目录

计算机网络详解

本章主要内容：

1. 因特网的边缘部分和核心部分，分组交换的概念

2. 计算机网络的性能指标

3. 计算机网络的体系结构（协议和服务的概念）

计算机网络的重要功能

✧ 连通性：彼此连通，交换信息（数据的传递）

✧ 共享：信息共享，软硬件共享（打印机共享、某一个大型的软件连接远程使用的共享）

网络、互连网络、因特网

网络：用交换机或集线器连接计算机就组成一个网络，同一个网络的计算机在同一个网段。

互连网络：通过路由器连接的多个网络，就是互连网络，路由器负责在不同的网段转发数据包。

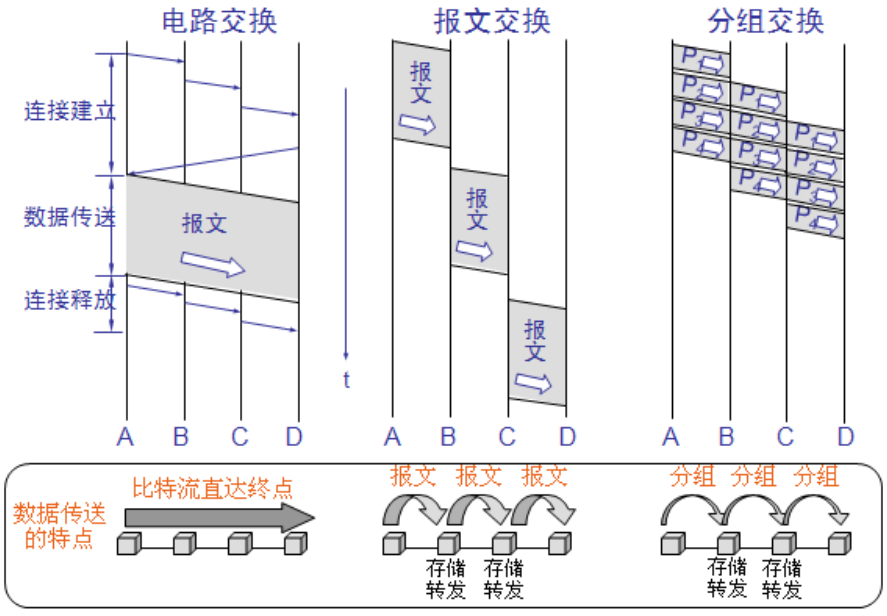
因特网：全球最大的互连网络（Internet）

因特网的组成部分

边缘部分：通信的计算机，计算机通信的方式（客户端服务端方式 C/S 方式、对等方式-迅雷下载相关的 BT 种子）

核心部分：网络（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）

三种数据交换方式：



1． 电路交换：计算机终端进行通信时，一方发起呼叫，独占一条物理线路，通信完成后，释放线路。

特点：实时性强，时延小，线路利用率低，通信效率低，不同类型终端用户之间不能通信

电路交换比较适用于信息量大、长报文，经常使用的固定用户之间的通信。

2． 报文交换：将用户的报文存储在交换机的存储器中，当所需要的输出电路空闲时，再将该报文发向接收交换机或终端，它以“存储—转发”方式在网内传输数据。

特点：中继电路利用率高，多个用户可以同时在一条线路上传送，可实现不同速率、不同规程的终端间互通，网络传输时延大，且占用大量的交换机内存和外存，实时性弱

报文交换适用于传输的报文较短、实时性要求较低的网络用户之间的通信

3．分组交换：将报文按照一定长度分割成许多小段的数据—分组，每个分组标识后，在一条物理线路上采用动态复用的技术，同时传送多个数据分组，交换机负责暂存分组并进行转发，接收端根据分组的标识整合成完成的报文

特点：高效、灵活、迅速、可靠

练习题目：试在下列条件下比较电路交换和分组交换。要传送的报文共 x（bit），从源站到目的站共经过 k 段链路，每段链路的传播时延为 d（s），数据率为 b（bit/s）。在电路交换时电路的建立时间为 s（s）。在分组交换时分组长度为 p（bit），且各结点的排队等待时间可忽略不计。问在怎样的条件下，分组交换的时延比电路交换的要小？

对于电路交换的时延：s + x/b + kd

对于分组交换的时延：x/b + (k-1)p/b + kd(这是最后一个分组 k-1 次的转发)

计算机网络的类别

广域网 WAN　　城域网 MAN　　局域网 LAN　　个人区域网 PAN

计算机网络的性能指标

1．速率：每秒传输多少比特(bit) 单位是 b/s(bps) kb/s Mb/s Gb/s

2．带宽：用来表示网络的通信线路传输数据的能力

3．吞吐量：在单位时间内通过某个网络或接口的数据量，包括全部上传和下载的流量

4．时延：指数据（一个数据包或 bit）从网络的一端传送到另一端所需要的时间

时延 = 传播时延（信道长度/电磁波在信道上的传播速率） + 处理时延 + 排队时延 + 发

送时延（数据帧长度/发送速率）

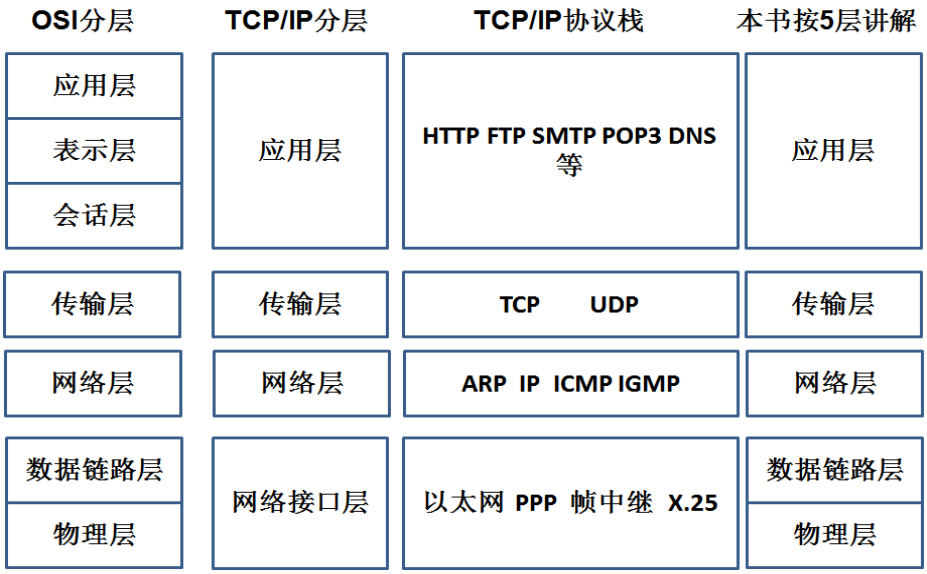
5．时延带宽积：把链路上的传播时延和带宽相乘，就会得到时延带宽积

6．往返时间：从发送方发送数据开始，到发送方接收到来自接收方的确认（发送方收到后立即发送确认），总共经历的时间

7．利用率：网络有百分之几的时间是被利用的（有数据通过），没有数据通过的网络利用率为零

U 是网络利用率，D 表示网络当前时延，D_0 表示网络空闲时的时延 $D=\frac{D_0}{1-U}$

计算机网络体系结构



- 物理层：通过媒介传输比特,确定机械及电气规范（比特 Bit）
- 数据链路层：将比特组装成帧和点到点的传递（帧 Frame）
- 网络层：负责数据包从源到宿的传递和网际互连（包 PackeT）
- 传输层：提供端到端的可靠报文传递和错误恢复（段 Segment）
- 会话层：建立、管理和终止会话（会话协议数据单元 SPDU）
- 表示层：对数据进行翻译、加密和压缩（表示协议数据单元 PPDU）
- 应用层：允许访问 OSI 环境的手段（应用协议数据单元 APDU）

这里面需要讨论的几个问题：

1．分层的好处是什么？

✧ 各层之间是独立的(每一层的变化并不会影响其它层,每一层并不关心其它层如何实现）

✧ 灵活性好

- ✧ 结构上可分割开
- ✧ 易于实现和维护
- ✧ 能够促进标准化工作

2．实体、服务、协议、服务访问点？

- ✧ 实体：发送和接受信息的硬件或软件的进程
- ✧ 服务：下层向上层提供服务，上层需要使用下层提供的服务来实现本层的功能（垂直的）
- ✧ 协议：控制两个进程的规则（两个进程是对等的，所以协议是水平的）

  包括语法（数据与控制信息的结构或格式）、语义（每一个结构或格式的具体含义）、同步（事件实现顺序的详细说明）三个部分
- ✧ 服务访问点：相邻的两层实体交换信息的地方

物理层

本章主要内容：

1． 物理层的功能

2． 数据通信的基本知识以及计算问题

3． 几种常用的信道复用技术

物理层的基本概念

物理层考虑的是怎样在连接计算机的传输媒体上传输比特流，主要确定与传输媒体的接口有关的一些特性：

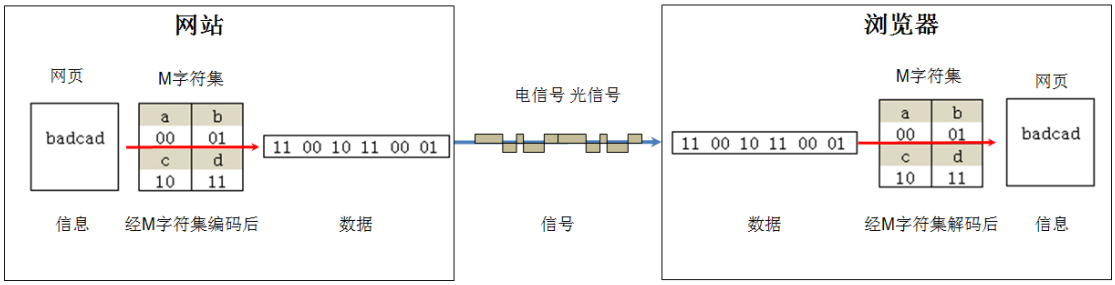
- ✧ 机械特性：指明接口所用的接线器的形状和尺寸，引线数目和排列，固定和锁定装置等等。平时常见的各种规格的接插件都有严格的标准化的规定。
- ✧ 电气特性：指明在接口电缆的各条线上出现的电压的范围（-5V 到+5V）
- ✧ 功能特性：指明某条线上出现的某一电平的电压表示何种意义（规定-5V 表示 0，+5 表示 1）
- ✧ 过程特性：指明对于不同功能的各种可能事件的出现顺序

数据通信的基本知识

数据通信常用术语：

- ✧ 信息（文字、图像、视频和音频等）
- ✧ 数据（将信息编码后的结果）

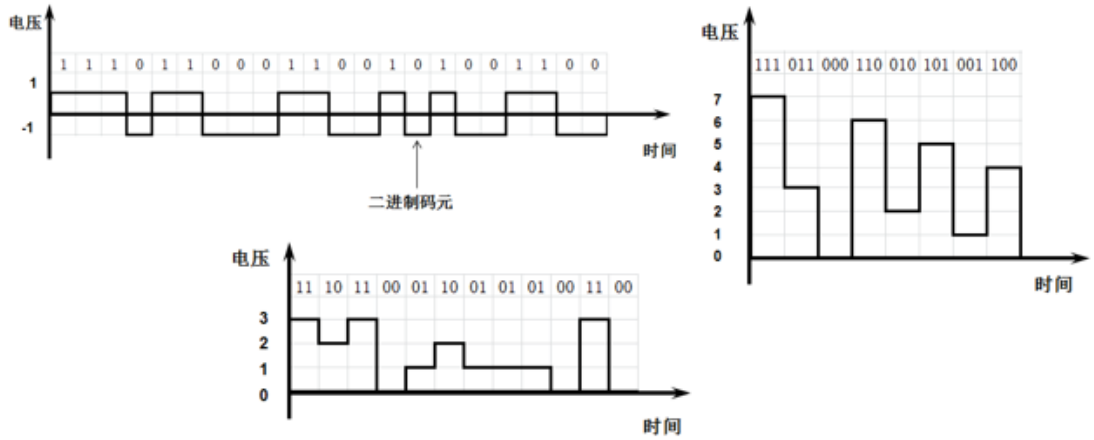
✧ 信号（数据在通信线路上传递需要变成电信号或光信号）



模拟信号——波形是一条连续的线（模拟信号如果在传输过程中如果出现信号干扰发生，波形发生变形，很难纠正）

数字信号——波形是离散分布的（数字信号波形失真可以进行修复）

码元：数字信号的基本波形就代表一个码元



上图可以看出一个码元如果承载 1 个二进制位，就有 2 种波形，承载 2 个二进制位，就有 4 种波形，承载 3 个二进制位，就有 8 种波形

信道和调制

信道是信息传输的通道，信息进行传输时所经过的一条通路

在信道上，一端是发送端，另一端是接收端，两端的通信方式有三种：

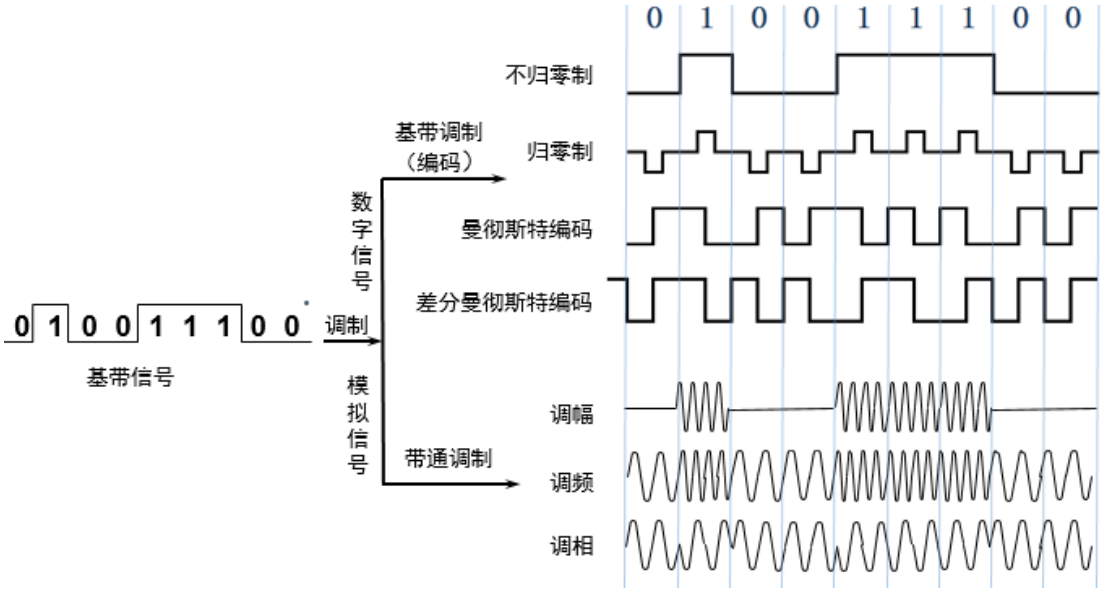
✧ 单工通信：信号只能从乙方传输，比如收音机接收，电视广播

✧ 半双工通信：可以双向传送，但必须交替进行，A 端说话 B 端接听，B 端说话 A 端接听，不能同时说和听

✧ 全双工通信：即信号可以同时双向传送。比如我们手机打电话，听和说可以同时进行

调制技术分为基带调制（对基带信号的波形进行变换，使它能够与信道特性相适应）和带通调制（使用载波（carrier）进行调制，把基带信号的频率范围搬移到较高的频段以便在信道中传输）

