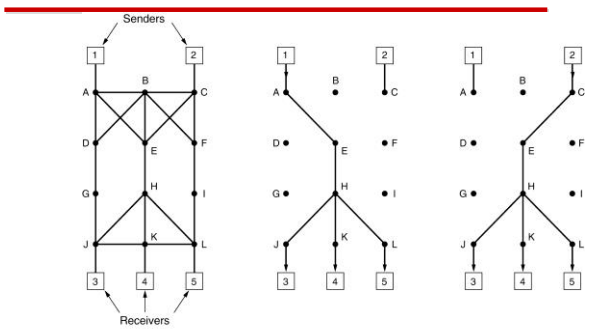
(b)

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

Integrated Services（综合服务）

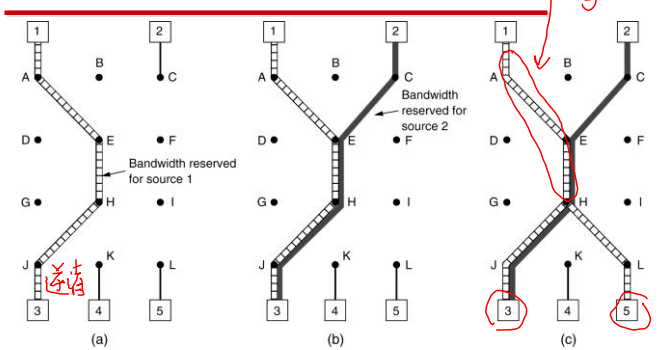
- Problem: in order to efficiently support streams, we cannot set up a single connection per stream, but need to integrate things. This is particularly the case with multicast applications, for which we then need to assume a large and frequently changing group of receivers.
- Basic idea: we set up multicast trees from sources to destinations, but this time take into account the bandwidth needed by all the receivers. Bandwidth can be reserved on pre-constructed trees. If there's not enough bandwidth as required by the receiver, a failure is reported back.

综合服务例子：RSVP协议



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

RSVP(Resource ReSerVation Protocol)

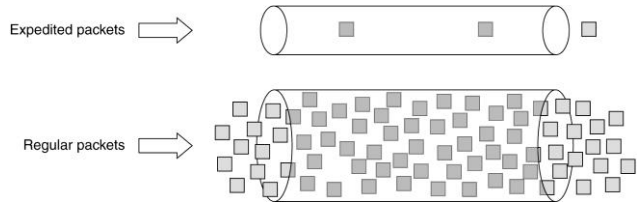


(a) Host 3 requests a channel to host 1. (b) Host 3 then requests a second channel, to host 2. (c) Host 5 requests a channel to host 1.

Differentiated Services（区分服务）

- Problem: *Integrated service require connection setup*. Instead, offer a means for local QoS decision-making → differentiated services (for routers in one administrative domain).
- Basic idea: for each class of applications, reserve resources. In other words: differentiate (in advance) the applications that the network will have to support.
- Example: let routers differentiate regular from expedited traffic(快速型转发服务). Packets belonging to either class will be marked as such.

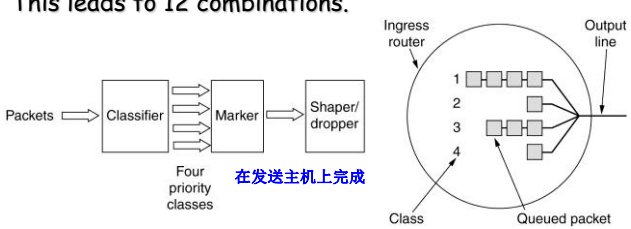
快速型转发



- 每条输出线路两个队列，一个用于快速类别的分组，另一个用于常规分组。
  - 分组到达，按其类别，进入不同队列
  - 路由器执行“加权的公平排队”分组调度机制
- 也可，每隔几个常规分组一个快速类别的分组

Differentiated Services: Assured Forwarding

- Basics: Distinguish four priority classed and three discard probabilities in case you're also shaping traffic. This leads to 12 combinations.

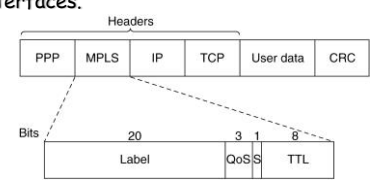


A possible implementation of the data flow for assured forwarding.

Label Switching and MPLS

- Essence: instead of having standard routers provide QoS on datagram routing, let them try to *establish connections(然后2.5层交换)*.
- Techniques: add a connection-id to datagrames and let routers take that ID as index into a table with already determined routes, identified by outgoing interfaces.

