应用层

1. 应用层概念

两种主要方式：P2P、C/S。

进程间通信：socket（应用层和传输层，ip地址和端口号）

应用层协议定义：

Types of messages exchanged

Message syntax

Message semantics

Rules

开放式协议和专用式协议

Application需要哪些运输服务？

Data integrity

Throughput

Timing

Security

Internet transport protocols services

TCP和UDP

TCP（全双工）

可靠数据传输、流量控制、拥塞控制、面向连接

UDP（全双工）

无连接，不可靠

1. Web和HTTP

1、Web

Web由多个object组成。每个对象可以被存储到不同的Web服务器上。

Object可以是HTML文件、JPEG图像等。

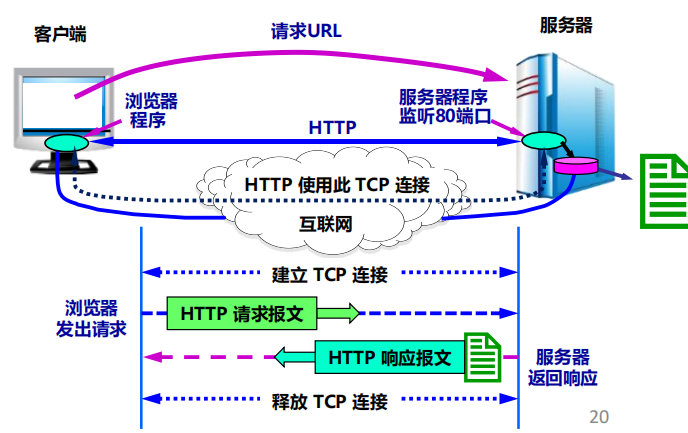
网页含有一个base HTML-file ，base HTML-file包括多个被引用的objects，每个objecs都可以通过URL寻址。

1. HTTP（Web客户端和Web服务器进行Web页面的请求、发送和接收的协议，用于传输）

·客户端：提出请求、接收（使用 HTTP 协议）和 “显示 "objects 的浏览器、。

·服务器： Web server发送（使用 HTTP 协议）对象以响应请求

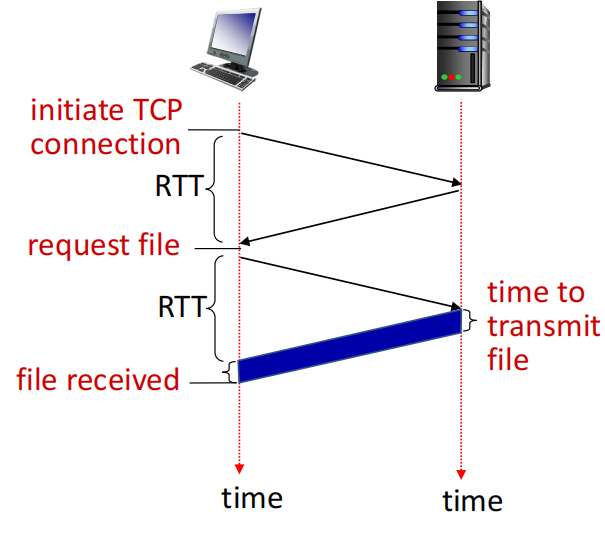
HTTP使用TCP协议。



Http的”stateless” ：server 不保存过去 client 的请求信息。

HTTP类型（Non-persistent HTTP和Persistent HTTP）

Non-persistent HTTP（http 0.9）



Non-persistent HTTP response time = 2RTT+ file transmission time

Persistent HTTP（HTTP 1.0/1.1）

同一个客户和服务器，在传输同一个WEB界面上的多个对象时，可以只建立一个TCP连接。

特别的，位于同一台服务器（域名一致）上的多个web界面，传输到同一个客户时，也可以只通过一个TCP连接来进行传输。

HTTP报文

①请求报文

第一行是request line，后面是head line，然后是entity body。

方法为GET时entity body 是空。

②响应报文

第一行是status line，后面是head line，然后是entity body（可以为空。报文主要部分，data data data data..）。

状态行：200 OK，404 NOT FOUND等

首部行里Content-length：发送数据的字节数（B）。

每个响应报文里封装一个对象。

**Cookies**

服务器创建的特殊标识符，用于跟踪用户

包含4个componets：

（识别码的形式放在这四个地方）

1. HTTP 响应报文的首部行
2. HTTP 请求报文的首部行

3) 保存在user host上、由浏览器管理的 cookie 文件

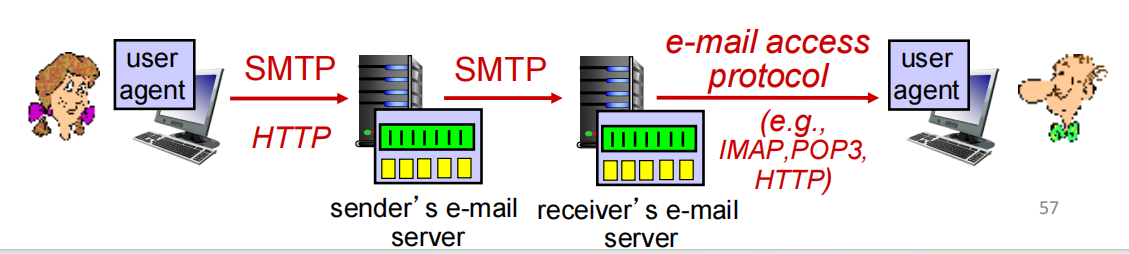
4) Web的后端数据库

Web caches（proxy server）

HTTP2/3

**Email**

**全过程都是TCP连接。**



SMTP（push）

TCP连接，使用持久连接。

三个阶段：

SMTP handshaking (greeting)

