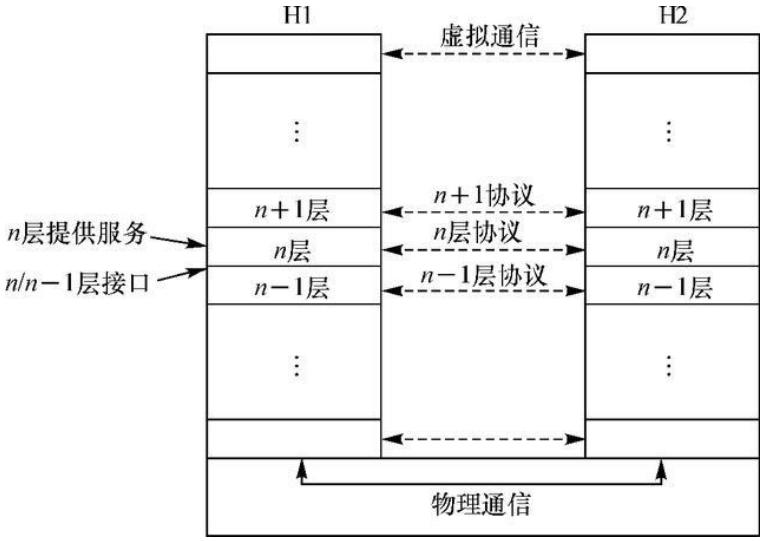
■ 知识点一：网络体系结构

1. 网络为何要分层

计算机网络通信技术显然不是件容易的事，两台计算机通信除了必须有的数据通道（无线通道也是通道）之外，还需要协议软件上的高度协调，软件协调通常是相当复杂的。

一般情况下，人们解决复杂问题的思路是先把大问题分解成多个子问题，再把相对容易解决的子问题逐一解决，计算机网络通信技术也不例外。

计算机网络采用分层体系结构，把大问题分成小问题。假设网络通信技术划分了N 层，从下到上依次为第1层，第2层，$\cdots$, 第N层，分层结构规定了每一层的功能。



上图描述的是主机H_1与主机H_2的分层情况。若主机H_1把数据传输给主机H_2，数据首先在主机H_1的最顶层第N 层形成，这个数据最终会被主机H_2的最顶层N层收到，原因是两台主机的各自的第N 层之间遵守了该层间的通信协议（协议是水平的），不过主机H_1第N层的数据并不是直接传输到H_2的第N 层，而是先通过层间接口先传输到自己的第$N-1$层，$N-1$ 层对N层下来的数据进行包装处理，包装后的数据再按照两主机$N-1$层之间的协议（协议还是水平的）最终会到达主机H_2的$N-1$ 层。

从上面的分析过程我们可以看出，主机H_1最顶层形成的发往H_2的数据先是在自己的体系结构内部经历了一个自上而下的过程，等数据到了主机H_2，又经历了一个自下而上的过程。主机H_1的每一层的数据最终会传输到H_2的对应层，但需要借助自己下一层的功能来完成这个传输，这个现象我们称之为下层为上层服务（服务是垂直的）。H_1体系结构的上层仅知道把该层的数据通过层间“接口”传输给下一层，至于下一层如何对待该数据，又最终如何把数据传输到对方的同一层，这是下一层的事，上层是不知道的，也无须知道。即所谓的层间“独立性”问题。

2. 两大有影响的网络体系结构

国际标准化组织ISO制定了著名的网络开放系统互联模型体系结构OSI/RM（Open Systems Interconnection reference model）简称OSI。OSI的七层协议体系结构概念清楚，理论也较完善，但因为比较繁琐和商业利益驱动等一系列原因被另一个网络体系结构TCP/IP所替代，后者在互联网INTERNET得到广泛的应用，成为事实上的工业标准。

3. TCP/IP协议簇

国际标准化组织制定的OSI模型共有七层，从下到上分别是物理层、链路层、网络层、传输层、会话层、表示层和应用层。

TCP/IP协议在一定程度上参考了OSI的体系结构，并对此作了适当的简化，具体是：

（1）OSI应用层、表示层、会话层三个层次提供的服务相差不是很大，在TCP/IP协议中，它们被合并为应用层一个层次

（2）传输层和网络层在体系结构中的地位十分重要，TCP/IP协议中它们还是作为独立的两个层次。

（3）数据链路层和物理层的内容相差不多，TCP/IP协议将它们归并为一层，名称叫网络接口层。

TCP/IP协议簇传输层和网络层的主要协议分别是是TCP和IP，用它们作为整个协议簇的名称，可见其重要性。传输层除了TCP协议，还有UDP协议，它们的主要区别在与前者是面向连接的可靠协议，后者是面向无连接的不可靠协议, 两个应用进程采用TCP 协议通信的过程分成连接建立、数据传输、连接释放三个阶段，TCP协议用一系列措施保证数据传输的正确性。

