

一般不使用的特殊IP地址				
网络号	主机号	作为源地址	作为目的地址	代表的意思
0	0	可以	不可	在本网络上的本主机 (DHCP协议)
0	host-id	可以	不可	在本网络上的某台主机host-id
全1	全1	不可	可以	只在本网络上进行广播 (各路由器均不转发)
net-id	全1	不可	可以	对net-id上的所有主机进行广播
127	非全0或全1	可以	可以	用于本地软件环回测试

划分子网的IPv4地址

为什么需要划分子网？

比如一个单位有300台主机，此时需要申请一个B类网络地址，很容易得知，分配出去300个IP地址后，申请得到的B类网络还剩下很多IP地址。

当单位扩大规模，需要再添加两个子网的时，又要为这两个子网分别申请B类地址，这会花费一些不必要的支出，实际上子网1申请到的B类地址还有很多没用到，我们希望能够将这些剩余地址应用到其他子网中(节约地址)。

如何实现子网划分？

32比特的子网掩码可以表明分类IP地址的主机号部分被借用了几个比特作为子网号

- 子网掩码使用连续的比特1来对应网络号和子网号(子网号来自原先的一部分主机号)
- 子网掩码使用连续的比特0来对应主机号
- 将划分子网的IPv4地址与其相应的子网掩码进行(逻辑与运算)【即掩码是1的部分】就可得到IPv4地址所在子网的网络地址

无分类编址的IPv4地址

为什么需要？

划分子网在一定程度上缓解了因特网发展中遇到的困难，但是数量巨大的C类网因为其地址空间太小并没有得到充分使用，而因特网的IP地址仍在加速消耗，整个IPv4地址空间面临全部耗尽的威胁

1993年，IETF发布了无分类域间路由选择CIDR

- CIDR 消除了传统的 A 类、B 类和 C 类地址，以及划分子网的概念
- CIDR 可以更加有效地分配 IPv4 地址空间
- CIDR 使用“斜线记法”，或称 CIDR 记法。即在 IPv4 地址后面加上斜线“/”，在斜线后面写上网络前缀所占比特数量

【举例】

128.14.35.7 / 20

网络前缀占用的比特数量：20
主机编号占用的比特数量：32-20=12

- CIDR 实际上是将网络前缀都相同的连续 IP 地址组成一个“CIDR 地址块”

路由聚合

如路由器 A 上连接同一网络的多台主机，路由器 B 与 A 相连。若 A 将所有主机的具体 IP 地址都报给 B，则路由器 B 中会增加多项路由条目。可实际上 B 向 A 中任意一个主机转发数据时都是走同一个端口，因此我们可以将这些网络的共同前缀提取出来成为新的网络号，同时将剩余主机号置 0 放入路由器 B 中

如：A 连接了 172.1.4.0/25 和 172.1.7.0/24，则提取公共前缀聚合后变为 172.1.4.0/22

4. IP 数据报发送转发过程

同一个网络之间的主机可以直接通信，不同网络之间的主机通信需要路由器中转

源主机如何判断目的主机是否和自己在同一个网络中？

将自身的 IP 地址与子网掩码相与得到自身的网络号 1，再将目的 IP 地址与自身子网掩码相与得到网络号 2。若两个网络号相等，则说明处在同一个网络

主机如何知道应该把 IP 数据报交给哪个路由器进行转发呢？

通过设置默认网关。所谓默认网关，即当路由表中查不到数据时会将数据发往的路由器端口 IP 地址

静态路由配置

静态路由配置是指用户或网络管理员使用路由器的相关命令给路由器人工配置路由表

- 这种人工配置方式简单、开销小。但不能及时适应网络状态的变化。**一般只在小规模网络中采用**
- 可能由于：①配置错误 ②聚合了不存在的网络 ③网络故障 而出现**路由环路错误**
- **默认路由为：0.0.0.0/0**。其作用是当路由表不知道往哪里转发时，就会往**默认路由指定的下一跳位置转发**，根据网络号最长匹配原理，**默认路由**网络号长度为0，因此一定是最后一个被匹配的条目
- **特定主机路由：具体主机IP/32**。网络号前缀长度32保证了这是第一个被匹配的静态条目

如何防止错误路由导致IP数据报永久兜圈？

1. 在IP数据报首部设置生存时间TTL字段
 - IP数据报进入路由器后，TTL字段的值减1。若TTL的值不等于0，则被路由器转发，否则被丢弃
2. 对于聚合后或由于网络故障而不存在的路由条目设置**黑洞路由**
 - 所谓**黑洞路由**，即路由器应该丢弃的路由

5. 路由选择协议

因特网所采用的的路由选择协议主要特点

- **自适应**：动态路由选择，能较好地适应网络状态的变化
- **分布式**：路由器之间交换路由信息
- **分层次**：将整个因特网划分为许多较小的自治系统AS

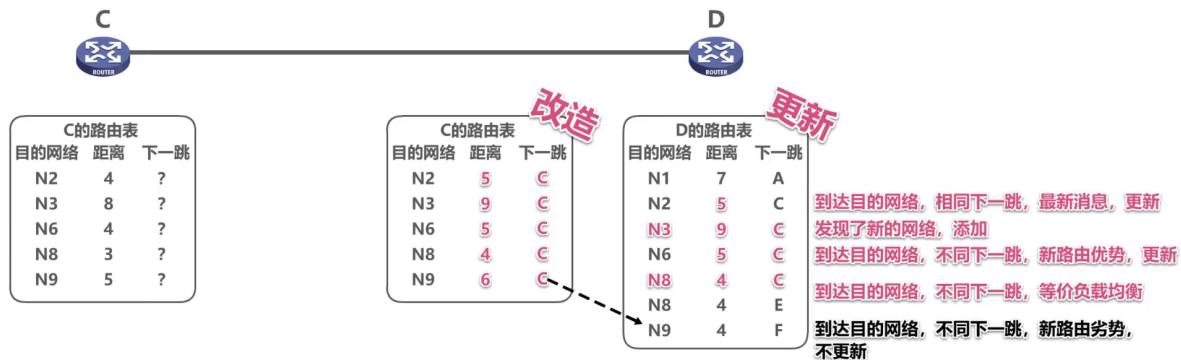
路由信息协议RIP

RIP使用跳数作为度量来衡量到达目的网络的距离

- 路由器到直连网络的距离定义为1
- 路由器到非直连网络的距离定义为所经过的路由器数+1

- 允许一条路径最多只能包含 15 个路由器。"距离"等于 16 时相当于不可达。因此，RIP 只适用于小型互联网
- RIP 认为好的路由就是"距离短"的路由，也就是所通过路由器数量最少的路由
- 当到达同一目的网络有多条"距离相等"的路由时，可以进行等价负载均衡
- RIP 包含以下三个要点
 - 仅和相邻路由器交换信息
 - 交换的是各自路由表的信息
 - 周期性交换信息

【举例】RIP 的路由条目的更新规则



RIP 存在的问题

存在"坏消息传播很慢"的问题，又称为**路由环路(两个路由器相互学习错误路由，造成循环)**或**距离无穷计数问题**，这是距离向量算法的一个固有问题，可以采取多种措施减少出现该问题的概率或减小该问题带来的危害

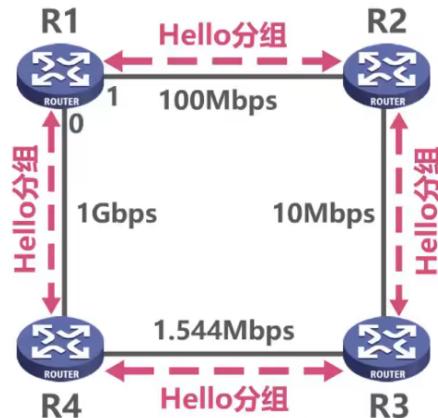
- 限制最大路径距离为 15 (16 表示不可达)
- 当路由表发生变化时就立即发送更新报文(即"**触发更新**")，而不是周期性发送
- 让路由器记录收到某特定路由信息的接口，而不让同一路由信息再通过此接口反方向传送(即"**水平分割**" "**毒性反转**")

开放最短路径优先OSPF

简单来说就是得到一个带权有向图，以当前路由器为起点，通过迪杰斯特拉算法得到到达某个点的最短路径

- OSPF是基于链路状态的，而不像RIP那样是基于距离向量的
- OSPF采用SPF算法计算路由，而不像RIP那样是基于距离向量的
- OSPF不限制网络规模，更新效率高，收敛速度快
- 链路状态是指本路由器都和哪些路由器相邻，以及相应链路的“代价”
 - “代价”的意思是费用、距离、时延、带宽等
- OSPF相邻路由器之间通过交互问候(Hello)分组，建立和维护邻居关系
 - Hello分组封装在IP数据报中，发往组播地址224.0.0.5
 - 发送周期为10秒
 - 40秒未收到来自邻居路由器的Hello分组，则认为该邻居路由器不可达

R1的邻居表			
邻居ID	接口	“死亡”倒计时	
R2	1	36秒	
R4	0	18秒	



- 使用OSPF的每个路由器都会产生链路状态通告LSA，包含以下内容
 - 直连网络的链路状态信息
 - 邻居路由器的链路状态信息
- LSA被封装在链路状态更新分组LSU中，采用洪泛法发送
- 使用OSPF的每个路由器都有一个链路状态数据库LSDB，用于存储LSA
- 通过各路由器洪泛法发送封装有自己LSA的LSU分组，各路由器的LSDB最终达到一致
- 使用OSPF的各路由器基于LSDB进行最短路径优先SPF计算，构建出各种到达其他各路由器的最短路径，即构建各自的路由表

OSPF五种分组

1. 问候(Hello)分组

用来发现和维护邻居路由器的可达性

2. 数据库描述(Database Description)分组

向邻居路由器给出自己的链路状态数据库中的所有链路状态项目的摘要信息

3. 链路状态请求信息(Link State Request)分组

向邻居路由器请求发送某些链路状态项目的详细信息

4. 链路状态更新(Link State Update)分组

路由器使用这种分组将其链路状态进行洪泛发送，即用洪泛法对全网更新链路状态

5. 链路状态确认(Link State Acknowledgement)分组

这是对链路状态更新分组的确认分组

OSPF基本工作过程

- 相邻路由器之间周期性发送**问候分组(Hello)**，以便建立和维护邻居关系
- 建立邻居关系后，**给邻居路由器发送数据库描述分组(DD)**，也就是将自己链路状态数据库中的所有链路状态项目的摘要信息发送给邻居路由器
- 收到数据库描述分组后，若发现自己缺少其中某些链路状态项目，则会**发送链路状态请求分组(LSR)**。
- 对方收到链路状态请求分组后，则会将其所缺少的链路状态项目的详细信息封装在**链路状态更新分组(LSU)**中发送回去
- 收到链路状态更新分组后，将这些信息添加到自己的链路状态数据库中

邻居关系的建立

一条总线上有多台主机，则它们互为邻居，因此每个路由器都要向其他路由器发送问候分组和链路状态更新分组。为了减少所发送分组的数量，则需要用相应方法对邻居关系进行删减

OSPF采用选举**指定路由器DR(Designated Router)**和**备用的指定路由器BDR(Backup Designated Router)**的方法减少邻居数目

- 所有的非 DR/BDR 只与 DR/BDR 建立邻居关系
- 非 DR/BDR 之间通过 DR/BDR 交换信息
- 当 DR 失效时由 BDR 顶上

边界网关协议BGP

尽力寻找一条能够到达目的网络且比较好的路由(不兜圈子)，而并非找最佳路由



内部网关协议IGP(如路由信息协议RIP或最短路径优先OSPF)