SMTP transfer of messages

SMTP closure

SMTP：在multipart message中发送多个对象

mail access protocols（pull）

都是TCP连接

POP3、IMAP、HTTP。

DNS

识别主机的两种方式：主机名和IP地址。

DNS：提供主机名到IP地址的转换服务。

包括两部分：

①由DNS server实现的分布式数据库

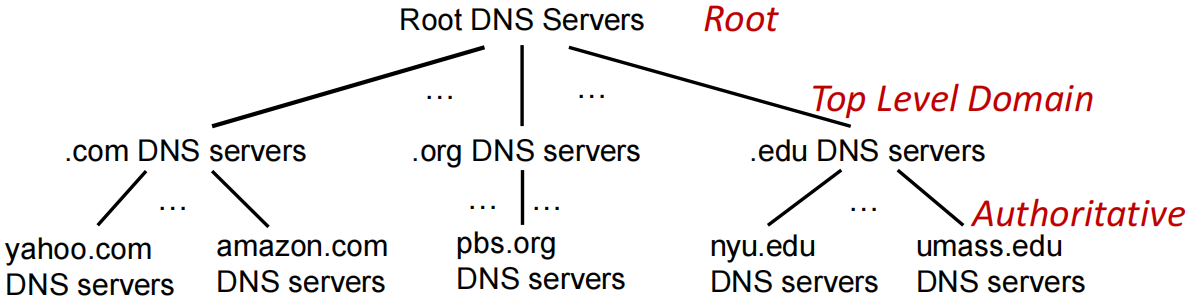
②使得主机能够查询这个数据库的应用层协议。（运行在UDP上）

Q: Why not 集中式 DNS?

 单点故障、流量、距离中央数据库太远、可维护性

A: 不能扩展。

DNS: a distributed, hierarchical database（分布式的、分层的数据库）



（local DNS不属于这个层级里面）

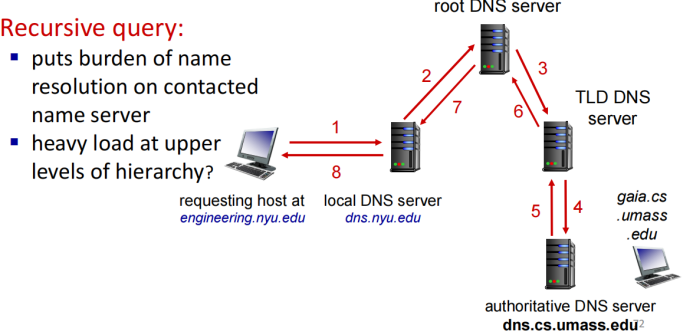
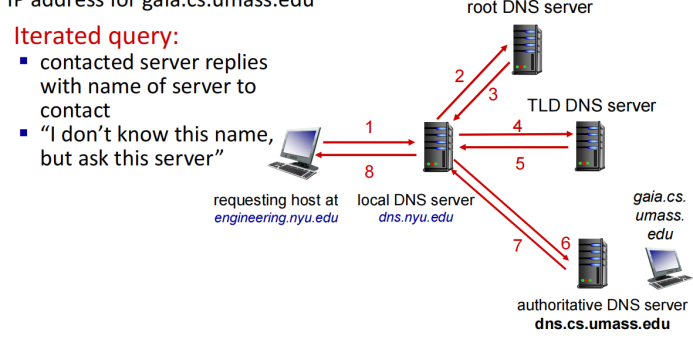
当主机进行DNS查询时，会先发送这个请求到local DNS。

