应用层：

传输层：提供进程的通信

网络层：提供主机的通信

数据链路层：

物理层

# 第三章 运输层

运输层为不同主机上的应用进程提供直接通信服务。

运输层的传输数据的分组是报文段。报文段是将上层的应用报文，划分成较小的块。并加上运输层的控制信息，形成报文段。

(网络层：网络层的传输单元数数据报)

运输层和网络层提供的服务的区别：运输层提供的是运行在不同主机上的进程之间的逻辑通信。而网络层提供的是主机之间的逻辑通信。

TCP和UDP的区别：TCP提供的是面向连接的、可靠的服务。而UDP提供的是无连接、不可靠的服务。

IP协议提供的是一种尽力而为的交付服务。

## 3.2多路复用和解复用

应用层的多个进程会复用运输层实体。因此就涉及到多路复用和解复用。

每个运输层的报文段中具有一些标识字段。在接收端，运输层检查字段，并标识出接收套接字，进而将报文段定向到该套接字。。将运输层报文段中的数据交付给正确的套接字的过程叫做多路分解。

多路复用就是在源主机中，从不同的套接字中收集数据块，并且添加标志字段，封装成报文段。

所以我们在运行一个新的应用程序时，一定要为他分配一个端口号。

端口号可以实现运输层的多路复用和解复用

对于UDP来说，也需要在报文段中，增加源IP 源端口 目标IP和目标端口。对于UDP来说，虽然有四个参数，但是决定其交付进程的因素是目标IP和目标端口。具有不同源端口和源IP但是目标端口和ip相同的报文段，会被交付到同一个进程。

对于TCP来说，需要在报文段中，增加源IP 源端口 目标IP和目标端口。对于TCP来说，决定其交付进程的因素是源ip和源端口、目标ip和目标端口。两个不同源ip和端口但是目标ip和端口相同的报文段，会被定向到两个不同的套接字。

(那么这样说，一个端口可能对应多个进程咯)

## 3.3 无连接传输：UDP

UDP协议在IP提供的服务的基础上，并没有额外扩展很多功能。只是增加了一个复用分解功能和少量的查错检测。

UDP报文格式

源端口+目标端口

长度+校验值

应用层数据

UDP报文的头部数据花销小。

## 3.4 可靠数据传输原理

支持可靠数据传输原理的协议是可靠数据传输协议（RDT）,例如TCP是在不可靠的IP的基础上实现的可靠传输。

在下层的提供服务不可靠的前提下，实现数据的可靠传输。

可靠数据传递：要实现不出错、不冲突、不重复、不丢失等目标

### 3.4.1 停止等待协议

当rdt(可靠数据传输协议)的发送端接收到上层的发送数据的请求，向发送端发送一个报文段，并且对报文段进行编号**(加入序号)**，比如说data+0，接收端如果接收到报文段，并且**校验**正确，向发送端发送ACK0。(确认分组同样进行编号)。发送端在接收到ACK0后，发送第二个报文段data+1，重复上述过程，并且当收到ACK1后，继续发送data0(数据编号只有一个比特，就只有0和1)。

当发送端发送data1，接收端经过校验后，发现数据出错，则接收端会向发送端发送上一条正确的报文段的ACK0,当发送端接收到ACK0时，他知道自己刚才发送的是DATA1,就知道自己发送错误，则会进行重发data1。

如果报文段在发送过程中丢失，停止等待协议提供超时重发机制。需要一个定时器timeout规定最长等待时间，超时则重传。这个timeout的时间有时候定死，有时候动态计算出来。

### 3.4.2 流水线可靠数据传输协议

对于停止等待协议，对于异常处理比较完备。但是对于高带宽的信道，信道浪费比较严重。在发送一个报文段后，需要等待回执再进行下一步操作。

因此设计了流水线可靠数据传输协议。

流水线可靠数据传输协议：允许发送方发送多个分组而无需等待确认。

因此，对于运作在流水线可靠传输协议下的报文段，序号编码需要唯一，因此需要多个bit的长度存储序号。

协议的发送方和接收方必须能缓存多个分组。

当出现丢失、损坏、延时等情况时，如何实现差错恢复：回退N步和选择重传。

先补充一下滑动窗口协议。(slide windows)

