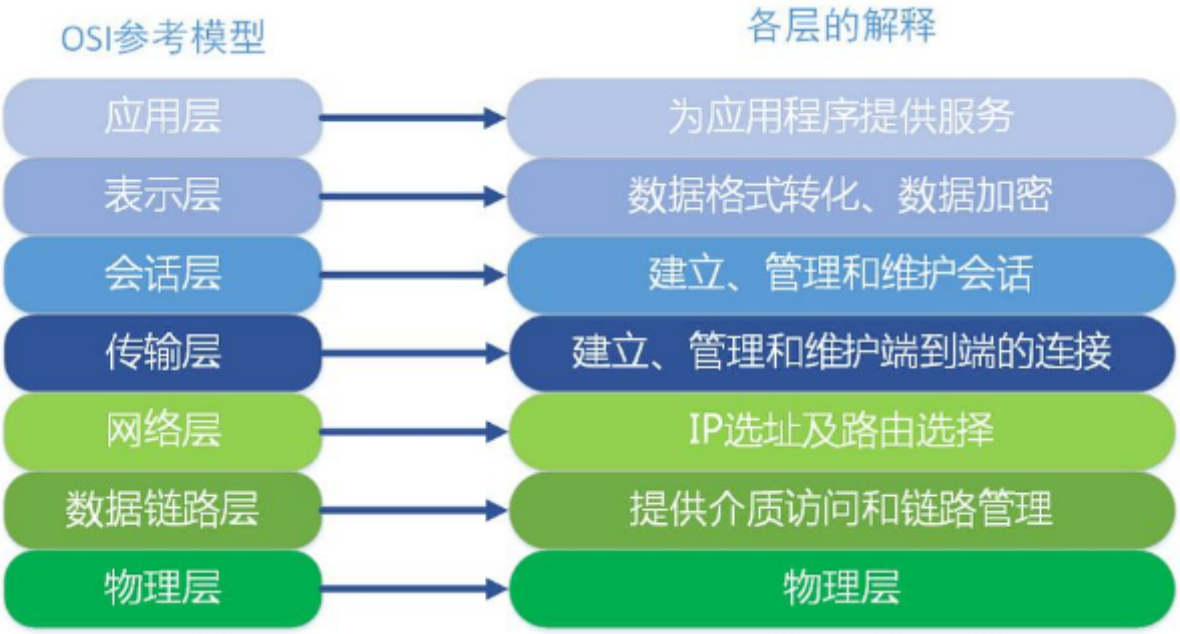
第二章 计算机网络体系结构

2.1 常用的计算机网络体系结构

2.1.1 OSI体系结构

1、为了使不同体系结构的计算机网络都能够互联，国际标准化组织于1977年成立了专门机构研究该问题，不久他们就提出了一个试图使各种计算机在世界范围内都能够互连成网的标准框架，也就是著名的“**开放系统互连参考模型**”，**简称为OSI，OSI体系结构有时候我们也称之为OSI模型。**

2、OSI是一个**七层协议**的体系结构：从下往上依次是**物理层、数据链路层、网络层、运输层、会话层、表示层、应用层。**



j

3、OSI试图达到一种理想境界，即全球计算机网络都遵循这个统一标准，因而全球的计算机将能够很方便地进行互连和交换数据。在20世纪80年代，许多大公司甚至一些国家的政府机构纷纷表示支持OSI。当时看来似乎在不久的将来全世界一定会按照OSI制定的标准来构造自己的计算机网络。

4、然而到了20世纪90年代初期，虽然整套的OSI国际标准都已经制定出来了，但由于基于TCP/IP的互联网已抢先在全球相当大的范围成功地运行了，而与此同时却几乎找不到有什么厂家生产出符合OSI标准的商用产品。因此人们得出这样的结论：**OSI 只获得了一些理论研究的成果，但在市场化方面则事与愿违地失败了。**

现今规模最大的、覆盖全球的、基于TCP/IP的互联网并未使用OSI标准。

5、OSI失败的原因可归纳为:

- **OSI的专家们缺乏实际经验，他们在完成OSI标准时缺乏商业驱动力;**
- **OSI的协议实现起来过分复杂，而且运行效率很低;**

- **OSI标准的制定周期太长，因而使得按OSI标准生产的设备无法及时进入市场;**
- **OSI的层次划分不太合理，有些功能在多个层次中重复出现。**

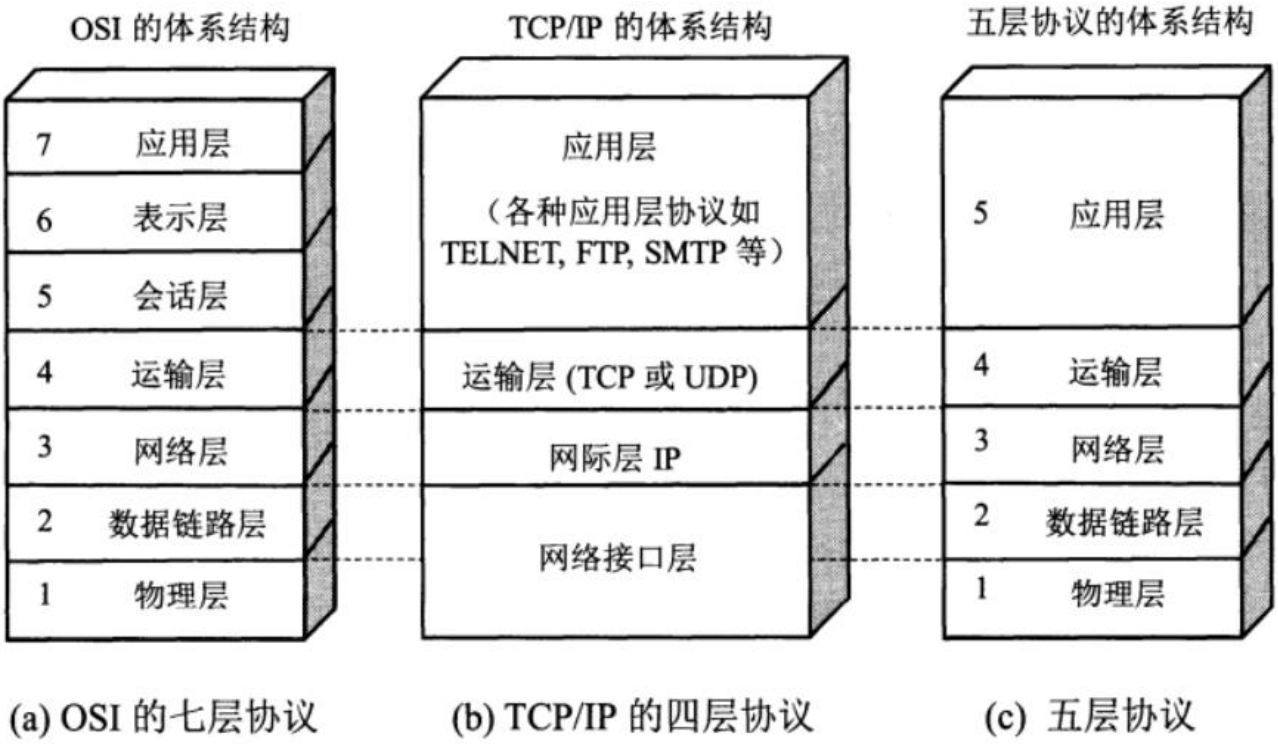
6、**OSI体系结构是法律上的国际标准， TCP/IP体系结构是事实上的国际标准**

2.1.2 具有五层协议的体系结构

1、TCP/IP是一个**四层**的体系结构，它包含**应用层、运输层、网际层和网络接口层**（用网际层这个名字是强调这一层是为了解决不同网络的互连问题)。

2、OSI的七层协议体系结构概念清楚，理论也比较完整，但是太过于复杂不实用。TCP/IP体系结构不同，但是现在却得到了非常广泛的应用。

3、在学习计算机网络的原理时往往采取折中的办法，即综合OSI和TCP/IP 的优点，**采用一种只有五层协议的体系结构**，这样既简洁又能将概念阐述清楚。有时为了方便，也可把最底下两层称为**网络接口层**。

