**计算机网络基础复习资料**

1. 因特网主机=端系统，包含PC、智能手机、PAD、智能电视等。
2. 网络核心部分要向网络边缘中的大量主机提供连通性，使边缘部分中的任何一个主机都能够向其他主机通信（即传送或接收各种形式的数据）。在网络核心部分起特殊作用的是路由器(router)。路由器是实现分组交换(packet switching)的关键构件，其任务是转发收到的分组，这是网络核心部分最重要的功能。

分组交换主要有两类，一类叫做路由器，一类叫作链路层交换机。两者的作用类似，都是转发分组，不同点在于转发分组所依据的信息不同。路由器根据分组中的IP地址转发分组，链路层交换机根据分组中的目的MAC地址转发分组。

1. 一个协议定义了在两个或多个通信实体之间交换的报文格式和次序，以及报文发送和/或接收一条报文或其他事件所采取的动作。一个协议的关键元素：报文格式、报文次序、动作。

网络协议三要素（1）语法：即数据与控制信息的结构或格式；（2）语义：即需要发出何种控制信息，完成何种动作以及做出何种响应；（3）同步：即事件实现顺序的详细说明。

1. 因特网标准由因特网工程任务组IETF研发。IETF的标准文档称为请求评论RFC。
2. 复用技术是指能在同一传输媒质中同时传输多路信号的技术，目的提高通信线路的利用率。

频分复用（FDM）的所有用户在同样的时间占用不同的带宽资源。

时分复用（TDM）则是将时间划分为一段段等长的时分复用帧（TDM 帧）。每一个时分复用的用户在每一个TDM 帧中占用固定序号的时隙。利用不同的时隙传送不同的信号。

统计时分复用（STDM)在时分复用的基础上根据实际情况“按需分配”。

1. 交换(switching)就是按照某种方式动态地分配传输线路的资源。

用于网络核心的交换技术主要有两种：电路交换(circuit switching)，分组交换(packet switching)

（1）电路交换：在通信进行过程中，网络为数据传输在传输路径上预留资源，这些资源只能被这次通信双方所使用；

（2）分组交换：数据被分成一个一个的分组，每个分组均携带目的地址，网络并不为packet传输在沿途packet switches上预留资源，packet switches为每个packet独立确定转发方向.

与电路交换不同，链路、交换机/路由器等资源被多个用户所共享，交换机在转发一个分组时的速度为其输出链路的full速度。

注：分组交换一般采用存储转发技术，分组在分组交换机中会经历一个排队(queuing)延迟。排队延迟与交换机的忙闲有关，大小可变。如果分组到达时缓存已满，则交换机会丢掉一个分组。分组交换网络有两大类：1）Datagram（数据报)网络，2）Virtual Circuit虚电路网络

1. 通讯介质及特点

导引型传输媒体：双绞线、同轴电缆、光纤

非导引型传输媒体：无线电通讯

1）双绞线（Twisted-Pair Copper Wire）：抗电磁干扰，模拟传输和数字传输都可以用；双绞线是目前高速LAN联网的主要方式。

2）同轴电缆（Coaxial Cable）广泛用于闭路电视中，容易安装、造价较低、网络抗干扰能力强、网络维护和扩展比较困难、电缆系统的断点较多，影响网络系统的可靠性。

3）光纤（Fiber Optics）传输损耗小，抗雷电和电磁干扰性好，保密性好，体积小，质量轻。

单模光纤比多模光纤性能更优，但价格更高。

4）无线电通讯（Radio）用无线电传输，优点：通讯信道容量大，微波传输质量高可靠性高，与电缆载波相比，投资少见效快。缺点：在传播中受反射、阻挡、干涉的影响。

1. 延时分类

1）传输时延（发送时延）

发送数据时，数据块从结点进入到传输媒体所需要的时间。也就是从发送数据帧的第一个比特算起，到该帧的最后一个比特发送完毕所需的时间。

2）传播时延

电磁波在信道中需要传播一定的距离而花费的时间。信号传输速率（即发送速率）和信号在信道上的传播速率是完全不同的概念。

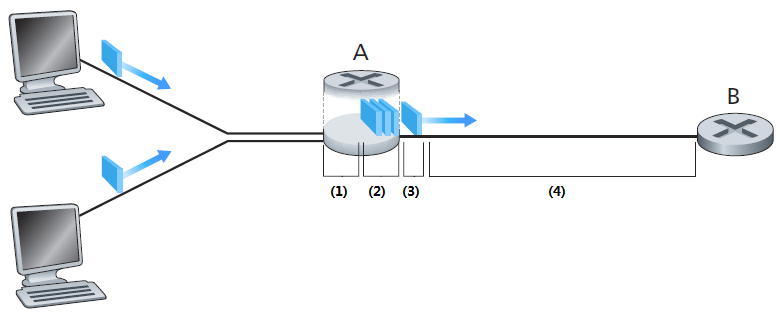
3）处理时延：交换结点为存储转发而进行一些必要的处理所花费的时间。

4）排队时延：结点缓存队列中分组排队所经历的时延。排队时延的长短往往取决于网络中当时的通信量。

注：排队延迟是节点延迟中最复杂、也是最有趣的部分。之所以最有趣，指目前或多研究工作就是针对排队延迟来进行的，包括调度算法、缓存策略等。排队延迟与网络设备的负载状况密切相关，不同分组所经历的排队延迟会随着负载的变化而变化

关于发送延迟和传播延迟，容易弄混。需要记住，传输延迟指将一个分组所有bit发送到link上所需的时间，与分组长度和发送速率有关，与两点之间的距离没有任何关系。而传播延迟指一位从链路的一端传播到另一端所需的时间，与link的长度和信号的传播速度有关。