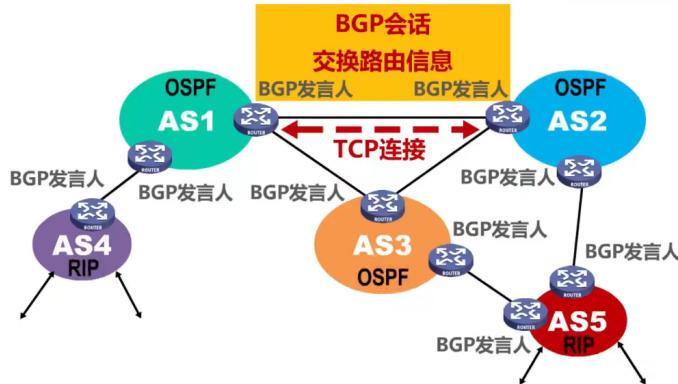
- 设法使分组在一个自治系统内尽可能有效地从源网络传输到目的网络
- 无需考虑自治系统外部其他方面的策略

外部网关协议EGP(如边界网关协议BGP)

- 在不同自治系统内，度量路由的"代价"(距离，宽带，费用等)可能不同。因此，对于自治系统之间的路由选择，使用"代价"作为度量来寻找最佳路由是不行的
 - 比如 A 系统路由选择度量是距离， B 系统是带宽.....那么 A 到系统 E 的路由怎样走最好呢？由于没有统一度量，所以不能直接得到最佳路由
- 自治系统之间的路由选择必须考虑相关策略(政治、经济、安全等)
 - 如中国的数据报尽量要绕开美国的自治系统

工作原理

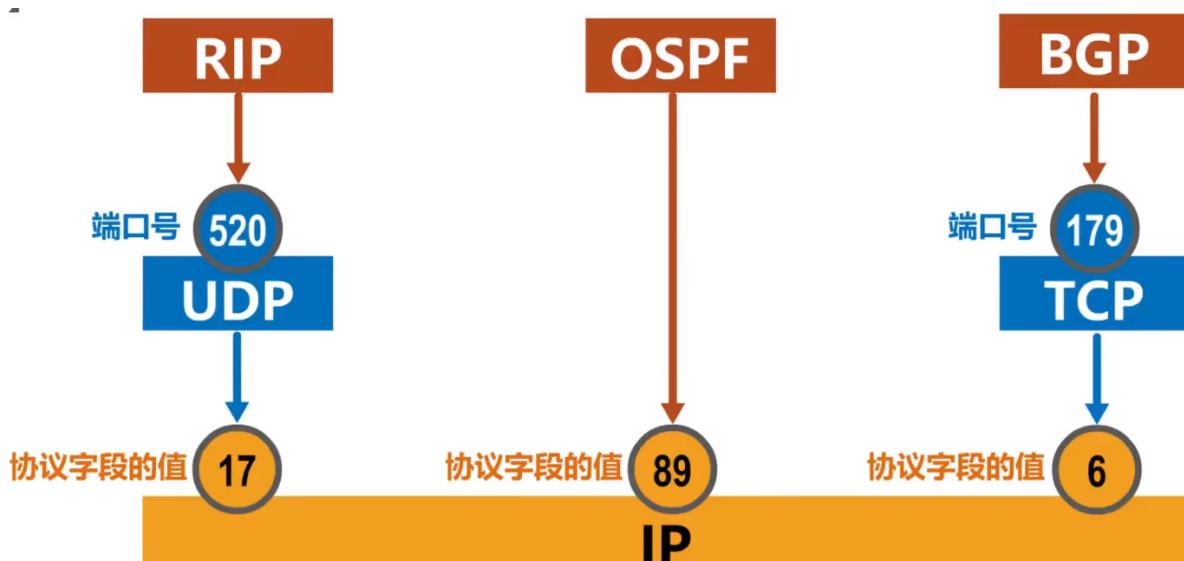
- 在配置BGP时，每个自治系统的管理员要选择至少一个路由器作为该自治系统的“**BGP发言人**”
- 不同自治系统的BGP发言人要交换路由信息，首先必须建立**TCP连接**，端口号为179
 - 在此TCP连接上交换BGP报文以建立**BGP会话**
 - 利用BGP会话**交换路由信息**（例如，增加新的路由，或撤销过时的路由，以及报告出错的情况等）
 - 使用TCP连接交换路由信息的两个BGP发言人，彼此称为对方的**邻站**（neighbor）或**对等站**（peer）
- BGP发言人除了运行BGP外，还必须运行自己所在自治系统所使用的内部网关协议IGP，例如OSPF或RIP。



- **BGP发言人交换网络可达性的信息**(要到达某个网络所要经过的一系列自治系统)
- 当**BGP发言人互相交换了网络可达性的信息后**，各**BGP发言人**就根据所采用的策略从收到的路由信息中**找出到达各自治系统的较好路由**。也就是构造出树形结构(防环路)的自治系统连通图

BGP-4的4中报文

1. **OPEN(打开)报文**: 用来与相邻的另一个**BGP发言人**建立关系，使通信初始化
2. **UPDATE(更新)报文**: 用来通告某一路由的信息，以及列出要撤销的多条路由
3. **KEEPALIVE(保活)报文**: 用来周期性地证实邻站的连通性
4. **NOTIFICATION(通知)报文**: 用来发送检测到的差错



6. IPv4数据报头部格式



- 版本

占4比特，表示IP协议版本。通信双方使用的IP协议版本必须一致。

目前广泛使用的IP协议版本为号为4(IPv4)

- 首部长度

占4比特，表示IP数据报首部长度。该字段取值以4字节为单位

最小十进制取值为5，表示IP数据报首部只有20字节【4字节单位，所以取值5对应20字节】固定部分

最大十进制取值为15，表示IP数据报首部包含20字节固定部分和最大40字节可变部分

- 可选字段

长度从1到40个字节不等。用来支持排错、测量及安全等措施

可选字段增加了IP数据报的功能，但这同时也使得IP数据报的首部长度成为可变的。这就增加了每一个路由器处理IP数据报的开销。实际上可选字段很少被使用

- 填充字段

确保首部长度为4字节长度的整数倍，使用全0进行填充

- 区分服务

占8比特，利用该字段的不同数值可提供不同等级的服务质量，只有在使用区分服务时，该字段才起作用。一般情况下不使用该字段

- 总长度

占16比特，表示IP数据报的总长度【首部+数据载荷】，最大取值为十进制65535，以字节为单位

- 标识

占16比特，属于同一个数据报的各分片数据报应该具有相同的标识
【可理解为ID】

IP软件维持一个计数器，每产生一个数据报，计数器值+1，并将此值赋给标识字段

- 标志

占3比特，各比特含义如下

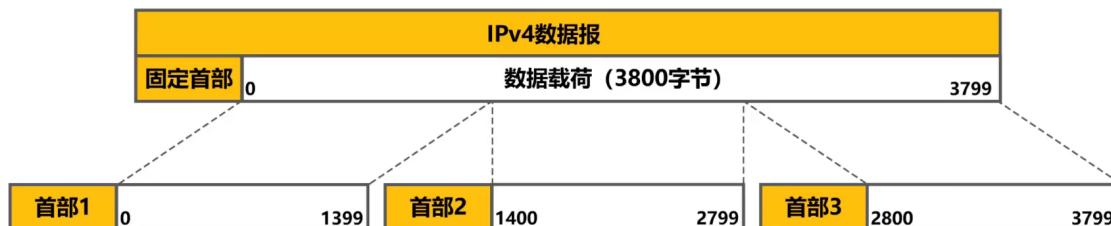
- DF位：1表示不允许分片，0表示允许
- MF位：1表示“后面还有分片”，0表示“这是最后一个分片”
- 保留位：必须为0

- 片偏移(必须是整数)

占13比特，指出分片数据报的数据载荷部分偏移其在原数据报的位置有多少单位

片偏移以8个字节为单位

【举例】对IPv4数据报进行分片



	总长度	标识	MF	DF	片偏移
原始数据报	3800+20	12345	0	0	0
分片1的数据报	1400+20	12345	1	0	0/8
分片2的数据报	1400+20	12345	1	0	1400/8
分片3的数据报	1000+20	12345	0	0	2800/8

- 生存时间

占8比特，最初以秒为单位，最大生存周期为255秒；路由器转发IP数据报时，将IP数据报首部中的该字段值减去IP数据报在本路由器上耗费的时间，若不为0【说明路由器消耗时间后还活着】就转发，否则丢弃

现在以"跳数"为单位，路由器转发IP数据报时，将IP数据报首部中的该字段值减1，若不为0就转发，否则丢弃【防止兜圈】

- 协议

占8比特，指明 IPv4 数据报的数据部分是何种协议数据单元

常用的一些协议和相应的协议字段值如下。

协议名称	ICMP	IGMP	TCP	UDP	IPv6	OSPF
协议字段值	1	2	6	17	41	89

- 首部检验和

占16比特，用来检测首部在传输过程中是否出现差错，比 CRC 检验码简单，称为因特网检验和

IP 数据报每经过一个路由器，路由器都要重新计算首部检验和，因为某些字段【生存时间、标志、片偏移等】的取值可能发生变化

由于 IP 层本身不提供可靠传输服务，并且计算首部校验和是一项耗时的操作，因此在 IPv6 中，路由器不再计算首部校验和，从而更快转发 IP 数据报

- 源IP地址和目的IP地址

各占32比特，用来填写发送该 IP 数据报的源主机 IP 地址和接收该 IP 数据报的目的主机

7. 网际控制报文ICMP

为了更有效地转发 IP 数据报和提高交付成功的几率，在网际层使用了网际控制报文协议 ICMP

- 主机或路由器使用 ICMP 来发送差错报告报文和询问报文
- ICMP 报文被封装在 IP 数据报中发送

回答报文类型

终点不可达

当路由器或主机不能交付数据报时，就向源点发送终点不可达报文。具体可再根据 ICMP 的代码字段细分为目的网络不可达、目的主机不可达、目的协议不可达、目的端口不可达、目的网络位置、目的主机未知等 13 种错误

源点抑制

当路由器或主机由于拥塞而丢弃数据报时，就向源点发送源点抑制报文，使源点知道应当把数据报发送速率放慢

时间超过

当路由器收到一个目的 IP 地址不是自己的 IP 数据报，会将其**生存时间 TTL 字段值减 1**。若结果不为 0，则将该 IP 数据报转发出去；**若结果为 0，除丢弃该 IP 数据报外，还要向源点发送时间超过报文**

当终点在预先规定的时间内不能收到一个数据报的全部数据报片时，就把已收到的数据报片都丢弃，也会向源点发送**时间超过报文**

参数问题

当路由器或目的主机收到 IP 数据报后，根据其首部中的检验和字段发现首部在传输过程中**出现了误码，就丢弃该数据报，并向源点发送参数问题报文**

改变路由(重定向)

路由器把改变路由报文发送给主机，让主机知道下次应将数据报发送给另外的路由器(可通过更好的路由)

如主机 1 的默认路由是 R1，信息经过 R1 时，R1 发现最佳路由不是自己，而是 R2，所以通过 ICMP 告知主机 1

询问报文类型

回送请求和回答

- ICMP 回送请求报文是由主机或路由器向一个特定的目的主机发出的询问
- 收到此报文的主机必须给源主机或路由器发送 ICMP 回送回答报文
- 这种询问报文用来测试目的站是否可达及了解其有关状态

时间戳请求和回答

- ICMP 时间戳请求报文是请某个主机或路由器回答当前的日期和时间
- 在 ICMP 时间戳回答报文中有 32 比特的字段，其中写入的整数代表从 1900 年 1 月 1 日起到当前时刻一共有多少秒
- 这种询问报文用来进行时钟同步和测量时间