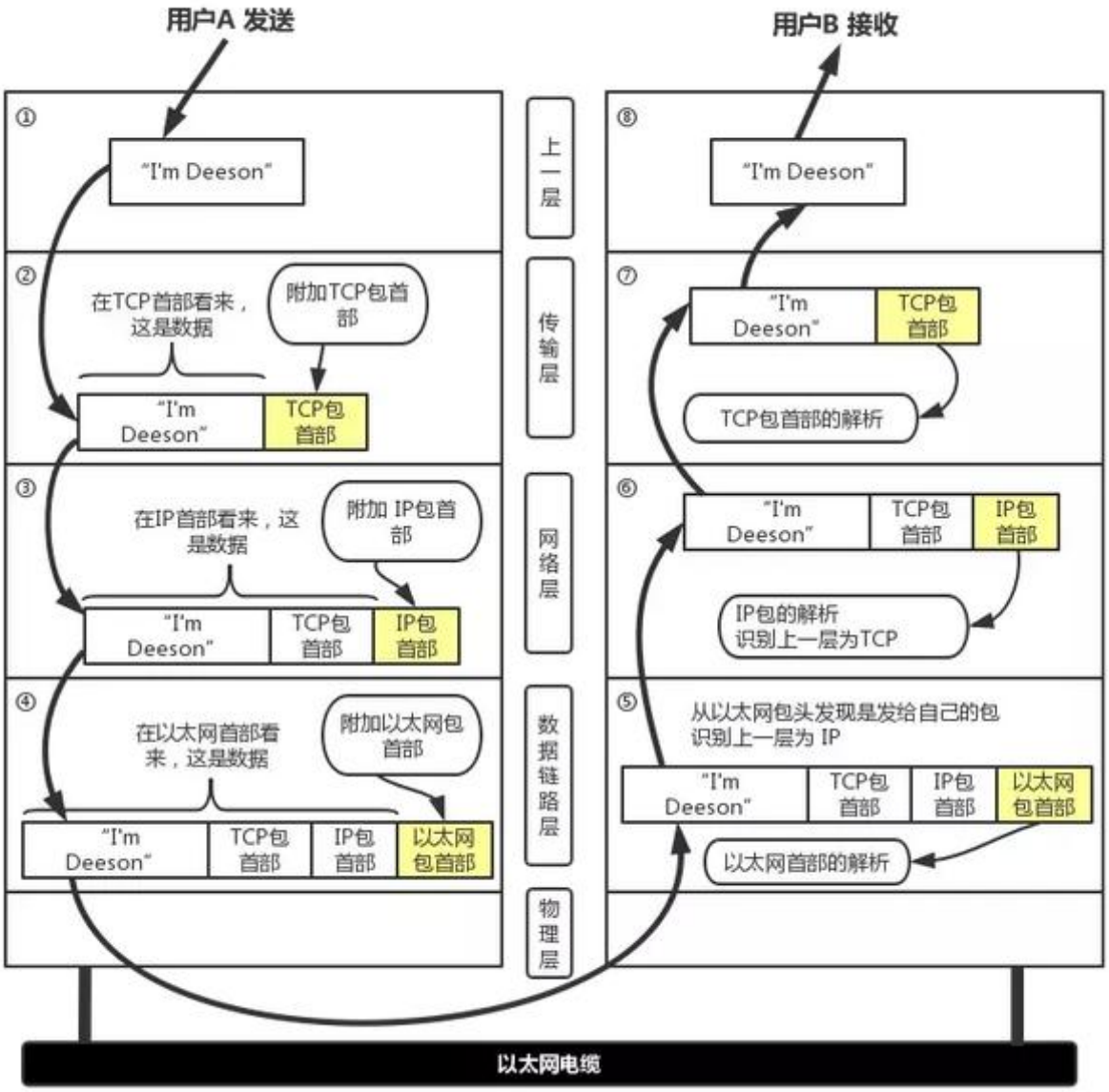


4、下面我们结合互联网的情况，自上而下地，非常简要的介绍一下各层的主要功能。

- **应用层（application layer)**
  - 应用层是体系结构中的最高层。应用层的任务是通过应用进程间的交互来完成特定网络应用。应用层协议定义的是应用进程间通信和交互的规则。这里的进程就是指主机中正在运行的程序。对于**不同的网络应用需要有不同的应用层协议**。在互联网中的应用层协议很多，如**域名系统DNS，支持万维网应用的 HTTP 协议，支持电子邮件的SMTP协议**，等等。**我们把应用层交互的数据单元称为报文(message)。**
- **运输层（transport layer)**
  - 运输层的任务就是负责**向两台主机中进程之间的通信提供通用的数据传输服务。**

- 运输层主要使用以下两种协议:
  - **传输控制协议TCP (Transmission Control Protocol)**：提供面向连接的、可靠的数据传输服务
  - **用户数据报协议UDP (User Datagram Protocol)** ：提供无连接的、尽最大努力(best-effort)的数据传输服务（不保证数据传输的可靠性）
- TCP和UDP协议都有固定的格式，数据在经过运输层时会根据所选择的运输协议在应用层传递过来的数据基础上加上对应协议的头部。

- **网络层（network layer）**
  - 主要作用是实现**两个网络系统之间的数据透明传送**，具体包括**路由选择，拥塞控制和网际互连**等。
  - 在发送数据时，网络层把运输层产生的报文段或用户数据报封装成分组或包进行传送。在TCP/IP体系中，由于网络层使用IP协议，因此分组也叫做**IP数据报**，简称为**数据报**。
  - 数据在经过网络层时会加上IP协议的头部
- **数据链路层（data link layer）**
  - 数据链路层常简称为链路层。我们知道，两台主机之间的数据传输，总是在**一**段**一**段的链路上传送的，这就需要使用专门的链路层的协议。在两个相邻结点之间传送数据时，数据链路层将网络层交下来的IP数据报**组装成帧(framing)**，在两个相邻结点间的链路上**传送帧(frame)**。每**一**帧包括数据和必要的**控制信息（如同步信息、地址信息、差错控制等）**。
- **物理层（physical layer）**
  - 利用传输介质为数据链路层提供物理连接，实现比特流的透明传输。
  - **物理层上所传输数据的单位是比特。**



2.2 物理层

2.2.1 物理层的基本概念

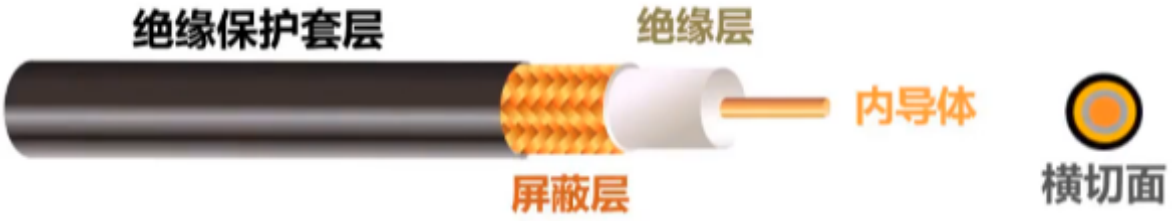
1、物理层主要解决在各种传输媒体上传输比特0和1的问题，进而给数据链路层提供透明传输比特流的服务

2、由于传输媒体的种类太多（例如**同轴电缆**、**光纤**、**无线电波**等），物理连接方式也有很多例如点**对点连接**、**广播连接**等，因此物理层协议种类也比较多。但是物理层为了解决在各种传输媒体上传输比特0和1的问题，无论是那种物理层协议都需要满足主以下四个任务：

2.2.2 传输媒体

1、导引型传输媒体：电磁波被导引沿着固体媒体传播

- **同轴电缆**



  - 可以从上图看出同轴电缆的各层都是共圆心的，也就是同轴心的
  - 同轴电缆有两种：
    - **基带同轴电缆**：数字传输，过去用于局域网
    - **宽带同轴电缆**：模拟传输，目前主要用于有线电视
  - 同轴电缆价格较贵且布线不够灵活和方便，随着集线器的出现，在局域网领域基本上都是采用双绞线作为传输媒体

- **双绞线**

  - **双绞线是最常用的传输媒体**，把两根互相绝缘的铜导线放在一起，然后按照一定规则绞合起来就构成了双绞线。
  - 常用的双绞线包含**八根铜导线**，每两根绞合成一条双绞线，绞合组合如下：

    - 蓝色线和蓝白双色线绞合
    - 橙色线和橙白双色线绞合
    - 绿色线和绿白双色线绞合
    - 棕色先和棕白双色线绞合
- 绞合的作用：
    - 抵御部分来自外界的电磁干扰
    - 减少相邻导线的电磁干扰
- 根据有无屏蔽层，双绞线分为：**无屏蔽双绞线**和**屏蔽双绞线**

无屏蔽双绞线UTP电缆

    - 无屏蔽
    - 屏蔽双绞线：在双绞线与外层绝缘封套之间有一个**金属屏蔽层**

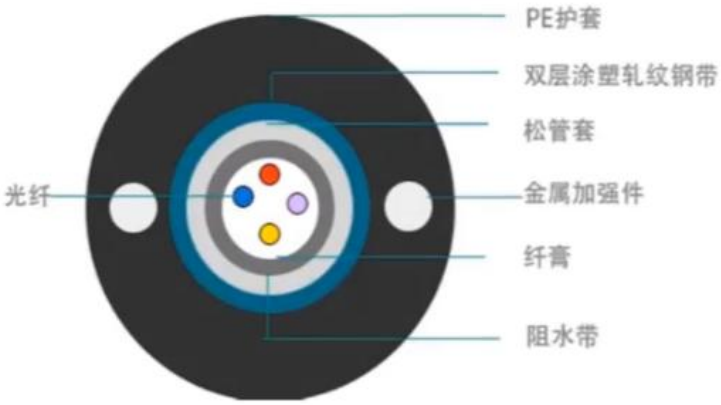
屏蔽双绞线STP电缆

- 常用的绞合线类别、带宽及典型应用

绞合线类别	带宽	线缆特点	典型应用
3	16MHz	2对4芯双绞线	模拟电话；曾用于传统以太网（10Mbit/s）
4	20MHz	4对8芯双绞线	曾用于令牌局域网
5	100MHz	与4类相比增加了绞合度	传输速率不超过100Mbit/s的应用
5E（超五类）	125MHz	与5类相比衰减更小	传输速率不超过1Gbit/s的应用
6	250MHz	与5类相比改善了串扰等性能	传输速率高于1Gbit/s的应用
7	600MHz	使用屏蔽双绞线	传输速率高于10Gbit/s的应用

- **光纤**

室外四芯光缆



光缆内部结构

    - 光纤的优点
        - 通信容量大(25000~30000GHz的带宽)
        - 传输损耗小，远距离传输时更加经济。
        - 抗雷电和电磁干扰性能好。这在大电流脉冲干扰的环境下尤为重要。
        - 无串音干扰，保密性好，不易被窃听。
        - 体积小，重量轻。

2、非导引型传输媒体：非导引型传输媒体是指自由空间

- **无线电波**
- **微波**
- **红外线**
- **可见光**

2.3 数据链路层

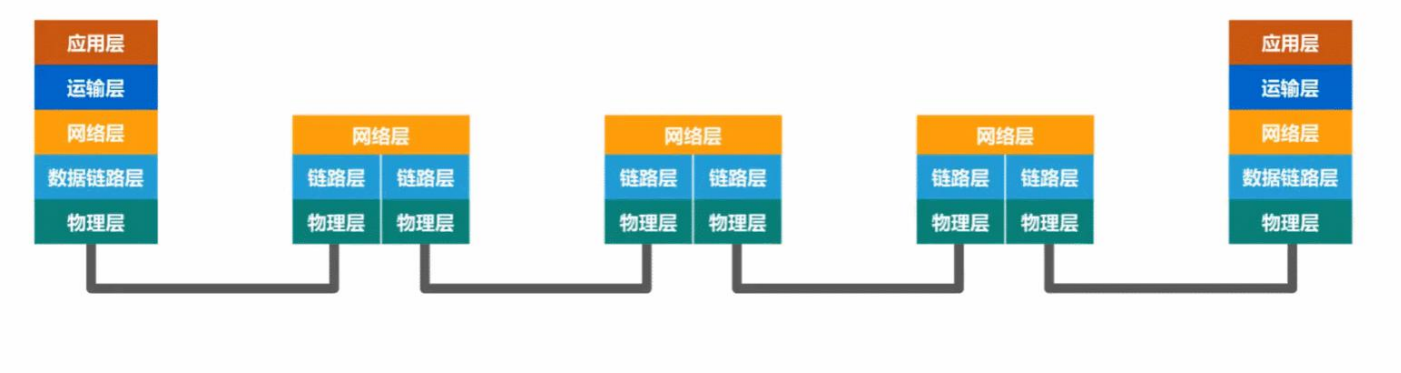
2.3.1 数据链路层概述

1、数据链路层在网络体系结构中所处的地位

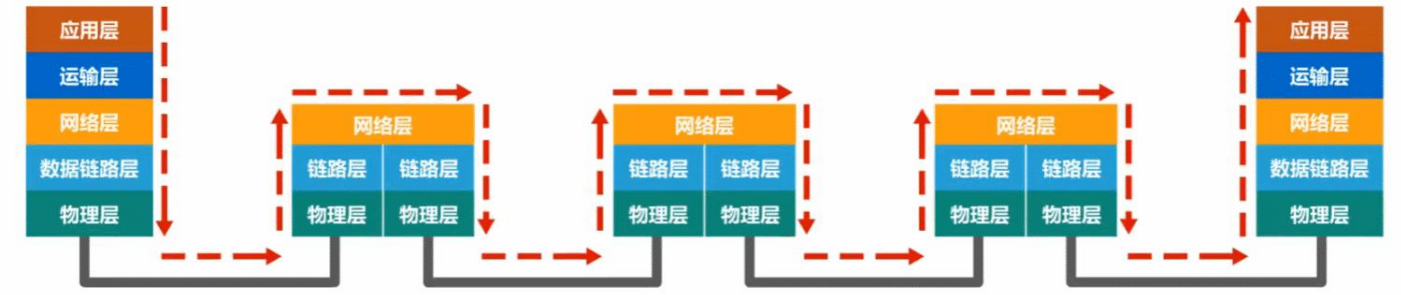
如下图所示：主机H1给主机H2发送数据，中间要经过三个路由器、电话网、局域网、广域网等多种网络。

