计算机网络和因特网（约5分）

• 什么是主机/端系统?

运行了各种应用程序并互相连接的计算设备，包括手机、电脑、服务器等等。（主机强调功能，端系统强调位置）

• 什么是分组?

当一个端系统要向另一个端系统发送数据时，发送端系统将数据分段，并为每段加上首部字节。由此形成的信息包叫分组。

• 什么是协议?

协议定义了在两个或多个通讯实体之间交换的报文格式和顺序，以及报文的发送/接收或其他事件所采取的操作。

• 客户、服务器?

不严格地说，客户通常是桌面PC、移动PC和智能手机等；
服务器通常是更为强大的机器，用于储存和发布Web网页、流视频、中继电子邮件等。

• 电路交换和分组交换的对比?

主要是**预不预定**！！！

电路交换有可预见的稳定的性能，而一旦电路建立，交换将变得简单而快速；
分组交换的性能优于电路交换，电路交换不考虑需求，预先分配了传输链路，这使得已分配而不需要的链路时间并未被利用；分组交换按需分配链路，链路传输能力将在所有需要在链路上传输的分组的用户之间逐分组地被共享。

• 节点处理时延、排队时延、传输时延、传播时延?

（1）处理时延 Process dproc

检查分组首部和决定将该分组导向何处所需要的时间是**处理时延**的一部分。处理时延也包括其他因素，如检查比特级别的差错所需要的时间——该差错出现在从上游节点向路由器 A 传输这些分组比特的过程中。高速路由器的处理时延通常是微秒或更低的数量级。在这种节点处理之后，路由器将该分组引向通往路由器 B 的链路之前的队列。（在第 4 章中，我们将研究路由器运行的细节。）

（2）排队时延 Waiting dquene

在队列中，当分组在链路上等待传输时，将经受**排队时延**。一个特定分组的排队时延长度将取决于先期到达的正在排队等待向链路传输的分组数量。如果该队列是空的，并且当前没有其他分组正在传输，则该分组的排队时延为 0。另一方面，如果流量很大，并且许多其他分组也在等待传输，该排队时延将很长。我们将很快看到，到达分组期待发现的分组数量是到达该队列的流量的强度和性质的函数。实际的排队时延可以是毫秒到微秒量级。

（3）传输时延 Submit 距离无关，dtrans

假定分组以先到先服务方式传输——这在分组交换网中是常见的方式，仅当所有已经到达的分组被传输后，才能传输刚到达的分组。用 L 表示该分组的长度，用 R 表示从路由器 A 到路由器 B 的链路传输速率。例如，对于 10Mbps 的以太网链路，速率 $R=10$Mbps；对于 100Mbps 的以太网链路，速率 $R=100$Mbps。**传输时延**是 L/R。这是将所有分组的比特推向链路（即传输，或者说发射）所需要的时间。实际的传输时延通常在毫秒到微秒量级。

（4）传播时延 Propagate dprop ~~Transport~~

一旦一个比特被推向链路，该比特需要向路由器 B 传播。从该链路的起点到路由器 B 传播所需要的时间是**传播时延**。该比特以该链路的传播速率传播。该传播速率取决于该链路的物理媒介（即光纤、双绞铜线等），其速率范围是 $2\times10^8\sim3\times10^8$m/s，这等于或略小于光速。该传播时延等于两台路由器之间的距离除以传播速率。即传播时延是 d/s，其中 d 是路由器 A 和路由器 B 之间的距离，s 是该链路的传播速率。一旦该分组的最后一个比特传播到节点 B，该比特及前面的所有比特被存储于路由器 B。整个过程将随着路由器 B 执行转发而持续下去。在广域网中，传播时延为毫秒量级。

• 吞吐量?

端到端。bit/s => bps

瞬时吞吐量

平均吞吐量

瓶颈链路

• 5层因特网协议栈？每层的作用？（术语：报文、报文段、数据报、帧）

应用层(application)

支持网络应用

应用层报文

一个端系统和另一个端系统中的应用程序交换信息的分组；

运输层(transport)

进程与进程数据传输、网络应用的数据传输

运输层报文段

运输层的分组；

应用层报文+运输层首部信息；

网络层(NetWork)

根据ip协议进行寻址、控制拥堵、端到端的数据传输

网络层数据报

网络层的分组，把运输层协议向网络层传输；

运输层的报文段和源和目的端系统地址等网络层首部信息；

数据链路层(Data Link)

负责相邻路由器/主机之间的直接的数据传输

链路层帧

对网络层分组+链路层首部信息，封装后的链路层分组；

物理层(Physical)

传输比特

应用层（约10分）

- **客户-服务器体系结构**

客户：

需要时启动，

是连接的初始化者，"speaks first"

有动态的IP地址

如：浏览器的客户端、电子邮件阅读器

服务器：

守护进程，保持运行

提供被用户需要的服务

有固定的IP地址

如：

网站服务器发送被需要的网页

邮件服务器传递邮件

- **对等体系结构**

没有一个总时运行着的服务器

任意两个端系统是直接交流的

对等体从其他对等体获取服务，并向其他对等体提供服务

有自我伸缩性

新的对等体在带来新的服务需求的同时带来了新的服务能力

对等体是间歇性连接，并改变IP地址

这导致了高拓展性，但是难以管理

如：

Gnutella, BitTorrent, Skype

- **TCP和UDP的区别?**

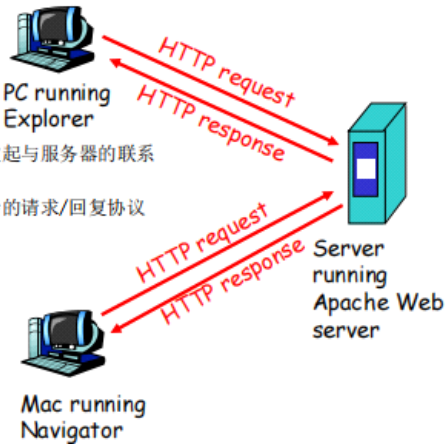
TCP 比较可靠

UDP 尽力而为的服务不能保证可靠

• 什么是HTTP?

Hyper Text Transport Protocol 超文本传输协议

- Client-server architecture 服务器始终打开并“知名”
  - Server is "always on" and "well known"
  - Clients initiate contact to server 客户端发起与服务器的联系
- Synchronous request/reply protocol 同步的请求/回复协议
  - Runs over TCP, Port 80
- Stateless
- ASCII format
  - Before HTTP/2



Web：

基础设施：

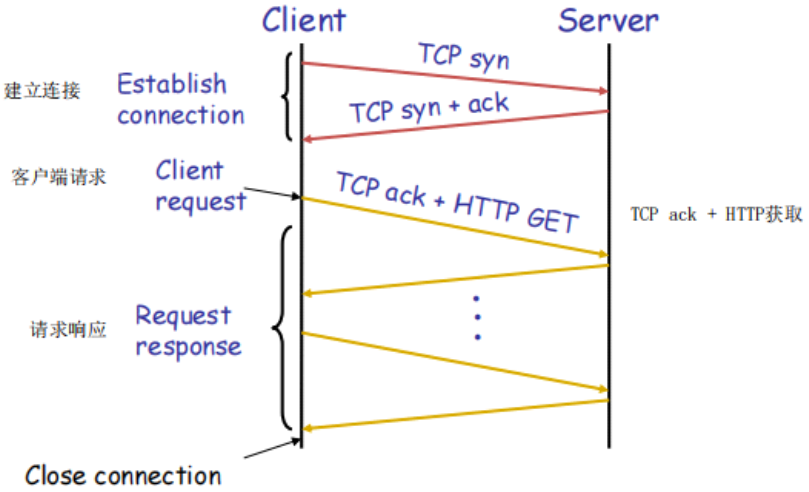
客户 & 服务器（DNS、CDN、Datacenters）

内容：

URL：名字 Uniform Resource Locator

HTML：内容

• HTTP的请求-响应行为?



还可以采取持续的连接

• URL由哪两部分组成?

存放对象的服务器主机名+路径名

<protocol>://<host>:<port>/<path>?query_str

- **因特网电子邮件的三个组成部分？每部分的主要作用？**

用户代理

允许用户阅读、回复、转发、保存和撰写报文。

邮件服务器（体系结构之核心）

邮箱管理和维护发送给用户的报文

维护一个不能交付的邮件消息的报文队列

邮件服务器之间的SMTP协议发送邮件消息

SMTP 简单邮件传输协议（主要协议）

使用TCP可靠数据传输服务

分两个部分，运行在发送方邮件服务器的客户端和运行在接收方邮件服务器的服务器端

每个报文都按照7比特ASCII码来进行编码

- **SMTP的基本操作？ （A向B发送一条报文的过程）**

！！没有中间服务器！！

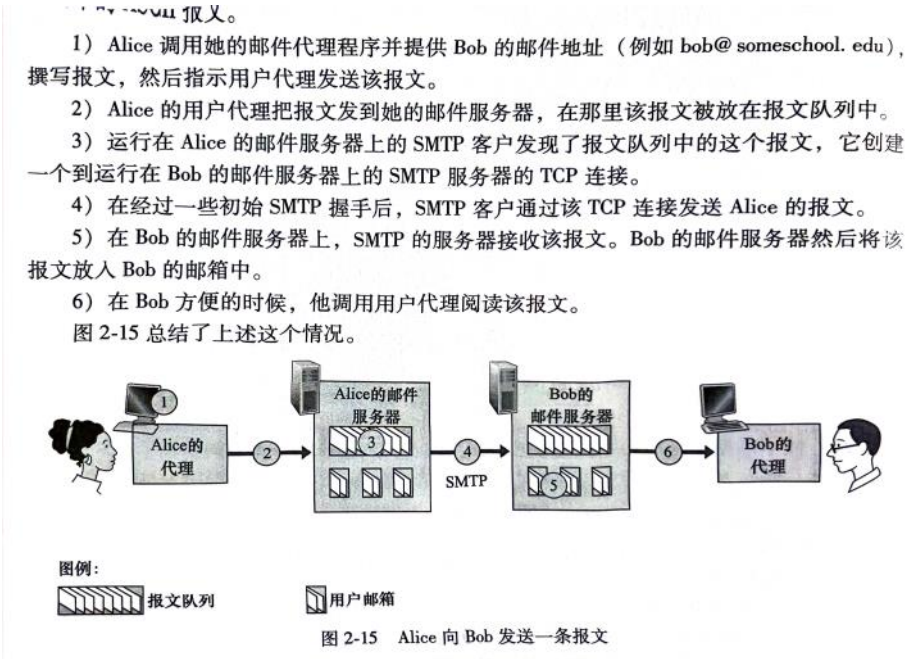


图 2-15 Alice 向 Bob 发送一条报文

Stage 1

电子邮件从本地**用户代理**转到**本地SMTP服务器**

用户代理充当SMTP客户端

本地服务器充当SMTP服务器

Stage 2

电子邮件由**本地服务器**中继到**远程SMTP服务器**

本地服务器现在充当SMTP客户端

Stage 3

远程用户代理使用邮件访问协议来访问远程服务器上的邮箱

使用 POP3 or IMAP4 or HTTP 协议

• 推协议、拉协议?

拉协议（Pull Protocol）：HTTP、FTP、POP3、IMAP

- 客户端必须主动向服务器发出请求，以获取所需的数据。
- 服务器仅在接收到客户端的请求后才会响应并发送数据。

推协议（Push Protocol）：SMTP

- 服务器可以主动向客户端发送数据，无需等待客户端发出请求。
- 客户端可以被动地接收来自服务器的数据更新。

• 域名系统的作用?

将 **主机名** 解析为 **IP 地址**

DNS 是

一个由分层的DNS服务器实现的分层式数据库

一个使得主机能够查询分布式数据库的应用层协议。

在许多名称服务器的层次结构中实现的**分布式数据库**

与**应用层协议主机**和名称服务器进行通信，以解析"域名"

负载平衡：对一个服务器名设置一组IP地址、防止频繁访问一个服务器

• DNS服务器的层次结构?

