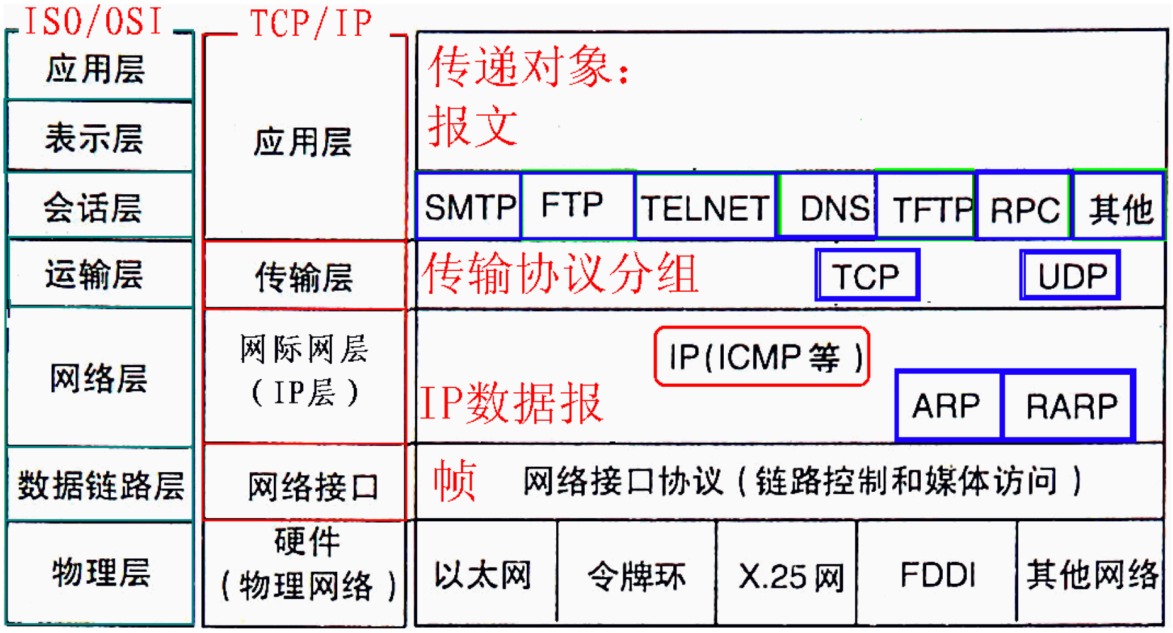
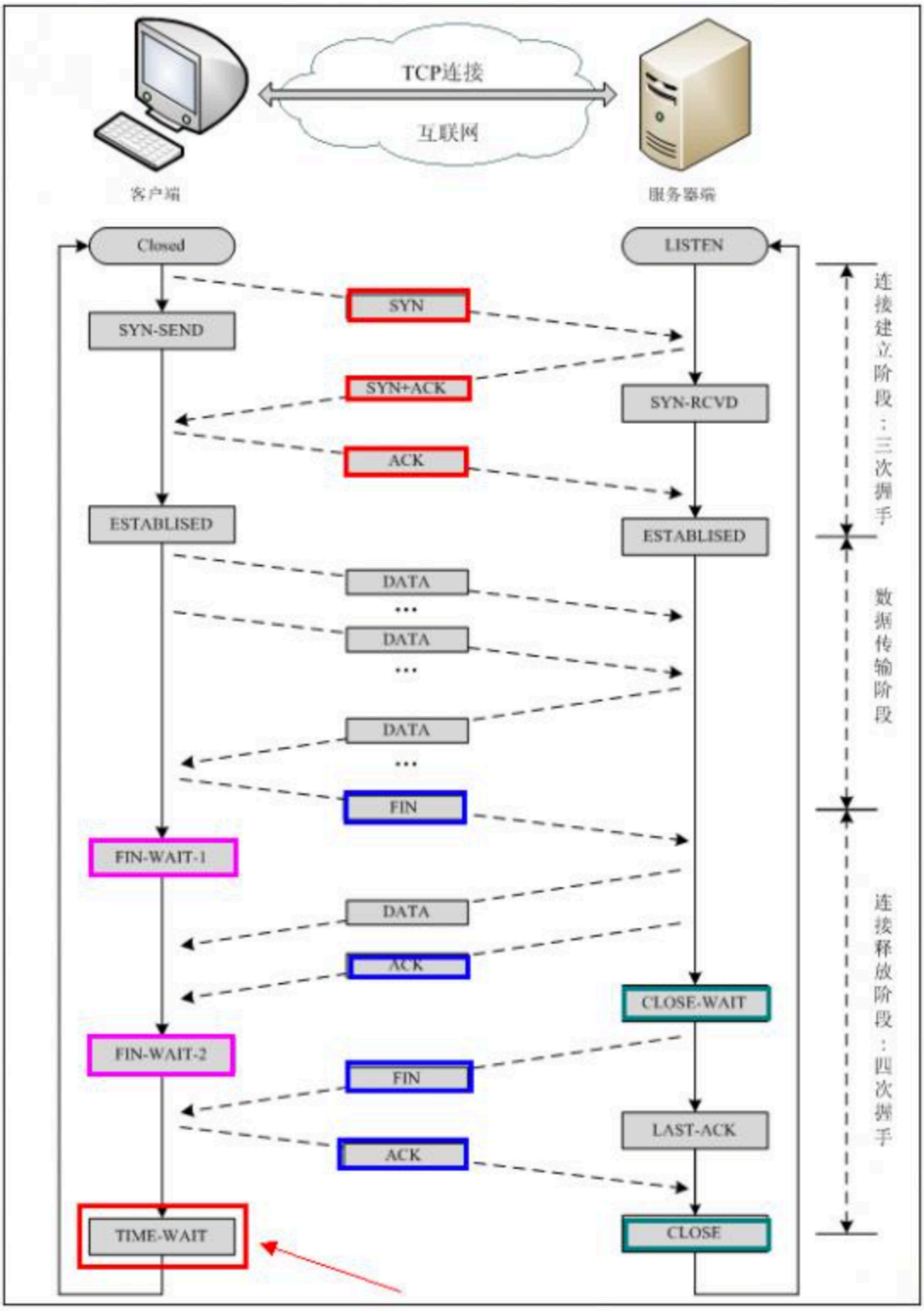
计算机网络基础