1. 分组从源主机出发，通过一系列路由器结点传输，到达目的主机。分组在下图中路由器结点内（之间）需经历的 (1) ~ (4) 不同类型的时延。



(1)处理时延：路由器检查分组首部并决定将该分组导向何处所需的时间；

(2)排队时延：当路由器队列中分组较多，分组在链路上等待传输所需经历的排队时间；

(3)传输时延：路由器结点将分组的比特推向（传输到）链路所需的时间；

(4)传播时延：比特从路由器结点A传播到路由器结点B所需要的时间。

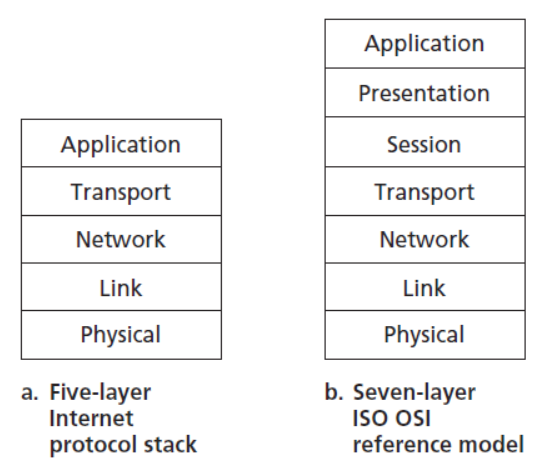
1. TCP/IP的体系结构

1）层次、功能、层次之间的关系

2）每层数据包的名称

3）每层地址

4）接口、协议、服务



因特网协议体系结构的几个层次及其作用。

（1）应用层：基于应用层协议提供网络应用功能，应用层协议如HTTP、SMTP、FTP等。应用层的信息分组称为报文（message）。

（2）运输层：在应用程序端点之间传输应用层报文，实现端到端的逻辑通信。运输层分组称为报文段（segment）。

（3）网络层：将数据报分组从一台主机移动到另一台主机，即提供主机之间的逻辑通信。网络层的分组称为数据报（datagram）。

（4）链路层：将网络层的数据报从一个节点传输到另一个节点。链路层分组称为帧（frame）。

（5）物理层：实现数据的比特流传输，提供信号传输介质。物理层的数据称为比特。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **层级** | **层名** | **分组名称** | **功能** | **层次之间的关系** | **每层地址** |
| 5 | 应用层 | Message报文 | 支持网络应用 | 一层嵌到另一层（每一层次都从上层的导数据，加上首部信息形成新的数据单元，将新的数据单元传递给下一层） | 不同的应用有不同的地址 |
| 4 | 传输层 | Segment报文段 | 负责应用进程间的通讯 | 端口号 |
| 3 | 网络层 | Datagram数据报 | 从源到目的地数据报的路由 | IP地址 |
| 2 | 数据链路层 | Frames帧 | 相邻节点之帧转发 | 网卡地址 |
| 1 | 物理层 | 无数据包 | 比特转发 | 无 |

应用层的地址不止有IP地址还有端口号，传输层、网络层为IP地址，链路层、物理层的地址为MAC地址。

接口在两层之间，协议是同层之间的，服务是下层为上层提供的。

1. 一个分组在因特网中传输，在源主机需将数据M封装成自顶向下不同层的分组。如下图所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 应用层分组名称：报文 |  |  |  | H1 | M |
| 运输层分组名称：报文段 |  |  | H2 | H1 | M |
| 网络层分组名称：数据报 |  | H3 | H2 | H1 | M |
| 链路层分组名称：帧 | H4 | H3 | H2 | H1 | M |
| 物理层数据：比特流 | 0、1比特流 | | | | |

1. 应用程序体系结构：client/server、P2P、混合C/S和P2P

客户服务器方式所描述的是进程之间的服务和被服务的关系。

客户是服务的请求方，服务器是服务的提供方。

Client/Server的好处是系统管理容易，问题是Server容易成为系统的瓶颈。

P2P中，没有在C/S中处于中心地位的Server，所有Host的地位平等，叫做Peers，因此这种系统也叫Peer to Peer。

P2P中没有必须总是在线的服务器，并且peer可以随时更换自己的IP。Gnutella是Pure P2P的一个很好的例子。

P2P的最大好处是系统可扩展性（scalability)强。由于每个peer既是Server又是Client, 随着系统中Peer的数量增多，系统的处理能力越强。

P2P的问题是可管理性，由于系统是完全分散的、无中心的，管理起来极其困难。

混合C/S和P2P即以上两种方式的结合。

1. HTTP通讯超文本传输协议

HTTP主要规定了消息的结构和client和server交换message的方式。

1) 浏览器首先建立与Server的TCP连接

2) 连接建立起来后，browser和server就向/从Socket发送/接收HTTP的消息。借助TCP的可靠数据传输，HTTP知道消息肯定会到达对方，这就是协议分层的好处。

HTTP是一种stateless(无状态)协议，server不保存任何client的任何状态信息。如果server在很短的时间内从browser接收到对某个object的两次请求，server就会发送两次response。

3）非流水线方式：客户在收到前一个响应后才能发出下一个请求。这比非持续连接的两倍RTT 的开销节省了建立TCP 连接所需的一个RTT 时间。但服务器在发送完一个对象后，其TCP 连接就处于空闲状态，浪费了服务器资源。

流水线方式：客户在收到HTTP 的响应报文之前就能够接着发送新的请求报文。一个接一个的请求报文到达服务器后，服务器就可连续发回响应报文。使用流水线方式时，客户访问所有的对象只需花费一个RTT时间，使TCP 连接中的空闲时间减少，提高了下载文档效率。

3）非持续连接：建立一次TCP连接，browser和server通过此连接只传输一个request消息和一个respond消息；

持续连接：建立一次TCP连接，browser和server通过此连接可以传输多个request消息和多个respond消息

* HTTP协议支持非持续连接和持续连接、非流水线和流水线多种方式。其默认模式是流水线的持续连接。
* Web的应用层协议是HTTP，它是Web的核心；它定义了在浏览器和Web服务器之间的传输报文格式和序列。
* HTTP协议使用TCP进行传输的重要原因是考虑到Web网页作为文本数据的传输可靠性。

1. HTTP请求报文中的方法字段

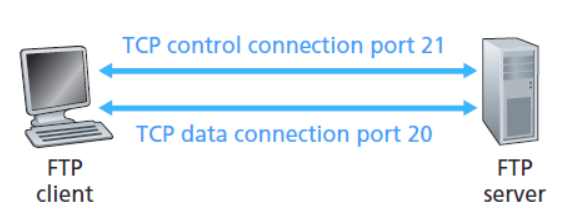
HTTP请求报文中的方法字段可以取种不同的值，包括GET、POST、HEAD、PUT和DELETE。当用户在浏览器中输入一个网址，希望获取网站的对象时，使用的是GET方法。当用户提交表单时，比如用户向搜索引擎提供搜索关键词时，HTTP客户常使用POST方法。当用户应浏览器要求而选中本地的图片进行上传时，HTTP客户使用的是PUT方法。当用户希望删除Web服务器中的图片时，可使用DELETE方法。

* 使用浏览器打开一个页面，在HTTP请求报文中通常使用的方法是GET。
* 用户在浏览器页面中提交表单时，在HTTP请求报文中使用的方法是POST。
* 当用户想要利用HTTP协议上传对象到Web服务器上时，在HTTP的请求报文中，所使用的方法是PUT。

