

计算机网络期末速通

1. 计算机网络概论

1.1 计网的定义、功能和组成

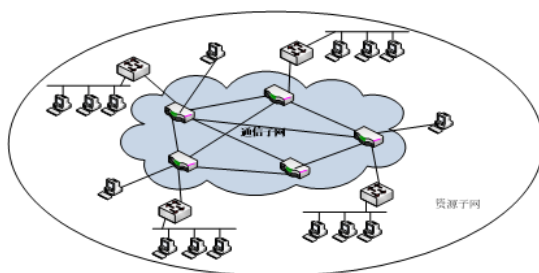
[计网的定义]

将地理上分散的、有独立工作能力多台计算机通过通信设备和通信线路连接, 在配有相应的网络通信软件的条件下, 实现数据通信和资源共享的系统.

[计网的功能]

- ① 建立数据通信.
- ② 实现资源共享.
- ③ 增加可靠性.
- ④ 提高系统处理能力.

[计网的逻辑结构]



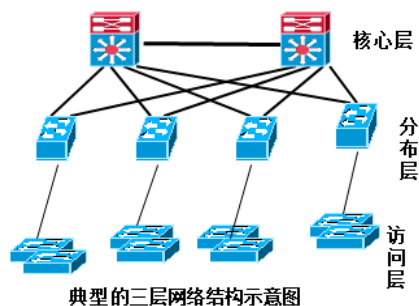
(1) 资源子网:

- ① 组成: 主机系统、终端设备、网络软件、数据资源.
- ② 功能: 数据处理, 提供网络资源和网络服务.

(2) 通信子网:

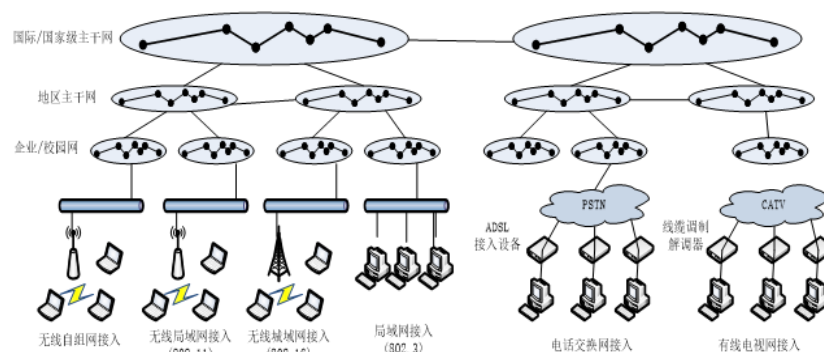
- ① 组成: 路由器、交换机、通信线路.
- ② 功能: 数据传输、路由、分组转发等通信处理.

[计网的三层组织结构]



- (1) 核心层: 提供核心三层交换骨干。
- (2) 分布层: 提供域间路由、地址汇聚。
- (3) 访问层: 提供物理网接入和隔离。

[互联网的组织机构]

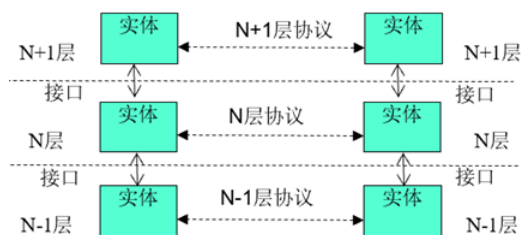


- ① 网络硬件、软件。
- ② 协议。
- ③ 存取控制。
- ④ 网络拓扑。

1.2 网络协议与网络体系结构

[网络协议]

- (1) 定义: 统一的信息交换规则, 规定信息格式后如何收发信息。
- (2) 协议三要素: 语法、语义、时序。



(3) 分层思想:

- ① 将网络系统功能分解为功能层, 用协议规定功能.
- ② 等功能层间用协议约束, 如协议数据单元、协议格式.
- ③ 相邻功能层间用接口交互, 如接口标准、使用服务.

[建立网络体系结构的必要性]

- (1) 解决通信问题: 信号、差错、寻址、交换、接口等问题.
- (2) 简化复杂问题: 设计、实现、抽象结构模型.
- (3) 层次化模型: 化整为零, 分而治之, 逐个击破.

[分层结构依据原则]

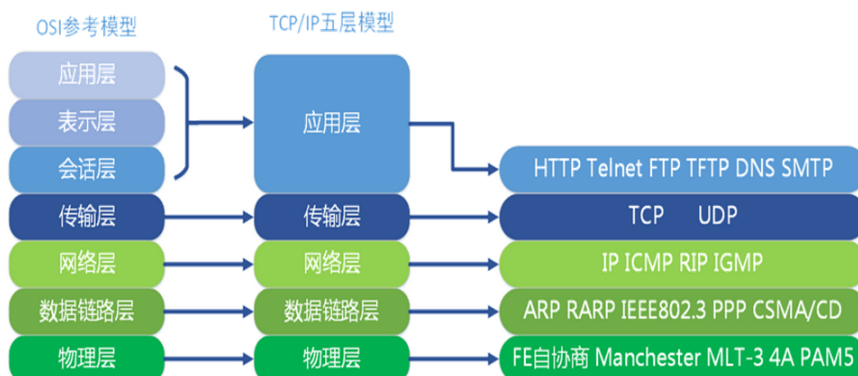
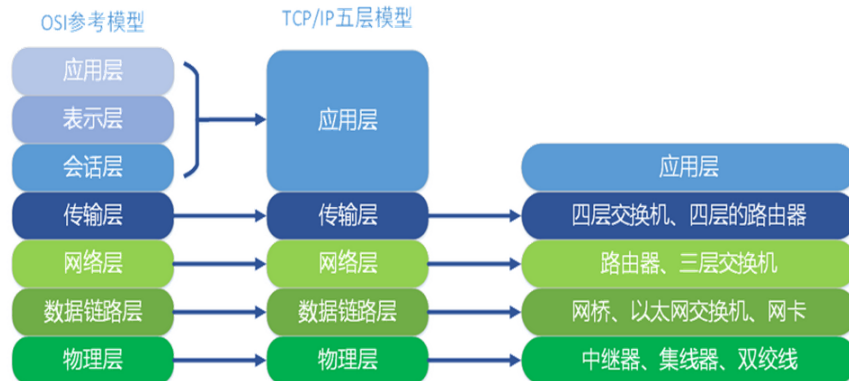
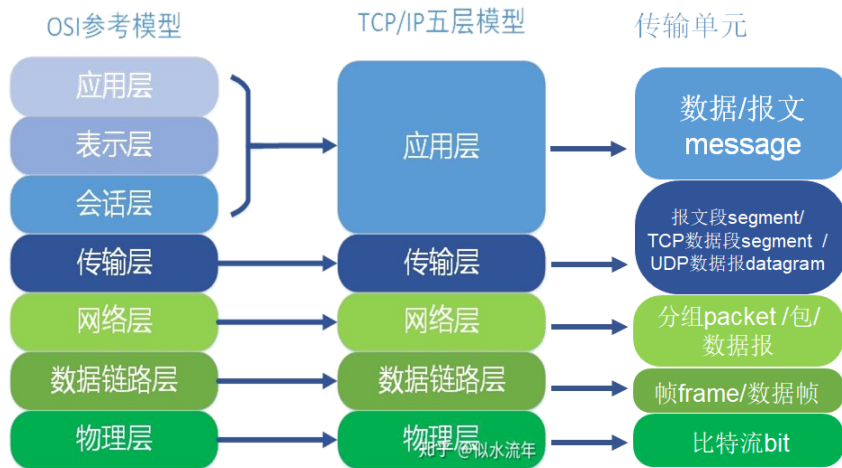
- (1) 明确规定每层实现的功能和服务.
- (2) 每层功能选择有利于标准化.
- (3) 层次划分不依赖于具体系统.
- (4) 高层使用下层提供的服务接口.

[封装协议数据单元]

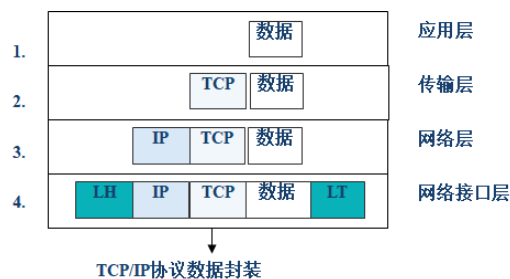
- (1) 封装: 上层协议数据单元添加头部信息, 形成新的协议数据单元, 交给相邻下层.
特别地, 网络层向数据链路层的封装还添加尾部.
- (2) 解封: 剥离相邻下层传递的协议数据的单元头部, 交给相邻上层.

[原理性的网络体系结构]

- (1) 物理层: 规定物理结构, 比特流透明传输.
- (2) 数据链路层: 实现逻辑链路上无差错的数据帧传输.
- (3) 网络层:
 - ① 功能: 实现网络传输, 解决寻址、路由、转发.
 - ② 协议: 协议: IP、ARP、RARP、ICMP.
- (4) 传输层: 实现进程间可靠、无差错的端到端通信.
- (5) 应用层:
 - ① 功能: 根据应用进程通信要求, 满足用户需要.
 - ② 协议:
 - (i) 应用支撑协议: DNS、DHCP、SNMP.
 - (ii) 网络应用协议: FTP、HTTP、SMTP、TELNET.



