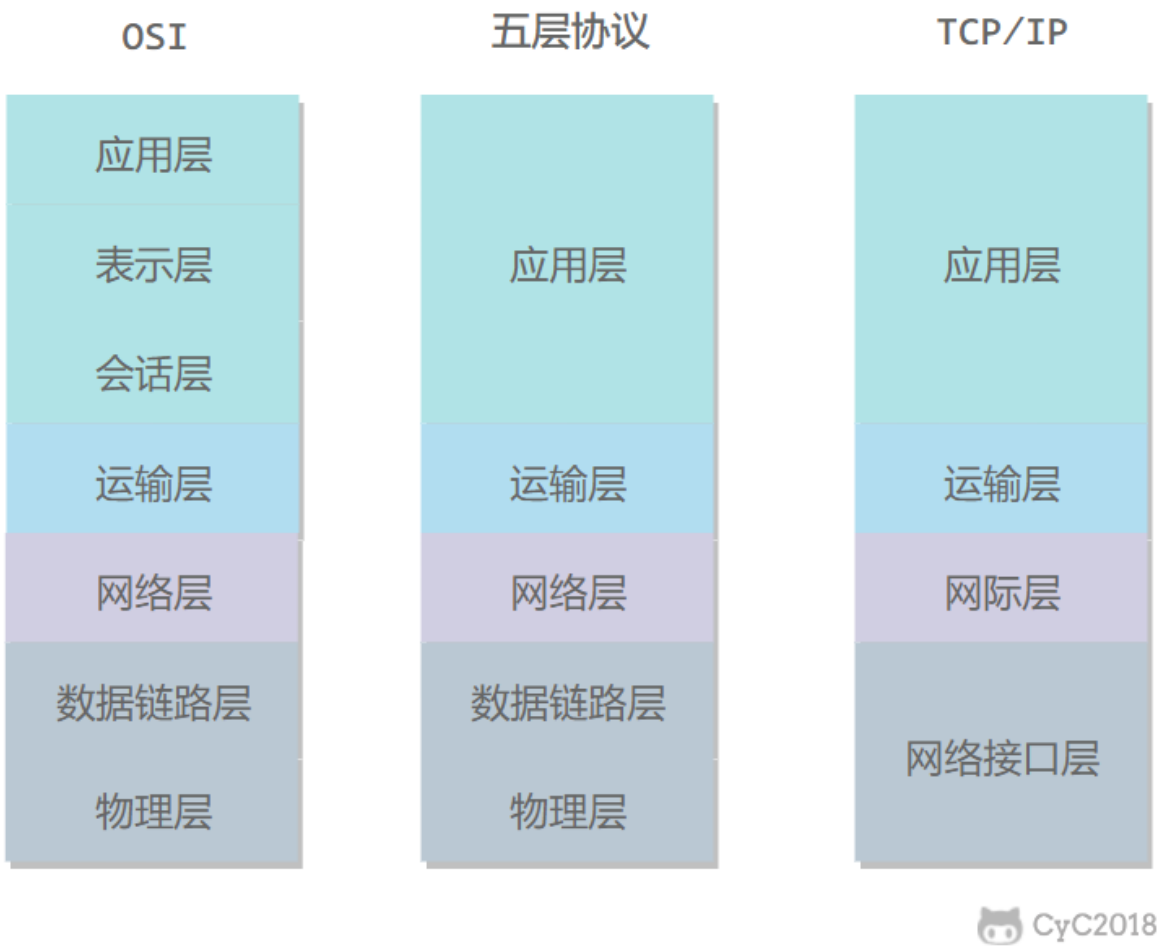
计算机网络

计算机网络体系结构



五层协议

- 应用层：为特定应用程序提供数据传输服务。
- 运输层：为进程提供通用的数据传输服务。
- 网络层：为主机提供数据传输服务。
- 数据链路层：网络层针对的还是主机之间的数据传输服务，而主机之间可以有很多链路，链路层协议就是为同一链路的主机提供数据传输服务。
- 物理层：考虑的是怎样在传输媒体上传输数据比特流，而不是指具体的传输媒体。