主要想法：层次结构

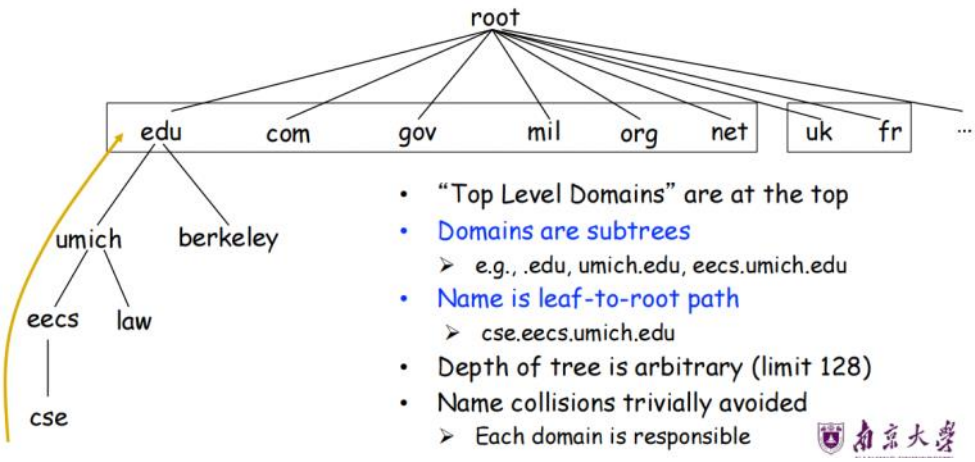
三个相互交织的层次结构

层次结构命名空间：而不是原始的平面命名空间

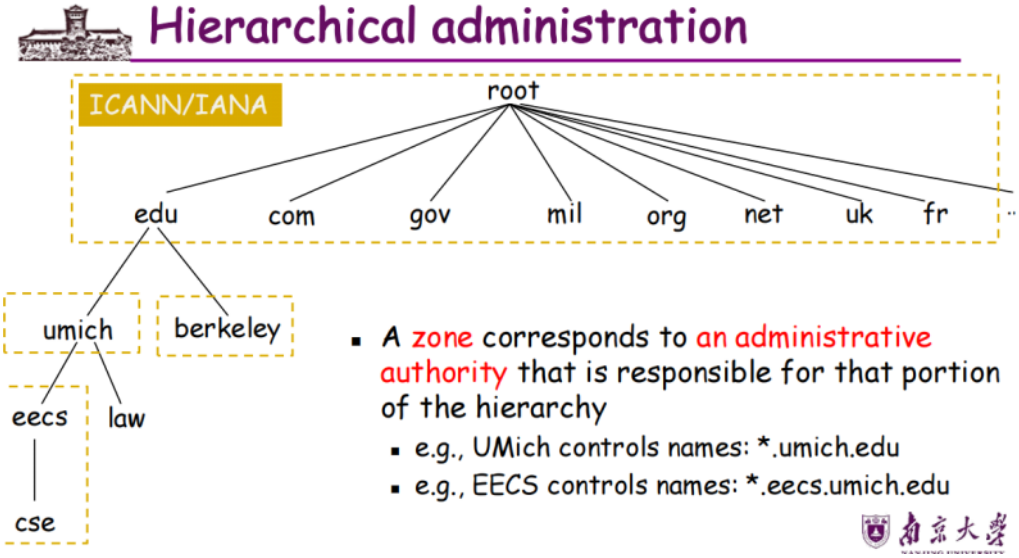
层次管理：而不是集中

分布式服务器的层次结构：而不是集中式存储

（A）层次结构命名空间：



（B）层次管理：



（C）分布式、层次数据库：

1. 根DNS服务器：返回TLD服务器的IP映射
2. 顶级域(TLD)DNS服务器 (com/edu(usa)/org/cn/uk)

   提供权威DNS服务器的IP地址
3. 权威DNS服务器

   组织的DNS服务器，提供主机名到IP地址的映射

   可以被组织或服务器提供商维护
4. 本地名称服务器

   - 不严格属于层次结构
   - 每个ISP（互联网服务提供商 | 住宅社区ISP，公司，大学等）都有一个，
   - 当主机进行DNS查询时，查询会被发送到它的本地DNS服务器
     - 有最近的名称到地址转换对的本地缓存（但可能已过时！）
     - 充当代理，将查询转发到层次结构中

每个服务器都存储一个（小的）总DNS数据库的子集

一个权威DNS服务器存储"资源记录"所有其管辖的DNS名称的域，

每个服务器需要知道其他服务器负责层次结构的其他部分

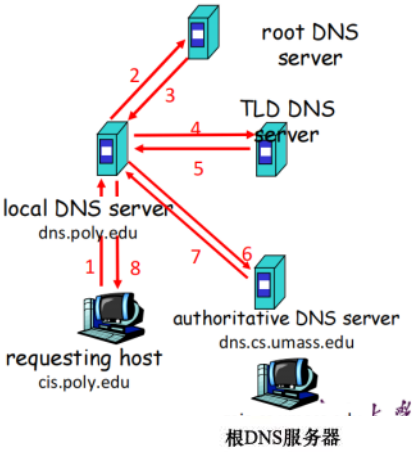
- 每个服务器知道根服务器
- 根服务器知道所有顶级域

e.g.

- Bob at cis.poly.edu wants IP address for Alice at gaia.cs.umass.edu

- Iterated query:
- Contacted server replies with name of next server to contact

- Host-Server: recursive query
- Server-Server: iterative query



• DNS中的递归查询和迭代

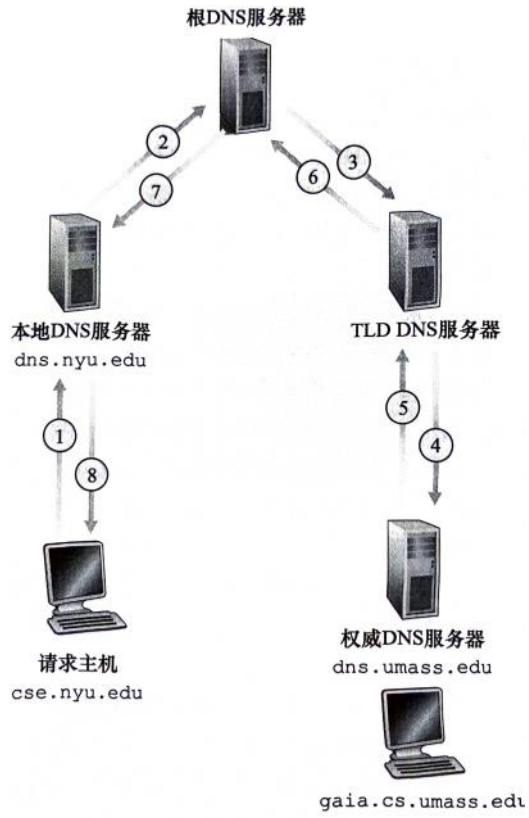


图 2-19　DNS 中的递归查询

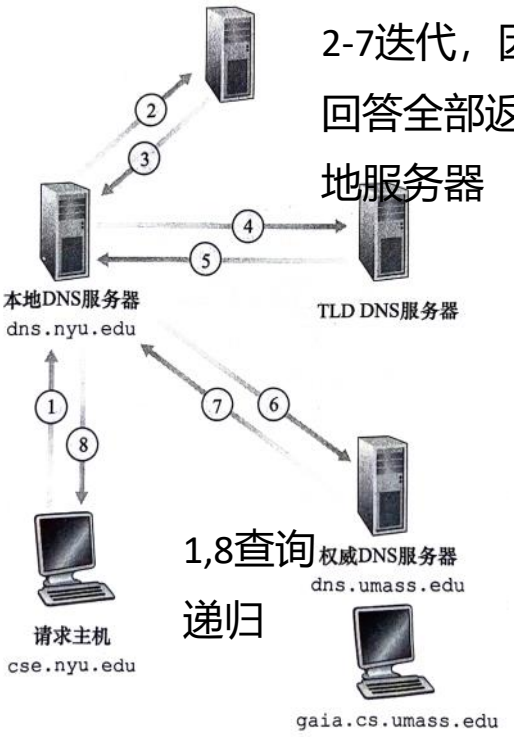


图 2-18　各种 DNS 服务器的交互

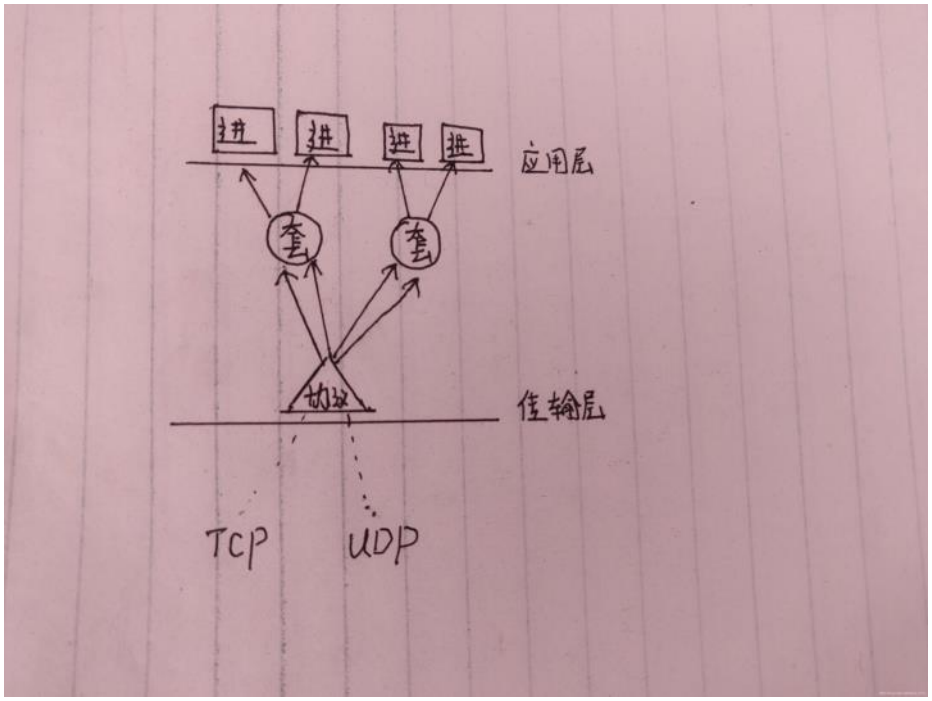
2-7迭代，因为所有回答全部返回给本地服务器

1,8查询递归

运输层（约20分）

套接字是什么?

1 进程 - 1/多个 套接字，用于定位进程



• 运输层的多路复用与多路分解?

多路分解：将运输层报文段中的数据交付到正确的套接字的工作

多路复用：在源主机从不同套接字中收集数据块，并为每个数据块封装上首部信息(这将在以后用于分解)，从而生成报文段，然后将报文段传递到网络层过程的工作。

• UDP套接字? TCP套接字?

UDP套接字：

二元组（目的IP地址 | 目的端口号）

两个UDP报文段有相同的目的IP地址和目的端口号，则会被定位到相同的进程。

TCP套接字：

四元组（源IP地址 | 源端口号 | 目的IP地址 | 目的端口号）

所以要4个一样才会定位到同一个进程。

• 为什么有些应用更适合用UDP?

这些应用不希望过分延时报文段传送，并且能容忍一些数据丢失。

• UDP中的检验和计算?

按每16个比特分组求和并回卷，最后求反码。

• TCP的肯定确认、否定确认、自动重传请求协议?