从五层协议原理体系结构的角度来看，主机应该具有体系结构中的各个层次，而路由器只需要具有体系结构中的网络层、数据链路层、物理层。网络中的各个设备通过传输媒体进行互连，主机H1将

需要发送的数据**逐层封装**后通过物理层将构成数据包的各个比特转换为电信号发送到传输媒体，数据包进入到路由器后，**从下网上逐层解封到网络层**，路由器根据数据包的**目的网络地址**和**自身的转发表**确定数据包的转发端口，然后从网络层向下逐层封装数据包，最后通过物理层将数据包发送到传输媒体，最后到达主机H2，主机H2在接收到数据包后再逐层解封。

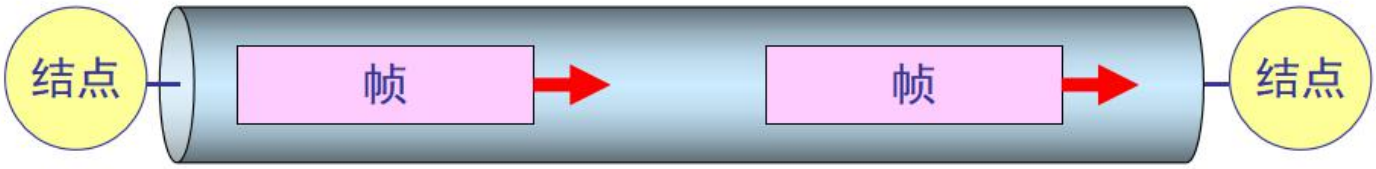


当我们研究数据链路层时，我们可以只关心数据链路层，而不考虑其他各层。我们可以想象，数据只在数据链路层从左至右沿水平方向传送。从数据链路层来看，主机H1到主机H2 的通信可以看作是在4段不同的链路上的通信所组成的。



所谓的**链路(Link)就是从一个结点到相邻结点的一段物理线路，而中间没有任何其他的交换结点。要在链路上传输数据，仅有链路还不够，还需要一些通信协议来控制这些数据的传输，如果把实现这些协议的硬件和软件加到链路上就构成了数据链路**（Data Link）

在数据链路上传输的数据包，又称为**帧。**（数据链路层是以帧作为单位传输和处理的）



> **注意：结点就是网络中的一台主机。**

数据链路层的协议有很多种，但是有三个基本问题是共同的。这三个基本问题就是：**封装成帧、透明传输和差错检测。**

2.3.2 封装成帧

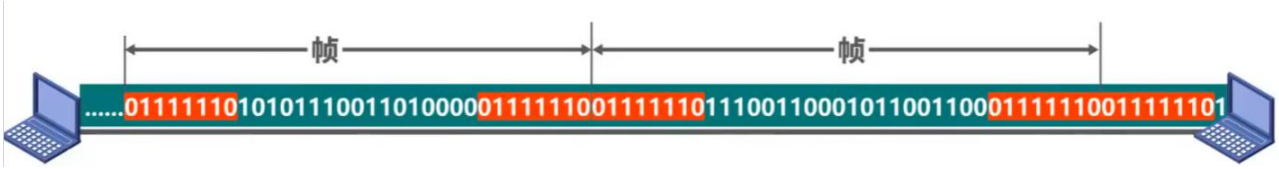
封装成帧是指数据链路层给上层交付的协议数据单元添加**帧头**和**帧尾**使之成为帧。

- **帧头和帧尾中包含有重要的控制信息**

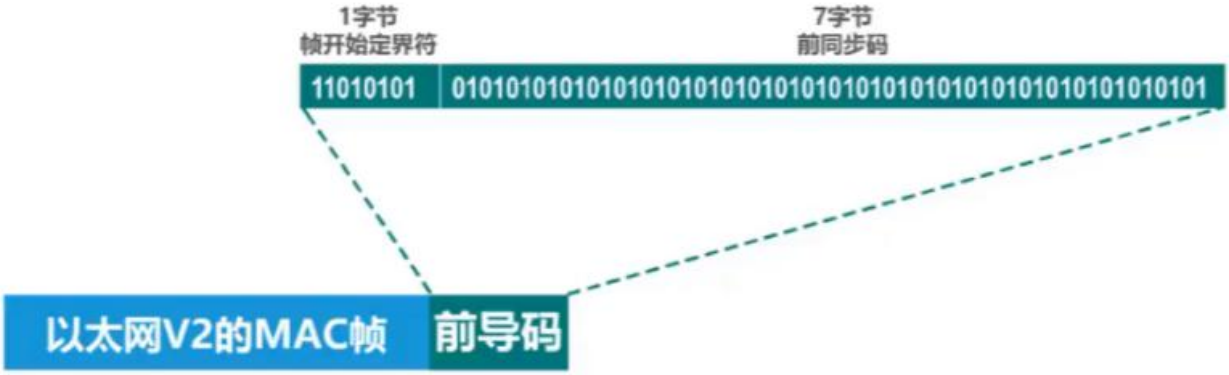
以太网V2的MAC帧（最大长度为1518字节）				
6字节	6字节	2字节	46 ~ 1500 字节	4字节
目的地址	源地址	类型	数 据 载 荷	FCS
帧头			上层交付的协议数据单元	帧尾

PPP帧的格式						
1字节	1字节	1字节	2字节	不超过1500字节	2字节	1字节
标志	地址	控制	协议	数 据 载 荷	FCS	标志
帧头				上层交付的协议数据单元		帧尾

- 帧头和帧尾的作用之一就是**帧定界**
  - 例如：PPP帧的第一个字节和最后一个字节就是帧定界，通过这两个字节就能够从物理层交付的比特流中提取出一个一个的帧。



- **并不是每种数据链路层协议的帧都包含有帧定界标志**，例如MAC帧在帧头和帧尾中是没有包含帧定界的标志的， 那么接收方是如何从物理层交付的比特流中提取出一个个的以太网帧的呢?
  - 第一步：数据链路层封装好MAC帧，将其交付给物理层
  - 第二步：物理层在MAC帧的前面添加8字节的**前导码**，前导码的**前7个字节**为**前同步码**，其作用是使接收方的时钟同步，之后的1个字节为帧开始定界符，表明其后紧跟着的就是MAC帧。

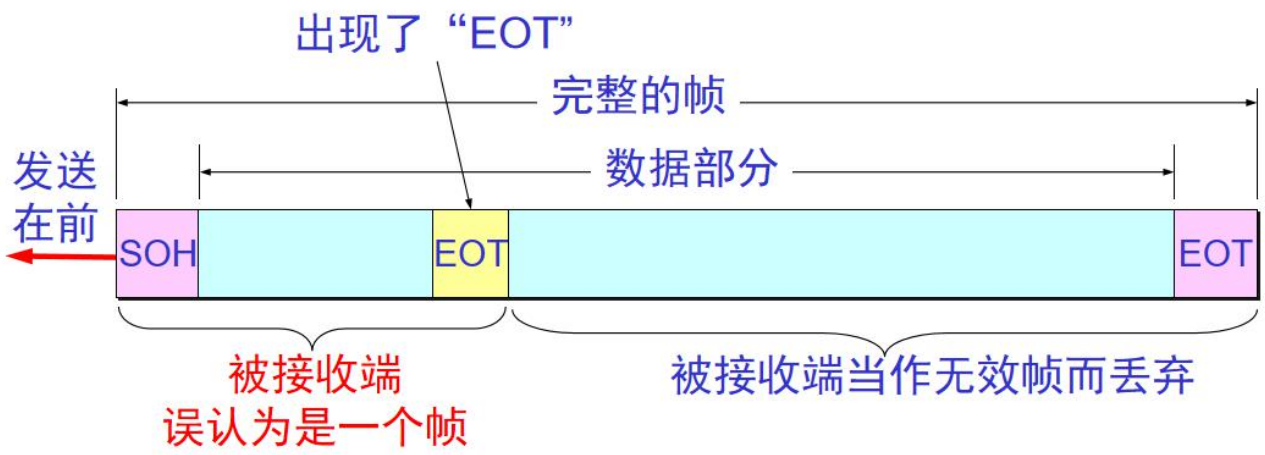


2.3.3 透明传输

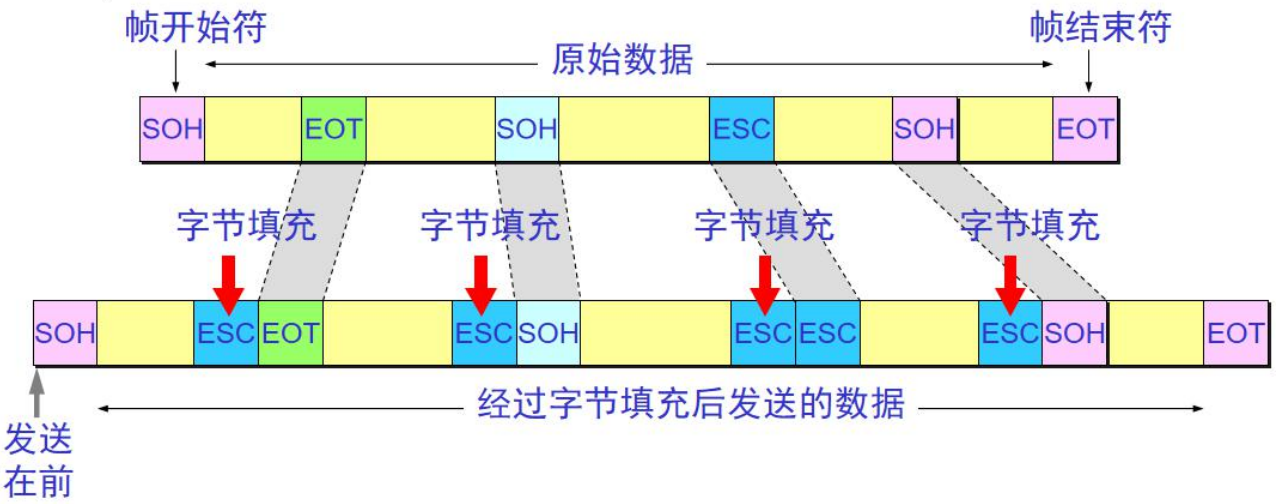
1、透明传输是指数据链路层对上层交付的传输数据没有任何限制，就好像数据链路层不存在一样。

- 当数据帧中的数据是帧定界标志时，发送端的数据链路层该如何处理呢?