OSI

其中表示层和会话层用途如下：

- **表示层**：数据压缩、加密以及数据描述，这使得应用程序不必关心在各台主机中数据内部格式不同的问题。
- **会话层**：建立及管理会话。

五层协议没有表示层和会话层，而是将这些功能留给应用程序开发者处理。

OSI仅仅是一种理论规范，实际开发中用到的是TCP/IP

TCP 的三次握手

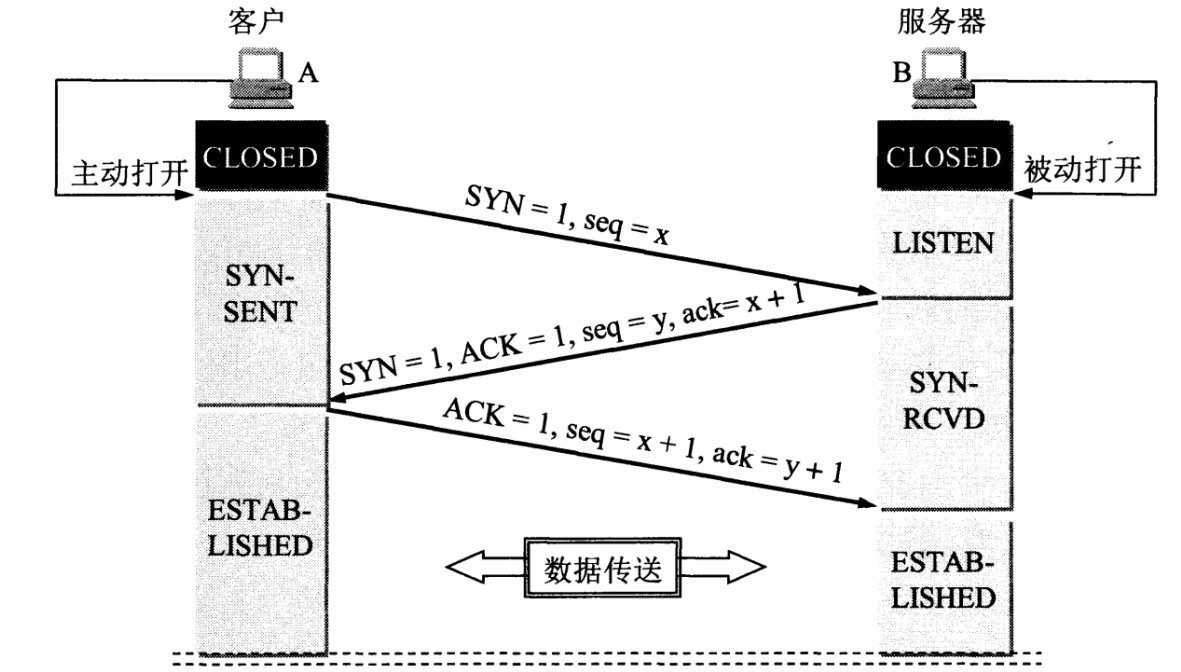


图 5-28 用三报文握手建立 TCP 连接

假设 A 为客户端，B 为服务器端。

- 首先 B 处于 LISTEN（监听）状态，等待客户的连接请求。
- A 向 B 发送连接请求报文，SYN=1，ACK=0，选择一个初始的序号 x。
- B 收到连接请求报文，如果同意建立连接，则向 A 发送连接确认报文，SYN=1，ACK=1，确认号为 x+1，同时也选择一个初始的序号 y。
- A 收到 B 的连接确认报文后，还要向 B 发出确认，确认号为 y+1，序号为 x+1。
- B 收到 A 的确认后，连接建立。

三次握手的原因

第三次握手是为了防止失效的连接请求到达服务器，让服务器错误打开连接。

客户端发送的连接请求如果在网络中滞留，那么就会隔很长一段时间才能收到服务器端发回的连接确认。客户端等待一个超时重传时间之后，就会重新请求连接。但是这个滞留的连接请求最后还是会到达服务器，如果不进行三次握手，那么服务器就会打开两个连接。如果有第三次握手，客户端会忽略服务器之后发送的对滞留连接请求的连接确认，不进行第三次握手，因此就不会再次打开连接。