[TCP/IP协议模型]



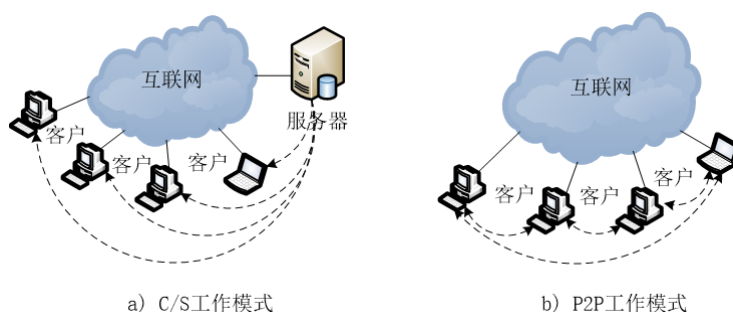
特点:

- (1) 在互联网应用层提供丰富的网络应用服务, 如C/S模式和P2P模式.
- (2) 在传输层支持面向连接、可靠、无差错的服务TCP和无连接、高效、快速的服务UDP.
- (3) 协议层较少, 实现简单、高效.
- (4) 实现多种异构网的互联, 网际协议IP为互联网核心.

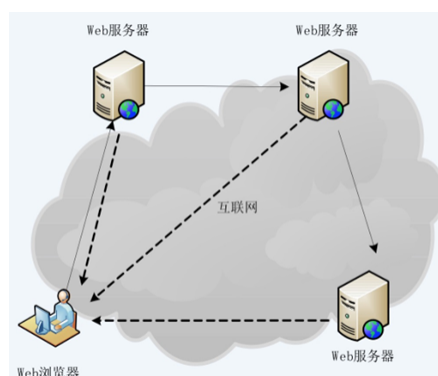
2. 应用层

2.1 Web服务与HTTP协议

[应用层的两种模式]

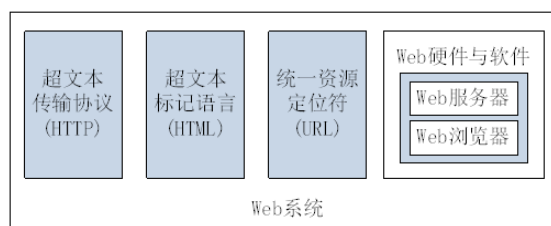


[Web的工作模式]



C/S模式.

[支持Web服务的关键技术]



(1) 超文本传输协议HTTP:

- ① Web服务应用层协议, 超文本在浏览器与Web服务器间的传输协议.
- ② 包括**非持续连接**、**持续连接**.

(2) 超文本标记语言HTML: 文档中的特殊数据格式, 一个文档可**链接**到另一文档.

(3) 统一资源定位符URL:

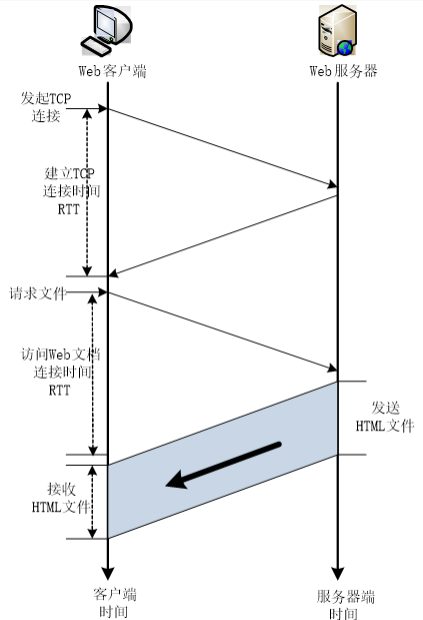
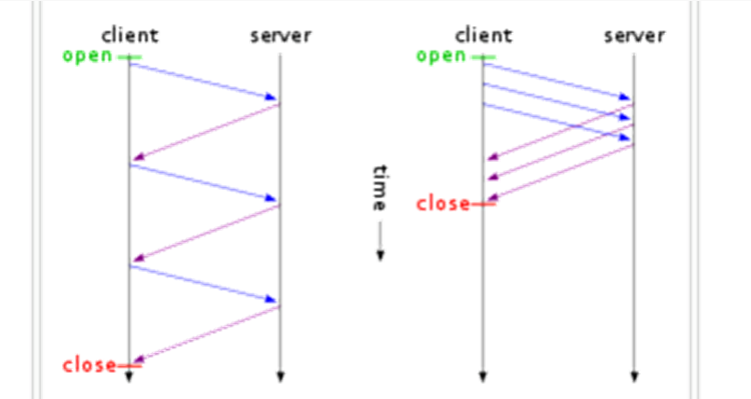
- ① 标识Web中的资源, 便于用户查找.
- ② 构成: 协议名称//主机名/文件名, 如"<http://www.szu.edu.cn/szu2007/index.html>"中, "http"为协议类型, "www.szu.edu.cn"为主机名, "szu2007/index.html"为路径和地址.

[HTTP的工作机制]

(1) 工作机制:

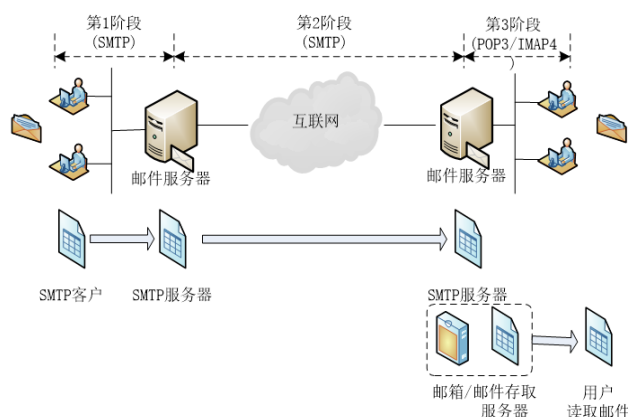
- ① 使用面向连接的TCP协议, 客户端Web服务器与Web服务器间建立TCP连接.
- ② TCP连接建立后, 浏览器进程发送HTTP请求报文, 并接受应答报文.
- ③ Web服务器接收HTTP请求报文, 并发送应答报文.
- ④ TCP提供可靠服务, 保证客户进程发送的HTTP请求能正确到达服务器端, 服务器进程发送HTTP应答报文能正确到达客户端.
- ⑤ Web服务器发送HTTP应答报文时不保存浏览器的请求状态信息, 故HTTP协议是无状态协议.

(2) 两种状态: 非持续连接、持续连接.

非持续连接 (HTTP 1.0)	持续连接
每次请求、相应都建立一次TCP连接	服务器发出响应后保持TCP连接, 相同的客户端进程与服务器端间的后续报文都通过该连接传送
	 <p>非流水线 客户端只有在接收到前一个响应时才能发出新的请求</p> <p>流水线 客户端在没有收到前一个响应时就发出新的请求</p>

2.2 电子邮件

[电子邮件的协议]



(1) 邮件传输协议:

- ① 客户端到邮件服务器、邮件服务器间: SMTP (Simple Mail Transfer Protocol, 简单邮件传输协议).

电子邮件应用程序发送邮件采用SMTP协议.

- ② 邮件服务器到客户端: POP-3(Post Office Protocol 3, 邮政协议第3版).

(2) 邮件存储访问协议

- ① IMAP-4 (Internet Message Access Protocol 4, Internet邮件访问协议).

电子邮件应用程序接收邮件采用POP-3协议或IMAP-4协议.

- ② MIME (Multipurpose Internet Mail Extensions, 通用Internet邮件扩展协议).

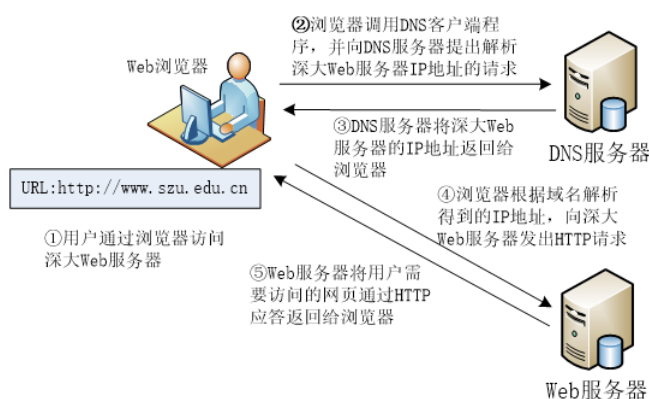
电子邮件应用程序邮件扩展采用MIME协议.

[基于Web的电子邮件]

用户代理为Web浏览器, 用户与邮箱服务器通信采用HTTP协议而非POP-3协议或IMAP-4协议, 但邮箱服务器间的通信仍采用SMTP协议.

