计算机网络

第1章 概述

1.分组交换

从源端向目的端发送一个报文，源将长报文划分为较小的数据块，称为分组。

源和目的之间通过**分组交换机（路由器和交换机**）传送。

存储转发：分组交换机开始输出该分组第一个bit前，必须接收完整个分组。

2.电路交换

频分复用（FDM）：频率被分为不同频段，每个频段用于不同连接传输数据。

时分复用（TDM）：时间被划分为固定时期的帧，并且每个帧又被分为固定数量时隙。每个帧内一个时隙用于一个连接传输数据。

对比：

分组交换和电路交换为网络链路和交换机移动数据的两种方式。

多数使用分组交换方式，它拥有更好的带宽共享，更简单有效，成本更低。

3.网络中的网络

ISP：互联网服务提供商（Internet Service Provider），电信、移动、联通

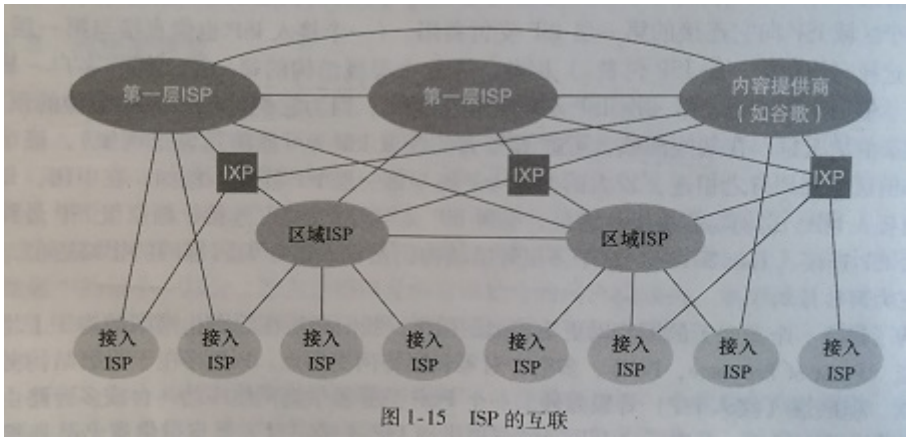


图 1-15 ISP 的互联

4.时延

处理时延：检验分组首部、决定发送方向等时延，一般明确告诉给出。

排队时延：分组在链路上等待传输的时延，不可预测，与拥塞程度有关。

传输时延：将分组bit推向链路所需要的时间，即跟机器有关，与链路传输速度无关。

传播时延：bit在两台路由器链路上传输所需时间，即链路长度/传播速度。

两个节点时延：将4个加起来。

5.吞吐量

就是**实际**传输或者接收数据的速度。

6.层次模型

五层模型：

应用层：报文，HTTP、SMT、DNS、DHCP 报文

传输层：TCP、UDP 报文段

网络层：IP、**ICMP** 报文段

链路层：以太网、WIFI 帧

物理层：传输bit

OSI七层模型（了解）：应用层-表示层-会话层-传输层-网络层-链路层-物理层

7.封装

每一层在上一层交付的数据的基础上加上自己协议的头部，封装实现该层协议功能。

第2章 应用层

1.应用程序体系结构

C/S：客户-服务器结构

P2P：程序在间断连接的主机对之间使用直接通信，服务器被用于跟踪用户IP地址，但报文在主机间直接发送。文件共享、QQ电话、视频通话等等。

2.HTTP

概览

Web页面由对象组成，HTML文件、图片、视频等均称为对象。

使用TCP连接，HTTP服务器不保存关于客户的任何信息，所以叫做无状态协议。

持续连接：所有请求和响应通过一个TCP连接发送。

非持续连接：每次请求与响应对是经过一个单独TCP连接发送。

（HTTP默认采用持续连接，但可以配置为非持续）

报文格式

请求

```
GET /admin_ui/rdx/core/images/close.png HTTP/1.1
Accept: /
Referer: http://xxx.xxx.xxx.xxx/menu/neo
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; .NET4.0C; .NET4.0E)
Accept-Encoding: gzip, deflate
Host: xxx.xxx.xxx.xxx
Connection: Keep-Alive
```

由请求行(request line)、请求头部(header)、空行 和 请求数据(request data) 四个部分组成

请求行包括： 请求方法，URL(包括参数信息)，协议版本这些信息（GET /admin_ui/rdx/core/images/close.png HTTP/1.1）

请求头部

一个个的key-value值，常见请求头（了解）

1. Host

   Host用于指定请求资源的主机名和端口号(可选，默认80)。比如：

   ```
   Host: www.baidu.com
   ```

2. User-Agent

   用户请求的代理软件，包括用户的操作系统，浏览器等相关属性。比如：

   ```
   User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.110 Safari/537.36
   ```

3. Referer

   代表当前访问的URL的上一个URL，也就是用户是从什么地方转到本页面的。比如说html中有图片需要显示，请求图片是就会加上该html所在的url。

   ```
   Referer: https://www.baidu.com/
   ```

4. Cookie

   Cookie是个非常重要的请求头，常用来表示请求者的身份。比如有些会话信息(SesssionId)会存在Cookie中。

   ```
   Cookie:BAIDUID=AAABBBCCCDDDEEEFFFGGG;BIDUPSID=ZYXWVUOPQRST;PSTM=1494145048;__cfduid=d9a1edfb6fa7a6a21167d12a07558b2551494568096;BD_CK_SAM=1;PSINO=1;BD_HOME=1;H_PS_PSSID=1421_21079_21672_20927;BD_UPN=12314353;sugstore=1
   ```

5. Range

   请求实体内容的一部分，多线程下载一定会用到该请求头。

6. Accept

   客户端接受什么类型的信息。类型用MIME表示。

   比如：

   ```
   Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
   ```

7. Accept-Charset

   客户端接受什么字符集的文本内容。常用的如UTF-8，GBK，iso-8859-1等。

8. Accept-Language

   客户端接受什么语言的文本内容。

9. Accept-Encoding

   客户端接受什么压缩格式的内容。如gzip压缩格式，deflate压缩，sdch压缩。

空行(CR+LF)：请求报文用空行表示header和请求数据的分隔

请求数据：GET方法没有携带数据， POST方法会携带一个body，用于表单数据

响应

HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Encoding: gzip
Content-Type: text/html
Date: Fri, 12 Oct 2018 06:36:28 GMT
Server: BWS/1.1

状态行，响应头，空行，数据(响应体)

状态行包括：HTTP版本号，状态码和状态值组成。

响应头类似请求头，是一系列key-value值

空白行：同上，响应报文也用空白行来分隔header和数据

响应体：响应的data

响应码（了解）

- 1XX：信息提示。表示请求已被服务器接受，但需要继续处理，范围为100~101。
- 2XX：请求成功。服务器成功处理了请求。范围为200~206。
- 3XX：客户端重定向。重定向状态码用于告诉客户端浏览器，它们访问的资源已被移动，并告诉客户端新的资源位置。客户端收到重定向会重新对新资源发起请求。范围为300~305。
- 4XX：客户端信息错误。客户端可能发送了服务器无法处理的东西，比如请求的格式错误，或者请求了一个不存在的资源。范围为400~415。
- 5XX：服务器出错。客户端发送了有效的请求，但是服务器自身出现错误，比如Web程序运行出错。范围是500~505。、

3.SMTP&POP3&IMAP

SMTP 的全称是“Simple Mail Transfer Protocol”，即简单邮件传输协议。它是**一组用于从源地址到目的地址传输邮件的规范，通过它来控制邮件的中转方式。**SMTP 协议属于 TCP/IP 协议簇，它帮助每台计算机在发送或中转信件时找到下一个目的地。SMTP 服务器就是遵循 SMTP 协议的发送邮件服务器。

POP3是Post Office Protocol 3的简称，即邮局协议的第3个版本,它**规定怎样将个人计算机连接到Internet的邮件服务器和下载电子邮件的电子协议。**它是因特网电子邮件的第一个离线协议标准,POP3允许用户从服务器上把邮件存储到本地主机（即自己的计算机）上,同时删除保存在邮件服务器上的邮件，而POP3服务器则是遵循POP3协议的接收邮件服务器，用来接收电子邮件的。

IMAP全称是Internet Mail Access Protocol，即交互式邮件存取协议，它是跟POP3类似邮件访问标准协议之一。不同的是，**开启了IMAP后，您在电子邮件客户端收取的邮件仍然保留在服务器上，同时在客户端上的操作都会反馈到服务器上**，如：删除邮件，标记已读等，服务器上的邮件也会做相应的动作。所以无论从浏览器登录邮箱或者客户端软件登录邮箱，看到的邮件以及状态都是一致的。

区别：

POP3协议允许电子邮件客户端下载服务器上的邮件，但是在客户端的操作（如移动邮件、标记已读等），不会反馈到服务器上，比如通过客户端收取了邮箱中的3封邮件并移动到其他文件夹，邮箱服务器上的这些邮件是没有同时被移动的 。

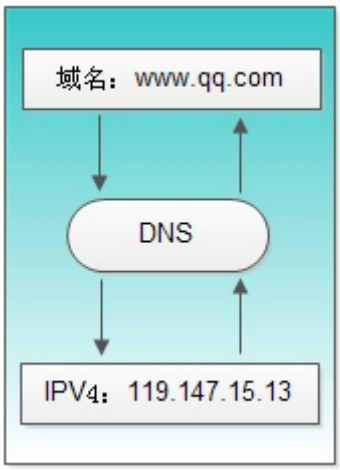
而IMAP提供webmail 与电子邮件客户端之间的双向通信，客户端的操作都会反馈到服务器上，对邮件进行的操作，服务器上的邮件也会做相应的动作。

4.DNS

概述

域名系统：进行主机名到IP地址转换的目录服务。

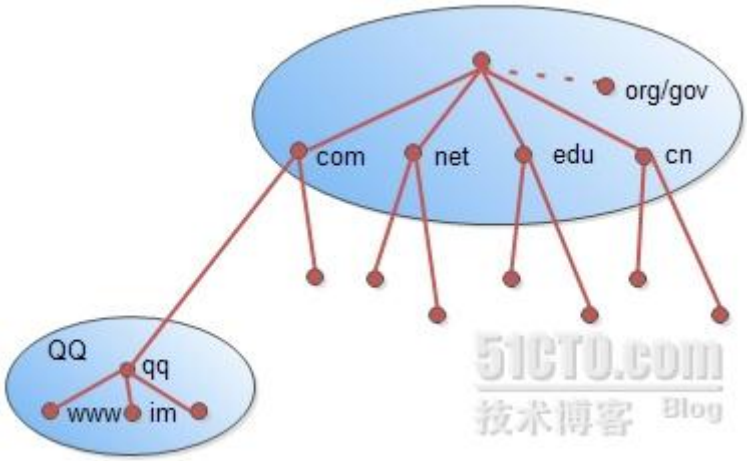
使用UDP，工作在53号端口。



DNS域名称

域名系统作为一个层次结构和分布式数据库，包含各种类型的数据，包括主机名和域名。DNS数据库中的名称形成一个**分层树状结构**称为域命名空间。域名包含单个标签分隔点，例如：im.qq.com。

完全限定的域名 (FQDN) 唯一地标识在 DNS 分层树中的主机的位置，通过指定的路径中点分隔从根引用的主机的名称列表。 下图显示与主机称为 im 内 qq.com DNS 树的示例。 主机的 FQDN 是 im.qq.com。

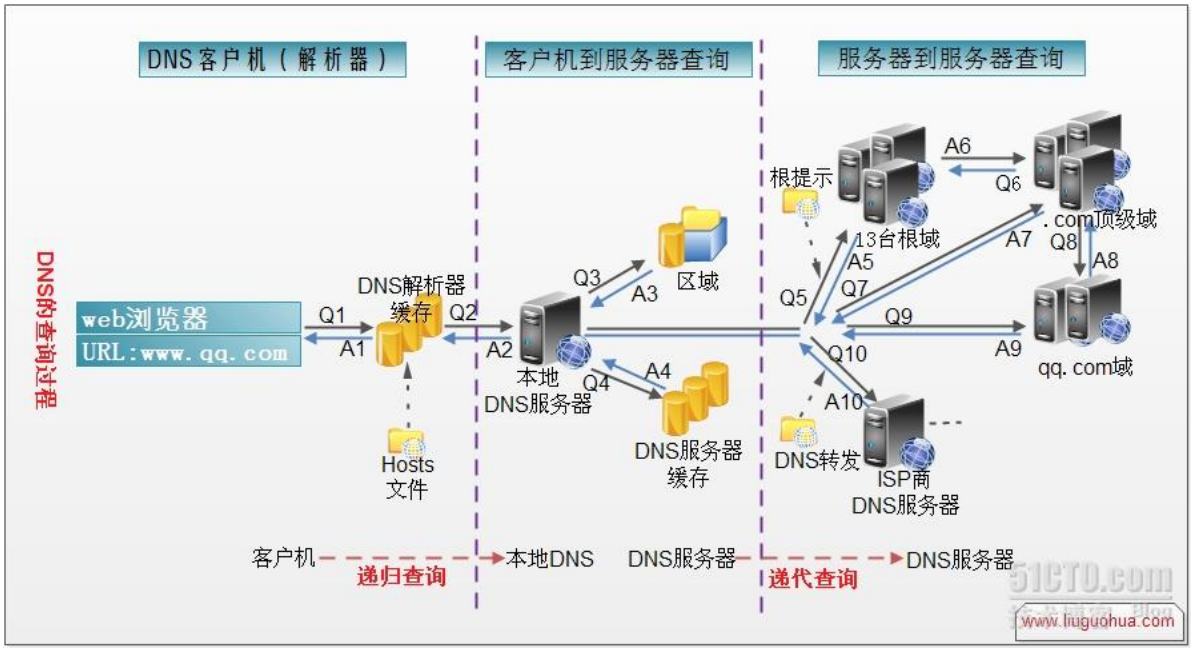


DNS域名称空间的组织方式

名称类型	说　明	示　例
根域	DNS域名中使用时，规定由尾部句点(.)来指定名称位于根或更高级别的域层次结构	单个句点(.)或句点用于末尾的名称
顶级域	用来指示某个国家/地区或组织使用的名称的类型名称	.com
第二层域	个人或组织在Internet上使用的注册名称	qq.com
子域	已注册的二级域名派生的域名，通俗的讲就是网站名	www.qq.com
主机名	通常情况下，DNS域名的最左侧的标签标识网络上的特定计算机，如h1	h1.www.qq.com

DNS域名称	组织类型
com	商业公司
edu	教育机构
net	网络公司
gov	非军事政府机构
Mil	军事政府机构
xx	国家/地区代码(cn表中国)
….	….