其中基带调制是数字信号转化为另一种数字信号，带通调制是数字信号转化为模拟信号



说明：

- ✧ 不归 0 制编码方式可能导致在传输过程中码元紊乱，没有时钟频率
- ✧ 归 0 制编码方式带有时钟频率，可是会影响发送速率，消耗了传输时间；
- ✧ 曼彻斯特编码方式表示位周期中心的向上跳变代表 0，位周期中心向下跳变的代表 1
- ✧ 差分曼彻斯特编码方式表示有跳变的表示 0，无跳变的表示 1，在这个过程中，编码本身有一个内部跳变。

信道极限容量：

奈氏准则：在任何信道中，码元传输的速率是有上限的，否则就会出现码间串扰的问题，使接收端对码元的判决（即识别）成为不可能

理想低通信道的最高码元传输速率=**2WBaud**

- ✧ W 是理想低通信道的带宽，单位为 HZ
- ✧ Baud 是波特，是码元传输速率的单位

香农公式：信道的极限信息传输速率（适用于模拟信号和数字信号）

有噪声的信道的极限信息传输速率 C：$C=W\log_2(1+S/N)\ (b/s)$

在式子中，W 为信道的带宽（以 HZ 为单位）S 为信道内所传信号的平均功率 N 为信道内部的高斯噪声功率

信噪比： 所谓信噪比就是信号的平均功率和噪声的平均功率之比，常记为 S/N，并用分贝

(dB)作为度量单位。即：

$$信噪比(dB)=10\log_{10}(S/N)(dB)$$

例如，当 S/N=10 时，信噪比为 10dB，而当 S/N=1000 时，信噪比为 30dB

传输媒体

导向传输媒体：双绞线、光纤、同轴电缆、光缆

非导向传输媒体：无线局域网、短波通信、微波通信、无线电频段

信道复用技术

频分复用：适用于模拟信号，ADSL 拨号上网就是利用了频分复用技术

频分复用的所有用户在同样的时间占用不同的频带宽度

时分复用：将时间划分为一段段等长的时分复用帧（TDM 帧）

时分复用的所有用户是在不同的时间占用同样的频带宽度

波分复用：为了提高光纤的传输信号的速率，也可以进行频分复用，由于光载波的频率很高，因此习惯上用波长而不用频率来表示所使用的光载波

码分复用：各用户使用经过特殊挑选的不同码型，因此彼此不会造成干扰

练习题目：假如基站发送了码片序列（0 0 -2 +2 0 -2 0 +2）。

A 手机的码片序列为（-1 -1 -1 +1 +1 -1 +1 +1）

B 手机码片序列为（-1 -1 +1 -1 +1 +1 +1 -1）

C 手机码片序列为（-1 +1 -1 +1 +1 +1 -1 -1）

问这三个手机，分别收到了什么信号？

A:（0+0+2+2+0+2+0+2）/8=1,证明有自己的数据为 1，

B：0+0+(-2)+(-2)+0+(-2)+0+(-2)/8=-1，证明有自己的数据为 0，

C：得到的结果为 0，证明没有自己的数据

数据链路层

本章主要内容：

1. 数据链路层的三个基本问题
2. 点对点协议以及以太网协议
3. 集线器、网桥、交换机、网卡的作用以及使用场景

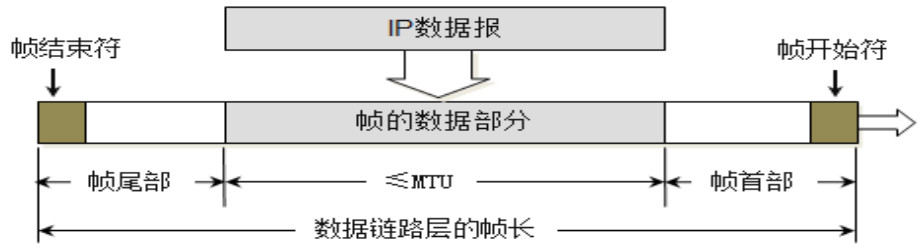
数据链路层的三个基本问题

封装成帧

封装成帧：将网络层的 IP 数据报的前后分别添加首部和尾部，这样就构成了一个帧

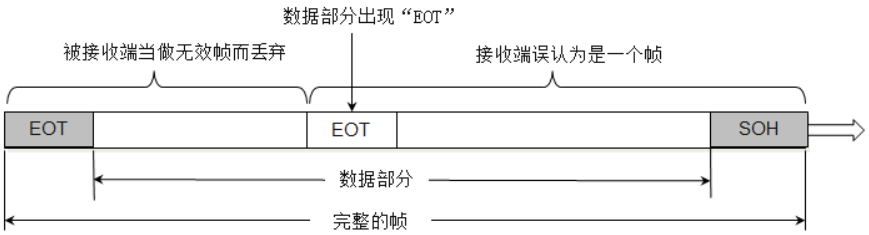
帧定界符：帧的首部和尾部有帧开始符和帧结束符

每一种数据链路层协议都规定了所能够传送的帧的数据部分长度的上限--即最大传输单元 MTU，以太网的 MTU 为 1500 个字节



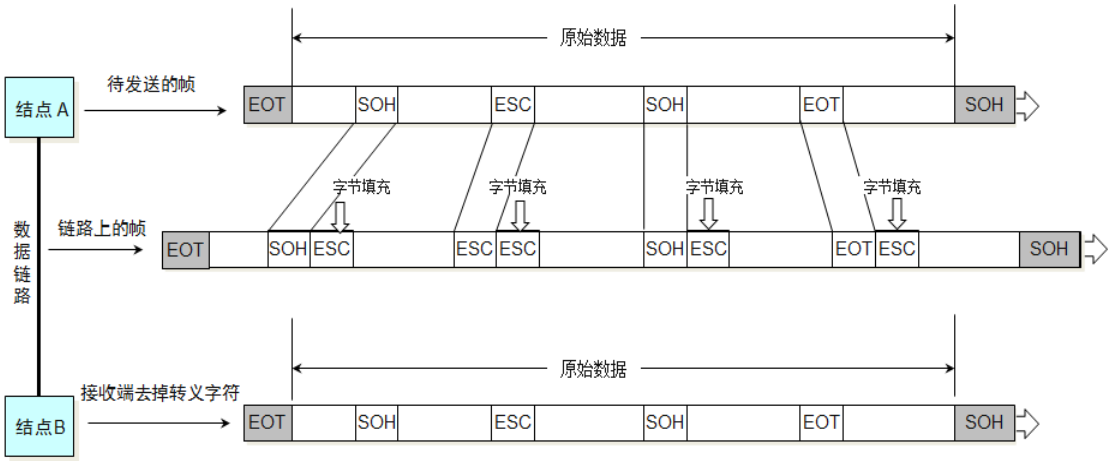
在 ASCII 字符代码表中，有两个字符专门用来做帧定界符，代码 SOH（Start Of Header）作为帧开始定界符，代码 EOT（End Of Transmission）作为帧结束定界符。

透明传输



解决透明传输问题：字节填充 或 字符填充(添加转义字符)

接收端的数据链路层在将数据送往网络层之前删除插入的转义字符。

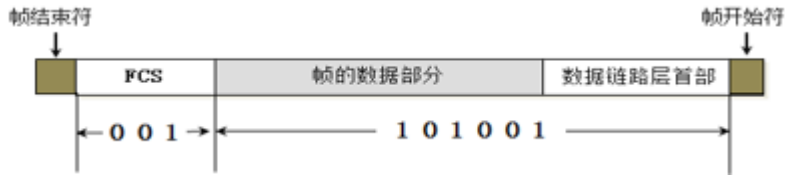


差错检测

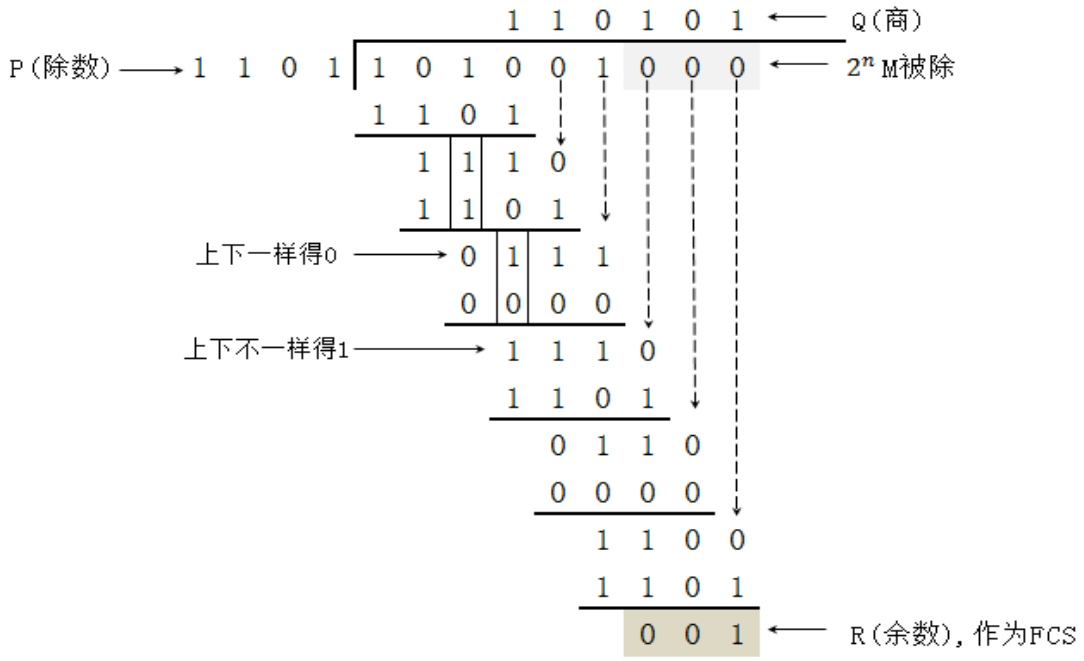
比特在传输过程中可能会产生差错：1 可能会变成 0，而 0 也可能变成 1

为了保证数据传输的可靠性，必须采用各种差错检测措施，目前在数据链路层广泛使用了循

环冗余检验 CRC(Cyclic Redundancy Check）的差错检验技术。



在数据后面添加帧检测序列 FCS（冗余码）



在被除数的后面添加的 0 的个数是除数的个数减去 1，得到的余数就是为了进行检测为添加的冗余码（帧检验序列）FCS

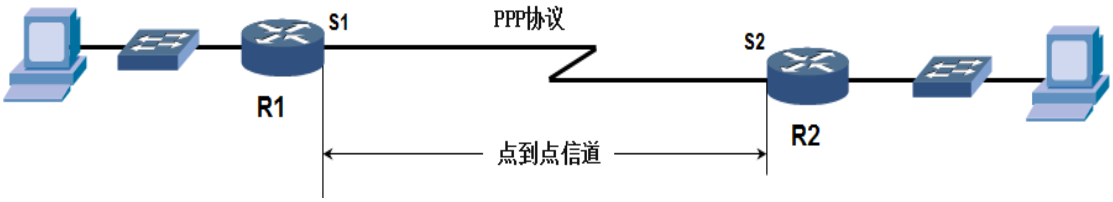
将原来的被除数+FCS 得到新的被除数，用新得到的被除数除以除数，得 R

(1) 若得出的余数 $R = 0$，则判定这个帧没有差错，就接受(accept)。

(2) 若余数 $R \neq 0$，则判定这个帧有差错，就丢弃。

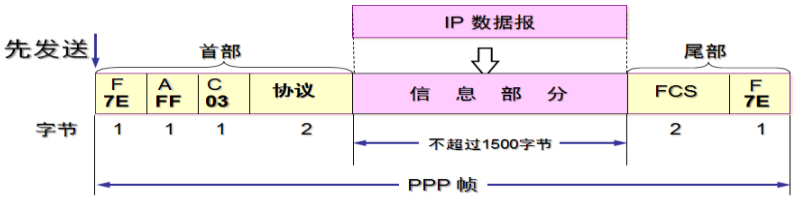
点到点的协议（PPP 协议）

点到点信道是指的一条链路上就一个发送端和接收端的信道，通常用在广域网链路



PPP 协议的特点：简单（不提供可靠传输）、封装成帧、透明传输、差错检测、支持多种网络层协议、支持多种类型的链路、最大传输单元

PPP 协议的帧格式：



标志字段 F：表示一个帧的开始和结束（PPP 帧的定界符）

地址字段 A：0xFF　　控制字段 C：0x03

在 PPP 协议上如何解决透明传输的问题？

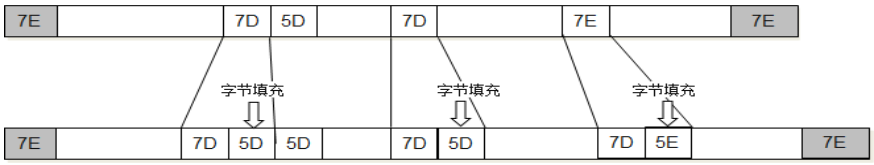
1．字节填充：以字节为单位–通过电压时钟同步

在异步传输的链路上，数据传输以字节为单位，PPP 帧的转义符定义为 0x7D，并使用字节填充。

✧ 把信息字段中出现的每一个 0x7E 字节转变成为 2 字节序列（0x7D，0x5E）。

✧ 若信息字段中出现一个 0x7D 的字节(即出现了和转义字符一样的比特组合),则把 0x7D 转变成为 2 字节序列（0x7D，0x5D）。

PPP帧字节填充



2．字符填充：0 比特填充–同步传输（以发送与接收端以帧为单位–时钟同步）

PPP帧零比特填充

练习题目：

【1】一个 PPP 帧的数据部分（用十六进制写出）是 7D 5E FE 27 7D 5D 7D 5D 65 7D 5E。试问真正的数据是什么（用十六进制写出）？

答：7E FE 27 7D 7D 65 7E。

【2】PPP 协议使用同步传输技术传送比特串 0110111111111100。试问经过零比特填充后变成怎样的比特串？若接收端收到的 PPP 帧的数据部分是 000111011111010111110110，问删除发送端加入的零比特后变成怎样的比特串？