发送方和接收方都需要缓冲区。发送方的缓冲区表示最多一次可以发送的报文段的个数。要发送的数据都存放在缓冲区中，以便差错检测。

接收方的缓冲区是为了解决接收方从发送方接收的数据速度和向上交付数据的速度的不匹配性，因此要设立缓冲区。

窗口和缓冲区的概念是不一样的。这个要记牢。

发送方的窗口：存放已经发送但是没收到确认的报文段所构成的空间。

一开始，未发送报文段时，发送窗口是空集。当发送一个报文段后，发送窗口的前端开始向前滑动一位，表示已发送一位但未确认。

接收窗口，存放可接收的报文段数据。

当接收窗口等于1时：

当接收窗口在“1”处，则不会接收其他序号的报文段。只能接收序号为“1”的报文段数据。当接收到需要为“1”的报文段数据后，接收窗口向前移动。(或者说向后移动)。当接收窗口处于“2”，并且此时收到序号为“3”的报文段，则先对序号为“3”的报文段进行丢弃，并且向发送端发送ACK1,表示自己需要序号为2的报文段。

当接收窗口大于1时：对于接收窗口内的报文段，都可以接收确认。假如接收窗口当前处在（ 0 1 2 3）时，收到序号为“3”的报文段，则变成（0 1 2 3）。当收到序号为“0”的报文段时，则变成（0 1 2 3），接收窗口向后移动，变成（1 2 3 4），序号为“0”的数据交付给上层。

### 3.4.3 回退N步协议

回退N步协议的接收窗口长度为1。由于接收窗口长度为1，所以对于回退N步协议，只能顺序接收报文段。

对于发送异常情况：回退N步协议会将发送窗口中的所有分组重新发送出去。

而接收窗口只有一个长度，因此他只会接收接收窗口内的分组。当接收方的数据乱序到达时，很多数据都会被丢弃。当引起超时重发时，发送方会将发送窗口中的所有分组重发(此时发送窗口中的第一个分组一定是没有被收到的，否则发送窗口就会向前移动)

GBN协议的发送端，只维护一个定时器，也就是应该最先收到ACK的分组的定时器。

### 3.4.4 选择性重发协议

选择性重发协议的接收窗口长度大于1。

对于发送异常情况，接收窗口收到了2 3 ，但是没有收到1，并且此时接收窗口已经发送了ACK2和ACK3，那么发送窗口就会把未接收到的分组重发。这就叫做选择性重发。

## 3.5 面向连接的TCP协议

所谓的面向连接，是指TCP协议要求，在应用进程在向另一个应用进程发送数据之前，这两个进程必须互相握手，确保连接建立。

TCP提供的是点到点的服务。

全双工，有流量控制

面向连接的可靠传输。

提供的是可靠的字节流服务。

MMS是什么？

### 3.5.2 TCP报文段结构

**TCP报文的组成**

TCP报文分为头部和数据段，数据段有最大字节数的限制（MMS 最大报文段）。而头部包含报文段的控制信息，包括目标端口和源端口，以及校验和字段。

**序号：**这个序号是指本报文段的数据部分占原始数据的第几个字节。比如说一个上层文件为5000个字节，MMS为1000，即数据段最大字节为1000.那么TCP就会为将这个文件分割成五个报文段，假设第一个报文段的序号x，那么第二个报文段的序号就为x+1000，第三个报文段的序号就为x+2000，以此类推。这个x一般不会是0或者1，而是数据传送双方约定好的一个值。

**确认号：**假设A向B发送数据，B接收到数据，并在向A发送的报文段的确认号中填入555，那就表示B已经收到了序号为554及之前的所有报文段。(这有点类似于回退N步，但是和选择重传似乎有些矛盾)

检验和和其他标志位等。(TCP的头部一般是20字节)

**往返时间的估计和超时**

往返时间：RTT 往返延时在100ms级别

如何计算估计定时器的定时大小？ 多次计算往返延迟，并计算平均值，计算偏差、方差等参数，最后可以综合得到一个定时时长。 往返延迟+四倍的标准差，作为定时器的定时时长。 还涉及到加权计算。