Dns服务的工作过程



1、在浏览器中输入www.qq.com域名，操作系统会**先检查自己本地的hosts文件是否有这个网址映射关系**，如果有，就先调用这个IP地址映射，完成域名解析。

2、如果hosts里没有这个域名的映射，则**查找本地DNS解析器缓存**，是否有这个网址映射关系，如果有，直接返回，完成域名解析。

3、如果hosts与本地DNS解析器缓存都没有相应的网址映射关系，首先会找TCP/IP参数中设置的**首选DNS服务器**，在此**我们叫它本地DNS服务器**，此服务器收到查询时，如果要查询的域名，包含在本地配置区域资源中，则返回解析结果给客户机，完成域名解析，此解析具有权威性。

4、如果要查询的域名，不由本地DNS服务器区域解析，但该服务器已缓存了此网址映射关系，则调用这个IP地址映射，完成域名解析，此解析不具有权威性。

5、如果本地DNS服务器本地区域文件与缓存解析都失效，则根据本地DNS服务器的设置（是否设置转发器）进行查询，如果未用转发模式，**本地DNS就把请求发至13台根DNS**，根DNS服务器收到请求后会判断这个域名(.com)是谁来授权管理，并**会返回一个负责该顶级域名服务器的一个IP**。本地DNS服务器收到IP信息后，将会联系负责.com域的这台服务器。这台负责.com域的服务器收到请求后，如果自己无法解析，它就会找一个管理.com域的下一级DNS服务器地址(qq.com)给本地DNS服务器。当本地DNS服务器收到这个地址后，就会找qq.com域服务器，重复上面的动作，进行查询，直至找到www.qq.com主机。

6、如果用的是转发模式，此DNS服务器就会把请求转发至上一级DNS服务器，由上一级服务器进行解析，上一级服务器如果不能解析，或找根DNS或把转请求转至上上级，以此循环。不管是本地DNS服务器用是是转发，还是根提示，最后都是把结果返回给本地DNS服务器，由此DNS服务器再返回给客户机。

DNS缓存

DNS服务器收到一个DNS回答后会将映射缓存到本地存储器中。

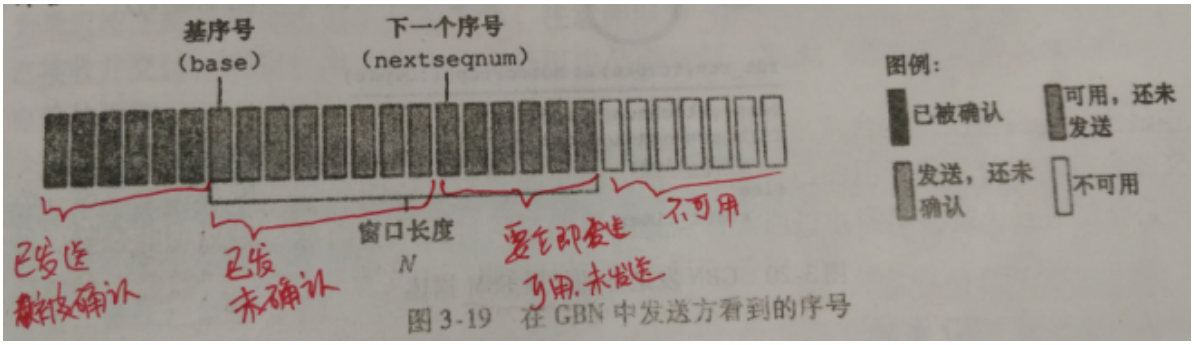
在一段时间后（通常两天）丢弃缓存信息。

第3章 传输层

1.可靠数据传输原理

GBN 回退N步/滑动窗口协议

在 GBN 协议中，允许发送方发送多个分组（当有多个分组可用时）而不需等待确认，但它也受限于在流水线中未确认的分组数不能超过某个最大允许数 N。



将**基序号（base）**定义为**最早的未确认分组**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确认的分组。[base, nextseqnum-1] 段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立即发送的分组，如果有数据来自于上层的话。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确认的分组（特别是序号为 base 的分组）已得到确认为止。

超时：使用定时器，如果出现超时，**发送方将重传所有已发送但还未确认过的分组。**

接收方接收到序号为n的分组后，为序号为n的分组发送一个ACK，**如果分组 k 为已接受并交付，则所有序号比 k 小的分组也已经交付。**因此，使用累积确认是 GBN 的一个自然的选择。

选择重传

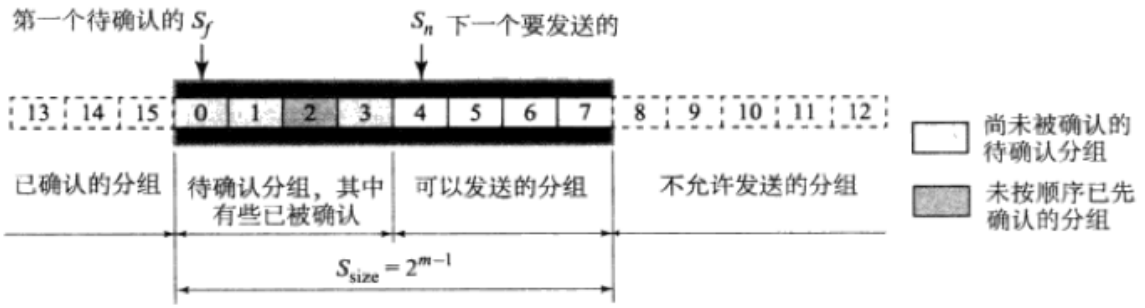


图 13.31 选择重传协议的发送窗口

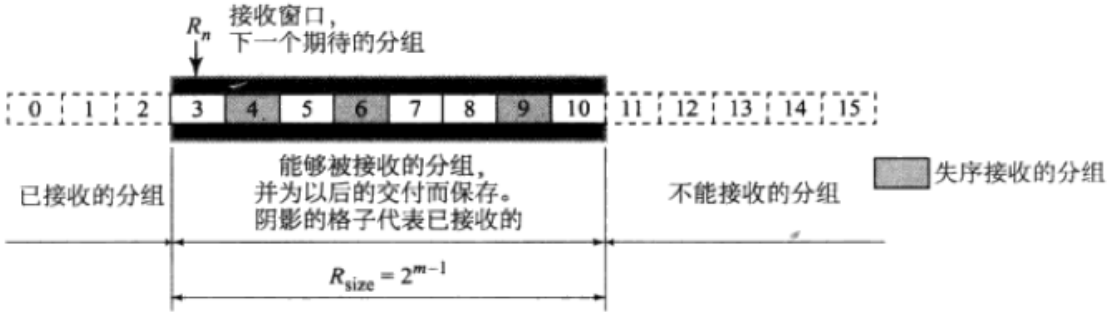


图 13.32 选择重传协议的接收窗口

选择重传的**接收窗口与发送窗口一样大**.选择重传协议允许与接收窗口一样多的分组**失序到达**,并保存这些失序到达的分组,**直到连续的一组分组被交付给应用层**.因为发送窗口与接收窗口是相同的,所以发送出来的所有分组都可以失序到达,而且会被保留直到交付为止.但是必须强调一点,在一个可靠的协议中,接收方永远不会把分组失序地交给应用层.在他们被交付给应用层之前,先要等待那些更早发出来的分组到达.

理论上选择重传协议要为**每个分组使用一个计时器**.当某个计时器超时后,只**有相应的分组被重传**.换而言之,返回N协议将所有的分组当做一个整体对待,而选择重传协议则分别对待每一个分组.但是大多数SR的运输层仅使用了一个计时器. 注意只使用一个计时器而做到跟踪所有发出去的分组的情况的做法是:标记发出分组,当ACK=Sf 时,**将窗口滑过所有连续的已确认的分组,如果还有未确认的分组,则重发所有检测到的未被确认的分组并重启计时器,如果所有分组都被确认了则停止计时器**.

在GBN中确认值(ACK) 是累计的,它定义了下一个希望接收的分组序号,同时也证实了此前所有的分组都已经被完好的接收了.在SR中,确认号(ACK)**只定义完好接收的那一个分组的序号**,并不反馈任何其他分组的信息.

2.TCP

概述

面向连接

全双工

点对点

缓存数据

MTU：最大传输单元，最大链路层帧长度；以太网为1500B

报文结构

●源、目标端口号字段：占16比特。TCP协议通过使用"端口"来标识源端和目标端的应用进程。

●顺序号字段：占32比特。用来标识从TCP**源端**向TCP**目标端发送的数据字节流**，它**表示在这个报文段中的第一个数据字节。**

●确认号字段：占32比特。只有ACK标志为1时，确认号字段才有效。**它包含目标端所期望收到源端的下一个数据字节。**

●头部长度字段：占4比特。给出头部占32比特的数目。没有任何选项字段的TCP头部长度为20字节；最多可以有60字节的TCP头部。

●标志位字段（U、A、P、R、S、F）：占6比特。各比特的含义如下：

◆URG：紧急指针（urgent pointer）有效。

◆ACK：为1时，确认序号有效。

◆PSH：为1时，接收方应该尽快将这个报文段交给应用层。

◆RST：为1时，重建连接。

◆SYN：为1时，同步程序，发起一个连接。

◆FIN：为1时，发送端完成任务，释放一个连接。

●窗口大小字段：占16比特。此字段用来进行**流量控制。**单位为字节数，这个值是**本机期望一次接收的字节数。**

●TCP校验和字段：占16比特。**对整个TCP报文段，即TCP头部和TCP数据进行校验和计算，并由目标端进行验证。**

●紧急指针字段：占16比特。它是一个偏移量，和序号字段中的值相加表示紧急数据最后一个字节的序号。

●选项字段：占32比特。可能包括"窗口扩大因子"、"时间戳"等选项。

连接过程

序号和确认号

- 字节序号

TCP 连接中，为传送的字节流（数据）中的**每一个字节按顺序编号。**也就是说，**在一次 TCP 连接建立的开始，到 TCP 连接的断开**，你要传输的所有数据的每一个字节都要编号。这个序号称为**字节序号。**

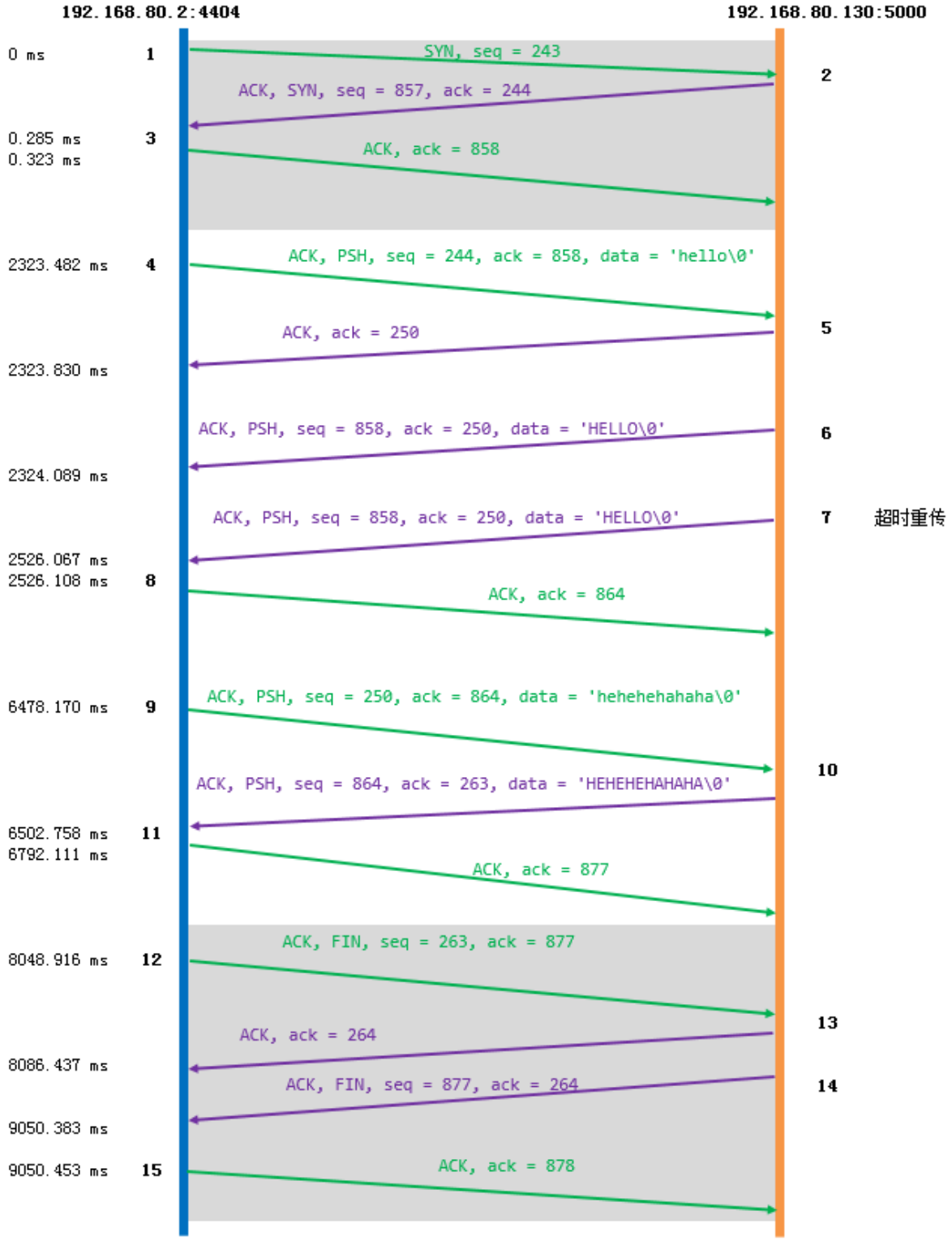
- 报文段序号

如果一个 TCP 报文段的序号为 301，它携带了 100 字节的数据，就表示这 100 个字节的数据的字节序号范围是 [301, 400]，该报文段携带的第一个字节序号是 301，最后一个字节序号是 400.