答：第一个比特串：经过零比特填充后编程 011011111011111000（加上下划线的 0 是填充的）另一个比特串：删除发送端加入的零比特后变成 000111011111-11111-110（连字符表示删除了 0）。

以太网协议（CSMA/CD 协议）

以太网标准：

试说明 10BASE-T 中的“10”、“BASE”和“T”所代表的意思。

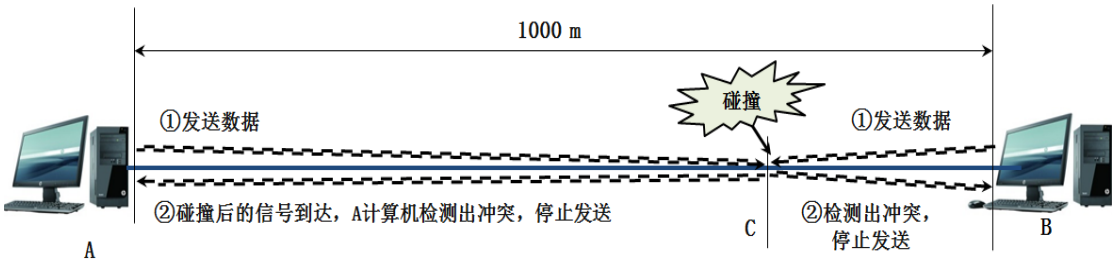
答：10BASE-T：“10”表示数据率为 10Mb/s，“BASE”表示电缆上的信号是基带信号，“T”表示使用双绞线的最大长度是 500m。

CSMA/CD 协议：

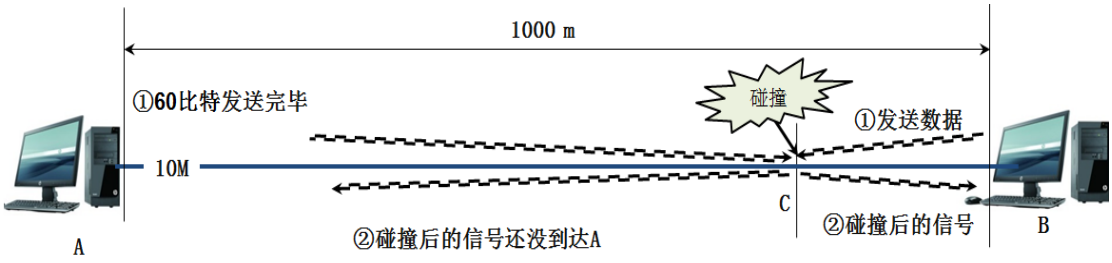
- “多点接入”表示许多计算机以多点接入的方式连接在一根总线上。
- “载波监听”是指每一个站在发送数据之前先要检测一下总线上是否有其他计算机在发送数据，如果有，则暂时不要发送数据，以免发生碰撞。
- “碰撞检测”就是计算机边发送数据边检测信道上的信号电压大小。

以太网最短帧：

A 计算机发送的信号和 B 计算机发送的信号在链路 C 处发生碰撞，碰撞后的信号相互叠加，在总线上电压变化幅度将会增加，发送方检测到电压变化超过一定的门限值时，就认为发生冲突，这就是**冲突检测**



如果以太网的帧的数据长度太短的话，不能检测到自己发的帧是否发生了冲突



以太网设计最大端到端长度为 5km（实际上的以太网覆盖范围远远没有这么大），单程传播时延为大约为 25.6μs，**往返**传播时延为 51.2μs，10M 标准以太网最小帧为：

10Mb/s(带宽)×51.2μs(传播时间) $=10^7$b/s×51.2×10^{-6}s=512b(64 个字节)

练习题目：假定 1km 长的 CSMA/CD 网络的数据率为 1Gb/s，设信号在网络上的传播速率为 200000km/s，求能够使用此协议的最短帧长

以太网最短帧 = 带宽 x 往返时间

答：最短帧为：1x10^9 x (1÷2x10^5 x 2) = 10000bit(1250 个字节)

冲突解决的办法：退避算法（确保了以太网的最短帧，但是当遇到冲突的时候，如何解决？）

计算机要想知道发送的帧在链路上是否发生碰撞必须等待 **2τ，2τ 称为争用期**

以太网使用**截断二进制指数退避算法**来解决碰撞问题

（1）确定基本退避时间，它就是争用期 2τ。以太网把争用期定为 51.2μs

（2）从离散的整数集合 [0，1，…，(2^k-1)] 中随机取出一个数，记为 r。重传应推后的时间就是 r 倍的争用期。上面的参数 k 按下面的公式计算：

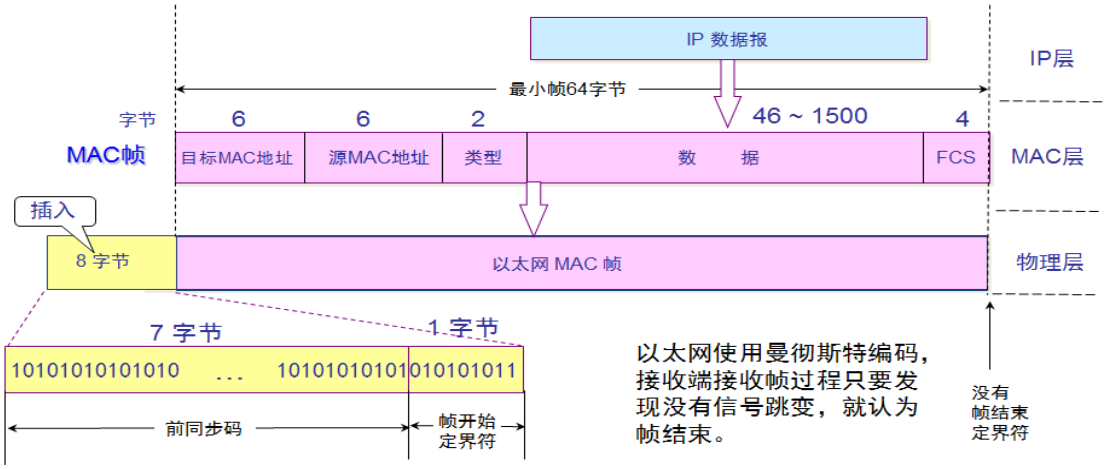
k=Min[重传次数，10]

可见当重传次数不超过 10 时，参数 k 等于重传次数；但当重传次数超过 10 时，k 就不再增大而一直等于 10

（3）当重传达 16 次仍不能成功时（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

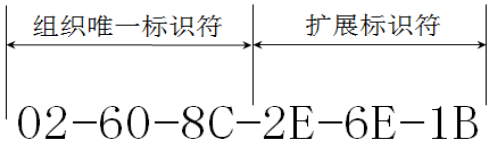
以太网帧格式：

以太网帧格式



- 前两个字段为目的 MAC 地址和源 MAC 地址字段
- 第三个字段是类型字段，用来标志上一层使用的是什么协议，以便把收到的 MAC 帧的数据上交给上一层的这个协议
- 第四个字段是数据字段，其长度在 46 到 1500 字节之间。当数据字段的长度小于 46 字节时，数据链路层就会在数据字段的后面加入一个整数字节的填充字段，以保证以太网的 MAC 帧长不小于 64 字节，接收端还必须能够将添加的字节去掉。
- 最后一个字段是 4 字节的帧检验序列 FCS（使用 CRC 检验）

MAC 地址有 6 个字节，这种 6 字节的 MAC 地址已被固化在网卡的 ROM 中。因此，MAC 地址也叫作硬件地址（hardware address）或物理地址。当这块网卡插入（或嵌入）到某台计算机后，网卡上的 MAC 地址就成为这台计算机的 MAC 地址了



（1）单播（unicast）帧（一对一），即收到的帧的 MAC 地址与本站的硬件地址相同

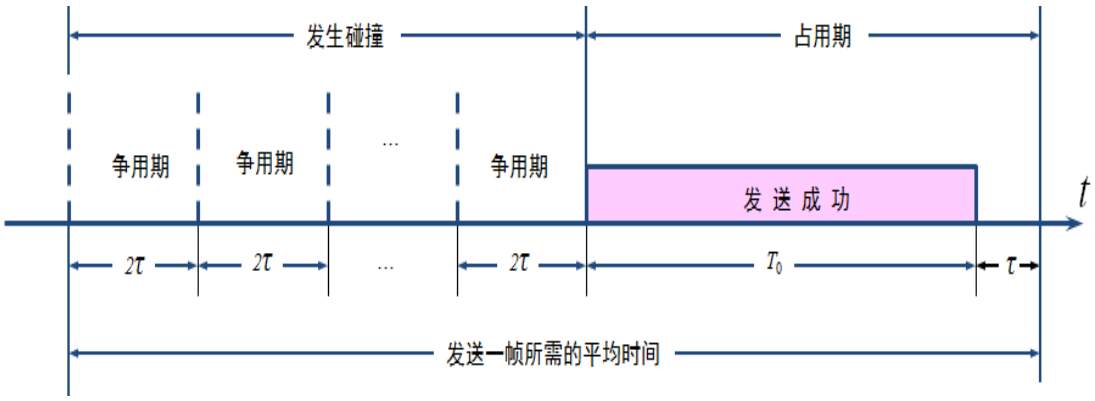
（2）广播（broadcast）帧（一对全体），即发送给本局域网上所有站点的帧（全 1 地址）

（3）多播（multicast）帧（一对多），即发送给本局域网上一部分站点的帧

以太网的信道利用率：

利用率是指的发送数据的时间占整个时间的比例。

如图所示，平均发送一帧所需要的时间，经历了 n 倍争用期 **2τ**，T_0 为发送该帧所需时间，**τ 为该帧传播时延。**



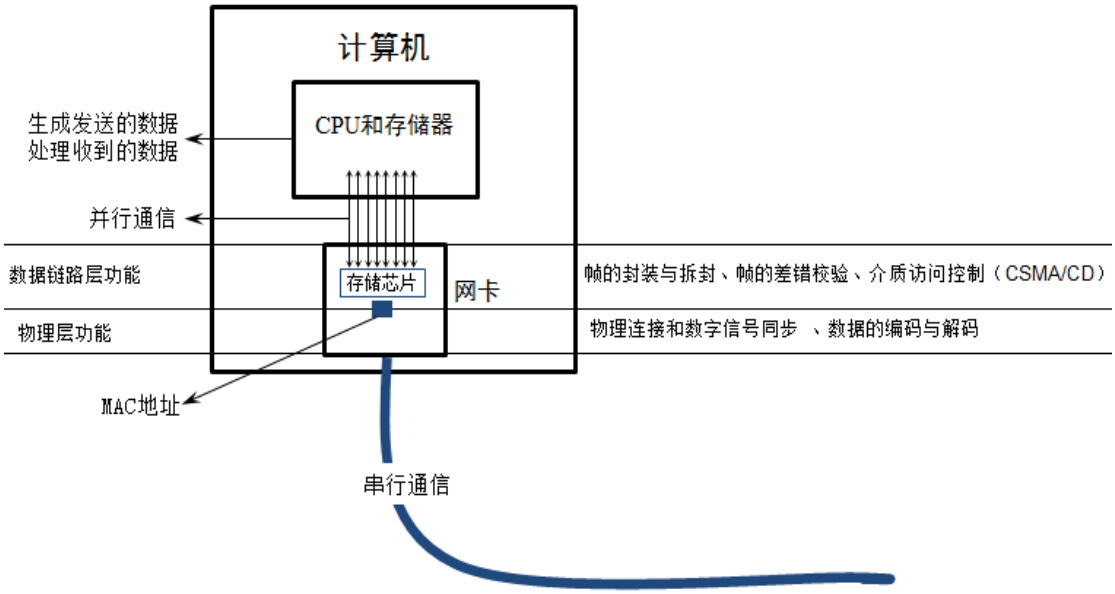
有冲突时的信道利用率：$S=\frac{T_0}{n2\tau+T_0+\tau}$