IP协议是网络层最重要的协议，它接收传输层TCP形成的数据（报文段）或UDP形成的数据（用户数据报），对其加上首部形成分组，然后“尽最大可能”地把分组传输到目标计算机。这种传输特性表明IP协议不是可靠的，IP 协议的分组交换网不同于传统电话通信的电路交换网，分组交换网的可靠性由端点保证，电路交换网的可靠性由中间的传输设备保证，这也就不难理解为何电话机比计算机便宜多了。

四层体系结构的TCP/IP 协议与七层体系结构的OSI 相比简单了不少，TCP/IP 协议在实际的应用中效率更高，成本更低。

■ 知识点二、网络连接设备

相距不远的两台计算机，可通过直连线路（有线或无线）直接通信。若两台计算机各自位于地球方向相反的两端，再通过直连线路把数据从一台发往另一台，根本就是不可能的了。

位于INTERNET上的、距离无论远近的两台计算机依然还可以通信，是因为它们之间有通信连接设备。通信连接设备可以是集线器、交换机、路由器和智能网关等，前三者居多。

1．集线器

集线器英文称为“Hub”。“Hub”就是“中心”的意思，“中心”的含义是指连接在集线器的计算机以集线器为中心互联互通。集线器的主要功能是对接收到的信号进行再生整形放大，以扩大网络的传输距离，它工作于OSI参考模型第一层，即“物理层”。集线器内部采用了总线型拓扑结构，在任意时刻数据传输都是单向的，维持在半双工（数据可以在两个方向，但不能同时）模式下。当某个接口收到数据，集线器会把数据转发到除接收端口之外的所有端口，这就是所谓的广播式发送。这种广播发送数据的方式有以下的不足：因把数据也发给了不是真正的接收者，很容易被某些别有用心的人截获利用，带来安全隐患；另外任何数据都采用广播式发送，极易造成网络拥塞，降低通信效率，集线器不需要配置。

Figure 1: 集线器

2．交换机

交换机与集线器外观上看上去差不多，工作原理差别就大了，换机工作在数据链路层，通过帧首部的目的MAC 地址转发数据，交换机上的两个端口之间的通道是相互独立的，可以实现全双工通信，交换机中存放了MAC地址与接口的对应表，交换机依据表中的地址找到转发端口，这种有目标的转发不像集线器式的“泛洪”式的广播方式转发，极大提高了通信效率。不过，需要注意的是，交换机对待“广播报文”与集线器没有区别，依然采用广播方式。交换机可以配置，主要配置vlan，实验没有涉及这方面内容。

Figure 2: 交换机

3．路由器

当有更多的用户加入网络时，必然需要更多的连接设备用以扩大网络规模，若采用连接设备是集线器或者交换机，按照前面的分析，广播域的范围会随着网络规模的扩大也随之扩大，后果就是导致通信效率的下降，因而采用集线器或交换机扩大规模已经比较大的网络是不可取的。

从理论上来说，开发一种既能扩大网络规模但又不扩大广播域的设备就可以解决上面的问题，事实上路由器就是这样的一款产品，用它连接两个网络并不会扩大广播域，其原因在于路由器会主动屏蔽发给它的所有广播信号。路由器能屏蔽广播信号仅仅是其作用之一，其更大作用则是路径选择等。路由器工作在网络层，当从某个接口收到分组，它能智能地“查看”分组首部中的目标IP地址，依据路由表选择转发出去的接口。

路由器内的路由表是靠静态路径录入或动态路由协议发现路径形成的记录到达目标网络条目的集合。最常见的路由协议有RIP和OSPF，它们对“最好路径”的理解不同，前者基于路径中的路由器数量最少，后者则基于链路带宽、延迟等因素，在多种链路组成的网络环境中，OSPF选路标准更合理。

路由器是用来连接网络的，其接口数比连接计算机的交换机接口数少许多。高档路由器都是模块化的，模块种类繁多，用以支持不同的链路，组网时需要根据实际情况灵活选择。

路由器必须配置，我们学习过静态路由、动态路由RIP、OSPF和DHCP等知识。

Figure 3: 路由器

■ 知识点三、对称秘码体制、非对称秘码体制、数字签名

广泛使用的TCP/IP协议（严格说是TCP/IP协议簇）是在美国国防部赞助下开发出来的，当时参与研究开发的单位仅限于几所大学和科研机构，彼此都是可信的，因而开发过程中没有过多地考虑通信的安全性问题。

随着源于TCP/IP协议的INTERNET的广泛普及，谁也没有料到，接入INTERNET的结点数量呈指数级增长，利用INTERNET解决的问题也越来越多。这时，INTERNET传输信息的安全问题也就逐渐地暴露出来了。

信息的安全性问题涉及许多方面，如用户A把某些涉密敏感信息发送给用户B时，不希望第三方看到, 属于信息机密性问题；

一些信息倒是不担心别人看到或者本来就是提供给别人看的，如国家政策、政府公文等，这些信息不允许篡改或伪造，属于信息的完整性问题；

还有一些信息一旦发送，发送者对此不能否认，属于信息不可否认性问题, 等等。

人们利用网络传输信息时, 为避免给网络用户造成损失, 所面临的上述安全问题都需要解决, 解决方法可以是重新开发一套新的安全网络通信体系结构, 或者在原有TCP/IP通信的基础上进行弥补, 前者相当于推到重来工作量巨大, 对已有的网络软硬件资源也是极大的浪费. 实际情况采用的后者, 研究人员在原有TCP/IP 协议的框架下增加了安全措施，这些措施涉及到了TCP/IP 协议簇的每一层，但其中更多的则是涉及网络层和传输层。

密码学是研究加/解密算法和密码的学科, 信息的安全性问题自然离不开密码学.