1. HTTP和FTP

HTTP和FTP都是文件传输协议，它们都运行TCP协议。FTP使用控制连接来传输控制信息，使用数据连接来传输文件；而HTTP协议是在传输文件的同一个TCP连接中发送控制信息（请求、响应报文），故FTP是带外发送控制信息，HTTP是带内发送控制信息。

HTTP协议的默认传输端口号是80，而FTP用于TCP控制连接的端口号是21，用于TCP数据连接的端口号是20。



FTP的控制连接贯穿了整个用户会话期间，故该连接是持续连接，而在每个会话中的每一次文件传输都需要建立一个新的数据连接，故数据连接是非持续的。

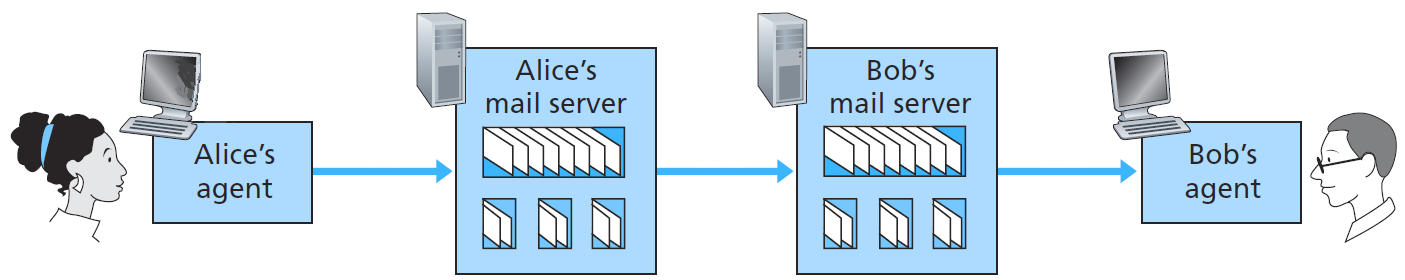
FTP服务器必须在整个会话期间保留用户的信息（将用户账户和控制连接联系起来，跟踪用户在文件服务器目录树上的位置等），故FTP又是有状态的。

而HTTP客户和服务器建立了连接后，客户可以一次性地获取多个对象，并且服务器无需跟踪客户的信息，故HTTP协议的连接是无状态的持续连接。

1. 电子邮件协议

Alice发送邮件给Bob，涉及多种电子邮件协议及其通信实体。

可以实现邮件收取的协议是POP3、IMAP、HTTP。



可使用HTTP和SMTP协议

可使用HTTP、POP3和IMAP协议

可使用SMTP协议

1. DNS的作用以及两种查询方式

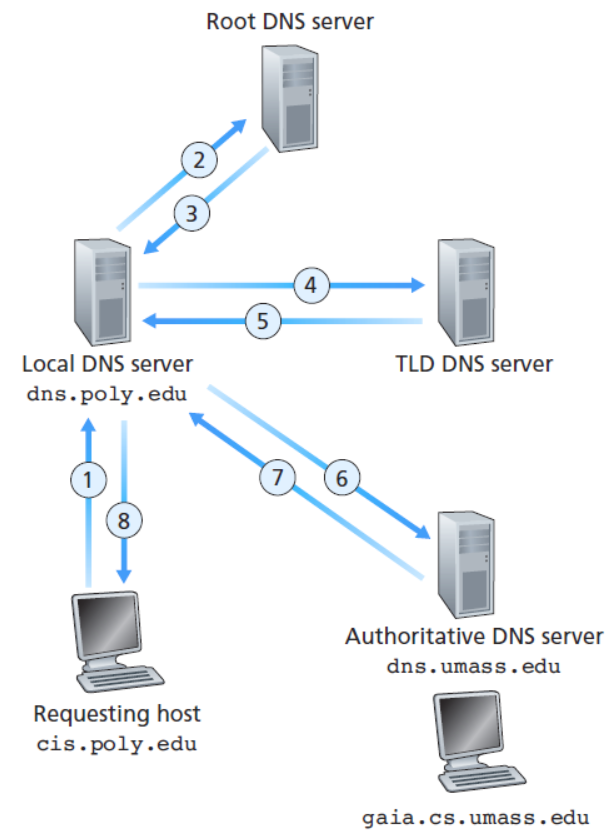
DNS 是域名解析系统(Domain Name System) 的缩写，它是由解析器和域名服务器组成的。用于便于人们使用的机器名字转换为IP地址

两种查询方式：

1）主机向本地域名服务器的查询一般都是采用递归查询。如果主机所询问的本地域名服务器不知道被查询域名的IP 地址，那么本地域名服务器就以DNS 客户的身份，向其他根域名服务器继续发出查询请求报文。

2）本地域名服务器向根域名服务器的查询通常是采用迭代查询。当根域名服务器收到本地域名服务器的迭代查询请求报文时，要么给出所要查询的IP 地址，要么告诉本地域名服务器：“你下一步应当向哪一个域名服务器进行查询”。然后让本地域名服务器进行后续的查询。

DNS迭代流程（1）~（8）。



* 某公司c有一台主机h，该主机具有的Internet域名应该为h.c.com。
* DNS使用UDP协议的主要原因是因为UDP协议在传输数据前无需像TCP那样经历三次握手，数据收发延时较小。
* 一台主机希望解析域名www.bit.edu.cn，如果这台服务器配置的域名服务器为202.120.66.88，Internet根域名服务器为10.1.2.3而存储www.bit.edu.cn与其IP地址对应关系的域名服务器为202.110.6.8，那么这台主机解析该域名时首先查询地址为202.120.66.88的域名服务器。

1. 运输层基本概念

* 多路分解将运输层报文段中的数据交付到应用层正确的套接字的工作称为多路分解。
* 计算机网络最本质的活动是分布在不同地理位置的主机之间的进程通信。
* 在因特网中，用来标识主机和在主机上的应用程序的是IP地址和端口号。
* 运输层协议除了提供可靠数据传输服务外，还可提供吞吐量确保、定时保证和安全性服务。
* TCP服务器进程与客户机进程通信时，需要生成欢迎套接字和连接套接字。
* TCP在传输数据前要进行三次握手，而UDP不需要任何准备即可进行数据传输。因此UDP不会引入建立连接的时延。

1. UDP的服务特点

UDP是一种无连接的、轻量级传输层协议，提供了最最健的服务模型。没有连接，直观上就应该比TCP更高效。

1）不可靠的数据传输：发送端将数据推入UDP Socket后，UDP并不保证数据最终会到达接收端，即使到达也不保证是按序到达；

2、没有拥塞控制机制：发送方可以以任意的速率向网络中发送数据，不管网络的拥塞状况。但发送的数据可能最终到达不了接收方，产生丢包。

优点：

1）应用可更好控制何时发送何种数据：无须建立连接，UDP可尽快将消息发给网络层；TCP可能需要重传在规定时间内没有到达的Segment。UDP没有建立连接所引入的延迟，这可能是DNS选择UDP而不是TCP的最主要原因。

2）实现简单：UDP因为是无连接的，主机因而无须维护连接状态，实现简单；

3）头部开销小：UDP的Segment头部字段共8个字节；而TCP的头部共包括20个字节.