在 TCP 协议中，一般采用**累积确认**的方式，即每传送多个连续 TCP 段，可以只对最后一个 TCP 段进行确认。

对方通过回复一个确认号，来表示确认已经接收到了哪个 TCP 段。比如发送方发送了一个**报文段序号**为 301 的 TCP 段，这个段携带了 100 字节数据，则接收方应当回复的确认号是 401，它表示接收方已经收到了**字节序号为** [0, 400] 的数据，现在期望你发送字节序号为 401 以及以后的数据。

（只有当 ACK 标志位被置位的时候，确认号这个字段才有效。）



三次握手：

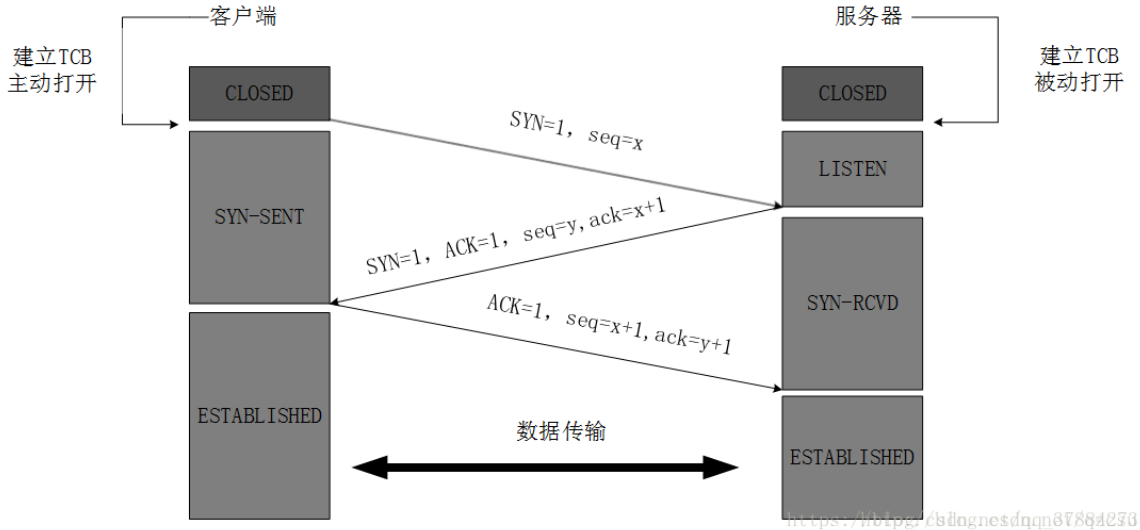
TCP是一个面向连接的协议，无论哪一方向另一方发送数据之前，都必须先在双方之间建立一条连接，建立一条连接有以下过程。

1、请求端（客户端）**发送一个SYN段**指明客户打算连接的服务器的端口，以及初始序列号（ISN），**这个SYN为报文段1**.

2、服务器发回包含服务器的初始序列号的SYN报文段（报文段2）作为应答。同时，将确认序号设置为**客户的ISN加1**以对客户的SYN报文段进行确认。一个SYN将占用一个字符。

3、客户必须将明确序号设置为服务器的**ISN加1**以对服务器的SYN报文段进行确认（报文段3）

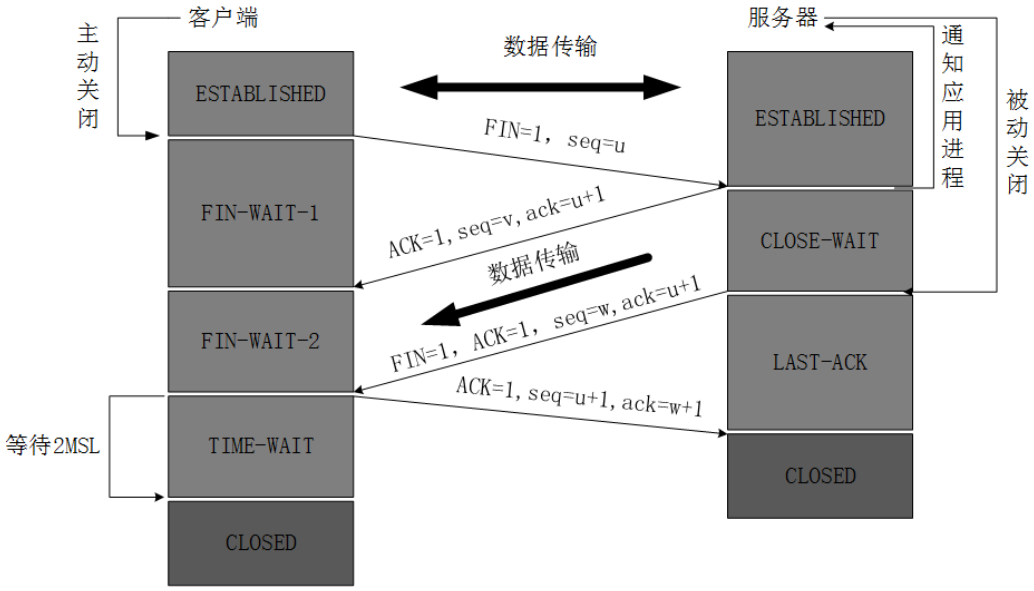
4、这三个报文段完成连接的建立，这个过程成为三次握手。



四次挥手：

1、客户端进程发出连接释放报文，并且停止发送数据。释放数据报文首部，**FIN=1，其序列号为seq=u（等于前面已经传送过来的数据的最后一个字节的序号加1）**，此时，客户端进入FIN-WAIT-1（终止等待1）状态。 TCP规定，FIN报文段即使不携带数据，也要消耗一个序号。
2、服务器收到连接释放报文，**发出确认报文，ACK=1，ack=u+1，并且带上自己的序列号seq=v**，此时，服务端就进入了CLOSE-WAIT（关闭等待）状态。TCP服务器通知高层的应用进程，客户端向服务器的方向就释放了，这时候处于半关闭状态，即客户端已经没有数据要发送了，但是服务器若发送数据，客户端依然要接受。这个状态还要持续一段时间，也就是整个CLOSE-WAIT状态持续的时间。
客户端收到服务器的确认请求后，此时，客户端就进入FIN-WAIT-2（终止等待2）状态，等待服务器发送连接释放报文（在这之前还需要接受服务器发送的最后的数据）。
3、**服务器将最后的数据发送完毕后**，就向客户端发送连接释放报文，**FIN=1，ack=u+1**，由于在半关闭状态，服务器很可能又发送了一些数据，假定此时的序列号为seq=w，此时，服务器就进入了LAST-ACK（最后确认）状态，等待客户端的确认。
客户端收到服务器的连接释放报文后，必须发出确认，ACK=1，ack=w+1，而自己的序列号是seq=u+1，此时，客户端就进入了TIME-WAIT（时间等待）状态。注意此时TCP连接还没有释放，必须经过2*MSL（最长报文段寿命）的时间后，当客户端撤销相应的TCB后，才进入CLOSED状态。
4、服务器只要收到了客户端发出的确认，立即进入CLOSED状态。同样，撤销TCB后，就结束了这次的

TCP连接。可以看到，服务器结束TCP连接的时间要比客户端早一些。



3.往返时延计算

P158讲得很清楚

4.TCP拥塞控制

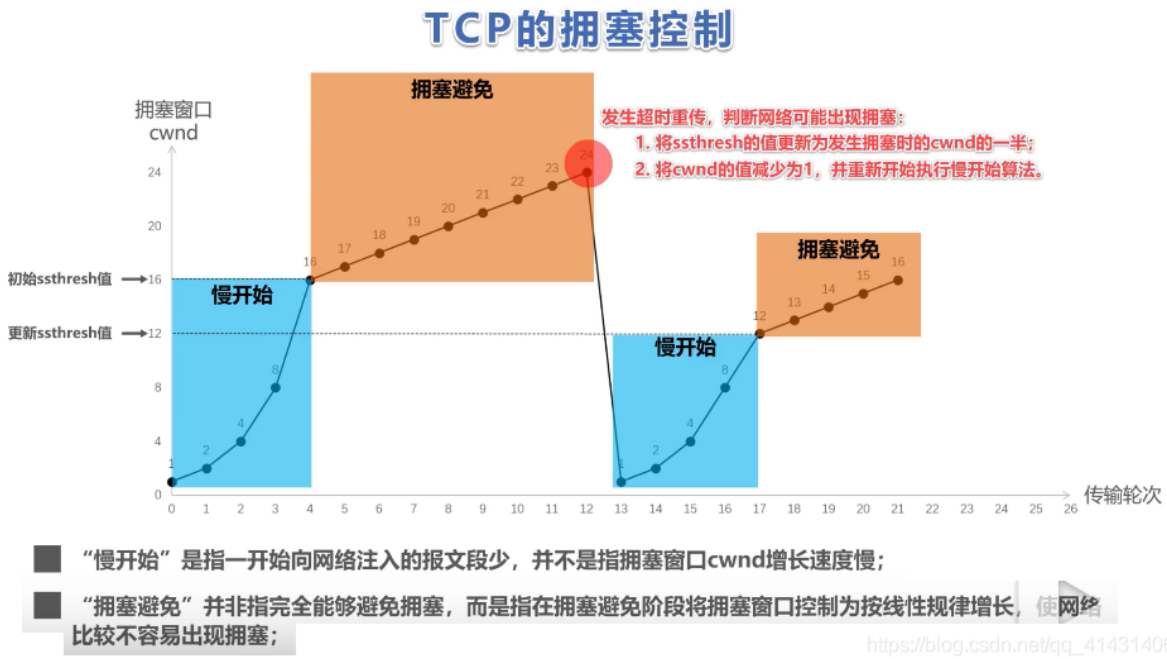
慢开始：
假设当前发送方拥塞窗口cwnd的值为1，而发送窗口swnd等于拥塞窗口cwnd，因此发送方当前只能发送一个数据报文段（拥塞窗口cwnd的值是几，就能发送几个数据报文段），接收方收到该数据报文段后，给发送方回复一个确认报文段，发送方收到该确认报文后，将拥塞窗口的值变为2，

发送方此时可以连续发送**两个**数据报文段，接收方收到该数据报文段后，给发送方一次发回2个确认报文段，发送方收到这两个确认报文后，将拥塞窗口的值加**2变为4**，发送方此时可连续发送4个报文段，接收方收到4个报文段后，给发送方依次回复4个确认报文，发送方收到确认报文后，将拥塞窗口加4，**置为8**，发送方此时可以连续发送8个数据报文段，接收方收到该8个数据报文段后，给发送方一次发回8个

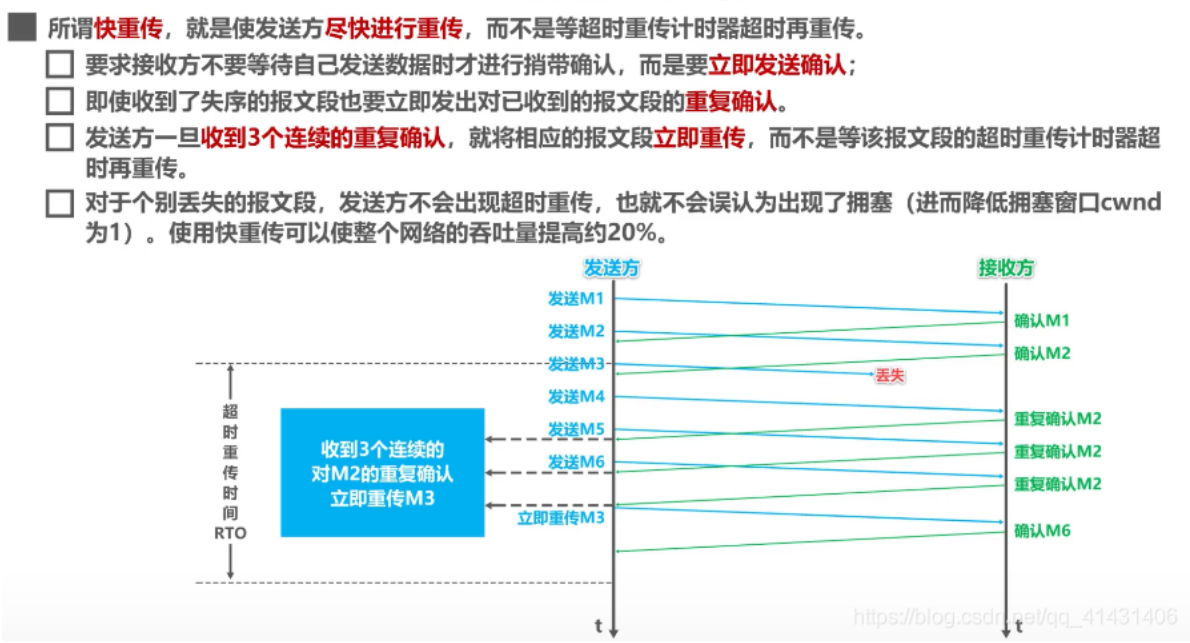
确认报文段，发送方收到这8个确认报文后，将拥塞窗口的值加**8变为16**，

当前的拥塞窗口cwnd的值已经**等于慢开始门限值，之后改用拥塞避免算法。**

拥塞避免：
也就是每个传输轮次，拥塞窗口cwnd只能**线性加一**，而不是像慢开始算法时，每个传输轮次，**拥塞窗口cwnd按指数增长。**同理，16+1……直至到达24，假设24个报文段在传输过程中丢失4个，接收方只收到20个报文段，给发送方依次回复20个确认报文段，一段时间后，丢失的4个报文段的重传计时器超时了，发送发判断可能出现拥塞，更改cwnd和ssthresh（慢启动阈值）.并重新开始慢开始算法，如图所示：