### 3.5.4 可靠数据传输

TCP协议提供可靠数据传输，比较像是SR和GBN的结合体。

TCP只设计一个定时器，虽然发出去很多报文段，但是只设计一个定时器。

解释一下TCP实现可靠传输的过程：

首先TCP支持流水线发送，一次性发送多个报文段，序号分别是 1 2 3 4 5，并开启一个定时器(这点很像GBN)，该定时器是针对序列号最小的报文段的(也就是最老的，)。当接收端接收到正确的报文段，并返回ack=2 ，表示此时接收方已经接收到1及之前的报文段，如果正确的话，会继续发送ack3 ….直到发送ack6 ，表示我此时已经收到 1 2 3 4 5。

如果出现数据丢失，定时器到达定时时间，那么发送端会发送最老的报文段。(这点很像SR) 。

TCP的接收端，发送给发送端的ACK,表示ACK之前的报文段都已经收到了，发送的ACK表示期待收到的报文段的数据。这种确认是一种累积确认。

所以说TCP是SR和GBN的结合体，有二者相似的地方。

**快速重传机制：**

当接收端已经接收到连续4个相同的ACKx,这就表示该ACKx后面的几个报文段已经被接收方接收，而该报文段出现丢失。那么即使此时没有到达重传时间，那么发送方也会对ACKx进行重传。(这部分概念理解不是很清楚)

### 3.5.5 TCP的流量控制

流量控制是指，发送方发送的数据太快，接收方来不及处理，那么就需要做流量控制。

在TCP的报文段中，有一个参数叫做接收窗口。当主机A向主机B发送报文段时，B向A发送TCP报文段作为ACK,那么此时会在接收窗口处填入空闲的的接收缓存大小。接收方根据该参数大小，判断是否需要暂停发送数据。

UDP不存在流量控制。

### 3.5.6 TCP的连接管理

TCP的连接需要经过三次握手过程。

TCP的连接，需要告知双方即将进行TCP的通信，并且为这次通信创建资源，包括缓冲区、控制变量的初始化，**确定双方的序号**。

两次握手无法保证双方都已经得知做好准备。

当发送方发送第一次握手，接收方发送第二次握手，说明接收方已经接受到第一次握手了，接收方做好准备。但第二次握手并不一定能到达发送方，那么此时我们的发送方不知道接收方已经做好准备。

如果是三次握手的话，第三次握手的发出，表示发送方已经知道接收方做好准备了。因此可以认为双方都做好准备了。第三次握手是否达到，无关紧要。

在握手的过程中，双方分别确认好各自的序号。（接收方的序号有什么用？）

三次握手过程：

这里认为是客户端向服务器请求建立连接。

客户端发送第一条报文段：包含SYN=1 表示连接建立请求。并且把自己的初始序号发出去：seq= client\_len

服务器接收到报文段后，向客户端发送第二条报文段，包含SYN=1,并且把自己的初始序号发出去seq= server\_len，并对服务器的seq进行确认，ack=client\_len+1

客户端接收到报文段后，向服务器发送第三条报文段，包含SYN=0,seq=client\_len+1,ack=server\_len+1.

四次挥手

对双向的连接，单独拆除。

客户端向服务器发送关闭连接请求。FIN ,服务器发送一个ACK ，表示OK.此时关闭了客户端到服务器的连接。

接下来服务器发送关闭连接请求。FIN,客户端发送一个ACK,表示OK，此时关闭了服务器到客户端的连接。

TCP的挥手过程并不完美，可能会出现一方断开连接，另一方维持连接的可能性。最后一次挥手是不可靠的。不能确认对方收到了关闭连接的请求。

## 3.6 拥塞控制原理

太多的数据需要网络传输，超过了网络的处理能力。和流量控制是不同的。

（拥塞控制原理还没掌握）

# 第四章 网络层

网络层的两个重要功能：转发和路由。

路由：从源端到目标端的路径规划。决定从路由器的输入端口到达的分组如何转发到输出端口。

转发：对于某一数据交换结点，实现从某一个端口入，从另一个端口出。(这个端口可不是端口号的那个端口)

网络层的数据单元：IP数据报。

路由表:是算出来的，根据路由表实现具体的转发过程。

路由器的主要作用：将数据报从入链路转发到出链路。

SDN:软件定义网络

## 4.3 路由器工作原理

路由器由以下几个结构组成：输入端口、输出端口、分组交换结构。

根据路由选择算法/协议(RIP OSPF BGP)生成路由表。

### 4.3.1 输入端口

要明确一个点，路由器的输入端口的数据，一定是从下层物理层到链路层，再解封装到网络层的。

物理层实现电信号或者电信号到数字信号的转换。

数据链路层实体，实现对帧头帧尾的检查，根据编码校验，检查一下有没有出错，并判断帧当中的目标mac和我的mac地址是否一致，决定是否要接收该帧。在接收完毕后，将帧当中的数据部分取出来(可能就是一个IP分组)，交给上层实体，一般也就是网络层实体。IP分组在网络层也需要排队，等待分组转发。根据路由表提供的转发路径，决定向哪个端口转发。

