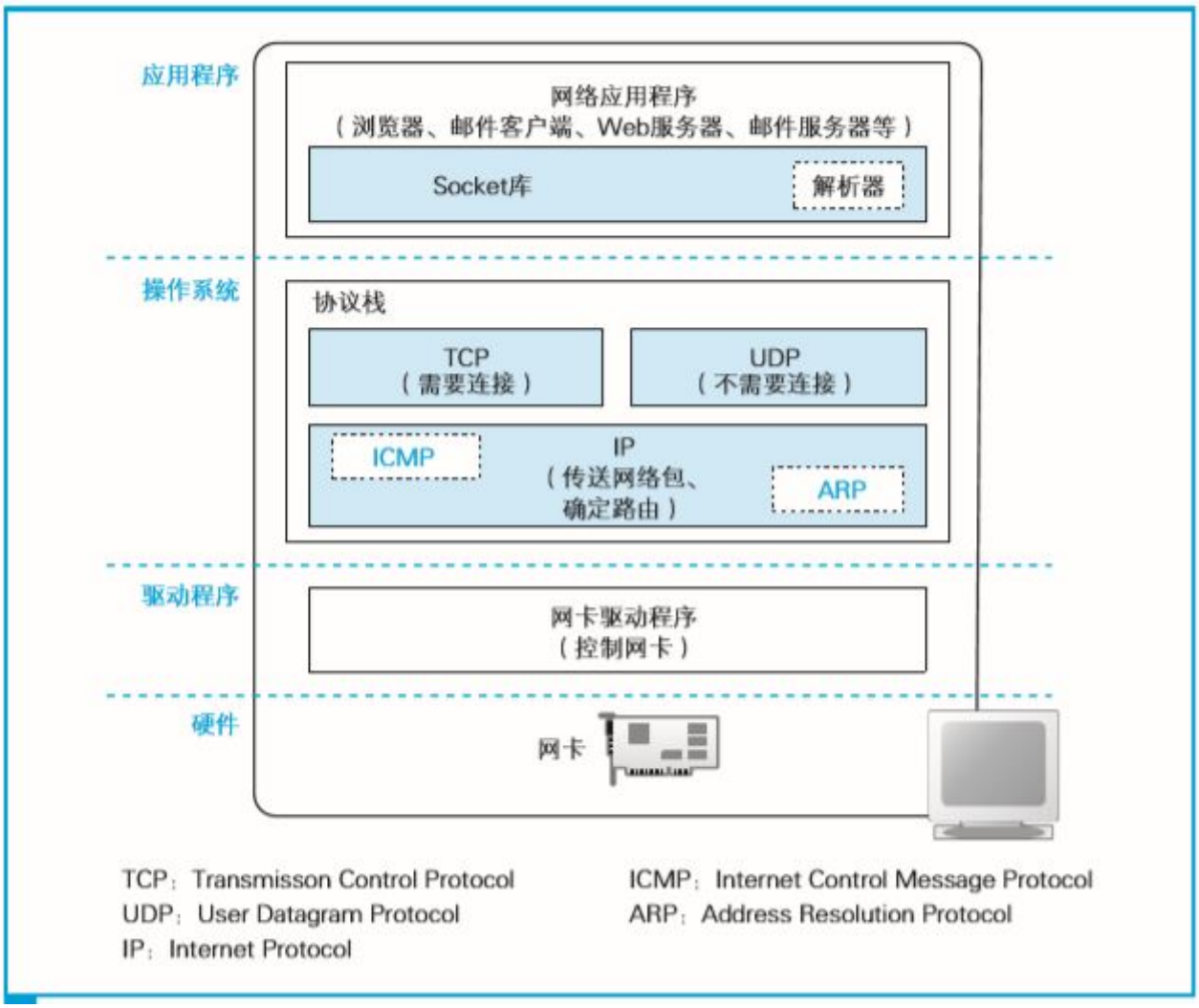
二、探索协议栈和网卡

本章内容：

- 创建套接字：介绍协议栈的内部结构、套接字的实体、以及创建套接字的操作过程。
- 连接服务器
- 收发数据 (重传)
- 从服务器断开连接并删除套接字
- IP与以太网的包收发操作
- 用UDP协议收发数据的操作