对称秘码体制、非对称秘码体制、报文摘要算法、消息认证码、数字签名和伪随机数生成器是密码学的六大工具, 下面我们分别了解其中的一些.

1．对称密码体制

设想用户A把信息M发送给用户B, 当事方A和B不希望M传递的过程中被第三方看到, 可以采取的方法是, A先对M进行处理, 然后发送处理后的结果. 对M处理实际上是进行某种数学运算, 运算结果我们用$E(M)$表示, B收到$E(M)$后, 会按照事前与A 约定好的方法从$E(M)$中把M还原出来, 结果我们用$D(E(M)) = M$表示. 第三方即使截获到$E(M)$, 因为不知道还原方法, 从而无法看到M, A与B也就达到了秘密通信的目的.

前面过程中的E我们称之为加密算法, D称之为解密算法, 二元组(E, D)表示一个特定的加/解密方法, 这个二元组是A 与B提前约定好的, 当然二元组(E, D)只有A和B知道, 其它的第三方是不能知道的, 这个不难理解.

当A与另一个网络用户C安全通信时, 就不能再用二元组(E, D)了, 需要改换成另一个加/解密二元组(E', D'). 由此看来, A 与多个用户安全通信, 就需要有多个加/解密二元组.

研究事实证明, 开发多个加解密二元组不是一件容易的事. 后来人们发现, 可以在加/解密二元组(E, D)基础上增加一个参数, A与B之外的其他用户通信只要选择不同的参数值就可以了. 具体来说, A与B之间的通信选择参数s_1, A与C之间的通信选择s_2, 即使大家都采用二元组(E, D), 也能做到安全通信, 这就避免了为不同用户对开发不同的二元组了, 这里的参数s_1和s_2就是所谓的密码或者密钥, s_1由A和B秘密保管, s_2由A和C秘密保管.

所有用户采用同一个加解密二元组(E, D), 只是不同用户对采用不同密钥的加密解密体制我们称之为对称加解密体制. “对称”这个称号的由来, 就是因为发送方与接收方共用(共享)同一个密钥.

■对称密码体制可以解决信息的机密性问题, 下面是应用模型.

已知二元组(E, D), 用户A与B之间的共享密钥是p, A把信息M安全发送给B, 流程是:

1. A和B拥有共享密钥p;
2. A利用p和加密算法E对信息M处理, 结果记作$E_p(M)$;
3. A把$E_p(M)$发送给B;
4. B收到$E_p(M)$;
5. B对收到的$E_p(M)$用密钥p和解密算法D处理, 结果记作$D_p(E_p(M)) = M$.

人们已经成功地开发出不少对称加解密算法, DES/3DES/AES就是其中的佼佼者, 这些加解密算法不但安全效率高, 且加解密速度快.

2．非对称密码体制

设想某网络有500个用户A_1, A_2, $\cdots$, A_{500}, 任何用户都可能与其他499个用户秘密通信, 每个用户需要保存$(500-1)=499$个密钥, 整个网络环境需要的密钥总数是

$$\frac{500\cdot(500-1)}{2}=124750$$

若用户数不是500而是5000, 那每个用户保存的密钥以及总密钥数将会更大.

对称密钥体制不便于密钥管理是大缺点。

上世纪70年代，人们发明了另外一种称为非对称密码的体制，它的出现使密码学的研究进入了一个崭新领域，原来许许多多棘手的问题由此迎刃而解。成为两种密码体制分水岭的上世纪70年代，在密码学的研究中具有划时代的意义，已载入史册。

非对称密码体制也涉及加密算法、解密算法与密钥. 每个用户有两个密钥, 一个是需要公开的密钥, 另外一个必须私藏的密钥. 因为加密密钥和解密密钥不同, 这是“非对称”名称的由来; 又因为每个用户的公钥需要公开, 所以非对称密码体制也称公钥密码体制.

假设某个具体的公钥密码算法加密算法为E, 解密算法为D, 对每个希望参与安全通信的用户, 先依据算法为其产生公钥和私钥. 假设为用户A产生公钥为p_a, 私钥s_a, 为用户B产生公钥p_b, 私钥为s_b. A的密钥对(p_a,s_a) 由A 唯一确定, B的密钥对(p_b,s_b) 由B唯一确定, 用户A和B都把自己的公钥公开, 私钥则秘密保管.

■ 非对称密码体制也可以解决信息的机密性问题, 下面是应用模型.

已知公钥密码算法二元组(E,D), 用户A把信息M安全发给用户B, 下面是流程.