输入端口会存有路由表的备份，以便输入端口知道该往哪里转发分组

输入端口缓存：是为了匹配链路层上传的IP数据报的速度和从输入端口转发出去的速度不一致性。因此不同输入端口中的一些IP数据报，可能需要转发到同一个输出端口，因此部分IP数据报就需要等待，因此需要设计一个输入端口的缓存。但是因此也会存在输入缓冲区溢出的情况，IP数据报的丢失。(也有可能在输出队列中被丢弃)

### 4.3.2 交换结构

将分组从输入缓冲区，传输到合适的输出端口。交换速率一般是输入输出电路速度的几倍。(交换速率一定会大于输入输出缓存速率)

三种典型的交换结构：基于内存交换、基于总线交换、基于互联网络交换。

基于内存的交换：

当分组到达交换结构时，被CPU拷贝到内存当中，并查看其目标IP地址。CPU根据目标IP地址，进行转发表的查找，转发给合适的输出端口。

转发速率受到内存带宽限制，并且一次只能转发一条分组。

基于总线的交换：

把分组广播至所有的输出端口，但是会在分组中加入输入端口信息，因此只会被合适的输出端口接收。基于总线的路由器，分组转发速率要大于基于内存的路由器。

### 4.3.3 输出端口

输出端口同样需要一个输出缓存：从分组交换结构进来的分组速率大于朝链路层封装转发出去的速率，同样可能存在分组丢失的情况，因为溢出了。这种丢失的IP数据报的内容，也存在一定的调度算法。

并不是先存入输出缓存的分组就先被转发出去，而是存在一定的调度规则。

调度算法默认是先入先出算法，但是也存在其他调度算法。

## 4.4 网际协议(IP)

网络层的协议可不止IP协议，还包括路由选择协议(RIP OSPF BGA) IP协议和 ICMP协议(因特网控制报文协议)

### 4.4.1 数据报格式

·版本号

涉及到IPv4还是IPv6

·源IP和目标IP

·TTL

Time to live

表示该IP数据报的寿命，每经过一个路由器，该值都会-1，当该值为0时，丢弃IP数据报。

·头部校验和

对头部信息进行校验，对数据部分不校验

·数据

IP数据包的数据段，即可以是上层的报文段(TCP or UDP),也可以是ICMP报文。

IP数据报会被封装到链路层的数据段中，而链路层的帧的数据部分有最长字段限制，比如说以太网的帧的数据长度。因此上层往下层走的时候，可能需要拆分IP数据报。

这种拆除，给每一个IP数据报的段，增加头部信息。

### 4.4.2 IPv4编址

什么叫子网：

子网内的所有IP地址，前面若干位相同，具备相同的子网号。

在一个子网内部，不同IP的主机之间，传递分组不需要路由器。

但是我们也可以认为，一些子网之间，通过路由器进行分组转发，我们把这些子网+路由器，也可以看成是非严格意义上的子网。

一个子网，在路由器当中，聚集成一个表象。通过某一路由器，分组注入子网内部。

IP地址：标识主机和网络接口设备的点

路由器一般有多个IP地址

主机也可以有多个IP地址

一个网卡对应一个IP，通过网卡才能在网络中实现数据发送。

网络的分类：

A

B

C

D

特殊IP地址：

子网部分全为0，表示本网络

主机部分全为0，表示本主机

主机部分全为1，表示广播地址

内网地址：只在局部网络内有意义，用来区分不同的设备，不会作为公网地址。

子网掩码：子网掩码可以定义，多少位之前作为网络号，其后作为主机号。

网络号相同的网络，我们可以认为在一个子网当中(大概可以这样说)

路由表的转发过程：首先根据路由表，计算从哪个端口输出，并且我们此时需要带上下一条的路由器的信息。首先我们根据路由表，已经知道下一条的路由器的IP了，根据IP，和一定的协议(ARP)，得到目标路由器的mac地址，封装起来，就可以在链路层中实现到达目标路由器的链路层。