快速重传



快速恢复

■ 发送方一旦收到3个重复确认，就知道现在只是丢失了个别的报文段。于是不启动慢开始算法，而执行快恢复算法；

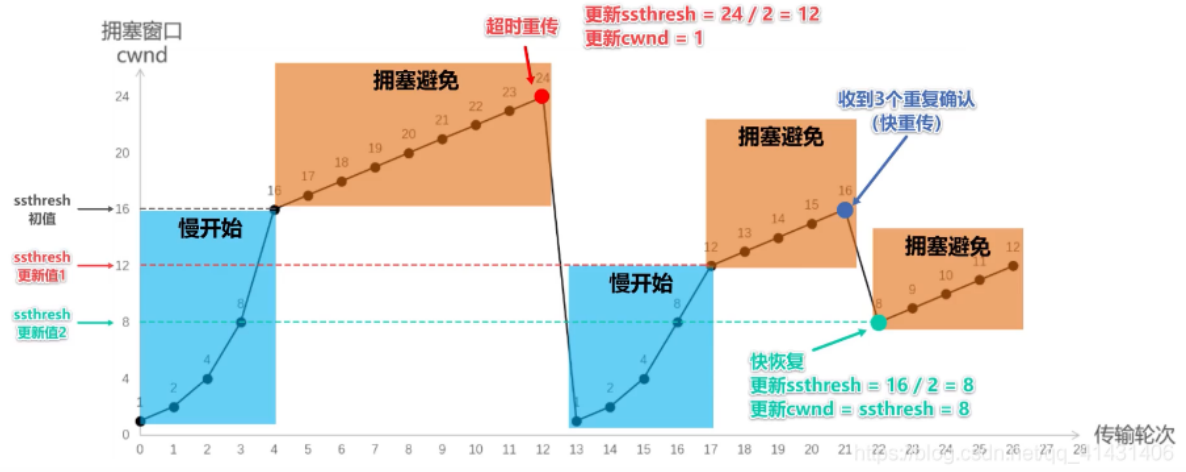
□ 发送方将慢开始门限ssthresh值和拥塞窗口cwnd值调整为当前窗口的一半；开始执行拥塞避免算法。

□ 也有的快恢复实现是把快恢复开始时的拥塞窗口cwnd值再增大一些，即等于新的ssthresh + 3。

◇ 既然发送方收到3个重复的确认，就表明有3个数据报文段已经离开了网络；

◇ 这3个报文段不再消耗网络资源而是停留在接收方的接收缓存中；

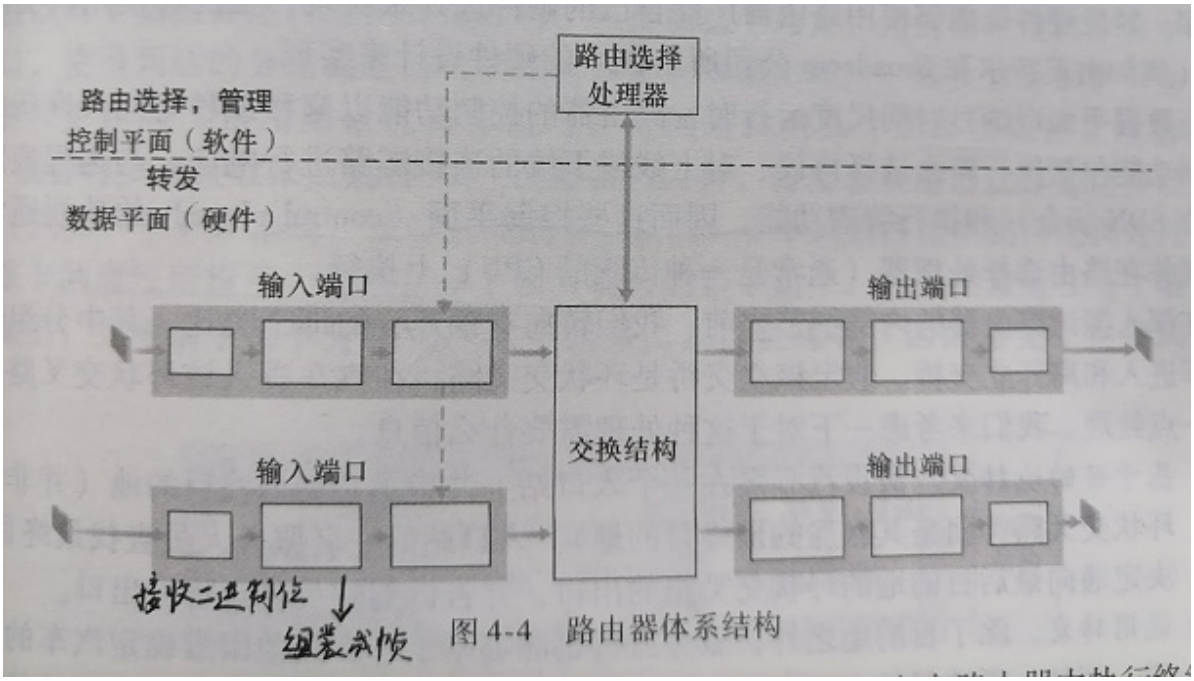
◇ 可见现在网络中不是堆积了报文段而是减少了3个报文段。因此可以适当把拥塞窗口扩大些。



第4章 网络层

数据平面

1.路由器工作原理



输入端口

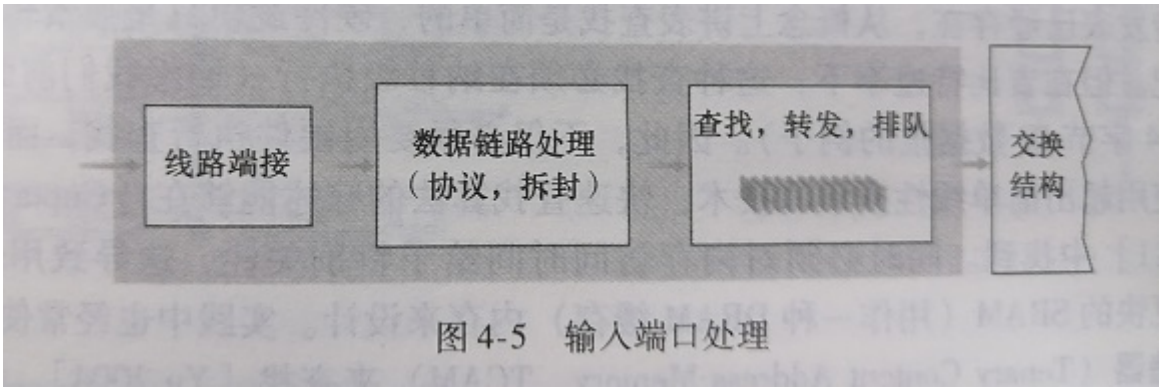


图 4-5　输入端口处理

交换结构

4.2.2　交换

交换结构位于一台路由器的核心部位，因为正是通过这种交换结构，分组才能实际地从一个输入端口交换（即转发）到一个输出端口中。交换可以用许多方式完成，如图 4-6 所示。

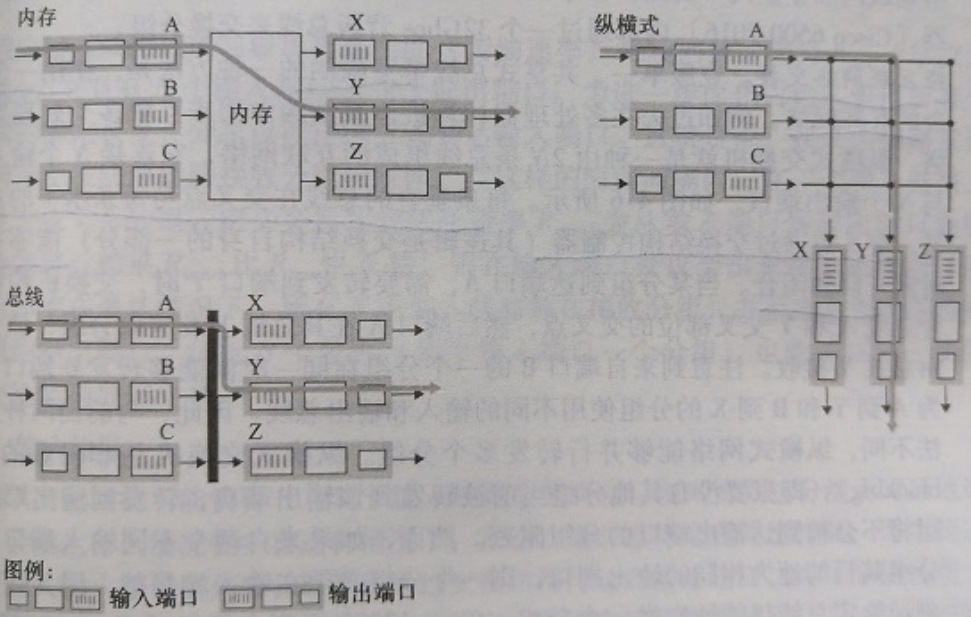


图 4-6　三种交换技术

输出端口

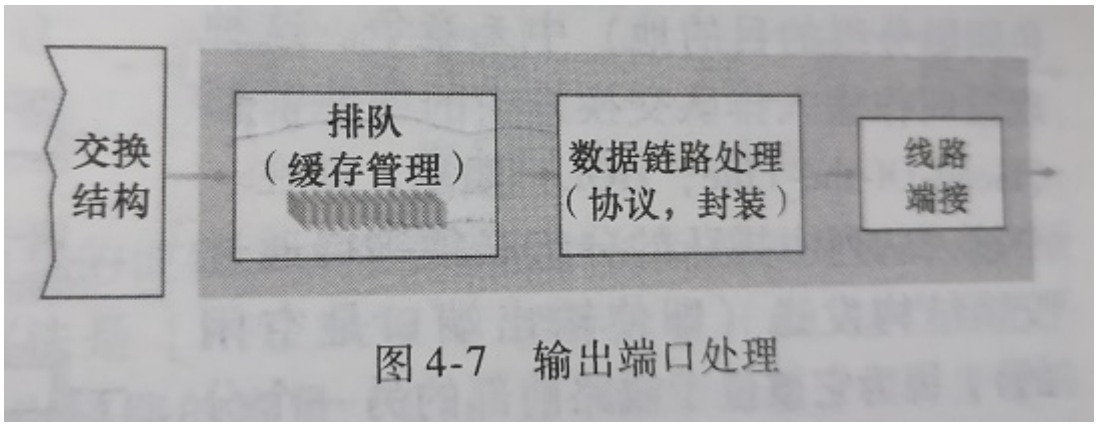


图 4-7　输出端口处理

基于目的的转发

路由器用分组目的地址的前缀与表中表项匹配，有多个匹配时，按最长前缀匹配。

2.IPv4

①版本号：ip协议的版本，字段占4位

②首部长度：字段占4位，给出的最大值为15。但是，从图中我们可以看到，首部有5行，每行为32位4个字节（未包括可变的选项字段），总共有20个字节，最大值15肯定是无法满足的，所以规定，首部长度这四位，是以4字节位为单位的（需要乘以4）字节