不发送ICMP的情况

1. 对 ICMP 差错报告报文不再发送 ICMP 差错报告报文
2. 对第一个分片的数据报片的所有后续数据报片都不发送 ICMP 差错报告报文
3. 对具有多播地址的数据报都不发送 ICMP 差错报告报文
4. 对具有特殊地址(如 127.0.0.0 或 0.0.0.0)的数据报不发送 ICMP 差错报告报文

ICMP应用举例

分组网间探测PING

- 用来测试主机或路由器间的连通性【eNLP 的 ping 命令】
- 应用层直接使用网际层的 ICMP (没有通过运输层的 TCP 或 UDP)
- 使用 ICMP 回送请求和回答报文

跟踪路由traceroute

用来测试IP数据报从源主机到达目的主机要经过哪些路由器

windows 版本

- tracert 命令
- 应用层直接使用网际层 ICMP
- 使用了 ICMP 回送请求和回答报文以及差错报告报文

实现方法

由主机发送出去的数据包中的生存时间字段TTL由1开始逐渐增加，每个路由器都会返回一个时间超过报文，由此达到跟踪路由器的目的

8. 虚拟专用网VPN

专用（私有）地址：

10.0.0.0~10.255.255.255(10/8地址块)

172.16.0.0~172.31.255.255(172.16/12地址块)

192.168.0.0~192.168.255.255(192.168/16地址块)

利用公用的因特网作为本机构各专用网之间的通信载体，这样的专用网又称为虚拟专用网。

虚拟专用网中各主机所分配的地址应该是本机构可自由分配的专用地址

如下图所示，同一机构内不同部门的内部网络所构成的虚拟专用网VPN又称为**内联网VPN**。

有时一个机构的VPN需要有某些外部机构（通常就是合作伙伴）参加进来。这样的VPN就称为**外联网VPN**。

在外地工作的员工需要访问公司内部的专用网络时，只要在任何地点接入到因特网，运行驻留在员工PC中的VPN软件，在员工的PC和公司的主机之间建立VPN隧道，即可访问专用网络中的资源。这种VPN称为**远程接入VPN**。



9. 网络地址转换NAT

NAT能使大量使用内部专用地址的专用网络用户共享少量外部全球地址来访问因特网上的主机和资源【为了节省IPv4地址】

由于绝大多数的网络应用都是使用运输层协议TCP或UDP来传送数据，因此可以利用运输层的端口号和IP地址一起进行转换。

这样，用一个全球IP地址就可以使多个拥有本地地址的主机同时和因特网上的主机进行通信。这种将端口号和IP地址一起进行转换的技术叫作**网络地址与端口号转换NAPT**

第5章 运输层

物理层、数据链路层以及网络层它们共同解决了将主机通过异构网络互联起来所面临的问题，实现了**主机到主机**的通信

但实际上在计算机网络中进行通信的真正实体是位于通信两端主机中的进程

如何为运行在不同主机上的应用进程提供直接的通信服务时运输层的任务，**运输层协议又称端到端协议**

1. 端口号

为了使运行不同操作系统的计算机的应用进程之间能进行网络通信，必须使用统一的方法对TCP/IP体系的应用进程进行标识，即端口号。

为什么不能使用进程标识符PID来区分各进程？

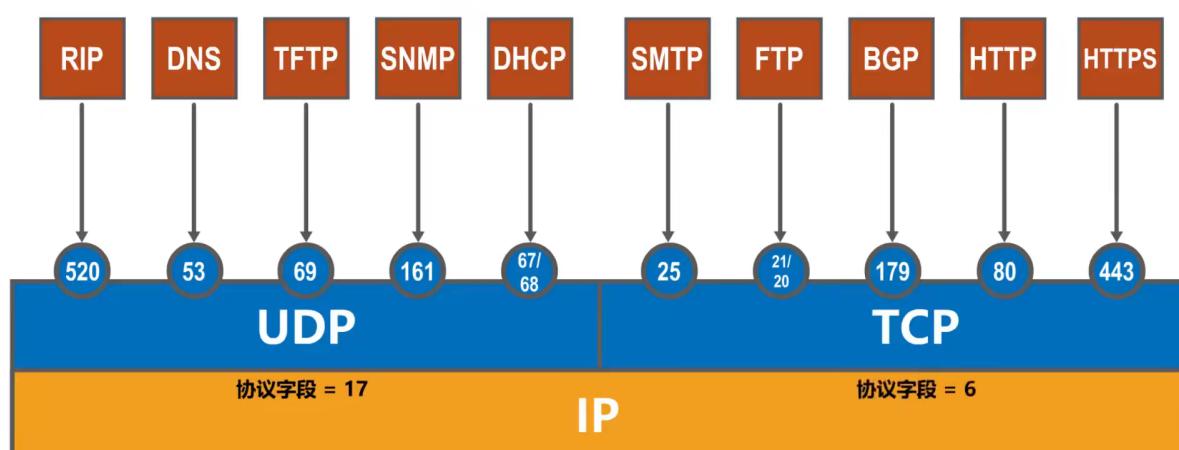
因为因特网上的计算机并不是使用统一的操作系统，**不同的操作系统使用不同格式的进程标识符**

端口号只具有本地意义，即端口号只是为了标识本计算机应用层中的各进程，在因特网中，**不同计算机中的相同的端口号是没有联系的**

端口号使用**16比特**表示，取值范围 0~65535。

- **熟知端口号：0~1023**，IANA把这些端口号指派给了TCP/IP体系中最重要的一些应用协议
- **登记端口号：1024~49151**，为没有熟知端口号的应用程序使用。使用这类端口号必须在IANA按照规定的手续登记，以防止重复。如Microsoft RDP微软远程桌面使用的端口是3389
- **短暂端口号：49152~65535**，留给客户进程选择暂时使用。当服务器进程收到客户进程的报文时，就知道了客户进程所使用的动态端口号。**通信结束后，这个端口号可供其他客户进程以后使用**

■ TCP/IP体系的应用层常用协议所使用的运输层熟知端口号

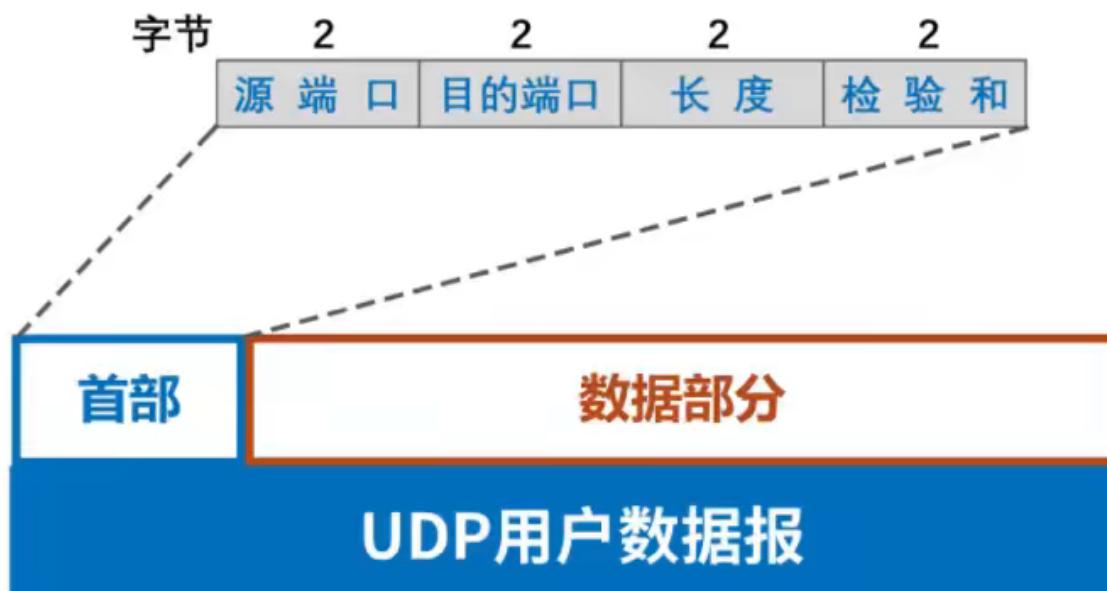


2. 复用

- 发送方的某些应用进程所发送的不同应用报文，在运输层使用UDP协议进行封装，这是UDP复用；若用TCP封装则称TCP复用
- 运输层使用端口号区分不同进程，不管使用何种协议封装的报文，在网络层都需要使用IP协议封装成IP数据报，这是IP复用，数据报中协议字段的值用来表明封装的是何种协议数据单元
- 根据协议字段的值，将IP数据报封装的协议数据单元上交运输层的过程叫IP分用
- 同理，UDP根据端口号将数据交给应用进程叫做UDP分用；TCP根据端口号将数据交给应用进程叫做TCP分用

3. 用户数据报协议UDP

- UDP是无连接的，随时可向目的主机发送报文，支持单播、多播和广播
- UDP收到应用层报文后直接为报文添加UDP首部就进行发送，即面向应用报文
- UDP数据报首部仅8字节



4. 传输控制协议TCP

- TCP时面向连接的。发送数据前需要“三报文握手”建立连接，数据传输结束后需要“四报文挥手”释放连接
- 仅支持单播

- **TCP是面向字节流的。**其将应用进程交付下来的数据块仅仅看作是一连串的字节流，TCP将这些字节流编号并存储在缓冲中；接收方一边接收数据，一边将缓冲中的数据交给应用进程。
- **接收方收到的字节流必须和发送方收到的字节流完全一致**
- **不会出现传输差错(误码、丢失、乱序、重复)**
- **TCP报文段首部最小20字节，最大60字节**



流量控制

一般来说，我们希望数据传输能快一些，但如果发送方把数据发送得过快，接收方就可能来不及接收，这会造成数据的丢失

流量控制就是让发送方的发送速率不要太快，要让接收方来得及接收

利用滑动窗口实现流量控制

- 发送方和接收方窗口保持一致，发送方窗口随着接收方窗口变化而变化（通过确认报文告知发送方）
- 发送方发送完窗口内数据后需要等到确认报文才会滑动窗口并继续发送，若窗口内的某个值很久没有收到回答报文，则超时重传报文

若接收方窗口调为0后，一段时间之后又调为200，此时向发送方传递确认报文，可此时报文丢失，则会造成发送方窗口始终为0，接收方以为发送方收到了确认报文而开始等待数据，造成死锁，如何解决？

当发送方窗口大小为0时，其隔一段时间就会发送一个1字节大小的零窗口探测报文，看看此时接收窗口大小是否进行调整

若发送的零窗口探测报文也丢失了，会造成新死锁吗？

不会。因为零窗口探测报文也有超时重传机制

拥塞控制

在某段时间，若对网络中某一资源的需求超过了该资源所能提供的可用部分，网络性能就要变坏，这种情况叫做拥塞

若出现拥塞而不进行控制，整个网络的吞吐量将随输入负荷的增大而下降

1. 慢开始和拥塞避免

- 发送方维护一个叫做**拥塞窗口** $cwnd$ 的状态变量，其值取决于网络的拥塞程度，并且动态变化
 - 拥塞窗口 $cwnd$ 的维护原则：只要网络没有出现拥塞，拥塞窗口就再增大一些(确认报文段窗口大小)，但只要网络出现拥塞，拥塞窗口就减少一些
 - 判断出现网络拥塞的依据：没有按时收到应当到达的确认报文(发送超时重传)
- 发送方将拥塞窗口作为**发送窗口** $swnd$ ，即 $swnd=cwnd$
- 维护一个**慢开始门限** $ssthresh$ 状态变量
 - 当 $cwnd < ssthresh$ 时，使用慢开始算法
 - 当 $cwnd > ssthresh$ 时，停止使用慢开始算法而改用拥塞避免算法
 - 当 $cwnd = ssthresh$ 时，既可使用慢开始算法，也可以使用拥塞避免算法