在不同的路由转发过程当中，主机号是没有用的，是根据网络号进行匹配的。但是在最后一跳中，主机号才有用。

如果一个目标IP没有和路由表中的任何一个匹配，那么一定会和默认网关匹配，默认网关可以认为是一个网络的出口。

**如何获取一个IP地址**

**·**网络管理员分配IP地址

·从服务器动态获取IP地址(DHCP)

动态主机配置协议：允许主机自动获取一个IP地址。

DHCP还允许一台主机得知他的子网掩码，以及第一跳路由地址等信息。需要DHCP server。

**网络地址转换**

当一个子网内部的ip都是内网ip时，子网内的主机又想和外部网络进行分组转发，就可以使用网络地址转换技术：NAT。在向外转发分组时，把源IP地址换成这个子网对外的公网ip，因此就可以对外进行分组转发。当收到外部网络的分组时，根据源ip地址，找到当前代替源ip地址的内网地址，并交付数据。这种映射关系需要被记录。

这种方法破坏了ip地址的封装性诶我感觉。

**IPv6**

IPv6有128位

### 4.4.3 因特网控制报文协议

ICMP ,ICMP协议被主机和路由器用来沟通彼此网络层的信息。最主要的用途是查错报告。

## 4.5 路由选择算法

路由信息的计算是以子网为基本单位的。

衡量路由选择算法的指标：跳转次数、延时、花费等

路由选择，实际上是规划从某一子网到另一子网的分组路径规划，路由选择算法有以下几种分类方式：

·全局算法和分散算法

根据可以分为全局式和分散式的算法。全局式算法，具有关于连通性和链路费用方面的完整信息，计算出从源路由器到目的路由器的完整路径规划。而分散式算法：每个节点有关于直接相连的链路费用知识后，就开始工作，逐渐和相邻节点交换信息，逐渐算出到目标路由的完整信息。

·静态算法和动态算法

静态算法的路由几乎不变化

而动态算法的路由会随着网络负载和拓扑发生变化而及时改变。目前一般都是动态算法。

### 4.5.1 链路状态路由选择算法

在获得整个网络拓扑和所有链路代价等信息，通过泛洪实现，再依据迪杰斯特拉算法，得到源路由器到目标路由器的最低费用路径。(前一半部分的网络拓扑和链路代价的获取，不是算法的考虑范围内)