- **当物理链路提供的是面向字符的传输服务时（物理链路以字符为单位传输数据）**：



数据链路层在交付数据给物理层时，对帧进行扫描，首先扫描到SOH，然后每扫描到一个SOH或者EOT就在前面加**转义字符ESC**，直至扫描到最后一个EOT，这种方式称之为：**字节填充(byte stuffing)或字符填充(character stuffing)**



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

- **当物理链路提供的是面向比特的传输服务时（物理链路以比特为单位传输数据）**：

在数据发送前采用**零比特填充法**：对数据进行扫描，每5个连续的比特1后面就插入1个比特0

考研真题：

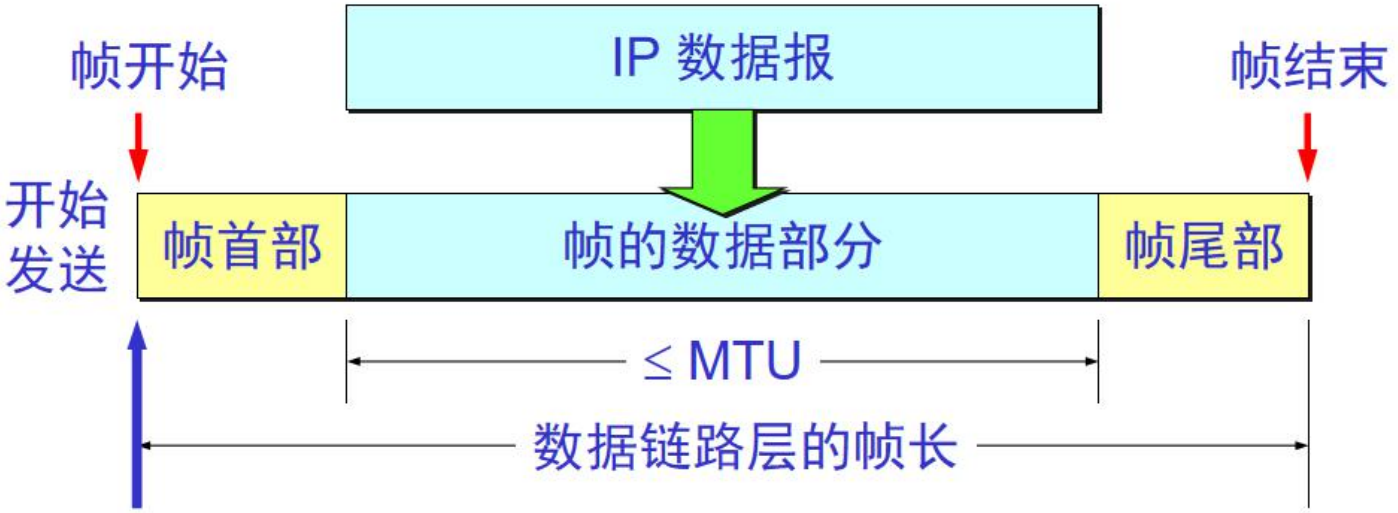
【2013年 题37】HDLC协议对0111110001111110组帧后对应的比特串为

A. 011111000011111010　　B. 01111100011111010111110

C. 01111100011111010　　D. 01111100011111001111101

2、为了提高帧的传输效率，应当使帧的数据部分的长度尽可能大些。

3、考虑到差错控制等多种因素，每一种数据链路层协议都规定了帧的数据部分的长度上限，即最大传送单元**MTU(Maximum Transfer Unit)**。



2.3.4 差错校验

1、实际的通信链路都不是理想的，比特流在传输过程中由于受到各种干扰可能会产生差错: 1可能会变成0,而0也可能变成1。这称为**比特差错**，或者称为**误码**。

2、一段时间内，传输错误的比特占所传输比特总数的比率称为**误码率（Bit Error Rate)**

3、接收方是如何知道数据在传输的过程中出现差错了呢？使用**差错检测码**来检测数据在传输过程中是否产生了比特差错，是数据链路层所要解决的重要问题之一。

4、在封装好的帧中利用若干个字节表示帧校验序列FCS字段。*FCS*:Frame Check Sequence(帧校验序列)。FCS字段由一些差错校验算法计算得出，常用的校验算法为：**循环冗余校验CRC(Cyclic Redundancy Check)。**

5、接收方在接收到数据后计算出一个FCS，然后将计算得出的FCS与接收到的数据帧中的FCS进行比较。

以太网V2的MAC帧（最大长度为1518字节）				
6字节	6字节	2字节	46 ~ 1500 字节	4字节
目的地址	源地址	类型	数 据 载 荷	FCS

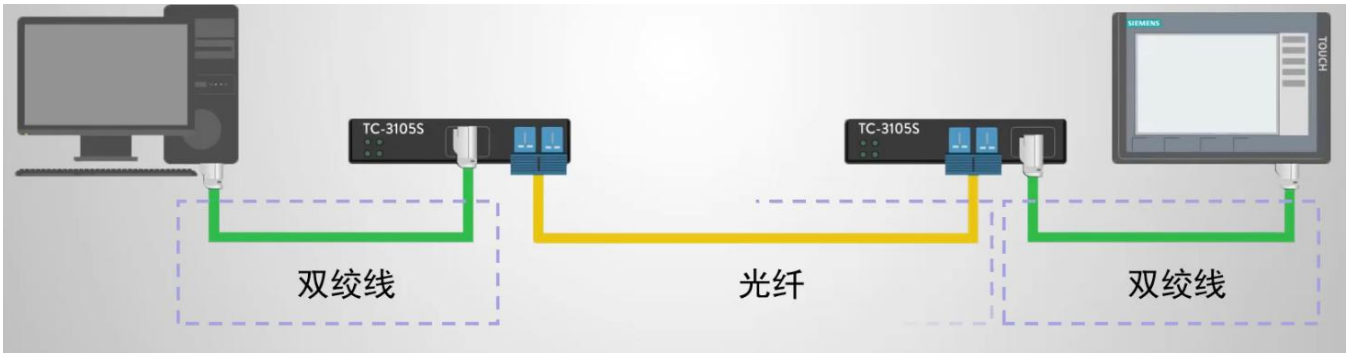
PPP帧的格式						
1字节	1字节	1字节	2字节	不超过1500字节	2字节	1字节
标志	地址	控制	协议	数 据 载 荷	FCS	标志

由于判断FCS是否正确只能检测出帧在传输过程中出现了差错，但并不能定位错误，因此**无法纠正错误。**接收方可以通过**检错重传**方式来纠正传输中的差错,或者仅仅是**丢弃检测到差错的帧**，这取决于数据链路层向其上层提供的是**可靠传输服务**还是**不可靠传输服务。**

2.3.4 以太网

1、**以太网概念**

- **以太网是一种计算机局域网技术。IEEE**（电气与电子工程师协会：Institute of Electrical and Electronics Engineers）组织的**IEEE 802.3**标准制定了以太网的技术标准，它规定了包括**物理层的连线、电子信号和介质访问层协议**的内容。**以太网是应用最普遍的局域网技术**，取代了其他局域网技术如令牌环、FDDI和ARCNET。
- 以太网是一种有线系统，最初使用**同轴电缆**进行数据传输，后来发展到使用双绞线和光纤并延续至今。



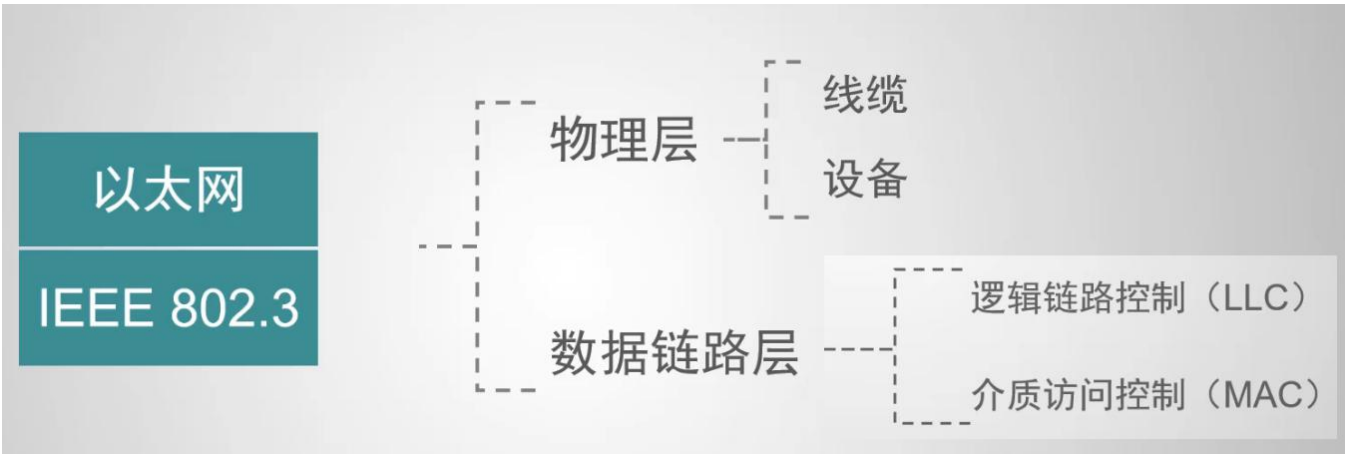
2、**以太网的两个标准**

- 1980年9月，**DEC公司**、**英特尔(Intel)公司**和**施乐公司**联合提出了10 Mbit/s 以太网规约的第一个版本**DIX V1**(DIX是这三个公司名称的缩写)。