③服务类型tos：指示期望获得哪种类型的服务，一般情况下不用，通常ip分组该字段的值为00H

④总长度：占16位，ip分组的**总字节数（首部+数据）**

最大ip分组的总长度，65535B；

最小的ip分组首部：20B；

最大的ip封装数据：65535-20=65515B

注意，在实际中不会有这种理想情况

⑤生存时间TTL：占8位，IP分组在网络中可以通过的**路由器数或跳步数**

路由器每转发一次分组，TTL减1

当TTL=0，说明寿命到了，**路由器将丢弃该IP分组**

⑥协议：指示ip分组封装的是哪个协议的数据包

可以实现复用/分解

6为TCP，表示封装的是TCP段

17为UDP，表示封装的是UDP段

⑦首部校验和：占16位，实现队IP分组**首部**的差错检测

计算校验和时，该字段置全0（与UDP差错检测是一样的）

关键：采用**反码算数运算求和（进位1加到第一位），和的求反码**作为首部校验和字段，结果全为1则是正确的

注意，**因为首部校验和是对首部进行校验，每一次路由器转发分组时，TTL都会减1，所以每次都要重新计算**

因为TCP校验和已经对TCP首部和数据进行了校验。

⑧源IP地址和目的IP地址：各占32位4个字节

发送分组的源主机/路由器（网络接口）

接收分组的目的主机/路由器（网络接口）

⑨选项字段：长度可变，范围在1~40B之间

用于网络的探测，可以携带安全、源选路径、时间戳、路由记录等内容

注意，这个字段实际上很少使用，因此，典型的ip数据报是20个字节（4个字节x5行）

⑩填充：首部是以4字节位单位的，因此要补足

数据报分片

发送主机通常发送的每一个数据报标识号+1

同一片具有相同标识号，用标识号区分哪些片为一个数据报分来的

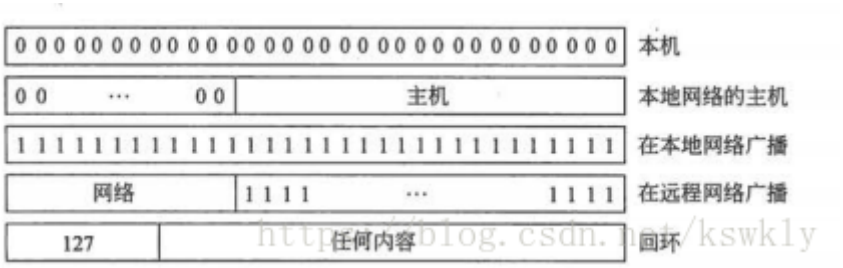
最后一片标志置位0，其他为1

偏移字段指定该片放在数据报哪个位置

IPv4编址

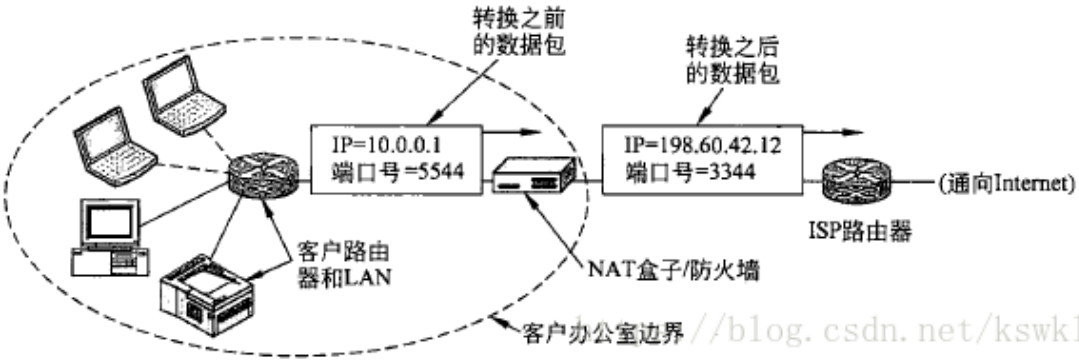
	10进制	2进制
IP地址	192 . 168 . 1 . 1	11000000 . 10101000.00000001 . 00000001
子网掩码	255 . 255 . 255 . 0	11111111 . 11111111.11111111 . 00000000

网络地址　　主机地址



NAT网络地址转换

IPv4的地址已经不够用了，但是ipv6还没有被普及，所以为了解决这个问题人们引入了NAT（network address translation），它的基本思路就是给每个家庭或者公司分配一个IP或者尽量少的IP，用这个IP地址来传输数据，这客户网络内部，每台计算机有唯一的一个内部IP地址，该地址用来传输内部数据。当客户网络要给外部的网络发送数据时，它首先进行一个地址转换，把内部IP地址转换成那个共享的公共IP地址,然后再进行发送。如下：（会更改源端口号）



内部IP的范围是固定的：
10.0.0.0~10.255.255.255/8
172.16.0.0~172.31.255.255/12
192.168.0.0~192.168.255.255/16

3.DHCP

DHCP(Dynamic Host Configuration Protocol),动态主机配置协议，是一个应用层协议。当我们将客户主机ip地址设置为动态获取方式时，DHCP服务器就会根据DHCP协议给客户端分配IP，使得客户机能够利用这个IP上网。

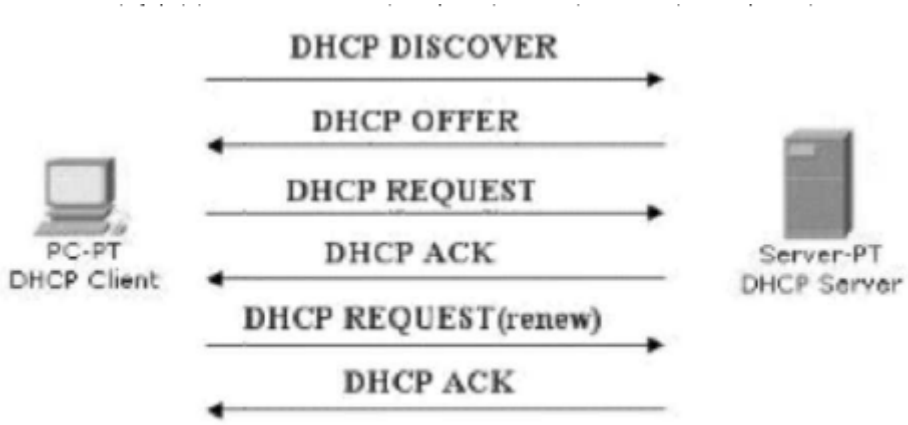


图 1 DHCP Client 与 DHCP Server 交互图

DHCP的实现分为4步，分别是：
第一步：Client端在局域网内发起一个DHCP　Discover包，目的是想发现能够给它提供IP的DHCP Server。
第二步：可用的DHCP Server接收到Discover包之后，通过发送DHCP Offer包给予Client端应答，意在告诉Client端它可以提供IP地址。
第三步：Client端接收到Offer包之后，发送DHCP Request包请求分配IP。
第四步：DHCP Server发送ACK数据包，确认信息。

控制平面

1.路由选择算法

集中式

Dijkstra（p246 填表）

分散式

距离向量（DV）

每个节点从直接相连的邻居接收信息，并计算到各个节点最短距离，每次更新后从新计算，直至没有信息交换。

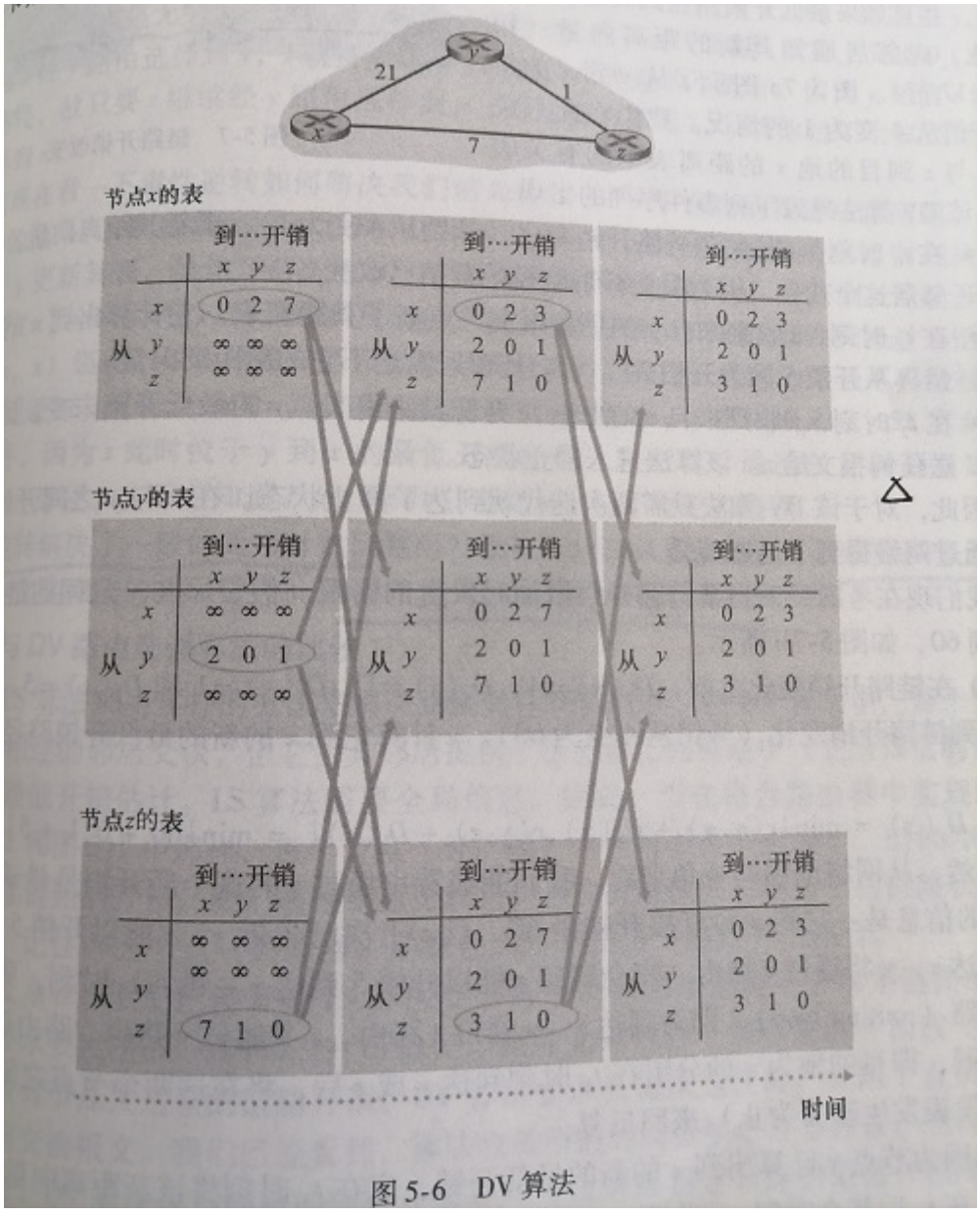
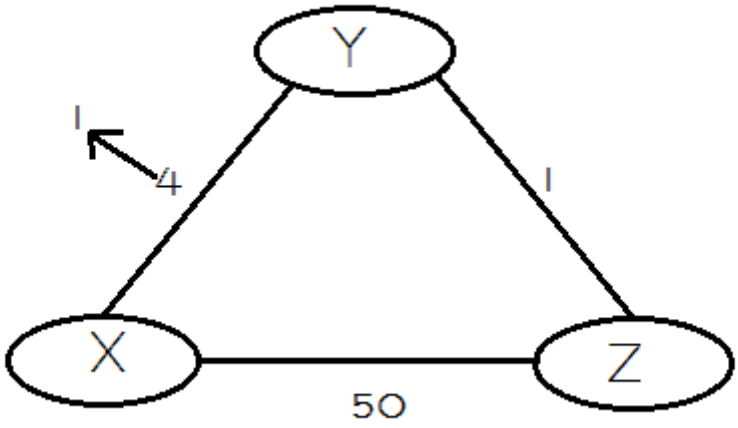


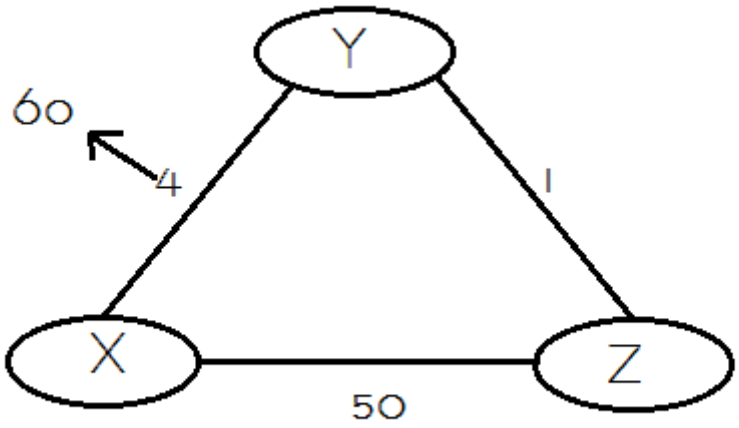
图 5-6　DV 算法

当某条链接的费用减少时，我们称之为有一个“好消息”。在网络中，**好消息的传递往往很迅速。**
例如，存在这样一个网络：



某一时刻，Y检测到它到X的链路费用由4减少为1，好消息当然要告诉大家了，于是它更新了自己的距离向量，并通知了Z。Z在收到Y的更新报文后，也更新了自己的距离向量（由5减为2），并向邻居们发送更新报文。而后，Y又收到了Z的更新报文，但它发现并没有改变自己的最低费用，于是保持不变。这样，仅仅经过了**两次迭代**网络就达到了静止。好消息通过网络得到了迅速传播。

但是，当链路费用增加（甚至断开）时，就不会这么简单了。
我们看下面这个例子：



还是X、Y、Z三个节点。此时Y检测到它到X的路径费用由4增加到了60。此时节点Z的距离向量为：d(X) = 5, d(Y) = 1, d(Z) = 0。于是Y在更新向量时发现，咦，Z到X的距离只有5诶，那可以先到Z再到X，于是Y的距离向量更新为：d(x) = 5 + 1 = 6, d(Y) = 0, d(z) = 1。我们可以发现，这个逻辑显然是错误的，**因为Z到X的距离为5的前提是要经过Y，但Y更新后的路径又要经过Z，这就形成了一个选路环路**（routing-loop）问题。因为Y的距离向量更新了（虽然是错误的），但它还是向Z发送了更新报文。Z收到更新报文后，比较了下邻居们到X的距离，发现经过Y的路径距离为1 + 6 = 7，小于直接到X的距离，于是Z也更新的自己的距离向量，然后又将更新后的距离向量发给Y。Y收到后又更新向量为8，然后再发给Z。。。这样循环往复，更新报文在Y和Z之间传来传去，直到**第44次迭代后**，Z算出它经由Y的路径费用大于50为止。此时，Z最终确定到X的最短路径费用是直接到达X的费用50，而Y也得到了最短路径是经Z到X的费用51。
可以看出，虽然最后还是得到了正确的信息（最后的50和51是正确的！），但坏消息的传播与好消息相比实在是慢太多了！而且，如果X和Y之间的费用为10000，Z和X的费用是9999时，就会出现计数到无穷（count-to-infinity）的问题！