1. UDP报文的首部为：

|  |  |
| --- | --- |
| 源端口号（0632） | 目的端口号（0045） |
| UDP报文段长度（001C） | 校验和（E217） |
| 数据 | |

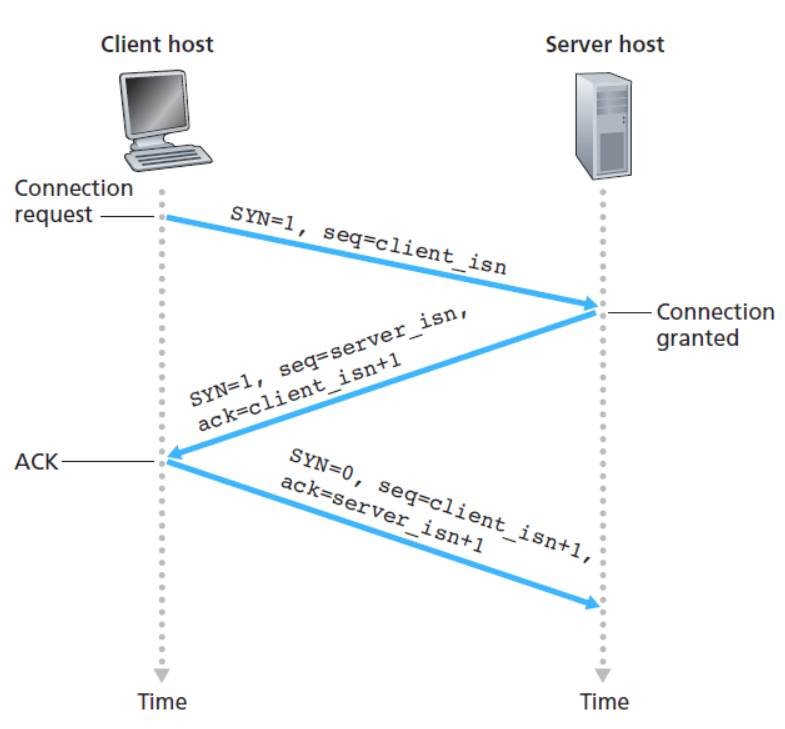
（1）根据UDP报文的格式，该UDP报文的源端口号为：0x0632 = 1586，目的端口号为：0x0045 = 69。因为目的端口 69 < 1023，所以目的端口是熟知端口，故该UDP报文是从客户发送给服务器（服务器程序是TFTP）。

（2）UDP报文段的长度为：0x001C = 28 字节，而UDP首都长度为 8 字节，故该UDP报文所包含的数据长度为 28 – 8 = 20 字节。

1. TCP的服务特点、流的概念

TCP (Transmission Control Protocol,传输控制协议) 是一种面向连接的协议，即数据传输之前要经过三次握手建立一条全双工连接，然后才能进行真正的数据传输。

TCP三次握手流程，注意seq、ack序号变化。



TCP除了是一种面向连接的协议外，还提供可靠的、按需到达的字节流数据传输、流控和拥塞控制。无头无尾，连续不断。

面向字节流。（TCP不采用停等式的传输，而用流水线的方式，且序列号是根据数据段的第一个字节填写的）

1. 可靠性传输原理

可靠性传输原理是由rdt1.0 rdt2.0 rdt2.1 rdt2.2 rdt3.0一步步累加而来的。

rdt1.0：接收方无返回确认信息

rdt2.0：接收方进行检错，并发送ACK 或NAK反馈给发送方

rdt2.1：加入序列号0和1rdt2.2：接收方不再发NAK而将ACK中加入序列号

rdt3.0：发送方引入定时器

以上都是停-等式（stop-and-wait）协议。为了解决stop-and-wait协议低效问题的方法非常简单，就是允许发送方可以在等待Receiver的ACK之前连续发送多个分组。这种技术叫做流水线。

流水线技术对可靠数据传输协议的影响：

1）更大的序列号范围。连续发送的并且是还没有得到ACK的多个分组必须要有唯一的序列号，否则引起混乱。

2）Sender和Receiver方需要存储空间来缓存分组。对于Sender来说，需要缓存已经发送出去但还没有得到ACK的分组；为了实现按序递交，接收方一般也需要存储空间。

序列号的范围和Buffer的大小取决于传输层协议如何相应分组丢失、差错以及过度延迟分组的方式。

解决流水线的差错恢复有两种基本方法：回退N步(Go-Back-N)和选择性重传(Selective Repeat)

GBN（Go-Back-N）允许发送方发送N个分组而无需确认，流水线中最多有N个等待确认消息的分组，允许使用的序列号范围可以看作是长度为N的一个窗口。随着协议的运行，这个窗口在序列号空间内向前滑动，因此这种协议也叫滑动窗口协议（sliding-window protocol)在此系统中，一个分组或其ACK的丢失可能造成GBN重传太多的分组。当信道差错率逐渐变大，信道会被不必要的重传分组所塞满。