1. 利用(E,D)为用户A产生公钥p_a, 私钥s_a, 为用户B产生公钥p_b, 私钥s_b;
2. A用B的公钥p_b和加密算法E对消息M进行处理, 处理结果记为$E_{p_b}(M)$;
3. A把$E_{p_b}(M)$发给用户B;
4. B对收到的$E_{p_b}(M)$进行处理, 结果记为$D_{s_b}(E_{p_b}(M))$;

我们说, 上面流程第3步的结果$D_{s_b}(E_{p_b}(M))$实际上就是M, 即

$$D_{s_b}(E_{p_b}(M))=M$$

这是公钥密码算法的设计要求所决定的.

第2步A发给用户B的信息$E_{p_b}(M)$, 要是被用户C截获了, C用自己的私钥计算$D_{s_c}(E_{p_b}(M))$, 该结果一定不等于M; 另外, 既然C知道了B的公钥p_b, 是否可以想法导出B的私钥s_b, 从而也可以作计算$D_{s_b}(E_{p_b}(M))$呢? 这在计算上也是办不到的, 原因都是公钥密码算法的特性所决定的.

前面的知识作个简单地总结: 对任何用户, 不能从公钥导出其私钥; 用谁的公钥加密, 只能用谁的私钥解密.

前面那个有500个用户A_1, A_2, $\cdots$, A_{500}的网络, 若他们采用公钥密码算法通信, 我们不难计算出每个用户只需要保存自己的私钥, 网络环境中的密钥总数等于$2\cdot500=1000$, 这比对称体制算法中的密钥数少多了, 这是公钥密码算法的优点所在.

目前, 比较流行的的公钥密码算法有RSA/ELGAMAL/IBC等.

3. 数字签名

生活中, 为了确认某个写有文字的纸质材料确实来自某个人, 我们可以让他在纸质材料上签上自己的名字, 如借款时用的借条. 要是两个单位打交道的话, 一般采用在材料上盖公章的方法, 可证明材料来自公章拥有单位.

通过电子信息方式通信的两个用户, 有什么办法也能达到上面的效果呢?

设想网络用户A把信息u发送给B, B为了确认收到的信息确实来自A, 可以让A对信息u进行特殊处理(相当于生活中的签字或者盖章), 这种处理方法应该是A特有的、独一无二的、其他用户无法效仿的等等, 在给出具体处理方法之前, 我们先给这种处理方法起个名字, 把它叫做“数字签名”. 经A数字签名后的电子材料, 也应该与生活中A手写的材料一样, 材料是完整的/不可仿冒的/不可否认的等等.

之前, 我们学习过公钥密码体制, 对参与通信的每个用户都有公钥和私钥, 这对数被该用户唯一确定. 假设公钥密码加解密算法是E, 解密算法是D, A的公钥是p, 私钥是s. 按照公钥密码算法原理, 对任意消息u, 都有

$$D_s(E_p(u))=u$$

和

$$E_p(D_s(u)) = u$$

前一个公式在解决数据的机密性问题方面有具体应用.

按照公钥密码算法原则, 算法E/D/任何用户公钥都是公开的, 每个用户的私钥由用户本人秘密保管. 我们分析公式$E_p(D_s(u)) = u$, 若网络用户A把信息u发给用户B, $D_s(u)$可以看作A用户利用自己的私钥对消息u进行处理, $E_p(D_s(u)) = u$可以看作用户B把消息u进行了还原. 用户A对消息进行$D_s(u)$的的处理是否可以看成对消息u的数字签名呢? 签名必须保证信息的完整性和不可否认性(注意机密性不需要), 完整性是指A发出的信息没有被篡改或者说篡改后能被发现, 不可否认性是指A发出了签名消息, 不能否认.

4. 数字证书

生活中有各式各样的“证书”, 如“驾照”/“工作证”/“计算机等级证”/“外语等级证”/“身份证”等等, 这些证书或者证明我们具有某种能力或者表明我们的身份, 它们一般由某个权威组织/单位颁发.

“数字证书”与上面的证书有些类似, 也是由某个权威组织颁发给用户的, 其目的就是证明某个公钥确实属于这个用户, 有时数字证书也称称公钥证书.

一般情况下, 数字证书包含有用户的名字/单位/邮箱/公钥等个人信息, 还有颁发机构的电子签名(正像平时证书的签章), 证书的颁发机构也称CA(Certification Authority).

从技术上讲，证书其实包含三部分，用户的信息，用户的公钥，还有CA中心对该证书里面的信息的签名。验证一份证书的真伪（即验证CA中心对该证书信息的签名是否有效），需要用CA 中心的公钥验证，而CA中心的公钥存在于对这份证书进行签名的证书内，故需要下载该证书，但使用该证书验证又需先验证该证书本身的真伪，故又要用签发该证书的证书来验证，这样一来就构成一条证书链的关系，这条证书链在哪里终结呢？答案就是根证书，根证书是一份特殊的证书，它的签发者是它本身，下载根证书就表明您对该根证书以下所签发的证书都表示信任，而技术上则是建立起一个验证证书信息的链条，证书的验证追溯至根证书即为结束。所以说用户在使用自己的数字证书之前必须先下载根证书。