然后有两种查询方式，

迭代查询（Iterated query）和递归查询（Recursive query）。通常从本机到local DNS是递归查询，其余查询是迭代查询。

下面两张图从主机到local DNS都是递归查询。

第一张图其余查询都是迭代查询，第二张图其余查询都是递归查询



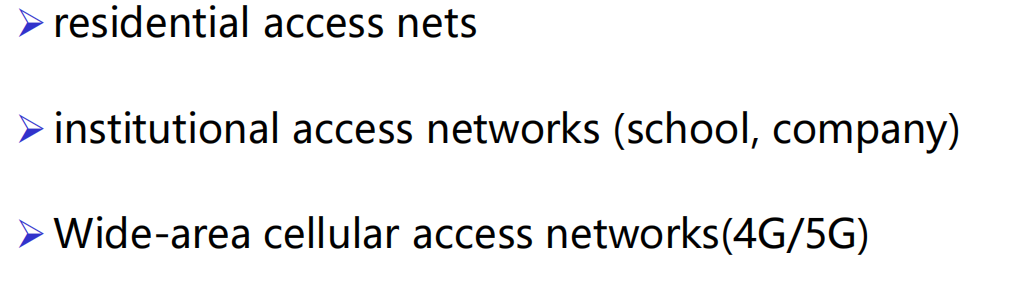
DNS缓存

DNS记录和报文

概论

**网络边缘** 包括hosts和接入网

接入网包括 住宅接入网、机构接入网和广域蜂窝接入网



住宅接入网

HFC（混合光纤同轴电缆）：FDM, asymmetric（指上传下载速率非对称），各家庭共享接入网络。

ADSL（非对称数字用户线）：FDM, asymmetric（指上传下载速率非对称），使用existing telephone line（电话线）

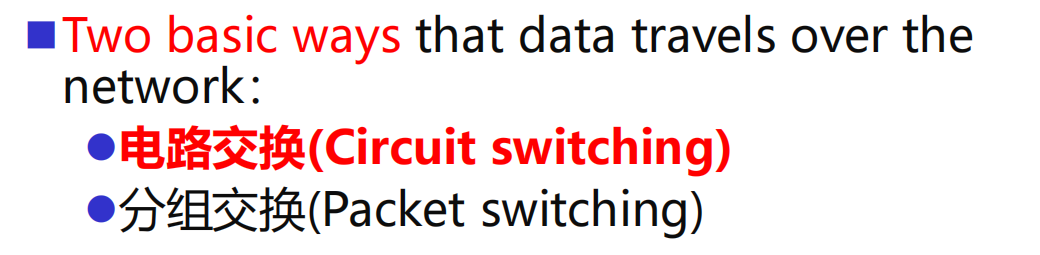
FTTH（光纤入户）

机构接入网

有线、无线和交换机、路由器。

广域蜂窝接入网

**网络核心**



电路交换：两节点在传输前建立起一个专用的传输通道，预留所需要的资源。

FDM和TDM。

分组交换：long messages被分成单独的小包 (packets)并被独立的发送。

分组交换的核心是存储转发。指每个packets必须完整的被当前节点接收后才被转发到下个节点。（传输该分组的第一个比特之前，必须接收完整个分组）

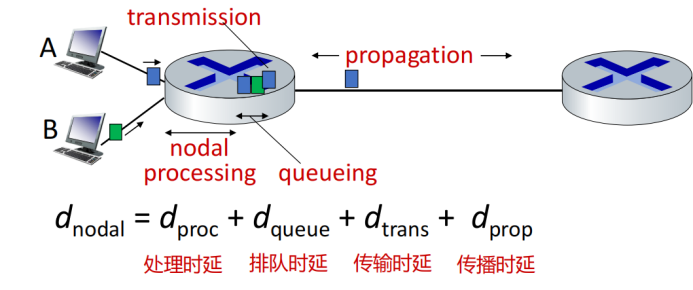
分组交换和电路交换的对比：

电路交换实时性更强，但是资源占用更多，更复杂，成本更高，效率较低。

分组交换实时性差（端到端时延不可预测），提供更好的带宽共享，更简单，更有效，成本更低，性能更好。

网络的网络

**时延**



一般只考虑传输时延（trans）和传播时延（prop）

传输时延：

L表示一个分组的长度，R表示链路传输速率，比如100Mbps的以太网链路，R=100Mbps。

Trans delay=L/R

传播时延：

S表示线路的长度（距离），V表示分组在传播时的速率，于是：

Prop delay=S/V

端到端时延（end -end）

N×节点时延=N×（处理时延+排队时延+传输时延+传播时延）

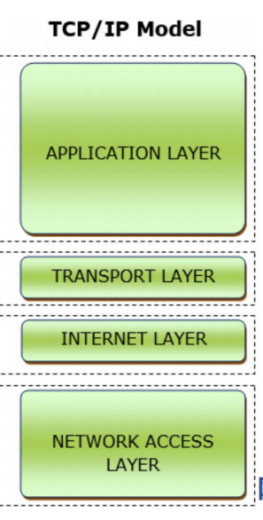
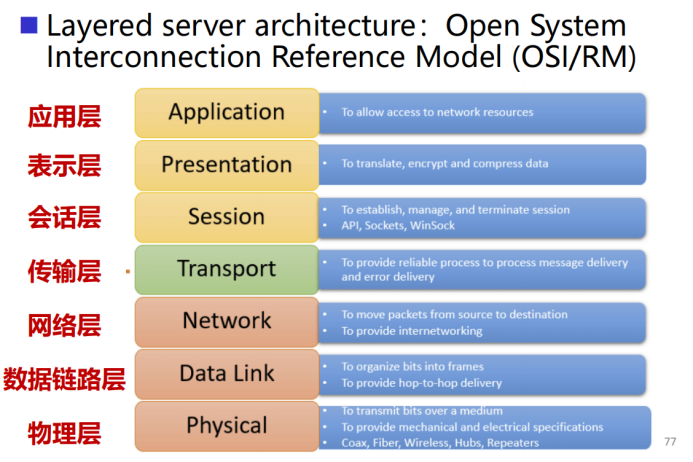
其中N是链路段数，于是一共有N+1个节点。

吞吐量（throughput）

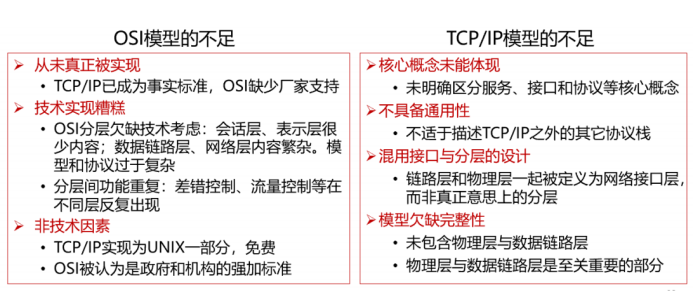
=min{R1，R2}。R1和R2分别为当前节点进入和输出的两条链路的传输速率。

协议分层

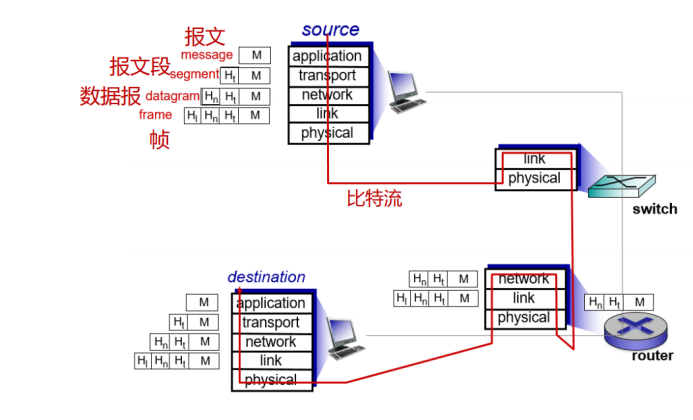
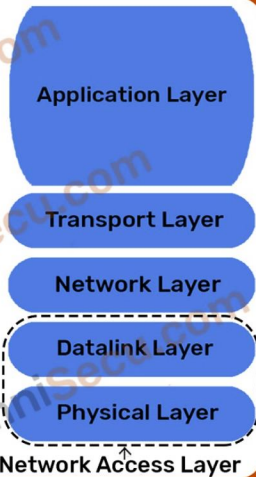
OSI模型和TCP/IP模型



两者的对比和不足。



我们所学习的是五层模型，即**应用层-传输层-网络层-数据链路层-物理层**



Protocol vs Service

**协议：对等层之间传输数据时的规定（水平的）**

**服务：下层对上层可以提供的功能等支持（垂直的）**

**传输层**

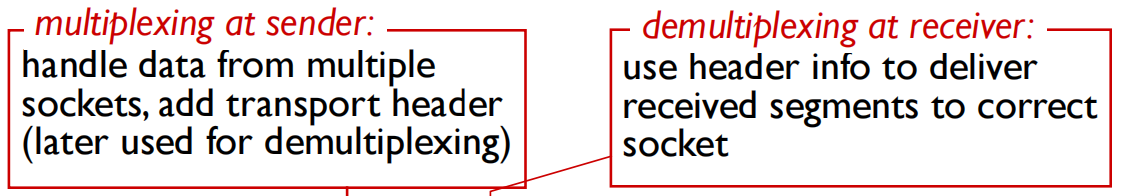
传输层是进程和进程之间的通信，网络层是主机和主机之间的通信

两种主要的传输协议：TCP和UDP

Multiplexing(多路复用)and Demultiplexing(多路分解)