TCP/IP四层协议

- 应用层 `SMTP,FTP,HTTP,DND,TELNET,RPC等等`
- 传输层 `TCP,UDP`
- 网络层 `IP,ICMP,ARP,RAPP`
- 网络接口 `DATA LINK`



TCP的三次握手和四次挥手



注：seq:"sequance"序列号；**ack**:"acknowledge"确认号；**SYN**:"synchronize"请求同步标志；；**ACK**:"acknowledge"确认标志"；**FIN**："Finally"结束标志。

TCP连接建立过程：首先Client端发送连接请求报文，Server段接受连接后回复ACK报文，并为这次连接分配资源。Client端接收到ACK报文后也向Server段发生ACK报文，并分配资源，这样TCP连接就建立了。

TCP连接断开过程：假设Client端发起中断连接请求，也就是发送FIN报文。Server端接到FIN报文后，意思是说"我Client端没有数据要发给你了"，但是如果你还有数据没有发送完成，则不必急着关闭Socket，可以继续发送数据。所以你先发送ACK，"告诉Client端，你的请求我收到了，但是我还没准备好，请继续你等我的消息"。这个时候Client端就进入FIN_WAIT状态，继续等待Server端的FIN报文。当Server端确定数据已发送完成，则向Client端发送FIN报文，"告诉Client端，好了，我这边数据发完了，准备好关闭连接了"。Client端收到FIN报文后，"就知道可以关闭连接了，但是他还是不相信网络，怕Server端不知道要关闭，所以发送ACK后进入TIME_WAIT状态，如果Server端没有收到ACK则可以重传。“，Server端收到ACK后，"就知道可以断开连接了"。Client端等待了2MSL后依然没有收到回复，则证明Server端已正常关闭，那好，我Client端也可以关闭连接了。Ok，TCP连接就这样关闭了！

为什么要三次挥手?

在只有两次“握手”的情形下，假设Client想跟Server建立连接，但是却因为中途连接请求的数据报丢失了，故Client端不得不重新发送一遍；这个时候Server端仅收到一个连接请求，因此可以正常的建立连接。但是，有时候Client端重新发送请求不是因为数据报丢失了，而是有可能数据传输过程因为网络并发量很大在某结点被阻塞了，这种情形下Server端将先后收到2次请求，并持续等待两个Client请求向他发送数据...问题就在这里，Cient端实际上只有一次请求，而Server端却有2个响应，极端的情况可能由于Client端多次重新发送请求数据而导致Server端最后建立了N多个响应在等待，因而造成极大的资源浪费！所以，“三次握手”很有必要！

为什么要四次挥手?

试想一下，假如现在你是客户端你想断开跟Server的所有连接该怎么做？第一步，你自己先停止向Server端发送数据，并等待Server的回复。但事情还没有完，虽然你自身不往Server发送数据了，但是因为你们之前已经建立好平等的连接了，所以此时他也有主动权向你发送数据；故Server端还得终止主动向你发送数据，并等待你的确认。其实，说白了就是保证双方的一个合约的完整执行！

UDP协议

UDP用户数据报协议，是面向无连接的通讯协议，UDP数据包括目的端口号和源端口号信息，由于通讯不需要连接，所以可以实现广播发送。UDP通讯时不需要接收方确认，属于不可靠的传输，可能会出现丢包现象，实际应用中要求程序员编程验证。

UDP与TCP位于同一层，但它不管数据包的顺序、错误或重发。因此，UDP不被应用于那些使用虚电路的面向连接的服务，UDP主要用于那些面向查询---应答的服务，例如NFS。相对于FTP或Telnet，这些服务需要交换的信息量较小。

每个UDP报文分UDP报头和UDP数据区两部分。报头由四个16位长（2字节）字段组成，分别说明该报文的源端口、目的端口、报文长度以及校验值。UDP报头由4个域组成，其中每个域各占用2个字节，具体如下：

（1）源端口号；

（2）目标端口号；

（3）数据报长度；

（4）校验值。

使用UDP协议包括：TFTP（简单文件传输协议）、SNMP（简单网络管理协议）、DNS（域名解析协议）、NFS、BOOTP。

TCP 与 UDP 的区别：TCP是面向连接的，可靠的字节流服务；UDP是面向无连接的，不可靠的数据报服务。

DNS协议

DNS是域名系统(DomainNameSystem)的缩写，该系统用于命名组织到域层次结构中的计算机和网络服务，**可以简单地理解为将URL转换为IP地址**。域名是由圆点分开一串单词或缩写组成的，每一个域名都对应一个惟一的IP地址，在Internet上域名与IP地址之间是一一对应的，DNS就是进行域名解析的服务器。DNS命名用于Internet等TCP/IP网络中，通过用户友好的名称查找计算机和服务。