TCP四次挥手

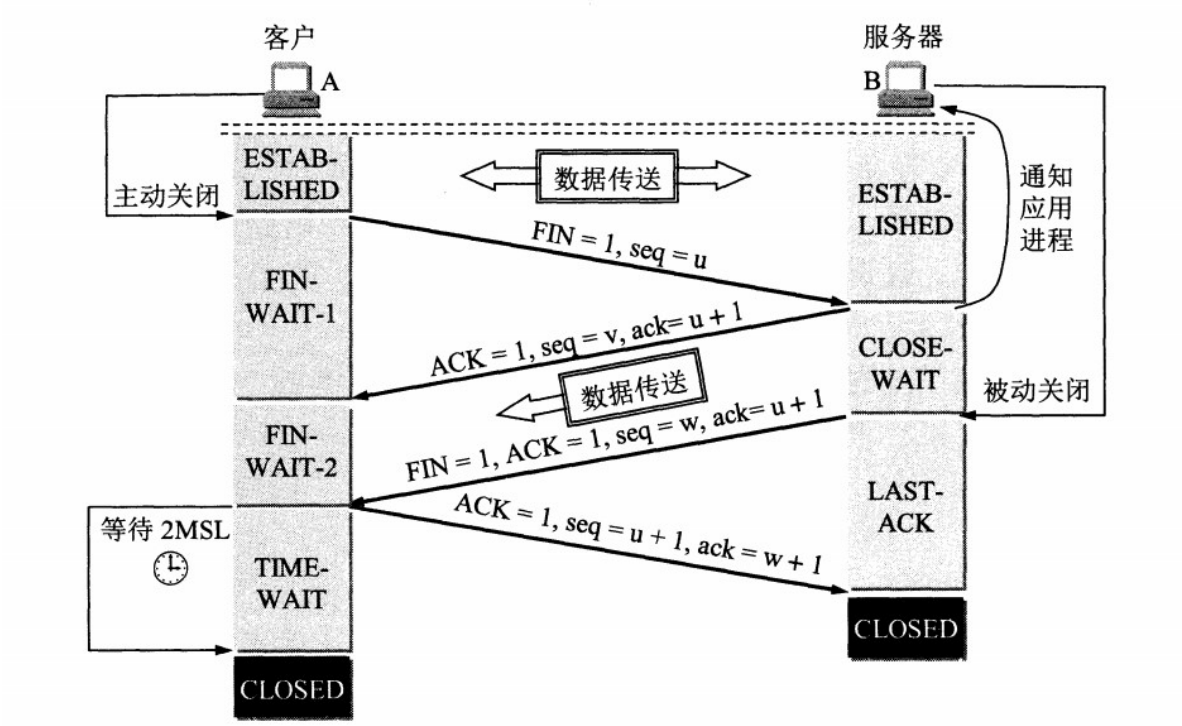


图 5-29　TCP 连接释放的过程

以下描述不讨论序号和确认号，因为序号和确认号的规则比较简单。并且不讨论 ACK，因为 ACK 在连接建立之后都为 1。

- A 发送连接释放报文，FIN=1。
- B 收到之后发出确认，此时 TCP 属于半关闭状态，B 能向 A 发送数据但是 A 不能向 B 发送数据。
- 当 B 不再需要连接时，发送连接释放报文，FIN=1。
- A 收到后发出确认，进入 TIME-WAIT 状态，等待 2 MSL（最大报文存活时间）后释放连接。
- B 收到 A 的确认后释放连接。

四次挥手的原因

客户端发送了 FIN 连接释放报文之后，服务器收到了这个报文，就进入了 CLOSE-WAIT 状态。这个状态是为了让服务器端发送还未传送完毕的数据，传送完毕之后，服务器会发送 FIN 连接释放报文。

TIME_WAIT

客户端接收到服务器端的 FIN 报文后进入此状态，此时并不是直接进入 CLOSED 状态，还需要等待一个时间计时器设置的时间 2MSL。这么做有两个理由：

- 确保最后一个确认报文能够到达。如果 B 没收到 A 发送来的确认报文，那么就会重新发送连接释放请求报文，A 等待一段时间就是为了处理这种情况的发生。
- 等待一段时间是为了让本连接持续时间内所产生的所有报文都从网络中消失，使得下一个新的连接不会出现旧的连接请求报文。