多路复用是给数据加上头部形成报文段。

多路分解是根据头部来分配到合适的socket来进行传输。



传输层用IP地址和端口号来标识报文段（segment）

网络层用IP地址来标识数据报（datagram）

UDP是无连接的，即对于所有的客户端都使用同一个socket进行数据收发。

Socket用来连接应用层和传输层，使得应用层可以使用传输层的协议和服务。

UDP socket只有目的ip和目的端口号，所以UDP多路分解只需要目的IP和目的端口号

TCP socket是四元组，目的ip，目的端口号，源ip，源端口号，所以TCP多路分解需要这个四元组。

UDP：

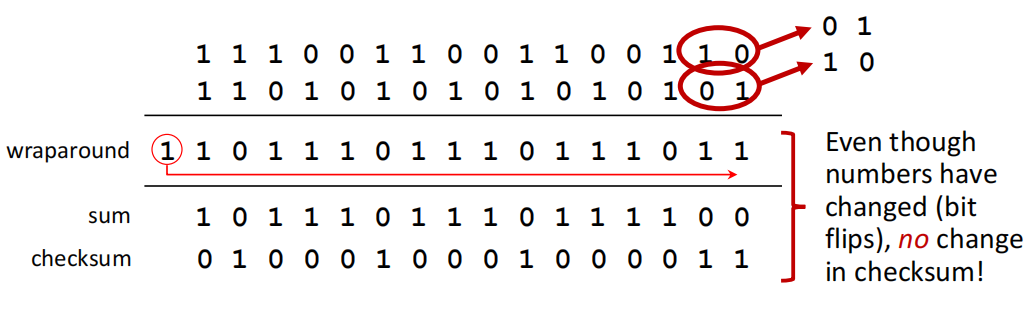
简单，不可靠（易丢失），无序交付；无连接；“尽力而为”（best-effort）

**UDP校验和计算**

两个数按位相加，最高位如果有溢出，加到最后面一位，再全部取反。

比如：1011＋1000=1 0011，那么把溢出的1加到最后一位，即0011+0001，是0100，再去取反，是1011

不能实现纠错，可以检验，但是也会失效（两个加数的同一位同时发生错误，即翻转）



可靠传输rdt

Rdt1.0-rdt3.0，回退N步GBN，选择重传SR

TCP：

Point to point；

可靠传输，按序交付，字节流；

拥塞（congestion）控制；

流量（flow）控制；

面向连接；

全双工通信（full duplex data）；

累积确认ACK；

流水线（pipelining）

TCP连接的状态完全保留在两个端系统中，与中间链路无关。

MSS（报文段的最大size，也就是一次传输最多能传输的字节数）

TCP相关：

Seq（Sequence numbers）:报文段中第一个字节的序号。每个字节占一个序号。

ACK：希望接下来从对方那里收到的报文段的 Seq。

采用累计确认的方式：**通过携带一个累计的确认号，来告诉对方，自己已经接收到的最大的连续的seq，并期望获得的下一个报文段的seq**

比如对方发送了seq为1000，2000，3000，4000的报文段，显然包含的字节的序号分别是1000-1999，2000-2999，3000-3999，4000-4999.

如果我收到1000的报文段后，我会发送ack=2000；如果我收到了1000和3000，那么我会发送ack=2000；如果我收到了1000，2000，3000，那么我会发送ack=4000.这就是累计确认。

EstimatedRTT、DevRTT

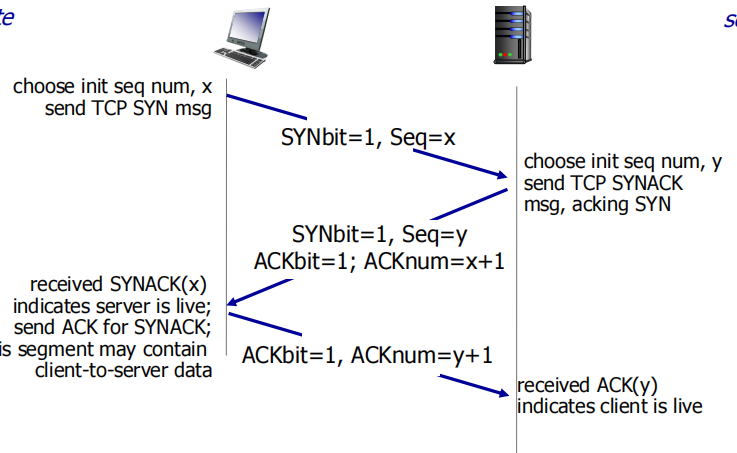
TCP可靠数据传输

超时重传、快速重传。

TCP流量控制

接收窗口rwnd：B在发给A的报文段中，告诉A自己的rwnd是多少，于是A可以根据rwnd来控制发送的流量。

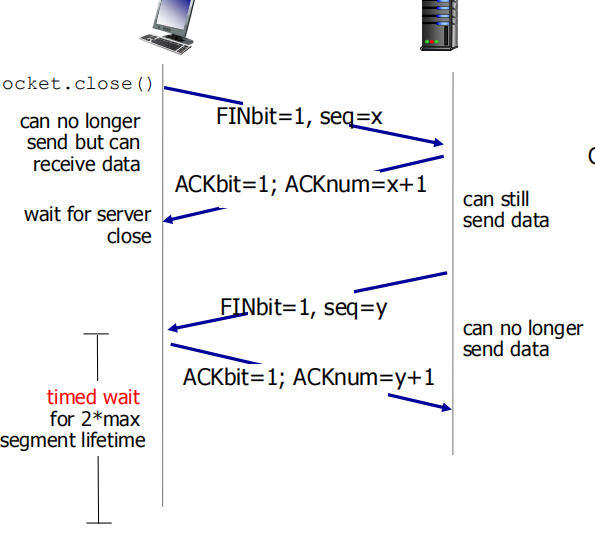
TCP连接和释放——三次握手和四次挥手



要注意的是客户端发送的第一个报文ACKbit=0，所以没有ack。

服务器发送的第一个报文（SYNbit=1，ACKbit=1）。

客户端发送的第二个报文（SYNbit=0，ACKbit=1）的seq=x+1，图中没有标出。



第二次挥手和第四次挥手的FINbit=0.