ACK(OK)

NAK (重传)

自动重传请求协议：ARQ

使用ACK & NAK，接收者可以让发送者知道是都要重传，基于这样的重传机制的可靠数据传输协议

三种协议功能

差错检测、接收方反馈、重传

• 停等协议的基本原理?

rdt2.0 - 相对的叫流水线，停等就是停着等一个分组ack

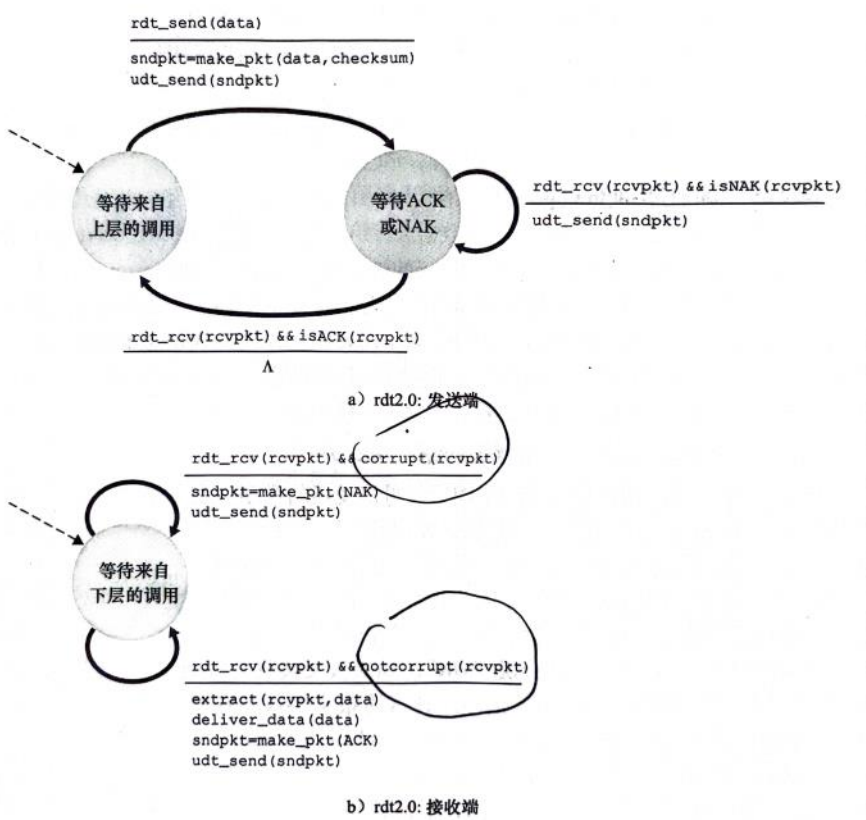


图 3-10 rdt2.0：用于具有比特差错信道的协议

发送方:

等待ACK/NAK，

如果NAK重发，如果ACK，等待来自上层的数据。

接收方：

差错检验决定发ACK/NAK

but 如何处理ACK/NAK受损?

直接问：你说什么？ (寄,直接无限循环)

搞一个能恢复差错的差错算法（只要信道别丢包，不然也寄）

发送方提供不清就重传（产生冗余分组，接收方不知道这个分组是新的还是因NAK重传的还是因听不清重传的冗余分组⇒加入序号机制）

- **比特交替协议的基本原理?**

rdt3.0 一个在可能出错和丢包的信道上可靠传输数据的协议

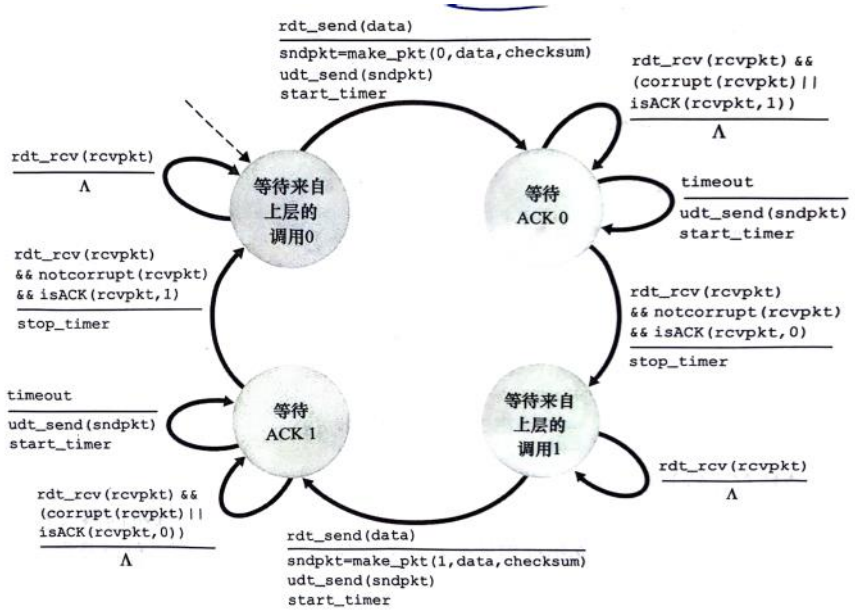


图 3-15　rdt3. 0 发送方

主要加入了两个机制

a. 分组序号机制，分组在0、1间交替

b. 定时器过期重传机制

- **回退N步(GBN)的基本原理?** 单个出错直接GO-BACK-N!

发送方滑动窗口累计确认；

接收方丢弃所有失序分组。

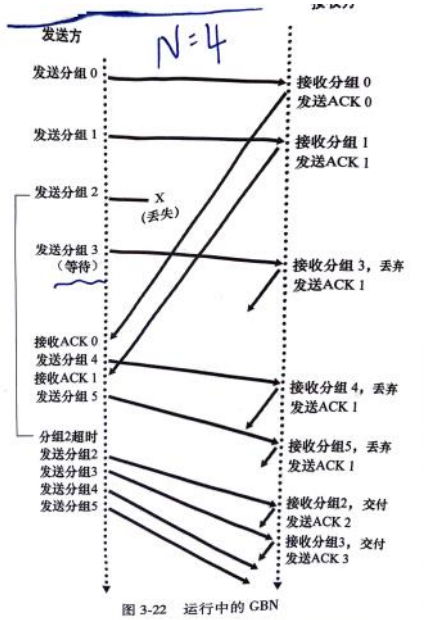


图 3-22　运行中的 GBN

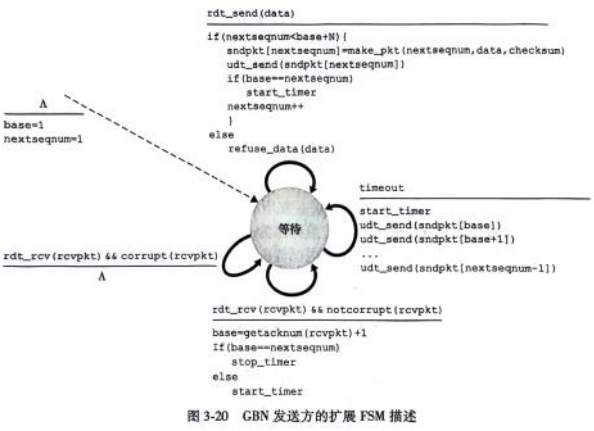


图 3-20　GBN 发送方的扩展 FSM 描述

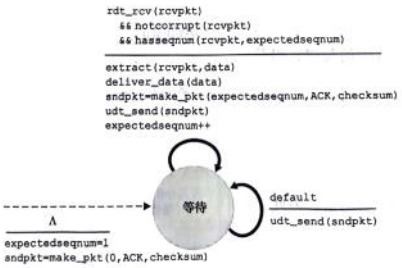
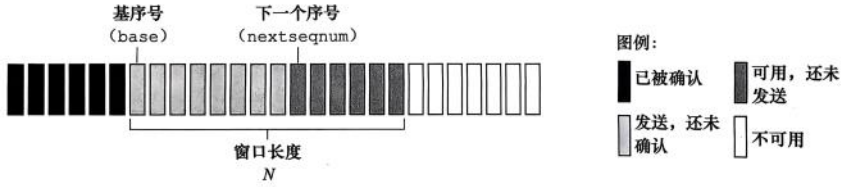


图 3-21　GBN 接收方的扩展 FSM 描述

图 3-22 运行中的 GBN

图 3-21 GBN 接收方的扩展 FSM 描述

图 3-19 显示了发送方看到的 GBN 协议的序号范围。如果我们将基序号（base）定义为最早未确认分组的序号，将下一个序号（nextseqnum）定义为最小的未使用序号（即下一个待发分组的序号），则可将序号范围分割成 4 段。在［0，base−1］段内的序号对应于已经发送并被确认的分组。［base，nextseqnum−1］段内对应已经发送但未被确认的分组。［nextseqnum，base+N−1］段内的序号能用于那些要被立即发送的分组，如果有数据来自上层的话。最后，大于或等于 base+N 的序号是不能使用的，直到当前流水线中未被确认的分组（特别是序号为 base 的分组）已得到确认为止。



注意发送方的定时器和超时的处理：

如果超时，重发所有[base,nseq)分组

收到ACK，若 base = nseq，停止计时器，否则重启。

注意发送方的确认方式:

累计确认，即如果ACK-i，代表i之前的所有分组都ACK

注意接收方的丢弃方式：

丢弃所有失序分组，即如果上次交付的是i，则如果序号不是i+1就丢弃，并发送ACK-i

• 选择重传(SR)的基本原理?

发送方分组单个确认；

接受方缓存失序分组。

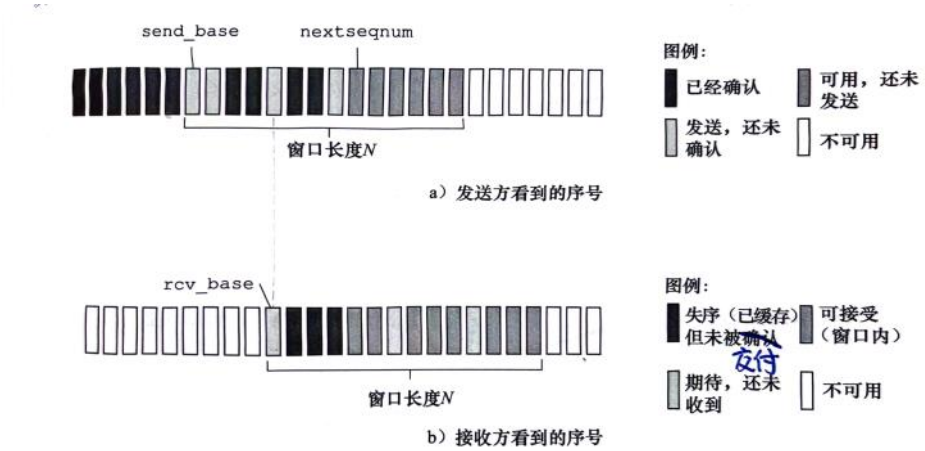


图 3-23 选择重传（SR）发送方与接收方的序号空间

1. 从上层收到数据。当从上层接收到数据后，SR 发送方检查下一个可用于该分组的序号。如果序号位于发送方的窗口内，则将数据打包并发送；否则就像在 GBN 中一样，要么将数据缓存，要么将其返回给上层以便以后传输。

2. 超时。定时器再次被用来防止丢失分组。然而，现在每个分组必须拥有其自己的逻辑定时器，因为超时发生后只能发送一个分组。可以使用单个硬件定时器模拟多个逻辑定时器的操作［Varghese 1997］。

3. 收到 ACK。如果收到 ACK，倘若该分组序号在窗口内，则 SR 发送方将那个被确认的分组标记为已接收。如果该分组的序号等于 send_base，则窗口基序号向前移动到具有最小序号的未确认分组处。如果窗口移动了并且有序号落在窗口内的未发送分组，则发送这些分组。

图 3-24　SR 发送方的事件与操作

1. 序号在［rcv_base，rcv_base+N−1］内的分组被正确接收。在此情况下，收到的分组落在接收方的窗口内，一个选择 ACK 被回送给发送方。如果该分组以前没收到过，则缓存该分组。如果该分组的序号等于接收窗口的基序号（图 3-23 中的 rcv_base），则该分组以及以前缓存的序号连续的（起始于 rcv_base 的）分组交付给上层。然后，接收窗口按向前移动分组的编号向上交付这些分组。举例子来说，考虑一下图 3-26。当收到一个序号为 rcv_base=2 的分组时，该分组及分组 3、4、5 可被交付给上层。

2. 序号在［rcv_base−N，rcv_base−1］内的分组被正确收到。在此情况下，必须产生一个 ACK，即使该分组是接收方以前已确认过的分组。重新确认已收到过小于窗口min号0的分组