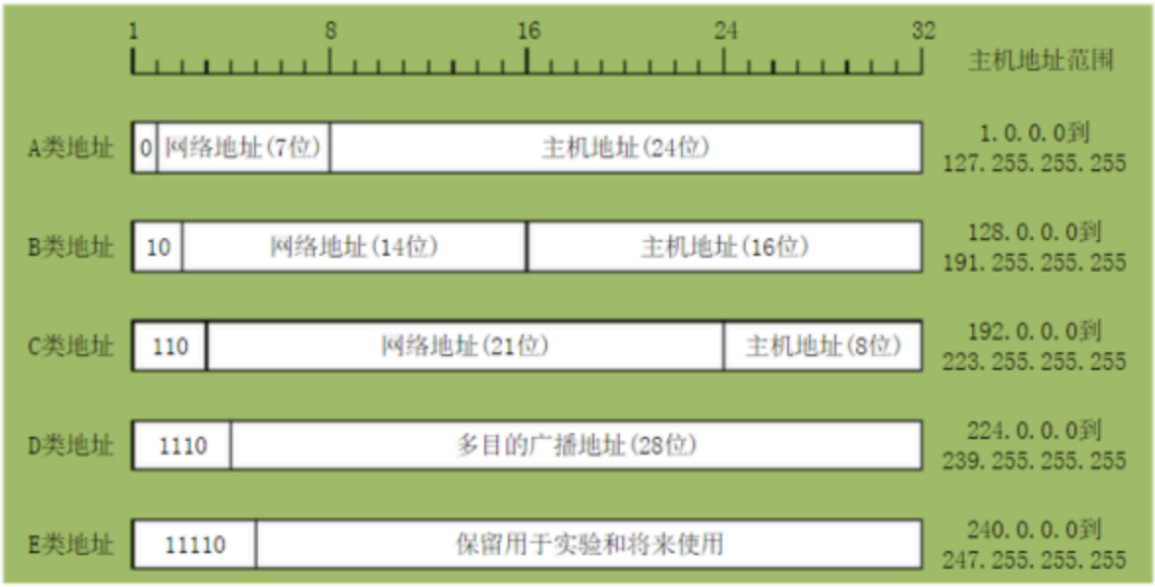
NAT协议

NAT网络地址转换(Network Address Translation)属接入广域网(WAN)技术，是一种将私有（保留）地址转化为合法IP地址的转换技术，它被广泛应用于各种类型Internet接入方式和各种类型的网络中。原因很简单，NAT不仅完美地解决了lP地址不足的问题，而且还能够有效地避免来自网络外部的攻击，隐藏并保护网络内部的计算机。

DHCP协议

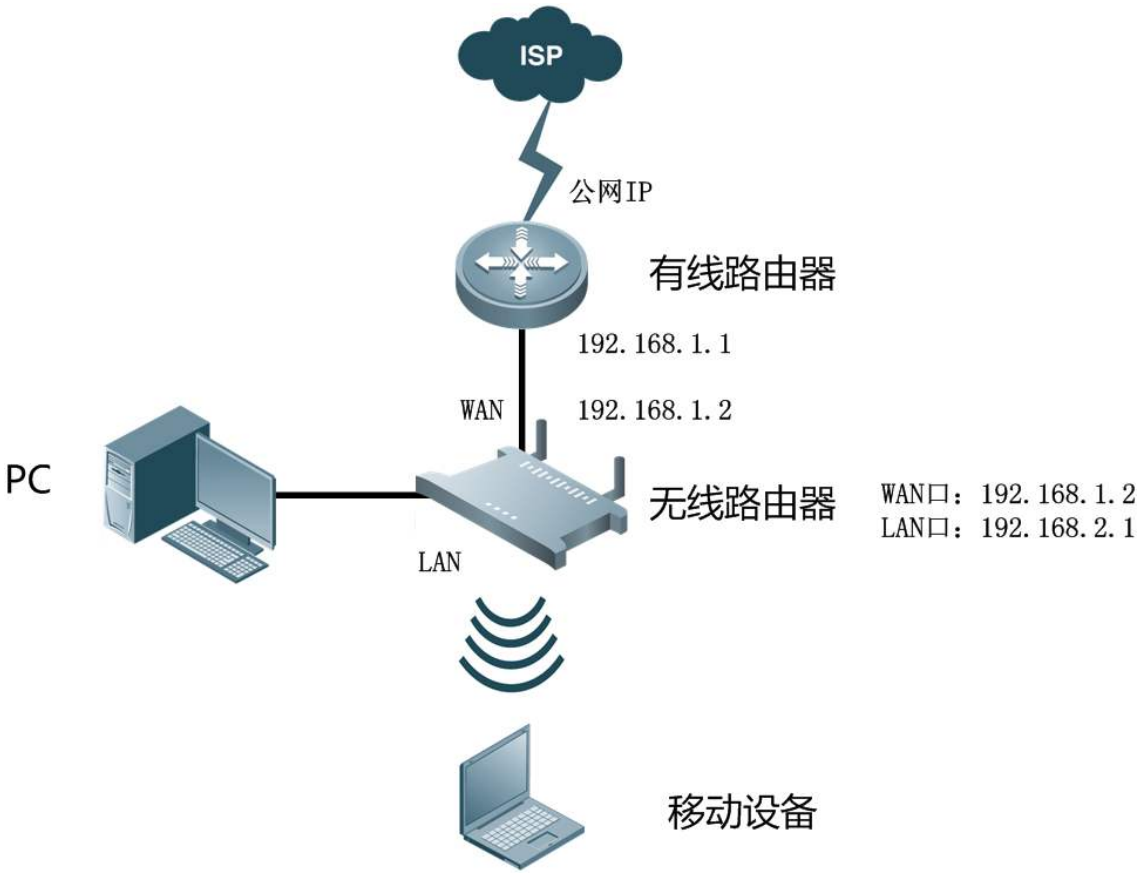
DHCP动态主机设置协议（Dynamic Host Configuration Protocol）是一个局域网的网络协议，使用UDP协议工作，主要有两个用途：给内部网络或网络服务供应商自动分配IP地址，给用户或者内部网络管理员作为对所有计算机作中央管理的手段。