根证书是CA认证中心给自己颁发的证书,是信任链的起始点。安装根证书意味着对这个CA认证中心的信任。

■ 知识点四、流量控制、IP地址、私有地址、公有地址、域名解析

流量控制：计算机A把信息发送给计算机B，因为各自不同的处理速度、存储能力包括线路带宽等因素，会造成发送和接收的不平衡，有时计算机B可能来不及接收处理A发来的数据时，只能丢弃。这样的话，A发送数据速度再快也没有意义，流量控制就是专门为这个现象开发的一项技术。大体意思是：通信时，接收端把自己的接收能力告诉发送端，发送端发送的数据量不能超出这个能力。

IP地址：互联网上成千上万台主机，都需要唯一标识符与其它主机区分，协议规定IP地址是计算机的标识符之一，每个参与INTERNET通信的计算机都需有唯一的IP地址。

IP是一个32位的二进制数数，其总数等于2^{32}。规定最左边的1位等于0时，叫做A类地址；最左边的两位等于10时叫B 类地址；最右边的三位等于110时叫C 类地址。

所有A类地址的总数等于2^{31}，把这些地址按照以下格式分成2^7块，

$$\overbrace{0\underbrace{0000000}_{24}\underbrace{**\cdots*}_{24},\quad 0\underbrace{0000001}_{24}\underbrace{**\cdots*}_{24},\quad \cdots,\quad 0\underbrace{1111111}_{24}\underbrace{**\cdots*}_{24}}^{2^7\text{ 块}}$$

每一块右边24个“*”位任意变化，每块包含的地址数等于2^{24}。

所有B类地址的总数等于2^{30}，把这些地址按照以下格式分成2^{14}块，

$$\overbrace{10000000.00000000\underbrace{**\cdots*}_{16},\quad 10000000.00000001\underbrace{**\cdots*}_{16},\quad \cdots,\quad 10111111.11111111\underbrace{**\cdots*}_{16}}^{2^{14}\text{ 块}}$$

每一块右边边16个“*”位任意变化，每块包含的地址数等于2^{16}。

所有C类地址的总数等于2^{29}，把这些地址按照以下格式分成2^{21}块，

$$\overbrace{110000000.00000000.00000000\underbrace{**\cdots*}_{8},\quad \cdots,\quad 110111111.11111111.11111111\underbrace{**\cdots*}_{8}}^{2^{21}\text{ 块}}$$

每一块右边边8个“*”位任意变化，每包含的地址数等于2^8。

上面的A/B/C类块更多情况下叫A/B/C类网络。

从上面的分法可以看出，A类网络个数最少，但每个网络内的地址数最多，C类网络个数最多，但每个网络内的地址数最少，B 类网络的情况则介于中间。

IANA（Internet Assigned Numbers Authority）负责规划和分配上述地址。

私有地址：，分配机构也考虑到一些计算机只是采用TCP/IP协议在局域网内通信，并不会与INTERNET相连，于是专门预留出一些地址供这类用户使用，称其为保留地址或者私有地址，具体有：

A类保留地址：10.0.0.0/8

B类保留地址：172.16.0.0/16，172.17.0.0/16，$\cdots$，172.31.0.0/16

C类保留地址：192.168.0.0/24，192.168.1.0/24，192.168.2.0/24，$\cdots$，192.168.255.0/24

公有地址：A/B/C三类地址中，非私有地址的叫做公有地址。

域名解析：IP地址是数，而且还是不容易记忆的数（主要因便于通信而设计）。为了方便使用计算机的用户，还有一种称为域名计算机标识符，例如大家经常看到的www.sina.com、www.nuist.edu.cn 等等，当我们与这种域名代表的计算机通信时，还得先把域名转换为数字型的IP地址。具体情况是，转化工作没有增加人的负担，是设备自动完成的，符号名式的名称到IP 地址式名称的转化过程就叫**域名解析**。

需要说明的是，预留A/B/C三类保留地址的初衷是提供给采用TCP/IP协议通信，但不与INTERNET 相连的计算机使用。但现实情况是，有些计算机已经采用或者可以采用这些保留地址参与INTERNET通信，这又如何解释？原来这背后有一个称为NAT 的技术在起作用，NAT又叫网络地址转换，保留地址通过NAT转换成共有地址就

能参与互联网通信了，这个非常实用的NAT 技术有效地解决了公有地址目前已耗尽，但又有很多用户希望加入互联网通信的矛盾。

■ 知识点五、路由、子网划分

路由：IP 地址为IP_1 的主机H_1向地址是IP_2 的主机H_2发送数据时，IP_1 和IP_2 作为源和目标被写在主机H_1网络层形成分组首部，该分组被互联网上有关的路由器转发，最终被送到H_2 的网络层。

路由器收到分组，依据分组首部的目标IP地址计算其所在的网络号，根据网络号选择转发接口。

举个例子:

当目标IP地址为98.1.1.1时，路由器根据$0 \leqslant 98 \leqslant 127$, 计算出98.1.1.1所在的网络号为98.0.0.0，然后依据网络号根据路由表选择转发接口;