SR（Selective Repeat）选择性重传就是Sender只重传那些出现错误的分组，而不是窗口中的所有分组。

* TCP报文段结构中有一个32比特的序号字段，该序号字段是按照字节流中的字节进行计数，而不是按照分组计数。
* TCP和UDP协议使用了16位来表示端口号，其中最常用的应用程序的端口号称为熟知端口，其数值范围是0-1023。
* TCP/UDP报文中源端口地址和目标端口地址可以相同，因为虽然端口地址一样，但其所在的主机是不同的。
* TCP报文中，若序号从1开始，确认号为1000表示已收到999字节。
* 主机甲与主机乙之间已建立一个TCP连接，主机甲向主机乙发送了两个连续的TCP段，分别包含300字节和500字节的有效载荷，第一个段的序列号为200，主机乙正确接收两个段后，发送给主机甲的确认序列号是1000。
* 当一个应用程序通知TCP数据已传送完毕时，TCP将单方面地关闭这个程序，此时报文段码位字段的FIN位被置1，指示发方已发送完数据。
* TCP使用三次握手协议来建立连接，握手的第一个报文段是由码位字段的SYN位被置为1来识别，表示请求连接。
* TCP使用三次握手协议来建立连接，第二个报文段码字段的ACK位和SYN位被置为1，指示对第一个报文的确认。
* 一个TCP连接总是以1KB的最大段长发送TCP段，发送方有足够的数据要发送。当拥塞窗口为16KB时发送了超时，如果用慢开始算法，接下来的4个RTT（往返时间）时间内的TCP段的传输是成功的，那么当第4个RTT时间内发送的所有TCP段都得到肯定应答时，拥塞窗口大小是9 KB。
* TCP可靠传输协议中，发送端滑动窗口大小为15，头15帧已发送，接收到的第1个ACK为ACK15，接收端已接收的帧为第14个帧。
* 在TCP协议中，终止连接需要经过4次握手阶段。
* TCP协议为了实现可靠服务，采用超时重传和累计确认技术，规定确认号为下一个希望接收的报文段的首字节序号。
* 假设主机A已收到了来自主机B的编号为0~535的所有字节，同时假设它打算发送一个报文段给主机B。主机A会在它发往主机B的报文段的确认号字段中填入 536。
* 假设主机A已收到了来自主机B的包含字节0~535的报文段，以及另一个包含字节900~1000的报文段。由于某种原因，主机A还没有收到字节536~899的报文段。则A到B的下一个报文段将在确认号字段中包含536。
* TCP拥塞控制算法主要包含三部分，（1）慢启动，（2）拥塞避免，（3）快速恢复。
* TCP拥塞控制中，当发送方定时器超时时，其ssthresh值将被设置为cwnd当前值的一半。

举例：主机A和B通过一个TCP连接通信，且主机B已经收到了来自A的直到字节348的所有字节。假定主机A随后向B发送两个报文段。第一个报文段和第二个报文段分别包含了50和70字节的数据。在第一个报文段中，序号是349，无论何时B收到A的报文段，它都会发送确认。假定两个报文段按序到达B，第一个确认丢失，第二个确认在第一个超时间隔后到达A。其时序图、报文段发送、确认及对应的序号和数据的字节编号如图所示。

Host B

Ack = 469

Ack = 469

Seq = 349, 50 bytes

Seq = 349, 50 bytes

Seq = 399, 70 bytes

Ack = 399

Host A

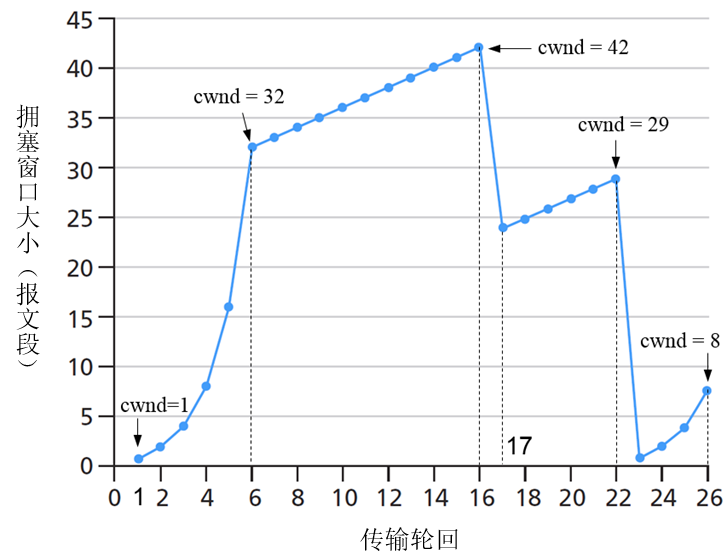
超时

间隔

超时

间隔

1. 如下TCP拥塞窗口演化图中，[1, 6] 和 [23, 26]属于TCP慢启动，[6, 16] 和 [17, 22]为TCP拥塞避免；在第16个传输轮回之后，出现了报文丢失，并且报文段的丢失是由于检测到三个冗余的ACK。而在第22个传输轮回之后，报文段的丢失却是由于检测到了超时。



在第1个传输轮回，ssthreshold的初始值是32。

在第18个传输轮回，ssthreshold的值是21（在第16个传输轮回，拥塞窗口大小为42，此后出现丢包，TCP将阈值ssthreshold设置为拥塞窗口大小的一半（cwnd/2），即ssthreshold = 42/2 = 21）。

1. 网络层基本概念

* IP协议的服务模型是尽力而为的服务模型。
* 网络层所实现的两个重要功能是转发和路由选择。

1. 因特网路由算法

路由算法根据网络拓扑信息来计算最小开销路径，根据网络拓扑信息是全局的或分散的，路由算法可以分为两大类：

1）全局路由算法：利用完整的、全局性的网络信息来计算最小开销路径。运行全局路由算法的路由器需要事先获取整个网络的节点之间的连接关系以及链路开销，然后才能计算从源到目的节点之间的最短路径。这种算法通常被称作“链路状态算法”。

2）分布式路由算法：以一种迭代的、分布式的方式计算最小开销路径。每个路由器只知道道与其直接相连的节点之间的链路开销，而不知道整个网络的完整的连接关系和开销。这种算法通常被称作“距离向量算法”

Internet的主要路由算法：

RIP中的路由更新消息在邻居之间利用RIP响应消息进行交换，每30秒交换一次。再用分布式路由算法进行计算。RIP协议使用跳数作为路由选择的度量标准。

OSPF中的Open指OSPF是一种开放性的路由协议并被认为是RIP的后继协议。OSPF与RIP相比有许多优点。本质上，OSPF是一种使用链路状态泛洪(flooding)的链路状态协议和Dijkstra最短路径算法。每个路由器均可获取整个自治系统的完整网络拓扑并独立地计算以自己为根的最短路径树。

RIP和OSPF两种为自治系统AS内部协议。

BGP是一种自治系统间的路由协议，也是事实上的标准。BGP为自治系统AS之间协议。

目前因特网中使用了两种常用的自治系统（AS）内部路由选择协议，分别是路由选择信息协议（RIP）与开放最短路径优先协议（OSPF）和一种AS间路由选择协议，即边界网关协议（BGP）。