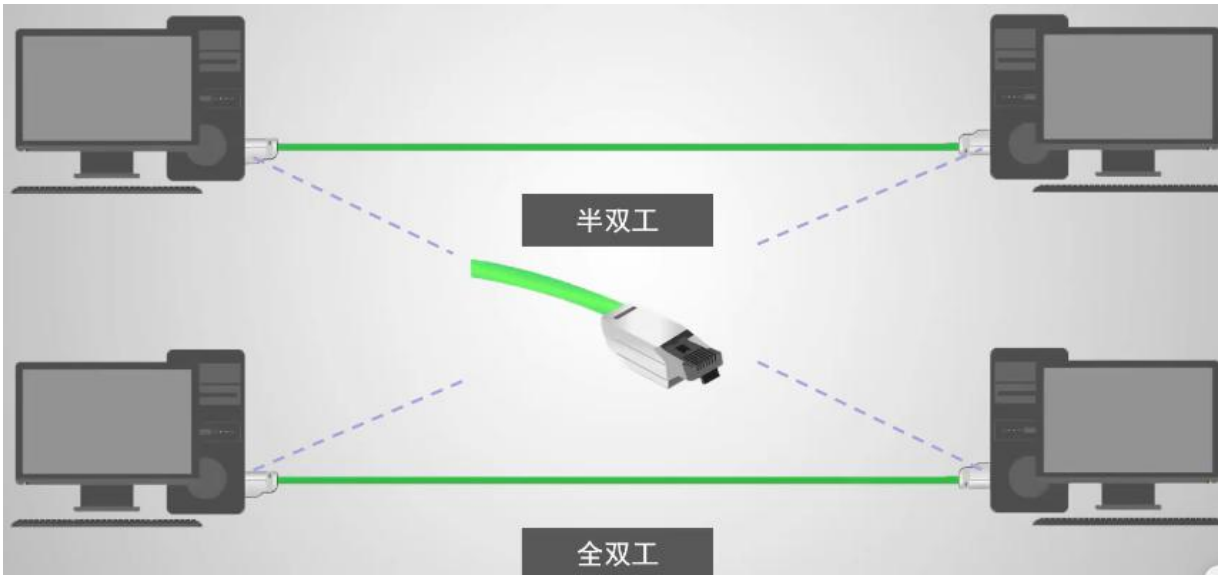
- 1982年又修改为第二版规约（实际上也就是最后的版本），即**DIX Ethernet V2**，成为世界上第一个局域网产品的规约。
- 1983年，**IEEE 802**委员会的**802.3工作组**制定了第一个IEEE的以太网标准**IEEE 802.3**[W-IEEE802.3]，数据率为**10 Mbit/s**。以太网的两个标准 DIX Ethernet V2与IEEE的802.3标准只有很小的差别，因此很多人也常把802.3局域网简称为"以太网"。
- IEEE 802委员会的介绍（引用自《计算机网络-谢希仁》）

  ① 注：IEEE 802 委员会是专门制定局域网和城域网标准的机构。目前（2016 年）其下属仍在活动的工作组只有 8 个，即：802.1——桥接/体系结构；802.3——以太网；802.11——无线局域网；802.15——无线个人区域网；802.16——宽带无线接入；802.19——无线共存(Wireless Coexistence)；802.21——媒体无关切换(Media Independent Handoff)；802.22 无线偏远地区网络(Wireless Regional Area Networks)。其余的都已经暂时或完全停止了活动。所有 802 标准都可从互联网上下载[W-IEEE802]。

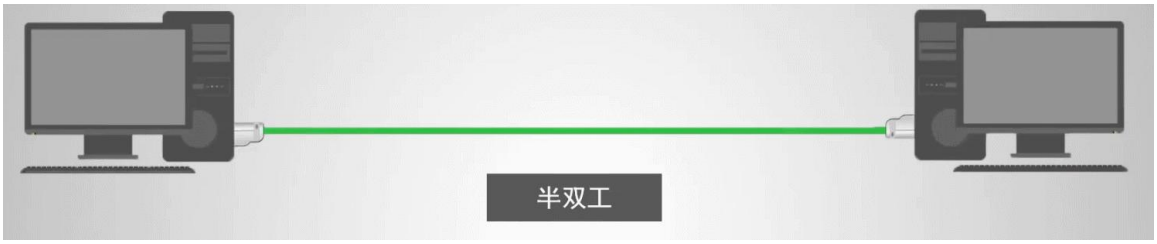
- IEEE 802.3 定义了以太网的**物理层**和**数据链路层**的介质访问控制部分，其中物理层由两个组件组成：**线缆**和**设备**，数据链路层可以分为两部分：**逻辑链路控制（LLC）**、**介质访问控制（MAC）**。



- 物理层
  - 线缆：以太网的通信线缆由最先的同轴电缆发展到今天的双绞线和光纤。
    - 双绞线两端配有**RJ45**八针连接器，这种八针连接器用于在半双工和全双工模式下进行数据的发送和接收。

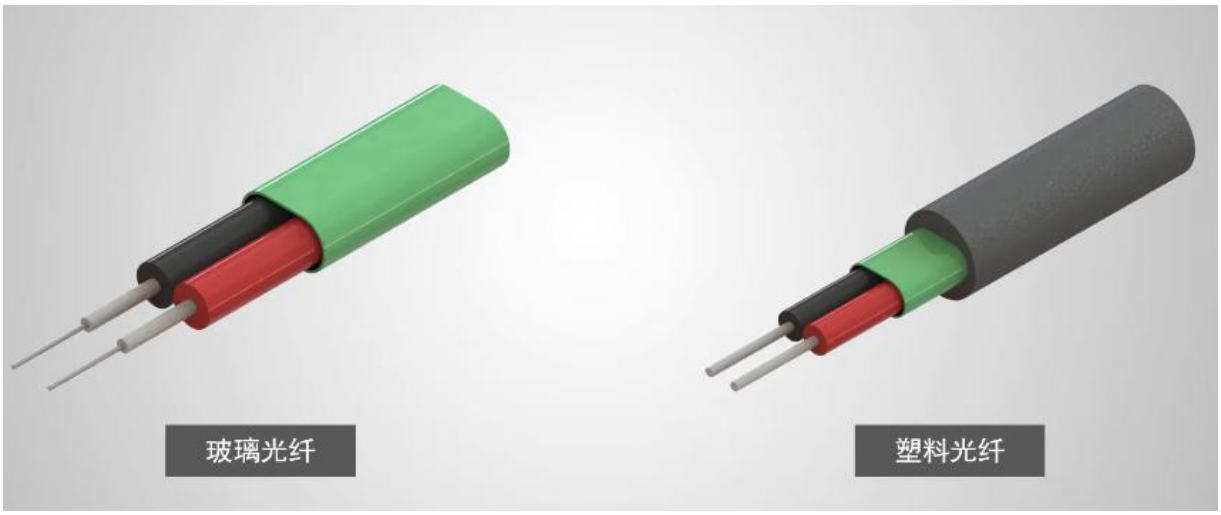


- 半双工模式：数据一次沿一个方向传输

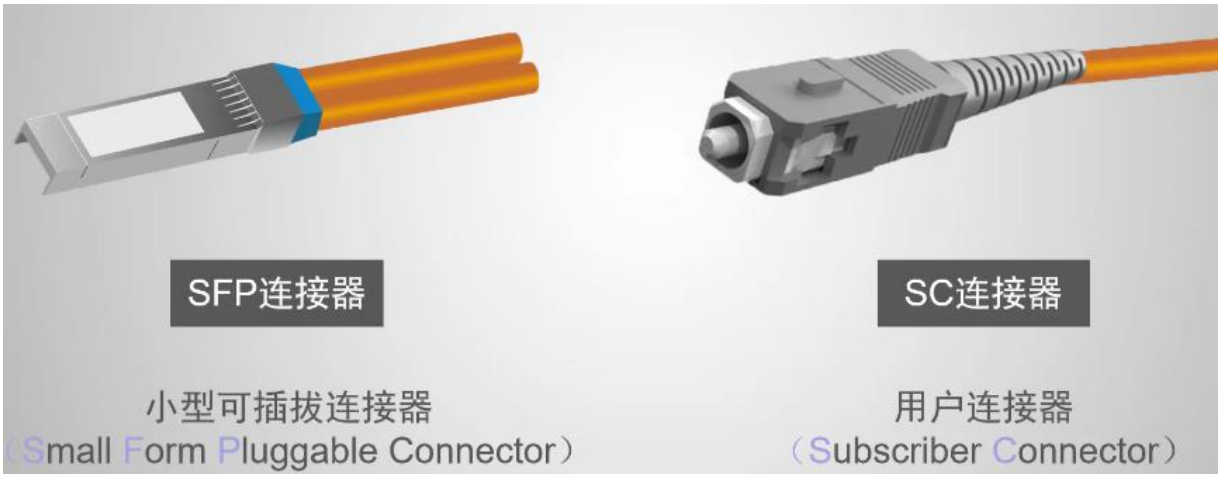


- 全双工模式：数据沿两个方向传输，以太网的全双工模式可以通过使用一对双绞线实现

- 光纤线缆：光纤线缆使用**玻璃光纤**或**塑料光纤**作为光脉冲的传输通道来传输数据



- 光纤电缆可以根据实际需求使用不同类型的连接器：**SFP连接器、SC连接器**



  - 设备：以太网设备由计算机、打印机等具有网络接口卡的设备所组成，常用的以太网设备有**路由 器、交换机、网桥**，而工作在数据链路层上的设备为交换机、网桥，路由器工作在五层体系结构中的网络层。