慢开始

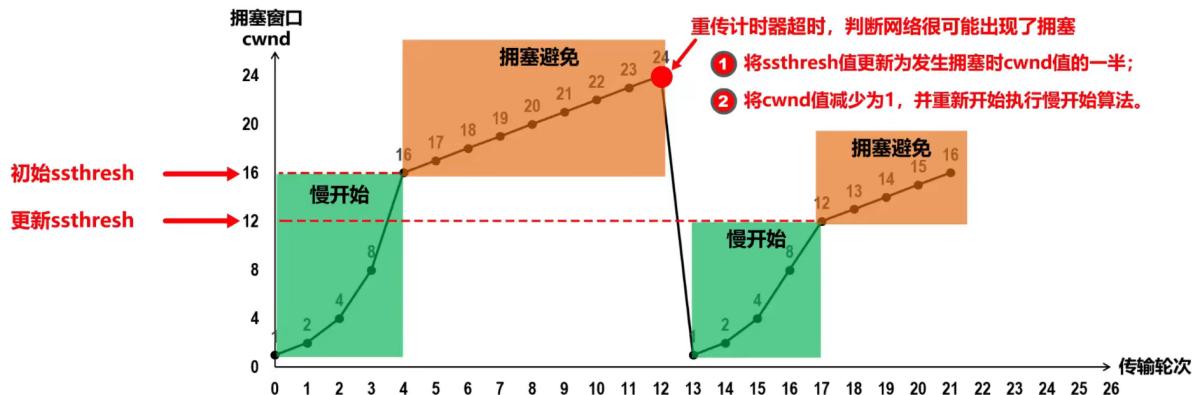
拥塞窗口从1开始，根据应答报文大小来扩大拥塞窗口，如发送方窗口2，应答大小2，则下次发送大小为4

拥塞避免

拥塞窗口每次只扩大1，而不是向慢开始那样根据发送方的返回窗口进行增加。

当窗口增加到一定大小，发送方发送的报文出现了超时重传，则判断网络可能出现了拥塞，此时将**拥塞窗口初始化为1**，同时将**慢开始门限** $ssthresh$ 设置为发生拥塞时窗口大小的一半

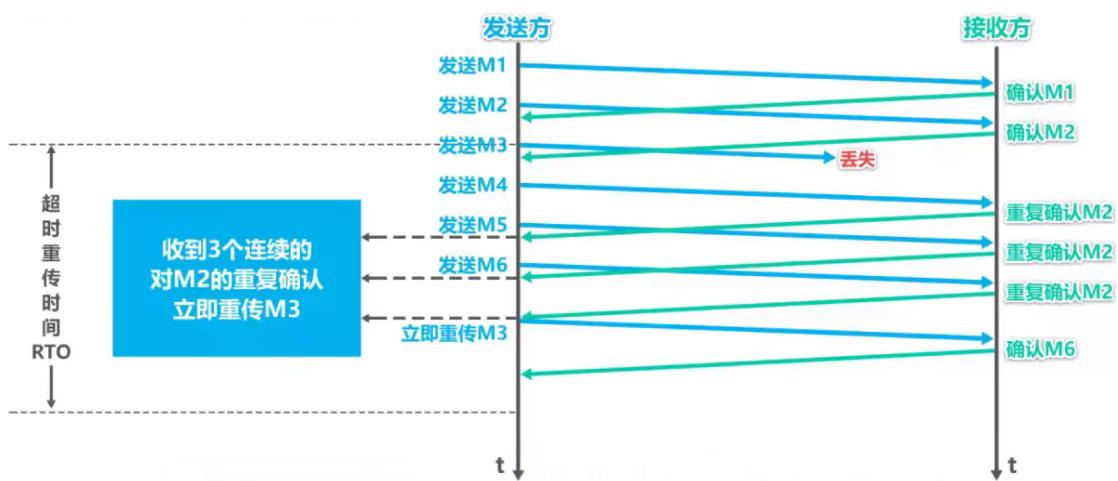
"拥塞避免"并非指完全能够避免拥塞，而是指在拥塞避免阶段将拥塞窗口控制为按线性规律增长，使网络比较不容易出现拥塞



2. 快重传和快恢复

快重传是使发送方尽快进行重传，而不是等待超时重传计时器超时再重传

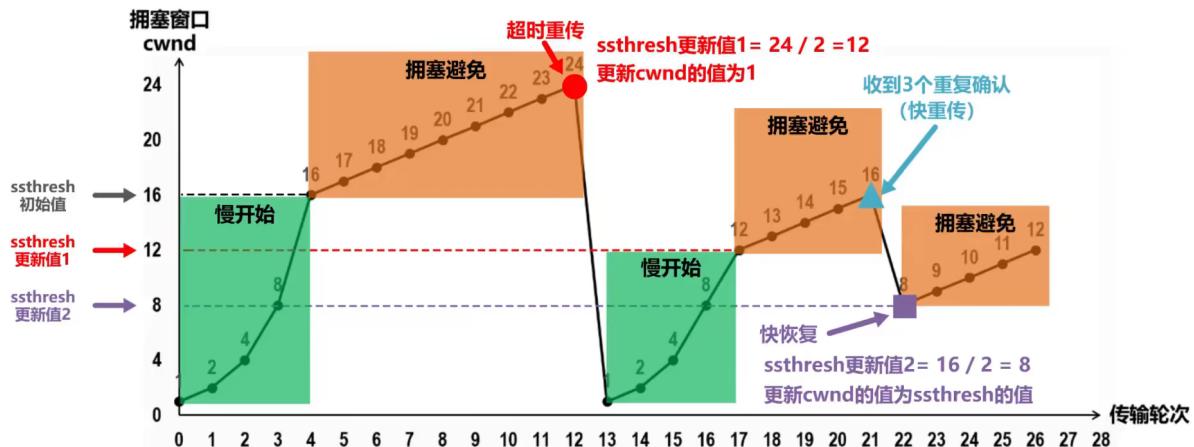
- 要求接收方不用等待自己发送数据时才进行捎带确认，而是要立即发送确认
- 即使收到了失序的报文段(说明有一段丢失了)也要立即发出对已收到的报文段的重复确认
- 发送方一旦收到3个连续的重复确认，就将相应的报文段立即重传，而不是等该报文段的超时重传计时器超时再重传
- 对于个别丢失的报文段，发送方不会出现超时重传，也就不会误认为出现了拥塞。使用快重传可以使整个网络的吞吐量提高约 20%



发送方一旦收到3个连续的重复确认，就知道现在只是丢失了个别报文段。也是不启动慢开始算法，而执行快恢复算法

- 发送方将慢开始门限ssthresh值和拥塞窗口cwnd值调整为当前窗口的一半，开始执行拥塞避免算法

- 也有的快恢复实现是把快恢复开始时的拥塞窗口 cwnd 值再增大一些，即等于新的 ssthresh+3
 - 既然发送方收到 3 个重复的确认，就表明有 3 个数据报文段已经离开了网络
 - 这三个报文段不再消耗网络资源而是停留在接收方的接收缓存中
 - 可见现在网络中不是堆积了报文段而是减少了 3 个报文段。因此可以适当把拥塞窗口扩大些



超时重传时间(RTO)选择

正常情况下，超时重传时间应该设为略大于往返时间。但是由于各区域的速率可能不一致，因此将超时重传时间设置为一个固定值是行不通的

利用每次测量得到的 RTT 样本，计算加权平均往返时间 RTT_S (平滑的往返时间)

显然，超时重传时间 RTO 应略大于加权平均往返时间 RTT_S

RFC6298建议使用下式计算超时重传时间RTO：

$$RTO = RTT_S + 4 \times RTT_D$$

加权平均往返时间 RTT_S

$$RTT_{SI} = RTT_I$$

$$\text{新的 } RTT_S = (1 - \alpha) \times \text{旧的 } RTT_S + \alpha \times \text{新的 } RTT \text{ 样本}$$

$$0 \leq \alpha < 1$$

已成为建议标准的RFC6298推荐的 α 值为 1/8，即 0.125。

RTT偏差的加权平均 RTT_D

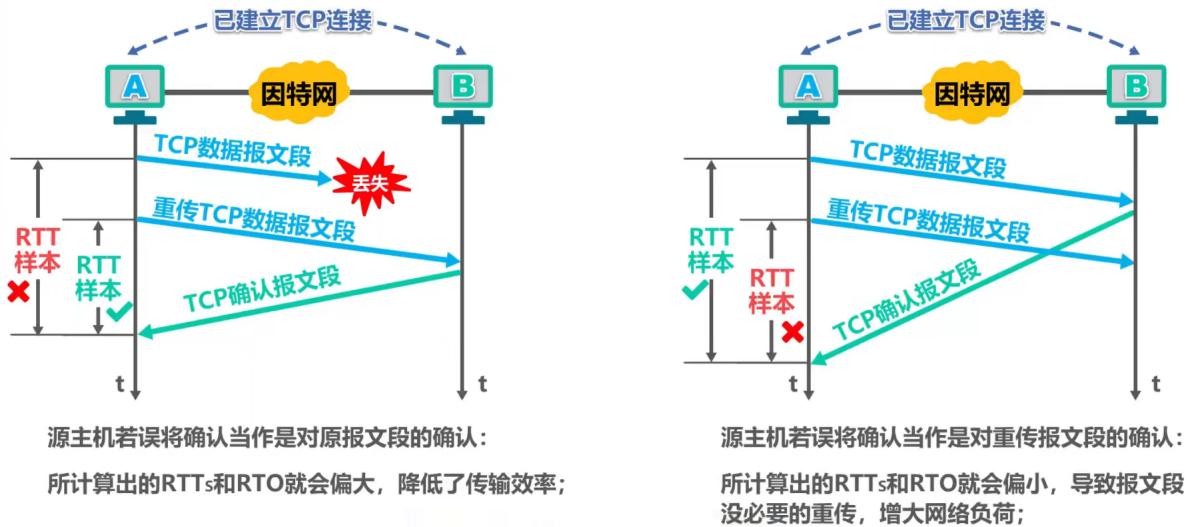
$$RTT_{DI} = RTT_I \div 2$$

$$\text{新的 } RTT_D = (1 - \beta) \times \text{旧的 } RTT_D + \beta \times |RTT_S - \text{新的 } RTT \text{ 样本}|$$

$$0 \leq \beta < 1$$

已成为建议标准的RFC6298推荐的 β 值为 1/4，即 0.25。

往返时间测量问题



针对出现超时重传时无法测准往返时间RTT的问题，有以下解决方法

在计算加权平均往返时间 RTT_S 时，只要报文段重传了，就不采用其往返时间 RTT 样本。也就是出现重传时，不重新计算 RTT_S ，进而超时重传时间RTO也不会重新计算。