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



- Note: Routes can be established on-demand, or when a router comes up (see book). Also note that the label is used per router to match incoming-to-outgoing interfaces. A label may be changed when a packet leaves the router.

Internetworking（1/2）

- Problem: We have all these networks, with all different protocol stacks, and now we just to let them talk to each other.
- Non-solution: Enforce all networks to run the same protocol stack. That's asking for a lot of trouble, and effectively saying that we are not allowed to make any progress.
- Solution: construct all kinds of gateways that connect to different kinds of networks.

Internetworking(2/2)

- Repeaters at the physical layer for boosting signals.
- Bridges/Switches to make the interconnection at the data link layer.
- Multi protocol routers for forwarding, and possibly splitting up packets (bridges can't do the later).
- Transport gateways for coupling **byte streams** in different networks.
  比如：把一个TCP连接和一个SNA连接“粘连”起来
- Application gateways, e.g., for handling electronic mail between OSI and TCP/IP networks.
  翻译消息语义，如：Internet电子邮件和X.400电子邮件之间的网关必须解析电子邮件消息，且改变各个头域。

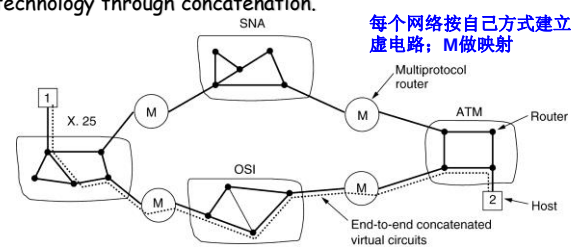
How Networks Differ（网络层）

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

- Don't worry: it's impossible to resolve all differences. The solution is to just take a simple approach (like the Internet). We now only consider the starred (*) issues.

Concatenated Virtual Circuits

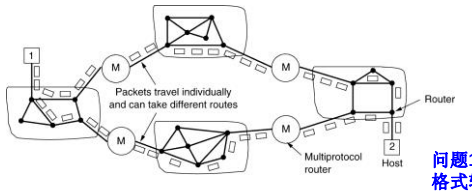
- Basic idea: Assume the constituent networks support virtual circuits. In that case, the internet can use virtual circuit technology through concatenation.



- Note: if one of the constituent networks does not support virtual circuits, or, for example, provides only unreliable data transport, simple concatenation will be hard.

Using Datagrams

- Basic idea: internetworking layer offers only datagram services: unreliable, unordered packet flow.



问题1：M做不同分组格式转换困难！

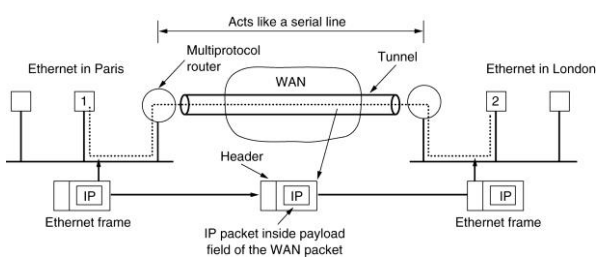
  - Main problem: Addressing – different networks use completely different addresses, M需要地址映射表，这条思路会引发很多问题。
  - Solution: You don't solve it, but instead consider, for example, IP as a universal network protocol. Lots of universal network protocols are available. //通用（互联）网络层协议

两种互连方法的比较

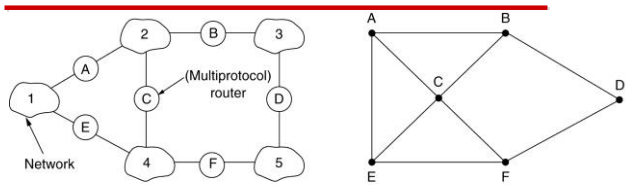
- 虚电路方法
  - 可以提前预留缓冲区、分组顺序可以得到保证、可以使用较短的头部，…
  - 路由器为每个打开的连接使用表空间、无法使用其它路由路径、易受沿途路由器失败的影响
  - 连接不可靠数据报网络的话，难以实现串联虚电路
- 数据报方法
  - 可能会拥塞、面对路由器失效有更好健壮性、需要更长的头部，…
  - 最大优点：可以包含未使用虚电路的子网,性能提升.

Tunneling

- Basic idea: We can solve a lot of the internetworking problems when we can assume that the source and destination are on the ~~same type~~ of network. In that case, we need only to tunnel packets through intermediate networks.