$$S_{\max}=\frac{T_0}{T_0+\tau}=\frac{1}{1+\frac{\tau}{T_0}}$$

τ 值和以太网连线的长度有关，这就意味着 τ 值要小，以太网网线的长度就不能太长。带宽一定的情况下 T0 和帧的长度有关，这就意味着，以太网的帧不能太短

网卡、集线器、网桥、交换机

网卡：工作在物理层和数据链路层



集线器：工作在物理层

集线器和计算机进行连接是通过网线（RJ45 接口），每个站到集线器的距离不超过 100m

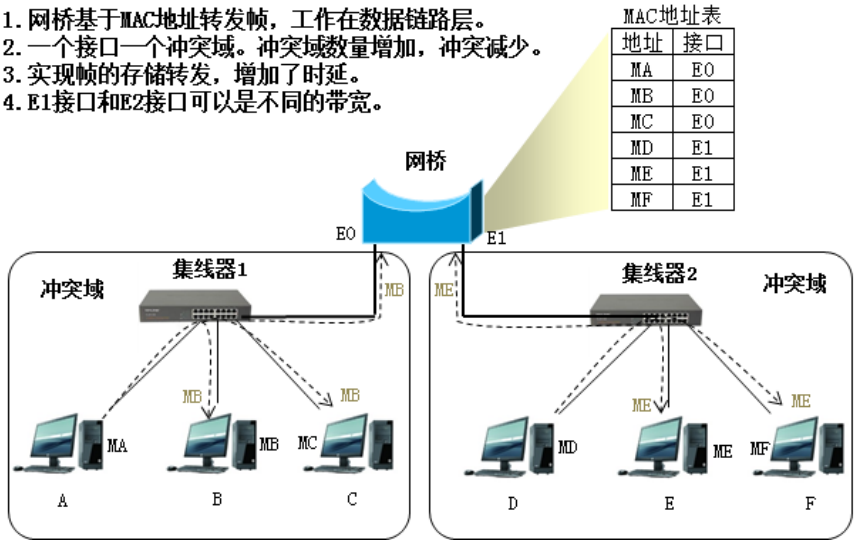
如何使用集线器实现计算机的数量和距离的扩展？

1．可以将多个集线器连接在一起形成一个更大的以太网，这不仅可以扩以太网中计算机的数量，还可以扩展以太网的覆盖范围。

2.可以使用光纤将两个集线器连接起来，集线器之间通过光纤连接，可以将相距几千米的集线器连接起来，需要通过光电转换器，实现光信号和电信号的相互转换。

使用集线器连接的计算机组成了一个大的冲突域

网桥：工作在数据链路层



网桥自动构造 Mac 地址表，并且是临时构造的（需要不断学习和更新）

网桥接入以太网时，MAC 地址表示空的，网桥会在计算机通信过程中自动构建 MAC 地址表

（1）自学习

网桥的接口收到一个帧，就要检查 MAC 地址表中与收到的帧源 MAC 地址有无匹配的项目，如果没有，就在 MAC 地表中添加该接口和该帧的源 MAC 地址对应关系以及进入接口的时间，如果有，则把原有的项目进行更新

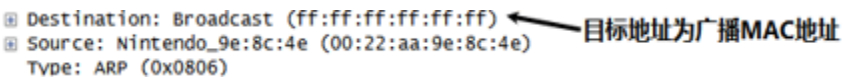
（2）转发帧

网桥接口收到一个帧，就检查 MAC 地址表中有没有该帧目标 MAC 地址对应端口，如果有，就会将该帧转发到对应的端口，如果没有，则将该帧转发到全部端口（接收端口除外）

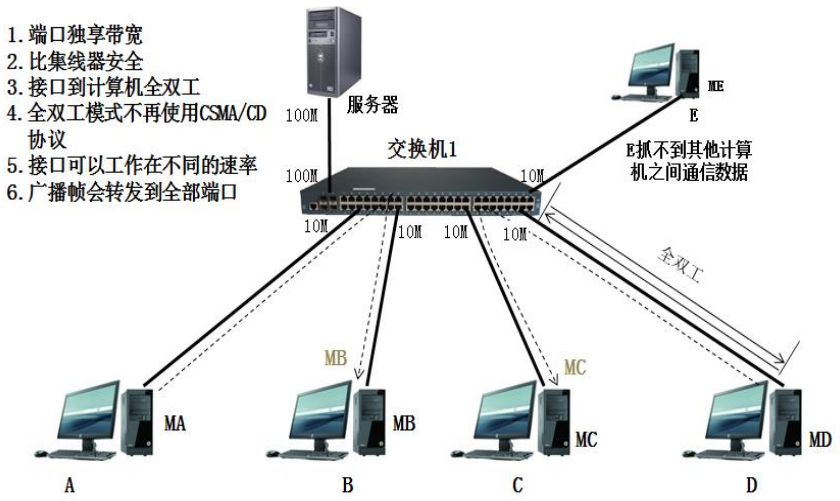
交换机：工作在数据链路层

网桥接口增多，网桥的接口就直接连接计算机了，网桥就发展成现在的**交换机**

交换机中的计算机如果发的是广播包，那么会转发广播帧到交换机连接的计算机的所有端口



交换机形成了一个广播域，路由器是隔绝广播的



IP 地址和子网划分

本章主要内容：

1. IP 地址、IP 地址划分类
2. 公网地址、私网地址、NAT 技术
3. 子网划分、合并网段

理解 IP 地址

IP 地址的组成：一部分为**网络标识**，一部分为**主机标识**，同一网段的计算机网络部分相同，路由器连接不同网段，负责不同网段之间的数据转发，交换机连接的是同一网段的计算机

IP 地址的格式：IP 地址用 32 位二进制来表示，也就是 32 比特，换算成字节，就是 4 个字节，这些位通常被分割为 4 个部分，每一部分 8 位二进制，中间使用符号 . 分开，IP 地址经常被写成十进制的形式，例如：172.16.30.56 IP 地址的这种表示法叫做“点分十进制表示法”

子网掩码的作用：子网掩码就是将某个 IP 地址划分成网络地址和主机地址两部分，区分哪一部分是网络地址，哪一部分是主机地址

判断 IP 地址所属于的网段：计算机用自己的 IP 地址与子网掩码进行与运算，得到网络号

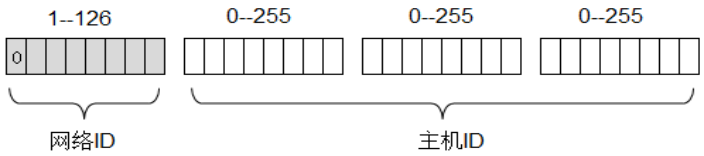
网关：网关就是到其他网段的出口，也就是路由器接口地址。路由器接口使用的地址可以是本网段中任何一个地址，不过通常使用该网段的第一个可用的地址或最后一个可用的地址，这是为了尽可能避免和网络中的计算机地址冲突

IP 地址分类

A 类地址：网络地址的最高位是 **0** 的地址为 A 类地址，网络 ID 是 0 不能用，127 作为保留

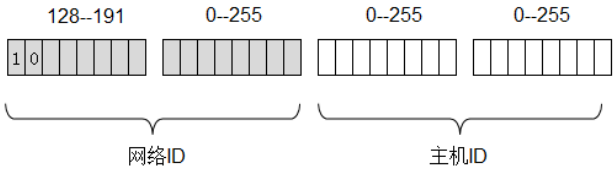
网段，因此 A 类地址的第 1 部分取值范围 1-126

A 类网络默认子网掩码为 **255.0.0.0**。主机 ID 由第 2 部分、第 3 部分和第 4 部分组成，每部分的取值范围 0-255，共 256 种取值，一个 A 类网络主机数量是 256×256×256=166777216，这里还需减去 **2**，主机 ID 全 0 的地址为网络地址，而主机 ID 全部为 1 的地址 000000000 为广播地址



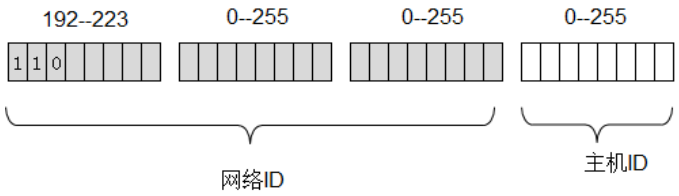
B 类地址：网络地址的最高位是 **10** 的地址为 B 类地址，IP 地址第 1 部分的取值范围为 128-191

B 类网络默认子网掩码为 **255.255.0.0**。主机 ID 由第 3 部分和第 4 部分组成，每个 B 类网络可以容纳的最大主机数量 256×256-2=65023



C 类地址：网络地址的最高位是 **110** 的地址为 C 类地址，IP 地址第 1 部分的取值范围为 192-223。

C 类网络默认子网掩码为 **255.255.255.0**。主机 ID 由第 4 部分组成，每个 C 类网络可以容纳的最大主机数量 256-2=254



D 类地址：网络地址的最高位是 **1110** 的地址为 D 类地址，D 类地址第 1 部分的取值范围为 224-239。用于多播（也称为组播）的地址，只能做目标地址，组播地址没有子网掩码。

E 类地址：网络地址的最高位是 **11110** 的地址为 E 类地址，第一部分取值范围 240-254，该部分地址不用掌握

保留的 IP 地址：

- **主机 ID 全为 0 的地址**：特指某个网段，比如 192.168.10.0 255.255.255.0，指 192.168.10.0 网段

- **主机 ID 全为 1 的地址**：特指该网段的全部主机，如果你的计算机发送数据包使用主机 ID 全是 1 的 IP 地址，数据链层地址用广播地址 FF-FF-FF-FF-FF-FF
- **127.0.0.1**：是本地环回地址，指本机地址，一般用来测试使用
- **169.254.0.0**：169.254.0.0-169.254.255.255 实际上是自动私有 IP 地址
- **0.0.0.0**：如果计算机的 IP 地址和网络中的其他计算机地址冲突，使用 ipconfig 命令看到的就是 0.0.0.0，子网掩码也是 0.0.0.0

公网地址、私网地址

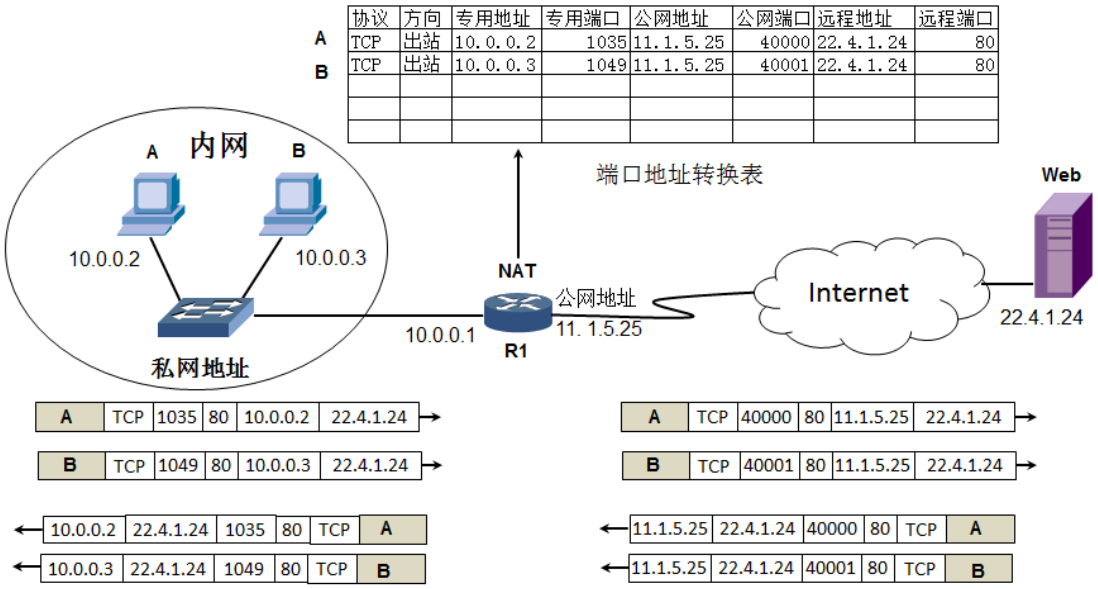
私网地址：地址可以被用于私有网络，在 Internet 没有这些 IP 地址，Internet 上的路由器也没有到私有网络的路由表

私网地址的分类：

- A 类：10.0.0.0 255.0.0.0，保留了一个 A 类网络
- B 类：172.16.0.0 255.255.0.0～172.31.0.0 255.255.0.0，保留了 16 个 B 类网络
- C 类：192.168.0.0 255.255.255.0～192.168.255.0 255.255.255.0，保留了 256 个 C 类网络

私网地址访问互联网：

私网地址访问 Internet 需要 NAT（网络地址转换）或 PAT（端口地址转换）



NAT 的过程：NAT 对应的路由器有一个地址映射表，A 给服务器通信的时候，NAT 会将私网地址转换为公网地址，当服务器给 A 通信，会直接给 NAT 通信，NAT 再将公网地址转换为私

网地址（仅替换的是 IP 地址）

PAT 的过程：除了 NAT 的 IP 地址的替换，同样替换源端口，将源端口 1035 替换为 40000，将源端口 1049 的替换为 40001，当数据发送回来的时候，可以对应端口给了不同的计算机