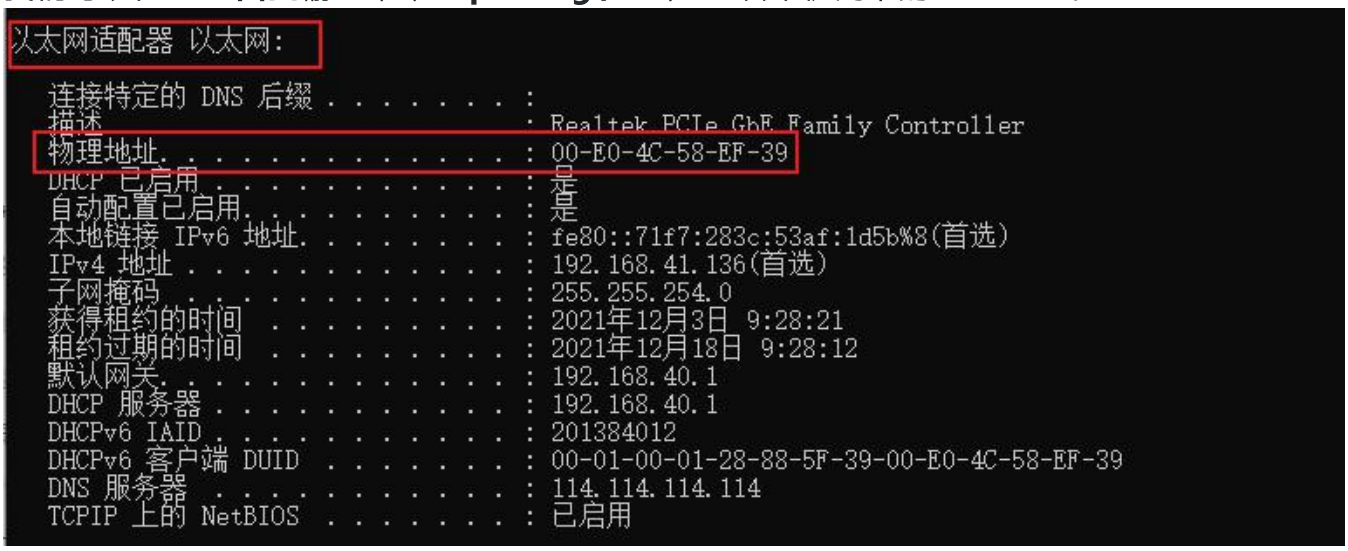
- 数据链路层

- 逻辑链路控制LLC：为网络层提供统一的接口以便数据在设备间传输。很多厂商生产的适配器上就仅装有 MAC 协议而没有 LLC 协议。
- 介质访问控制MAC：使用分配给网络接口卡的硬件地址来标识特定的计算机或设备接口，通过这种方法来表示数据传输的源地址和目的地址。

2.3.5 以太网的MAC层

1、MAC层的硬件地址

- **MAC地址 （英语：Media Access Control Address）** ，直译为**媒体存取控制位址** ，也称为**局域网地址** （LAN Address），**MAC位址** ，**以太网地址** （Ethernet Address）或**物理地址** （Physical Address），它是一个用来确认网络设备位置的位址。
- IEEE 802标准为局域网规定了一种**48位（6字节）的全球地址（一般简称为“地址”），这个地址会固化在适配器**的**ROM**中。
- IEEE 的**注册管理机构 RA** 负责向厂家分配地址字段的**前三个字节(即高位 24 位)**。地址字段中的后三个字节(即低位 24 位)由**厂家自行指派**，称为**扩展标识符**，必须保证生产出的适配器没有重复地址。世界上凡要生产局域网适配器的厂家都必须向IEEE购买由这三个字节构成的这个号（即地址块)，这个号的正式名称是组织唯一标识符。
  - > 例如，3Com公司生产的适配器的MAC地址的前三个字节是 02-60-8C。地址字段中的后三个字节（即低位24位）则由厂家自行指派，只要保证生产出的适配器没有重复地址即可
- 一个地址块可以生成**2^24**个（二百八十多万亿个）不同的地址。这种 48 位地址称为**MAC-48**，它的通用名称是**EUI-48**。
- 一般情况下，用户主机会包含两个网络适配器：**有线局域网适配器（有线网卡）和无线局域网适配器（无线网卡)**。每个网络适配器都有一个全球唯一的MAC地址。而**交换机和路由器往往拥有更多的网络接口，所以会拥有更多的MAC地址**。综上所述，严格来说，MAC地址是对网络上各接口的唯一标识，而不是对网络上各设备的唯一标识。
- 我们可以在**DOS窗口**输入命令：**ipconfig /all** ， 查看本机网卡的MAC地址。



2、MAC地址的格式

- **MAC地址的表示方法**

IEEE 802局域网的MAC地址格式

扩展的唯一标识符EUI
EUI-48

组织唯一标识符OUI（由IEEE的注册管理机构分配）								网络接口标识符（由获得OUI的厂商自行随意分配）							
第一字节								第二字节							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
十六进制	X		X		X		X		X		X		X		X

标准表示法：XX-XX-XX-XX-XX-XX　例如：00-0C-CF-93-8C-92

其他表示法：XX:XX:XX:XX:XX:XX　例如：00:0C:CF:93:8C:92

XXXX.XXXX.XXXX　例如：000C.CF93.8C92

- **MAC地址的含义**

组织唯一标识符OUI（由IEEE的注册管理机构分配）								网络接口标识符（由获得OUI的厂商自行随意分配）							
第一字节								第二字节							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0

0：全球管理
1：本地管理

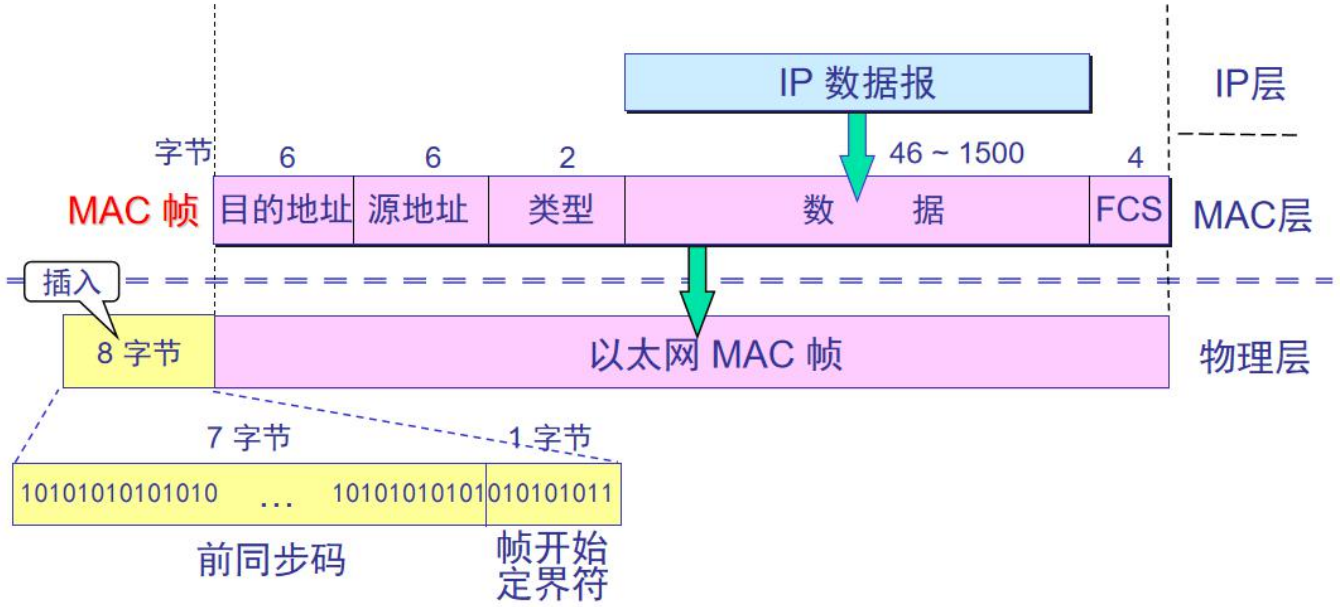
0：单播
1：多播

第一字节的b1位	第一字节的b0位	MAC地址类型	地址数量占比	总地址数量
0	0	全球管理 单播地址 厂商生产网络设备（网卡，交换机，路由器）时固化	1/4	2^{48}=281,474,976,710,656（二百八十多万亿）
	1	全球管理 多播地址 标准网络设备所支持的多播地址，用于特定功能	1/4	
1	0	本地管理 单播地址 由网络管理员分配，覆盖网络接口的全球管理单播地址	1/4	
	1	本地管理 多播地址 用户对主机进行软件配置，以表明其属于哪些多播组 注意：剩余46位全为1时，就是广播地址FF-FF-FF-FF-FF-FF	1/4	

3、MAC 帧的格式

数据链路层在网络层交付的IP数据包前面加上“**目的地址**”、“**源地址**”、“**类型**”字段，并且在最后加入4字节的**FCS**字段，组成一个**以太网MAC帧**，然后再交付给物理层

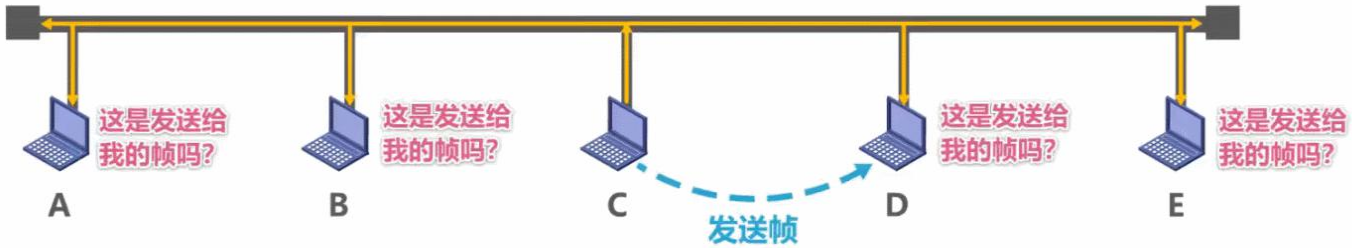
- 目的地址：接收主机的MAC地址
- 源地址：发送主机的MAC地址
- 类型：类型字段标志上一层使用的是什么**协议**，以便把收到的 MAC 帧的数据上交给上一层的这个协议。



> 注意：当数据字段的长度小于 46 字节时，应在数据字段的后面加入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。

4、MAC地址的识别

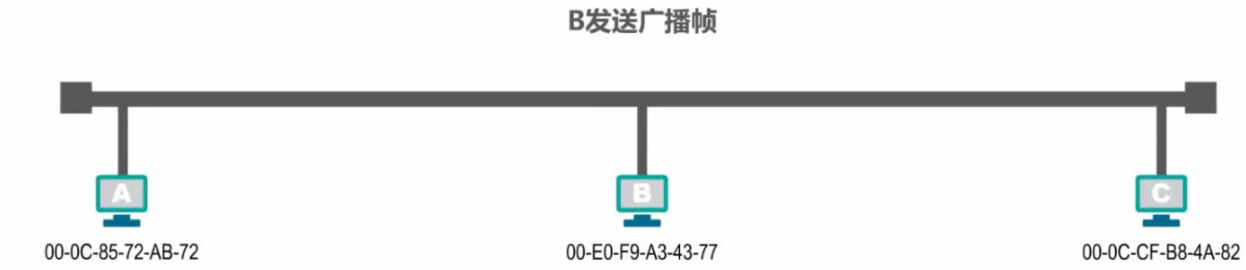
当多个主机连接到同一个广播信道上，要实现两个主机之间的通信，每个主机发送的帧中包含了目的地址和源地址，广播信道上的每一台主机都能够收到该帧，接收到帧的主机将帧中的目的地址与保存在网络适配器的电可擦除可编程只读存储器**EEPROOM**中的MAC地址进行比较，如果匹配则接受该帧，否则就丢弃该帧。