TCP 协议如何保证可靠传输

1. 应用数据被分割成 TCP 认为最适合发送的数据块。
2. TCP 给发送的每一个包进行编号，接收方对数据包进行排序，把有序数据传送给应用层。
3. **校验和：** TCP 将保持它首部和数据的检验和。这是一个端到端的检验和，目的是检测数据在传输过程中的任何变化。如果收到段的检验和有差错，TCP 将丢弃这个报文段和不确认收到此报文段。
4. TCP 的接收端会丢弃重复的数据。
5. **流量控制：** TCP 连接的每一方都有固定大小的缓冲空间，TCP的接收端只允许发送端发送接收端缓冲区能接纳的数据。当接收方来不及处理发送方的数据，能提示发送方降低发送的速率，防止包

丢失。TCP 使用的流量控制协议是可变大小的滑动窗口协议。 （TCP 利用滑动窗口实现流量控制）

6. **拥塞控制：** 当网络拥塞时，减少数据的发送。
7. **ARQ协议：** 也是为了实现可靠传输的，它的基本原理就是每发完一个分组就停止发送，等待对方确认。在收到确认后再发下一个分组。
8. **超时重传：** 当 TCP 发出一个段后，它启动一个定时器，等待目的端确认收到这个报文段。如果不能及时收到一个确认，将重发这个报文段。

TCP,UDP 协议的区别

类型	特点			性能		应用场景	首部字节
	是否面向连接	传输可靠性	传输形式	传输效率	所需资源		
TCP	面向连接	可靠	字节流	慢	多	要求通信数据可靠 （如文件传输、邮件传输）	20-60
UDP	无连接	不可靠	数据报文段	快	少	要求通信速度高 （如域名转换）	8个字节 （由4个字段组成）

UDP 在传送数据之前不需要先建立连接，远地主机在收到 UDP 报文后，不需要给出任何确认。虽然 UDP 不提供可靠交付，但在某些情况下 UDP 确是一种最有效的工作方式（一般用于即时通信），比如： QQ 语音、 QQ 视频 、直播等等。

TCP 提供面向连接的服务。在传送数据之前必须先建立连接，数据传送结束后要释放连接。 TCP 不提供广播或多播服务。由于 TCP 要提供可靠的，面向连接的传输服务（TCP的可靠体现在TCP在传递数据之前，会有三次握手来建立连接，而且在数据传递时，有确认、窗口、重传、拥塞控制机制，在数据传完后，还会断开连接用来节约系统资源），这一难以避免增加了许多开销，如确认，流量控制，计时器以及连接管理等。这不仅使协议数据单元的首部增大很多，还要占用许多处理机资源。TCP 一般用于文件传输、发送和接收邮件、远程登录等场景。

TCP 滑动窗口

窗口是缓存的一部分，用来暂时存放字节流。发送方和接收方各有一个窗口，接收方通过 TCP 报文段中的窗口字段告诉发送方自己的窗口大小，发送方根据这个值和其它信息设置自己的窗口大小。

发送窗口内的字节都允许被发送，接收窗口内的字节都允许被接收。如果发送窗口左部的字节已经发送并且收到了确认，那么就将发送窗口向右滑动一定距离，直到左部第一个字节不是已发送并且已确认的状态；接收窗口的滑动类似，接收窗口左部字节已经发送确认并交付主机，就向右滑动接收窗口。

接收窗口只会对窗口内最后一个按序到达的字节进行确认，例如接收窗口已经收到的字节为 {31, 34, 35}，其中 {31} 按序到达，而 {34, 35} 就不是，因此只对字节 31 进行确认。发送方得到一个字节的确认之后，就知道这个字节之前的所有字节都已经被接收。