互联网路由

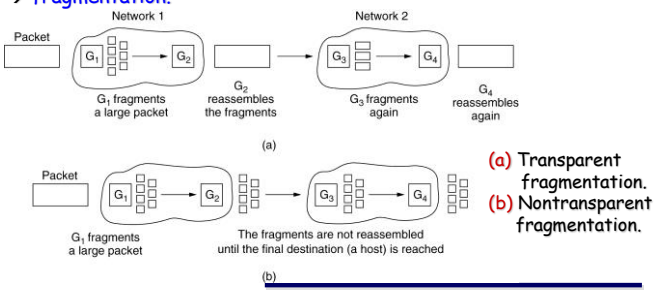


(a) An internetwork. (b) A graph of the internetwork.

- 在多协议路由器上需要使用路由算法，两级路由算法：
  - 在每个网络内部使用的内部网关协议（网关=路由）
  - 在网络间使用的**外部网关协议**(路由选择还与收费、政治等问题有关)
    注:这里采用"两级路由"的主要原因是互联,而非可扩展性;另外,就fig.a来说,路由器A实际上运行着3个路由算法(假定1,2各有自己的路由协议),但不参与域内转发.
- 过程描述
  - LAN中的互联网分组→本地多协议路由器→直接下一跳（本地主机\路由器）或是隧道下一跳→ … →目标网络→目标主机

Fragmentation

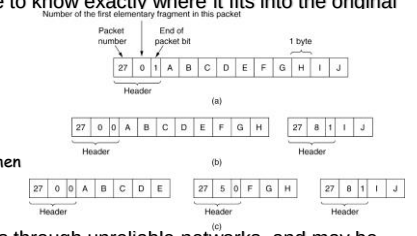
- Problem: Different networks may impose different maximum packet sizes. This means that we may have to split a packet into smaller ones when forwarding it through an intermediate network → fragmentation.



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

Fragmentation - Reassembly

- Problem: when we create fragments, how do we paste them together again:
  - A fragment may be fragmented again by successive intermediate networks → when a fragment arrives at the destination, we have to know exactly where it fits into the original packet. Solution:



(a) Original packet
(b) (c) Fragmentation when the elementary data size is 1 byte.

  - Fragments may pass through unreliable networks, and may be lost → we need to decide what to do about lost fragments. Solution: Discard the entire packet. VS. 重传丢失的部分.

Outline

- 10大Internet设计原则
- The Network Layer in the Internet
  - IP Packet & Addresses
  - IP Routing Protocol
    - OSPF – The Interior Gateway Routing Protocol
    - BGP – The Exterior Gateway Routing Protocol
  - Internet Control Protocols
  - Internet Multicasting*
  - Mobile IP*
  - IPv6*

10大Internet设计原则（1/2）

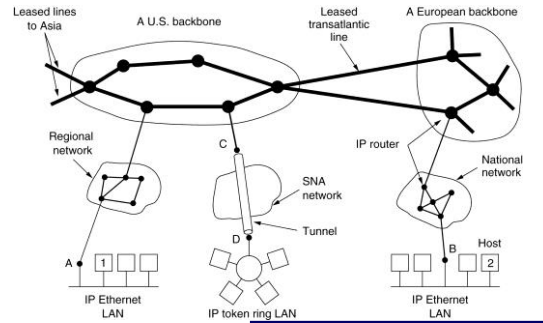
- 保证它能够工作
  - 有了多个原型系统可以成功相互通信之后，才可以最终确定设计或者确定标准。这才是正确的工作方式。
- 尽可能使它简单
  - 任何时候都应该使用最简单的方案。若一项特性并非绝对本质的特性，那么就不该考虑之。
- 做出明确的选择
  - 用两种或多种方法来做同样的事情简直是在自找麻烦，需择一。
- 尽可能做到模块化
  - 协议栈思想的源泉，强调层次性、独立性
- 期望具备异构性
  - 异构的客观存在，要求网络设计必须简单、通用和灵活

10大Internet设计原则（2/2）

- 避免使用固定不变的选择和参数
  - 若不可避免要使用参数的话，最好是让发送方和接收方协商，而不是定义固定的参数值
- 寻找一个好的设计，它不必是最完美的
  - 即忽略一些怪异的特例，不必大幅修改好的设计适应之，应将支持怪异特例的负担转移到那些对此有特殊需求的人身上
- 对于发送操作一定要严格，而对于接收操作要有一定的容忍度
  - 只发送那些严格符合标准的分组，但容许收到不完全符合标准的分组并试图对它们进行处理
- 要考虑伸缩性
  - 主要反映在不要使用中心化的数据库，且须考虑负载的均分分布
- 要考虑性能和代价
  - 若一个网络的性能很差或代价特别高，没有人会使用这样的网络