3. 其他情况。忽略该分组。

图 3-25　SR 接收方的事件与操作

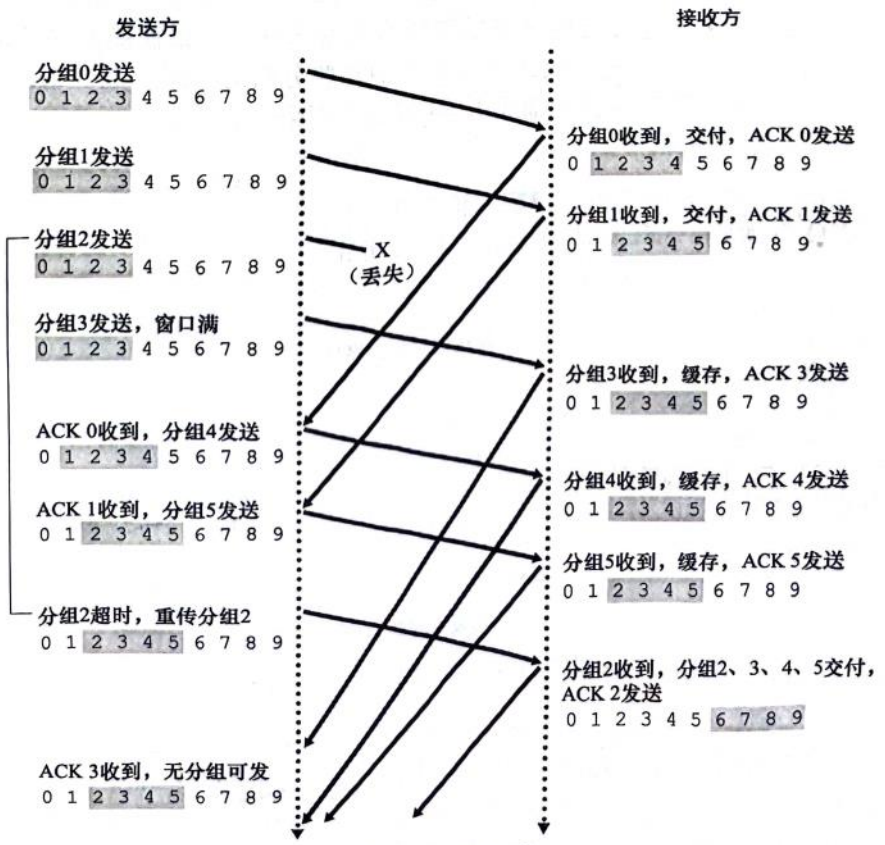


图 3-26　SR 操作

注意 窗口长度小于序列空间时，可能无法工作，是新分组还是重传?

- **什么是流量控制?**

因接收方缓存有限而抑制发送方发送

消除发送方使接收方缓存溢出的可能性。

发送方维护rwnd接收窗口.

接收方在建立连接时，为此次连接分配一个RcvBuffer的缓存大小，

接收方发给发送方时，告诉发送方还剩多少rwnd = RcvBuffer - HasStored

Trick：when rwnd = 0，发送方发1字节的数据报文。

- **什么是拥塞控制？-- 计算题1**

因为网络拥塞而抑制发送方发送

- **TCP的3次握手?**

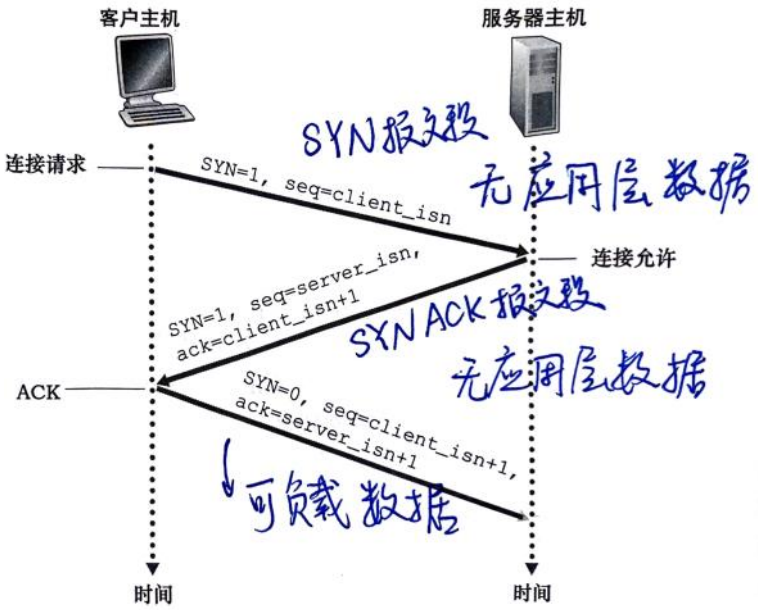


图 3-39　TCP 三次握手：报文段交换

- **TCP拥塞控制方法：慢启动、拥塞避免、快速恢复?**

cwnd (拥塞窗口)

丢包事件的定义：超时 / 3次冗余ACK

简单的概括 -- Reno 版本(考试的版本--应该)：

超时归一，进入慢启动，

冗余折半，进入快速恢复，

慢启动超阈值(cwnd ≥ ssthresh)进入拥塞避免，

快速恢复成功(new ack)进入拥塞避免。

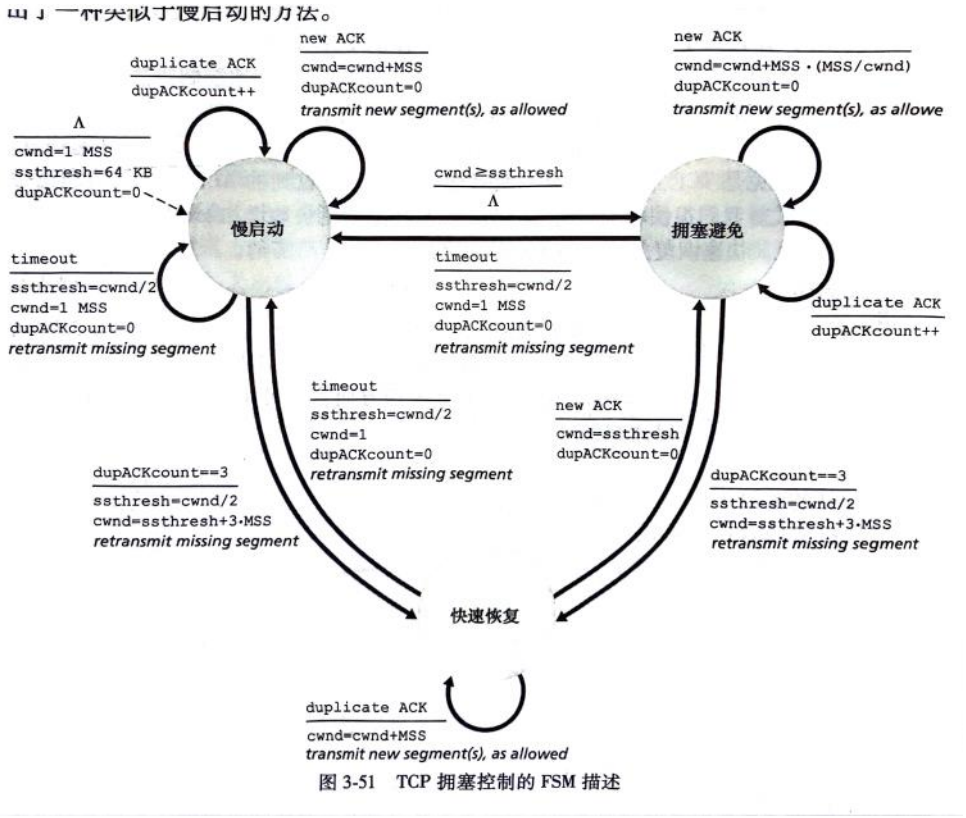


图 3-51 TCP 拥塞控制的 FSM 描述

Tahoe版本将冗余ACK的处理和超时处理一样。

补充：

TCP的要点：书 P151 - P152

首部20字节

序号 & 确认号(以字节为单位，确认号是想要的下一字节的序号)

TCP 提供累计确认 (也就是就算缓存失序报文，但是不会立刻发失序报文的下一个字节的ACK)

TCP 没有规定如何处理失序报文段

$LastByteSent - LastByteAcked \leq min(cwnd, rwnd)$

UDP：首部8字节

网络层：数据平面（约15分）

• 数据平面主要作用——转发是什么?

将分组从一个输入链路接口转移到适当的输出链路接口的路由器本地操作。

转发在很短的时间尺度(ns)发生，因此通常是硬件实现。

• 控制平面的主要作用——路由选择?

确定分组从源目的地所采用的端到端路径的网络范围处理过程，(s)的时间尺度。

• 路由器的4个组件?

输入端口

交换结构

输出端口

路由选择处理器

• 路由器中的最长前缀匹配规则?

寻找符合匹配的、前缀最长的那个

• 三种交换技术?

经内存交换：类似计算机

copy in 内存后再取出

吞吐量 < 1/2 * B (B为每秒读/写内存的最多分组数)

经总线交换：使用一个总线

总线上只能有一个分组。

经互联网络交换：纵横式交换机

2N个总线，N个输入+N个输出端口

只要输出端口没人用，我就可以去，并且不堵车。

• 分组调度：先进先出、优先权排队、循环和加权公平排队?

FIFO 略

优先权：分成两个队列，一个队列优先度更高

循环和加权公平：

若干个队列，每个加个权，循环服务，

服务的时间占比 $w_i / \sum w_j$ 分母是队列不为空的权之和

理想化 $r_i = R * w_i / \sum w_j$

- **IPv4编址：二进制/十进制IP地址、子网、无类别域间路由选择?**

2/10IP地址 略，10进制名字叫点分十进制记法

子网 形如 a.b.c.d/x，/x记法是子网掩码

无类别域间路由选择：

因特网的地址分配策略

使用子网寻址，IP被划分为两个部分，a.b.c.d/x, x是前缀

- **动态主机配置协议(DHCP)的工作原理?**

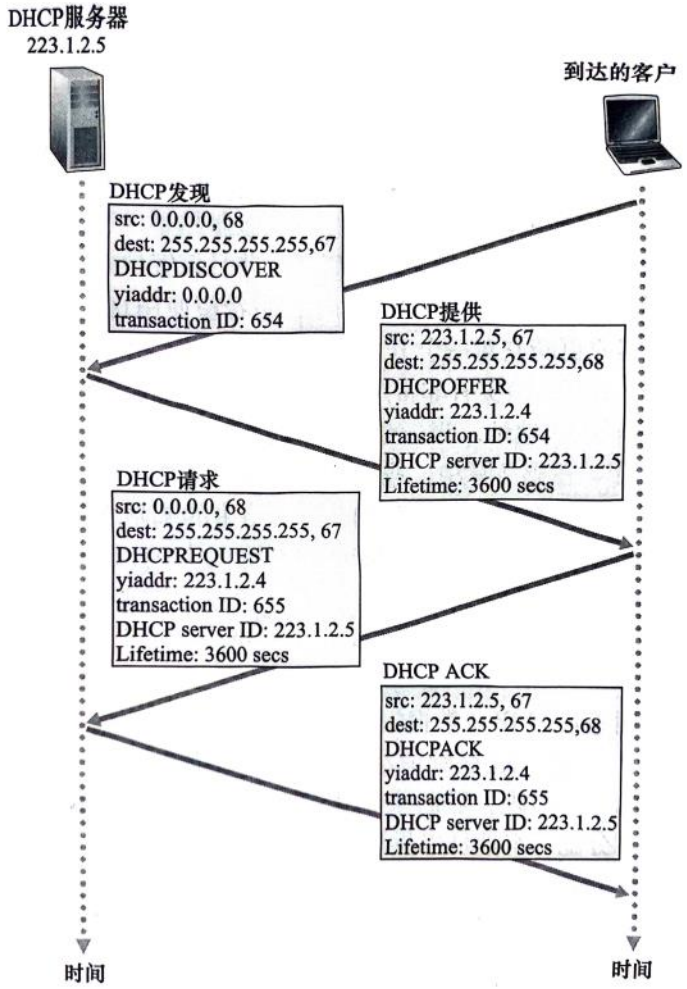


图 4-24　DHCP 客户-服务器交互

注意点：

全部要广播。

子网没有DHCP服务器时，需要DHCP中继代理(通常是一台路由器)