此方法的漏洞如下：如果报文段时延突然增大很多，并且之后很长一段时间都会保持这种时延。因此在原来得出的重传时间内，不会收到确认报文段，于是重传，造成死锁

修正方法：报文段每重传一次，就把超时重传时间RTO增大一些，典型的做法是将RTO的值取为旧RTO的2倍

可靠传输

具体实现

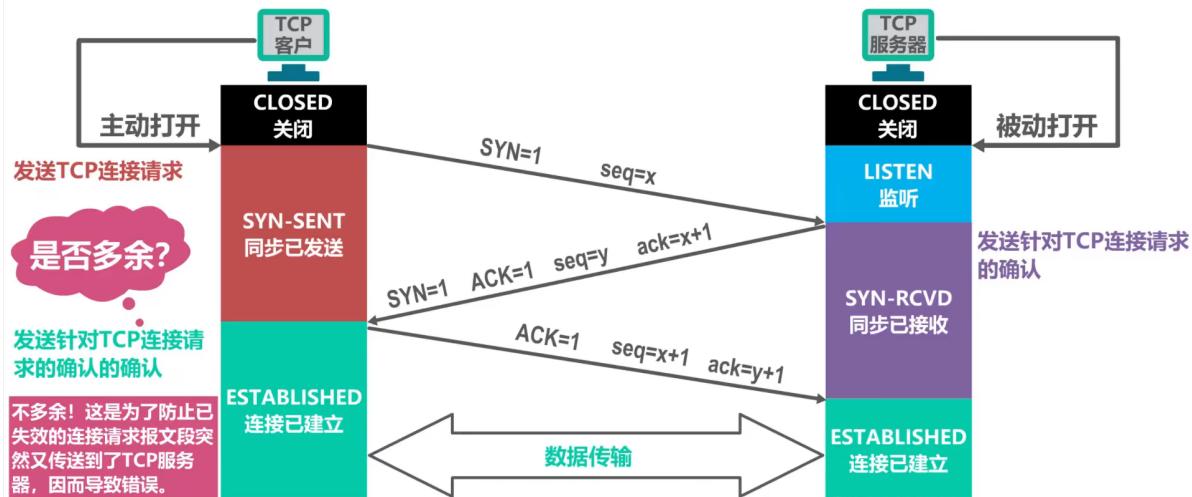
- 虽然发送方的发送窗口是根据接收方的接收窗口设置的，但在同一时刻，发送方的发送窗口并不总是和接收方的接收窗口一样大。
 - 网络传送窗口值需要经历一定的时间滞后，并且这个时间还是不确定的。
 - 发送方还可能根据网络当时的拥塞情况适当减小自己的发送窗口尺寸。
- 对于不按序到达的数据应如何处理，TCP并无明确规定。
 - 如果接收方把不按序到达的数据一律丢弃，那么接收窗口的管理将会比较简单，但这样做对网络资源的利用不利，因为发送方会重复传送较多的数据。
 - TCP通常对不按序到达的数据是先临时存放在接收窗口中，等到字节流中所缺少的字节收到后，再按序交付上层的应用进程。
- TCP要求接收方必须有累积确认和捎带确认机制，这样可以减小传输开销。接收方可以在合适的时候发送确认，也可以在自己有数据要发送时把确认信息顺便捎带上。
 - 接收方不应过分推迟发送确认，否则会导致发送方不必要的超时重传，这反而浪费了网络的资源。

TCP标准规定，确认推迟的时间不应超过0.5秒。若收到一连串具有最大长度的报文段，则必须每隔一个报文段就发送一个确认[RFC 1122]。
 - 携带确认实际上并不经常发生，因为大多数应用程序很少同时在两个方向上发送数据。
- TCP的通信是全双工通信。通信中的每一方都在发送和接收报文段。因此，每一方都有自己的发送窗口和接收窗口。在谈到这些窗口时，一定要弄清楚是哪一方的窗口。

运输连接管理

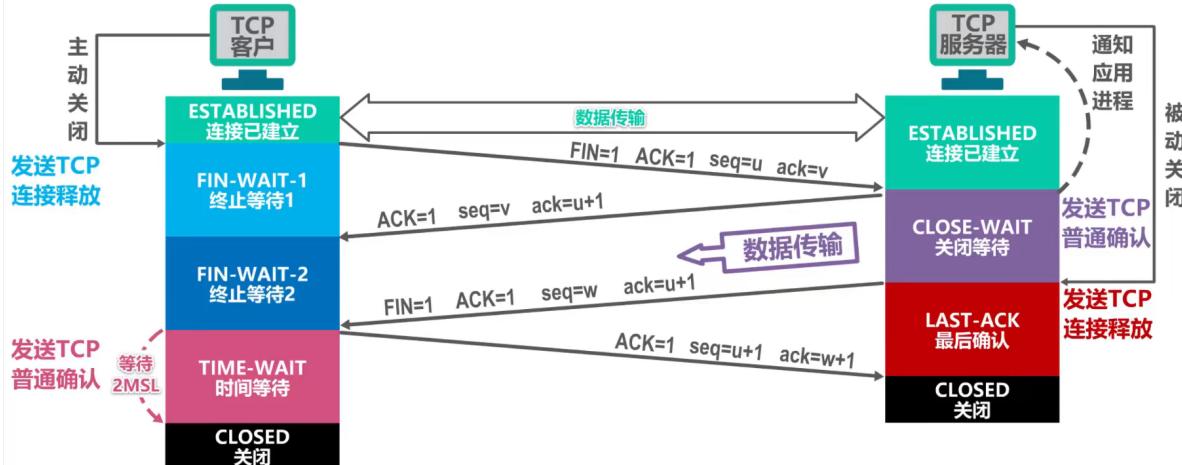
TCP的运输连接管理就是使运输连接的建立和释放都能正常地进行

■ TCP使用“三报文握手”建立连接



- **SYN** 为 1 的报文段不能携带数据, 但会消耗一个序列号 **seq**
- **ACK=1** 代表这是普通确认报文段, **ack=x+1** 表示这是对报文段序列号 **seq=x** 的确认

■ TCP通过“四报文挥手”来释放连接



MSL(Maximum Segment Lifetime)意思是最长报文段寿命, RFC793建议为2分钟。

- 客户端发起关闭请求, 一去一回后进入半关闭状态【客户端不再发送数据, 服务端可能还会发】
- 服务器将自己剩余的数据发送完后也发送一个关闭请求, 接着客户端给予回应后服务器关闭, 客户机则要等到一段时间后完全关闭(防止发给服务器的确认报文丢失)



- TCP服务器进程每收到一次TCP客户进程的数据，就重新设置并启动**保活计时器**（2小时定时）。
- 若保活计时器定时周期内未收到TCP客户进程发来的数据，则**当保活计时器到时后，TCP服务器进程就向TCP客户进程发送一个探测报文段**，以后则每隔75秒钟发送一次。若一连发送10个探测报文段后仍无TCP客户进程的响应，TCP服务器进程就认为TCP客户进程所在主机出了故障，接着就关闭这个连接。

首部格式



- **源端口**: 占16比特，写入源端口号，用来标识**发送该TCP报文段的应用进程**
- **目的端口**: 占16比特，写入目的端口号，用来标识**接收该TCP报文段的应用进程**
- **序号**: 占32比特，取值范围 $[0, 2^{32} - 1]$ ，序号增加到最后一个后，下一个序号就又回到0。作用是指出本TCP报文段数据载荷的第一个字节的序号
- **确认标志位ACK**: 取值为1时确认号字段才有效，为0时确认号字段无效
- **确认号**: 占32比特，取值范围 $[0, 2^{32} - 1]$ ，序号增加到最后一个后，下一个序号就又回到0。可理解为**若确认号=n，则表明到序号n-1为止的所有数据都已正确接收，期望接收序号为n的数据**
- **数据偏移**: 占4比特，并以4字节为单位
用来指出TCP报文段的数据载荷部分的起始处距离TCP报文段的起始处。这个字段**实际上是指出TCP报文段的首部长度**

首部固定长度为20字节，因此数据偏移字段的最小值(0101)₂；首部最大长度为60字节，因此数据偏移字段最大值为(1111)₂

- **保留字段：**占6比特，保留为今后使用，但是目前应置为0
- **窗口：**占16比特，以字节为单位。指出**发送本报文段一方的接收窗口**窗口值作为接收方让发送方设置其发送窗口的依据，这是**以接收方的接收能力来控制发送方的发送能力**，称为流量控制
- **检验和：**占16比特，检测范围**包括TCP报文段的首部和数据载荷两部分**在计算校验和时，要在TCP报文段的前面加上12字节的伪首部
- **同步标志位SYN：**在TCP连接建立时用来同步序号
- **终止标志位FIN：**用来释放TCP连接
- **复位标志位RST：**用来复位TCP连接
当RST=1时，表明TCP连接出现了异常，必须释放连接，然后再重新建立连接；RST置1还用来拒绝一个非法的报文段或拒绝打开一个TCP连接
- **推送标志位PSH：**接收方的TCP收到该标志位为1的报文段会尽快上交**应用进程**，而不必等到接收缓存都填满后再向上交付
- **紧急标志位URG：**取值为1时**紧急指针字段有效**；取值为0时**紧急指针字段无效**。
- **紧急指针：**占16比特，以字节为单位，用来指明紧急数据的长度
当发送方有紧急数据时，可**将紧急数据插队到发送缓存的最前面**，并立刻封装到一个TCP报文段中进行发送。紧急指针会指出本报文段数据载荷部分包含了多长的紧急数据，紧急数据之后是普通数据
- **选项：**增加选项可以增加TCP的功能
 - **最大报文段长度MSS选项：** TCP报文段数据载荷部分的最大长度
 - **窗口扩大选项：**为了扩大窗口(提高吞吐率)
 - **时间戳选项：**
 - 用来计算往返时间RTT
 - 用于处理序号超范围的情况，又称为防止序号绕回PAWS
 - **选择确认选项：**实现选择确认功能

- 填充：由于选项长度可变，因此使用填充来保证报文段首部能被4整除

第6章 应用层

解决通过应用进程的交互来实现特定网络应用的问题

应用层是计算机网络体系结构的最顶层，是设计和建立计算机网络的最终目的

客户/服务器方式(C/S)

• 客户/服务器

- 客户和服务器是指通信中所涉及的两个应用进程
- 客户/服务器方式所描述的是进程之间服务和被服务的关系
- **客户是服务请求方，服务器是服务提供方**
- **服务器总是处于运行等待状态，并等待客户的服务请求。服务器具有固定端口号(例如HTTP服务器的默认端口号为80)，而运行服务器的主机也具有固定的IP地址**
- 基于c/s方式的应用服务通常是**服务集中型的**，即应用服务集在网络中比客户计算机少得多的服务器计算机上
 - 由于一台服务器计算机要为多个客户机提供服务，在c/s应用中，常会出现**服务器计算机跟不上众多客户机请求的情况**
 - 为此，在c/s应用中，常用**计算机群集(或服务器场)**构建一个强大的**虚拟服务器**

对等方式(P2P方式)

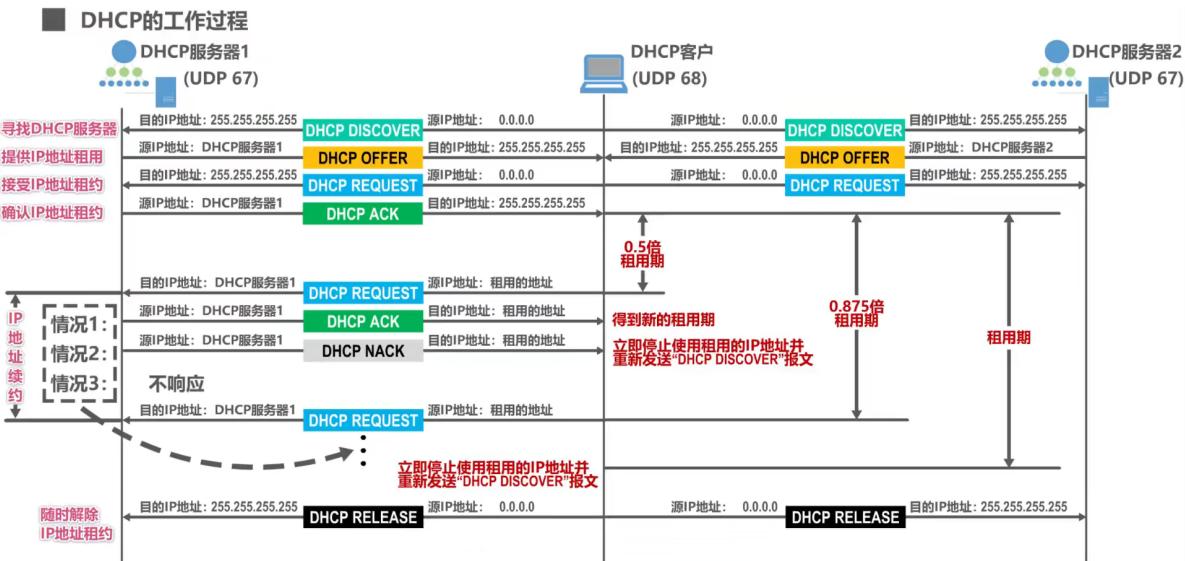
■ 对等 (Peer-to-Peer, P2P) 方式

□ 在P2P方式中，**没有固定的服务请求者和服务提供者**，分布在网络边缘各端系统中的应用进程是对等的，被称为**对等方**。**对等方相互之间直接通信**，每个对等方既是服务的请求者，又是服务的提供者。