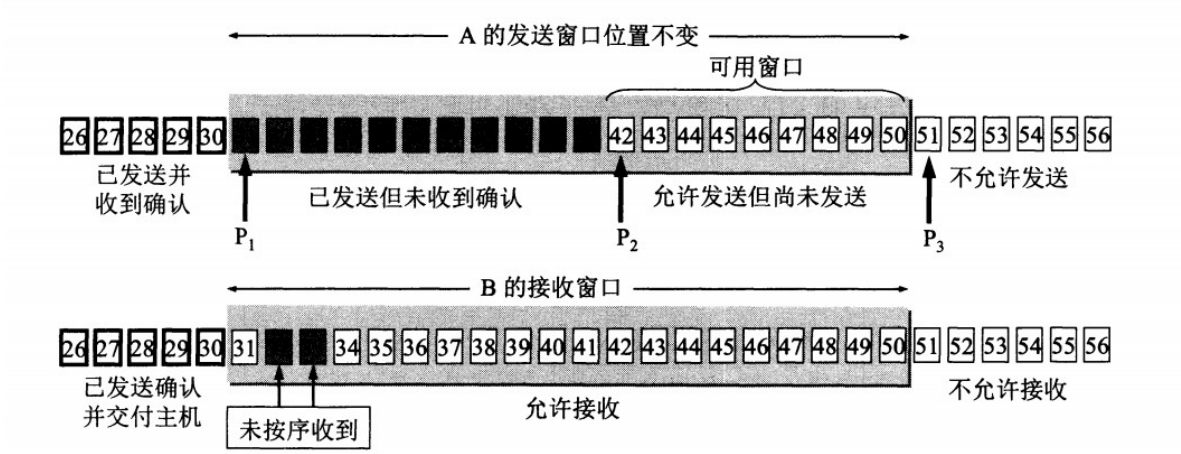


图 5-16　A 发送了 11 个字节的数据

TCP 流量控制

流量控制是为了控制发送方发送速率，保证接收方来得及接收。

接收方发送的确认报文中的窗口字段可以用来控制发送方窗口大小，从而影响发送方的发送速率。将窗口字段设置为 0，则发送方不能发送数据。

TCP 拥塞控制

如果网络出现拥塞，分组将会丢失，此时发送方会继续重传，从而导致网络拥塞程度更高。因此当出现拥塞时，应当控制发送方的速率。这一点和流量控制很像，但是出发点不同。流量控制是为了让接收方能来得及接收，而拥塞控制是为了降低整个网络的拥塞程度。