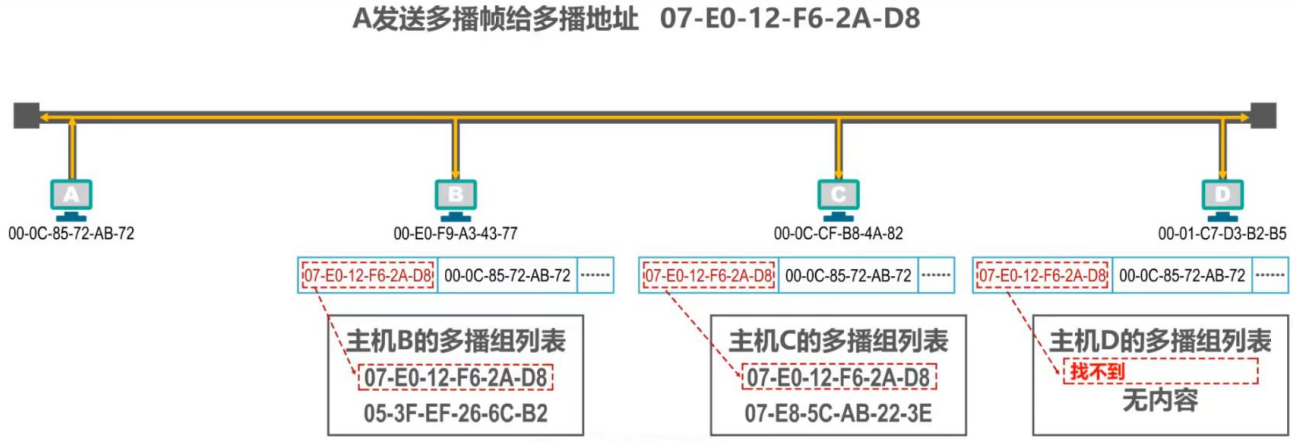
根据MAC地址为单播MAC地址还是广播MAC地址还是多播MAC地址，在计算机网络中“**发往本站（本主机）的帧**”分为三种：**单播(unicast)帧（一对一）、广播(broadcast)帧（一对全体）、多播(multicast)帧（一对多）。**

- **单播MAC地址**

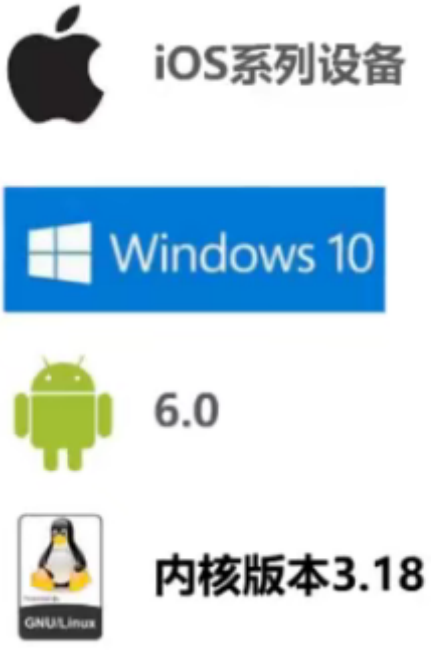
- **广播MAC地址**



- **多播MAC地址**



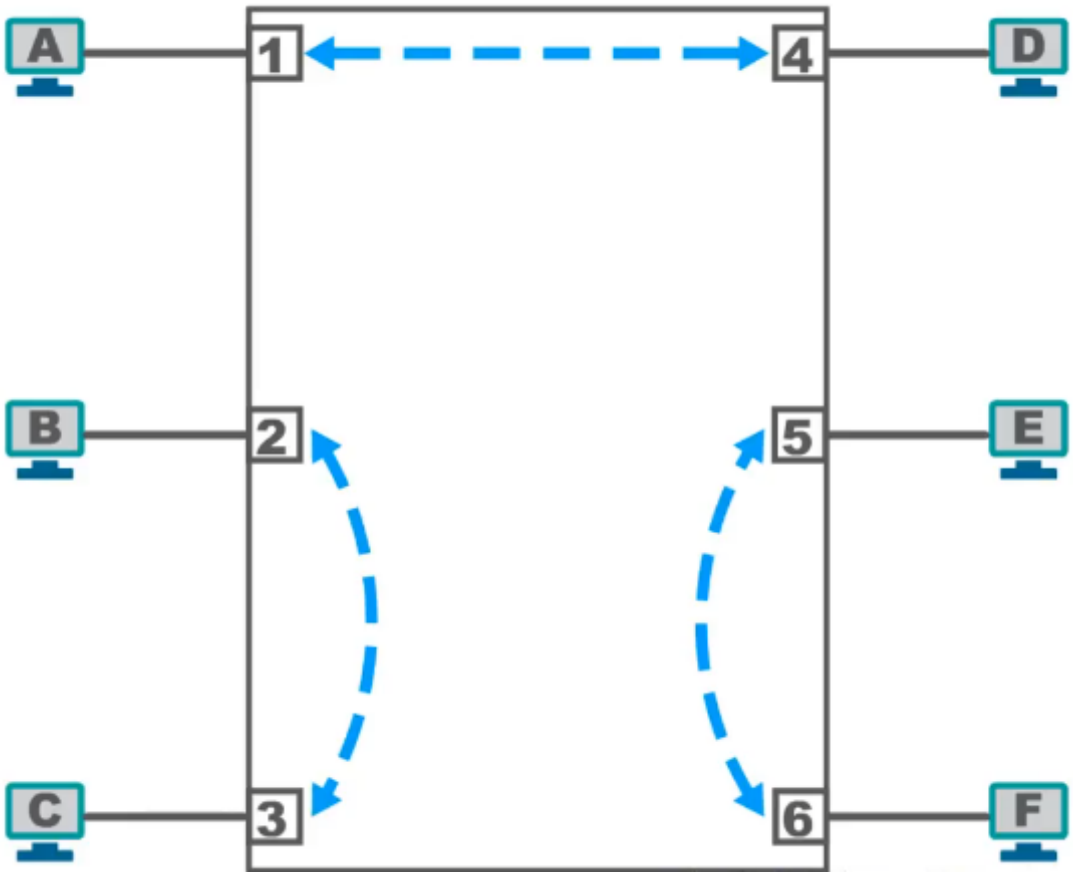
- **随机MAC地址**：据斯诺登介绍，美国国家安全局有一套系统通过监视电子设备的MAC地址来跟踪城市中每个人的行动，因此苹果率先在**ios**系列设备扫描网络时采用随机MAC地址技术，随后**Windows10**，**安卓6.0**以及**内核版本3.18的Linux**系统也提供随机MAC地址功能。 目前大多数移动设备都采用了随机MAC地址技术。



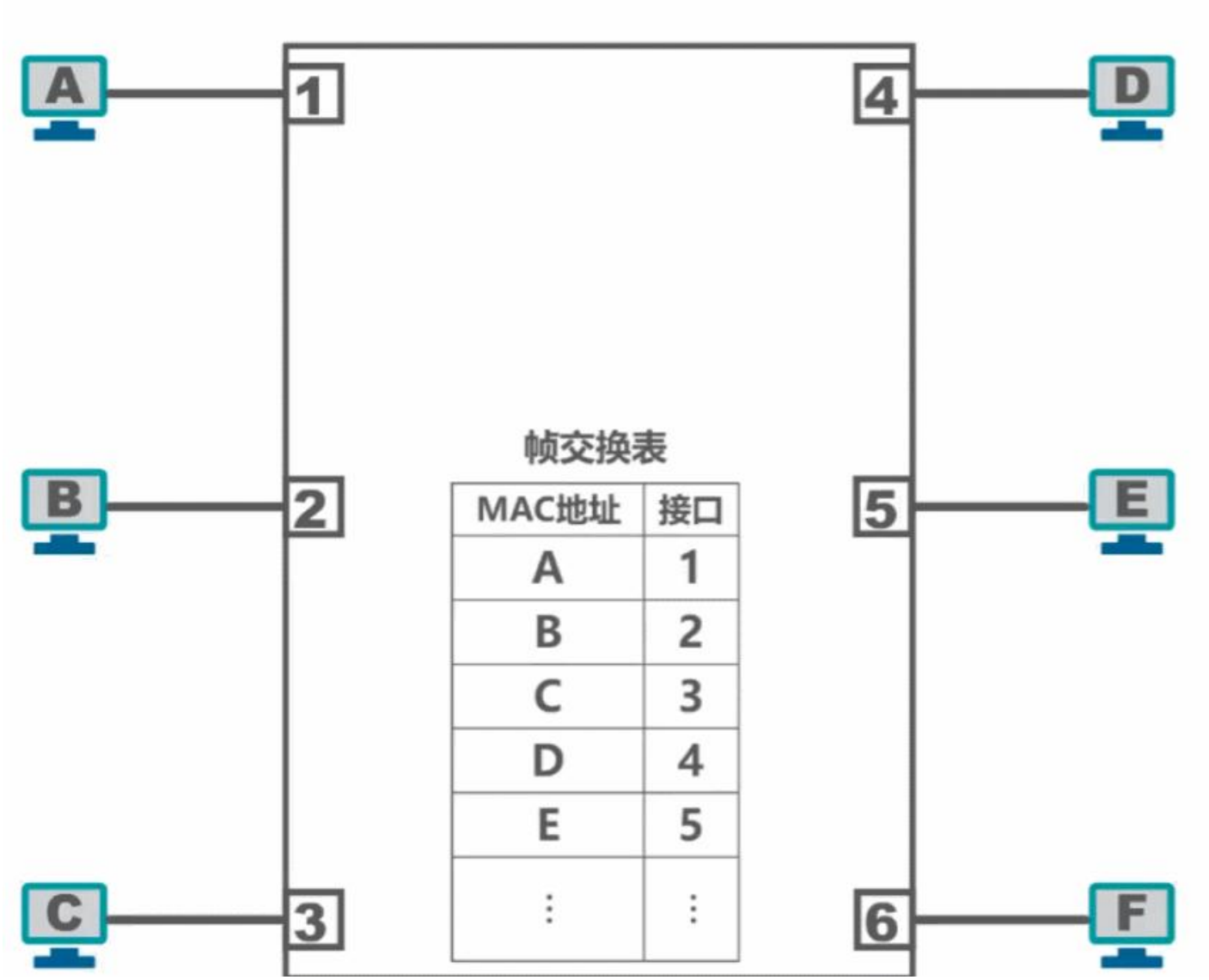
2.3.6 以太网交换机

1、以太网交换机的基本功能

- 以太网交换机是基于**以太网传输数据**的交换机，以太网交换机通常都有多个接口，每个接口都可以直接与一台主机或另一个以太网交换机相连，一般都工作在**全双工方式**。
- 以太网交换机具有并行性，能同时连通多对接口，使多对主机能同时通信。