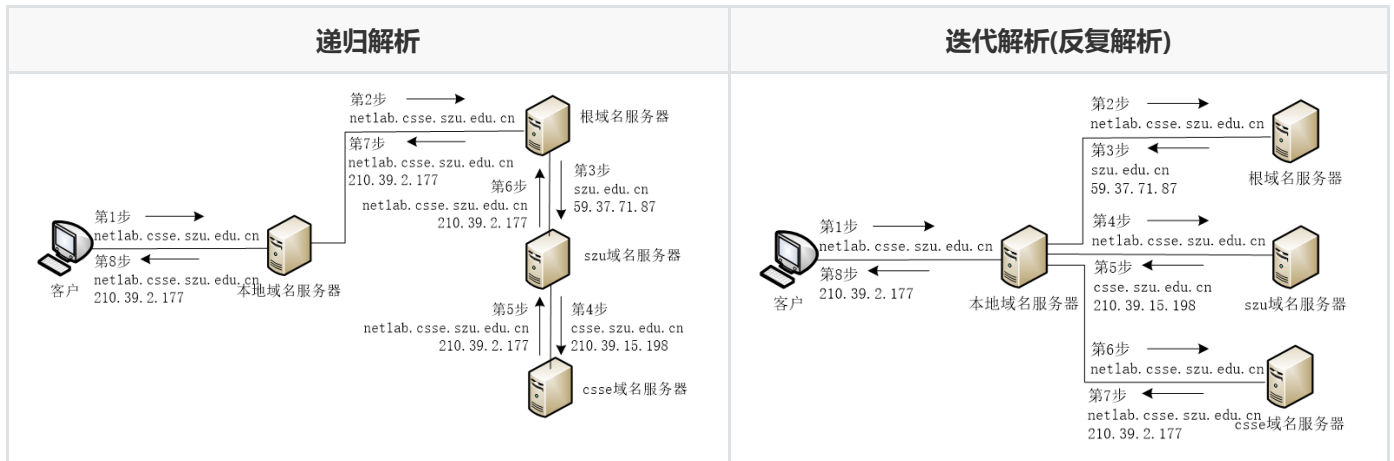
2.3 域名系统与DNS

[DNS]



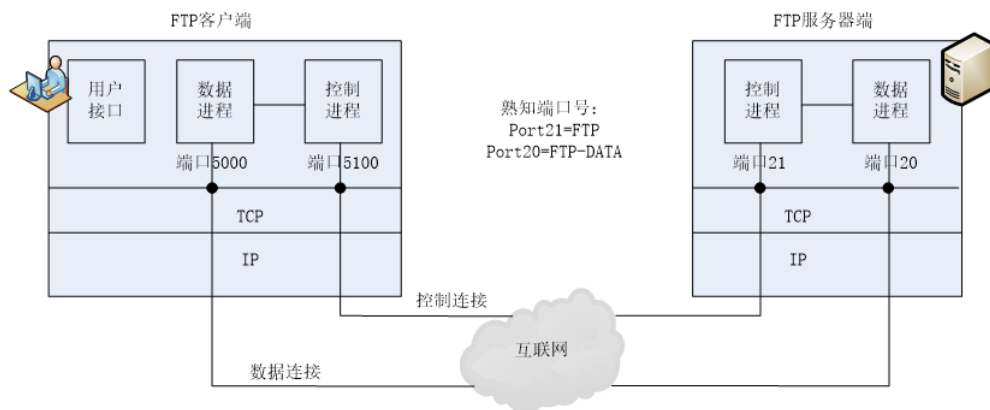
DNS (Domain Name System, 域名系统)将主机域名转化为主机IP地址.

[域名解析算法]



2.4 FTP

[FTP文件服务]



(1) C/S模式, 传输层采用TCP协议.

(2) 模式:

- ① 主动模式: 控制信息使用端口 21, 数据信息使用端口 20.
- ② 被动模式: 控制信息使用端口 21, 数据信息的端口由服务端和客户端协商而定.

3. 传输层

看练习题即可.

4. 网络层

4.1 网络层的功能

[网络层的特点]

- (1) 无连接、不可靠的分组传送服务协议, 尽力而为服务.
 - ① 不维护分组发送后的状态信息, 每个分组传输独立.
 - ② 不保证可靠, 不丢失、按序.
- (2) 点-点的网络层通信协议: 两主机间通信的路径由多个路由器和点-点链路组成.

[网络层的功能]

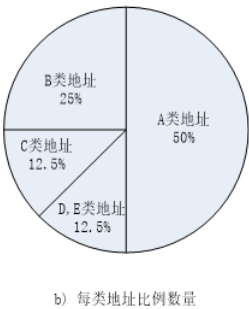
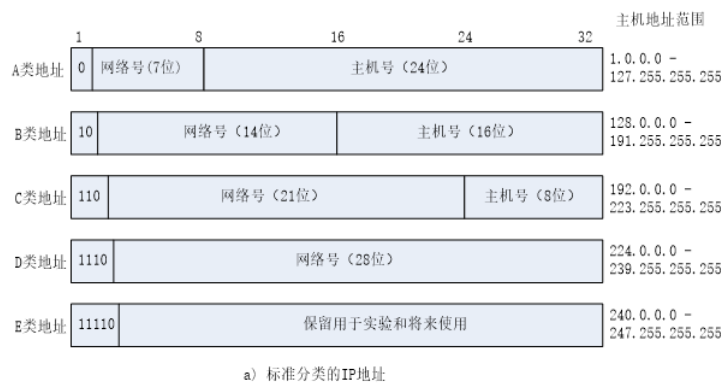
- (1) 逻辑地址分配: IP地址分配.
- (2) 寻址和路由.
- (3) 分段和重组.
- (4) 错误管理: ICMP.
- (5) 为传输层屏蔽物理网络的差异, 掩盖不同物理网和协议的差异, 实现异构互联.

4.2 IPv4

[IP地址的点分十进制表示法]

32 位地址, 用点分十进制 $x.x.x.x$ 表示, 其中 $x \in [0, 255]$.

[标准IP地址的分类]



[划分子网的三级地址结构]



- (1) 三级地址结构: 网络号-子网号-主机号.
- (2) 路由器处理到来的分组, 将子网掩码与IP地址按位与得到网络地址, 即网络号-子网号.
- (3) 子网内部主机寻址, 将子网掩码按位取反后与IP地址按位与得到主机号.

[例] 对一个B类网络 156.26.0.0 进行子网划分, 约有 210 个子网需求.

[解] 因 $2^7 < 210 < 2^8$, 则取 8 位子网号, 子网掩码 255.255.255.0 .

可用的IP地址:

- ① 子网 1: 156.26.1.1 ~ 156.26.1.254 .
- ② 子网 2: 156.26.2.1 ~ 156.26.2.254 .
- ...
- ③ 子网 254: 156.26.254.1 ~ 156.26.254.254 .

[例] 某公司申请了一个C类IP地址 202.60.31.0 , 该共公司有 100 名员工在销售部, 50 名员工在财务部, 50 名员工在设计部. 位这三个部分分别组建子网.

[解]

- (1) 因 $2^6 < 100 < 2^7$, 则子网 1 需 7 位主机号.

取子网掩码 255.255.255.128 , 则:

主机IP地址	11001010 00111100 00011111 00000000	202.60.31.0
子网掩码	11111111 11111111 11111111 10000000	255.255.255.128 或 /25
与值	11001010 00111100 00011111 00000000	202.60.31.0

故子网 1 的地址范围: 202.60.31.1 ~ 202.60.31.126 .

- (2) 因 $2^5 < 50 < 2^6$, 则子网 2 和子网 3 分别需 6 位主机号.

取子网掩码 255.255.255.192 , 则:

主机IP地址	11001010 00111100 00011111 10000000	202.60.31.128
子网掩码	11111111 11111111 11111111 11000000	255.255.255.192 或 /26
与值	11001010 00111100 00011111 10000000	202.60.31.128

故子网 2 的地址范围: 202.60.31.129 ~ 202.60.31.190 .

(3) 下一地址 202.60.31.191 中 $(191)_{10} = (1011\ 1111)_2$, 则第四个字节的主机号(后 6 位)全 1, 留作广播地址.

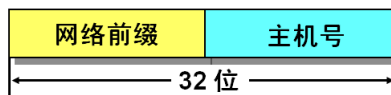
下一地址 202.60.31.192 中 $(192)_{10} = (1100\ 0000)_2$, 则第四个字节的主机号部分全 0, 留作网络地址.

故子网 3 的地址范围: 202.60.31.193 ~ 202.60.31.254 .

4.3 CIDR

[CIDR]

(1) IP地址从用子网掩码的三级编址回到二级编址.



(2) **斜线记法(或CIDR记法)**: 在IP地址后加一个斜线"/"和网络前缀所占的位数, 即三级编址中子网掩码的 1 的个数.

(3) 网络前缀相同的连续的IP地址组成**CIDR地址块**.

[例] 128.14.32.0/20 的地址块有 $2^{32-20} = 2^{12}$ 个地址, 起始地址为 128.14.32.0 .

该地址块的最小地址为 128.14. (0010 0000)₂.0 , 即 128.14.32.0 .

该地址块的最大地址为 128.14. (0010 1111)₂.255 , 即 128.14.47.255 .

[路由聚合]

(1) 一个CIDR地址块可表示多个地址, 这种地址的聚合称为**CIDR聚合**.

(2) CIDR地址的特点: 地址聚合、路由聚合.

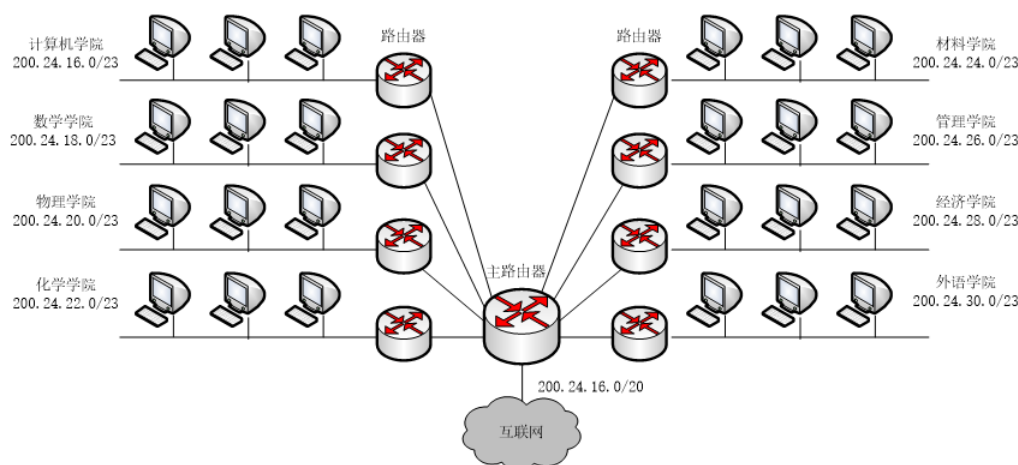
路由聚合也称**构成超网**.

(3) 优点:

- ① 使得路由表中的一个项目可表示多个传统分类地址的路由.
- ② 减小路由器间的路由选择信息的交换, 提高互联网性能.