四次挥手的ACKbit其实都是1.图中没有标出。

TCP拥塞控制

发送方已经发送但尚未得到确认的数据量<=min{cwnd，rwnd}。

AIMD:加性增，乘性减。

①慢启动：每过一个rtt会将cwnd变成两倍（实质是收到一个ack就会增加1个MSS的cwnd）。

②当慢启动达到慢启动阈值后，就会进入拥塞避免阶段。每个RTT后 cwnd加1。

③当出现重传时，结束拥塞避免。

如果是超时重传，会把慢启动阈值ssthresh设置为cwnd/2，接着将当前cwnd设置为1，进入慢启动。

如果是快速重传，会把慢启动阈值ssthresh设置为cwnd/2，接着将当前cwnd设置为ssthresh+3，进入快速恢复。

④快速恢复阶段，每当再收到一个冗余ack后cwnd+1。（一般不会出现，因为是在收到三个冗余ACK后进入快速恢复阶段，并且这之后立刻重传这个报文段，那么接下来就会收到新的ack。）

当出现超时重传时，会把慢启动阈值ssthresh设置为cwnd/2，接着将当前cwnd设置为1，进入慢启动。

当收到新的ACK时，进入拥塞避免状态。每个RTT后cwnd＋1。直到再出现重传。

（一般不用考虑太多，只这样想：收到3个冗余ACK后进入快速恢复阶段，ssthresh变成cwnd/2，当前cwnd降到cwnd/2+3，然后就是拥塞避免阶段）

网络层

数据平面：本地的，每个路由器内部的，确定到达路由器输⼊端⼝的数据报如何转发到路由器输出端⼝。

控制平面：全局的，确定数据报如何从源主机到达目的主机。

Internet的服务模式（service model）：best-effort service

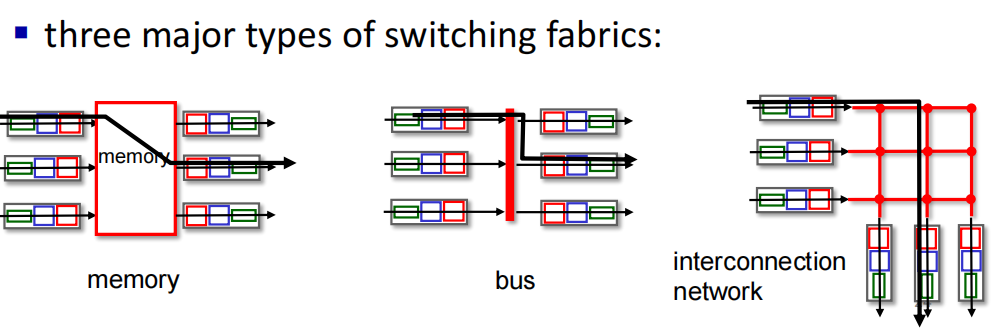
网络层的两种服务：虚电路服务和数据报服务。

虚电路服务类似于电路交换，有连接的建立。

数据报服务无连接的建立，例如IP。

最长前缀匹配原则：当查找给定⽬标地址的转发表条⽬时，如果有多个能匹配上的，使⽤最⻓的那个与⽬标地址匹配的地址前缀。

路由器内部交换结构（Switching fabrics）：



以及内部的一些调度策略

IP地址

IP数据报分片（IP Datagram Fragmentation ），MTU是分片后一个数据报最大的长度。

比如原来一个ID为422的数据报是2400字节，MTU是700字节。

（默认头部是20字节）

那么IP负载一共是2380，每个片的最大负载是680，需要4个片。

前三个片都680+20=700，标志位为1；最后一个片是340+20=260，标志位为0.

并且这四个片的ID都是422.

一个Offset=680÷8=85，从0开始计。

那么第一个片的offset是0，第二个就是85，第三个就是170，第四个就是255.