2.1 创建套接字

2.1.1 协议栈的内部结构



TCP和UDP:

- 浏览器、邮件等一般应用程序收发数据时用 TCP；

- DNS 查询等收发较短的控制数据时用 UDP

IP:

- ICMP用于告知网络包传送过程中产生的错误以及各种控制消息
- ARP用于根据IP地址查询相应的以太网MAC地址

2.1.2 套接字的实体就是通信控制信息

存放控制信息的内存空间 (里记录了用于 控制通信操作的控制信息，例如通信对象的IP地址、端口号、通信操作的 进行状态等) 就是套接字的实体

协议栈是根据套接字中记录的控制信息来工作.

- windows可以使用netstat 命令显示套接字内容

2.1.3 调用socket时的操作

- 创建套接字时，首先分配一个套接字所需的内存空间，然后向其中写入初始状态。
- 将表示这个套接字的描述符告知应用程序。应用程序在向协议栈进行收发数据委托时就需要提 供这个描述符。

2.2 连接服务器

2.2.1 连接的意思

连接实际上是通信双方交流控制信息。

2.2.2负责保存控制信息的头部

控制信息大体分为两类;

- 头部中记录的信息。
  TCP的头部信息:

字段名称		长度（比特）	含　义
TCP头部（20字节～）	发送方端口号	16	发送网络包的程序的端口号
	接收方端口号	16	网络包的接收方程序的端口号
	序号（发送数据的顺序编号）	32	发送方告知接收方该网络包发送的数据相当于所有发送数据的第几个字节
	ACK 号（接收数据的顺序编号）	32	接收方告知发送方接收方已经收到了所有数据的第几个字节。其中，ACK 是 acknowledge 的缩写
	数据偏移量	4	表示数据部分的起始位置，也可以认为表示头部的长度
	保留	6	该字段为保留，现在未使用
	控制位	6	该字段中的每个比特分别表示以下通信控制含义。 URG：表示紧急指针字段有效 ACK：表示接收数据序号字段有效，一般表示数据已被接收方收到 PSH：表示通过 flush 操作发送的数据 RST：强制断开连接，用于异常中断的情况 SYN：发送方和接收方相互确认序号，表示连接操作 FIN：表示断开连接
	窗口	16	接收方告知发送方窗口大小（即无需等待确认可一起发送的数据量）
	校验和	16	用来检查是否出现错误
	紧急指针	16	表示应紧急处理的数据位置
	可选字段	可变长度	除了上面的固定头部字段之外，还可以添加可选字段，但除了连接操作之外，很少使用可选字段