- 目前，在因特网上流行的P2P应用主要包括P2P文件共享、即时通信、P2P流媒体、分布式存储等。
- 基于P2P的应用是**服务分散型的**，因为服务不是集中在少数几个服务器计算机中，而是分散在大量对等计算机中，这些计算机并不为服务提供商所有，而是为个人控制的桌面计算机和笔记本电脑，它们通常位于住宅、校园和办公室中。
- P2P方式的最突出特性之一就是它的**可扩展性**。因为系统每增加一个对等方，不仅增加的是服务的请求者，同时也增加了服务的提供者，**系统性能不会因规模的增大而降低**。
- P2P方式**具有成本上的优势**，因为它通常不需要庞大的服务器设施和服务器带宽。为了降低成本，服务提供商对于将P2P方式用于应用的兴趣越来越大。

动态主机配置协议DHCP

通过 DHCP 服务器为指定网段主机分配 IP 地址



- DHCP 客户机首先进行广播寻找 DHCP 服务器("DHCP发现"消息), 源地址为本机 0.0.0.0, 目的地址为广播地址 255.255.255.255
- DHCP 服务器收到后, 返回"DHCPOFFER"消息, 其中包含提供给 DHCP 客户机的 IP 地址和相关配置信息。源地址为 DHCP 服务器地址, 目的地址为广播地址(因为此时目的客户机还没有IP地址)
- 客户机可能会收到多个DHCPOFFER消息, 一般以收到的第一个为准。此时客户机知道 DHCP 服务器可以给它分配地址, 因此发送"DHCPRREQUEST"报文来请求分配 IP 地址, 报文的源地址为 0.0.0.0, 目的地址为 255.255.255.255
- DHCP 收到请求信息后, 查看其中**事务ID**是否相符, 若不符则丢弃; 符合则从地址池中取得一个 IP 地址, 并通过 ARP 协议确认此地址未被使用后, 将其封装进"DHCPCONFIRM"信息中, 报文源地址为 DHCP 服务器地址, 目的地址为广播地址
- 客户机收到 DHCP 确认信息后, 查看其中**事务 ID**是否相符, 不符则丢弃; 符合则**再次使用ARP确认IP地址没有被使用, 确认成功后将此IP地址应用(有一定租约)**。
- **当IP地址租约达到0.5倍时间时**, 客户机会再次向 DHCP 服务器发送请求信息, 此时 DHCP 服务器会出现以下三种情况
 1. 收到请求后, 返回一个**确认报文**, 其中有新的 IP 地址租期
 2. 收到请求后, 返回**否认报文**, 则客户机收到后立刻停止使用IP地址并重新发送"DHCP发现"报文

3. 不响应。则在租期达到**0.875倍**时，**DHCP**客户必须重新发送"**DHCP 请求**"报文，继续等待**DHCP**服务器可能做出的反应。若依然无反应，则租用期到后，客户机必须立刻停止使用当前**IP 地址**
- **客户端可随时终止DHCP服务器提供的租用期**，这时只需要向**DHCP**服务器发送**DHCP 释放**报文即可。源地址**0.0.0.0**，目的地址

255.255.255.255

域名系统DNS

我们通过输入网址来访问网页，可实际上计算机间的通信是通过**IP 地址**，所以网址的本质上是**IP 地址**，将网址与**IP 地址**映射起来就是**DNS**的作用

因特网是否可以只适用一台DNS服务器？

这种做法不可取。因为因特网的规模很大，如果只有一个服务器，那么一旦其出现故障，整个因特网就会瘫痪

因此现实中采用**层次结构的命名树**作为主机的名字(即**域名**)，并使用分布式的**域名系统 DNS**

DNS使大多数域名都在本地解析，仅少量解析需要在因特网上通信，因此系统效率很高。由于**DNS**是分布式系统，即使单个计算机出了故障，也不会妨碍整个系统的正常运行

- 因特网采用**层次树状结构的域名结构**
- 域名的结构由若干个分量组成，各分量之间用“点”隔开，分别代表不同级别的**域名**。

… .三级域名.二级域名.顶级域名

 - 每一级的**域名**都由英文字母和数字组成，不超过63个字符，不区分大小写字母。
 - 级别最低的**域名**写在最左边，而级别最高的**顶级域名**写在最右边。
 - 完整的**域名**不超过255个字符。
- 域名系统既不规定一个**域名**需要包含多少个下级**域名**，也不规定每一级的**域名**代表什么意思。
- 各级**域名**由其上一级的**域名管理机构**管理，而最高的**顶级域名**则由因特网名称与数字地址分配机构**ICANN**进行管理。

【举例】湖南科技大学网络信息中心的**域名**

nic.hnust.edu.cn

四级域名 三级域名 二级域名 顶级域名

■ 顶级域名TLD (Top Level Domain) 分为以下三类:

国家顶级域名nTLD 采用ISO 3166的规定。如cn表示中国, us表示美国, uk表示英国、等等。

通用顶级域名gTLD 最常见的通用顶级域名有七个, 即: **com (公司企业)**、**net (网络服务机构)**、**org (非营利性组织)**、**int (国际组织)**、**edu (美国教育结构)**、**gov (美国政府部门)**、**mil (美国军事部门)**。

反向域arpa 用于反向域名解析, 即IP地址反向解析为域名。

| 注意区分

■ 在**国家顶级域名下注册的二级域名均由该国家自行确定**。例如, 顶级域名为jp的日本, 将其教育和企业机构的二级域名定为ac和co, 而不用edu和com。

■ 我国则将**二级域名**划分为以下两类:

类别域名 共七个: ac (科研机构)、**com (工、商、金融等企业)**、**edu (教育机构)**、gov (政府部门)、net (提供网络服务的机构)、mil (军事机构) 和org (非营利性组织)。

行政区域名 共34个, 适用于我国的各省、自治区、直辖市。例如: bj为北京市、sh为上海市、js为江苏省, 等等。

■ 域名和IP地址的映射关系必须保存在域名服务器中, 供所有其他应用查询。显然不能将所有信息都储存在一台域名服务器中。DNS使用**分布在各地的域名服务器**来实现域名到IP地址的转换。

■ 域名服务器可以划分为以下四种不同的类型:

根域名服务器

根域名服务器是最高层次的域名服务器。每个根域名服务器都知道所有的顶级域名服务器的域名及其IP地址。因特网上共有**13个**不同IP地址的根域名服务器。尽管我们将这13个根域名服务器中的每一个都视为单个的服务器, 但“每台服务器”实际上是由许多分布在世界各地的计算机构成的**服务器群集**。当本地域名服务器向根域名服务器发出查询请求时, 路由器就把查询请求报文转发到离这个DNS客户最近的一个根域名服务器。这就加快了DNS的查询过程, 同时也更合理地利用了因特网的资源。**根域名服务器通常并不直接对域名进行解析, 而是返回该域名所属顶级域名的顶级域名服务器的IP地址**。

顶级域名服务器

这些域名服务器负责**管理在该顶级域名服务器注册的所有二级域名**。当收到DNS查询请求时就给出相应的回答 (可能是最后的结果, 也可能是下一级权限域名服务器的IP地址)。

权限域名服务器

这些域名服务器负责**管理某个区的域名**。每一个主机的域名都必须在某个权限域名服务器处注册登记。因此权限域名服务器知道其管辖的域名与IP地址的映射关系。另外, 权限域名服务器还知道其下级域名服务器的地址。

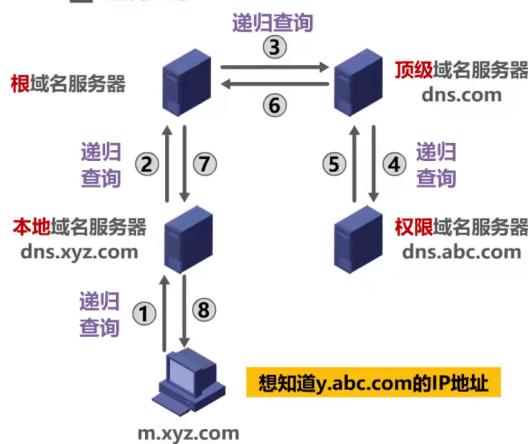
本地域名服务器

本地域名服务器不属于上述的域名服务器的等级结构。当一个主机发出DNS请求报文时, 这个报文就首先被送往该主机的本地域名服务器。本地域名服务器起着代理的作用, 会将该报文转发到上述的**域名服务器的等级结构中**。每一个因特网服务提供者ISP, 一个大学, 甚至一个大学里的学院, 都可以拥有一个本地域名服务器, 它有时也称为**默认域名服务器**。本地域名服务器离用户较近, 一般不超过几个路由器的距离, 也有可能就在同一个局域网中。本地域名服务器的IP地址需要直接配置在需要域名解析的主机中。

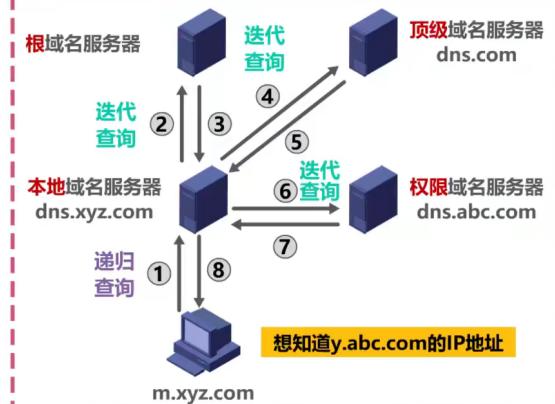
■ 域名解析的过程

递归查询

迭代查询



由于递归查询对于被查询的域名服务器负担太大, 通常采用以下模式: 从请求主机到本地域名服务器的查询是递归查询, 而其余的查询是迭代查询。



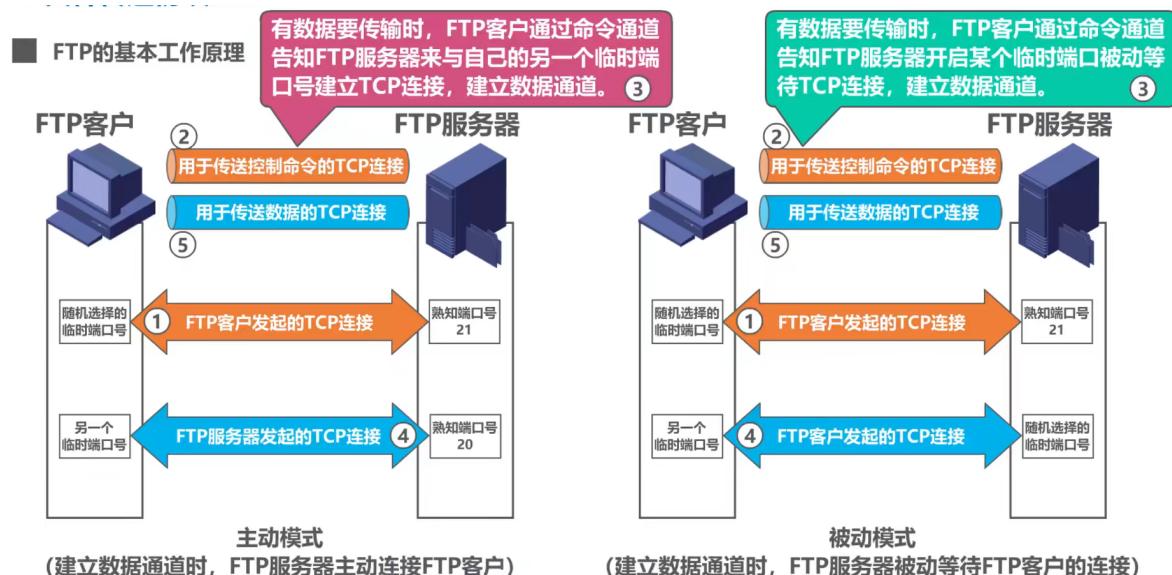
- 为了提高DNS的查询效率，并减轻根域名服务器的负荷和减少因特网上的DNS查询报文数量，在域名服务器中广泛地使用了**高速缓存**。高速缓存用来存放最近查询过的域名以及从何处获得域名映射信息的记录。
- 由于域名到IP地址的映射关系并不是永久不变，为保持高速缓存中的内容正确，**域名服务器应为每项内容设置计时器并删除超过合理时间的项**（例如，每个项目只存放两天）。
- 不但在本地域名服务器中需要高速缓存，在用户主机中也很需要。许多用户主机在启动时从本地域名服务器下载域名和IP地址的全部数据库，维护存放自己最近使用的域名的高速缓存，并且只在从缓存中找不到域名时才向域名服务器查询。同理，主机也需要保持高速缓存中内容的正确性。