[例] 若校园网管理中心获得了 200.24.16.0/20 的地址块, 希望将其划分为 8 个等长的较小的地址块, 则可取 12 位主机地址的前 3 位划分地址块, 即:

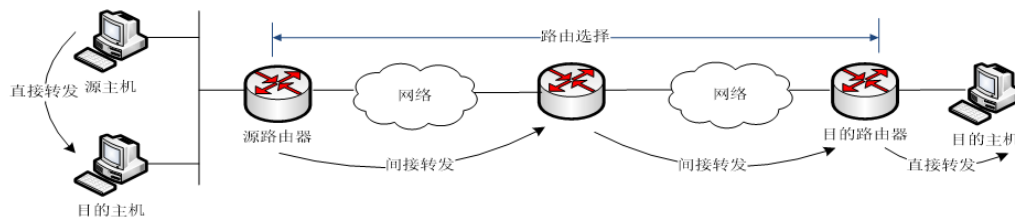
校园网地址	200.24.16.0/20	<u>11001000 00011000 00010000 00000000</u>
学院 1 地址	200.24.16.0/23	<u>11001000 00011000 00010000 00000000</u>
学院 2 地址	200.24.18.0/23	<u>11001000 00011000 00010010 00000000</u>
学院 3 地址	200.24.20.0/23	<u>11001000 00011000 00010100 00000000</u>
学院 4 地址	200.24.22.0/23	<u>11001000 00011000 00010110 00000000</u>
学院 5 地址	200.24.24.0/23	<u>11001000 00011000 00011000 00000000</u>
学院 6 地址	200.24.26.0/23	<u>11001000 00011000 00011010 00000000</u>
学院 7 地址	200.24.28.0/23	<u>11001000 00011000 00011100 00000000</u>
学院 8 地址	200.24.30.0/23	<u>11001000 00011000 00011110 00000000</u>



4.4 路由表与路由汇聚

[分组转发与路由选择]

- (1) 默认路由器(网关): 第一跳路由器.
- (2) 路由问题: 路由转发、路由算法.
- (3) 分组转发:



- ① 直接转发: 到达目的网络.
- ② 间接转发: 查找路由表转发.

(4) 特殊路由:

① 默认路由: 路由选择过程中, 若路由表无明确指明到达目的网络的路由信息, 则将该分组转发到默认路由指定的路由器地址.

② 特定主机路由: 路由表基于网络地址, 但允许为一个特定的主机地址建立路由表项.

[路由选择算法]

(1) 采用表驱动的路由选择算法, 查表决定下一条路由.

(2) 评价路由算法的依据:

- ① 算法需正确、稳定、公平.
- ② 算法尽量简单.
- ③ 算法能适应网络拓扑和通信量的变化.
- ④ 算法需最佳.

(3) 分类:

- ① 静态路由选择: 非自适应.
- ② 动态路由选择: 自适应, 自动更新路由表.

[路由表]

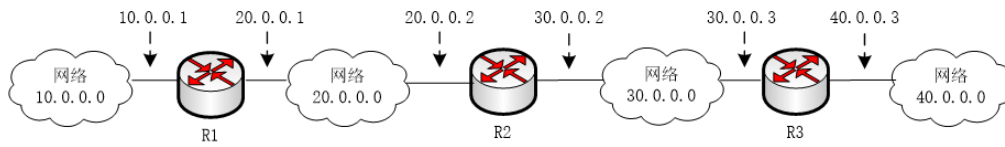
(1) 由路由选择算法创建和维护.

(2) 二元组 (N, R) , 其中 N 为目标网络地址, R 为网络 N 的下一跳路由地址.

(3) 子网路由选择的三元组 (M, N, R) , 其中 M 为目标网络的子网掩码.

进行子网的路由选择时, 取出接收的分组中的目的地址, 与路由表中的子网掩码按位与, 将结果与目的网络比较.

[例] [标准路由选择算法] 写出下图所示的网络中路由器 R_2 的路由表:



[解]

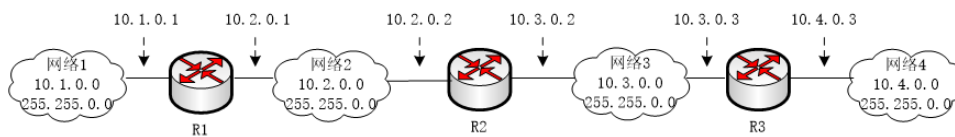
目的网络	下一跳路由
20.0.0.0	直接转发
30.0.0.0	直接转发
10.0.0.0	20.0.0.1
40.0.0.0	30.0.0.3

(1) 网络 20.0.0.0 和 30.0.0.0 直接与 R_2 相连, 则 R_2 收到属于这两个网络的IP分组时直接转发给目的主机即可.

(2) R_2 收到属于网络 10.0.0.0 的分组时, 通过地址为 10.2.0.1 的接口转发给 R_1 .

(3) R_2 收到属于网络 40.0.0.0 的分组时, 通过地址为 30.0.0.3 的接口转发给 R_3 .

[例] [子网的路由选择] 写出下图所示的网络中路由器 R_2 的路由表:



[解]

子网掩码	目的网络	下一跳路由
255.255.0.0	10.2.0.0	直接转发
255.255.0.0	10.3.0.0	直接转发
255.255.0.0	10.1.0.0	10.2.0.1
255.255.0.0	10.4.0.0	10.3.0.3

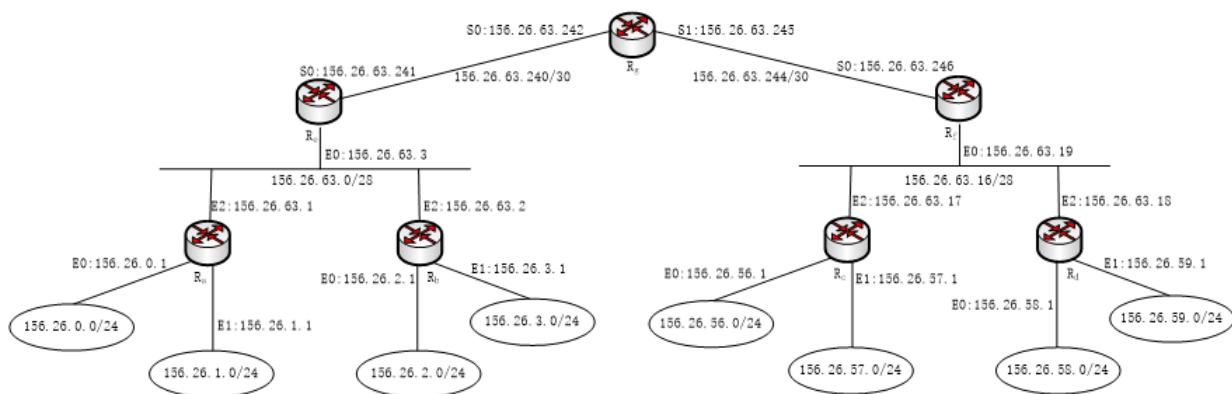
(1) R_2 收到目的网络为 10.2.12.1 的分组时, 将其与子网掩码 255.255.0.0 按位与得到网络地址 10.2.0.0, 对应路由表的第一项, 直接转发给网络 10.2.0.0.

(2) R_2 收到目的网络为 10.4.112.10 的分组时, 将其与子网掩码 255.255.0.0 按位与得到网络地址 10.4.0.0, 对应路由表的第四项, 通过地址为 10.3.0.3 的接口转发给 R_3 .

[IP路由汇聚]

- (1) 路由汇聚是减少路由表项数的重要手段. 路由表项数越少, 路由器查询时间越短, 延时越小.
- (2) 基于CIDR, 网络前缀越长, 主机地址数越少, 寻址目的主机越容易.
- (3) 路由表项的二元组: (网络前缀, 下一跳地址).
- (4) 最长前缀匹配: 选择有最长公共前缀的路由.

[例] 写出下图所示的网络中路由器 R_g 的路由表, 并进行路由汇聚.



[解] 若用静态路由表, 则表项较多, 故用动态路由表.

路由器	输出接口
156.26.63.240/30	S_0 (直接连接)
156.26.63.244/30	S_1 (直接连接)
156.26.63.0/28	S_0
156.26.63.16/28	S_1
156.26.0.0/24	S_0
156.26.1.0/24	S_0
156.26.2.0/24	S_0
156.26.3.0/24	S_0
156.26.56.0/24	S_1
156.26.57.0/24	S_1
156.26.58.0/24	S_1
156.26.59.0/24	S_1

上述路由表的后 8 项为 8 个子网, 考虑合并为两项.

考察地址 156.26.0.0/24 ~ 156.26.3.0/24 的第三个字节的最长公共前缀:

0	<u>00000000</u>
1	<u>00000001</u>
2	<u>00000010</u>
3	<u>00000011</u>

最长公共前缀长为 6, 则这四个地址的网络前缀为 22, 可合并为 156.26.0.0/22.

同理地址 156.26.56.0/24 ~ 156.26.59.0/24 可合并为 156.26.56.0/22.

汇聚后 R_g 的路由表:

路由器	输出接口
156.26.63.240/30	S_0 (直接连接)
156.26.63.244/30	S_1 (直接连接)
156.26.63.0/28	S_0
156.26.63.16/28	S_1
156.26.0.0/22	S_0
156.26.56.0/22	S_1

R_g 收到目的地址为 156.26.2.37 的分组时, 在路由表中查找一个最佳的匹配路由, 一次匹配如下:

156.26.2.37/22	<u>10011100 00011010 00000010</u> 00100101
156.26.0.0/22	<u>10011100 00011010 00000000</u> 00000000

目的地址与地址 156.26.0.0/22 的前缀有 22 位匹配, R_g 将该分组通过接口 S_0 转发.