- 套接字（协议栈中的内存空间）中记录的信息。
  例如，Windows和 Linux操作系统的内部结构不同，协议栈的实现方式不同，必要的控制信息也就不同

2.2.3 连接操作的实际过程

三次握手：

2.3 收发数据

- 缓存区、计时器
- MTU：表示一个网络包的最大长度(一般是1500字节)。
- MSS： 一个数据包中所能容纳的最大数据长度。

2.3.2 对较大数据进行拆分

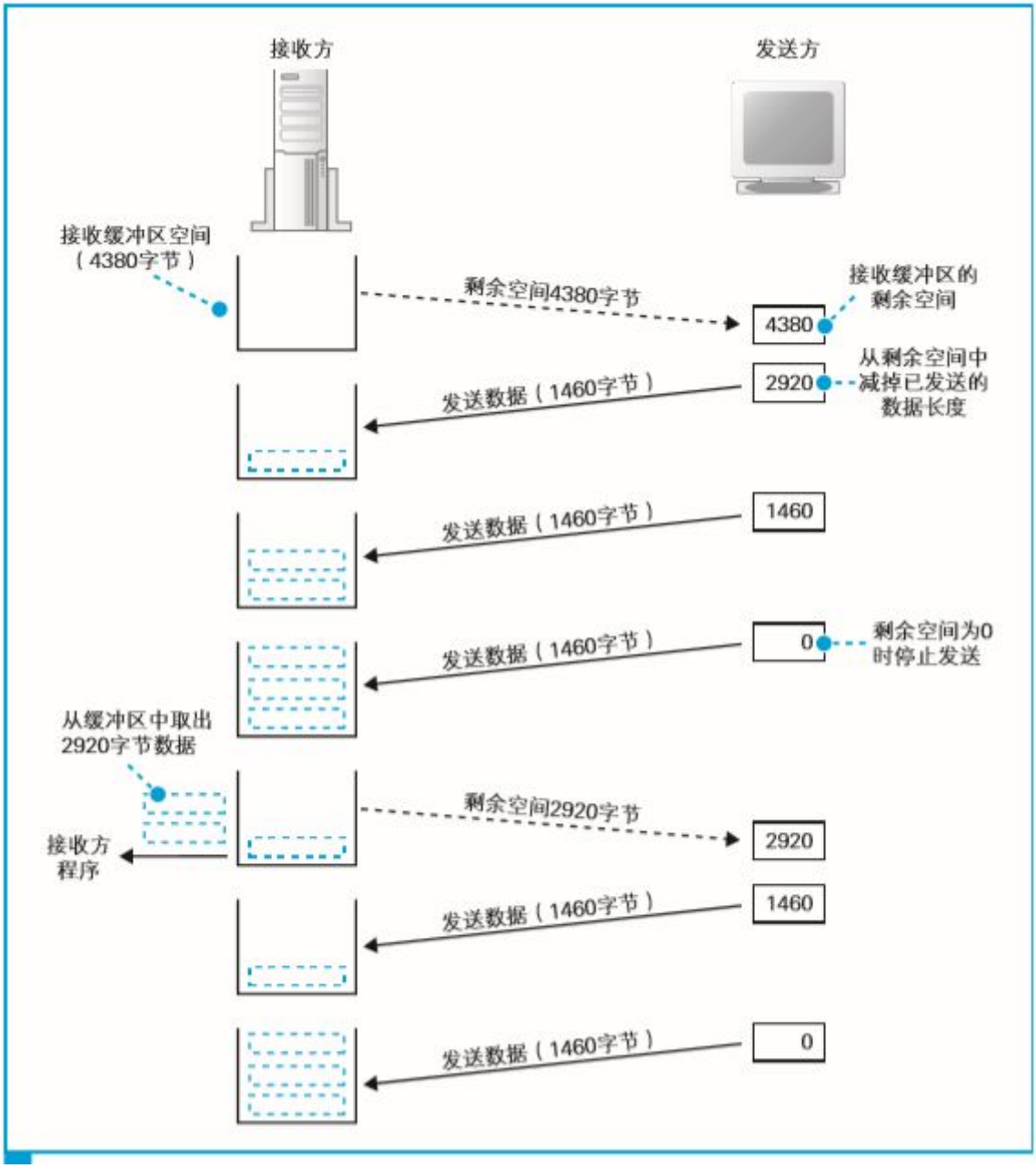
2.3.3 使用ACK号确认数据包已收到

- 通过“序号”和“ACK 号”可以确认接收方是否收到了网络包

2.3.4 根据数据包平均往返时间调整等待时间：TCP会在发送数据 的过程中持续测量ACK号的返回时间，如果ACK号返回变慢，则相应 延长等待时间；相对地，如果ACK号马上就能返回，则相应缩短等待 时间

2.3.5 使用窗口有效管理ACK号

- 滑动窗口
  为防止缓冲区溢出造成的丢包：接收方需要告诉发送方自己最多能接收多少数据， 然后发送方根据这个值对数据发送操作进行控制，这就是滑动窗口方式的 基本思路。



2.3.6 ACK与窗口的合并

2.3.7 接收HTTP响应消息

2.4 从服务器断开并删除套接字

- 四次挥手

2.5 IP与以太网的包收发操作

- 路由器根据目标地址判断下一个路由器的位置 (IP协议，使用IP头部)
  路由器中有一张路由表
- 集线器在子网中将网络包传输到下一个路由 (以太网协议，使用MAC头部)
  集线器里有一张用于以太网协议的表