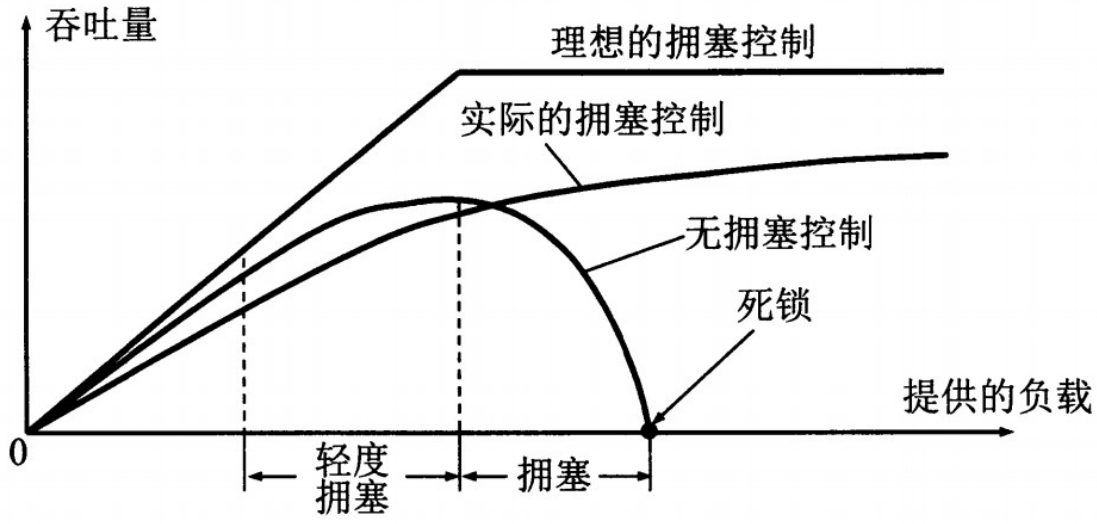


图 5-23　拥塞控制所起的作用

TCP 主要通过四个算法来进行拥塞控制：慢开始、拥塞避免、快重传、快恢复。

发送方需要维护一个叫做拥塞窗口（cwnd）的状态变量，注意拥塞窗口与发送方窗口的区别：拥塞窗口只是一个状态变量，实际决定发送方能发送多少数据的是发送方窗口。

为了便于讨论，做如下假设：

- 接收方有足够大的接收缓存，因此不会发生流量控制；
- 虽然 TCP 的窗口基于字节，但是这里设窗口的大小单位为报文段。

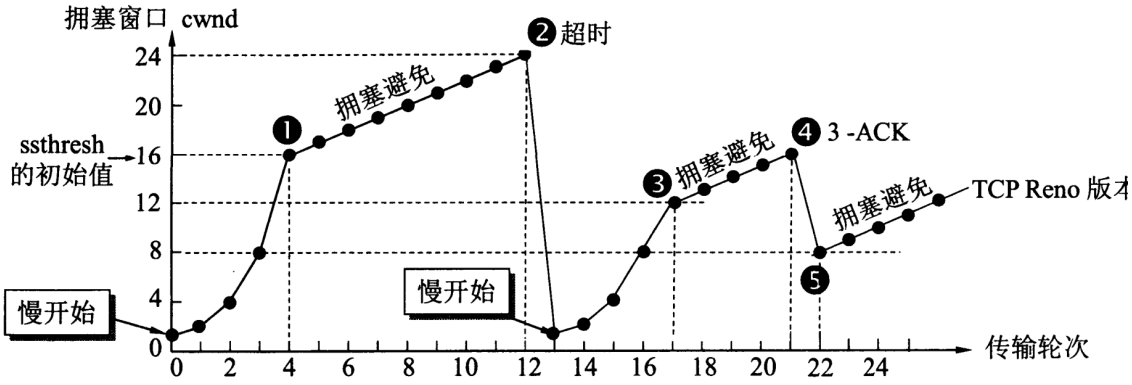


图 5-25　TCP 拥塞窗口 cwnd 在拥塞控制时的变化情况

1. 慢开始与拥塞避免

发送的最初执行慢开始，令 cwnd = 1，发送方只能发送 1 个报文段；当收到确认后，将 cwnd 加倍，因此之后发送方能够发送的报文段数量为：2、4、8 ...

注意到慢开始每个轮次都将 cwnd 加倍，这样会让 cwnd 增长速度非常快，从而使得发送方发送的速度增长速度过快，网络拥塞的可能性也就更高。设置一个慢开始门限 ssthresh，当 cwnd >= ssthresh 时，进入拥塞避免，每个轮次只将 cwnd 加 1。

如果出现了超时，则令 ssthresh = cwnd / 2，然后重新执行慢开始。

2. 快重传与快恢复

在接收方，要求每次接收到报文段都应该对最后一个已收到的有序报文段进行确认。例如已经接收到 M1 和 M2，此时收到 M4，应当发送对 M2 的确认。

在发送方，如果收到三个重复确认，那么可以知道下一个报文段丢失，此时执行快重传，立即重传下一个报文段。例如收到三个 M2，则 M3 丢失，立即重传 M3。

在这种情况下，只是丢失个别报文段，而不是网络拥塞。因此执行快恢复，令 ssthresh = cwnd / 2 ，cwnd = ssthresh，注意到此时直接进入拥塞避免。