[注] 汇聚方式不同, 答案可能不同, 但均正确.

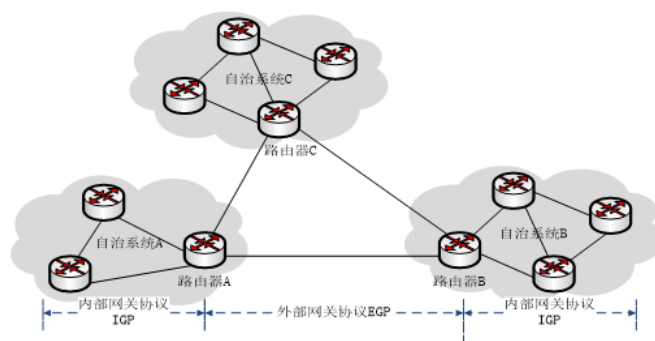
4.5 路由表的建立与路由选择协议

[自治系统]

(1) 互联网由**自治系统**(Autonomous System, AS)互联而成.

(2) 自治系统内的路由选择: 自治管理, 有权自主决定内部采用何路由协议, 与其他AS采用的路由协议无关.

[路由选择协议的分类]



(1) 内部网关协议(Interior Gateway Protocol, IGP): AS内部使用的路由选择协议, 与其它AS使用的路由选择协议无关.

例: RIP、OSPF.

(2) 外部网关协议(External Gateway Protocol, EGP): 连接不同AS边界的路由器间交换信息的协议.

[向量距离路由算法]

(1) 向量距离(Vector-Distance, V-D)路由算法也称Bellman-Ford算法.

(2) 思想: 周期性地向相邻路由器告知本地路由器可到达的网络和到达该网络的距离(跳数), 按最短路原则刷新路由表.

(3) 路由向量 (V, D) 中, V 表示该路由器到目的网络的地址, D 表示距离(跳数).

(4) 过程:

① 初始化路由表.

② 只包含与该路由器有直接相连的路由信息, 距离都为 0, 即本地路由.

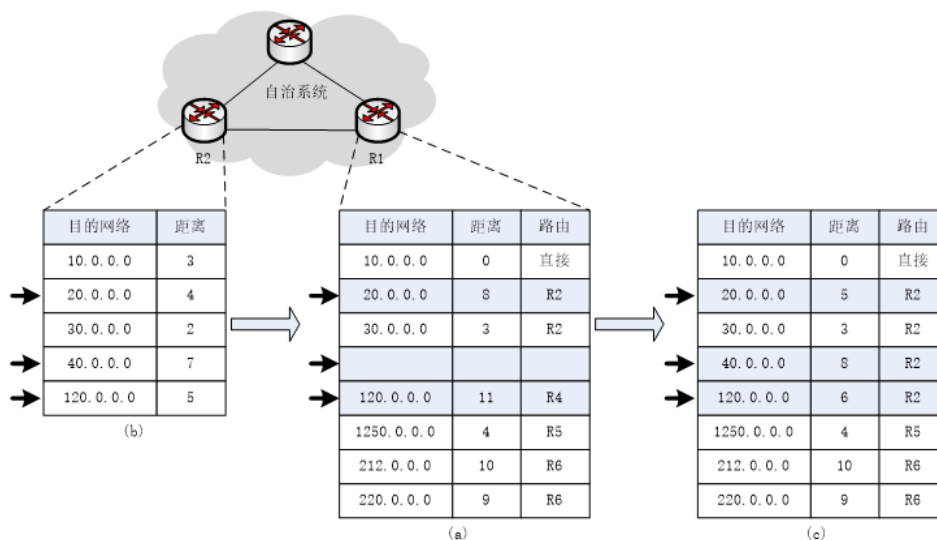
③ 更新路由表, 若得到信息"路由器 X 到目的网络 Y 的距离为 n ":

(i) 若原路由表中无到 Y 的表项, 则增加一个到 Y 的表项.

(ii) 若原路由表中有到 Y 的表项" X 经路由器 Z 到 Y 的距离为 m ", 且 $m > n + 1$, 则更新到 Y 的下一跳路由为 X .

(5) 最大跳数为 15.

[例] 如下图是一个AS. 表(a)是更新前 R_1 的路由表, 表(b)是 R_2 发送的 (V, D) 报文, 求 R_1 更新后的路由表.



[解] 箭头标记的三项需更新.

[路由信息协议]

(1) 路由信息协议(Routing Information Protocol, RIP)在向量距离路由选择算法的基础上, 规定了AS内部路由器间的路由信息交互格式和错误处理方式, 设置了周期更新定时器、延迟定时器、超时定时器、清除定时器.

(2) 周期定时器每 30 s 在相邻路由器间交换一次路由信息.

(3) 若收到同一网络有多条距离相同的路径, 则按"先入为主"的原则, 取用第一条路径的信息, 直至该路径失效或被取代.

(4) 根据向量距离路由选择算法, 若出现开销小的路径则修改表项, 否则一直保持.

(5) 每个路由表项设有超时计时器, 在表项被修改时开始计时, 若 180 s 后未收到刷新信息, 则该路径出现故障, 将该项设置为"无效", 但不删除该项.

[最短路径优先协议]

(1) 最短路径优先协议(Open Shortest Path First, OSPF).

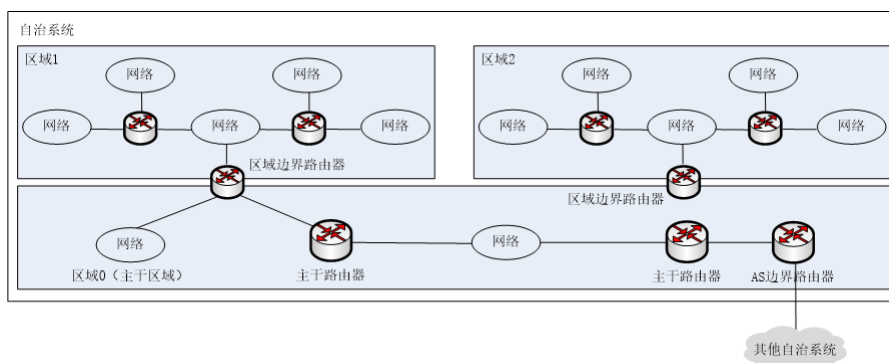
(2) 特点:

① 使用链路状态协议, 而RIP使用距离向量路由协议.

② 路由器周期性地发送链路状态信息(费用、距离、延时、带宽), AS内所有路由器都能形成链路状态数据库(可用端口、已知可达路由、链路状态信息).

③ 链路状态变化时, 用泛洪法向所有路由器发送该信息, 而RIP仅向相邻路由器通报.

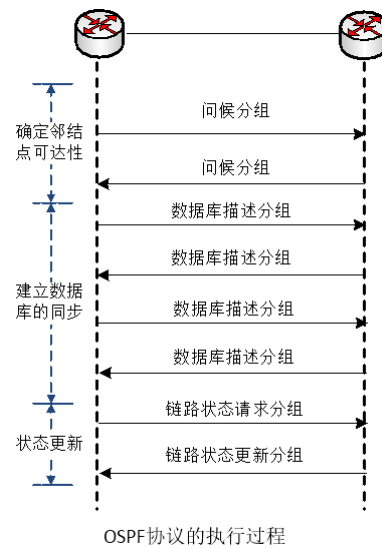
(3) AS的内部结构:



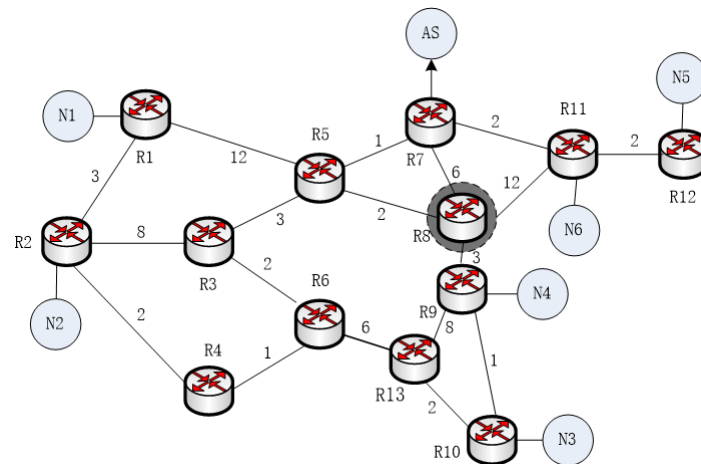
① 主干区域: 由主干路由器组成.

② 区域: 通过区域边界路由器与主干路由器连接.

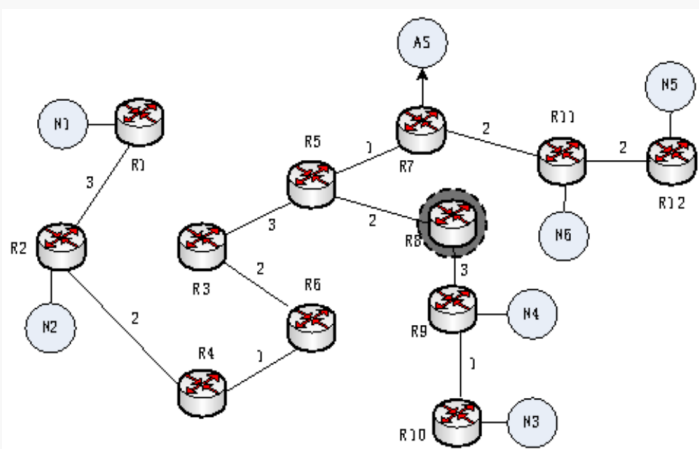
(4) 执行过程:



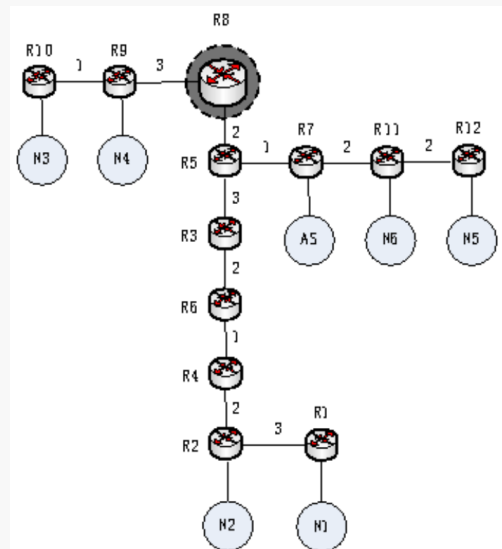
[例] 用OSPF协议求下图所示的AS中从路由器 R_8 出发到网络 $N_1 \sim N_5$ 的最短路.