提供报文一般包含收到的发现报文的事务ID，推荐IP地址，网络掩码以及IP地址租用期。

- **网络地址转换的工作原理?**

行为就如同一个具有单一 IP 地址的单一设备。在图 4-25 中，所有离开家庭路由器流向更大因特网的报文都拥有一个源 IP 地址 138. 76. 29. 7，且所有进入家庭的报文都拥有同一个目的 IP 地址 138. 76. 29. 7。从本质上讲，NAT 使能路由器对外界隐藏了家庭网络的细节。（另外，你也许想知道家庭网络计算机是从哪儿得到其地址，路由器又是从哪儿得到它的单一 IP 地址的。在通常的情况下，答案是相同的，即 DHCP！路由器从 ISP 的 DHCP 服务器得到它的地址，并且路由器运行一个 DHCP 服务器，为位于 NAT-DHCP 路由器控制的家庭网络地址空间中的计算机提供地址。）

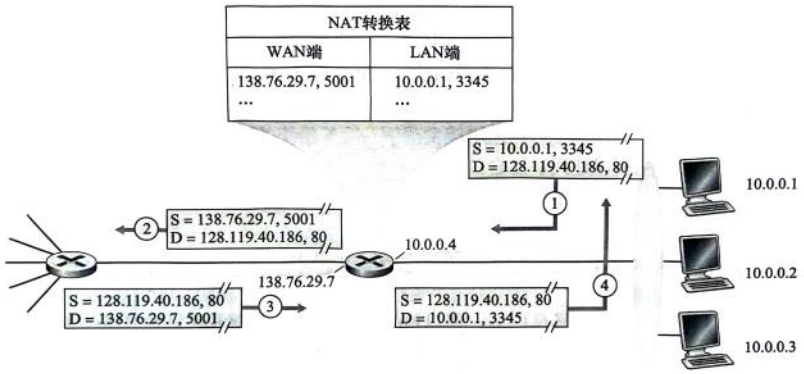


图 4-25　网络地址转换

如果从广域网到达 NAT 路由器的所有数据报都有相同的目的 IP 地址（特别是对 NAT 路由器广域网一侧的接口），那么该路由器怎样知道它应将某个分组转发给哪个内部主机呢？技巧就是使用 NAT 路由器上的一张 **NAT 转换表**（NAT translation table），并且在表项中包含了端口号及其 IP 地址。

- **IPv4向IPv6迁移中采用的建隧道方法?**

简单来说就是，在隧道始末，

把IPv6整个数据报塞到 IPv4 的有效载荷里面，然后再去除。

个涉及数十亿台机器的怀志日现在更是不可想象的。

在实践中已经得到广泛采用的 IPv4 到 IPv6 迁移的方法包括建隧道（tunneling）[RFC 4213]。除了 IPv4 到 IPv6 迁移之外的许多其他场合的应用都具有建隧道的关键概念，包括在第 7 章将涉及的全 IP 蜂窝网络中也得到广泛使用。建隧道依据的基本思想如下：假定两个 IPv6 节点（如图 4-27 中的 B 和 E）要使用 IPv6 数据报进行交互，但它们是经由中间 IPv4 路由器互联的。我们将两台 IPv6 路由器之间的中间 IPv4 路由器的集合称为一个**隧道**（tunnel），如图 4-27 所示。借助于隧道，在隧道发送端的 IPv6 节点（如 B）可将整个 IPv6 数据报放到一个 IPv4 数据报的数据（有效载荷）字段中。于是，该 IPv4 数据报的地址设为指向隧道接收端的 IPv6 节点（在此例中为 E），再发送给隧道中的第一个节点（在此例中为 C）。隧道中的中间 IPv4 路由器在它们之间为该数据报提供路由，就像对待其他数据报一样，完全不知道该 IPv4 数据报自身就含有一个完整的 IPv6 数据报。隧道接收端的 IPv6 节点最终收到该 IPv4 数据报（它是该 IPv4 数据报的目的地），并确定该 IPv4 数据报含有一个 IPv6 数据报（通过观察在 IPv4 数据报中的协议号字段是 41［RFC 4213］，指示该 IPv4 有效载荷是 IPv6 数据报），从中取出 IPv6 数据报，然后再为该 IPv6 数据报提供路由，就好像它是从一个直接相连的 IPv6 邻居那里接收到该 IPv6 数据报一样。

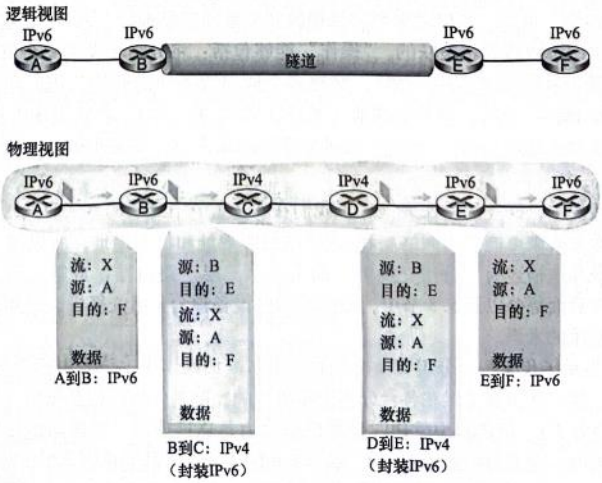


图 4-27 建隧道

网络层：控制平面（约15分）

• 每路由器控制、逻辑集中式控制?

每路由器控制：

每台路由器运行一种路由选择算法，每台路由器都包含转发和路由选择功能，并有一个路由选择组件，用于与其他路由器通讯。

主要有OSPF(开放最短路)和BGP(边界网关)协议。

逻辑集中式控制：

逻辑集中式控制器计算并分发转发表以供每个路由器使用。

控制器以一种定义良好的协议与每台路由器中的一个控制代理CA进行交互，以配置和管理该路由器的转发表。

• 链路状态算法(LS)的工作原理?

原理：Dijkstra算法；性质：集中式路由选择算法。

输入：网络拓扑+链路开销、源节点

输出：源节点的最短路径

实践中，通过每个节点向网络中的所有其他节点广播链路状态分组来完成输入，分组包含了它所连接的链路的标识和开销；实践中(如使用OSPF协议)，经常由**链路状态广播**算法来完成。

问题：当链路成本依赖于交通量时，可能会出现路由振荡

• 距离向量算法的工作原理?

原理：Ballman-Ford；性质：分散式...。

见： DV_algorithm.py · MojoisMe/Nju-Cn-Lab - 码云 - 开源中国 (gitee.com)

特性：好消息传播快，坏消息传播慢。

无限计数问题

原因：路径选择环路

zx的最短路：z->y->x,

yx新的最短路：y->z->y-(旧)>x

不完全的解决方法：毒性逆转

z->y->x, then z tells y: Dz(x) = infinity

• 什么是OSPF?

自治系统(AS)内部的路由选择算法，一种链路状态协议。

使用泛洪链路状态信息和Dijkstra最低开销路径算法，

各链路开销是管理员配置的。

使用OSPF时，路由器向AS内所有其他路由器广播路由选择信息。

当一条链路状态发生改变时，或者一个周期结束(至少每隔30min)，路由器广播链路状态信息。

优点：

安全、支持选择多条相同开销的路径、综合支持单播与多播路由选择、支持在单个AS中的层次结构。

• 边界网关协议BGP的工作原理?

ISP之间的路由选择算法，自治系统间的路由选择协议。

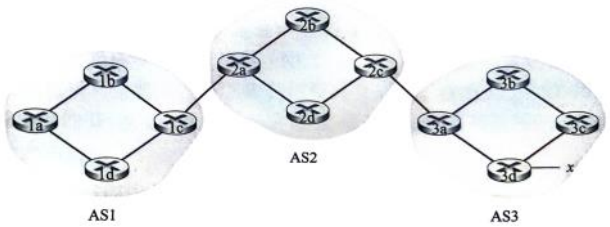


图 5-8　具有 3 个自治系统的网络。AS3 包括一个具有前缀 x 的子网

我们考虑这样一个任务：向图 5-8 中显示的所有路由器通告对于前缀 x 的可达性信息。在高层次上，这是简明易懂的。首先，AS3 向 AS2 发送一个 BGP 报文，告知 x 存在并且位于 AS3 中；我们将该报文表示为“AS3 x”。然后 AS2 向 AS1 发送一个 BGP 报文，告知 x 存在并且能够先通过 AS2 然后进入 AS3 进而到达 x；我们将该报文表示为“AS2 AS3 x”。以这种方式，每个自治系统不仅知道 x 的存在，而且知道通向 x 的自治系统的路径。

一点，我们现在重温图 5-8 中的例子。在 BGP 中，每对路由器通过使用 179 端口的半永久 TCP 连接交换路由选择信息。每条直接连接以及所有通过该连接发送的 BGP 报文，称为 BGP 连接（BGP connection）。此外，跨越两个 AS 的 BGP 连接称为外部 BGP（eBGP）连接，而在相同 AS 中的两台路由器之间的 BGP 会话称为内部 BGP（iBGP）连接。图 5-8 所示网络的 BGP 连接的例子显示在图 5-9 中。对于直接连接在不同 AS 中的网关路由器的每条链路而言，通常有一条 eBGP 连接；因此，在图 5-9 中，在网关路由器 1c 和 2a 之间有一条 eBGP 连接，而在网关路由器 2c 和 3a 之间也有一条 eBGP 连接。

两个属性：

AS-PATH: 到达x的路径

NEXT-HOP: 该路径下一跳的IP地址(路由器某接口的IP)

热土豆路由选择：网关间的路由选择——自私的算法

分组被喻为烫手山芋，每个网关都希望早点把他丢到其他AS里面。

所以，选择网关间开销最短的，而不考录丢给其他网关后的开销。

实践中的选择：逐步选择,不再自私

定义了偏好值

a. 选择偏好值最高的
b. 选择AS跳数最短的
c. 热土豆选择
d. 使用BGP标识符选择【?】

• SDN体系结构的4个关键特征? P270

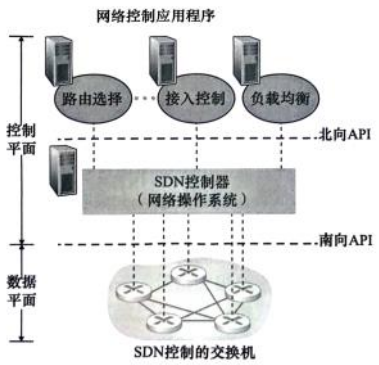


图 5-14　SDN 体系结构的组件：SDN 控制的交换机、SDN 控制器和网络控制应用程序

基于流的转发

数据平面与控制平面分离

网络控制功能：位于数据平面交换机外部

可编程的网络

- **什么是ICMP?**

因特网控制报文协议

用途：主机和路由器用其彼此沟通网络层的信息，典型用途为差错报告

- **网络管理的关键组件？ P280**

管理服务器：应用程序

被管设备

数据：配置数据、运行数据

设备统计

网络管理代理：运行在被管设备上，(类似路由选择代理)

网络管理协议

三种方式管理网络：

CLI 命令行接口

SNMP/MIB

通过**SNMP**查询/设置设备的**管理信息库(MIB)**对象中包含的数据。

NETCONF/YANG

- **什么是SNMP?**

Simple Network Management Protocol **简单网络管理协议**

是一个应用层协议，用于管理服务器和代表管理服务器执行的代理之间传递网络管理控制和信息报文。

最常使用的是请求响应模式：

SNMP管理服务器向代理发送请求，代理接收后执行某些操作，然后发送回复。

陷阱报文：

非请求报文，代理通知服务器：因异常情况导致MIB对象值已经改变。

链路层和局域网（约15分）

• 什么是节点、链路?

节点:

运行链路层协议的任何设备(主机、路由器、交换机、WiFi接入点等)

链路:

沿着通信路径连接相邻节点的通信信道

• 链路层提供的可能服务包括?

成帧:

链路接入:

MAC(介质访问控制)协议规定

可靠交付:

通常用于易于产生高差错率的链路(如无线链路)

差错检测和纠正

• 链路层在何处实现?

网络适配器的芯片

• 奇偶校验、检验和、循环冗余检测的基本原理?