What is the Internet

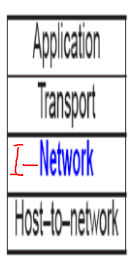
- view it as a collection of autonomous systems connected together by a bunch of backbones:



IP的设计初衷就是互联不同的网络

Internet Model

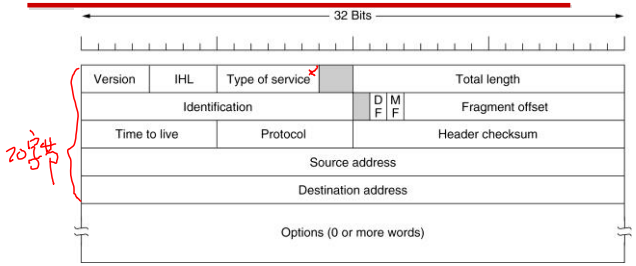
- An application offers a data stream to the transport layer, using either connection-oriented or connectionless services.
- The transport layer breaks up the data stream into datagrams, and passed these to the network layer.
- The datagrams are routed through the internet, occasionally fragmented when needed.
- Routers always pass the datagram to the underlying data link layer (generally LANs and (dial-up or leased) telephone lines).



Internet network layer

- IP
  - Addressing
  - Datagram format
  - Fragmentation and packet handling
- ICMP,ARP,RARP,DHCP…
  - Error reporting
  - Signaling
- Routing: defining paths and compiling forwarding tables
  - RIP
  - OSPF
  - BGP

IP Header



IHL: length of the header(32); TOS: 3-bit priority plus 3-bit flag (delay, throughput, reliability)
TLen: length header + payload(8) ID: datagram id DF: don't fragment this datagram
MF: there are more fragments after this one
Foff: Offset in original datagram where this fragment belongs. 8字节是基本分段单位
TTL: Maximum number of hops allowed Prot: Globally defined ids for transport port.
HCS: Checks the headers (has to be calculated at each hop)

IP Options

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

- Security is hardly used: specifying that a datagram is "really top secret" is not such a good idea.
- Source routing can be effectively used to force routes: debugging, politics, and mobile hosts.
- Record route and Timestamping are mainly used for debugging
- Note: Not all routers actually support these options

主机、路由器都需要IP地址

- 例子：连接3个网络的两个路由器地址分配情况

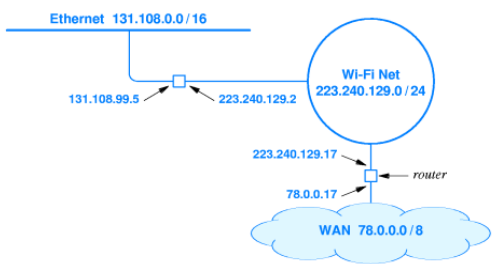
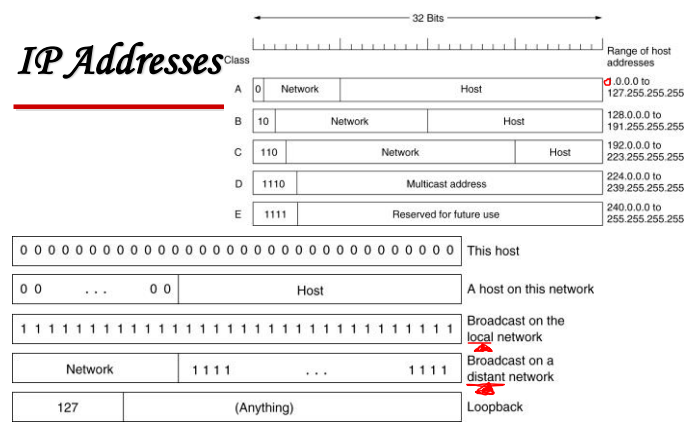


Figure 18.9 An example of IP addresses assigned to two routers. Each interface is assigned an address that contains the prefix of the network to which the interface connects.

IP Addresses

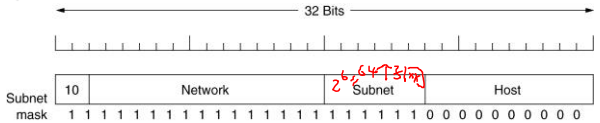


Class	M.nets	M.hosts
A	126	16777214（2^26-2）
B	16382	65536
C	2097150	254

现在这种分类的编址方案已经不再使用！

Subnet Masking

- Problem: All hosts on the same network must have the same network number. This may cause a single organization to acquire several classes of addresses (e.g. each time it adds a LAN), that would subsequently need to be announced worldwide (WHY?)
- Solution: Use a single network address for the entire organization, and internally divide the host address space into a subnet address and a host id:

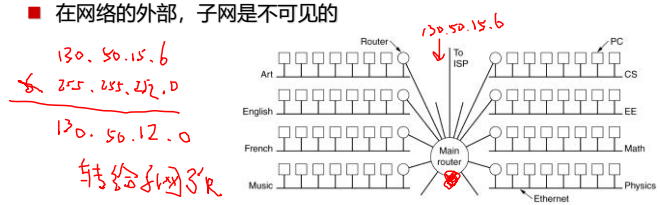


- Note: we're introduced a 3-level routing hierachy.