子网划分

子网划分主要做的两件事情：

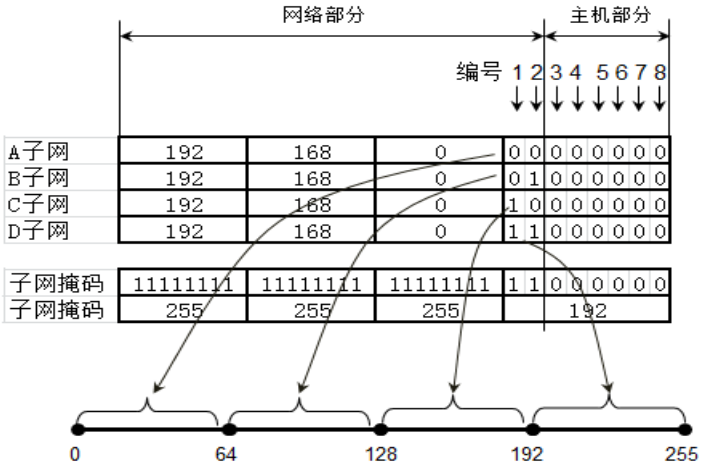
✧ 确定子网掩码的长度

✧ 确定子网中第一个可用的 IP 地址和最后一个可用的 IP 地址

等长子网划分：

等长子网划分就是将一个网段等分成多个网段，也就是等分成多个子网

C 类子网划分为 4 个等长的子网，子网掩码是 255.255.255.192



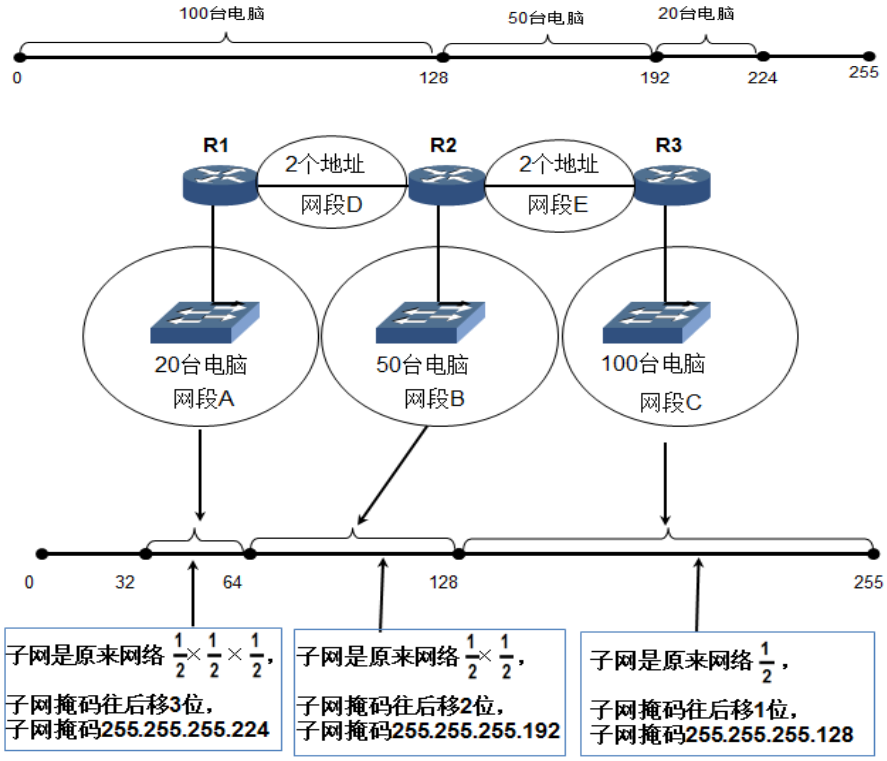
规律：如果一个子网是原来网络 $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$，子网掩码往后移2位。

每个子网的最后一个地址都是本子网的广播地址，不能分配给计算机使用，A 子网的 63、B 子网的 127、C 子网的 191 和 D 子网的 255

	网络部分			主机位全1
A子网	192	168	0	0 0 1 1 1 1 1 1
				63
B子网	192	168	0	0 1 1 1 1 1 1 1
				127
C子网	192	168	0	1 0 1 1 1 1 1 1
				191
D子网	192	168	0	1 1 1 1 1 1 1 1
				255
子网掩码	11111111	11111111	11111111	1 1 0 0 0 0 0 0
子网掩码	255	255	255	192

变长子网划分：

规律：如果一个子网地址块是原来网段的 $(\frac{1}{2})^n$，子网掩码就在原网段的基础上后移 n 位，不等长子网，子网掩码也不同



CIDR 表示子网掩码：判断 192.168.0.101/26 所属的子网 该 IP 和子网掩码做与运算

合并网段

子网掩码往左移 1 位，能够合并两个连续的网段，但不是任何连续的网段都能合并

网段合并的规律，子网掩码左移 1 位能够将能够合并两个网段，左移 2 位，能够合并四个网段，左移 3 位，能够合并 8 个网段

判断两个子网是否能够合并

判断连续的 2 个网段是否能够合并，只要第一个网络号能被 2 整除，就能够通过左移 1 位子网掩码合并（对于一个数字来说，如果能够被 2 整除，则最后一位为 0，能被 4 整除的，最后两位为 0，能够被 8 整除的最后三位为 0）

判断四个网段是否能够合并

要合并连续的四个网络，只要第一个网络的网络号写成二进制后面两位是 00，这四个网段就能合并，只要一个数能够被 4 整除，写成二进制最后两位肯定是 00

判断是超网还是子网

如果该网段的子网掩码比默认子网掩码长，就是子网

如果该网段的子网掩码比默认子网掩码短，则是超网

静态路由和动态路由

本章主要内容：

1．网络层实现的功能，网络通畅的条件以及不通畅的情况

2．静态路由，路由汇总，默认路由

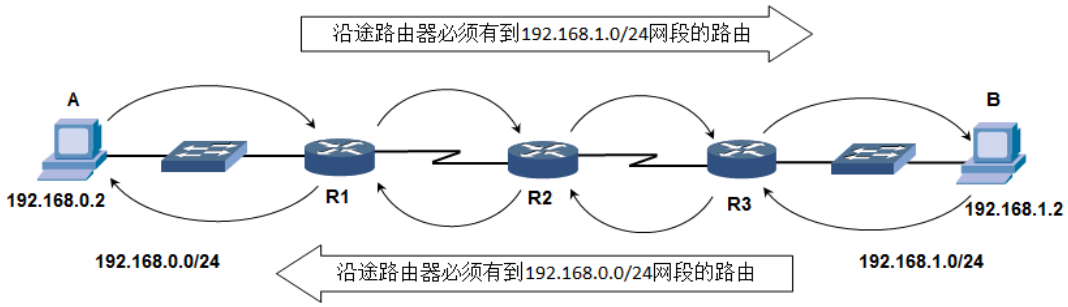
3．动态路由 IP 协议（RIP 协议、OSPF 协议）

路由-网络层实现的功能

网络层的功能：网络中路由器来实现，为每一个 IP 数据包单独选择转发路径（提供不可靠的交付）

通俗一点来讲，网络中的路由器为每一个数据包单独的选择转发路径，网络层不提供服务质量的承诺。也就说路由器直接丢弃传输过程中出错的数据包，如果网络中待发的数据包太多，路由器处理不了就直接丢弃，路由器也不判断数据包是否重复，也不确保数据包按发送顺序到达终点。

网络通畅的条件：要求数据包必须能够到达目标地址，同时数据包必须能够返回发送地址。

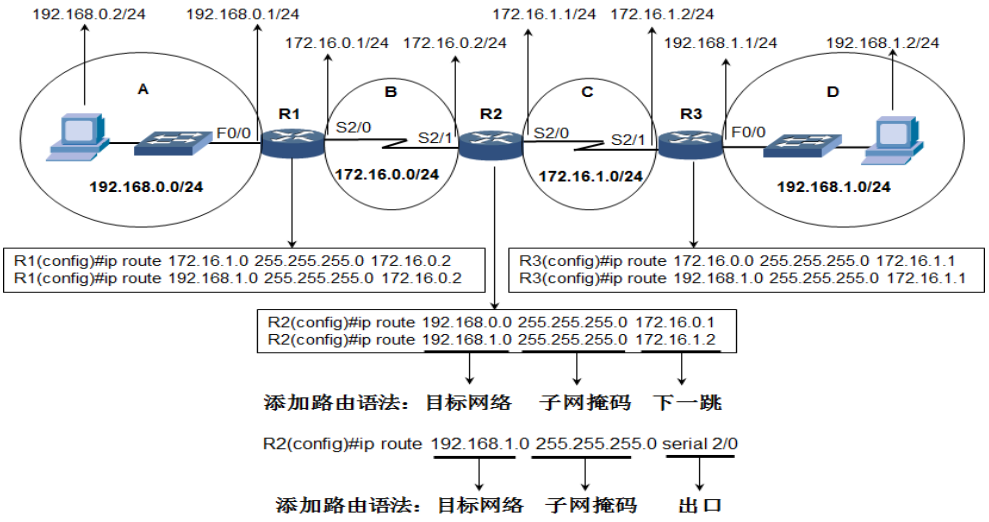


网络不通畅的情况：目标主机不可到达、请求超时

静态路由

管理员人工添加网络中的结点到直连网段和非直连网段的路由的方式就是静态路由。

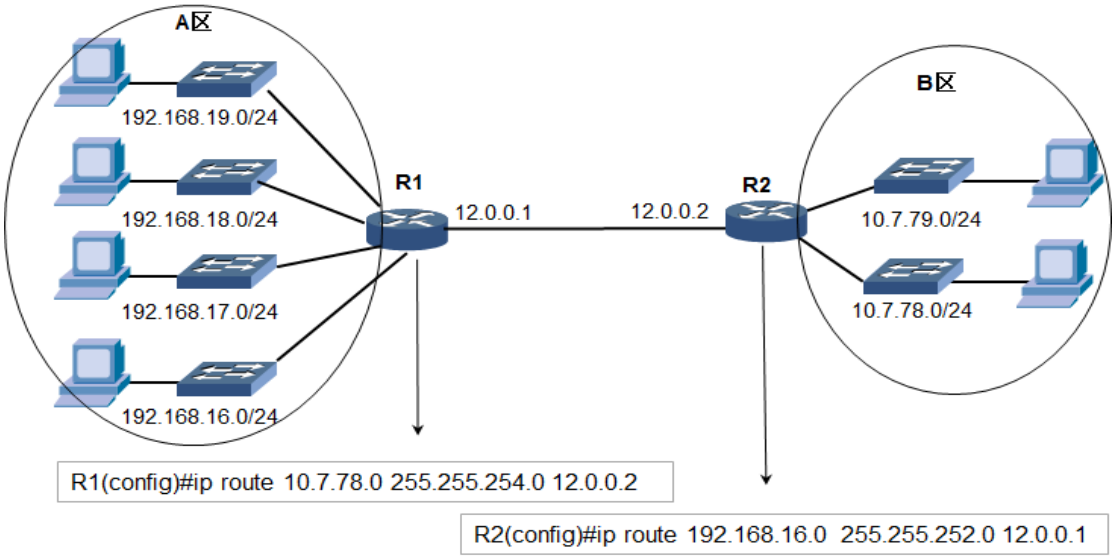
配置静态路由：



路由器只关心到某个网段如何转发数据包，因此我们在路由器上添加路由，必须是到某个网段（子网）的路由，不能添加到某个特定地址的路由。

路由汇总：

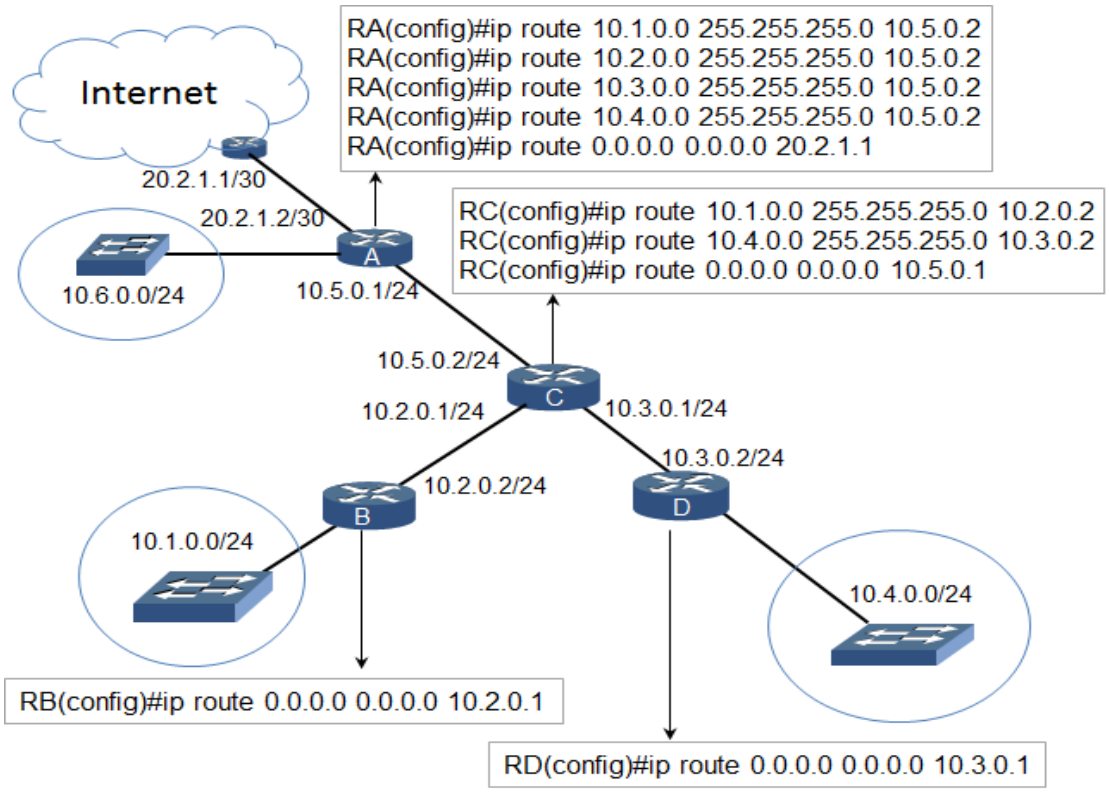
- ✧ 如果路由器把全球所有的网段都添加到路由表，那将是一个非常庞大的路由表，庞大的路由表势必会增加处理时延。
- ✧ 将物理位置连续的网络分配地址连续的网段，就可以在边界路由器上将远程的网络合并成一条路由，这就是路由汇总。