毒性逆转方法（The Reverse-Poison(Split-horizon) Hack）

上述的选路环路问题可以通过毒性逆转的技术加以避免。它的基本思想是：**如果Z的最短路径要通过邻居Y，那么它将告诉Y自己到目的节点的距离是∞。**这样，Z向Y撒了一个善意的谎言，使得只要Z经过Y选路到X，它就会一直持续讲述这个谎言，这样Y也就永远不会尝试从Z选路到X了，也就避免了环路问题。（可见，有时善意的谎言也不是件坏事，也是为了你好啊T^T）
我们将毒性逆转技术应用于上例。Y在更新自己的距离向量时，发现Z到X的距离是∞，于是它将d(x)无奈地更新为60，并向Z发送了更新报文。Z收到报文后更新自己的d(X)为50（直接选路到X），并发给Y更新报文（此时因为Z不需要经过Y进行选路，因此将告诉Y自己到X的距离为50）。Y在接收到Z的报文后，重新将距离更新为1 + 50 = 51，并告诉Z自己到X的距离是∞（实际是51）。Z收到报文后，发现最低耗费并没有改变，因此算法进入静止状态。

但是，当涉及3个或更多节点（而不仅仅是两个直接相连的邻居节点）的环路将不能被毒性逆转技术检测到

2.BGP

AS：自治系统，即不同区域有不同的路由器自治系统，由多个路由器组成的网络。

OSPF：开放最短路优先，自治系统内部路由选择协议。

BGP：边界网关协议，是一种实现自治系统AS（Autonomous System）之间的路由可达，并选择最佳路由的距离矢量路由协议。

目的：
为方便管理规模不断扩大的网络，网络被分成了不同的自治系统。1982年，外部网关协议EGP（Exterior Gateway Protocol）被用于实现在AS之间动态交换路由信息。但是EGP设计得比较简单，只发布网络可达的路由信息，而不对路由信息进行优选，同时也没有考虑环路避免等问题，很快就无法满足网络管理的要求。

BGP是为取代最初的EGP而设计的另一种外部网关协议。不同于最初的EGP，BGP能够进行路由优选、避免路由环路、更高效率的传递路由和维护大量的路由信息。

虽然BGP用于在AS之间传递路由信息，但并不是所有AS之间传递路由信息都需要运行BGP。比如在数据中心上行的连入Internet的出口上，为了避免Internet海量路由对数据中心内部网络的影响，设备采用静态路由代替BGP与外部网络通信。

BGP的优点：
BGP从多方面保证了网络的安全性、灵活性、稳定性、可靠性和高效性。

BGP采用认证和GTSM的方式，保证了网络的安全性。

BGP提供了丰富的路由策略，能够灵活的进行路由选路，并且能指导邻居按策略发布路由。

BGP提供了路由聚合和路由衰减功能由于防止路由震荡，有效提高了网络的稳定性。

BGP使用TCP作为其传输层协议（目的端口号179），并支持与BGP与BFD联动、BGP Tracking和BGP GR和NSR，提高了网络的可靠性。

在邻居数目多、路由量大且大部分邻居具有相同出口的策略的场景下，BGP使用按组打包技术极大的提高了BGP打包发包性能。

BGP基本概念：
自治系统ASAutonomous System：
AS是指在一个实体管辖下的拥有相同选路策略的IP网络。**BGP网络中的每个AS都被分配一个唯一的AS号，用于区分不同的AS。**AS号分为2字节AS号和4字节AS号，其中2字节AS号的范围为1至65535，4字节AS号的范围为1至4294967295。支持4字节AS号的设备能够与支持2字节AS号的设备兼容。

BGP分类：
BGP按照运行方式分为EBGP（External/Exterior BGP）和IBGP（Internal/Interior BGP）。

EBGP：运行于**不同AS之间的BGP称为EBGP**。为了防止AS间产生环路，当BGP设备接收EBGP对等体发送的路由时，会将带有本地AS号的路由丢弃。
IBGP：运行于**同一AS内部的BGP称为IBGP**。为了防止AS内产生环路，BGP设备不将从IBGP对等体学到的路由通告给其他IBGP对等体，并与所有IBGP对等体建立全连接。为了解决IBGP对等体的连接数量太多的问题，BGP设计了路由反射器和BGP联盟。

3.SDN控制平面

基于流的转发：SDN控制交换机分组转发，能够基于传输层、网络层或链路层首部任意数量字段进行，即决定哪个口出去的依据可以有很多共同匹配。

数据平面与控制平面分离：交换机只执行简单的匹配流表，转发功能，控制平面来生成流表。

网络控制：服务器上运行选路算法生成流表。

可编程：服务器上选路算法可编程控制。

OpenFlow

OpenFlow，一种网络通信协议，属于数据链路层，能够控制网上交换器或路由器的转发平面（forwarding plane），借此改变网络数据包所走的网络路径。

OpenFlow能够启动远程的控制器，经由网络交换器，决定网络数据包要由何种路径通过网络交换机。这个协议的发明者，将它当成SDN的启动器。

OpenFlow允许从远程控制网络交换器的数据包转送表，透过新增、修改与移除数据包控制规则与行动，来改变数据包转送的路径。比起用访问控制表 和路由协议，允许更复杂的流量管理。同时，OpenFlow允许不同供应商用一个简单，开源的协议去远程管理交换机（通常提供专有的接口和描述语言）。

OpenFlow协议用来描述控制器和交换机之间交互所用信息的标准，以及控制器和交换机的接口标准。协议的核心部分是用于OpenFlow协议信息结构的集合。

4.ICMP

ICMP：因特网控制报文协议，用于差错报告，如HTTP会话时错误报文、ping命令等

第5章 链路层

1.CRC

计算方法、校验方法

2.多路访问链路协议

信道划分协议

时分多路复用（TDM）

频分多路复用（FDM）

（参见第一章）

随机接入协议

时隙ALOHA&ALOHA（了解）

载波监听多路访问CSMA

CSMA/CD

载波监听就是利用电子技术检测总线上有没有其他计算机也在发送。**载波监听实际上就是检测信道。**在发送前，每个站不停地检测信道，是为了获得发送权；在发送中检测信道，是为了及时发现有没有其他站的发送和本站发送的碰撞，这就是碰撞检测。总之，载波监听是全程都在进行的。

碰撞检测就是**边发送边监听。**就是网卡边发送数据边检测新岛上的信号电压的变化情况，以便判断自己在发送数据的时候其他站是否也在发送数据。当几个站同时在总线上发送数据时，总线上的信号电压变化幅度将会**增大**（互相叠加），当网卡检测到的信号电压超过一定的门限值时，说明总线上至少有两个站同时在发送数据，表明产生了碰撞（冲突），所以也称为冲突检测。这时，由于接收的信号已经识别不出来，所以**任何一个正在发送的站就会立即停止发送数据，然后等待一段随机事件以后再次发送。**

（传播时延对碰撞检测的影响）

二进制指数后退法

以太网使用**截断二进制指数退避算法**来确定碰撞后重传的时机。这种算法让发生碰撞的站在停止发送数据后，不是等待信道变为空闲后立即再发送数据，而是退避一个随机的时间。一以减小再次发生碰撞的概率。具体算法如下：

- 协议规定了基本退避时间为2t，具体的争用期时间通常取51.2us,对于10Mb/s的以太网，在争用期内可发送512bit，即64字节。
- 从离散的整数集合【0，1，……（2^k-1）】中随机取一个数，记为r，重传应推后的时间就是r倍的争用期。
- 当重传次数不超过10时，参数k等于重传次数；当重传次数超过10时，k就不再增大一直等于10。
- 当重传16次仍然不能成功时（这表明同时打算发送数据的站太多，以致连续发生冲突），则丢弃该帧，并向高层报告。

轮流协议

（了解）

轮询协议

令牌传递协议

3.ARP

MAC地址：网卡适配器具有的物理地址，唯一。

ARP：地址解析协议，将网络层IP地址与链路层MAC地址进行转换。

每台主机或路由器内存中有一个ARP表，包含IP地址到MAC地址映射关系。

IP 地址	MAC 地址	TTL
222. 222. 222. 221	88-B2-2F-54-1A-0F	13:45:00
222. 222. 222. 223	5C-66-AB-90-75-B1	13:52:00

图 6-18　在主机 222. 222. 222. 220 中的一个可能的 ARP 表

TTL，指示删除每个映射的时间。

发送数据到子网外

很明显，子网先通过ARP找到网关对应接口MAC地址，通过网关发送到子网外。

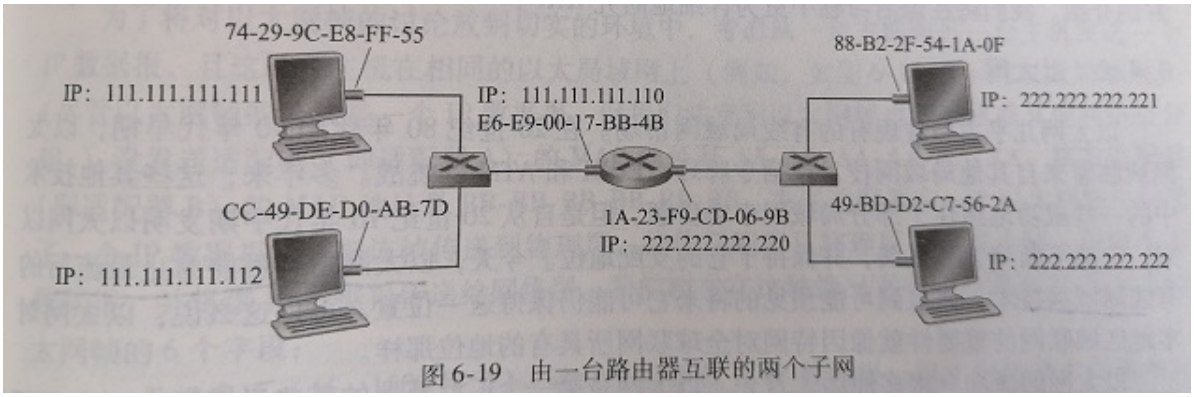
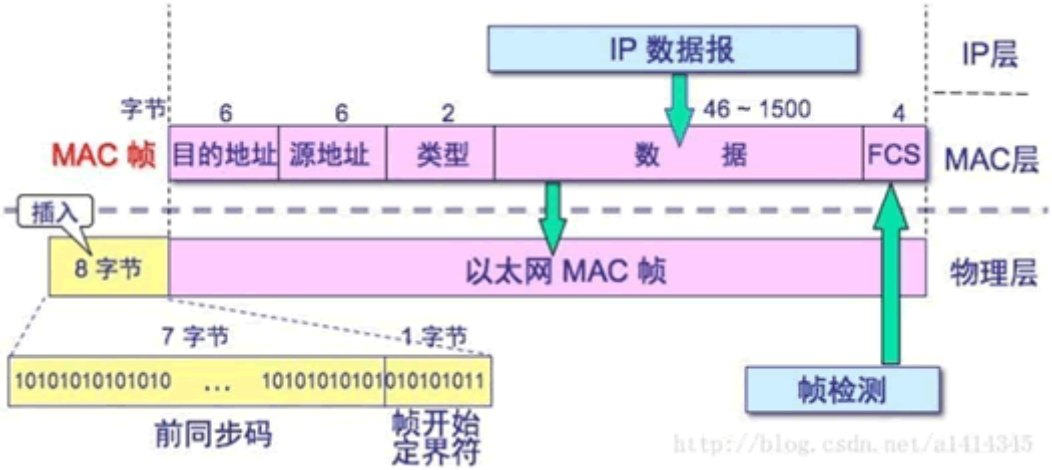


图 6-19　由一台路由器互联的两个子网

4.以太网

以太网是目前最流行的有线局域网技术。

以太网帧结构



字段	含义
前同步码	用来使接收端的适配器在接收 MAC 帧时能够迅速调整时钟频率，使它和发送端的频率相同。前同步码为 7 个字节，1 和 0 交替。
帧开始定界符	帧的起始符，为 1 个字节。前 6 位 1 和 0 交替，最后的两个连续的 1 表示告诉接收端适配器：“帧信息要来了，准备接收”。
目的地址	接收帧的网络适配器的物理地址（**MAC 地址**），为 6 个字节（48 比特）。作用是当网卡接收到一个数据帧时，首先会检查该帧的目的地址，是否与当前适配器的物理地址相同，如果相同，就会进一步处理；如果不同，则直接丢弃。
源地址	发送帧的网络适配器的物理地址（**MAC 地址**），为 6 个字节（48 比特）。
类型	上层协议的类型。由于上层协议众多，所以在处理数据的时候必须设置该字段，**标识数据交付哪个协议处理**。例如，字段为 0x0800 时，表示将数据交付给 IP 协议。
数据	也称为效载荷，表示交付给上层的数据。以太网帧数据长度**最小为 46 字节，最大为 1500 字节**。如果不足 46 字节时，会填充到最小长度。最大值也叫最大传输单元（MTU）。 在 Linux 中，使用 ifconfig 命令可以查看该值，通常为 1500。
帧检验序列 FCS	检测该帧是否出现差错，占 4 个字节（32 比特）。发送方计算帧的**循环冗余码校验（CRC）**值，把这个值写到帧里。接收方计算机重新计算 CRC，与 FCS 字段的值进行比较。如果两个值不相同，则表示传输过程中发生了数据丢失或改变。这时，就需要重新传输这一帧。