当目标IP地址为156.1.1.1时，路由器根据$128 \leqslant 156 \leqslant 191$, 计算出156.1.1.1所在的网络号为156.1.0.0，然后依据网络号根据路由表选择转发接口;

当目标IP地址为200.1.1.1时，路由器根据$192 \leqslant 200 \leqslant 223$, 计算出200.1.1.1所在的网络号为200.1.1.0，然后依据网络号根据路由表选择转发接口;

路由表记录的是目标网络与路由器接口的对应关系，而不是目标IP地址与接口的对应关系，因为如果是后者的话，路由器需要记录的路由表就太大了。举个通俗易懂的例子，这有点像人们开车从“无锡”走京沪高速去“北京故宫”时，在进入京沪高速的入口处，指示牌上标注的目标地是北京而不是北京故宫的道理一样。

大家知道，两个1位二进制数可作一种称为与的运算“$\wedge$”，运算的规则如下

$$0 \wedge 0 = 0, \quad 0 \wedge 1 = 0, \quad 1 \wedge 0 = 0, \quad 1 \wedge 1 = 1$$

我们让32位的IP地址98.1.1.1与32位的数据255.0.0.0的对应为作运算$\wedge$，容易得出

$$98.1.1.1 \wedge 255.0.0.0 = 98.0.0.0$$

这种运算也可以得出98.1.1.1所在的网络号。事实上，路由器用这种方法计算目标IP地址位于的网络号时，其速度要明显快于作比较的速度，把255.0.0.0这个数据称为A类网络地址的掩码，并将其放在路由器中的路由表中每个A类网络号的后面。同理B和C类网络地址的掩码分别是255.255.0.0和255.255.255.0，分别放在路由器路由表B类网络号和C类网络号的后面。

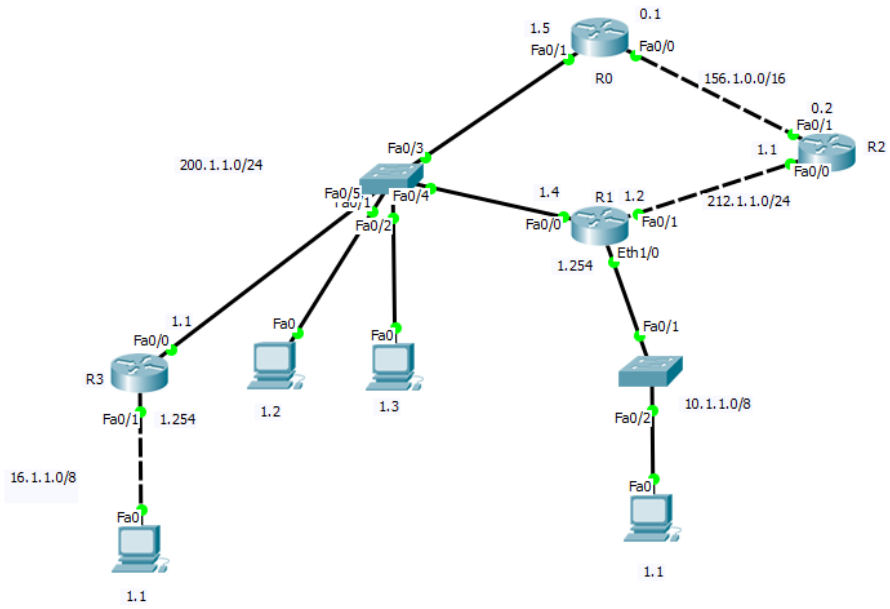


Figure 4: 路由表格式说明图

如图4表示的网络环境，有5个网络，分别是

16.0.0.0/8
200.1.1.0/24
156.1.1.0/16
212.1.1.0/24
10.0.0.0/8

查看R3的路由表，如下

O 10.0.0.0/8 [110/11] via 200.1.1.4, 00:20:57, FastEthernet0/0

C 16.0.0.0/8 is directly connected, FastEthernet0/1

O 156.1.0.0/16 [110/2] via 200.1.1.5, 00:20:57, FastEthernet0/0

C 200.1.1.0/24 is directly connected, FastEthernet0/0

O 212.1.1.0/24 [110/2] via 200.1.1.4, 00:20:57, FastEthernet0

每一行就是路由表的一个条目，网络号斜杠后面的数字表示该网络的掩码，如网络10.0.0.0的掩码为255.0.0.0，从地址为16.1.1.1的计算机发往地址为10.1.1.1计算机的包，首先送到路由器R3，R3取出包的首部中的目标IP地址10.1.1.1，从上到下分别与路由表中每个网络斜杠后面的掩码作与运算，运算结果与该行的网络号作比较，相等时把数据包发给“下一跳”，不相等则到下一行继续作相同的运算。