[解] 以 R_8 为起点做Dijkstra算法即可.



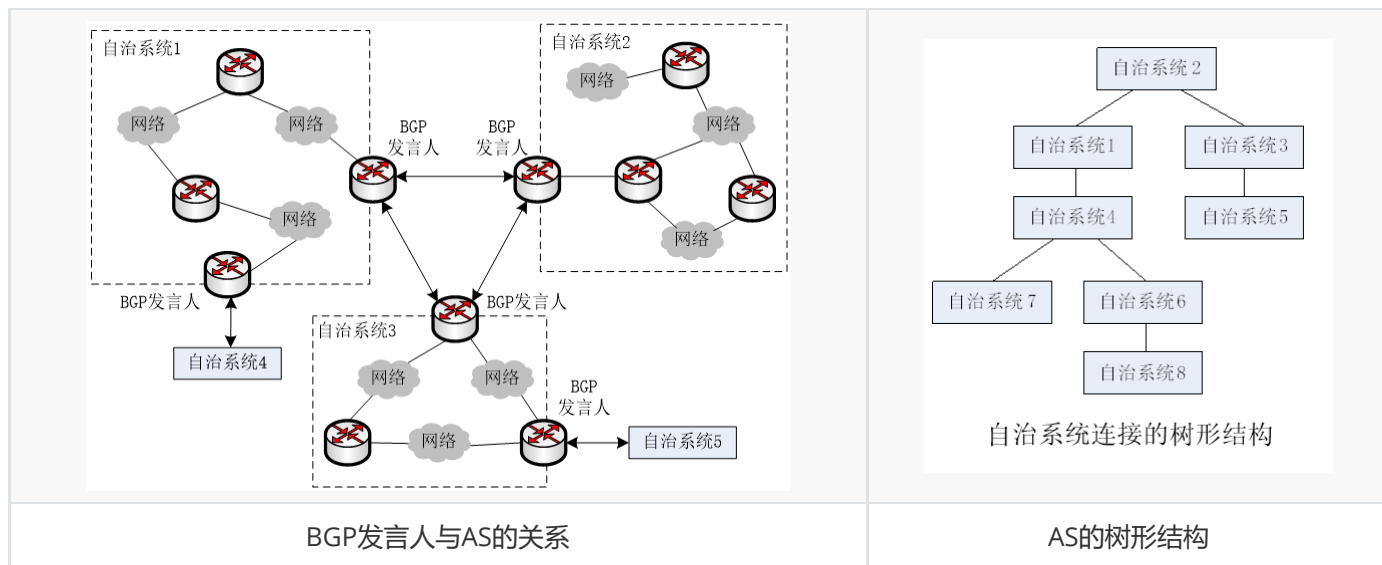
最短路径



以 R_8 为根的最短路径树

[边界网关协议]

- (1) 边界网关协议(Border Gateway Protocol, BGP)在不同AS的路由器间交换路由信息.
- (2) 考虑安全、经济等因素, 寻找较好的路由, 而非最佳路由.
- (3) 使用路径向量路由协议, 不同于RIP和OSPF.
- (4) 每个AS至少有一个路由器作为该AS的**BGP发言人**.



4.6 ICMP

[互联网控制报文协议]

- (1) IP无连接、不可靠, 无差错控制.
- (2) 互联网控制报文协议(Internet Control Message Protocol, ICMP)不能独立于IP单独存在(作为IP的辅助协议), 用于解决IP不可靠的问题, 工作在网络层.
- (3) 需封装为IP分组交给数据链路层.
- (4) 用于IP分组转发过程中检测错误, 由路由器向源主机报告差错原因, 只能报告差错, 不能纠正差错.
- (5) 传输层(高层)要得到可靠传输需用其他机制保证.

[ICMP的报文类型]

(1) 差错报告:

① 路由重定向.

② 数据报超时:

(i) 路由器转发分组时, 若TTL字段值减1后为0, 则丢弃该分组, 并向源主机发送超时报文.

(ii) 分组所有分段在有限时间内(第1个分段到达, 启动计时器)未能到达目的主机(分组无法组装)时, 若多个分组出现该情况, 导致目的主机不能接受新的分组, 出现"死锁", 则丢弃接收到的分段, 并向源主机发送超时报文.

③ 目的站不可达.

④ 源站抑制: 路由器因拥塞而丢弃分组时, 向源主机发送抑制报文.

⑤ 数据报参数错: 路由器发现以上四种差错情况外的错误字段, 则丢弃该分组, 并向源主机发送出错报文.

(2) 查询报文:

① 时间戳请求与应答.

② 回应请求与应答.

③ 地址掩码请求与应答.

④ 路由器查询与通告:

(i) 主机需获取目标路由器是否正常工作, 向目标路由器发送询问与通告报文.

(ii) 主机广播或多播询问报文, 收到询问报文的路由器回复通告报文, 广播其路由信息.

(iii) 无主机询问时, 路由器也可周期性地发送通告报文, 通告自己的存在和它知道的网络中所有的路由器.

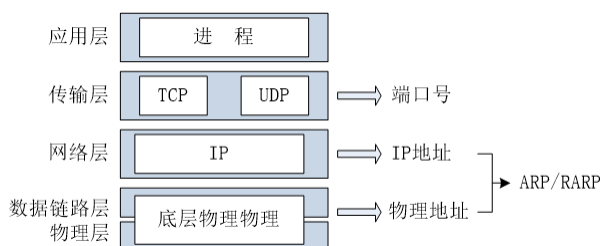
4.7 ARP与RARP

[物理地址]

(1) 网卡物理地址存储器中存储单元对应的实际地址称为物理地址或MAC(Media Access Control Address)地址.

(2) 由 6 个两位的十六进制数用冒号或横杠连接而成, 共 64 位.

[IP地址与物理地址的映射]



(1) IP逻辑地址: 实现不同物理网设备(主机、路由器等)的地址统一.

(2) 分组数据包最终在物理网络上使用物理地址传输.

(3) 地址映射方法:

① 静态映射: 人工维护地址映射表.

不足:

(i) 映射表不能及时反映网络设备变化, 如添加新的主机或路由器.

(ii) IP地址不变的情况下更换网卡, 即MAC地址改变时, 需重新映射.

(iii) 主机物理地址发生变化, 即IP地址改变, 但MAC地址不变时, 需重新映射,

② 动态映射: ARP建立映射.

[地址解析协议]

(1) 正向地址解析协议(Address Resolution Protocol, ARP): 将IP地址解析为MAC地址.

(2) 反向地址解析协议(Reverse Address Resolution Protocol, RARP): 将MAC地址解析为IP地址.

(3) 解析过程: 用ARP请求分组、应答分组.

4.8 IPv6

[IPv6地址]

- (1) 有 128 位, 用冒分十六进制表示.
- (2) 零位压缩法: 将中间包含的很长一段 0 压缩为双冒号 "::".

要点:

- ① 冒号在一个地址中只能出现一次.
- ② 不能压缩一个位段的有效 0.

[例] IPv6地址 $21DA : 0000 : 0000 : 0000 : 02AA : 000F : FE08 : 9C5A$ 可压缩为:

$21DA : 0 : 0 : 0 : 2AA : F : FE08 : 9C5A$, 可进一步压缩为: $21DA : 2AA : F : FE08 : 9C5A$.

[例] 不能将IPv6地址 $FF02 : 30 : 0 : 0 : 0 : 0 : 5$ 压缩为 $FF2 : 3 :: 5$, 而应压缩为 $FF02 : 30 :: 5$.

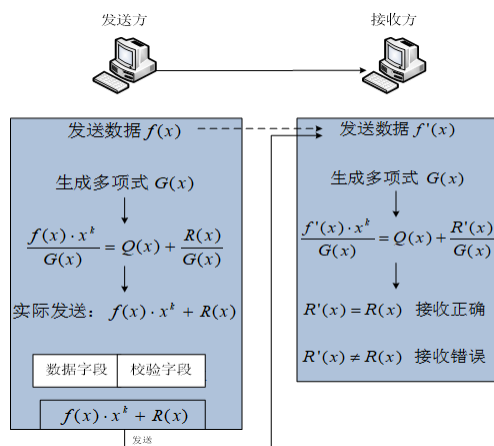
5. 数据链路层

5.1 差错控制

[循环冗余码工作原理]

(1) 循环冗余码(Cyclic Redundancy Code, CRC)又称多项式码, 如数据 1011011 对应的多项式为 $f(x) = x^6 + x^4 + x^3 + x + 1$.

(2) 过程:



① 发送方生成数据多项式 $f(x) \cdot x^k$, 其中 k 为生成多项式的次数. 如生成多项式 $G(x) = x^4 + x^2 + x + 1$ 的 $k = 4$.