- **IP地址**
    - IP地址是一个32位的二进制数，通常被分割为4个“8位二进制数”（也就是4个字节）。IP地址通常用“点分十进制”表示成（a.b.c.d）的形式，其中，a,b,c,d都是0~255之间的十进制整数。



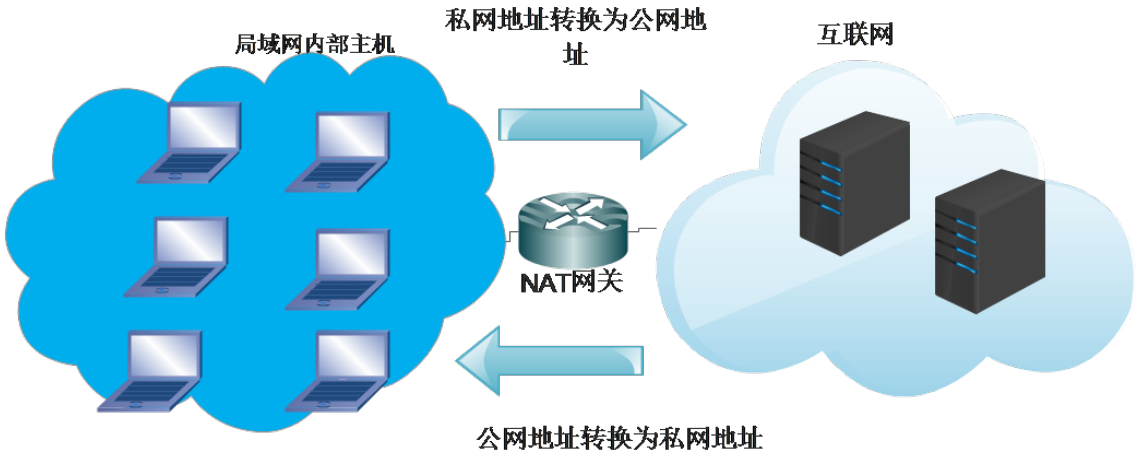
- **子网掩码**
  - 子网掩码(subnet mask)又叫网络掩码、地址掩码、子网络遮罩，它是一种用来指明一个IP地址的哪些位标识的是主机所在的子网，以及哪些位标识的是主机的位掩码。

    子网掩码不能单独存在，它必须结合IP地址一起使用。子网掩码只有一个作用，就是将某个IP地址划分成网络地址和主机地址两部分。
    子网掩码是一个32位地址，用于屏蔽IP地址的一部分以区别网络标识和主机标识，并说明该IP地址是在局域网上，还是在远程网上。
  - 通过子网掩码，就可以判断两个IP在不在一个局域网内部。
  - 子网掩码可以看出有多少位是网络号，有多少位是主机号
    - IP地址二进制&子网掩码二进制 = 网络地址二进制 --->十进制
    - IP地址二进制&(~子网掩码二进制) = 主机地址
- **路由器**
  - 路由器的实质是一种将网络进行互联的专用计算机，路由器在TCP/IP中又称为网关。
  - 下图是一个简单的的路由器结构