以太网向网络层提供不可靠服务，当某帧没有通过CRC校验即丢弃。

5.交换机

过滤：决定一个帧应该转发到某个接口还是丢弃；

转发：决定一个帧应该被导向哪个接口并移动该帧到接口。

交换机过滤与转发借助于**交换机表**。

自学习功能，即插即用设备。

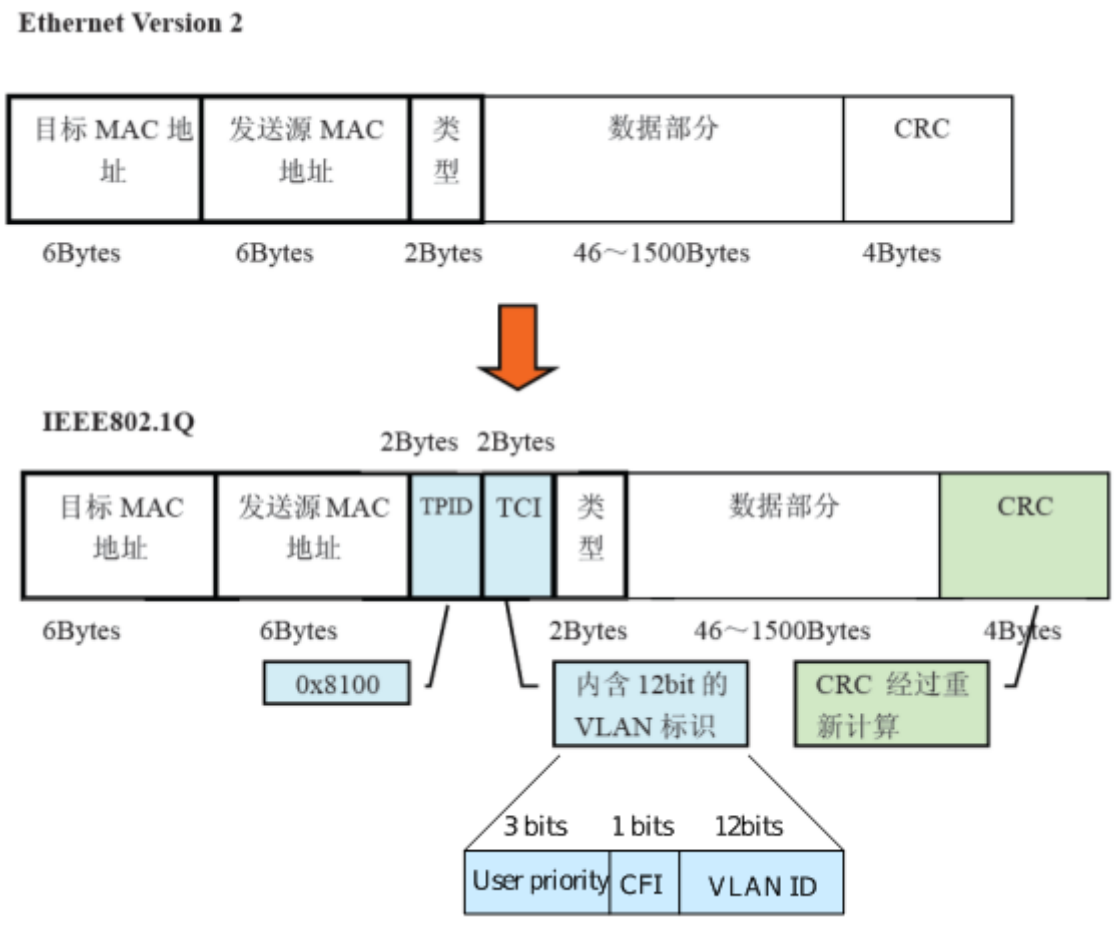
地址	接口	时间
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	3	9:36
…	…	…

图 6-22　图 6-15 中最上面交换机的交换机表的一部分

6.VLAN

VLAN(Virtual LAN)，翻译成中文是“虚拟局域网”。LAN可以是由少数几台家用计算机构成的网络，也可以是数以百计的计算机构成的企业网络。VLAN所指的LAN特指**使用路由器分割的网络**——也就是广播域。

简单来说，同一个VLAN中的用户间通信就和在一个局域网内一样，**同一个VLAN中的广播只有VLAN中的 成员才能听到**，而不会传输到其他的VLAN中去，从而控制不必要的广播风暴的产生。同时，若没有路由，不同VLAN之间不能相互通信，从而提高了不同工作组之间的信息安全性。网络 管理员可以通过配置VLAN之间的路由来全面管理网络内部不同工作组之间的信息互访。



TPID (Tag Protocol Identifier，也就是EtherType) **标签协议标识符**
是IEEE定义的新的类型，表明这是一个加了802.1Q标签的帧。TPID包含了一个固定的值0x8100。

TCI (Tag Control Information) **标签控制信息**
包括用户优先级(User Priority)、规范格式指示器(Canonical Format Indicator)和 VLAN ID。
①User Priority：该字段为3-bit，用于定义用户优先级，总共有8个(2的3次方)优先级别。IEEE 802.1P为3比特的用户优先级位定义了操作。最高优先级为7，应用于关键性网络流量，如路由选择信息协议（RIP）和开放最短路径优先（OSPF）协议的路由表更新。优先级6和5主要用于延迟敏感（delay-sensitive）应用程序，如交互式视频和语音。优先级4到1主要用于受控负载（controlled-load）应用程序，如流式多媒体（streaming multimedia）和关键性业务流量（business-critical traffic） - 例如，

SAP 数据 - 以及“loss eligible”流量。优先级0是缺省值，并在没有设置其它优先级值的情况下自动启用。
②CFI：CFI值为0说明是规范格式，1为非规范格式。它被用在令牌环/源路由FDDI介质访问方法中来指示封装帧中所带地址的比特次序信息。
③VID：该字段为12-bit， VLAN ID 是对 VLAN 的识别字段，在标准 802.1Q 中常被使用。支持4096(2的12次方) VLAN 的识别。在4096可能的VID 中，VID = 0 用于识别帧优先级。 4095(FFF)作为预留值，所以 VLAN 配置的最大可能值为4094。 所以有效的VLAN ID范围一般为1-4094。

第6章 无线网络

1.体系结构

802.11体系结构的组成
802.11体系结构的组成包括：**无线站点STA**（station），**无线接入点AP**（access point），独立基本服务组IBSS（independent basic service set），**基本服务组BSS**（basic service set），分布式系统DS（distribution system）和扩展服务组ESS（extended service set）。

一个无线站点STA通常由一台PC机或笔记本计算机加上一块无线网卡构成，无线网卡分为台式机用的PCI或ISA插槽的网卡和笔记本电脑用的PCMCIA网卡，此外无线的终端还可以是非计算机终端上的能提供无线连接的嵌入式设备(例如802.11手机)。

无线接入点AP可以看成是一个无线的Hub，它的作用是提供STA和现有骨干网络（有线或无线的）之间的桥接。AP可以接入有线局域网，也可以不接入有线局域网，但在多数时候AP与有线网络相连，以便能为无线用户提供对有线网络的访问。AP通常由一个无线输出口和一个以太网接口(802.3接口)构成，桥接软件符合802.1d桥接协议。

802.11在网络构成上采用单元结构，将整个系统分成许多单元，每个单元称为一个BSS（基本服务组），多个BSS构成一个ESS（扩展服务组），不含AP的BSS称为IBSS（独立基本服务组）。

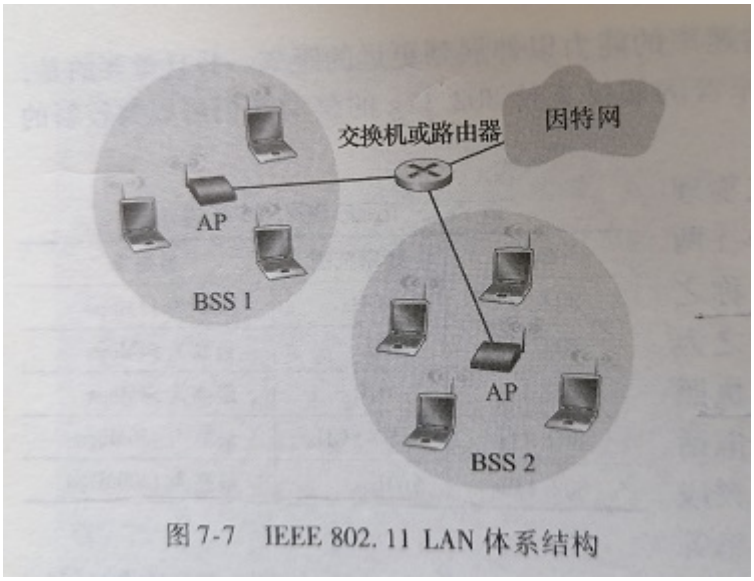


图 7-7 IEEE 802.11 LAN 体系结构

主动扫描与被动扫描

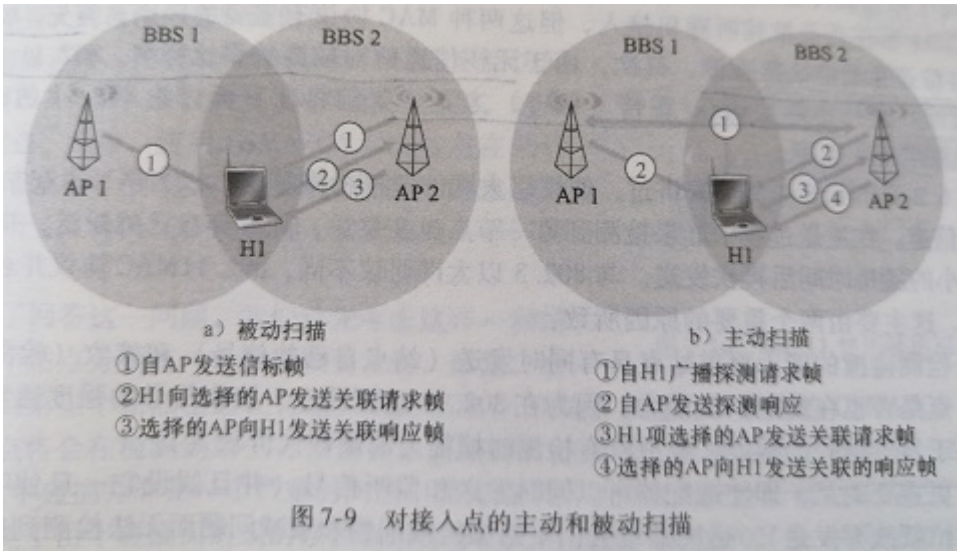


图 7-9　对接入点的主动和被动扫描

2.MAC

CSMA/CA： 带**碰撞避免**的CSMA

由于无线信道具有相对较高的误比特率，所以使用链路层确认重传方案。

DCF Interframe Space(DIFS):在DCF协议中，**节点在开始发送数据之前需要监测信道是否空闲。**如果信道已经空闲，则节点仍需等待DIFS段时间才开始发送数据；而如果在DIFS时间段内任一时刻信道被监测为忙，则节点不得不推迟它的数据发送。DIFS和SIFS间的计算关系如下：

SIFS Short Interframe Space(SIFS):在802.11系列无线局域网中SIFS是固定值，SIFS是**最小的帧间间隔**，因此采用SIFS的节点具有访问无线链路的最高优先级。它等于节点从发送状态切换到接收状态并能正确解码所需要的时间，或者从接收状态转为发送状态所需要的时间，在SIFS过期后可能发送的数据包包括ACK、CTS帧，不同标准中规定的SIFS值不同。

工作原理

所有无线站点在发送数据前先进行载波侦听操作，以查看通信线路是否空闲，如果其他站点正在传输数据，那么该侦听站点就会随机等待一段时间（再等待大于DIFS的空闲时间后，启动退避机制），然后再进行重新发送。如果侦听到介质空闲（这里的空闲是指侦听到大于DIFS的空闲时间后），那么它就会先发送一个比较短的请求发送信息（RTS），RTS 消息是由目的地址和需要占用的传输时间组成的，这样一来，其他的站点就会知道必须等待多长的时间才能够发送。目的端收到RTS后，就会发出一个允许发送的短消息（CTS）（等待SIFS的时间），告诉源站点可以发送数据而不必担心会发生冲突。然后源站点开始发送数据（等待SIFS时间后发送数据）。数据到达目的端后，目的端就会发送一条ACK信号（等待SIFS时间），表示数据已经接收到了。如果没有收到ACK的话，MAC层就会重传那段数据（重传数据时，会重新启动退避机制）。

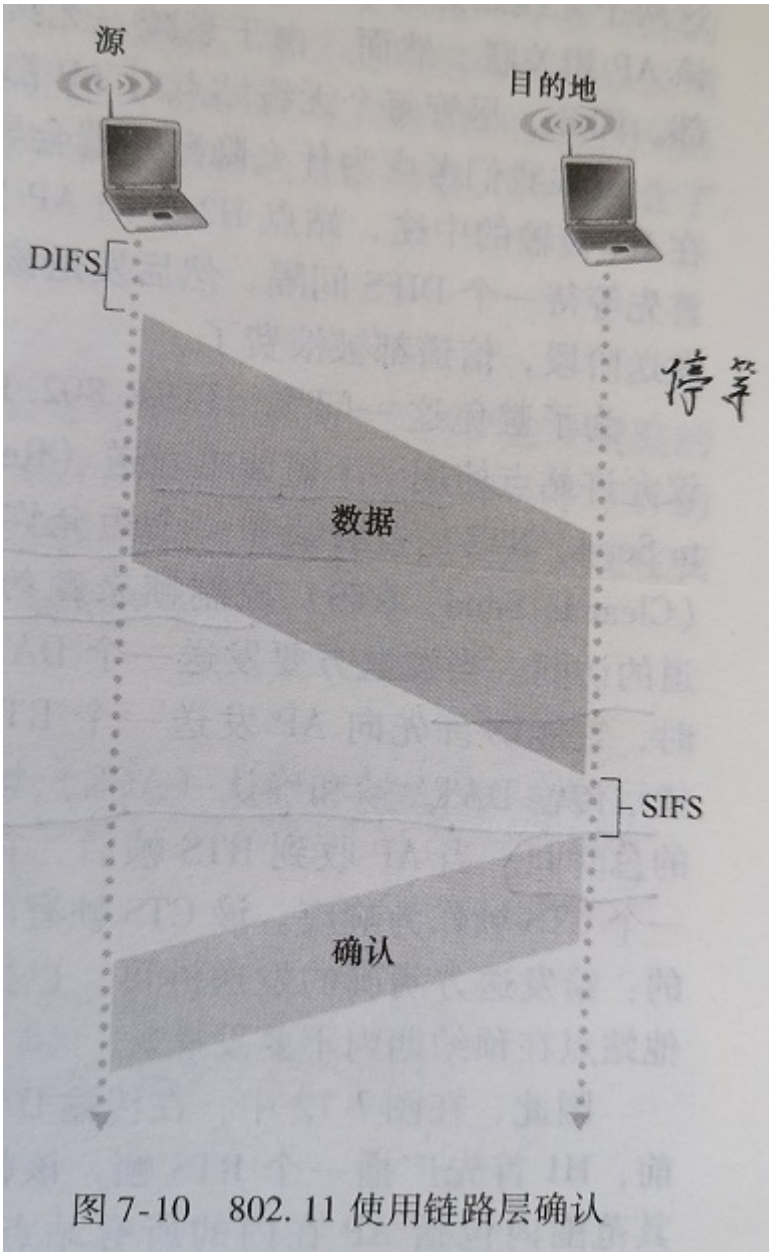


图 7-10　802.11 使用链路层确认

处理隐藏终端

首先等待一个 DIFS 间隔，然后
发送阶段，信道都被浪费了。

为了避免这一问题，IEEE 802.11 协议允许站点使用一个短请求发送（Request to Send，RTS）控制帧和一个短允许发送（Clear to Send，CTS）控制帧来预约对信道的访问。当发送方要发送一个 DATA 帧时，它能够首先向 AP 发送一个 RTS 帧，指示传输 DATA 帧和确认（ACK）帧需要的总时间。当 AP 收到 RTS 帧后，它广播一个 CTS 帧作为响应。该 CTS 帧有两个目的：给发送方明确的发送许可，也指示其他站点在预约期内不要发送。