子网划分： IP地址分成了A/B/C三类，设想某企业申请到了C类地址段200.100.1.0/24，企业局域网络有200台计算机需要通信，因为协议规定局域网内的每个计算机都要占用唯一的地址，我们可以从200.100.1.0 ~ 200.100.1.255中选取200个地址分配给计算机。余下的56个地址可以作为局域网扩展时使用。不过需要特别说明的是余下的56 个地址实际可以使用的只有56-2=54个。因为当IP地址指派给某局域网内的计算机时，按照协议规定，编号最最小的200.100.1.0 代表网络号，编号最大的200.100.1.255代表网络的广播地址，这两个地址是不能给计算机使用的，所以整个局域网内可以分配的地址等于256-2=254。

A/B/C三类网络，最少的C类网，也有256个地址，IANA最初以A/B/C三类网络为单位进行的地址分配方式太不利于地址的节约使用了。

后来出现了一种称为子网划分的技术，能够比较好的节约地址使用。下面以C类网络举例说明子网划分的原理，这个原理也适合A 类网络和B类网络。

200.100.1.0/24是C类地址网络，斜杠后面的数字表示地址块的左边24位是网络位，余下的32-24=8位表示主机位，8位的主机位可以表示的地址范围是200.100.1.0 ~ 200.100.1.255，这是没有子网划分的情况。

我们把这256个地址200.100.1.0 ~ 200.100.1.255分成

$$\begin{cases} 200.100.1.\mathbf{0}0000000 \\ 200.100.1.\mathbf{0}0000001 \\ \vdots \\ 200.100.1.\mathbf{0}1111111 \end{cases} \quad \text{和} \quad \begin{cases} 200.100.1.\mathbf{1}0000000 \\ 200.100.1.\mathbf{1}0000001 \\ \vdots \\ 200.100.1.\mathbf{1}1111111 \end{cases}$$

两部分（注：为了方便描述，IP地址最后数字用二进制表示），前一部分最后的8位二进制数的最左位等于0，后一部分最后8 位二进制数的最左位等于1. 前一部分用200.100.1.0/25表示，后一部分用200.100.1.128/25 表示，这里的25表示网络位，主机位就是32-5=7位，每部分有$2^7 = 128$个地址，这就叫做对C类网络200.100.1.0/24进行二分子网划分。

C类网络200.100.1.0/24二分子网划分后产生2个子网，每个网络依然有网络号和广播地址。

第1个子网的200.100.1.0/25 网络号和广播地址分别是为200.100.1.0 和200.100.1.127；

第2个子网200.100.1.128/25 网络号和广播地址分别是为200.100.1.128 和200.100.1.255。

网络号总是地址中的最小者，广播地址总是地址中的最大者。

按照同样的思路，我们可以把200.100.1.0/24分成四部分，它们是

$$\left\{\begin{array}{c}200.100.1.\mathbf{00}000000\\200.100.1.\mathbf{00}000001\\\vdots\\200.100.1.\mathbf{00}111111\end{array}\right.,\left\{\begin{array}{c}200.100.1.\mathbf{01}000000\\200.100.1.\mathbf{01}000001\\\vdots\\200.100.1.\mathbf{01}111111\end{array}\right.,\left\{\begin{array}{c}200.100.1.\mathbf{10}000000\\200.100.1.\mathbf{10}000001\\\vdots\\200.100.1.\mathbf{10}111111\end{array}\right.,\left\{\begin{array}{c}200.100.1.\mathbf{11}000000\\200.100.1.\mathbf{11}000001\\\vdots\\200.100.1.\mathbf{11}111111\end{array}\right.$$

请注意这四部分各有$\frac{256}{4}=64$个地址，每一部分最后的8位二进制数的左边两位都相同。这个四个部分表示为

$$\begin{array}{c}200.100.1.0/26\\200.100.1.64/26\\200.100.1.128/26\\200.100.1.192/26\end{array}$$

这就叫做对C类段200.100.1.0/24进行四分子网划分。四部分的掩码都是255.255.255.192。

问题1：写出对C类段202.190.1.0/24的二分、四分、八分、十六分、三十二分、六十四分子网划分方式。

C类段地址段经过子网划分产生了多个网络，具体到每个子网络内的地址数明显减少了，但“掐头去尾”这个特性还是不能变化, 也就是说，子网内的地址数即使再少，头尾两个地址也不能给计算机使用，因为它们分别代表子网号和子网广播地址。

问题2：某个计算机的IP地址为212.3.56.7/26，试求该计算机位于的子网号？这个子网是哪个地址段几分子网划分的结果？子网的广播地址是多少？该子网实际能分配给设备接口的地址个数是多少？

问题3：某个计算机的IP地址为193.4.3.56/28，试求该计算机位于的子网号？这个子网是哪个地址段几分子网划分的结果？子网的广播地址是多少？该子网实际能分配给设备接口的地址个数是多少？