默认路由：

在路由器上添加添加到 `0.0.0.0 0.0.0.0` 网段的路由，就是默认路由。

- ✧ 子网掩码 `0.0.0.0` 网段包括了全球所有 IPv4 地址，也就是全球最大的网段
- ✧ 任何一个目标地址都与默认路由匹配，默认路由是在路由器没有为数据包找到更为精确匹配的路由，最后匹配的一条路由。（优先级最低）



动态路由

所有的动态路由协议都属于 IP 协议

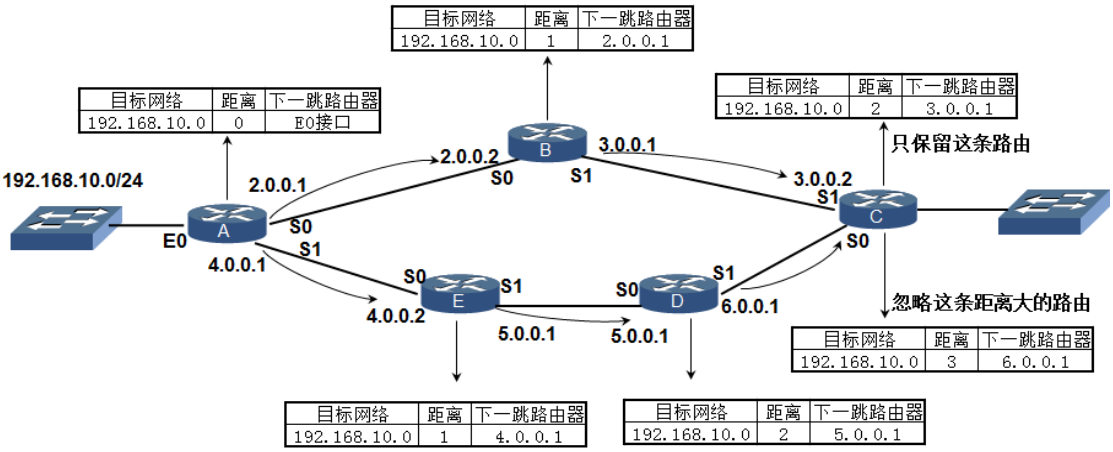
RIP 协议：

路由信息协议 RIP 是一个真正的距离矢量路由选择协议。

- ✧ 它每隔 30 秒就送出自己完整的路由表到所有激活的接口。如果隔几个 30 秒后，没有收到某个网段的路由信息，从路由表中删除该路由信息。
- ✧ RIP 协议选择最佳路径的标准就是跳数，认为到达目标网络经过的路由器最少的路径就是最佳路径。
- ✧ 默认它所允许的最大跳数为 15 跳，也就是说 16 跳的距离将被认为是不可达的。
- ✧ 在小型网络中，RIP 会运转良好，但是对于使用慢速 WAN 连接的大型网络或者安装有大量路由器的网络来说，它的效率就很低了。

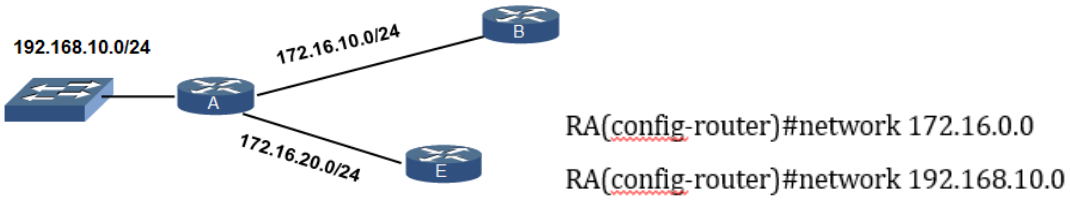
如果学到到一个网段有两条路径，只保留最佳路径。如果当两个网段都能够等价到达的话，则进行自动的负载均衡！会自动添加两条路由信息的。

举例说明 192.168.10.0/24 网段如何通过 RIP 协议通告给网络中的其他路由器的。

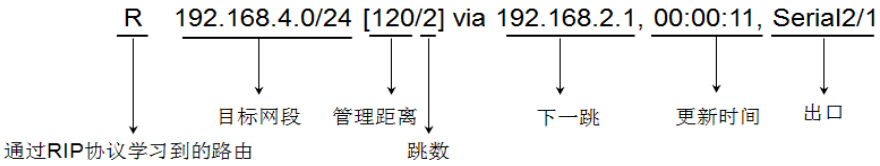


在上图中，各个路由器都运行了 RIP 协议，A 路由器会通告路由信息（到 192.168.10.0 这个网段距离是 0），通告信息的原地址是 2.0.0.1，目标地址是 2.0.0.2，那么 B 会根据 A 的通告信息，得知到192.168.10.0这个网段的距离是1，下一跳是刚才的原地址2.0.0.1，同时 B 路由器也会通告给 C 路由器，C 会根据通告信息得知到 192.168.10.0 这个网段的距离是 2，下一跳是刚才的原地址 3.0.0.1，同时，A 路由器也会向 E，E 向 D，D 向 C 发通告信息，其中 C 得到两条通告信息，选择最佳路径。

配置 RIP 协议：



show ip route

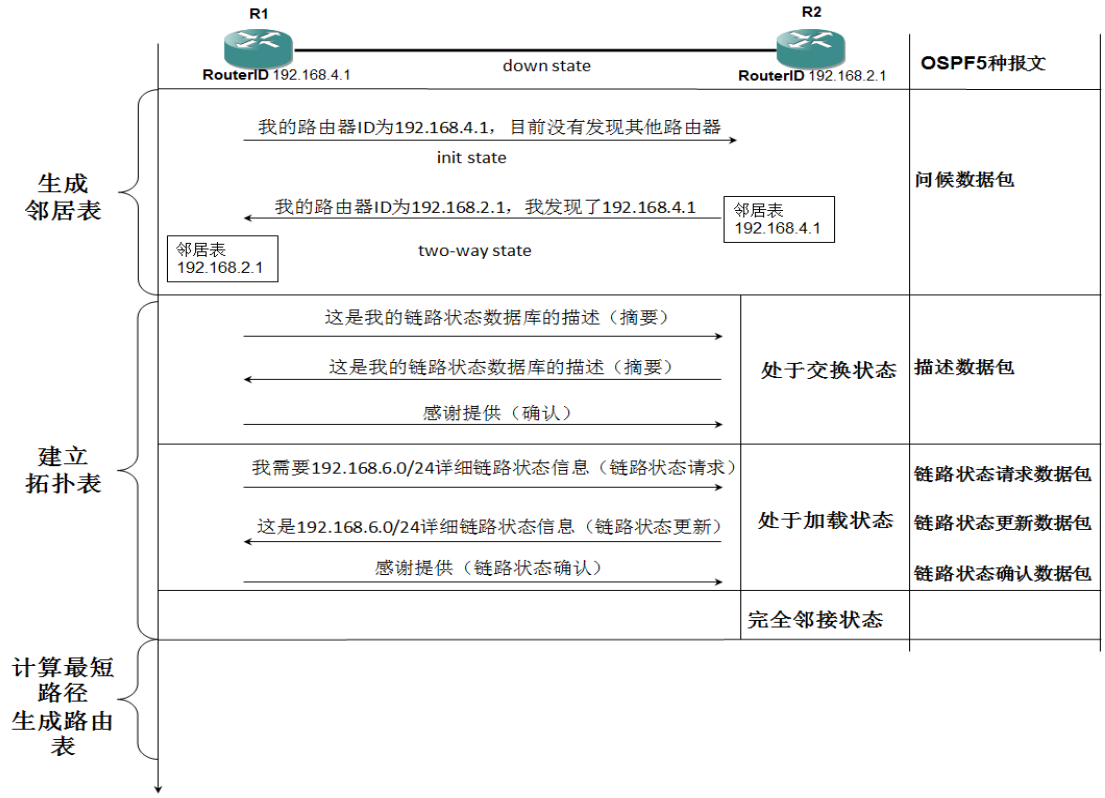


OSPF 协议：

✧ OSPF 协议选择最佳路径的标准是带宽，带宽越高计算出来的开销越低。到达目标网络的各个链路累计开销最低的，就是最佳路径。

✧ 运行 OSPF 协议的路由器有 3 张表，邻居表、链路状态表（网络中有多少路由器，每个路由器连接那些网段）和路由表。

下面以这三张表的产生过程为线索，来分析在这个过程中，路由器发生了那些变化，从而说明 OSPF 协议的工作过程。



邻居表：周期性的发送 Hello 数据包，判断邻居是否健在，构成邻居表

链路状态表：相互交换邻居表之后，构建一张链路状态表

路由表：根据链路状态计算最短路径（标准是带宽）

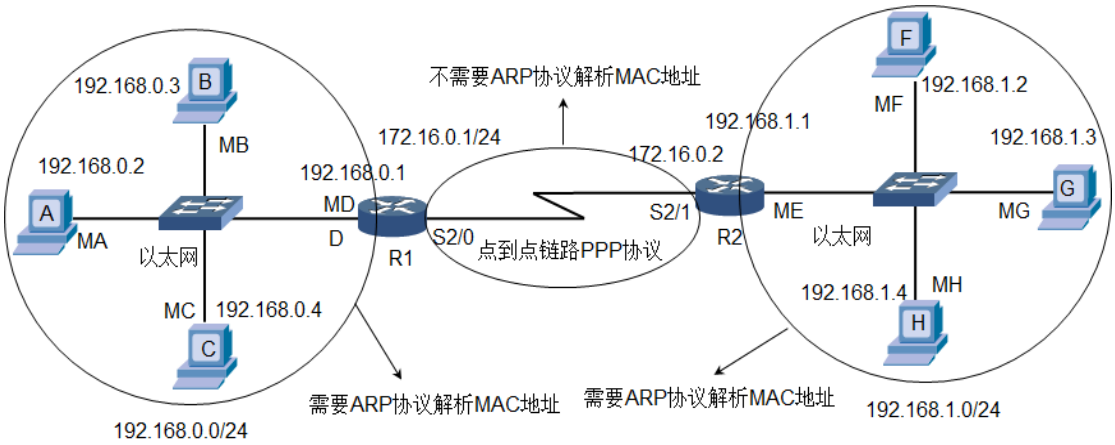
网络层协议

本章主要内容：ARP 协议、ICMP 协议、IGMP 协议

ARP 协议

✧ ARP 协议的作用，将以太网中的计算机的 IP 地址解析成 MAC 地址

✧ 点到点链路使用 PPP 协议，不需要 ARP 协议（PPP 协议不需要 MAC 地址）



ARP 协议是建立在网络中各个主机互相信任的基础上的，计算机 A 发送 ARP 广播帧解析计算机 C 的 MAC 地址，A 无法知道 H 和 G 的 MAC 地址，无法跨网段 ARP 解析

ICMP 协议

ICMP 是 Internet 控制报文协议，用于在 IP 主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网络本身的消息。

IGMP 协议

Internet 组管理协议称为 IGMP 协议，是因特网协议家族中的一个组播协议。该协议运行在主机和组播（多播）路由器之间，IGMP 协议是网络层协议。

IGMP 实现如下双向的功能：

1. 主机通过 IGMP 通知路由器希望接收或离开某个特定组播组的信息。
2. 路由器通过 IGMP 周期性地查询局域网内的组播组成员是否处于活动状态（如果没有成员，那么就不让路由器去接受某个特定的组播信息），实现所连网段组成员关系的收集与维护。

传输层

本章主要内容：

1. 传输层的两个协议（应用场景、服务和端口、与应用层的关系）
2. TCP 协议和 UDP 协议的特点
3. 可靠传输
4. 流量控制
5. 拥塞控制
6. TCP 建立连接和释放连接

传输层的两个协议

传输层协议应用的场景：

网络中的计算机通信无外乎有以下两种情况：

- ✧ 要发送的内容多，需要将发送的内容分成多个数据包发送。

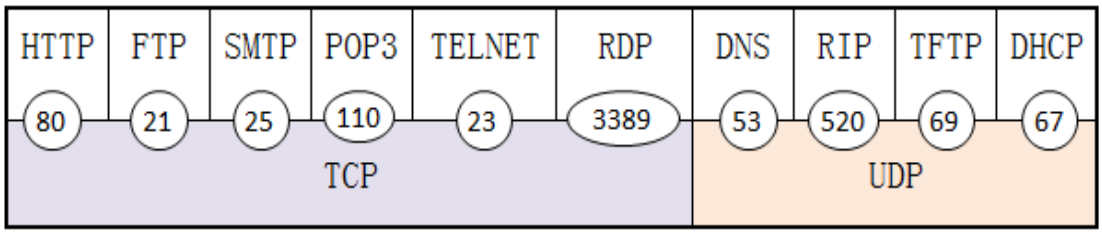
  （TCP）下载电影、QQ 传输文件

- ✧ 要发送的内容少，一个数据包就能发送全部内容。

  （UDP）域名解析、QQ 聊天、QQ 语音聊天、流媒体多播发送视频流

传输层协议和应用层协议之间的关系：

传输层协议加一个端口号来标识一个应用层协议，展示了传输层协议和应用层协议之间的关系。

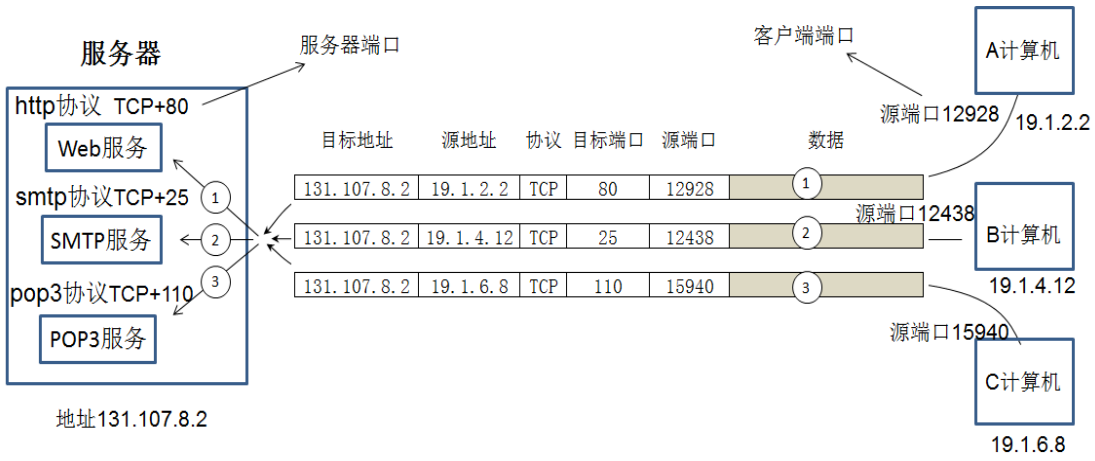


一些常见的应用层协议和传输层协议：

- HTTP 默认使用 TCP 的 80 端口标识
- FTP 默认使用 TCP 的 21 端口标识
- SMTP 默认使用 TCP 的 25 端口标识
- POP3 默认使用 TCP 的 110 端口
- HTTPS 默认使用 TCP 的 443 端口
- DNS 使用 UDP 的 53 端口