奇偶校验-二维奇偶校验：求一下异或得校验比特，略

检验和：TCP/UDP 每16位bit相加回卷，略

循环冗余(CRC)：链路层常用！以太网帧使用！作业题有，略

$$R = \text{remainder}\frac{D \cdot 2^r}{G}$$

图 6-7 举例说明了在 $D=101110$，$d=6$，$G=1001$ 和 $r=3$ 的情况下的计算过程。在这种情况下传输的 9 个比特是 101110011。你应该自行检查一下这些计算，并核对一下 $D \cdot 2^r = 101011 \cdot G$ XOR R 的确成立。

国际标准已经定义了 8、12、16 和 32 比特生成多项式 G。CRC-32 32 比特的标准被多种链路级 IEEE 协议采用，使用的一个生成多项式是：

$G_{CRC-32} = 100000100110000010001110110110111$

每个 CRC 标准都能检测小于 $r+1$ 比特的突发差错。（这意味着所有连续的 r 比特或者更少的差错都可以检测到。）此外，在适当的假设下，长度大于 $r+1$ 比特的突发差错以概率 $1-0.5^r$ 被检测到。每个 CRC 标准也都能检测任何奇数个比特差错。有关 CRC 检测实现的讨论可参见［Williams 1993］。CRC 编码甚至更强的编码所依据的理论超出了本书的范围。教科书［Schwartz 1980］对这个主题提供了很好的介绍。

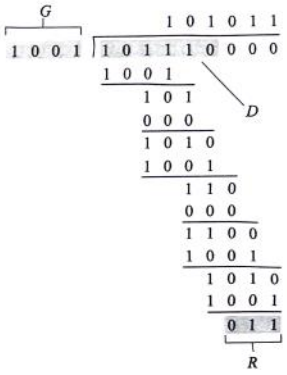


图 6-7 一个简单的 CRC 计算

- **多路访问协议分为哪三种类型？每种类型的工作原理？**

多路访问问题：

如何协调多个发送和接收节点对于一个共享信道的访问。

信道划分协议：

TDM 时分 为每个安排一个slot(时隙)

FDM 频分 为每个安排一个频段

CDMA 码分多址 (下一章节) 为每个安排一种不同的编码

随机接入协议：

节点以全速发送，有碰撞时，等待一个随机时延，然后继续发，直到OK

时隙ALOHA/ALOHA (不是很重要?)

CSMA (!!)

轮流协议：

轮询协议/轮流协议：

选择一个主节点，循环的方式轮询每个节点(向其发送报文告诉能传输的帧的最多数量)，在该节点传输某些帧后，再告诉下一个节点。

主节点通过观察信道上是否缺乏信号来决定一个节点何时完成了发送。

令牌传递：

只有有令牌才能发，节点完成最大数目帧的发送后，以固定顺序交给下一个节点。

- **什么是CSMA?**

CS：载波侦听

说话前先听，发送前，先要确认一段时间信道空闲。

MA：多路访问

因为有信道传播时延(d_{prop})，所以仍会碰撞。

- **什么是CSMA/CD?**

CD：碰撞检测

边说边听，发送时如果监测到其他节点也在传输，自身立即停止传输，并在重复“侦听-当空闲时传输”前等待一段随机时间。

随机时间一般是二进制指数后退：

传输一个帧，已经经历了n(n大于10按10计算)次碰撞，我rand一个K$\in \{0,1,..,2^n-1\}$,然后等待512*K的比特时间(即submit512Kbit所需要的时间)

效率 = $1/(1+\frac{5d_{prop}}{d_{trans}})$

- **CSMA & CSMA/CD的区别是什么?**

加了CD，通过不传输一个无用的、因干扰而损坏的帧，改善性能

- **MAC地址的表示方式?**

链路层地址，长度6字节，每个字节16进制表示

e.g.：AA-AA-AA-AA-AA-AA

- **地址解析协议(ARP)的工作原理?**

ARP将IP解析为MAC地址，放入该帧的首部作为目的主机的MAC地址。

注意，ARP只为在同一个子网的主机和路由器接口解析IP地址。

每台主机内有个ARP表Cache(带TTL寿命机制)，能命中则直接发，不能则进行查询ARP报文(类似DNS,但又不太一样)。

a. 广播ARP查询报文
b. 目标收到查询报文后，回复响应ARP报文，注意这里是标准帧而不广播

补充，跨子网的数据传输：

子网1主机1.1向子网2主机2.1发送数据

目标IP地址：主机2.1的IP地址

目标MAC地址：路由器与子网1接口的MAC地址。

路由器的每个接口都有一个IP地址和适配器(MAC地址）。

- **交换机和路由器的区别？P324**

交换机运行在链路层，使用MAC地址进行转发；

即插即用、相对高的分组过滤和转发速率；

活跃拓扑被限制在树内、对广播风暴不提供任何保护。

路由器运行在网络层。

没有生成树限制、允许丰富的拓扑结构，针对广播风暴提供防火墙保护；

不是即插即用、需要给自身和连接到它的主机分配IP地址，处理时间较长，英文发音不统一(逆天)。

- **多协议标签交换技术MPLS？P329**

在链路层首部和网络层首部增加一个MPLS首部，

该首部由 20bit固定长度标签、3bit实验位、1bitS位、5bitTTL构成。

MPLS使能的路由器，又称标签交换路由器，

根据标签而不是IP地址，来确定转发接口（faster！）

维护一个入标签、出标签、目的地IP、出接口的表，

- **数据中心网络的等级拓扑？P333**

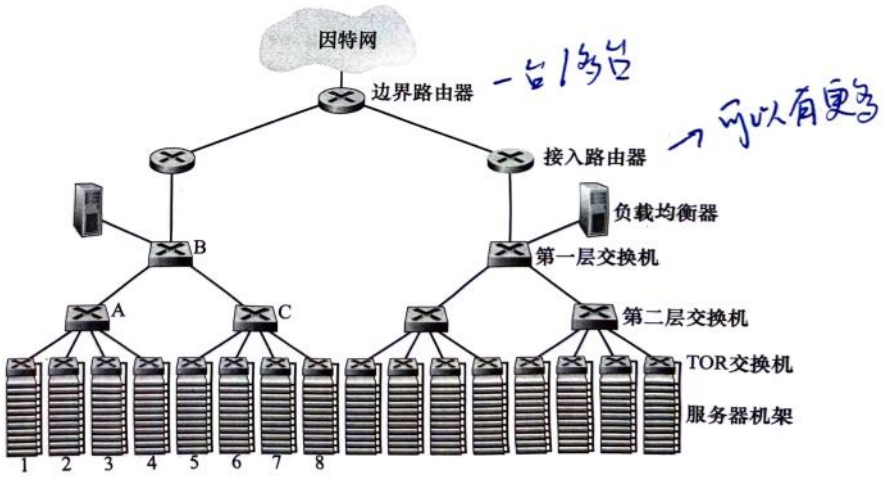


图 6-30　具有等级拓扑的数据中心网络

- 作业-关于集线器和交换机

到与该总线连接的所有适配器并被其处理。回忆一下，我们在 6.3.2 节中讨论了以太网的具有二进制指数回退的 CSMA/CD 多路访问协议。

到了 20 世纪 90 年代后期，大多数公司和大学使用一种基于集线器的星形拓扑以太网安装替代了它们的局域网。在这种安装中，主机（和路由器）直接用双绞对铜线与一台集线器相连。**集线器**（hub）是一种物理层设备，它作用于各个比特而不是作用于帧。当表示一个 0 或一个 1 的比特到达一个接口时，集线器只是重新生成这个比特，将其能量强度放大，并将该比特向其他所有接口传输出去。因此，采用基于集线器的星形拓扑的以太网也是一个广播局域网，即无论何时集线器从它的一个接口接收到一个比特，它向其所有其他接口发送该比特的副本。特别是，如果某集线器同时从两个不同的接口接收到帧，将出现一次碰撞，生成该帧的节点必须重新传输该帧。

在 21 世纪初，以太网又经历了一次重要的革命性变化。以太网安装继续使用星形拓扑，但是位于中心的集线器被**交换机**（switch）所替代。在本章后面我们将深入学习交换以太网。眼下我们仅知道交换机不仅是“无碰撞的”，而且也是名副其实的存储转发分组交换机就可以了；但是与运行在高至第三层的路由器不同，交换机仅运行在第二层。

无线网络和移动网络（约10分）

• 什么是无线主机、无线链路、基站?

无线主机:

智能手机、平板电脑、物联网设备等(可以不移动)

无线链路:

主机通过无线通信链路连接到一个基站/无线主机

基站:

负责向与之关联的无线主机发送数据并从主机中获取数据。

• 什么是基础设施模式、自组织网络?

与基站关联的主机通常被称为以**基础设施模式**运行

在**自组织网络**中，无线主机没有这样的基础设置与之相连，主机本身必须提供诸如路由选择、地址分配以及类似于DNS的名字转换等服务。

• 什么是多径传播、信噪比？隐藏终端问题?

多径传播:

当电磁波的一部分受物体和地面反射，在发送方和接收方之间走了不同长度的路径时，就出现了多径传播。

信噪比SNR:

所收到的信号(传输信息)和噪声强度的相对测量 | 单位分贝

接收到的信号振幅与噪声幅度比值以10为底的对数的20倍

隐藏终端问题:

A 与 C 之间存在物理等阻挡，妨碍AC互相听见，

就可能出现，A向B、C向B同时传输并产生干扰的事件。

• CDMA码分多址的基本工作原理?

每个发送方有一个编码长度为M，每个值为± 1，称为c_i

称一个比特时间为1比特时隙时间，分成M个微时隙。

$$I(b) = \begin{cases} 1(b=1) \\ -1(b=0) \end{cases}$$

编码时，输出的就是Z序列：

$$\mathrm{Z_{i,m}} = \mathrm{I}(b_i) * c_m$$

解码时，使用公式：

$$b = I^{-1}(\frac{1}{M}\sum_{m=1}^{M} Z_{i,m} * c_m)$$

假设干扰的输出比特信号是加性的。

这样即使有干扰，仍然可以得到正确的b。

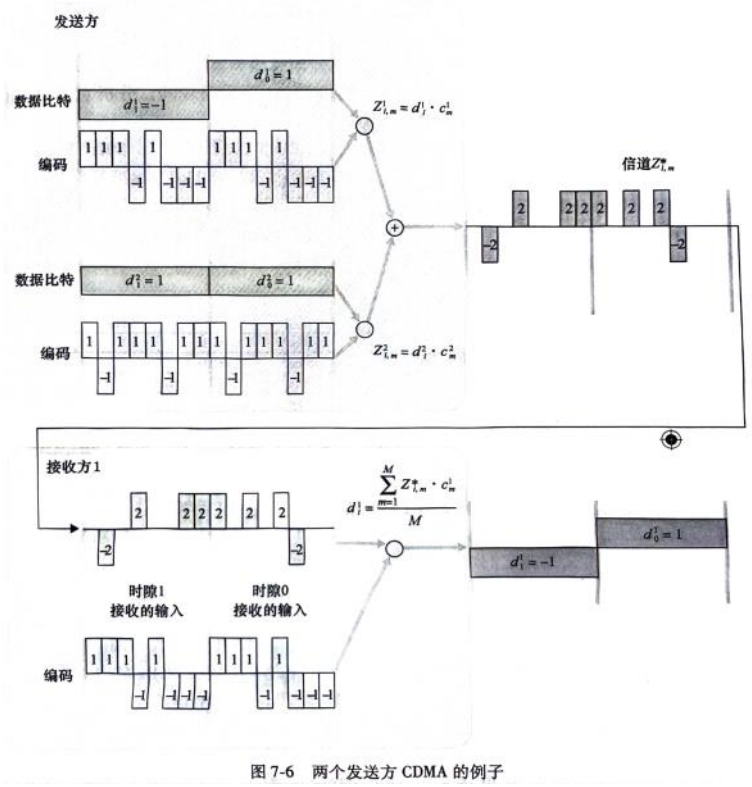


图 7-6　两个发送方 CDMA 的例子

- **被动扫描、主动扫描?**

被动扫描：

扫描信道和监听信标帧的过程

主动扫描：

无线主机发起。

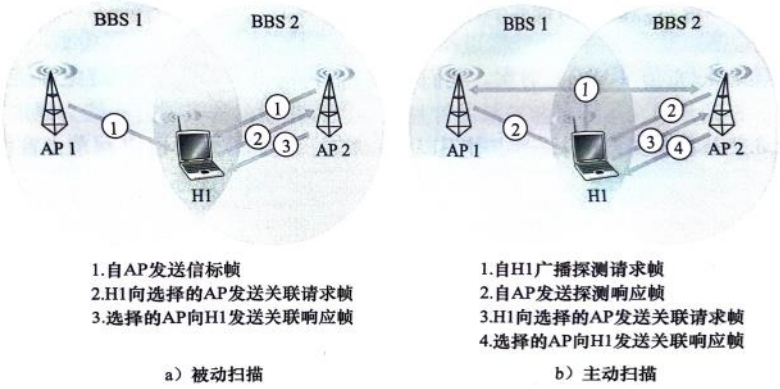


图 7-9　对接入点的主动和被动扫描

• CSMA/CA的工作原理?