各子网起始IP地址：*.*.4.1; *.*.8.1; *.*.12.1; ……
掩码:255.255.252.0

例：

- 某大学有一B类网络地址
  - 通过4个中继器连接而成的以太网，允许的主机个数无论如何都使用不了64k个IP地址?
  - 又要互连新的网段，申请新的网络地址?
- 子网技术
  - 让A、B和C类网络地址代表一组网络，而不是一个网络：掩码
  - 路由表项四类：（网络，0）、（当前网络，主机）、（当前网络、子网，0）、（当前网络、当前子网，主机）
  - 转发：分组的目标地址 & 网络子网掩码 决定下一跳
  - 在网络的外部，子网是不可见的



解决IP地址匮乏的措施

- CIDR
  - Classless InterDomain Routing（无类别域间路由）
  - 一种目前正在使用的、并且给与Internet额外喘息空间的（编址、寻址）方案。
  - 基本思想：将剩余的IP地址以可变大小块的方式进行分配，而不管它们所属的类别。意味着”转发”过程变得更为复杂!
- NAT
  - Network Address Translation（网络地址转换）

CIDR

- Assign class C addresses in contiguous blocks of 256 addresses. Also, partition the address space into four zones:

194.0.0.0	–	195.255.255.255	Europe
198.0.0.0	–	199.255.255.255	North America
200.0.0.0	–	201.255.255.255	Central/South America
202.0.0.0	–	203.255.255.255	Asia and the Pacific

A series of contiguous blocks is assigned, together with a 32-bits mask…

CIDR - Example

Site	# Addr.	Range	Notation
S1	2048	194.24.0.0 – .7.255	194.24.0.0/21
S2	1024	194.24.8.0 – .11.255	194.24.8.0/22
–	1024	194.24.12.0 – .15.255	194.24.12.0/22
S4	4096	194.24.16.0 – .31.255	194.24.16.0/20

地址如何分配及掩码如何得到的？P375

- Update all routers (in Europe) with three entries, each consisting of a (base address, mask)-pair. When an IP datagram arrives at a router, the latter tries to find the base address by using the masks:

$$(194.24.17.4 \&\& 255.255.248.0) \neq 194.24.0.0 \quad \surd$$
$$(194.24.17.4 \&\& 255.255.252.0) \neq 194.24.8.0 \quad \surd$$
$$(194.24.17.4 \&\& 255.255.240.0) = 194.24.16.0 \quad OK$$

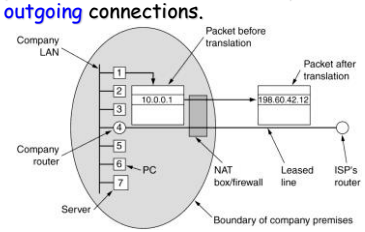
- Note: the "/21" notation indicates that the mask consists of 21 1-bits, followed by 0-bits only.

CIDR – Aggregate Entries

- Note: In principle, all routers worldwide need to store the masks to do the appropriate routing.
- Observation: for many routers outside 194.0.0.0, the only thing they see is that there are (at least) 3 network addresses for which packets follow the same route. These entries can be aggregated into 194.24.0.0/19 with a single submask of 19/13(1/0) bits.

Network Address Translation

- Essence: 申请一些 IP address for local networks that operate behind a special router (which can also operate as a firewall), and allow only outgoing connections.



- Example: When a connection is set up from address 10.0.0.1, port X, the router sends it off using source address 198.60.42.12 (its ISP-supplied address) on port Y, and registers the mapping X←→Y. when a reply comes in for port Y, it is sent back to 10.0.0.1 on port X. //How can you allow incoming connections?

ICMP

- Basic idea: we need to inform hosts and routers when things go wrong, or, likewise, should be able to send queries to get status information. Encapsulate control messages in normal IP datagrams.
  - Distinguish several message types (destination unreachable, time exceeded, etc.)
  - Each message type is further specified by a code. Example: a destination can be unreachable (message type=3) because of an unsupported protocol (code=2)
  - An ICMP message includes the header of the IP datagram that caused the message to be sent.
- Question: what should we do when an ICMP message is lost?
- Question: How can be recognize ICMP messages when they are encapsulated in IP datagrams?