服务和端口之间的关系：

Linux 操作系统为网络中计算机提供服务的服务，一旦启动就会使用 TCP 或 UDP 的某个端口侦听客户端的请求



TCP 协议和 UDP 协议的特点

UDP 协议的特点：

1. UDP 是无连接的，即发送数据之前不需要建立连接（当然发送数据结束时也没有连接可释放），因此减少了开销和发送数据之前的时延。
2. UDP 不保证可靠交付

3. UDP 是面向报文的，发送方的 UDP 对应用程序交下来的报文（不对报文合并和拆分），在添加首部后就向下交付给网络层
4. UDP 没有拥塞控制（QQ 上的语音聊天）
5. UDP 支持一对一、一对多（多播）、多对一和多对多的交互通信
6. UDP 的首部开销小，只有 8 个字节，比 TCP 的 20 个字节的首部要短

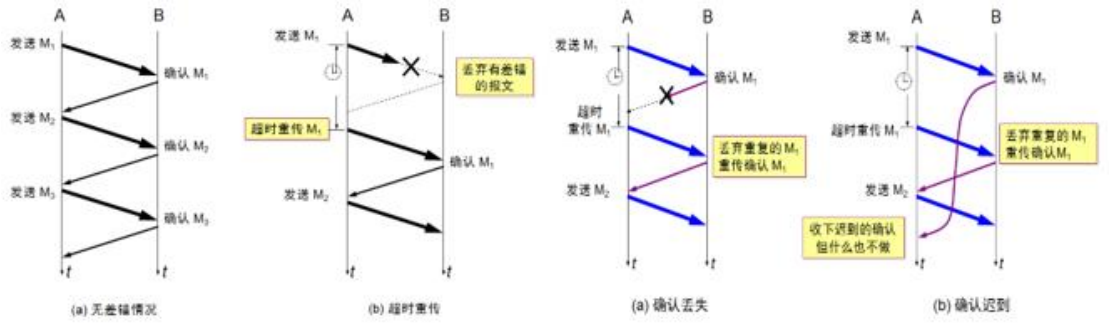
TCP 协议的特点：

TCP 协议是能够实现数据分段传输、可靠传输、流量控制、网络拥塞避免等功能，因此 TCP 报文的首部要比 UDP 报文首部字段要多，并且首部长度不固定。

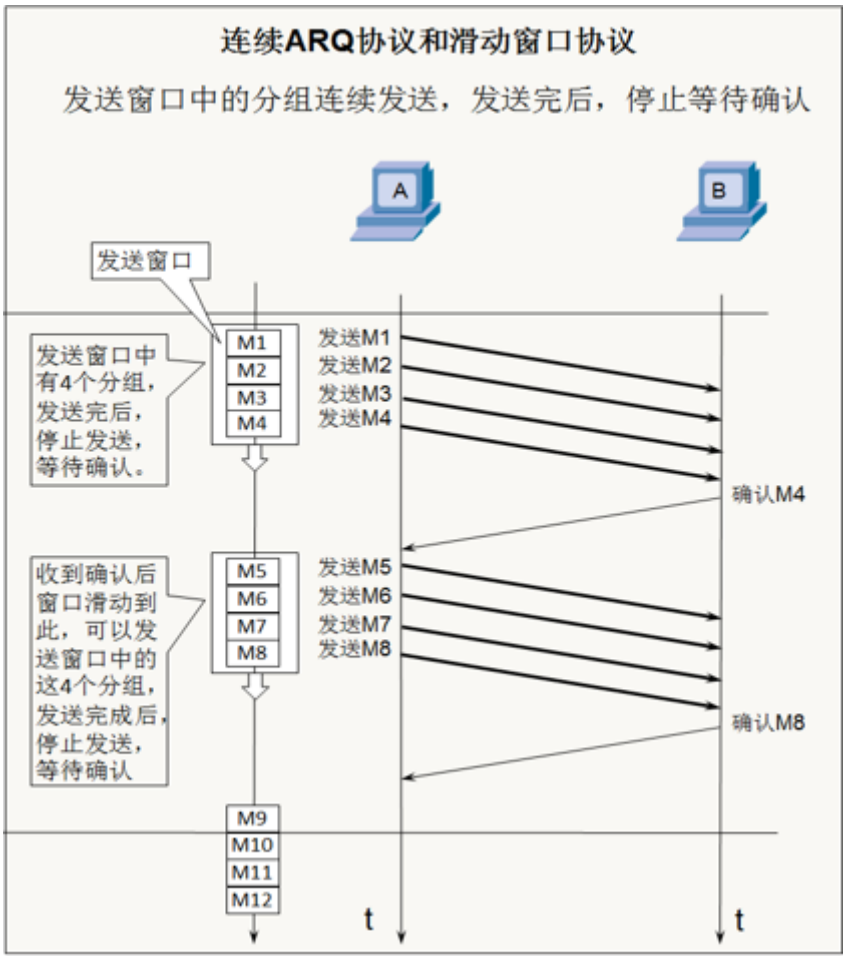
1. TCP 是面向连接的传输层协议。要建立连接进行三次握手，释放连接进行四次挥手
2. 每一条 TCP 连接只能有两个端点，每一条 TCP 连接只能是点对点的（一对一）
3. TCP 提供可靠交付的服务，通过 TCP 连接传送的数据，无差错、不丢失、不重复、且按序发送。
4. TCP 提供全双工通信。TCP 允许通信双方的应用进程在任何时候都能发送数据。TCP 连接的两端都设有发送缓存和接收缓存，用来临时存放双向通信的数据。在发送时，应用程序把数据传送给 TCP 的缓存后，就可以做自己的事，而 TCP 在合适的时候把数据发送出去。在接收时，TCP 把收到的数据放入缓存，上层的应用进程在合适的时候读取缓存中的数据。
5. 面向字节流，TCP 中的“流”指的是流入到进程或从进程流出的字节序列（对上层的应用程序来说接受的是字节流，传递的也是字节流）

可靠传输

停止等待协议实现可靠传输：



改进的停止等待协议：连续 ARQ 协议（连续发一组数据包）和滑动窗口协议（收到确认，窗口向后移）



改进的确认--选择性确认：

TCP 通信时，发送 4 个数据包，如果第三个数据包丢失，最后确认的是 M2，这样原先已经正确传输的分组（M4）也可能重复发送，降低了 TCP 性能。为改善这种情况，发展出 SACK 选择确认技术，使 TCP 只重新发送丢失的包，不用发送后续所有的分组，而且提供相应机制使接收方能告诉发送方哪些数据丢失，哪些数据已经提前收到等。

超时重传时间：

在停止等待协议的出错情况中，有超时重传的情况，那超时重传的时间（比 TCP 往返传输时间多一点）应该设置为多大？

TCP 往返传输时间（RTT）的测量可以采用两种方法：

1. TCP Timestamp 选项

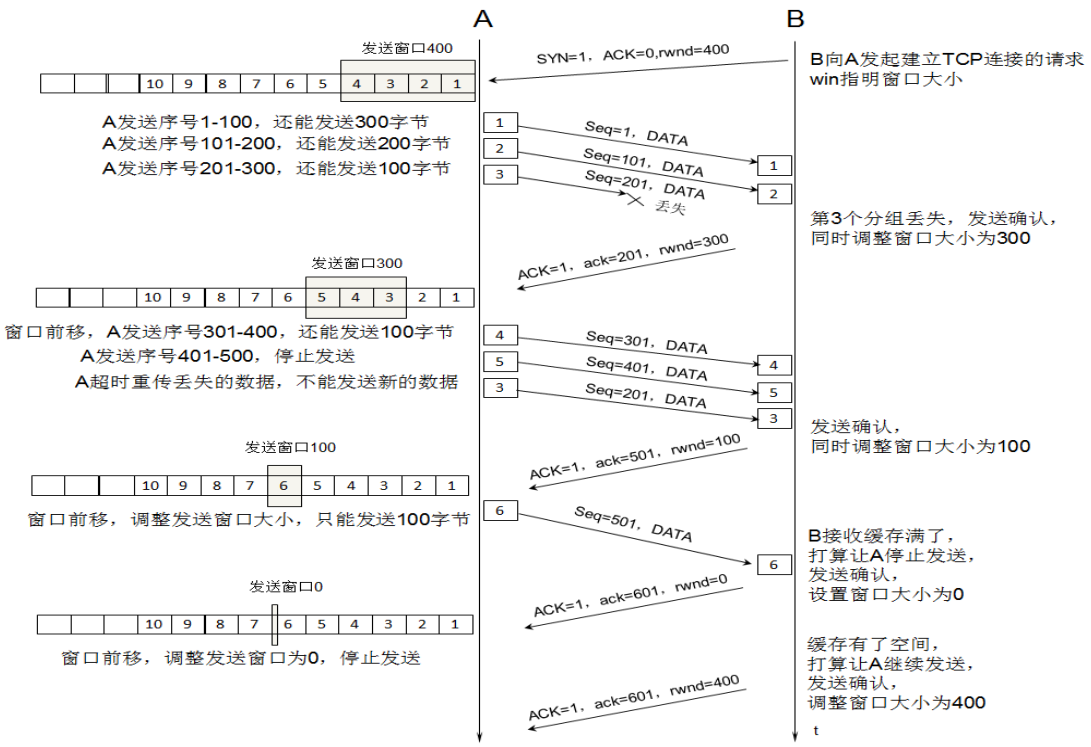
TCP 时间戳选项可以用来精确的测量 RTT。发送方在发送报文段时把当前时钟的时间值放入时间戳字段，接收方在确认该报文段时把时间戳字段值复制到时间戳回送回答字段。因此，发送方在收到确认报文后，可以准确地计算出 RTT 来。RTT=当前时间-数据包中 Timestamp 选项的回显时间。

2．重传队列中数据包的 TCP 控制块

在 TCP 发送窗口中保存着发送而未被确认的数据包，数据包 skb 中的 TCP 控制块包含着一个变量， tcp_skb_cb→when，记录了该数据包的第一次发送时间，当收到该数据包确认，就可以计算 RTT，RTT=当前时间-when。这就意味着发送端收到一个确认，就能计算新的 RTT