1. 路由器

路由器的交换结构是路由器的核心，其交换方式大致可分为内存、总线、纵横（互联网络）三种方式。

当路由器接收的IP 报文的TTL 值等于0 时，采取的策略是丢掉该分组。

当路由器接收的报文的目的IP地址在路由器路由表中没有对应的表项时，采取策略是丢掉该分组。

路由器用分组的目的地址的前缀与自身路由表中的表项进行匹配。当有多个匹配时，路由器使用最长前缀匹配规则。

路由器根据数据包的目的IP地址作出转发决定。

1. 数据报分片

1）不同链路层协议能够携带的最大传输单元MTU不同，为了将超长的IP分组挤到链路层分组的有效载荷字段。

源发送的某个分组可能需要在某个路由器处分割成多个更小的分组(fragment,片)，以便能够封装在Frame中。

2）某个分组的所有片需要在将其交给传输层协议之前进行重组。根据端到端原则，分片的重组由端系统完成，而不是由路由器完成。端系统的\网络层协议收到fragment后，根据其头部携带的identification（标识）、flag（分片标志）以及fragment offset（片偏移量）等字段信息来对片进行排序，重组等。

属于某个分组的所有fragments具有相同的identifier，根据flag和offset字段的值判断时都收到了所有的fragment并对他们进行排序。当一个分组的一个或多个Fragment没有收到，目的端系统将丢弃这个分组的所有已经收到的Fragment。

3）在路由器里分组在终端系统里组装。

分片在源主机和中间路由器都有可能产生，但片到达目的主机后才进行重新组装。

路由器的数据链路层收到报文的 MTU 大于该路由器将要发出接口的最大 MTU 时，采取的策略是将该分组分片。

1. IP地址

* 应用程序PING 发出的是ICMP 请求报文。
* 主机地址101.11.100.100/255.255.252.0的广播地址是101.11.103.255；10.1.0.1/17的广播地址是10.1.127.255。
* 某公司申请到一个C 类IP 地址，但要连接6 个的子公司，最大的一个子公司有26 台计算机，每个子公司在一个网段中，则子网掩码应设为255.255.255.224。
* 国际上负责分配IP地址的专业组织划分了几个网段作为私有网段，可以供人们在私有网络自由分配使用，包括10.0.0.0/8、172.16.0.0/12、192.168.0.0/16。
* 交换机表中的表项包含MAC地址、通向MAC地址的交换机接口、表项放置在表中的时间。
* 因特网协议中，A类地址第一个字节的范围是1~126。
* 一个B类IP地址最多可用来划分子网的位数是14。
* 一个B类网络，有5位掩码加入缺省掩码用来划分子网，每个子网最多可分配的主机数是2046。
* 一台主机可以分配多个IP地址。一个子网中的主机都有相同的网络号。
* 在C类地址下，默认主机比特位数为8位，但该类地址可支持的主机数为254台。
* 目前因特网的地址分配策略称为无类别域间路由选择（CIDR）。它将子网寻址的概念一般化了，将32比特的IP地址划分成两部分，a.b.c.d/x，其中x指示了地址的x最高比特构成了IP地址的网络部分。比如/21 和 255.255.248.0表示同一个含义。

1. 学院要新建3个实验室，每个实验室有25台主机，已申请了一个C类地址192.168.10.0。请对该C类地址进行子网划分，写出子网掩码及每个实验室的网络号、主机IP地址范围、广播地址（**不使用全0子网和全1子网**）。

（1）确认子网掩码，2*x* > 3 + 2（3个子网，1个全0子网，1个全1子网），得到*x* = 3。所以，子网掩码为255.255.255.11100000，即为255.255.255.224。

（2）每个实验室的网络号、主机IP地址范围、广播地址：

1）第一个子网的网络号为：192.168.10.001 00000，即为192.168.10.32；主机IP地址范围为192.168.10.001 00001 ~ 192.168.10.001 11110，即192.168.10.33~192.168.10.62；广播地址为192.168.10.001 11111，即为192.168.10.63。

2）第二个子网的网络号为：192.168.10.010 00000，即为192.168.10.64；主机IP地址范围为192.168.10.010 00001 ~ 192.168.10.010 11110，即192.168.10.65~192.168.10.94；广播地址为192.168.10.010 11111，即为192.168.10.95。

3）第三个子网的网络号为：192.168.10.011 00000，即为192.168.10.96；主机IP地址范围为192.168.10.011 00001 ~ 192.168.10.011 11110，即192.168.10.97~192.168.10.126；广播地址为192.168.10.011 11111，即为192.168.10.127。

1. 学校实验中心，包含**7**个小机房，每个机房不超过30台计算机。该实验中心申请使用教育网C类IP地址，网络ID为202.196.8.0/24，为了更好管理机房网络,避免机房之间的广播流量，各个小机房网络通过单台路由器实现互联互通。请列出每个机房对应的子网地址、可用IP地址范围、广播地址、子网掩码。要求有解题思路；

提示：对C类IP地址202.196.8.0/24进行子网划分，这里解答时假设**允许子网ID为全0，广播地址全1（即其中一个子网的网络地址与待划分网络的网络地址相同，都是202.196.8.0；其中一个子网的广播地址与待划分网络的广播地址相同，都是202.196.8.255）**，每个机房使用一个子网。

7个机房，即7个子网，子网需要借用主机为3位（23），故子网掩码为255.255.255.11100000，即为255.255.255.224。

子网地址共有8个：202.196.8.\*（\*为0、32、64、96、128、224、160、192 ）

可用IP范围：

202.196.8.1~202.196.8.30

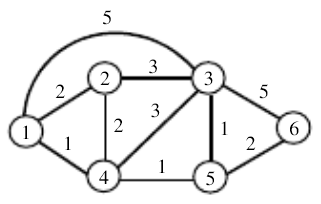
202.196.8.33~202.196.8.62

……

202.196.8.192~202.196.8.254

1. 最短路径Dijkstra算法

以图E-1的网络为例来讨论算法，即寻找从源结点到网络中其他各结点的最短路径。为方便起见，设源结点为结点1。然后一步一步地寻找，每次找一个结点到源结点的最短路径，直到把所有的点都找到为止。



图E-1 求最短路径算法的网络举例

令*D*(*v*)为源结点(记为结点1)到某个结点*v*的距离，它就是从结点1沿某一路径到结点*v*的所有链路的长度之和。再令*l*(*i*, *j*)为结点*i*至结点*j*之间的距离。整个算法只有以下两个部分：