慢开始和快恢复的快慢指的是 cwnd 的设定值，而不是 cwnd 的增长速率。慢开始 cwnd 设定为 1，而快恢复 cwnd 设定为 ssthresh。

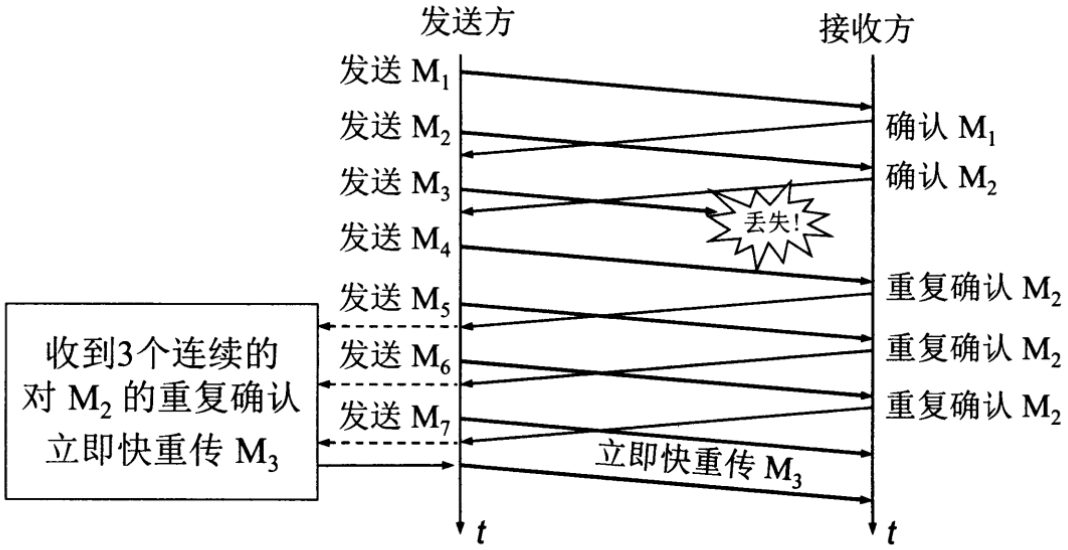


图 5-26　快重传的示意图

交换机和路由器

主要的区别体现在一下几个方面：

（1）外形上：
从外形上我们区分两者，交换机通常端口比较多看起来比较笨重，而路由器的端口就少得多体积也小得多，实际上右图并不是真正的路由器只是集成了路由器的功能，除此之外还有交换机的功能（LAN口就是作为交换机的端口来使用，WAN是用于连接外网的端口），而两个天线则是无线AP接入点（即是通常所说的无线局域网wifi）。

（2）工作层次不同：
最初的交换机工作在OSI开放式系统互联模型的数据链路层，也就是第二层，而路由器则工作在OSI模型的网络层，就是第三层。也就是由于这一点所以交换机的原理比较简单，一般都是采用硬件电路实现数据帧的转发，而路由器工作在网络层，肩负着网络互联的重任，要实现更加复杂的协议，具有更加智能的转发决策功能，一般都会在在路由器中跑操作系统，实现复杂的路由算法，更偏向于软件实现其功能。

（3）数据的转发对象不同：
交换机是根据MAC地址转发数据帧，而路由器则是根据IP地址来转发IP数据报/分组。数据帧是在IP数据包/分组的基础上封装了帧头（源MAC和目的MAC等）和帧尾（CRC校验码）。而对于MAC地址和IP地址大家也许就搞不明白了，为何需要两个地址，实际上IP地址决定最终数据包要到达某一台主机，而MAC地址则是决定下一跳将要交互给哪一台设备（一般是路由器或主机）。而且，IP地址是软件实现的，可以描述主机所在的网络，MAC地址是硬件实现的，每一个网卡在出厂的时候都会将全世界唯一的MAC地址固化在网卡的ROM中，所以MAC地址是不能被修改的，但是IP地址是可以被网络管理人员配置修改的。

（4）”分工“不同
交换机主要是用于组建局域网，而路由器则是负责让主机连接外网。多台主机可以通过网线连接到交换机，这时就组建好了局域网，就可以将数据发送给局域网中的其他主机，如我们使用的飞秋、极域电子教室等局域网软件就是通过交换机把数据转发给其他主机的，当然像极域电子教室这样的广播软件是利用广播技术让所有的主机都收到数据的。然而，通过交换机组建的局域网是不能访问外网的（即是Internet）,这时需要路由器来为我们”打开外面精彩世界的大门“，局域网的所有主机使用的都是私网的IP，所以必须通过路由器转化为公网的IP之后才能访问外网。

（5）冲突域和广播域
交换机分割冲突域，但是不分割广播域，而路由器分割广播域。由交换机连接的网段仍属于同一个广播域，广播数据包会在交换机连接的所有网段上传播，在这种情况下会导致广播风暴和安全漏洞问题。而连接在路由器上的网段会被分配不通的广播域，路由器不会转发广播数据。需要说明的是单播的数据包在局域网中会被交换机唯一地送往目标主机，其他主机不会接收到数据，这是区别于原始的集线器的，数据的到达时间由交换机的转发速率决定，交换机会转发广播数据给局域网中的所有主机。

最后需要说明的是：路由器一般有防火墙的功能，能够对一些网络数据包选择性过滤。现在的一些路由器都具备交换机的功能（如上图右），一些交换机具备路由器的功能，被称为3层交换机，广泛使用。相比较而言，路由器的功能较交换机要强大，但是速度也较慢，价格昂贵，三层交换机既有交换机的线性转发报文的能力，又有路由器的良好的路由功能因此得到广泛的使用。

当然关于路由器和交换机的一些介绍远不止这些，上述所说是主要的一些区别，同时也是本人对路由器和交换机的浅显认识，如有其他一些较明显的区别特征望给出宝贵意见。