网络号+子网号+主机号。根据子网掩码来区分，子网掩码为0的地方是主机号。

目前使用CIDR编址。跟上述差不多，一些相关名称不同。

子网划分。子网内全0和全1不能用。比如10.6.6.0/24这个子网，就只有254个IP地址分给主机。10.6.6.0和10.6.6.255都不能用。

DHCP：动态主机配置协议——主机从它所在网络获取自己IP地址的方式。

告诉主机的不仅是是主机自己的IP地址，还有默认网关（也就是主机的第一跳)的ip地址、DNS服务器的名称和IP地址、网络掩码。

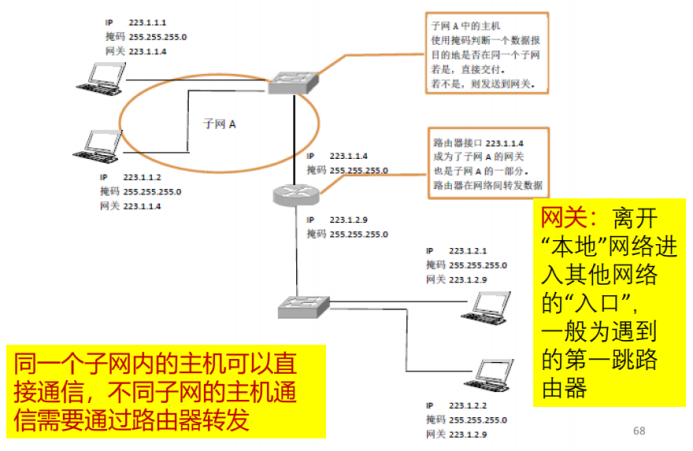
DHCP过程：

DHCP Discover：客户端发送广播来寻找DHCP服务器。

DHCP Offer：DHCP服务器为客户端提供IP地址和其他消息。

DHCP Request：客户端选择一个DHCP服务器并请求使用。

DHCP ACK：服务器确认并分配IP地址后，发送ACK。



NAT（网络地址转换）：本地网络中所有主机共用一个IP地址，但是不同的端口号。

即这个网络中的主机在发送报文段时，源IP为这个网络的统一IP；

接收报文段时，目的IP也是这个统一IP。

SDN（软件定义网络）：

基于流转发、

控制平面和数据平面分离、

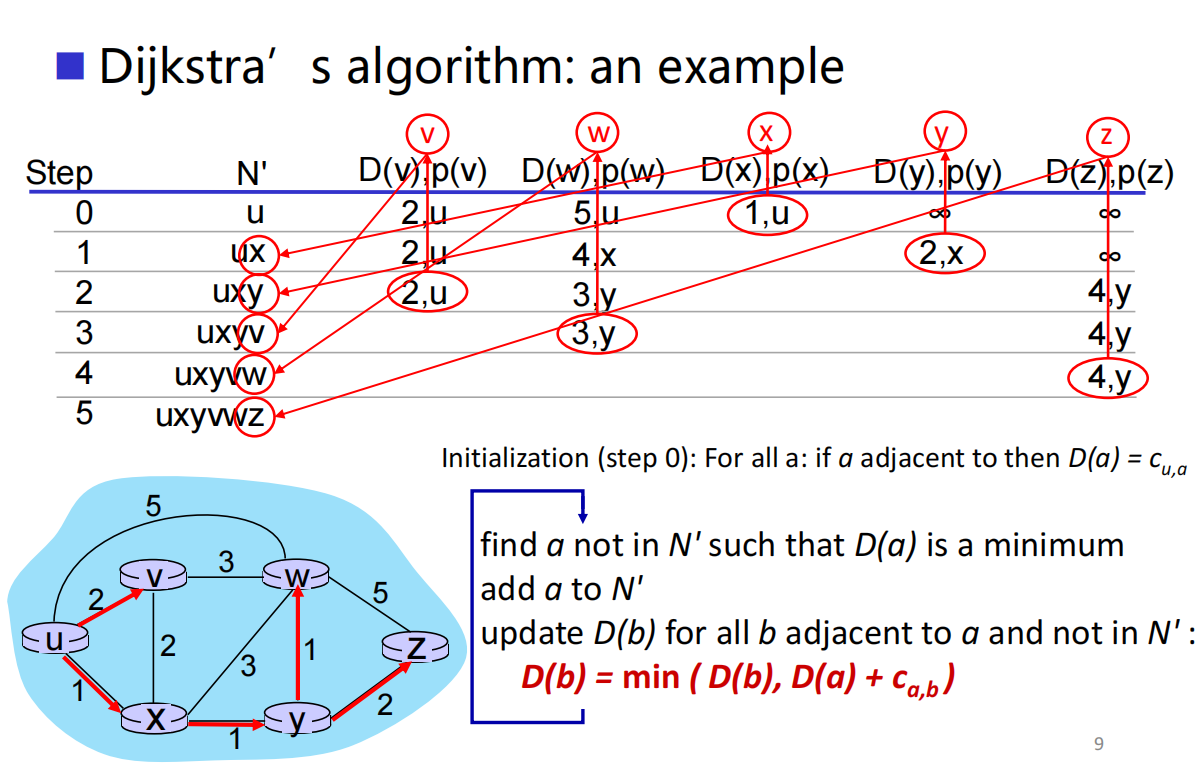
将控制函数从设备中提取出来整合到一起、

可编程网络。

控制平面

路由算法

Dijkstra，一个源节点，到多个目的节点。



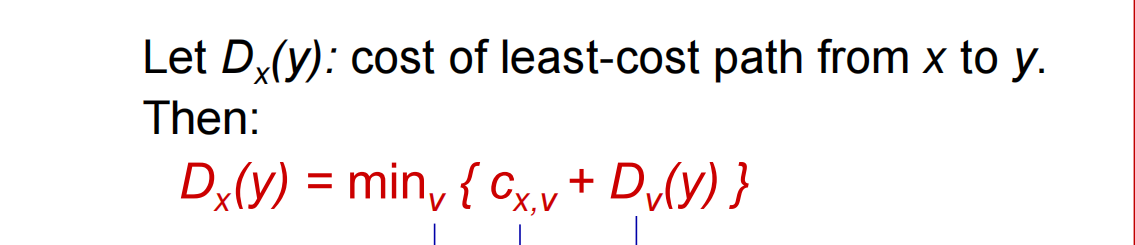