对二进制乘法, $f(x) \cdot x^{12}$ 等价于将比特流左移 12 位后放入余数.

② $f(x) \cdot x^k$ 在模 2 意义下除以(即异或)生成多项式 $G(x)$ 得到余数多项式, 即 $\frac{f(x) \cdot x^k}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$.

③ 将 $f(x) \cdot x^k + R(x)$ 作为整体, 发送给接收方.

若要发送的数据 $D = f(x)$ 有 d 位, CRC校验 $R = R(x)$ 有 r 位, 则发送的 DR 有 $(d + r)$ 位.

④ 接收方对接收到的数据对应的多项式 $f'(x)$ 做同样的运算 $\frac{f'(x) \cdot x^k}{G(x)} = Q(x) + \frac{R'(x)}{G(x)}$, 得到余数多项式 $R'(x)$.

⑤ 根据 $R'(x)$ 与 $R(x)$ 是否相等来判断是否出错.

[例] CRC检错.

(1) 发送数据位: 1010001, 有 7 位.

(2) 生成多项式: 10111, 次数 $k = 4$.

(3) 将数据乘 2^4 , 即左移 4 位, 得到乘积 10100010000.

(4) 乘积在模 2 意义下除以生成多项式得到余数 1101, 计算过程如下:

$$\begin{array}{r}
 \begin{array}{l} G(x) \rightarrow 10111 \end{array} \overline{) \begin{array}{l} 1001111 \leftarrow Q(x) \\ 10100010000 \leftarrow f(x) \cdot x^k \\ 10111 \\ 11010 \\ 10111 \\ 11010 \\ 10111 \\ 11010 \\ 10111 \\ 11010 \\ 10111 \\ 1101 \leftarrow R(x) \end{array}
 \end{array}$$

(5) 将余数比特流附加到发送数据的尾部, 即:

$$\begin{array}{ccc}
 \boxed{1010001} & \boxed{1101} & \\
 \text{发送数据} & \text{CRC校验码} & \\
 \hline
 \boxed{10100011101} & & \\
 \text{发送比特流} & & \\
 \text{带CRC校验码的发送数据比特流} & &
 \end{array}$$

(6) 若传输过程无差错, 则接受方收到的数据比特流(含CRC码)能被相同的生成多项式整除.

[海明码]

(1) 又称汉明码, 是线性调制码.

(2) 用奇偶校验位, 在数据后增加一些比特, 验证数据的有效性.

a_{n-1}	a_{n-2}	a_r	a_{r-1}	a_0
k位信息位				r位监督位		
<div style="display: flex; align-items: center; justify-content: space-between;"> ← $n=k+r$ → </div>						

(3) 用一个以上的校验位, 不仅能验证数据是否有效, 还能在出错时指明错误位置.

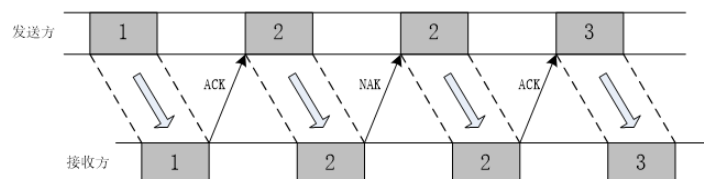
① 为检测 d bit 的错误, 需 $(d + 1)$ 的海明距离.

② 为纠正 d bit 的错误, 需 $(2d + 1)$ 的海明距离.

5.2 流量控制与拥塞控制

[单帧停止等待协议]

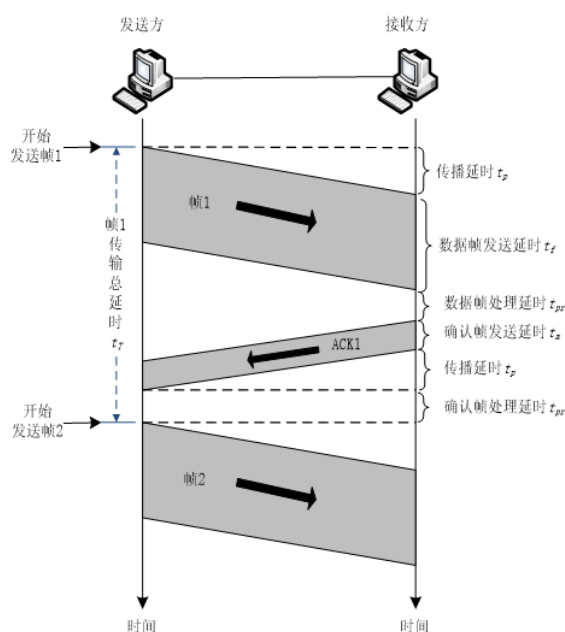
(1) 思想: 发送方每发送一帧需等待确认帧返回, 再发送下一帧. 若发送方收到否认帧, 则重新发送出错的数据帧.



(2) 优点: 协议简单, 实现容易.

(3) 缺点: 帧传输效率低下.

(4) 效率分析:



① 帧传输总延时 $t_T = t_p + t_f + t_{pr} + t_a + t_p + t_{pr} = 2t_p + 2t_{pr} + t_f + t_a$.

因节点对帧的处理延时 t_{pr} 小于帧发送延时 t_r 和传播延时 t_p , 故可忽略.

确认帧常短, 其发送延时 t_a 可忽略. 故 $t_T \approx t_f + 2t_p$.

② 帧传输效率 $U = \frac{t_f}{t_T} = \frac{t_f}{t_f + 2t_p}$. 设 $\alpha = \frac{\text{传播延时}}{\text{发送延时}} = \frac{t_p}{t_f}$, 则 $U = \frac{1}{1 + 2\alpha}$.

[例] 设电磁波在电缆中的传播速度为 $2 \times 10^8 \text{ m/s}$, 连接收发双方的传输介质长 1000 m , 传播延时 $t_p = 5 \times 10^{-6} \text{ s}$, 节点的发送速率为 10 Mbps .

(1) 若数据帧长 100 位, 求帧传输效率.

(2) 若数据帧长 1000 位, 求帧传输效率.

[解]

$$(1) \text{ 发送延时 } t_f = \frac{100}{10 \times 10^6} \text{ s} = 1 \times 10^{-5} \text{ s}.$$

$$\alpha = \frac{t_p}{t_f} = \frac{5 \times 10^{-6}}{1 \times 10^{-5}} = 0.5, \text{ 帧传输效率 } U = \frac{1}{1 + 2\alpha} = 0.5.$$

$$(2) \text{ 发送延时 } t'_f = \frac{1000}{10 \times 10^6} \text{ s} = 1 \times 10^{-4} \text{ s}.$$

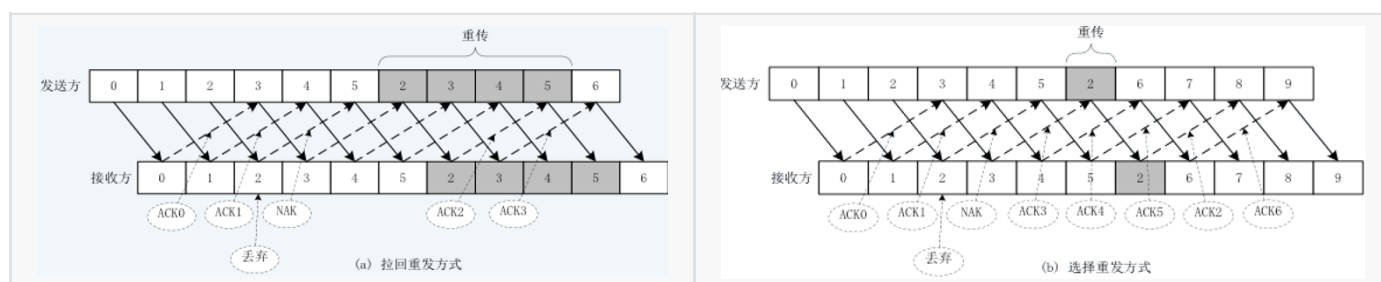
$$\alpha' = \frac{t_p}{t'_f} = 0.05, \text{ 帧传输效率 } U' = \frac{1}{1 + 2\alpha'} \approx 0.91.$$

[注] 传播延时一定时, 发送的数据帧越长, 发送延时越大, α 越小, 传输效率越高. 故在 $(t_f + 2t_p)$ 时间内不出错的前提下, 连续发送多个帧可提高传输效率.

[连续发送ARQ协议]

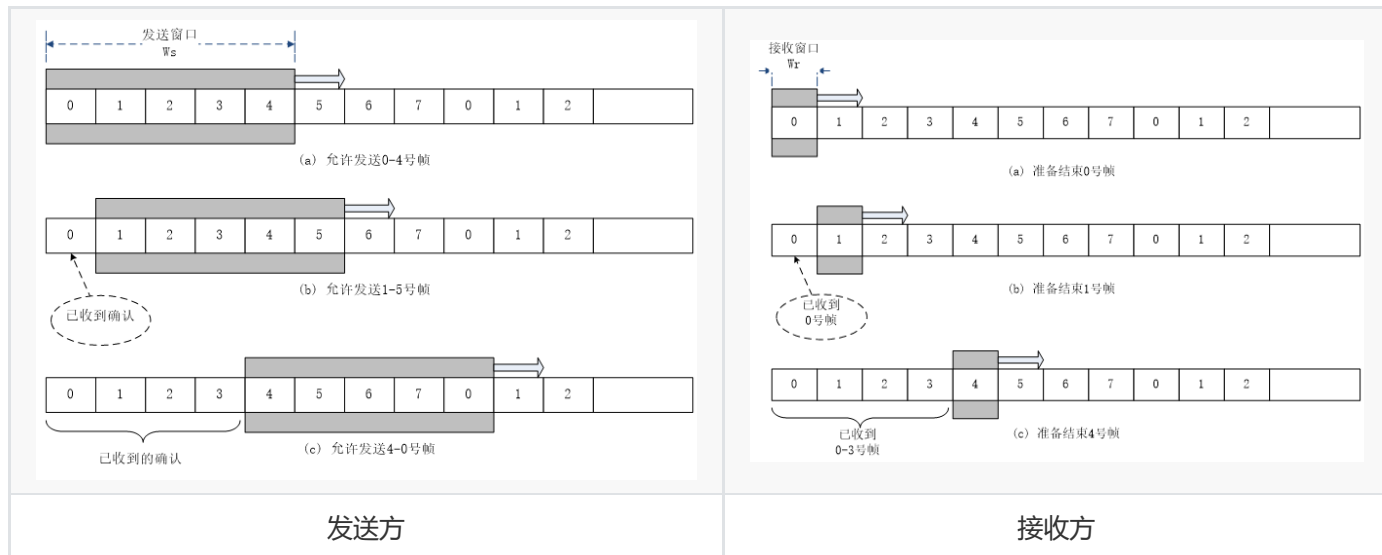
(1) 发送方连续向接收方发送数据帧, 接收方校验收到的数据帧后, 向发送方返回相应的应答帧.