流量控制

流量控制就是发送端根据接收端的接收能力调整发送速度

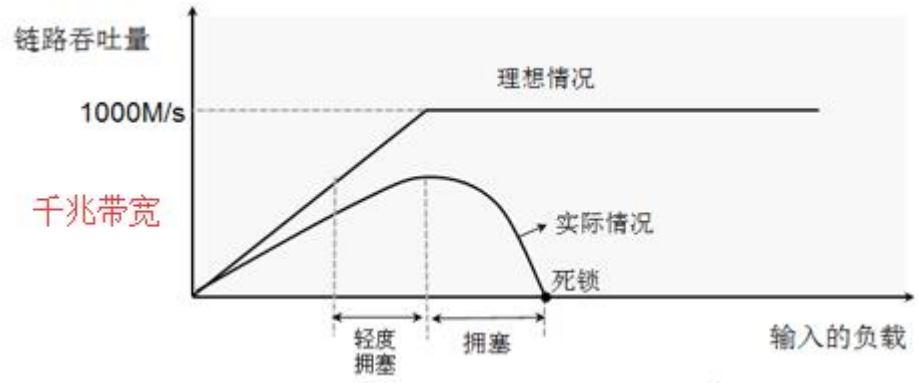


拥塞控制

防止发送方发的太快，使得网络来不及处理，从而导致网络拥塞

拥塞控制的原理：

避免输入的负载达到某一个特定的值，出现拥塞状态

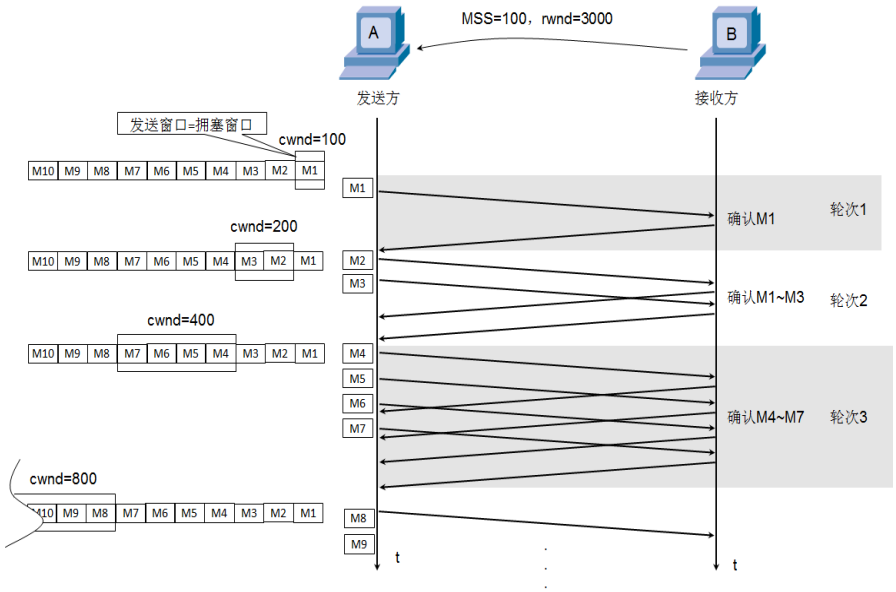


✧ 在理想情况下，输入的负载当小于 1000M 时，吞吐量呈线性变化，当输入的负载大于 1000M 时，会丢弃一些数据包，只能达到 1000M 的吞吐量

✧ 真实情况，当处于某一个输入的负载时，已经存在了丢包现象，随着输入的负载越多，丢包对严重，处于轻度拥塞状态，当输入的负载到达某一个值时，出现拥塞，严重导致死锁

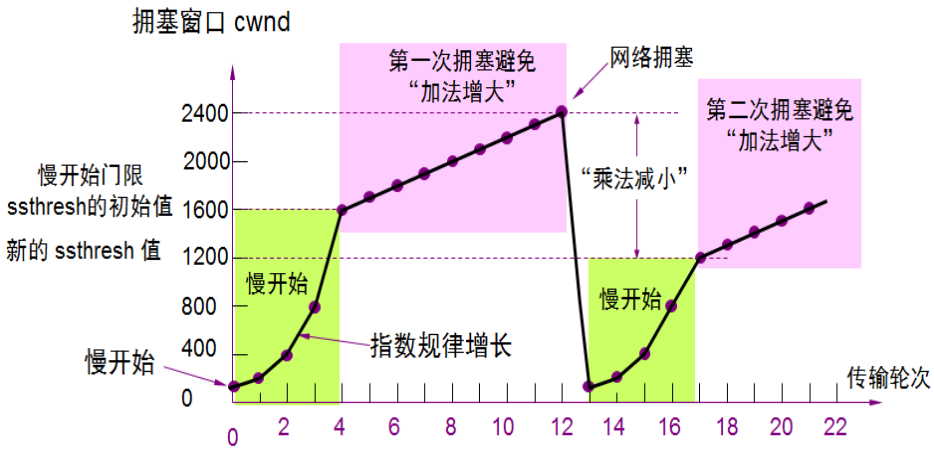
拥塞控制方法：

1．慢开始算法：发送方刚开始根据拥塞窗口先发送 1 个窗口数据（一个窗口数据是 100 个字节），当收到接收方的确认时，证明没有丢失，那么拥塞窗口会调整成 2 个窗口数据，当再次收到接收方的确认时，证明没有丢失，那么拥塞窗口会调整成 4 个窗口数据，当再次收到确认的时候，会将拥塞窗口调整成 8 个窗口数据



2．拥塞避免

当拥塞窗口到达 16 个窗口数据的时候，发现出现了丢包现象，那么就会采用拥塞避免算法

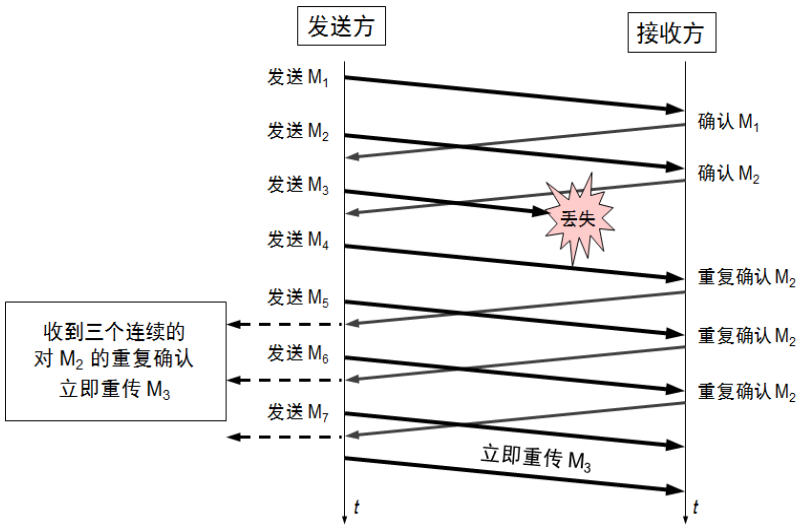


由上图可知，当到达第一次的慢开始门限，会进行拥塞避免算法，不再采用指数增长，采用加法增长的方式，当到达 2400 时，发现有丢包现象，再次增加会产生网络拥塞，那么就用 2400 去除以 2，形成新的慢开始门限(1200)，依次类推。

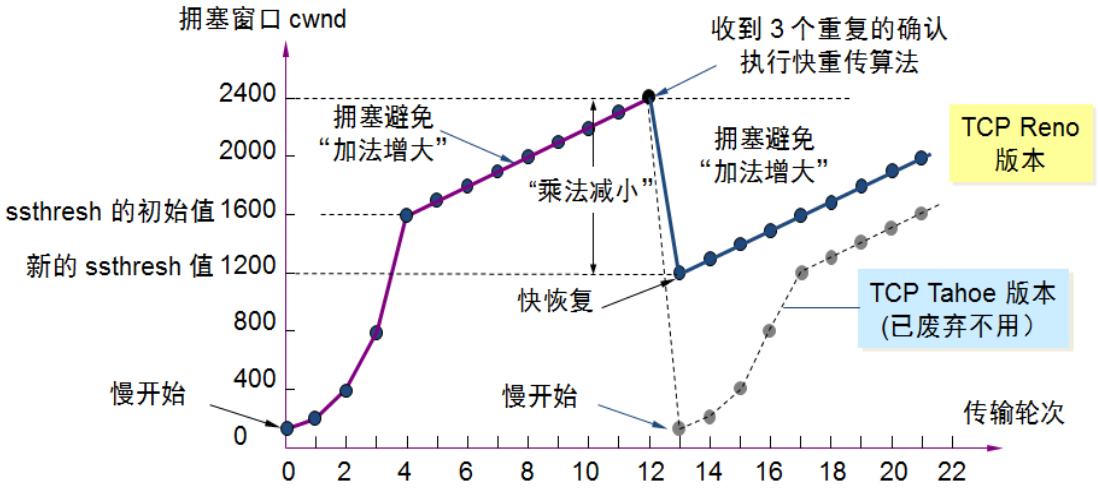
3．改进的拥塞控制方法：快重传和快恢复

快重传算法：

- ✧ 快重传算法首先要求接收方每收到一个失序的分组后就立即发出重复确认（为的是使发送方及早知道有分组没有到达对方）而不要等待自己发送数据时才进行捎带确认。
- ✧ 快重传算法规定，发送方只要一连收到三个重复确认就应当立即重传对方尚未收到的报文段 M3，而不必继续等待为 M3 设置的重传计时器到期。



当收到三个重复的确认的时候，则执行快恢复算法：



直接得到新的慢开始门限，直接从慢开始门限进行线性增长

发送窗口的上限：

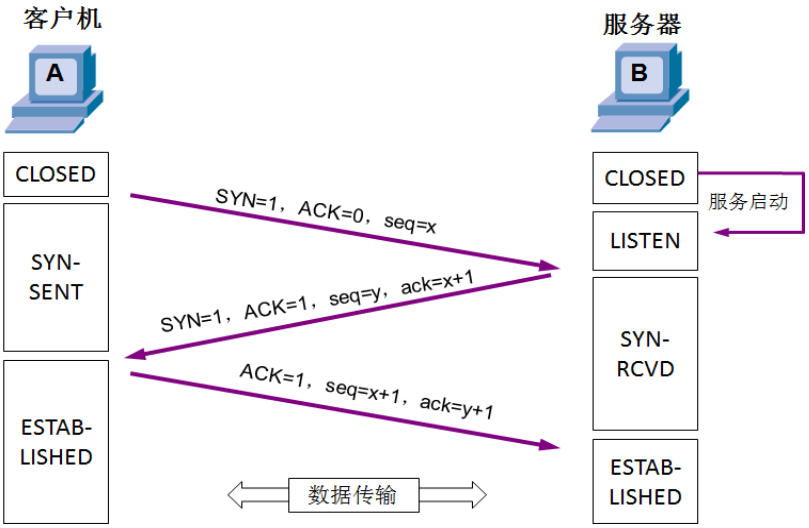
将拥塞控制和流量控制一起考虑，发送方的窗口的上限值应当取为接收方窗口 rwnd 和拥塞窗口 cwnd 这两个变量中较小的一个，也就是说：

发送方窗口的上限值 = Min [rwnd，cwnd]

- 当 rwnd<cwnd 时，是接收方的接收能力限制发送方窗口的最大值。
- 反之，当 cwnd<rwnd 时，则是网络的拥塞限制发送方窗口的最大值。
- 也就是说，rwnd 和 cwnd 中较小的一个控制发送方发送数据的速率。

TCP 连接建立和释放

TCP 连接建立：



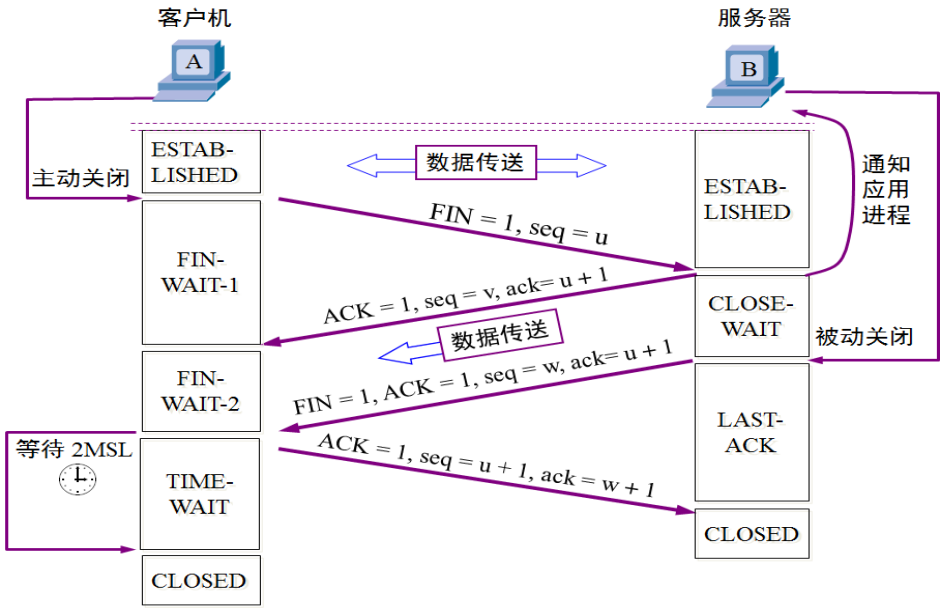
- 第一次握手：初识状态下，两者都处于 CLOSED 状态，客户端发送 SYN=1，ACK=0，（当 SYN=1 而 ACK=0 时，表明这是一个连接请求报文段）同时选择一个初识序号 seq=x 的数据包到服务器，并进入 SYN_SEND 状态，等待服务器确认
- 第二次握手：服务器收到请求报文，必须确认客户的 SYN=1，ACK=1（对方若同意建立连接，则应在响应的报文段中使 SYN=1 和 ACK=1）同时发送确认号为 ack=x+1，同时自己也选择一个初识序号 seq=y，此时服务器进入 SYN_RECV 状态
- 第三次握手：客户端收到服务器的确认报文，向服务器发送确认包 ACK=1，同时发送确认号 ack=y+1，此包发送完毕，客户端和服务器进入 ESTABLISHED 状态，完成三次握手。

为什么一定要三次握手？

比如现在 A 给 B 通信发送了一条连接建立请求，但是这个连接请求发送的非常慢，因此 A 发送了一个新的连接建立请求，此时 B 收到了新的请求，和 A 建立连接，这个时候 A 原来发的连接请求到了，B 会再发送一次确认连接建立，但是此时 A 不认为这个连接建立是 A 发送

的，此时 B 用的是原来的连接请求，A 认为是新的连接请求，这样，双方死锁，所以需要三次握手。

TCP 连接释放：



- 第一次挥手：初识状态，客户端和服务器进入 ESTABLISHED 状态，客户端发送一个 FIN（终止 FIN 用来释放一个连接，当 FIN=1 时，表明此报文段的发送方的数据已发送完毕，并要求释放传输连接）并发送一个 seq=u 序号，A 此时处于 FIN_WAIT1 状态
- 第二次挥手：被动关闭方收到 FIN 包后，发送一个 ACK 给对方，确认序号为收到序号+1，发送一个序号 seq = v，B 此时处于 CLOSE_WAIT 状态
- 第三次挥手：被动关闭方发送一个 FIN=1 ACK=1 seq=w ack=u+1，此时 A 收到之后处于 FIN_WAIT2 状态，B 此时处于 LAST_ACK 状态
- 第四次挥手：主动关闭方收到 FIN 后，然后发送一个 ACK=1 seq=u+1 ack=w+1 给被动关闭方，此时 B 处于 CLOSED 状态，但是 A 需要等待一个 2MSL 时间，此时处于 TIME_WAIT 状态，之后才变成 CLOSED 状态。

为什么 TIME_WAIT 状态还需要等 2MSL 后才能返回到 CLOSED 状态？

如果在 A 发确认报文的时候，中途丢失，那么 B 就需要重新发送一个 FIN=1 的报文，这个时候如果 A 处于 CLOSED 状态，那么就不能再发送确认报文