Address Resolution Protocol(1/2)

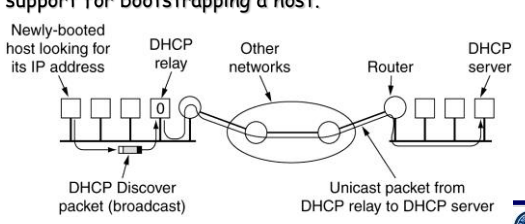
- Situation: Addressing hosts using IP addresses is great, but these addresses are not recognized by the hardware of those hosts.
  - Example: a host on an Ethernet LAN will only read messages encapsulated in frames containing that host's hardware address.
- Problem: How do we find out the hardware (i.e. datalink) addresses of a host, given its Internet address?

Address Resolution Protocol(2/2)

- The protocol used to discover network addresses is called Address Resolution Protocol (ARP)
  - S1. Router: ask each host on the LAN whether they have the requested IP address. This is done by encapsulating the query as an ARP message in a datalink frame, and broadcasting it.
  - S2. Host: recognize it is dealing with an ARP message, checks whether it has the requested address, and if so, sends a reply back with its datalink address. Question: How can the host recognize an ARP message?
  - S3. Router: Recognizes a reply ARP message, and (generally) caches the IP address with the datalink address. It can then forward IP datagrams to the correct host, encapsulating them in datalink frames.
- Usually the IP-to-Ethernet mapping of the router is also included in the request so all nodes can store it

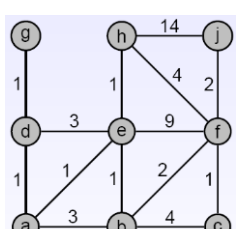
Reverse Address Resolution Protocol

- Problem: So how does a host know its own IP address?
- Solution: the host uses a limited broadcast (i.e. restricted to its own network) to query a RARP server for its IP address. The RARP server maintains a table of (datalink address, IP address) mappings.
- Observation: RARP has largely been replaced by Dynamic Host Configuration Protocol (DHCP). It does the same, but a bit like support for bootstrapping a host.



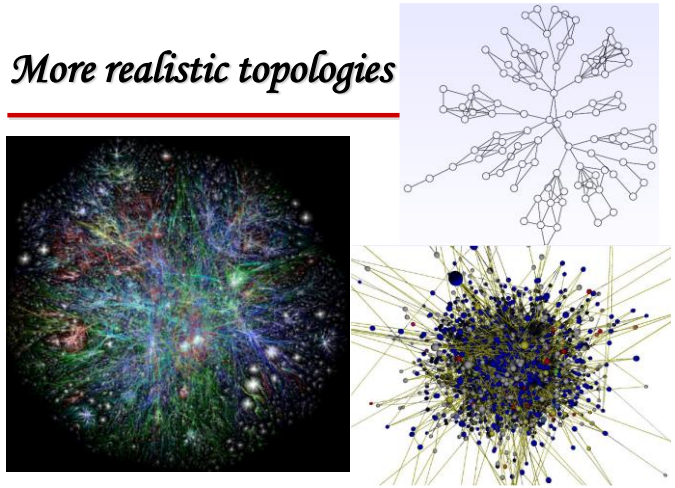
Network model

- So far, we have studied routing over a "flat" network model



- Also, our objective has been to find the least-cost paths between sources and destinations

More realistic topologies



Hierarchical Routing

- The network flat model is too simplistic for at least two important reasons:
  - Scalability
    - Hundreds of millions of hosts in today's Internet
    - Transmitting routing information (e.g., LSAs) would be too expensive
    - Forwarding would also be too expensive
  - Administrative autonomy
    - One organization might want to run a distance-vector routing protocol, while another might want to run a link-state protocol
    - An organization might not want to expose its internal network structure
- Today's Internet is organized in *ASs*
  - Independent administrative domains
  - Gateway routers connect an AS with other AS.

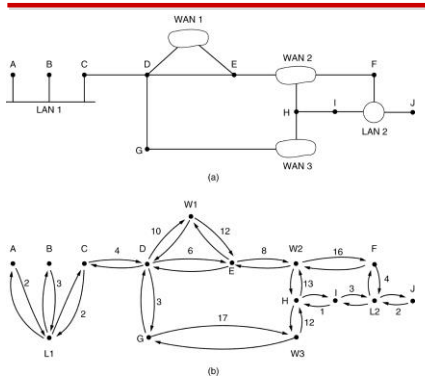
IP Routing

- Basic idea: Make a distinction between routing in an autonomous system, and between autonomous systems:
  - Intra-AS routing is concerned with getting packets from source to destination. It should do this as best as possible (optimal routing). Interior Gateway Routing.
    - E.g., RIP(distance vector) mainly used in low-tier IPSs and Small enterprises and OSPF(link state) used by tier-1 ISP.
  - Inter-AS routing has to deal with a lot of politics. For example, some ASes should not be traversed at all, whereas some do not accept "foreign" packets. Exterior Gateway Routing.
    - BGP is the Internet standard.