(2) 发送方维持一个一定大小的发送窗口, 位于发送窗口内的所有分组都可连续发送, 中途无需等待对方确认.



[滑动窗口协议]

- (1) 发送方连续发送帧的数量受接收方控制, 根据接收方的接受缓冲区的剩余空间来限制发送帧的节奏, 避免拥塞.
- (2) 滑动窗口: 限制已发送但未确认的数据帧的数量, 以控制流量.
- (3) 设置发送窗口 W_s 对发送方进行流量控制, W_s 值表示未收到接收方的确认信息时发送方最多可发送的数据帧数.
- 特别地, $W_s = 1$ 时为单帧停止等待协议.



- (4) 发送序号与发送窗口最大值的关系: 设接受窗口 $W_r = l$, 发送窗口 $W_s \leq 2^n - l$ 时, ARQ协议才能正确运行.

[例] 设滑动窗口协议中 $n = 3$, 即发送序号用 3 bit 编码.

设发送窗口 $W_s = 8$, 发送方发完 0 ~ 7 的 8 个帧后发送窗口被填满, 发送暂停. 有如下两种情况:

- ① 所有确认帧都正确到达发送方, 则发送方发送 8 个新的数据帧, 编号 0 ~ 7.
- ② 所有确认帧都丢失, 发送方重发 8 个旧的数据帧, 编号 0 ~ 7.

则 $W_s = 8$ 时接收方无法判定新旧数据帧.

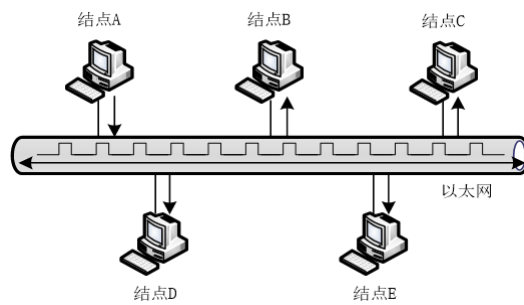
5.3 以太网

[载波侦听多路访问/冲突检测技术]

- (1) 载波侦听多路访问/冲突检测技术(CSMA/CD):

- ① 载波侦听: 监听总线占用.
- ② 多路访问: 平等访问总线.
- ③ 冲突检测: 冲突不可避免.

(2) CSMA/CD的工作特点: 先听后发、边听边发、冲突停止、延迟重发.



(3) 数据发送流程:

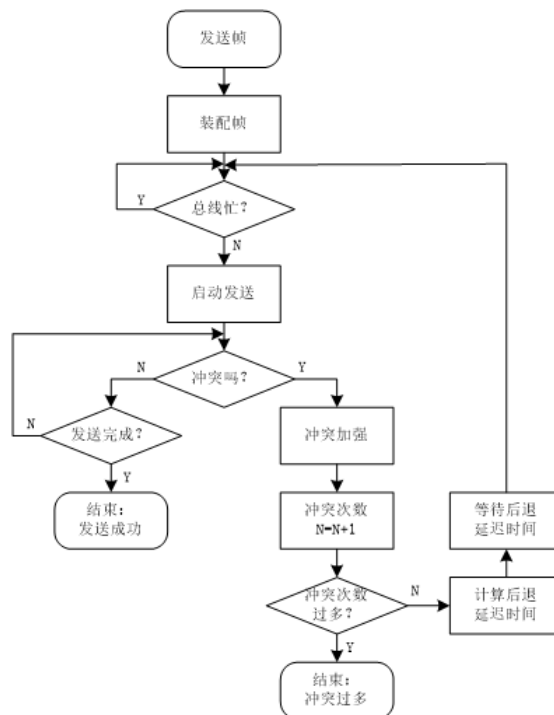
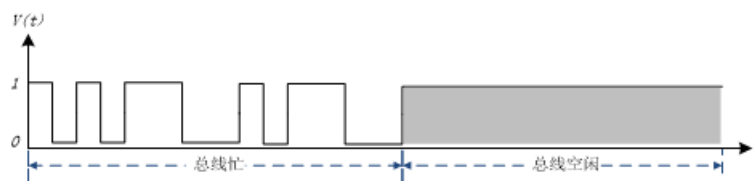
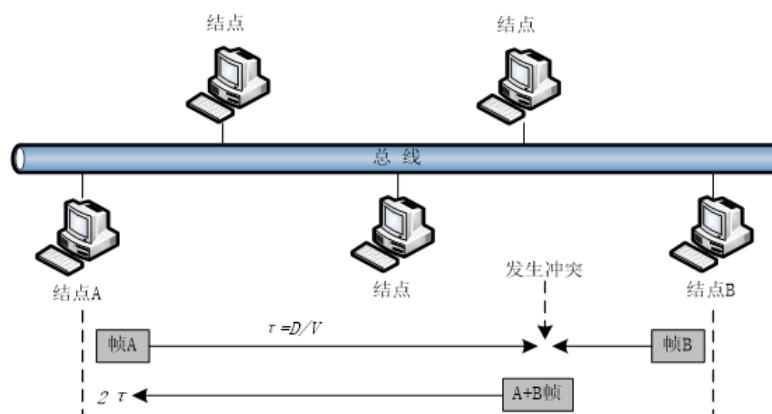


图8-32 以太网结点数据发送流程

(4) 总线忙闲状态判断:



(5) 冲突窗口:



6. 物理层

6.1 脉冲编码调制

[脉冲编码调制]

(1) 脉冲编码调制(Pulse Code Modulation, PCM)将模拟信号数字化.

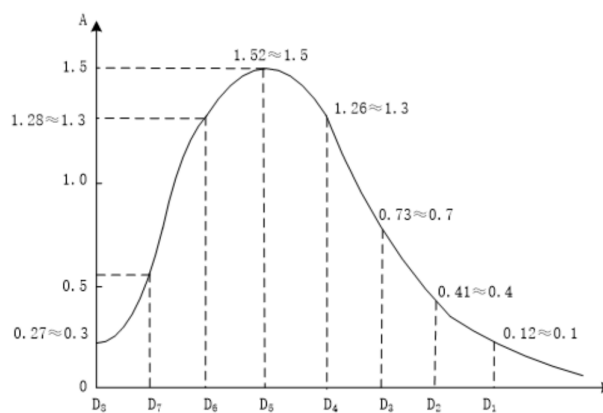
(2) PCM的操作:

① 采样: 每隔一段时间, 取模拟信号电平.

采样频率 $f = \frac{1}{T} \geq 2B$, 其中 T 为采样周期, B 为信道带宽.

或 $f = \frac{1}{T} \geq 2f_{\max}$, 其中 f_{\max} 为信道允许通过的信号最高频率.

② 量化:



③ 编码:

样本	量化级	二进制编码	编码信号
D ₁	1	0001	
D ₂	4	0100	
D ₃	7	0111	
D ₄	13	1101	
D ₅	15	1111	
D ₆	13	1101	
D ₇	6	0110	
D ₈	3	0011	

6.2 数据传输速率

[数据传输效率]

(1) 数据传输速率指每秒传输的构成数据代码的二进制比特数, 单位为比特每秒(bit/s), 记作 bps .

节点向传输介质发送数据的速率称为发送速率.

(2) 二进制数据的数据传输速率 $S = \frac{1}{T}$ bps, 其中 T 为发送每个比特所需的时间.

(3) 常用单位:

① $1 \text{ Kbps} = 1 \times 10^3 \text{ bps}$.

② $1 \text{ Mbps} = 1 \times 10^6 \text{ bps}$.

③ $1 \text{ Gbps} = 1 \times 10^9 \text{ bps}$.

④ $1 \text{ Tbps} = 1 \times 10^{12} \text{ bps}$.

存储计算用二进制, 如 $1 \text{ Kb} = 1024 \text{ b}$, 而计算速率用十进制.

(4) 调制速率(也称波特率) B 用于模拟信号传输, 单位为 $1/\text{s}$, 称为 1 baud .

数据传输速率 S (bps) 与波特率 B (baud) 的关系: $S = B \cdot \log_2 k$, 其中 k 为多相调制的相数.

[例] 设以太网的传输速率为 10 Mbps , 帧为 1500 Byte . 求以太网发送一帧的延时.

[解] $t = \frac{1500 \times 8 \text{ bit}}{10 \times 10^6 \text{ bps}} = 1.2 \text{ ms}$.

[例] 求十六相调制($k = 16$)中波特率 1200 baud 对应的数据传输速率(bps).

[解] $S = B \cdot \log_2 k = 1200 \times 4 \text{ bps} = 4800 \text{ bps}$.