CSMA：载波侦听 多路访问

CA collision avoidance：碰撞避免

前提：链路层的确认/重传(ARQ)

　　目标站点收到并检测OK后，等待一个**短帧间间隔**

过程：

1. 监听信道空闲，在**分布式帧间间隔**的短时间后发送
2. 截断二进制指数回退算法，在侦听到空闲时，计数值-1
3. 计数值 = 0时，发送并等待确认
4. OK重传次数=0，如果还要传回到2；否则重传次数+1，回到2。

处理隐藏终端：RTS/CTS交换序列

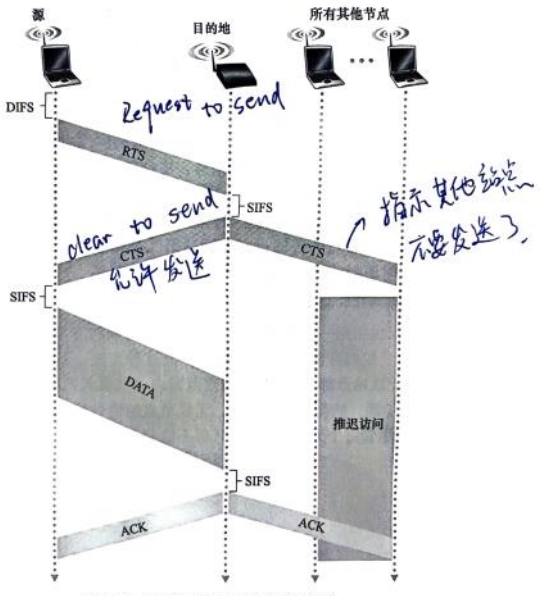


图 7-12　使用 RTS 和 CTS 帧的碰撞避免

RTS 和 CTS 帧的使用能够以两种重要方式来提高性能：

- 隐藏终端问题被缓解了，因为长 DATA 帧只有在信道预约后才被传输。
- 因为 RTS 和 CTS 帧较短，涉及 RTS 和 CTS 帧的碰撞将仅持续短 RTS 和 CTS 帧的持

• 4G LTE架构的部件?

表 7-2 LTE 部件和类似的 WLAN（WiFi）功能

LTE	描述	类似的 WLAN 功能
移动设备	端用户的 IP 使能的无线/移动设备（如智能手机、平板电脑、便携机）	主机，端系统
基站	无线接入 LTE 网络的网络侧	接入点，尽管 LTE 基站执行的许多功能未能在 WLAN 中找到
移动性管理实体	用于移动设备服务的协调器：鉴别、移动性管理	接入点，尽管 MME 执行的许多功能未能在 WLAN 中找到
归属用户服务器	位于移动设备的归属网络中，在归属和访问网络中提供鉴别、访问特权	WLAN 无对应部件
服务网关和 PDN 网关	蜂窝运营商网络中的路由器，协调转发到运营商网络之外	在接入 ISP 网络中的 iBGP 和 eBGP 路由器
无线电接入网	移动设备与基站之间的无线链路	在移动设备与 AP 之间的 802.11 无线链路

- **到移动设备的间接路由？到移动设备的直接路由？ 习题P13/14**

解决通信者向移动设备发消息的问题。

归属网络(注册地) 被访网络(所在地) 两者不同则叫漫游(roaming!)

间接路由：

2. 到移动设备的间接路由

让我们再次考虑想要向移动设备发送数据报的通信者。在**间接路由**方法中，通信者简单地将数据报定位到移动设备的永久地址，并将数据报发送到网络中，不需要知道移动设备是在其归属网络中还是在被访网络中；因此，移动性对通信者来说是完全透明的。像往常一样，这些数据报首先被路由到移动设备的归属网络。这些情况在图 7-26 的步骤 1 中进行了说明。

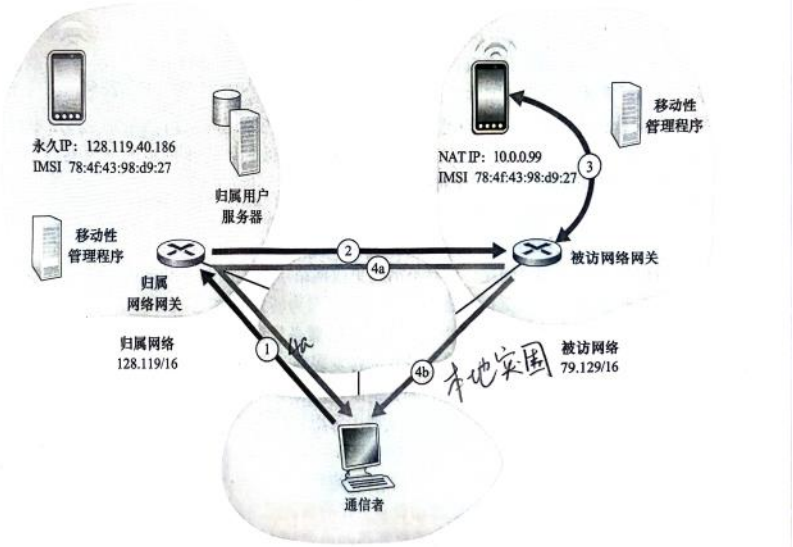


图 7-26 对移动设备的间接路由

现在让我们把注意力转向 HSS。HSS 负责与被访网络进行交互，以跟踪移动设备的位置以及归属网络的网关路由器。这种网关路由器的一项工作是监视到达的数据报，该数据报的地址为本归属网络的某设备，但目前驻留在被访网络中。归属网络网关截获该数据报，与 HSS 协商确定移动设备所在的被访网络，并将该数据报转发给被访网络的网关路由器，即图 7-26 中的步骤 2。然后，被访网络的网关路由器将数据报转发给移动设备，即图 7-26 中的步骤 3。如果使用 NAT 转换，如图 7-26 所示，则由被访网络的网关路由器进行 NAT 转换。

更详细地考虑归属网络的重新路由是有启发意义的。显然，归属网络网关需要将到达的数据报转发到被访网络中的网关路由器。另一方面，最好保持通信者的数据报完整，因为接收数据报的应用程序应该不知道数据报是通过归属网络转发的。这两个目标都可以通过让归属网关将对应的初始完整数据报封装在一个新的（更大的）数据报中来实现。然后，这个较大的数据报被寻址并发送到被访网络的网关路由器，该路由器将对数据报进行解封装。也就是说，从较大的封装数据报中移除对应的初始数据报，并将初始数据报转发

直接路由：

直接路由（direct routing）克服了三角路由选择的低效问题，但却是以增加复杂性为代价的。在直接路由方法中，如图 7-27 所示，通信者首先发现移动设备所在的被访网络。这是通过在移动设备的归属网络中查询 HSS 来完成的，假设（如在间接路由的情况下）移动设备的被访网络在 HSS 中进行了注册，如图 7-27 中的步骤 1 和步骤 2 所示。然后通信者将数据报从其网络直接通过隧道发送到移动设备的被访网络的网关路由器。

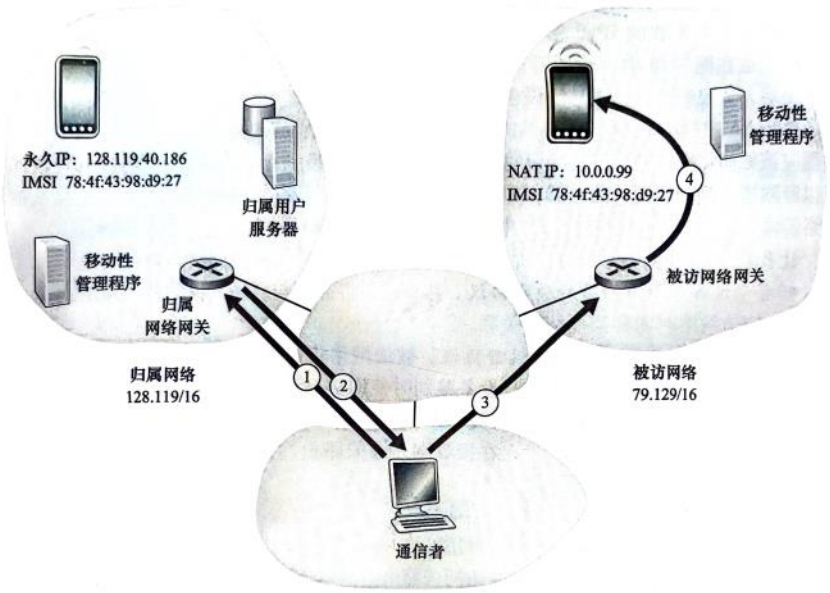


图 7-27　直接路由到某移动设备

尽管直接路由克服了三角路由选择问题，但它引入了两个重要的其他挑战：

- 通信者需要一个移动用户定位协议来查询 HSS，以获得移动设备的被访网络（图 7-27 中的步骤 1 和步骤 2）。这是移动设备向其 HSS 注册位置所需的协议之外的附加协议。
- 当移动节点从一个外部网络移到另一个外部网络时，如何将数据报转发到新的外部网络？在间接路由的情况下，这个问题很容易通过更新归属网络中的 HSS 来解决，并且更改隧道端点使其终止于新被访网络的网关路由器。然而，若使用直接路由，则在被访网络中，这种变化不是那么容易处理，因为 HSS 只在会话开始时由通信者查询。因此，需要额外的协议机制在每次移动设备移动时主动更新相应的协议。本章最后通过两个习题探究了此问题的解答。

- **移动设备从源基站切换到目标基站的步骤?**

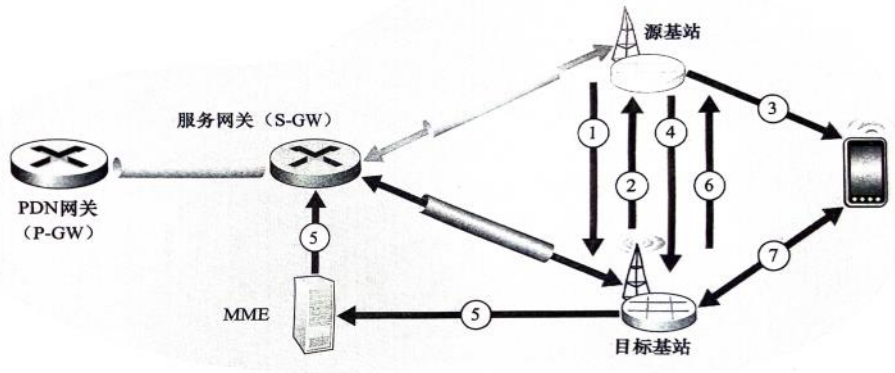


图 7-30　将移动设备从源基站切换到目标基站的步骤

1）当前（源）基站选择目标基站，并向目标基站发送切换请求消息。

2）目标基站检查自己是否有资源来支持移动设备及其业务质量要求。如果是，则在其无线电接入网上为该设备预分配信道资源（例如，时隙）和其他资源。这种资源预分配将移动设备从前面讨论的耗时基站关联协议中解放出来，允许尽可能快地执行切换。目标基站向源基站确认一个切换请求确认报文，该报文包含移动设备需要与新基站关联的目标基站的所有信息。