最优路由算法。

链路状态路由选择算法的核心是迪杰斯特拉算法，该算法时间复杂度较高。

### 4.5.2 距离向量路由选择算法

和相接的路由器，交换路由信息，比较当前路由器前往目标路由器的最短路径应该从哪个路由器出发。进而分布式地找到到达目标的路径。最终迭代得到所有的路由。

感觉上是flord算法

那么什么时候更新路由呢，当当前路由到达相邻路由的代价发生变化，或者是从相邻路由得到的到达其他路由结点的代价发生变化时，应该对路由表进行更新。

链路状态路由选择算法和距离向量路由选择算法的比较：

链路状态路由选择算法传递的消息多

但是距离向量路由算法的收敛会较慢

## 4.6 因特网中的路由选择

路由可以分为两个层面，分别是内部网关协议和子网之间的路由协议

前面讲的都是算法，现在要讲具体的应用和协议。

子网内部使用的协议有RIP和OSPF,分别是基于距离矢量和链路状态路由算法的协议。

·RIP协议

RIP协议规定，分组最大跳转15次。每隔30s，相邻的两个路由器交换距离矢量，做更新。或者是相邻路由器发出距离矢量请求，也会进行交换。

RIP协议规模相对较小。

RIP协议的路由器维护的路由表当中，有以下几个内容：每一个表项都有目的子网、前往目的子网的下一跳路由器、和到达目的子网的跳数。

·OSFP协议

依据链路状态路由算法实现

### 4.6.3 子网之间的路由选择：BGP 边界网关协议

严格的来说，不能说是子网， 而应该说是自治区。

把大的互联网，分成各个小封地。也就是分为自治系统内部。

在自治系统内部，自治系统自己选择使用合适路由协议，但是在自治系统之间，采用统一的路由选择协议。

自治区内部有很多子网， 并通过一个默认网关，与其他自治区交换路由信息。

BGP分为外部BGP和内部BGP。网关会向外发出消息，可以把要发往AS内部的分组都给我，然后再由网关进行内部分组分发。

# 第五章 链路层

链路层的两种信道方式，第一种是广播信道。第二种是点对点的通信链路。链路层以结点为单位，之间的通信信道称作链路，

网络接口卡是链路层的主体，包含设备mac地址。

每个网卡，实现链路层和物理层的功能。

链路层的作用:一个结点，通过链路，将帧发送到另外一个结点。链路层可以实现可靠的服务。

链路层可以实现查错和纠正，这种纠正在链路层就实现，不需要由上层重发。

## 5.2 差错检测和纠错技术

### 5.2.1奇偶校验

加入一个标志位，奇校验满足所有的1之和为奇数；偶校验满足所有的1之和为偶数。

### 5.2.2 校验和方法

### 5.2.3 循环冗余检测

CRC校验

把发送的比特串，看成是系数为0和1的一个多项式。

发送和接收双方，约定了一个生成多项式序列，R次多项式序列。(长度为R+1)

在我们发送的序列之后，加上长度为r的EDC序列，保证该整体序列能被我们定义的生成多项式整除。

## 5.3 多路访问链路和协议

链路层有两类信道，分别是点对点和广播信道，

对于多路访问链路来说，要解决访问冲突的问题。在链路中，多结点同时发出访问请求，会发生碰撞

多路访问链路需要遵守MAC协议(多路访问控制协议)

**多路访问控制协议可以分为信道划分协议、随机接入协议、和轮流协议。**

### 5.3.1 信道划分协议

时分复用、频分复用和码分多址(CDMA) Code division multiple access

### 5.3.2 随机接入访问

载波侦听多路访问(CSMA)

当某一结点在发送帧之前，都会进行侦听，当确定当前信道没有其他结点发送帧后，才会进行帧的发送。理论上，这样可以避免帧的冲突，但是由于存在信号传播时延，所以不能完全避免冲突

CSMA:在发送之前侦听。

具有碰撞检测的载波侦听多路访问(CSMA/CD)

在发送过程当中， 同样进行侦听，当检测到冲突时，及时终止传输，提高信道利用率。中止传输后，适配器等待一个随机时间量，然后进行侦听和发送过程。

CSMA/CA

具有冲突避免的载波侦听多路访问(CSMA/CA)

在无线介质中，冲突检测不好用，冲突避免好用一些。 但是CSMA/CA 无法完全避免冲突，因此要选择ACK

以太网中不需要ACK,而无线网需要ACK

## 5.4 交换局域网

·ARP协议

实现IP地址到MAC地址的转换。MAC地址的作用，在一个物理网络内部，标识所有结点，实现帧的分发。

MAC地址的匹配，是通过硬件实现的

在网络层当中，在路由中实现跳转时，接收端在接收到IP数据报分组后，会把目标路由或者说是目标结点的MAC地址封装进去。

特殊的MAC地址：48位全为1，表示这是一个广播地址。链路上的结点都会收到这条帧。

为什么在一个子网内部，IP数据报是通过一跳可达呢。

### 5.4.2 以太网

以太网是目前最主流的LAN技术。

集线器：是一种物理层设备，他作用于各个bit而不是作用于帧。当表示一个0或者1到达接口时，集线器重新生成并且放大该bit，也就是增强能量，并且将该bit向所有其他接口传输。

集线器的多个端口，只能有一个端口在发送数据，不允许多个端口同时发送数据。

以太网的链路可以选择同轴电缆，是一种总线型的链路，是广播信道。但是会出现“一旦电缆的一部分损坏”整条信道出现问题。

因此出现了集线器型的链路，通过集线器转发bit。使得各条信道相对独立。集线器会将bit向所有其他接口传输。

后来升级成交换机：交换机在内部实现bit的定位转发。

交换机的不同结点，可以同时进行发送和接收，并不是统一时刻只能进行两个结点之间的发送和接收。（这是和集线器的最大区别）

根据帧当中的目标mac地址，判断是否接收该数据帧

以太网的物理层媒介，可以是光纤，同轴电缆和双绞线。

# 存疑：

对socket怎么理解。(套接字)

虚电路和数据报

内部路由选择协议和外部路由选择协议