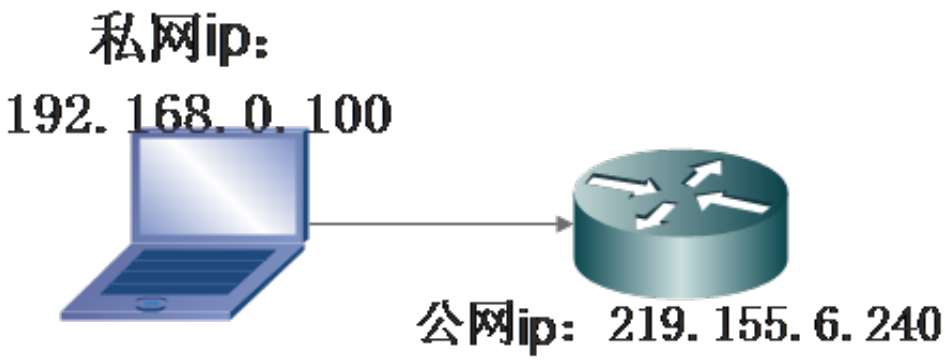
NAT

- 示意图

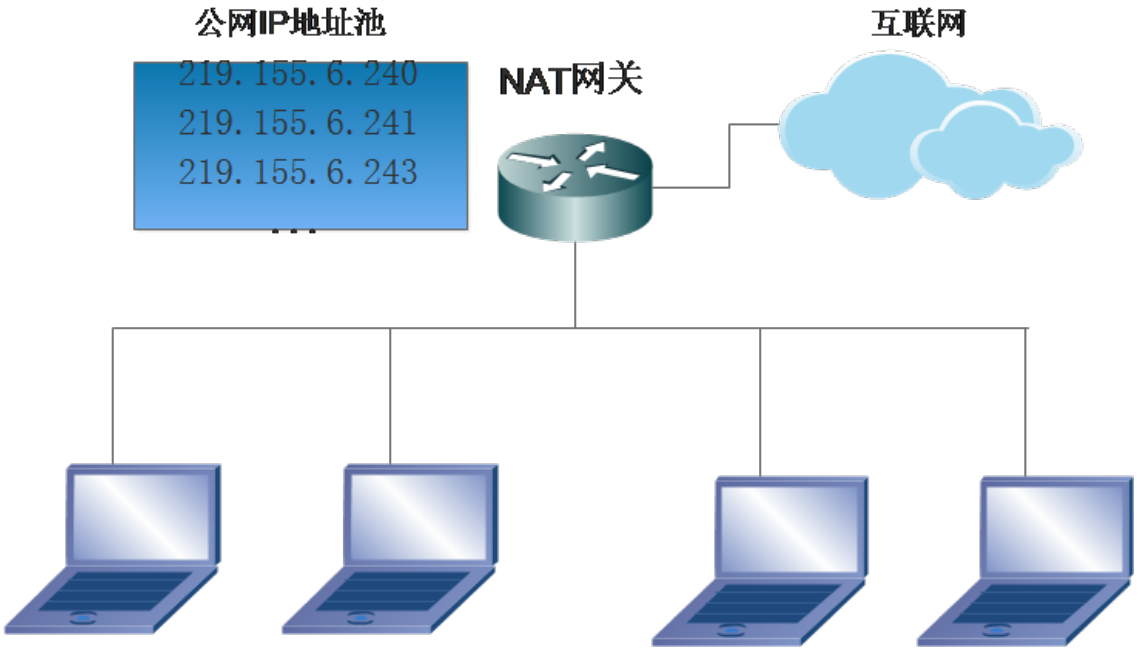


- 特点:
  - 网络被分为私网和公网两个部分，NAT网关设置在私网到公网的路由出口位置，双向流量必须都要经过NAT网关;
  - **网络访问只能先由私网侧发起，公网无法主动访问私网主机；**
  - NAT网关在两个访问方向上完成两次地址的转换或翻译，出方向做源信息替换，入方向做目的信息替换；
  - NAT网关的存在对通信双方是保持透明的；
  - NAT网关为了实现双向翻译的功能，需要维护一张关联表，把会话的信息保存下来。
- **静态NAT**

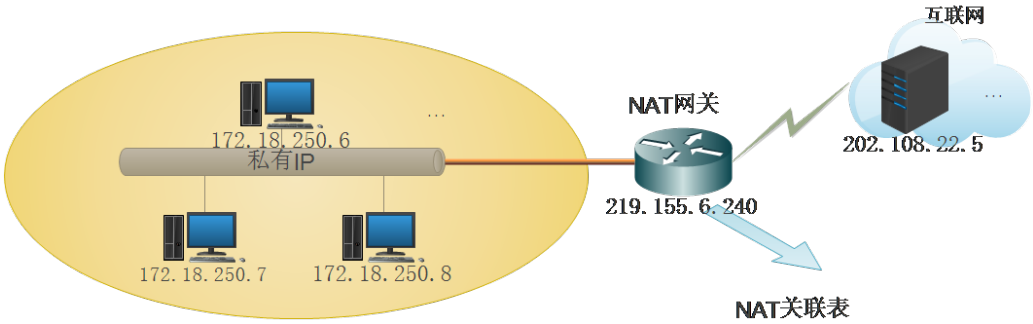
- **如果一个内部主机唯一占用一个公网IP，这种方式被称为一对一模型。此种方式下，转换上层协议就是不必要的，因为一个公网IP就能唯一对应一个内部主机。显然，这种方式对节约公网IP没有太大意义，主要是为了实现一些特殊的组网需求。比如用户希望隐藏内部主机的真实IP，或者实现两个IP地址重叠网络的通信**



- **动态NAT**
  - 它能够将未注册的IP地址映射到注册IP地址池中的一个地址。不像使用静态NAT那样，你无需静态地配置路由器，使其将每个内部地址映射到一个外部地址，但必须有足够的公有因特网IP地址，让连接到因特网的主机都能够同时发送和接收分组