(1) 初始化

令*N*表示网络结点的集合。先令*N* = {1}。对所有不在*N*中的结点*v*，写出

*D*(*v*) = *l*(1, *v*), 若结点 *v*与结点1直接相连

*D*(*v*) = ∞, 若结点 *v*与结点1不直接相连

在用计算机进行求解时，可以用一个比任何路径长度大得多的数值代替∞。对于上述例子，可以使*D*(*v*) = 99。

(2) 寻找一个不在*N*中的结点*w*，其*D*(*w*) 值为最小。把*w*加入到*N*中。然后对所有不在*N*中的结点*v*，用 [*D*(*v*), *D*(*w*) + *l*(*w*, *v*)]中的较小的值去更新原有的*D*(*v*)值，即：

*D*(*v*) ← Min[*D*(*v*), *D*(*w*) + *l*(*w*, *v*)] (E-1)

(3) 重复步骤(2)，直到所有的网络结点都在*N*中为止。

表E-1是对图E-1的网络进行求解的详细步骤。可以看出，上述的步骤(2)共执行了5次。表中带圆圈的数字是在每一次执行步骤(2)时所寻找的具有最小值的*D*(*w*) 值。当第5次执行步骤(2)并得出了结果后，所有网络结点都已包含在*N*之中，整个算法即告结束。



现在我们对以上的最短路径树的找出过程进行一些解释。

因为选择了结点1为源结点，因此一开始在集合*N*中只有结点1。结点1只和结点2, 3和4直接相连，因此在初始化时，在*D*(2)，*D*(3)和*D*(4)下面就填入结点1到这些结点相应的距离，而在*D*(5)和*D*(6)下面填入∞。

下面执行步骤1。在结点1以外的结点中，找出一个距结点1最近的结点*w*，这应当是*w* = 4，因为在*D*(2)，*D*(3)和*D*(4)中，*D*(4) = 1，它的之值最小。于是将结点4加入到结点集合*N*中。这时，我们在步骤1这一行和*D*(4)这一列下面写入①，数字1表示结点4到结点1的距离，数字1的圆圈表示结点4在这个步骤加入到结点集合*N*中了。

接着就要对所有不在集合*N*中的结点（即结点2, 3, 5和6）逐个执行 (E-1) 式。 对于结点2，原来的*D*(2) = 2。现在*D*(*w*) + *l*(*w*, *v*) = *D*(4) + *l*(4, 2) = 1 + 2 = 3 > *D*(2)。因此结点2到结点1距离不变，仍为2。

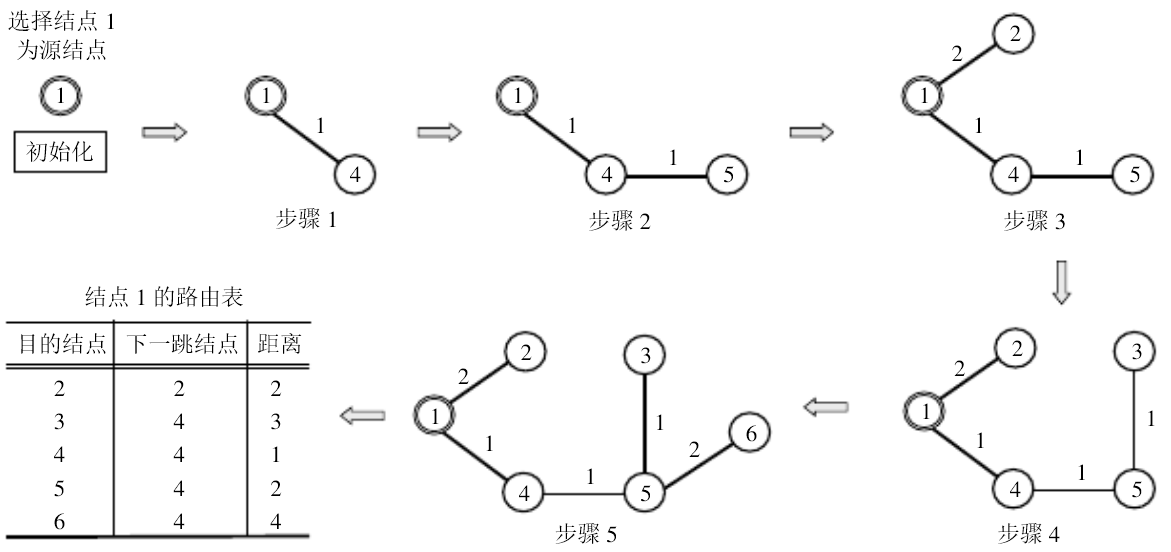
对于结点3，原来的*D*(3) = 5。现在*D*(*w*) + *l*(*w*, *v*) = *D*(4) + *l*(4, 3) = 1 + 3 = 4 < *D*(3)。因此结点3到结点1的距离要更新，从5减小到4。

对于结点5，原来的*D*(5) = ∞。现在*D*(*w*) + *l*(*w*, *v*) = *D*(4) + *l*(4, 5) = 1 + 1 = 2 < *D*(5)。因此结点5到结点1的距离要更新，从∞减小到2。

对于结点6，现在到结点1的距离仍为∞。

步骤1的计算到此就结束了。

下面执行步骤2。在结点1和4以外的结点中，找出一个距结点1最近的结点*w*。现在有两个结点（结点2和5）到结点1的距离一样，都是2。我们选择结点5（当然也可以选择结点2，最后得出的结果还是一样的）。以后的步骤类似。



图E-2 用Dijkstra算法求出最短路径树的各个步骤和在结点1的路由表

最后就得出以结点1为根的最短路径树。图E-2给出了各步骤执行后的结果。从最短路径树可清楚地找出从源结点(结点1)到网内任何一结点的最短路径。图E-2还给出了在结点1的路由表。此路由表指出对于发往某个目的结点的分组，从结点1发出后的下一跳结点（在算法中常称为“后继结点”）和距离。当然，像这样的路由表，在所有其他各结点中都有一个。但这就需要分别以这些结点为源结点，重新执行算法，然后才能找出以这个结点为根的最短路径树和相应的路由表。

1. 考虑下面的网络：对于标明的链路费用，用Dijkstra最短路径算法可以计算出从 *x* 到所有网络节点的最短路径，如表所示。

7

4

3

3

2

4

8

12

8

6

3

6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Step* | *N’* | *D(t),p(t)* | *D(u),p(u)* | *D(v),p(v)* | *D(w),p(w)* | *D(y),p(y)* | *D(z),p(z)* |
| 0 | *x* | ∞ | ∞ | 3,x | 6,x | 6,x | 8,x |
| 1 | *xv* | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 2 | *xvu* | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 3 | *xvuw* | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 4 | *xvuwy* | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 5 | *xvuwyt* | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 6 | *xvuwytz* | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |

根据上表，可得到用Dijkstra算法求的每个步骤的最短路径树和在结点 *x* 的路由表，如下所示：

结点*x*

为源节点

初始化

步骤1

3

3

步骤2

3

步骤3

3

6

3

3

步骤4

6

3

6

步骤5

3

4

6

3

6

步骤6

3

4

8

6

3

6

结点 *x* 的路由表：

|  |  |  |
| --- | --- | --- |
| 目的结点 | 下一跳结点 | 距离 |
| *t* | *v* | 7 |
| *u* | *v* | 6 |
| *v* | *v* | 3 |
| *w* | *w* | 6 |
| *y* | *y* | 6 |
| *z* | *z* | 8 |

1. 差错检测和纠正技术

* **一比特奇偶校验**

发送方：

1）在要发送的信息D（d位）后面附加一个奇偶校验位

2）使“1”的个数是奇数（奇校验）或偶数（偶校验）

3）一起传输发送（d+1位）。

0111000110101011

1

*d*位数据

校验位

偶校验

接收方：

1）检测收到的信息（d+1位）中“1”的个数。

2）偶校验：发现奇数个“1”，至少有一个比特发生差错（奇数个比特差错）。

3）奇校验：发现偶数个 “1”，至少有一个比特发生差错。

特点：

1）可以查出任意奇数个错误，但不能发现偶数个错误。

2）若比特差错概率很小，差错独立发生，一比特奇偶校验可满足要求。

3）若差错集中一起“突发”（突发差错），一帧中未检测到的差错的概率达到50%。

* **因特网校验和**

**发送方：**

* 将数据的每两个字节当作一个16位的整数，可分成若干整数；
* 将所有16 位的整数求和；
* 对得到的和逐位取反，作为检查和，放在报文段首部，一起发送。

**接收方：**

* 对接收到的信息 (包括检查和)按与发送方相同的方法求和。
* 全“1”：收到的数据无差错；
* 其中有“0”：收到的数据出现差错。

0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0

0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0

**1** 0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 1

0 1 0 0 1 0 1 0 1 1 0 0 0 0 1 0

1回卷

和

检验和（取反）

1 0 1 1 0 1 0 1 0 0 1 1 1 1 0 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

无差错，和为：

* **循环冗余检测CRC**

**基本思想：**

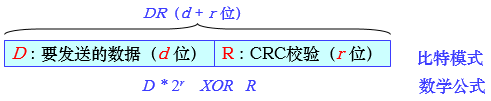
* 设发送节点要把数据D（*d* 比特）发送给接收节点。
* 发送方和接收方先共同选定一个生成多项式 G（*r*+1比特），最高有效位 (最 左边)是1。

**发送方：**

* 计算出一个 *r*位 附加比特 *R，*添加到 *D* 的后面产生*DR*（*d+r* 比特）
* *DR* 能被 *G* 模 2 运算整除*，*一起发送。

**接收方：**用 *G* 去除接收到的 *DR（d+r* 比特*）*

* 余数非0：传输发生差错；
* 余数为0：传输正确，去掉尾部 *r* 位，得所需数据 *D*。



1. MAC的两种方式以及CSMA/CD

多路访问控制协议可以大致分为三类：

1)信道划分协议；

2)随机访问协议；

3)轮转协议。

CSMA(载波侦听多路访问)中，节点在传输Frame之前侦听信道，如果信道空闲时才开始发送整个Frame。不过，CSMA中的节点在开始发送Frame后就要发送整个Frame，不管在该Frame的发送过程中有没有碰撞产生。

CSMA/CD与CSMA类似，节点在发送数据之前首先侦听信道，如果信道忙，则延后一段时间继续侦听信道，直到信道空闲才能开始发送；另外，CSMA/CD节点在发送Frame的同时继续侦听信道，如果检测到碰撞，则立即中止Frame的发送。CSMA/CD对CSMA的改善是显而易见的。

1. ARP协议

地址解析协议(Address Resolution Protocol)的工作是从IP地址得到对应的MAC地址。

每个主机的ARP模块维护ARP表，ARP表的结构一般为<IP Address，MAC Address, TTL>。其中，TTL指示表项从开始创建到从表中删除的时间。

同一个子网内部的ARP处理过程：

（1）主机A构造一个ARP查询消息，向子网内所有主机广播，消息中包含欲解析主机(B)的IP地址。

（2）主机B收到ARP查询消息后，向A回答自己的MAC地址。

ARP表是自动生成的，无须手工操作。同时，ARP表也起着Cache的作用。

1. 以太网技术

HUB、交换机的区别、交换机的自学习、交换机的切入式转发

**1）HUB、交换机的区别**

集线器（HUB）本质上是一个物理层设备，它作用于单个bit而不是Frame。Hub将收到的信号进行再生和放大，并广播所收到的每一位。

由于Hub工作在物理层，所以它没有实现CSMA/CD，要靠主机中的网络适配器来检测冲突。

Hub可以收集信息，提供一定的网络管理功能。比如，将一个故障站点断开连接。

交换机是数据链路层设备，利用存储转发机制处理Frame。交换机收到一个Frame后，检查其中的目的MAC地址，查找本地MAC地址表来决定Frame的出口。当Frame被转发到一个LAN Segment时，利用CSMA/CD来访问Segment的广播信道。

**2）交换机的自学习**

若从 A 发出的帧从接口 x 进入了某交换机，那么从这个接口出发沿相反方向一定可把一个帧传送到 A。

交换机每收到一个帧，就记下其源地址和进入交换机的接口，作为交换表中的一个项目。

在建立建环表时是把帧首部中的源地址写在“地址”这一栏的下面。

在转发帧时，则是根据收到的帧首部中的目的地址来转发的。这时就把在“地址”栏下面已经记下的源地址当作目的地址，而把记下的进入接口当作转发接口。

**3）交换机的切入式转发**

交换机载转发时不用等收集到一个完整的帧就开始转发。

* 传统以太网的最大传输包长MTU是1518字节。
* IP地址具有层次寻址结构，因为它分为网络部分和主机部分，而MAC地址分配由IEEE分配前半部分，适配器生产公司自己分配后半部分，属于平面结构。
* 链路层交换机可以隔离冲突域，但不能隔离广播域，而路由器可以隔离广播域。
* 现代全双工的交换机在局域网中由于可以在发送数据的同时接收数据，而不会有碰撞，故没有必要使用MAC层协议，如CSMA/CD。
* 在一台交换机上，要将端口划分成不同的组，以实现不同组的流量及访问相互隔离，可使用虚拟局域网技术（VLAN）。