Interior Gateway Routing: OSPF

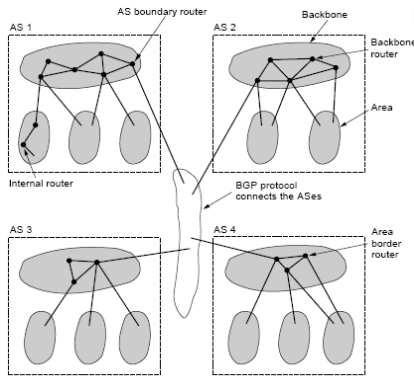
- Open Shortest Path First: link state routing protocol, replacement for (still widely used, but inadequate) RIP (routing information protocol). Requirements:
  - Openness: the algorithm should be made publicly available so that anyone could implement it.
  - Support for different distance metrics (hops, delays, costs, etc.)
  - Dynamic and efficient adaptability to changing topologies.
  - Support routing based on type of service (already specified in IP, but no one actually used it). Especially important for real-time (multimedia) traffic.
  - Support for load balancing: when a route is heavily used, another one should be selected.
  - Support hierarchical routing.
  - Offer security.
  - Support IP tunneling.

OSPF – Routing Graphs



- Note: OSPF build different graphs, depending on the distance metrics it uses (delay, throughput, reliability). Of course, physical links are always taken into account.
- Note: Each LAN is represented by a designated router that acts as a representative of the other routers on that LAN.

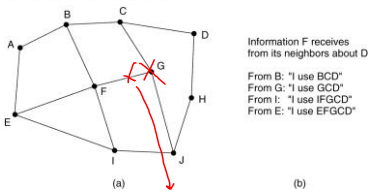
OSPF – Hierarchical Routing



- Note: We can use the same algorithm for the areas and the backbone. In fact, the backbone behaves as just another area. This means it should be contiguous.

Exterior Gateway Routing: BGP

- Basic idea: To ensure that routing polices are met, they are generally configured manually into each BGP router. In other words: routers do not automatically use the routes they find, but have to check whether it is allowed. Neighboring routers maintain a connection to simplify message reliability.
- Routing: is based on distance vector routing, but paths rather than distances are announced:

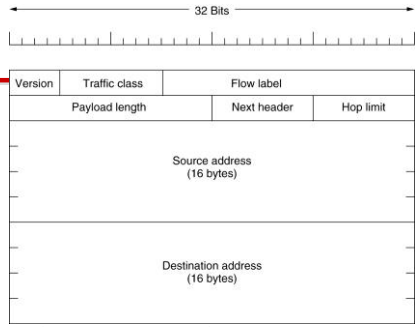


- Note: we do not have a count-to-infinity problem here, because a router decides on an entire path.

IPv6

- Problem: the current version of IP can not support enough addresses, but also lacks the flexibility to act as the basis for a large variety of users. The goals:
  - Support billions of hosts.
  - Reduce the size of routing tables.
  - Simplify the protocol to enable faster routers.
  - Provide (better) security
  - Better support for type of service
  - Support scopes with multicasting
  - Support roaming hosts without address changes.
  - Coexistence of the old and new protocol, and evolution of the new one.

IPv6 - Header



- Note: the flow label is used to set up a pseudoconnection between source and destination. It identifies a flow for which, for example, bandwidth has been reserved.
- Note: A simpler header is almost impossible – further info is provided by next headers.
- Note: No checksum, and no fragmentation fields.

8000:0000:0000:0000:0123:4567:89AB:CDEF
8000::123:4567:89AB:CDEF

IPv6 – Address Space

- Big difference: IPv6 uses 16-byte addresses. This is really a lot: 7*10^25 addresses per square meter. It does allow us to be less efficient with address allocation: 72% is unassigned.