3）源基站接收到切换请求确认报文，并将目标基站的身份信息和信道接入信息告知移动设备。此时，移动设备可以开始向新的目标基站发送/接收数据报。从移动设备的角度来看，切换已经完成！然而，在网络内部仍有一些工作要做。

4）源基站也将停止向移动设备转发数据报，而是将它接收到的任何隧道化的数据报转发给目标基站，目标基站随后将这些数据报转发给移动设备。

5）目标基站通知 MME 它（目标基站）将是为移动设备服务的新基站。MME 依次向服务网关和目标基站发出信号，以重新配置服务网关到基站隧道，使其在目标基站而不是源基站终止。

6）目标基站向源基站确认隧道已被重新配置，从而允许源基站释放与该移动设备关联的资源。

7）此时，目标基站也可以开始向移动设备发送数据报，包括源基站在切换过程中转发给目标基站的数据报，以及从服务网关重新配置后通过隧道到达的新数据报。它还可以将从移动设备接收到的出方向的数据报转发给到服务网关的隧道。

补充：

1. 移动设备和基站关联
2. 移动设备的网元控制平面配置
3. 移动设备转发隧道的数据平面设置
4. 移动设备从一个基站切换到另一个基站

计算机网络中的安全（约10分）

- **安全通信包括哪四个方面的性质?**

机密性：截取后无法理解

报完完整性：防止恶意篡改或意外改动

端点鉴别：双方能确信对面是声称的那个人

运行安全：防止利用蠕虫等发起的Dos攻击

- **对称密钥系统与公开密钥系统的区别?**

对称密钥：

$K_B\left(K_A(m)\right) = m$

K_A, K_B只有A、B自己知道，并提前约定好了。

A向B发送用K_A加密，B用K_B解密

公开密钥：

$K_A^-\left(K_A^+(m)\right) = m$

K_A^-只有A知道，K_A^+全世界都知道。

向A发，用A的公钥加密，A用自己的私钥解密。

- **RSA算法的工作原理?**

略，作业有，见书P409

- **密码散列函数的性质?**

想要找到**任意**两个报文x,y，

$H(x) = H(y)$ 在计算上是不可能的。

- **数字签名的基本原理?**

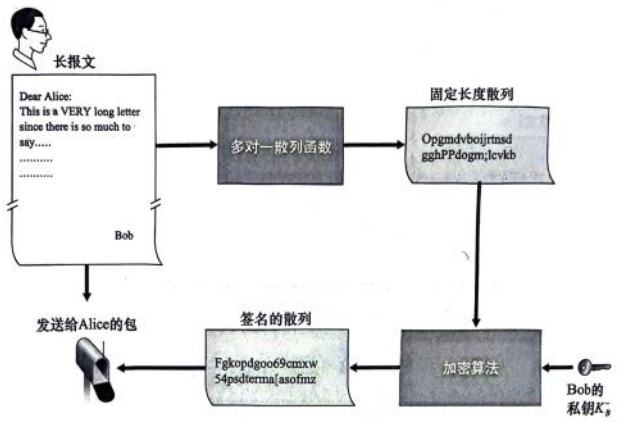


图 8-11　发送数字签名的报文

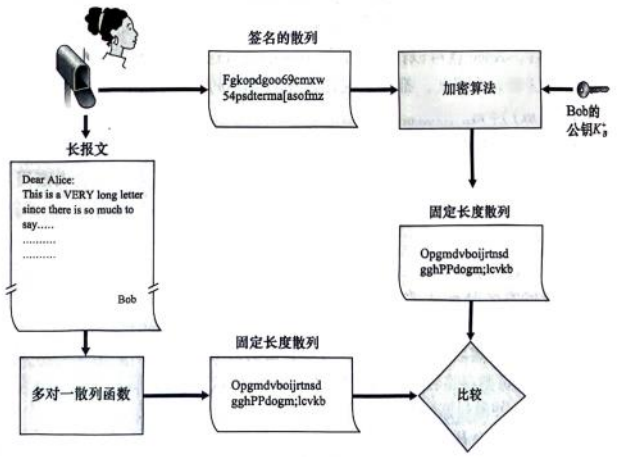


图 8-12　验证签名报文

- **鉴别协议及其安全性?**

1.0 直接 I am Alice

寄

2.0 加上IP地址

寄

3.0 加上口令

有人会窃听，寄

3.1 口令加密

有人会录音，寄

4.0 不重数 + 对称秘密密钥加密

对称系统的问题：如何共享密钥?

（中间人攻击？）

5.0 不重数 + 公开密钥密码体系

使用公钥密码可以绕过共享密钥系统的一个难题

仅仅因为公钥分发而不够安全。

幸运的是，我们能够使用证书来安全地分发公钥

但是仍然无法避免中间人攻击。

- **IPSec协议族中的两个重要协议?**

AH鉴别首部协议

提供源鉴别和数据完整性服务

不提供机密性服务

ESP封装安全性载荷协议

提供源鉴别、数据完整性、机密性服务

- **防火墙包括哪三类？这三类的主要区别?**

传统分组过滤器：

查看数据报

根据一些策略制定规则决定如何处理。

表 8-6　用于某路由器接口的访问控制列表

动作	源地址	目的地址	协议	源端口	目的端口	标志比特
允许	222. 22/16	222. 22/16 的外部	TCP	>1023	80	任意
允许	222. 22/16 的外部	222. 22/16	TCP	80	>1023	ACK
允许	222. 22/16	222. 22/16 的外部	UDP	>1023	53	—
允许	222. 22/16 的外部	222. 22/16	UDP	53	>1023	—
拒绝	全部	全部	全部	全部	全部	全部

状态过滤器：

跟踪TCP连接，在控制列表中加一项是否核对链接。

在经过传统分组过滤后，核对链接判断是否接受。

方法是可能的：因为防火墙能够通过观察三次握手（SYN、SYNACK 和 ACK）来观察一条新连接的开始；而且当它看到该连接的一个 FIN 分组时，它能够观察该连接的结束。当防火墙经过比如说 60 秒还没有看到该连接的任何活动性，它也能够（保守地）假设该连接结束了。某防火墙的一张连接表例子显示在表 8-7 中。这张连接表指示了当前有 3 条进行中的 TCP 连接，所有的连接都是从该机构内部发起的。此外，该状态过滤器在它的访问控制列表中包括了一个新栏，即“核对连接”，如表 8-8 中所示。注意到表 8-8 与表 8-6 中的访问控制列表相同，只是此时它指示应当核对其中两条规则所对应的连接。

表 8-7　用于状态过滤器的连接表

源地址	目的地址	源端口	目的端口
222. 22. 1. 7	37. 96. 87. 123	12699	80
222. 22. 93. 2	199. 1. 205. 23	37654	80
222. 22. 65. 143	203. 77. 240. 43	48712	80

表 8-8　用于状态过滤器的访问控制列表

动作	源地址	目的地址	协议	源端口	目的端口	标志比特	核对连接
允许	222. 22/16	222. 22/16 的外部	TCP	>1023	80	任意	
允许	222. 22/16 的外部	222. 22/16	TCP	80	>1023	ACK	X
允许	222. 22/16	222. 22/16 的外部	UDP	>1023	53	—	
允许	222. 22/16 的外部	222. 22/16	UDP	53	>1023	—	X
拒绝	全部	全部	全部	全部	全部	全部	

应用程序网关：

用程序网关除了看 IP/TCP/UDP 首部外，还基于应用数据来做策略决定。一个**应用程序网关**（application gateway）是一个应用程序特定的服务器，所有应用程序数据（入和出的）都必须通过它。多个应用程序网关可以在同一主机上运行，但是每一个网关都是有自己的进程的单独服务器。

为了更深入地了解应用程序网关，我们来设计一个防火墙，它只允许内部客户的受限集合向外 Telnet，不允许任何外部客户向内 Telnet。这一策略可通过将分组过滤（在一台路由器上）和一个 Telnet 应用程序网关结合起来实现，如图 8-35 所示。路由器的过滤器配置为阻塞所有 Telnet 连接，但从该应用程序网关 IP 地址发起的连接除外。这样的过滤器配置迫使所有向外的 Telnet 连接都通过应用程序网关。现在考虑一个要向外部 Telnet 的内部用户。这个用户必须首先和应用程序网关建立一个 Telnet 会话。在网关（网关监听进入的 Telnet 会话）上一直运行的应用程序提示用户输入用户 ID 和口令。当这个用户提供这些信息时，应用程序网关检查这个用户是否得到许可向外 Telnet。如果没有，网关则中止这个内部用户向该网关发起的 Telnet 连接。如果该用户得到许可，则这个网关：①提示用户输入它所要连接的外部主机的主机名；②在这个网关和某外部主机之间建立一个 Telnet 会话；③将从这个用户到达的所有数据中继到该外部主机，并且把来自这个外部主机的所有数据都中继给这个用户。所以，该 Telnet 应用程序网关不仅执行用户授权，而且同时充当一个 Telnet 服务器和一个 Telnet 客户，在这个用户和该远程 Telnet 服务器之间中继信息。注意到过滤器因为该网关发起向外部的 Telnet 连接，将允许执行步骤②。

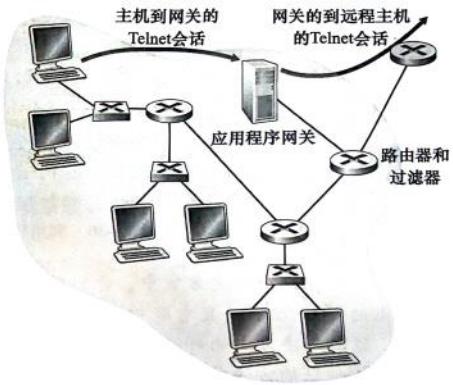


图 8-35　由应用程序网关和过滤器组成的防火墙

应用程序网关并非没有缺点。首先，每一个应用程序都需要一个不同的应用程序网关。第二，因为所有数据都由网关转发，付出的性能负担较重。当多个用户或应用程序使用同一个网关计算机时，这成为特别值得关注的问题。最后，当用户发起一个请求时，客户软件必须知道如何联系这个网关，并且必须告诉应用程序网关如何连接到哪个外部服务器。

- **入侵检测系统与防火墙的主要区别?**

防火墙（Firewall）

1. **定义**：防火墙是一种网络安全系统，用于监控和控制进出网络的流量，根据预定

的安全规则来允许或拒绝数据包。

2. **功能**：防火墙主要执行以下任务：
    - 过滤数据包：检查进入和离开网络的数据包，根据规则决定是否允许它们通过。
    - 阻止未授权访问：防止未授权用户访问内部网络资源。
    - NAT（网络地址转换）：允许多个设备共享单个公网IP地址。
3. **主动性**：防火墙通常是主动的，因为它实时监控网络流量并根据规则做出决策。
4. **目的**：防火墙的目的是防止恶意流量进入网络，并保护网络不受外部威胁。

入侵检测系统（IDS）

1. **定义**：IDS是一种用于检测网络或系统中未授权或恶意活动的系统。
2. **功能**：IDS执行以下任务：
    - 监控网络流量：实时或近实时地监控网络流量，寻找可疑行为。
    - 签名匹配：与已知的攻击模式（签名）进行匹配，以识别潜在的攻击。
    - 异常检测：分析流量模式，识别与正常行为显著不同的异常行为。
3. **主动性**：IDS通常是被动的，它监控流量但不主动阻止流量。它可以发出警报，但不会直接干预。
4. **目的**：IDS的目的是检测和报告潜在的安全威胁，以便采取适当的响应措施。

主要区别

- **干预方式**：防火墙可以阻止恶意流量，而IDS主要提供警报。
- **反应时间**：防火墙通常在数据包到达时立即做出反应，而IDS可能需要时间来分析流量并生成警报。
- **复杂性**：防火墙相对简单，主要基于规则；IDS更复杂，需要分析流量模式和行为。
- **依赖性**：IDS可能依赖于防火墙来执行其检测到的威胁的阻止操作。

总的来说，防火墙和IDS是互补的。防火墙作为第一道防线，阻止未授权访问和恶意流量，而IDS作为第二道防线，监控和报告可能绕过防火墙的攻击。在实际应用中，它们经常一起使用，以提供更全面的网络安全保护。