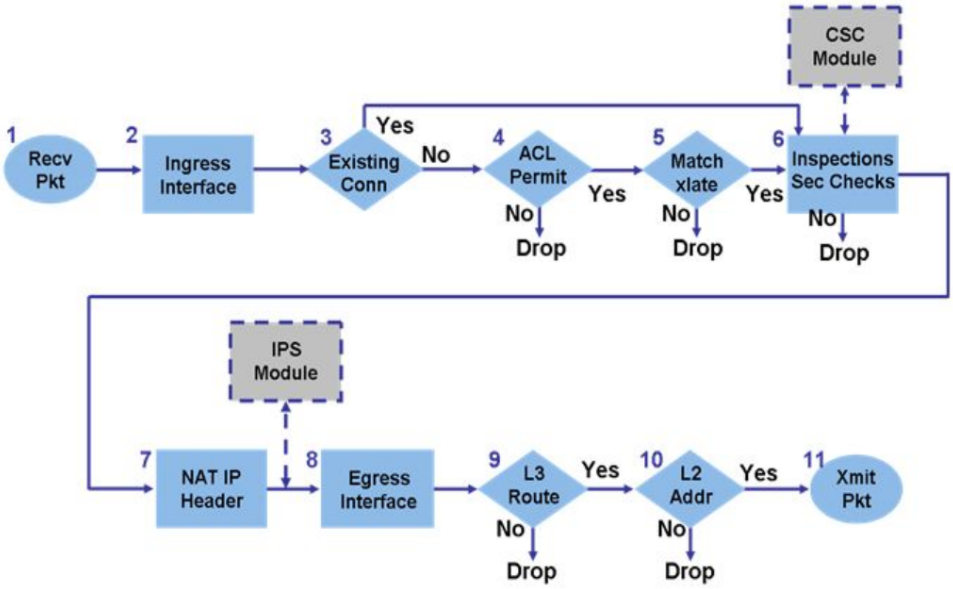
- NAT重载
  - 这是最常用的NAT类型。NAT重载也是动态NAT，它利用源端口将多个私网ip地址映射到一个公网ip地址(多对一)。那么，它的独特之处何在呢?它也被称为端口地址特换(PAT)。通过使用PAT(NAT重载)，只需使用一个公网ip地址，就可将数千名用户连接到因特网。其核心之处就在于利用端口号实现公网和私网的转换。



协议	私网的私有ip地址	公网Ip+端口号	目的地址Ip+端口号
TCP	172. 18. 250. 6	219. 155. 6. 240:1723	202. 108. 22. 5:80
TCP	172. 18. 250. 7	219. 155. 6. 240:1026	202. 108. 22. 5:80
TCP	172. 18. 250. 8	219. 155. 6. 240:1492	202. 108. 22. 5:80

防火墙

- 数据包在防火墙中的传递过程



1. 数据包抵达接口
2. 匹配connection表项
3. ACL检查
4. 匹配地址转换
5. 深度包检查
6. IP头转换
7. 抵达外出接口
8. 3层路由表查找
9. 2层mac地址查找
10. 发送数据包

ARP/RARP协议

> **地址解析协议，即ARP（Address Resolution Protocol），是根据IP地址获取物理地址的一个TCP/IP协议。**主机发送信息时将包含目标IP地址的ARP请求广播到网络上的所有主机，并接收返回消息，以此确定目标的物理地址；收到返回消息后将该IP地址和物理地址存入本机ARP缓存中并保留一定时间，下次请求时直接查询ARP缓存以节约资源。地址解析协议是建立在网络中各个主机互相信任的基础上的，网络上的主机可以自主发送ARP应答消息，其他主机收到应答报文时不会检测该报文的真实性就会将其记入本机ARP缓存；由此攻击者就可以向某一主机发送伪ARP应答报文，使其发送的信息无法到达预期的主机或到达错误的主机，这就构成了一个ARP欺骗。**ARP命令可用于查询本机ARP缓存中IP地址和MAC地址的对应关系、添加或删除静态对应关系等。**

ARP工作流程举例：

主机A的IP地址为192.168.1.1，MAC地址为0A-11-22-33-44-01；

主机B的IP地址为192.168.1.2，MAC地址为0A-11-22-33-44-02；

当主机A要与主机B通信时，地址解析协议可以将主机B的IP地址（192.168.1.2）解析成主机B的MAC地址，以下为工作流程：

（1）根据主机A上的路由表内容，IP确定用于访问主机B的转发IP地址是192.168.1.2。然后A主机在自己的本地ARP缓存中检查主机B的匹配MAC地址。

（2）如果主机A在ARP缓存中没有找到映射，它将询问192.168.1.2的硬件地址，从而将ARP请求帧广播到本地网络上的所有主机。源主机A的IP地址和MAC地址都包括在ARP请求中。本地网络上的每台主机都接收到ARP请求并且检查是否与自己的IP地址匹配。如果主机发现请求的IP地址与自己的IP地址不匹配，它将丢弃ARP请求。

（3）主机B确定ARP请求中的IP地址与自己的IP地址匹配，则将主机A的IP地址和MAC地址映射添加到本地ARP缓存中。

（4）主机B将包含其MAC地址的ARP回复消息直接发送回主机A。

（5）当主机A收到从主机B发来的ARP回复消息时，会用主机B的IP和MAC地址映射更新ARP缓存。本机缓存是有生存期的，生存期结束后，将再次重复上面的过程。主机B的MAC地址一旦确定，主机A就能向主机B发送IP通信了。