因此，在图 7-12 中，在传输 DATA 帧前，H1 首先广播一个 RTS 帧，该帧能被其范围内包括 AP 在内的所有站点听到。AP 然后用一个 CTS 帧响应，该帧也被其范围内包括 H1 和 H2 在内的所有站点听到。站点 H2 听到 CTS 后，在 CTS 帧中指明的时间内将抑制发送。RTS、CTS、DATA 和 ACK 帧如图 7-12 所示。

RTS 和 CTS 帧的使用能够在两个重要方面提高性能：

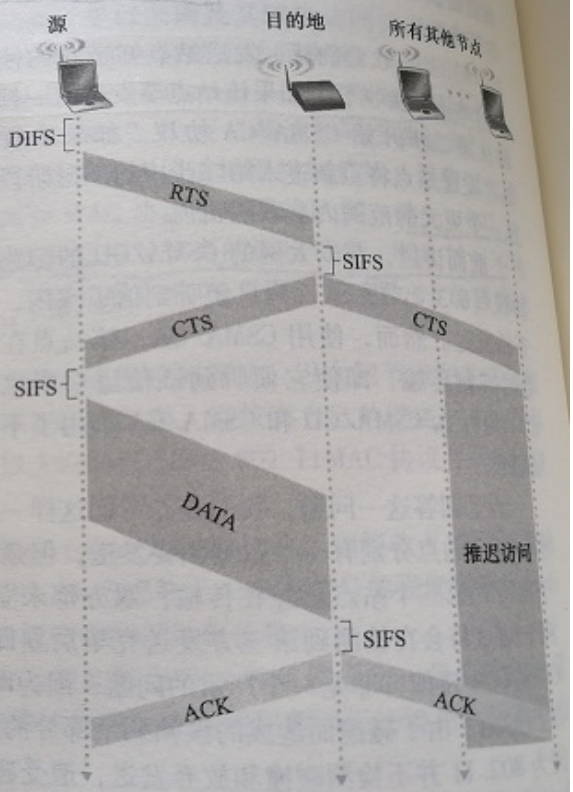


图 7-12　使用 RTS 和 CTS 帧的碰撞避免

第7章 计算机安全

1.密码学

对称密钥加密

流密码（了解）

流密码算法，或者叫序列密码，算法大概的原理是，每次加密都通过密钥生成一个**密钥流**，解密也是使用同一个密钥流，明文与同样长度的密钥流进行异或运算得到密文，密文与同样的密钥流进行异或运算得到明文。

块密码

块密码算法也叫分组密码算法，从字面意思就可以知道，它把**加密和解密序列分成了一个个分组，最后把每一块序列合并到一起**，形成明文或者密文。根据不同的分组加密方式，每个分组之间可以有联系，也可以没有联系

DES、3DES、AES

特点：加密算法较简单，计算量不高，但关键在于密钥怎么传输?

公开密钥加密

公钥和私钥（K+，k-）

公钥加密只有私钥可解，私钥加密使用公钥解。

RSA

特点：加密算法计算量大，无法加密大文件。

2.报文完整性与数字签名

报文完整性

密码散列函数，以m为输入，计算得到一个固定字符串H(m)，使得找不到任意两个不同报文x和y使得H(x)=H(y);

（使得入侵者不能伪造报文内容）

缺点：入侵者可以使用自己报文和自己散列函数。

报文鉴别码：

需要一个共享比特串s（鉴别密钥）

计算H(m+s)，即为报文鉴别码

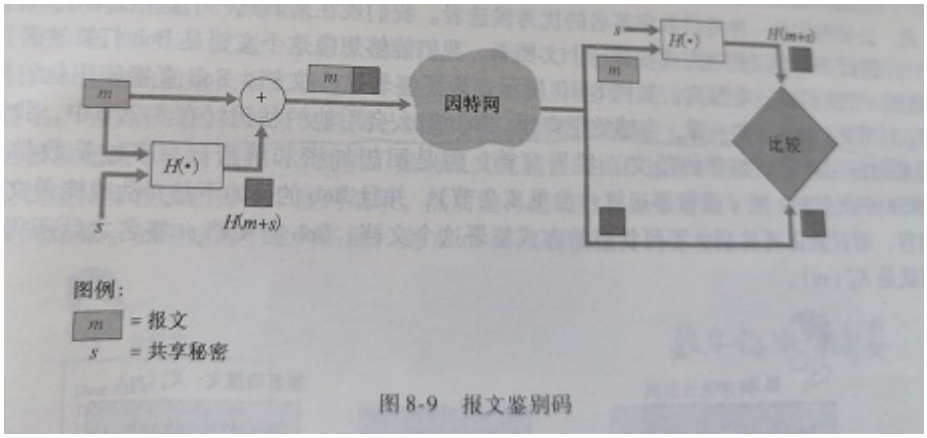


图 8-9　报文鉴别码

简而言之，即对**报文+H(m)**进行了**对称加密。**

数字签名

通俗来讲，对一段报文用自己私钥加密即为签名。

对计算的散列报文进行签名（私钥），K-(H(m))

验证时，将明文求Hash，再与解密后的签名比较。

公钥认证

认证中心，CA：验证一个实体与其公钥绑定，并颁发证书。

端点鉴别

一个实体经计算机网络向另一个实体证明身份的过程；

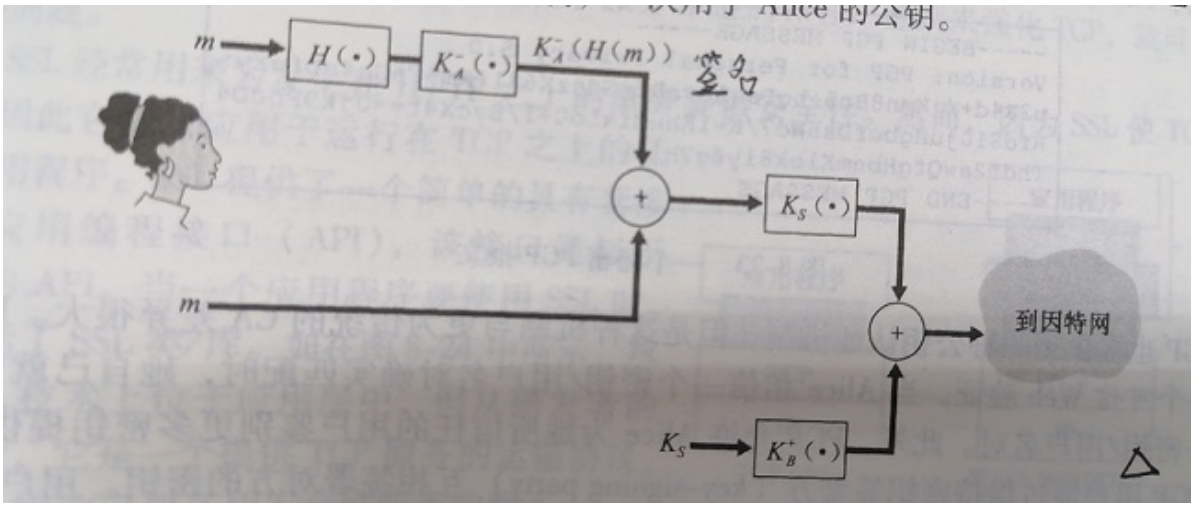
3.安全电子邮件示例

使用散列函数(Hash)对明文进行加密 H(m)

将Hash后生成的摘要用私钥加密 K-(H(m))

将明文与加密后摘要组合，再使用对称密钥加密 Ks(K-H(m) + m)

将对称密钥使用公钥加密并与前面组合，发送给对方。



一些要考的解答题：

必考：

1.时延计算

节点与节点时延：处理时延+排队时延+传输时延+传播时延

端到端时延：假定源到目的有N-1台路由器，且无拥塞（无排队时延）

=N*节点间时延

2.DNS工作过程与如何提高效率

工作过程：

输入域名后，操作系统会先查询本地host文件是否有映射关系

再在查询本地DNS缓存

再询问本地DNS服务器

再查询根域名服务器，会返回该URL顶级域名服务器IP地址

再查询顶级域名服务器，如果没有则查询它下辖服务器，直至找到返回。

提高效率：

1.树状结构，分级查找

2.域名服务器多级缓存

3.目前IP网络层给上层（包含UDP）提供的是尽最大努力数据传输服务服务，UDP给上层应用层提供的也是尽最大努力服务，能否将两层合并，不设计UDP呢？为什么？

不能。

1.协议本身来看：UDP为传输层协议，主要用于将应用层交付的数据打包后交给网络层，最重要的一点是需要识别是哪个用户，即端口号。而IP协议作用于网络层，用于将传输层打包好的数据报传到目的IP地址的主机（传输主要是链路层），IP协议不涉及到是哪个程序交付的，不设计UDP只能将数据报传到指定位置，而无法交付给对应程序。

2.应用层来看：有些应用层数据需要追求效率，不使用TCP传输，如果没有UDP直接交给网络层，数据将无法被目的主机识别交予哪个进程；应用层数据只有交付给传输层的接口，而无法交付给网络层。

3.网络层来看：网络层只处理传输层交付的数据，只关心如何传到目的主机，而不关心给主机哪一个具体进程，如果合并，破坏了分层结构，网络层协议也会复杂很多，从而降低网络层效率。

4.TCP序列号、确认号计算填空

序列号：是传输报文段首字节的字节流编号

确认号：是期望收到的下一个字节的序号，即seq+传输数据字节数

（详情参见TCP）

5.TCP往返时间计算

需要一个EstimatedRTT（估计时间）、一个DevRTT（偏差）

EstimatedRTT = 0.875*EstimatedRTT* + *0.125*SampleRTT

DevRTT = 0.75*DevRTT* + *0.25*|SampleRTT - EstimatedRTT|

TimeoutInterval = EstimatedRTT + 4*DevRTT

（详情参见书上P158）

6.传统路由器选路转发与SDN差异

传统：每台路由器均运行路由选择算法，每台路由器包含转发与路由选择两种功能。

SDN：路由器从物理上分离，远程控制器计算和分发转发表以供每台路由器使用。

7.路由器转发数据依据

路由表

8.DHCP工作原理

（详情参加DHCP）

9.Dijkstra填表

书上P246

10.BGP与自治系统编号

参加书上P258-P260

11.CRC计算

不多说了

12.设计数据传输加密过程

（参考安全邮件示例）

可能会考：

1.IP首部校验和为什么只计算首部，而不是整个IP数据报，为什么每个路由器在转发一个数据报前都要重新计算该校验和。

IP协议位于网络层，它的数据封转格式是在传输层数据基础之上加了一个IP头。因此，IP数据包报的数据部分其实是传输层数据报（TCP或UDP），而TCP或UDP在封转数据时已经有校验字段对它本身携带的数据进行校验。因此，如果再在IP头部添加对数据区域的校验字段的话，一是会重复校验，二是会增加数据包的非数据部分长度，降低了协议的效率。

因为IP数据报每经过一个路由器，他的TTL字段值就会减1，所以需要重新计算校验和

2.VLAN设计在哪些方面体现出它的应用优势，是否存在某些方面的应用缺点?

优点：

1.限制广播域。广播域被限制在一个VLAN内，提高了网络处理能力。

2.增强局域网的安全性。VLAN的优势在于VLAN内部的广播和单播流量不会被转发到其它VLAN中，从而有助于控制网络流量、减少设备投资、简化网络管理、提高网络安全性。

3.灵活构建虚拟工作组。用VLAN可以划分不同的用户到不同的工作组，同一工作组的用户也不必局限于某一固定的物理范围，网络构建和维护更方便灵活。

缺点：

VLAN常见有三种配置，基于端口、静态与动态

1.基于端口的VLAN，当用户位置改变时，相应地要对网线进行迁移；

2.静态VLAN因为端口和VLAN的不一致往往会直接导致一个VLAN的人员不能正常访问他原先所在的VLAN之中。

3.动态VLAN在建立初期工作量较大。

3.如果我们自己编写的网络程序运行时，获取的键盘输入是域名，怎么编写程序让程序得到解析出的IP地址?

调用操作系统的Socket库；

Socket中gethosstbyname()，生成发送给DNS服务器的查询消息，接收DNS服务器的返回消息，从响应中取出IP地址返回给应用层。