文件传送协议FTP

将某台计算机中的文件通过网络传送到可能相距很远的另一台计算机中，是一项基本的网络应用，即**文件传送**

FTP的常见用途是在计算机之间传输文件，尤其是**用于批量传输文件**。FTP的另一个常见用途是**让网站设计者将构成网站内容的大量文件批量上传到他们的Web服务器**

- FTP提供交互式的访问**，运行客户指明文件的类型与格式(如指明是否使用**ASCII码**)，并**允许文件具有存取权限**(如访问文件的用户必须经过授权，并输入有效的口令)
- FTP屏蔽了各计算机系统的细节**，因而适合于在异构网络中任意计算机之间传送文件

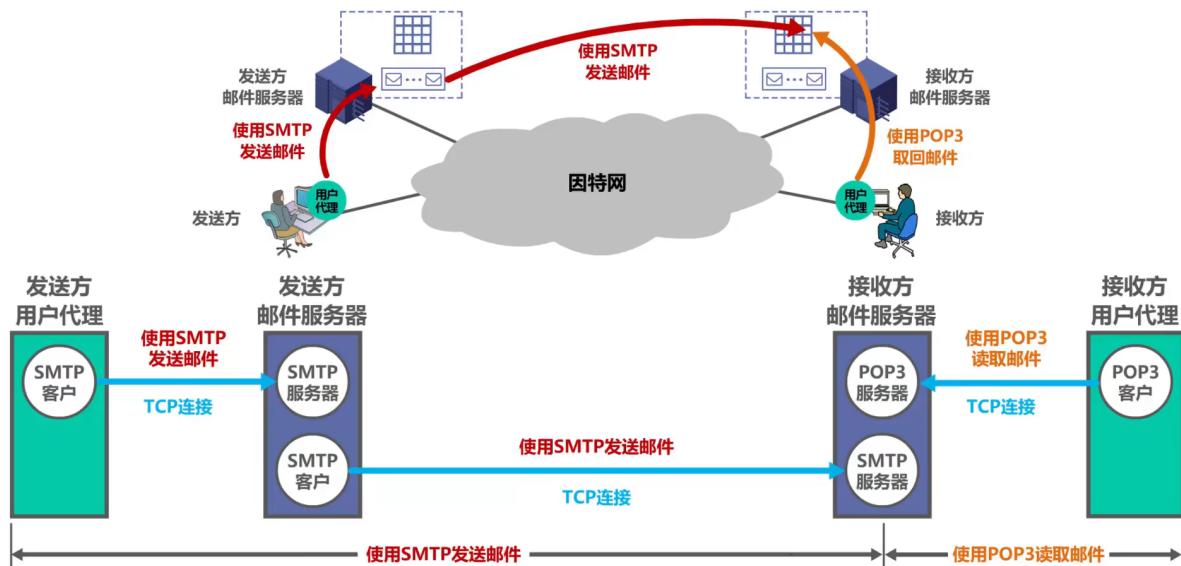


- 用于**传送控制命令的TCP连接**在整个会话过程都**保持开启状态**
- 用于**传送数据的TCP连接**只会在**有数据传送时开启**，**数据结束传送后就关闭**

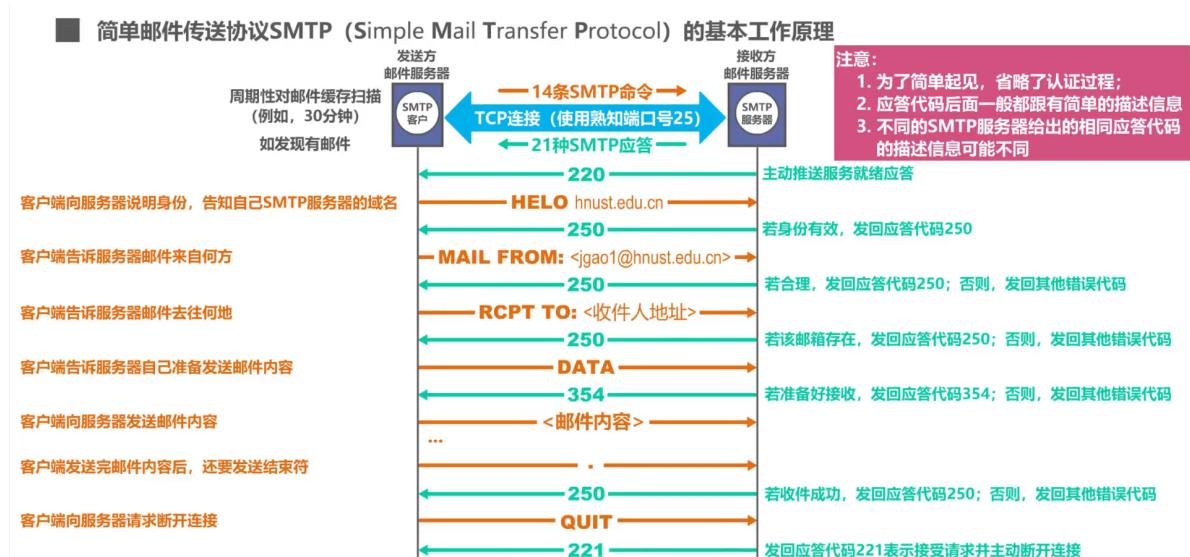
电子邮件

电子邮件系统的三个主要组成构件：用户代理，邮件服务器，以及电子邮件所需的协议

- **用户代理**是用户与电子邮件系统的接口，又称为**电子邮件客户端软件**
- **邮件服务器**是电子邮件系统的基础设施。因特网上所有的**ISP**都有邮件服务器，其功能是**发送和接收邮件**，同时还要负责维护用户的邮箱
- **协议**包括**邮件发送协议(如 SMTP)**和**邮件读取协议(如 POP3, IMAP)**



简单邮件传送协议SMTP



- **SMTP协议只能传送ASCII码文本数据**，不能传送可执行文件或其他的二进制对象。
- **SMTP不能满足传送多媒体邮件**（例如带有图片、音频或视频数据）的需要。并且许多其他非英语国家的文字（例如中文、俄文、甚至带有重音符号的法文或德文）也无法用SMTP传送。
- 为解决SMTP传送非ASCII码文本的问题，提出了**多用途因特网邮件扩展MIME** (Multipurpose Internet Mail Extensions)
 - 增加了**5个新的邮件首部字段**，这些字段提供了有关邮件主体的信息。
 - 定义了**许多邮件内容的格式**，对多媒体电子邮件的表示方法进行了标准化。
 - 定义了**传送编码**，可对任何内容格式进行转换，而不会被邮件系统改变。
- 实际上，MIME不仅仅用于SMTP，也用于后来的同样面向ASCII字符的HTTP。

邮件读取协议

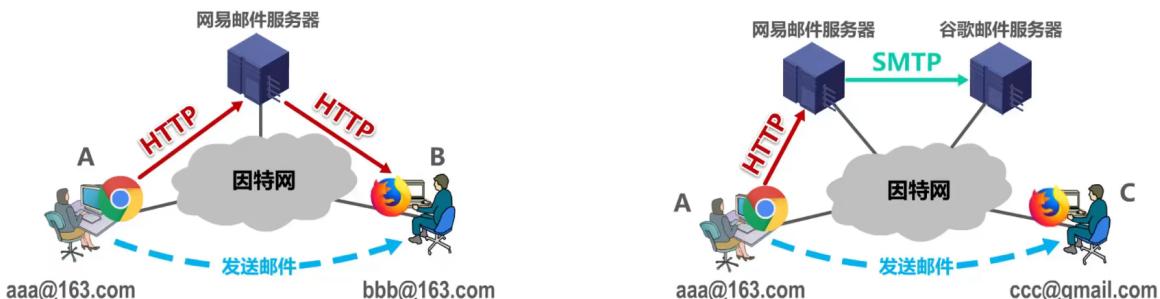
■ 常用的邮件读取协议有以下两个：

- **邮局协议POP** (Post Office Protocol) , POP3是其第三个版本，是因特网正式标准。
非常简单、功能有限的邮件读取协议。用户只能以**下载并删除方式**或**下载并保留方式**从邮件服务器下载邮件到用户方计算机。**不允许用户在邮件服务器上管理自己的邮件。**（例如创建文件夹，对邮件进行分类管理等）。
- **因特网邮件访问协议IMAP** (Internet Message Access Protocol) , IMAP4是其第四个版本，目前还只是因特网建议标准。
功能比POP3强大的邮件读取协议。**用户在自己的计算机上就可以操控邮件服务器中的邮箱**，就像在本地操控一样，因此IMAP是一个联机协议。
- POP3和IMAP4都采用**基于TCP连接的客户/服务器方式**。POP3使用熟知端口110，IMAP4使用熟知端口143。

基于万维网的电子邮件

■ 基于万维网的电子邮件

- 通过**浏览器登录**（提供用户名和口令）**邮件服务器万维网网站**就可以撰写、收发、阅读和管理电子邮件。这种工作模式与IMAP很类似，不同的是用户计算机无需安装专门的用户代理程序，只需要使用通用的万维网浏览器。
- 邮件服务器网站通常都提供非常强大和方便的邮件管理功能，用户可以在邮件服务器网站上管理和处理自己的邮件，而不需要将邮件下载到本地进行管理。