逆地址解析协议，即RARP，功能和ARP协议相对，其将局域网中某个主机的物理地址转换为IP地址，比如局域网中有一台主机只知道物理地址而不知道IP地址，那么可以通过RARP协议发出征求自身IP地址的广播请求，然后由RARP服务器负责回答。

RARP协议工作流程：

（1）给主机发送一个本地的RARP广播，在此广播包中，声明自己的MAC地址并且请求任何收到此请求的RARP服务器分配一个IP地址；

（2）本地网段上的RARP服务器收到此请求后，检查其RARP列表，查找该MAC地址对应的IP地址；

（3）如果存在，RARP服务器就给源主机发送一个响应数据包并将此IP地址提供给对方主机使用；

（4）如果不存在，RARP服务器对此不做任何的响应；

（5）源主机收到从RARP服务器的响应信息，就利用得到的IP地址进行通讯；如果一直没有收到RARP服务器的响应信息，表示初始化失败。

路由选择协议

常见的路由选择协议有：RIP协议、OSPF协议。

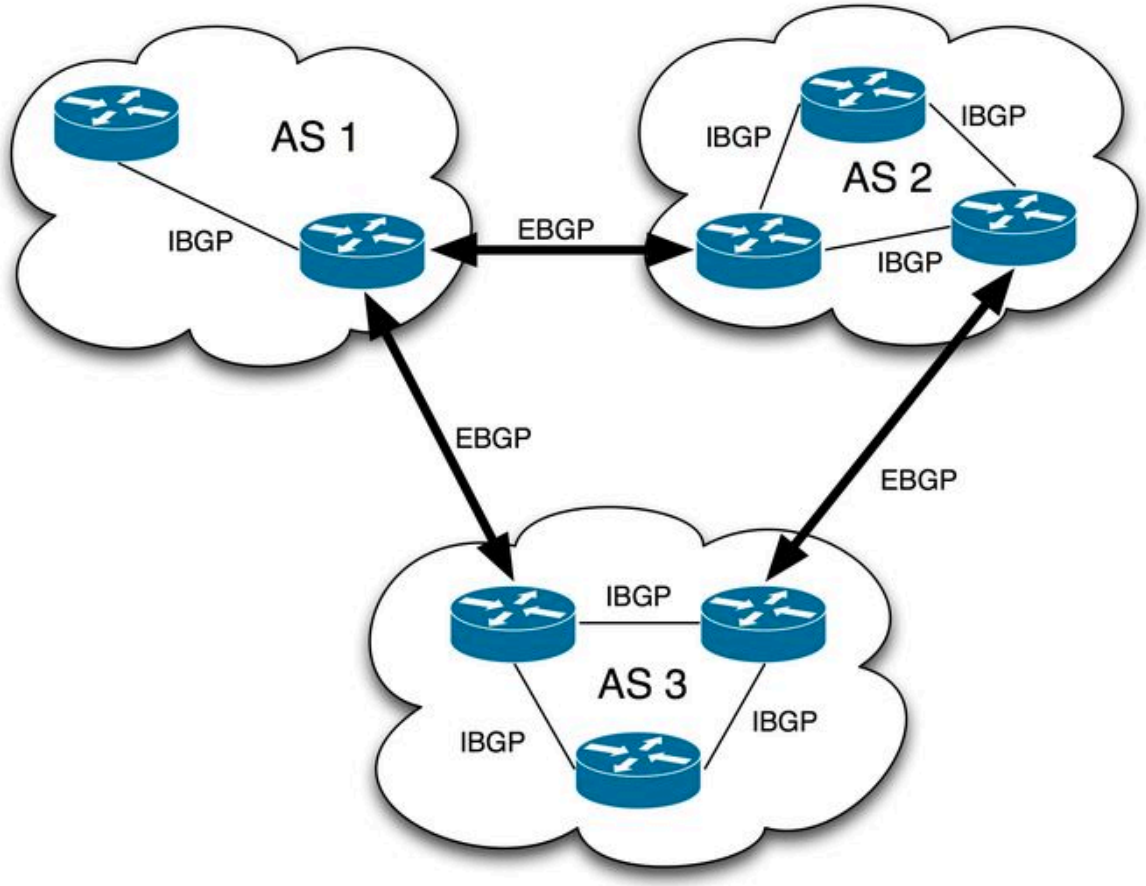
RIP协议： 底层是贝尔曼福特算法(Bellman–Ford algorithm)，它选择路由的度量标准（metric)是跳数，最大跳数是15跳，如果大于15跳，它就会丢弃数据包。

OSPF协议： Open Shortest Path First开放式最短路径优先，底层是迪杰斯特拉算法（Dijkstra algorithm），是链路状态路由选择协议，它选择路由的度量标准是带宽，延迟。

BGP协议: Border Gataway Protocol BGP是互联网上一个核心的去中心化自治路由协议。(目前是最重要也是互联网上唯一使用的路由协议)。

- AS（Autonomous system）： 自治系统，指在一个（有时是多个）组织管辖下的所有IP网络和路由器的全体，它们对互联网执行共同的路由策略。也就是说，对于互联网来说，一个AS是一个独立的整体网络。而BGP实现的网络自治也是指各个AS自治。每个AS有自己唯一的编号。
- IGP（Interior Gateway Protocol）： 内部网关协议，在一个AS内部所使用的一种路由协议。一个AS内部也可以有多个路由器管理多个网络。各个路由器之间需要路由信息以知道子网络的可达信息。IGP就是用来管理这些路由。代表的实现有RIP和OSPF。
- EGP（Exterior Gateway Protocol）： 外部网关协议，在多个AS之间使用的一种路由协议，现在已经淘汰，被BGP取而代之。