D（x）是到x节点的最短路径长度，p（x）是该最短路径到达x之前的前向节点。

每一步不断把上一步中的最短路径的目的节点加入到已有集合中。

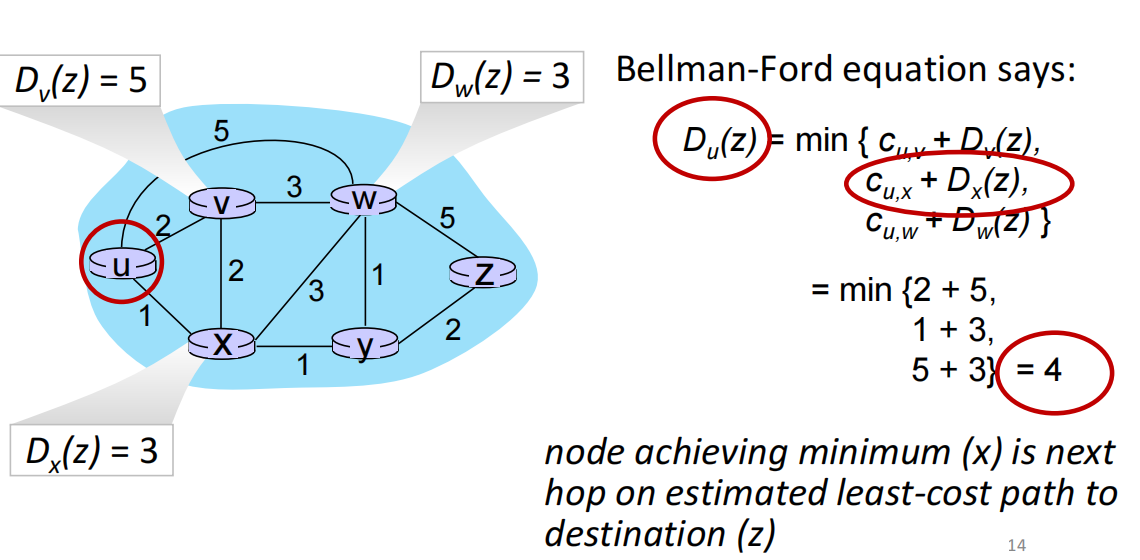
路由震荡Route oscillations：

链路的cost与方向性有关，并且是依赖流量的（volume-dependent）

Distance vector，基于Bellman-Ford。



也就是x到y的最短距离=min{ x和某节点v的直接距离+v到y的最短距离 }



每个时间步里，每个节点都要做：

接收来自邻居的距离向量（上一个时间布计算出来的结果）、计算新的自己的距离向量、发送自己的距离向量给邻居（邻居会在下个时间步收到）。

在初始化（时间步为0）时，每个节点只知道与自己直接相连的节点的距离。

毒性逆转

如果X到Y的最短路径经过了u，x会告诉u，这个最短距离为正无穷。

并且在下一个时间步开始时，u会更新自己的DV：X到Y的最短距离为正无穷。

AS：自洽系统，又称domains（域名）。将很多很多的路由器聚合到一起的系统。

AS内路由算法：RIP、EIGRP、OSPF。

OSPF：开放最短路径优先调度

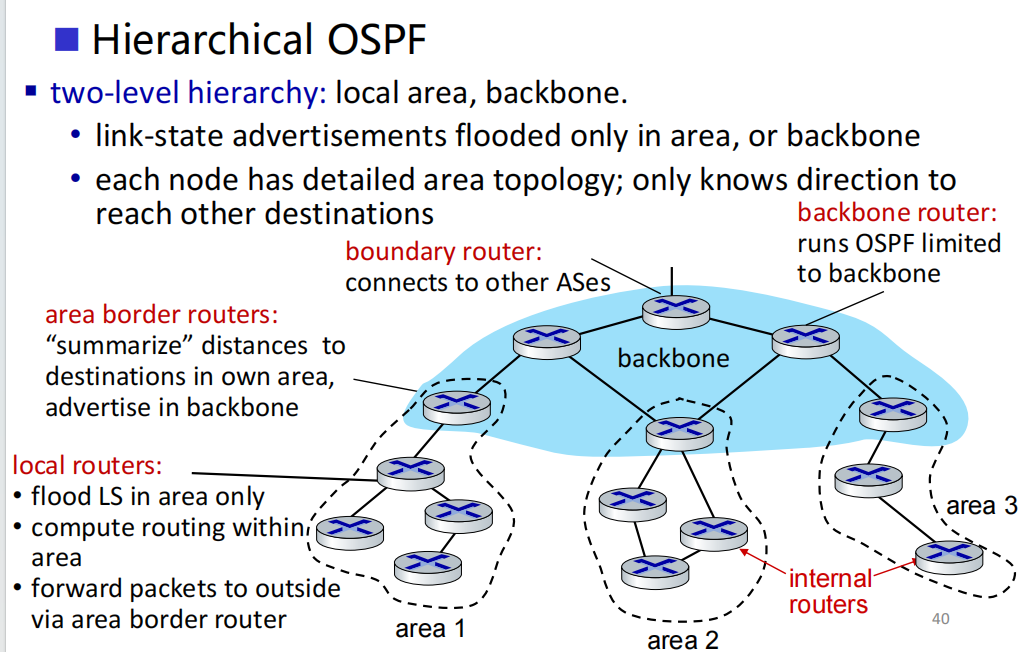
特点：

①直接使用IP在整个AS中来通告OSPF信息（而不是使用TCP等协议）

②OSPF信息会包含链路的多种信息（带宽、时延等）

③每个路由器都包含整个AS的拓补网络，使用Dijkstra算法。

分层OSPF：（local router 和backbone router)