万维网

万维网并非某种特殊的计算机网络。它是一个大规模的、联机式的信息储藏所，是运行在因特网上的一个分布式应用

万维网利用网页之间的超链接将不同网站的网页链接成一张逻辑上的信息网

浏览器最重要的部分是**渲染引擎**，也就是**浏览器内核**。负责对网页内容进行解析和显示

- 不同的浏览器内核对网页内容的解析也有不同，因此同一网页在不同内核的浏览器里显示的效果可能不同
- 网页编写者需要在不同内核的浏览器中测试网页显示效果

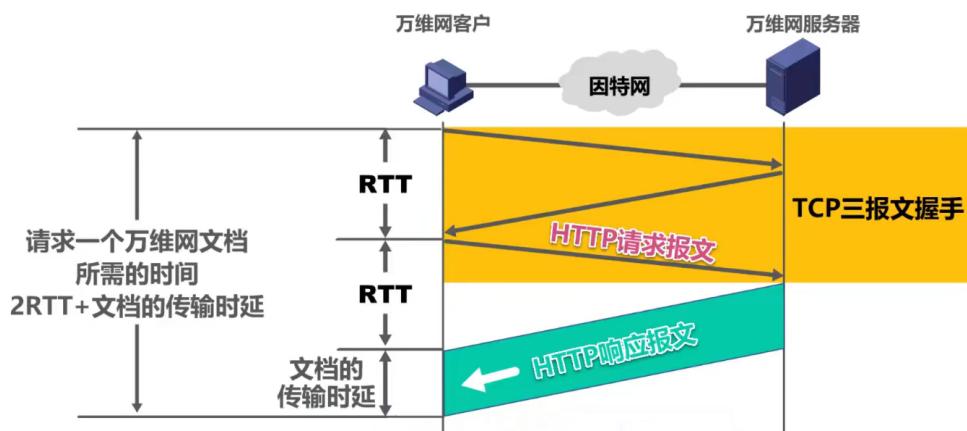
- 为了方便地访问在世界范围的文档，万维网使用统一资源定位符 URL 来指明因特网上任何种类"资源"的位置
- URL 的一般形式由以下四个部分组成

<协议>://<主机>:<端口>/<路径>

超文本传输协议HTTP

HTTP 定义了浏览器(即万维网进程)怎样向万维网服务器请求万维网文档，以及万维网服务器怎样把万维网文档传送给浏览器

- HTTP/1.0采用**非持续连接**方式。在该方式下，每次浏览器要请求一个文件都要与服务器建立TCP连接，当收到响应后就立即关闭连接。
 - 每请求一个文档就要有两倍的RTT的开销。若一个网页上有很多引用对象（例如图片等），那么请求每一个对象都需要花费2RTT的时间。
 - 为了减小时延，**浏览器通常会建立多个并行的TCP连接同时请求多个对象**。但是，这会大量占用万维网服务器的资源，特别是万维网服务器往往要同时服务于大量客户的请求，这会使负担很重。

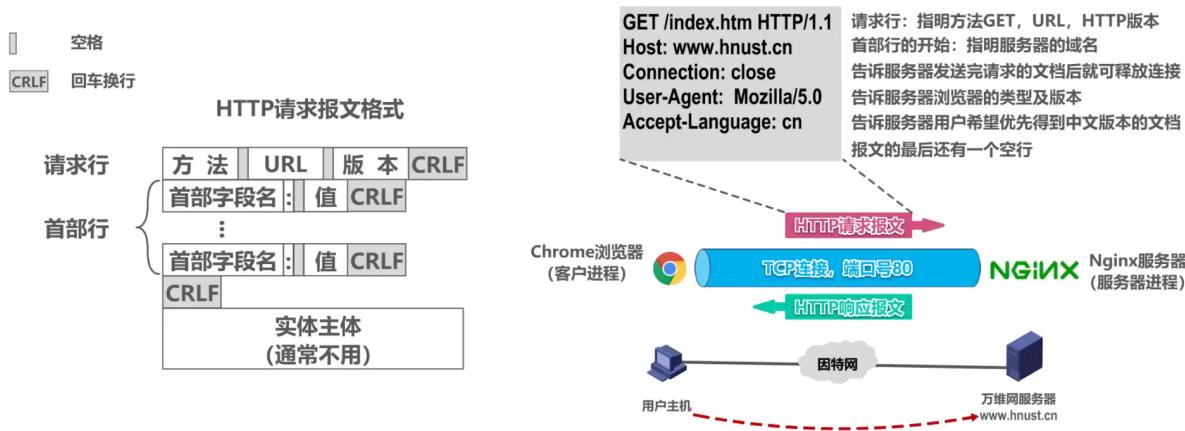


- HTTP/1.1采用**持续连接**方式。在该方式下，万维网服务器在发送响应后仍然保持这条连接，使同一个客户（浏览器）和该服务器可以继续在这条连接上传送后续的HTTP请求报文和响应报文。这并不局限于传送同一个页面上引用的对象，而是只要这些文档都在同一个服务器上就行。
 - 为了进一步提高效率，HTTP/1.1的持续连接还可以使用**流水线**方式工作，即浏览器在收到HTTP的响应报文之前就能够连续发送多个请求报文。这样的一个接一个的请求报文到达服务器后，服务器就发回一个接一个的响应报文。这样就节省了很多个RTT时间，使TCP连接中的空闲时间减少，提高了下载文档的效率。

HTTP请求报文

■ HTTP的报文格式

HTTP是面向文本的，其报文中的每一个字段都是一些ASCII码串，并且每个字段的长度都是不确定的。



HTTP响应报文



Cookie

Cookie提供了一种机制使得万维网服务器能够“记住”用户，而无需用户主动提供用户标识信息。也就是说，Cookie是一种对无状态的HTTP进行状态化的技术

■ 使用Cookie在服务器上记录用户信息



万维网缓冲与代理服务器

■ 万维网缓存与代理服务器

- 在万维网中还可以使用缓存机制以提高万维网的效率。
- 万维网缓存又称为**Web缓存** (Web Cache)，可位于客户机，也可位于中间系统上，位于中间系统上的Web缓存又称为**代理服务器** (Proxy Server)。
- Web缓存把最近的一些请求和响应暂存在本地磁盘中。当新请求到达时，若发现这个请求与暂时存放的请求相同，就返回暂存的响应，而不需要按URL的地址再次去因特网访问该资源。

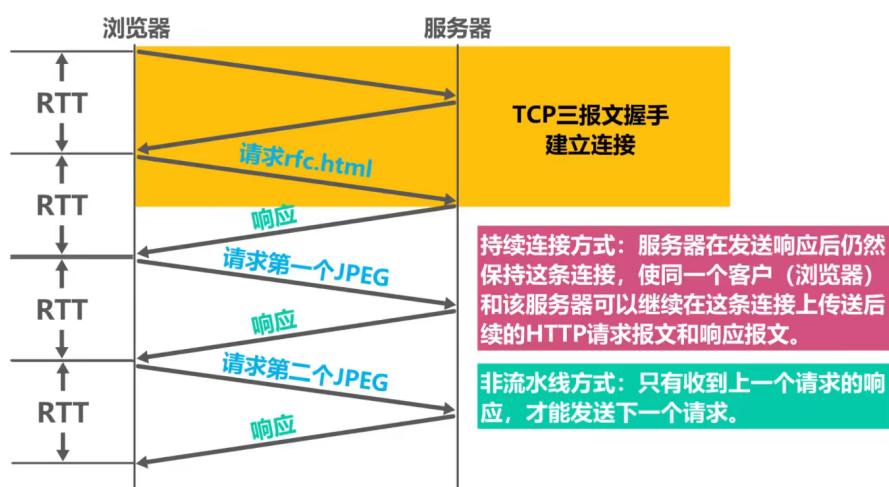


如果原始服务器中的文档已经修改，但是代理服务器中仍然有缓存，那么主机访问文档时是否会访问到未修改的文档从而发送错误呢？

实际上，原始服务器会为每个响应的对象设定一个**修改时间字段**和一个**有效日期字段**。当请求到达代理服务器时，若文档未过期则直接返回响应报文，否则向原始服务器发送请求。请求头部中有**If-modified-since**字段，记录了上次文档的修改时间，如果代理服务器中文档的修改时间与原始服务器中文档的修改时间一致，说明文档没有发生变化，因此原始服务器返回一个空文档，否则返回新文档

【修改自 2011年 题47 第（3）问】假设HTTP1.1协议以持续的非流水线方式工作，一次请求-响应的时间为RTT，rfc.html页面引用了2个JPEG小图像，则浏览器从开始建立TCP连接到收到全部内容为止，需要多少个RTT？

【解析】



防火墙

防火墙主要用于保护一个网络区域免受来自另一个网络区域的网络攻击和网络入侵行为，主要部署在网络边界，对进出网络的访问行为进行控制，安全防护是其核心特性，比如在企业网中，在与 Internet 接口处布置防火墙，可以起到过滤病毒、阻止黑客攻击等好处

安全领域

安全区域 zone 是一个或多个接口的集合，是防火墙区别路由器的主要特性，防火墙通过安全区域来划分网络，标识报文流动的“路线”，一般来说，当报文在不同的安全区域流动时，才会受到控制。

在华为防火墙中，一个接口只能加入到一个安全区域

默认安全领域

华为防火墙的默认安全区域： Trust、 DMZ 和 Untrust

区域名	受信任程度	安全级别	何时使用
Trust	高	85	定义内部用户所在网络
DMZ	中等	50	定义内部服务器所在网络
Untrust	低	5	定义 Internet 等不安全的网络
Local	代表防火墙本身	100	

凡是防火墙主动发出的报文均可认为是 Local 区域发出的，凡是需要防火墙响应并处理的报文均可认为是 Local 区域接收。 Local 不能添加接口，因为防火墙的所有接口都隐含属于 Local 区域

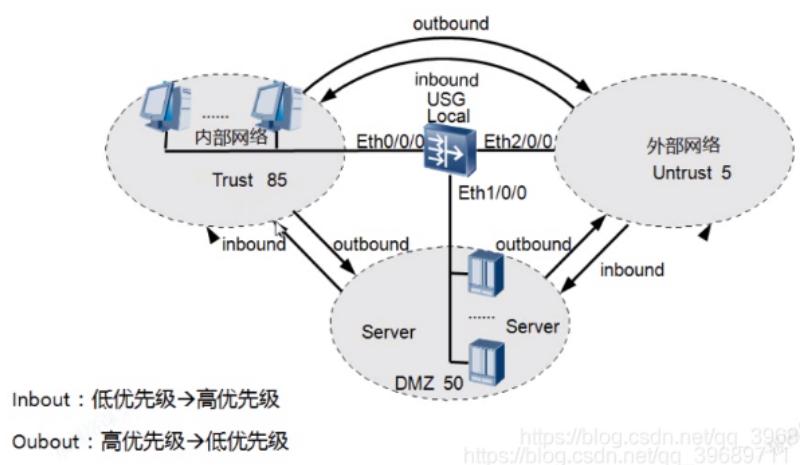
在网络数量较少网络中，使用默认的安全区域就可以满足划分网络的需求，反之，则需要创建新的安全区域。

安全域间

安全域间是**两个安全区域的唯一道路**，也可以用来描述流量的传输通道，任意两个安全区域都可构成一个安全域间。如果希望对经过这条通道的流量进行控制，就必须配置安全策略。

报文在两个安全区域之间流动时，报文从低级别的安全区域向高级别的安全区域流动时为入方向 **Inbound**，报文从高级别的安全区域向低级别的安全区域流动时为出方向 **Outbound**。

通信双方一定会交互报文，即安全区域的两个方向上都有报文的传输，通过设置安全区域，防火墙的安全区域之间有等级明确的域间关系，不同的安全区域代表不同网络，防火墙成为连接各个网络的节点，以此为基础，防火墙可以对各网络之间流动的报文实施管控。



防火墙内不允许定义优先级一样的区域，同等优先级的区域无法识别

Inbound、Outbound

三种工作模式

- **路由模式**：采用路由模式时，可以完成 ACL 包过滤、NAT 转换等功能，**Trust** 与 **Untrust** 区域间有一台防护墙，防火墙左右接口 IP 不同，需要使用路由表指导报文的转发，此种模式就是路由。
- **透明模式**：透明模式的防火墙支持 ACL 规则检查，防攻击检查、流量监控等功能，报文在防火墙当中不仅仅像是交换机的二层处理，还会对报文进行高层分析处理，两个区域 IP 属于同一网段，防火墙接口没有配置 IP，工作成交换机模式。

- **混合模式**: Trust与Untrust区域都使用同一网段，两台防火墙，组成主备关系，防火墙工作为路由模式，左右流量是二层转发，上下流量是三层转发。