Prefix	Usage	Fraction
0000 0000	Reserved (incl. IPv4)	1/256
0000 0001	Unassigned	1/256
0000 001	OSI NSAP addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Unassigned	1/16
001	Globally unique unicast addr.	1/8
010	Provider-based addresses	1/8
011	Unassigned	1/8
100	Geographic-based addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 1110 0	Unassigned	1/512
1111 1110 10	Link local unicast addr.	1/1024
1111 1110 11	Site local unicast addr.	1/1024
1111 1111	Multicast	1/256

ICMPv6 (1/2)

- Note: ICMPv6 integrates a number of issues that were separated in IPv4, and extends some of IPv4's capabilities.

  four error messages:
  - Unreachable destination (no known route, administratively forbidden, no transport-layer server at destination)
  - Packet too big
  - Hop limit exceeded
  - Invalid packet (think of errors with next headers, illegal options, etc.)

ICMPv6 (2/2)

- Informational messages:
  - Echo messages ("pinging")
  - Router-related messages
  - Neighbor discovery (replaces IPv4's ARP)
  - Redirect messages (tell a node about a better first-hop node toward a destination)
  - Mobility-management messages
- Note: ICMPv6 also handles to a great extent the dynamic allocation of addresses (autoconfiguration). Routers advertise prefixes; hosts choose the rest and test for conflicts on the local LAN. This largely replaces DHCP.

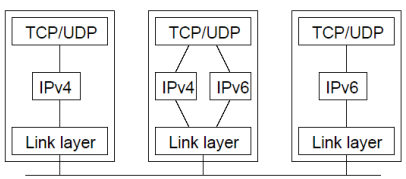
IPv6 - Security

- Illustrative example: there was a lot of discussion on where and how to incorporate security in IPv6:
  - If you are really concerned about security, would you trust anything else but end-to-end encryption? Question: What does this mean!?
  - Having security in the network layer offers a generally useful service to many applications. Those that don't want to use it, just ignore it.
  - Network-layer protocols have to run in every country. Some countries disallow crytosystems that the government can't decrypt easily.
  - Are the default crypto-algorithm good enough? For example, MD5 has been cracked
- This main issue here, as with almost every protocol, is to decide in which layer we should put functionality. There are many people who argue that only end-to-end solutions should be applied. The rest (i.e. general solutions) will never be good enough.

Interoperability

- Problem: IPv6 is not going to take over in a split second. The two protocols will have to live side-by-side for a very long time. There are three solutions:
  - Dual-stack techniques
  - Tunneling techniques
  - Network Address Translation and Protocol Translation (NAT-PT)

Dual – stack Techniques



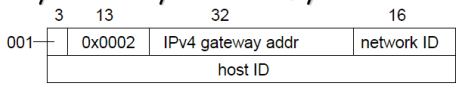
- Note: Dual-stack nodes have a separate IPv4 and IPv6 address.
- Note: You can also have complete dual-stack networks: all routers run both protocols. Requires a lot of redundancy (e.g. routing tables).

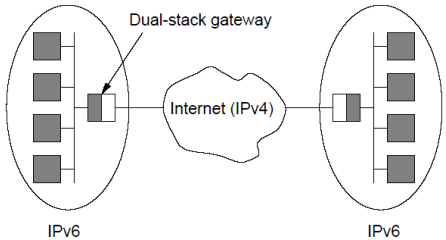
Tunneling Techniques

- Basic solution: simply forward IPv6 packets as payload encapsulated in IPv4 packets. Note that fragmentation and such can simply be applied to the IPv4 packet.
- Manual tunneling: Explicitly set up a tunnel between two dual-stack machines.
- Automatic tunneling: use IPv4-compatible IPv6 addresses and route these through a dual-stack network. Uses addresses with an all-zeros 96-bit prefix.

Automatic tunneling: 6to4

- Essence: encode the IPv4 address of your network's gateway into every address of your local IPv6 network.





NAT - PT

- Essence: translate the header info in an IPv4 packet to an IPv6 header and vice versa. Simply copy the payload field.
- Main drawback: hosts need to stick to their local network's address (as with 6to4)

Mobile IPv6

- Essence: works the same as with mobile IPv4, but some things can be done better when nodes speak IPv6:
  - There is no need for a foreign agent: a mobile IPv6 host can get its own care-of address.
  - The mobile hosts piggybacks its home address with every packet. The IP layer pretends that this is the true source address (useful for firewalls).
  - A mobile IPv6 host locally maintains a home-to-care-of address binding, and replaces the former with the latter when sending packets.

小结

- 网络层为传输层提供分组的逐跳转发并最终到达目的端的传送服务
  - 路由表的构造及选路算法是关键任务
  - 网络拥塞处理及QoS保证措施是“传送质量”保证的机制
- 网络互联
  - 由于MTU的不同，网络层还需要考虑分段处理问题
  - IP是事实上的网络互联标准，与ICMP等协议一起构成了Internet的互联网层协议集合
  - 下一代互联网核心技术IPv6