2.5.2 包收发操作概览

包收发操作的起点：TCP模块委托IP模块发送包的操作。传给IP模块的数据：TCP头部和数据以及指定通信对象的IP地址。

IP模块负责添加如下两个头部
（1） MAC 头部：以太网用的头部，包含 MAC 地址
（2） IP 头部：IP 用的头部，包含 IP 地址

无论要收发的包是控制包还是数据包，IP对各种类型的包的收 发操作都是相同的

2.5.3 生成包含接收方IP地址的IP头部

IP头部内容：

字段名称		长度（比特）	含　义
IP 头部（20 字节～）	版本号	4	IP 协议版本号，目前使用的是版本 4
	头部长度（IHL）	4	IP 头部的长度。可选字段可导致头部长度变化，因此这里需要指定头部的长度
	服务类型（ToS）	8	表示包传输优先级。最初的协议规格里对这个参数的规定很模糊，最近 DiffServ 规格重新定义了这个字段的用法
	总长度	16	表示 IP 消息的总长度
	ID 号	16	用于识别包的编号，一般为包的序列号。如果一个包被 IP 分片，则所有分片都拥有相同的 ID
	标志（Flag）	3	该字段有 3 个比特，其中 2 个比特有效，分别代表是否允许分片，以及当前包是否为分片包
	分片偏移量	13	表示当前包的内容为整个 IP 消息的第几个字节开始的内容
	生存时间（TTL）	8	表示包的生存时间，这是为了避免网络出现回环时一个包永远在网络中打转。每经过一个路由器，这个值就会减 1，减到 0 时这个包就会被丢弃
	协议号	8	协议号表示协议的类型（以下均为十六进制）。 TCP: 06 UDP: 11 ICMP: 01
	头部校验和	16	用于检查错误，现在已不使用
	发送方 IP 地址	32	网络包发送方的 IP 地址
	接收方 IP 地址	32	网络包接收方的 IP 地址
	可选字段	可变长度	除了上面的头部字段之外，还可以添加可选字段用于记录其他控制信息，但可选字段很少使用