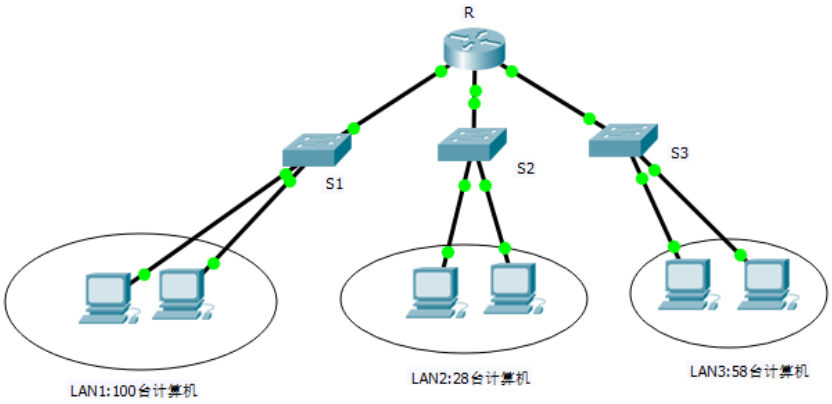


Figure 5: 网络拓扑图

问题4：某企业网络结构如图4所示，申请到C类地址段201.3.4.0/24，三个局域网内的计算机数量如图示，给出地址段201.3.4.0/24一种具体的子网划分方式，能够满足这三个局域网内计算机地址的需求，给出每个局域网地址范围和网络掩码？给出每个局域网内的子网号和广播地址？

问题5：某单位有6个局域网采用路由设备将其相连，6个局域网内部的计算机数量分别为20,23,30,27,16,19。若单位申请到C类地址段200.3.16.0/24，给出一种地址分配方式？写出每个局域的掩码？

问题6: 互联网上的两台主机通信, 总是先动态地选择一条通路, 数据报沿着当前通路进行传送. 如图所示的通路, 主机PC1发出的信息先后经过两个中间结点(通常是路由器)到达另一个主机PC2.

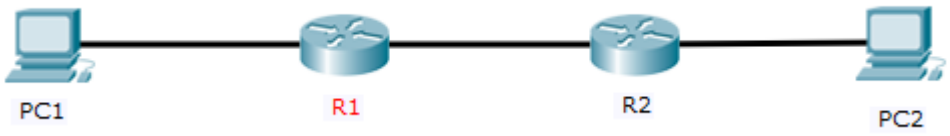


Figure 6: 问题6示意图

因为各种原因, 通路中的中间结点设备路由器收到一个分组后, 可能将其丢弃(例如收到的数据报被CRC检查出错而丢弃). 按照这种说法, PC1与PC2之间有3段链路, PC1发往PC2的数据报通过三段链路时, 数据报可能被R1或者R2丢弃. 当然, 传输协议的可靠机制会将丢弃的分组重传, 要保证丢弃的数据报最终被PC2收到.

假设图中的R1和R2丢弃分组的概率为p, 试计算:

1. PC1发出的每一个分组在一次传输过程中平均经过几段链路?

2. PC1发出的每一个分组平均要传送几次才成功?

分析解答:

1. PC1发出的每个分组可能走1段链路、2段链路和3段链路. 走1段链路的概率是p，走2段链路的概率是$p(1-p)$，走3 段链路概率是$(1-p)^2$. 那么, PC1发出的每个分组所走链路数的平均数l是一个数学期望值, 它可以表示为

$$\begin{aligned} l =& 1 \cdot p + 2 \cdot p(1-p) + 3 \cdot (1-p)^2 \\ =& p^2 - 3p + 3 \end{aligned}$$

即PC1发往PC2的每个分组, 所走链路的平均数为$p^2 - 3p + 3$(这个数值在某些特殊情况下可以理解, 例如当$p = 0$时, 有$l = 3$, 其含义是R1和R2都不丢弃分组, 分组自然走3段路; 当$p = 1$时, 有$l = 1$, 其含义是R1和R2收到的分组一定丢弃, 分组自然走1段路)

2. 每一次发送分组, 分组要是走三段链路, 就说明发送成功. 走三段链路的概率为$(1-p)^2$, 也就是一次发送就成功的概率为$(1-p)^2$, 令$\alpha = (1-p)^2$. 两次发送成功的概率为$(1-\alpha)\alpha$, 三次发送成功的概率为$(1-\alpha)^2\alpha$, 四次发送成功的概率为$(1-\alpha)^3\alpha$, $\cdots$. 因此, 一个分组发送成功, 平均发送的次数就是

$$\begin{aligned} T =& 1 \cdot \alpha + 2 \cdot \alpha(1-\alpha) + 3 \cdot \alpha(1-\alpha)^2 + \cdots + n \cdot \alpha(1-\alpha)^{n-1} + \cdots \\ =& \frac{\alpha}{1-\alpha}[(1-\alpha) + 2(1-\alpha)^2 + 3(1-\alpha)^3 + \cdots + n(1-\alpha)^n + \cdots] \\ =& \frac{\alpha}{1-\alpha} \sum_{k=1}^{\infty} k(1-\alpha)^k \\ =& \frac{\alpha}{1-\alpha} \cdot \frac{1-\alpha}{(1-(1-\alpha))^2} \quad \left(\text{这一步利用了一个公式} \sum_{k=1}^{\infty} kq^k = \frac{q}{(1-q)^2} \right) \\ =& \frac{1}{\alpha} \\ =& \frac{1}{(1-p)^2} \end{aligned}$$