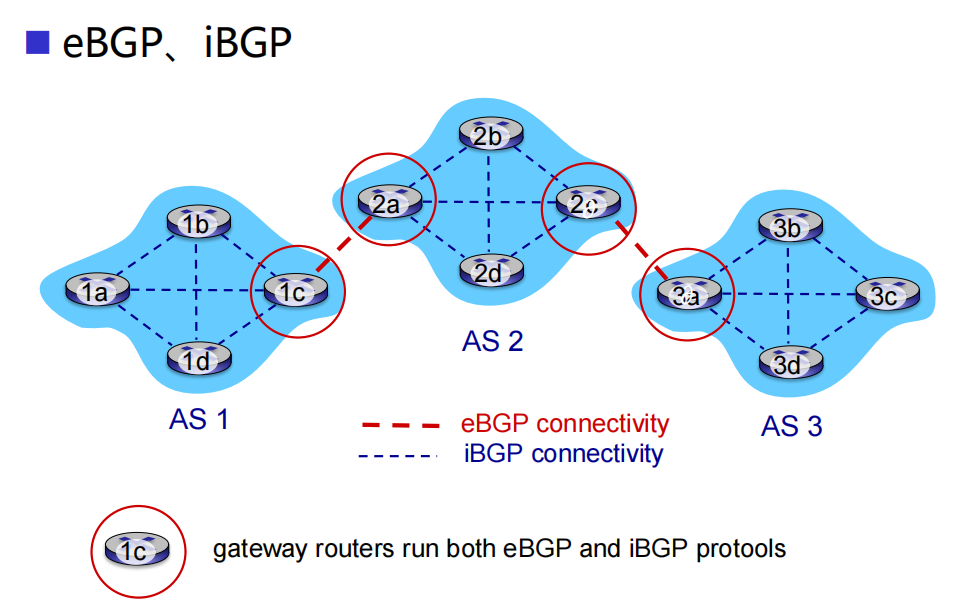
AS间路由算法：BGP

基于**策略（policy）**的路由协议

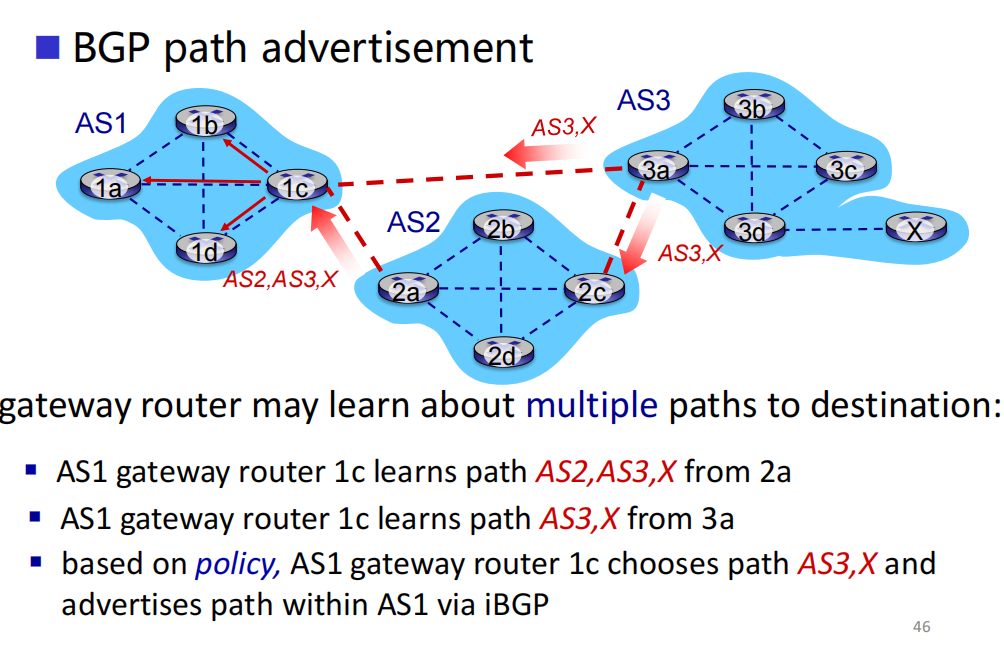
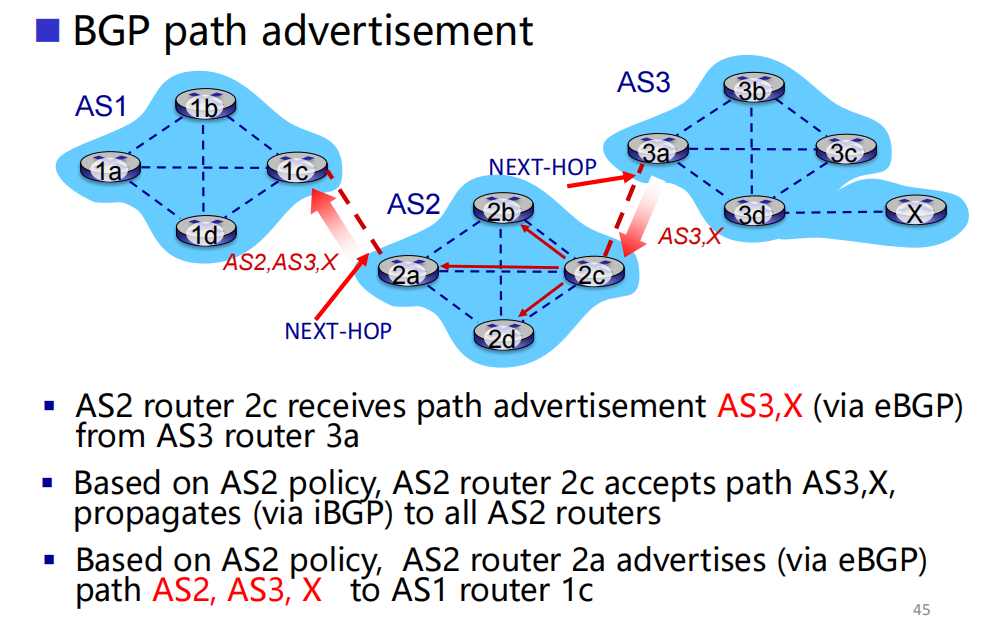
**使用TCP来传播可达性信息（永远是向源路由器传递目的路由器的可达性信息，即这个可达性信息的目的路由器，是想要发送数据出去的路由器）**

eBGP是获取其他邻近AS的可达性信息。（只在网关路由器上）

iBGP是将这个可达性信息在本AS内传播。（在网关路由器和内部路由器上）



BGP信息：前缀＋下一跳信息。前缀指已经经过的AS，下一跳指使用特定路由器来到下一跳的AS.



上图例子是为了获得X的可达性信息。

X在AS3中通过OSPF告诉3a自己的可达性信息（？），

2c收到来自3a的相关信息（通过eBGP）：AS3,X。（距离当前路由器越近的AS放在左边，最右边是希望获取可达性信息的目标路由）

基于AS2的BGP策略，2c通过iBGP在AS2内部传播这个可达性信息：AS3,X（已经经过的AS的信息）。于是所有的AS2内部路由器都收到了这个信息。

基于AS2的BGP策略，2a通过eBGP将这个信息传播给1c：AS2,AS3,X（已经经过的AS的信息）。

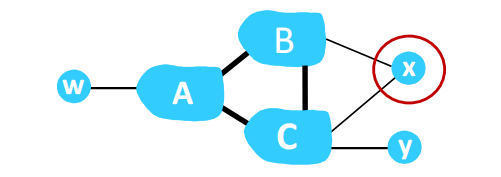
之后再根据AS1的BGP策略，在AS1内部传播这个可达性信息。

内部某路由器（比如1a）知道了可以通过某网关路由1c到达X，（这里假定只能通过1c来到达X，而没有其他的网关路由可以到达X）那么就会在1a内部路由表添上<X，n>（假定经过OSPF，1a知道到达1c的最短路径，是从1a的n接口发出的。这里要注意n是1a的接口，因为是1a要发送信息到X，不要搞错方向）

再考虑如果可以通过多个网关路由来到达X，比如考虑2d，它通过iBGP知道：可以通过2a或者2c到达X，那么我们就根据Hot Potato算法，来选择最早能离开当前AS的网关路由器（选择这样的网关路由：从2d到这个网关的cost最小。要注意的是算从2d到2a或者2c的cost是通过OSPF实现的）

假如这里选择的是2a，那么同上，在2d的内部路由表添上<X，m>，其中m是经过OSPF，2d知道到2a的最短路径是从2d的m接口发出的。

基于策略：



如果B不想要通过x到达C，那么x就不会向B传达C的可达性信息。

和OSPF对比：

OSPF是将AS内部的可达性信息，在AS内部进行传递，使得AS内部的各路由器知道到内部其他路由器的最短距离。

eBGP是将邻近其他AS内子网的可达性信息传递到本AS的网关路由器上，

iBGP是将这个可达性信息在本AS内传播，使得本AS内的路由器获取到其他AS内的子网可达性信息。

ICMP总线控制报⽂协议

在IP数据报中携带着ICMP报文。（IP数据报就是指网络层的数据报datagram）

①错误报告

②echo request/reply（Ping、Traceroute）

链路层