IP 头部的“接收方 IP 地址”填写通信对象的
网卡根据客户端的路由表来进行判断

2.5.4 生成以太网用的MAC头部

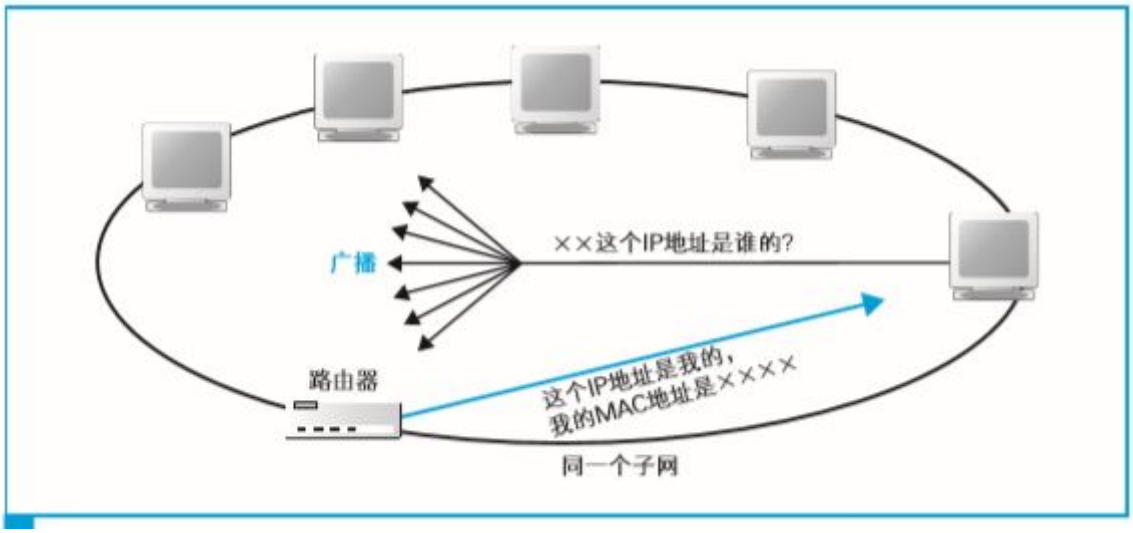
MAC头部字段：

字段名称		长度（比特）	含　义
MAC头部（14字节）	接收方 MAC 地址	48	网络包接收方的 MAC 地址，在局域网中使用这一地址来传输网络包
	发送方 MAC 地址	48	网络包发送方的 MAC 地址，接收方通过它来判断是谁发送了这个包
	以太类型	16	使用的协议类型。下面是一些常见的类型，一般在 TCP/IP 通信中只使用 0800 和 0806 这两种。 0000-05DC：IEEE 802.3 0800　：IP 协议 0806　：ARP 协议 86DD　IPv6

问题：接收方MAC地址如何确定?
使用IP地址查询MAC地址的操作。

2.5.5 通过ARP查询目标路由器的MAC地址

使用ARP缓存，若盖目标路由器的MAC地址不存在，则使用：
使用ARP协议：利用广播方法



缓存问题：当IP地址发生变化时，ARP缓存的内容就会和现实发生差异。
解决办法：，ARP缓存中的值 在经过一段时间后会被删除，一般这个时间在几分钟左右。

2.5.6 以太网的基本知识

当一台计算机发送信号时，信号就会通过网线流过整个网络，最终到达所有的设备。
与接收者地址匹配的 设备就接收这个包，其他的设备则丢弃这个包，
目前使用的交换式集线器：信号只会流到根据MAC地址指定的设备，而不会到达其他设备了。

2.5.7 将IP包转换成电或光信号发送出去。

将服务器的响应包从IP传递给TCP

若服务器返回的包的接收方IP地址和客户端网卡的地址一致，检查确认之后
我们就可以接收这个包了。若包是分片的，IP模块还需要将它们还原成原始的包。

若接收方IP地址不是自己的地址，不接受该包。此外IP模块会通过ICMP消息将错误告知发送方
ICMP规定了各种类型的消息：

消息	类型	含　义
Echo reply	0	响应 Echo 消息
Destination unreachable	3	出于某些原因包没有到达目的地而是被丢弃，则通过此消息通知发送方。可能的原因包括目标 IP 地址在路由表中不存在；目标端口号不存在对应的套接字；需要分片，但分片被禁用
Source quench	4	当发送的包数量超过路由器的转发能力时，超过的部分会被丢弃，这时会通过这一消息通知发送方。但是，并不是说遇到这种情况一定会发送这一消息。当路由器的性能不足时，可能连这条消息都不发送，就直接把多余的包丢弃了。当发送方收到这条消息时，必须降低发送速率
Redirect	5	当查询路由表后判断该包的入口和出口为同一个网络接口时，则表示这个包不需要该路由器转发，可以由发送方直接发送给下一个路由器。遇到这种情况时，路由器会发送这条消息，给出下一个路由器的 IP 地址，指示发送方直接发送过去
Echo	8	ping 命令发送的消息。收到这条消息的设备需返回一个 Echo reply 消息，以便确认通信对象是否存在
Time exceeded	11	由于超过了 IP 头部中的 TTL 字段表示的存活时间而被路由器丢弃，此时路由器会向发送方发送这条消息
Parameter problem	12	由于 IP 头部字段存在错误而被丢弃，此时会向发送方发送这条消息

2.6 UDP协议的收发操作

2.6.1 不需要重发的数据用UDP发送更高效

TCP：为了实现可靠性。并且为了实现高效的传输，要避免重发已经送达的包，而是只重发那些出错或者未送达的包。
UDP：数据很短，只需要一个包。考虑用UDP. 不可靠的。

UDP没有TCP的接收确认、窗口等机制，在收发数据之前也不需要交换控制信息，即不需要建立和断开连接的步骤。
知识单纯的发送包而已。若应用程序收不到对方的回复，会重新发送一遍数据。

2.6.2 音频和视频数据