由于BGP就是为了替换EGP而创建，它的地位与EGP相似。但是BGP也可以应用在一个AS内部。因此BGP又可以分为IBGP（Interior BGP： 同一个AS之间的连接）和EBGP（Exterior BGP：不同AS之间的BGP连接）。既然EGP已经被替代了，那EBGP的存在比较好理解，但是IGP协议都还活得好好的（这里指的是OSPF），那IBGP的意义何在？IGP的协议是针对同一个AS网络来设计的，一个自治网络的规模一般都不大，所以设计的时候就没有考虑大规模网络的情况。而当一个自治网络足够大时，OSPF存在性能瓶颈（后面会说明）。BGP本身就是为了在Internet工作，其设计就是为了满足大型网络的要求，所以大型私有IP网络内部可以使用IBGP。总的来说，这几类路由协议，小规模私有网络IGP，大规模私有网络IBGP，互联网EBGP



BGP可以说是最复杂的路由协议。它是应用层协议，其传输层使用TCP，默认端口号是179。因为是应用层协议，可以认为它的连接是可靠的，并且不用考虑底层的工作，例如fragment，确认，重传等等。BGP是唯一使用TCP作为传输层的路由协议，其他的路由协议可能都还到不了传输层。

TCP连接的窗口是65K字节，也就是说TCP连接允许在没有确认包的情况下，连续发送65K的数据。而其他的路由协议，例如EIGRP和OSPF的窗口只有一个数据包，也就是说前一个数据包收到确认包之后，才会发送下一个数据包。当网络规模巨大时，需要传输的数据也相应变大，这样效率是非常低的。这也是它们不适合大规模网络的原因。而正是由于TCP可以可靠的传输大量数据，且互联网的路由信息是巨大的，TCP被选为BGP的传输层协议，并且BGP适合大规模网络环境。

BGP如何工作

BGP是一种路径矢量协议（Path vector protocol）的实现。因此，它的工作原理也是基于路径矢量。首先说明一下，下面说的BGP route指的是BGP自己维护的路由信息，区分于设备的主路由表，也就是我们平常看见的那个路由表。BGP route是BGP协议传输的数据，并存储在BGP router的数据库中。并非所有的BGP route都会写到主路由表。每条BGP route都包含了目的网络，下一跳和完整的路径信息。路径信息是由AS号组成，当BGP router收到了一条 路由信息，如果里面的路径包含了自己的AS号，那它就能判定这是一条自己曾经发出的路由信息，收到的这条路由信息会被丢弃。

这里把每个BGP服务的实体叫做BGP router，而与BGP router连接的对端叫BGP peer。每个BGP router在收到了peer传来的路由信息，会存储在自己的数据库，前面说过，路由信息包含很多其他的信息，BGP router会根据自己本地的policy结合路由信息中的内容判断，如果路由信息符合本地policy，BGP router会修改自己的主路由表。本地的policy可以有很多，举个例子，如果BGP router收到两条路由信息，目的网络一样，但是路径不一样，一个是AS1->AS3->AS2，另一个是AS1->AS2，如果没有其他的特殊policy，BGP router会选用AS1->AS2这条路由信息。policy还有很多其他的，可以实现复杂的

控制。

除了修改主路由表，BGP router还会修改这条路由信息，将自己的AS号加在BGP数据中，将下一跳改为自己，并且将自己加在路径信息里。在这之后，这条消息会继续向别的BGP peer发送。而其他的BGP peer就知道了，可以通过指定下一跳到当前BGP router，来达到目的网络地址。

所以说，BGP更像是一个可达协议，可达信息传来传去，本地根据收到的信息判断决策，再应用到路由表。

在浏览器中输入 www.baidu.com 后执行的全部过程

现在假设如果我们在客户端（客户端）浏览器中输入http://www.baidu.com,而baidu.com为要访问的服务器（服务器），下面详细分析客户端为了访问服务器而执行的一系列关于协议的操作：

1）客户端浏览器通过DNS解析到www.baidu.com的IP地址220.181.27.48，通过这个IP地址找到客户端到服务器的路径。客户端浏览器发起一个HTTP会话到220.161.27.48，然后通过TCP进行封装数据包，输入到网络层。

2）在客户端的传输层，把HTTP会话请求分成报文段，添加源和目的端口，如服务器使用80端口监听客户端的请求，客户端由系统随机选择一个端口如5000，与服务器进行交换，服务器把相应的请求返回给客户端的5000端口。然后使用IP层的IP地址查找目的端。

3）客户端的网络层不用关系应用层或者传输层的东西，主要做的是通过查找路由表确定如何到达服务器，期间可能经过多个路由器，这些都是由路由器来完成的工作，不作过多的描述，无非就是通过查找路由表决定通过那个路径到达服务器。

4）客户端的链路层，包通过链路层发送到路由器，通过邻居协议查找给定IP地址的MAC地址，然后发送ARP请求查找目的地址，如果得到回应后就可以使用ARP的请求应答交换的IP数据包现在就可以传输了，然后发送IP数据包到达服务器的地址。