- **以太网交换机工作在数据链路层（也包括物理层）**，它收到帧后，在帧交换表中查找帧的目的MAC地址所对应的接口号，然后通过该接口转发帧。



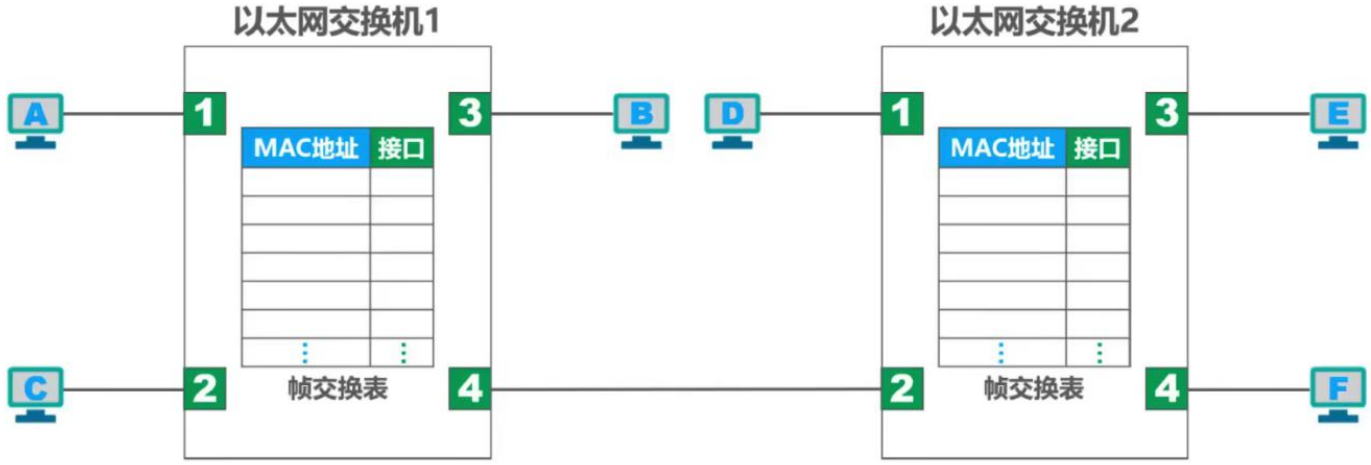
- 帧的两种转发方式：
  - 存储转发：交换机在转发之前必须**接收整个帧**，并进行**错误校检**，如无错误再将这一帧发往目的地址。帧通过交换机的转发时延随帧长度的不同而变化。
  - 直接交换：采用基于硬件的**交叉矩阵**（交换机只要检查到帧头中所包含的目的地址就立即转发该帧，而无需等待帧全部的被接收，也不进行错误校验。由于以太网帧头的长度总是固定的，因此帧通过交换机的转发时延也保持不变。）
- 以太网交换机是一种即插即用设备，其内部的**帧交换表**是通过**自学习算法**自动的逐渐建立起来的。

2、以太网交换机自学习和转发帧的流程

刚上电时以太网交换机内部的帧交换表是空的，随着网络中各主机间的通信，通过**自学习算法**自动的逐渐建立起来**帧交换表**。

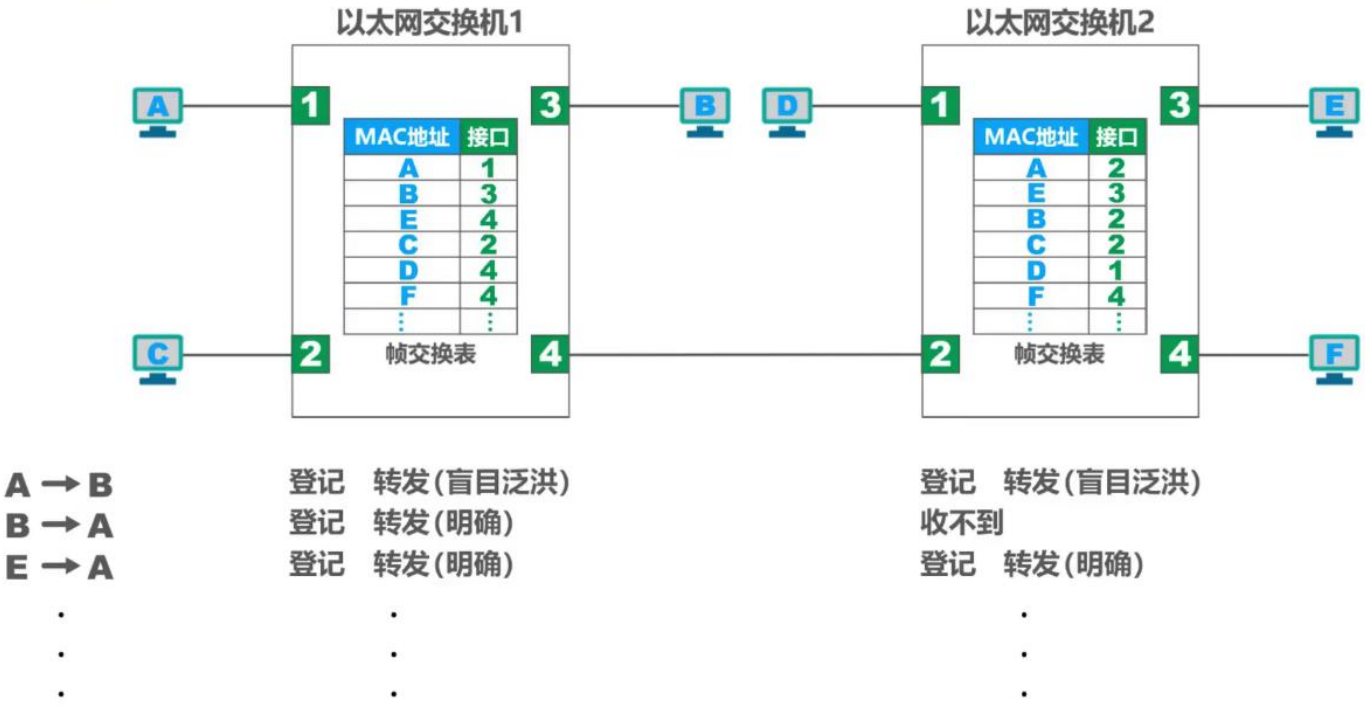
如下图所示相互连接的“以太网交换机1”和“以太网交换机2”各自连接了三台主机，构成了一个交换式以太网。为了简单起见各台主机的MAC地址我们用一个大写字母表示，并且假设各台主机知

道网络中其他各主机的MAC地址。



下面我们来了解以太网交换机是如何进行自学习和转发帧的：

- 主机A发送数据给主机B
  - 该帧从交换机1的接口1进入交换机，交换机1首先进行登记工作，将该帧中的源MAC地址A记录到自己的帧交换表中，并且将接口号1与MAC地址A相对应也记录到帧交换表中。以上的登记工作就称之为交换机的自学习。
  - 交换机1对该帧进行转发，该帧中的目的MAC地址是B，在帧交换表中查找MAC地址B，发现找不到就对该帧进行盲目地转发（也成为泛洪），也就说会将该帧在除接口1以外的其他所有接口进行转发。
  - 主机B在接收到该帧后，根据该帧的目的MAC地址B与自己的MAC地址比较，发现相等就说明该帧是发送给自己的，于是主机B接受该帧，主机C则会丢弃该帧。
  - 该帧从交换机1的接口4通过交换机2的接口2进入交换机2，交换机2首先进行登记工作，将该帧中的源MAC地址A记录到自己的帧交换表中，并且将接口号2与MAC地址A相对应也记录到帧交换表中
  - 交换机2对该帧进行转发，该帧中的目的MAC地址是B，在帧交换表中查找MAC地址B，发现找不到就对该帧进行盲目地转发（也成为泛洪），也就说会将该帧在除接口2以外的其他所有接口进行转发。
  - 主机D、E、F丢弃该帧

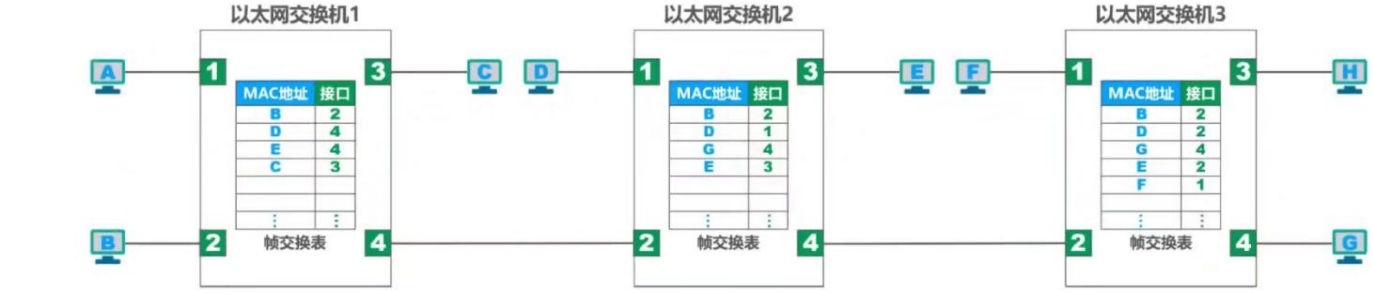


> 注意：在帧交换表中每条记录都有自己的有效时间，到期自动删除。这是因为MAC地址与交换机接口的对应关系并不是永久性的（对应的接口可能会更换计算机）

【习题】为简单起见，主机A，B，C，D，E，F，G，H的MAC地址与其主机名称相同。主机间依次如下通信：

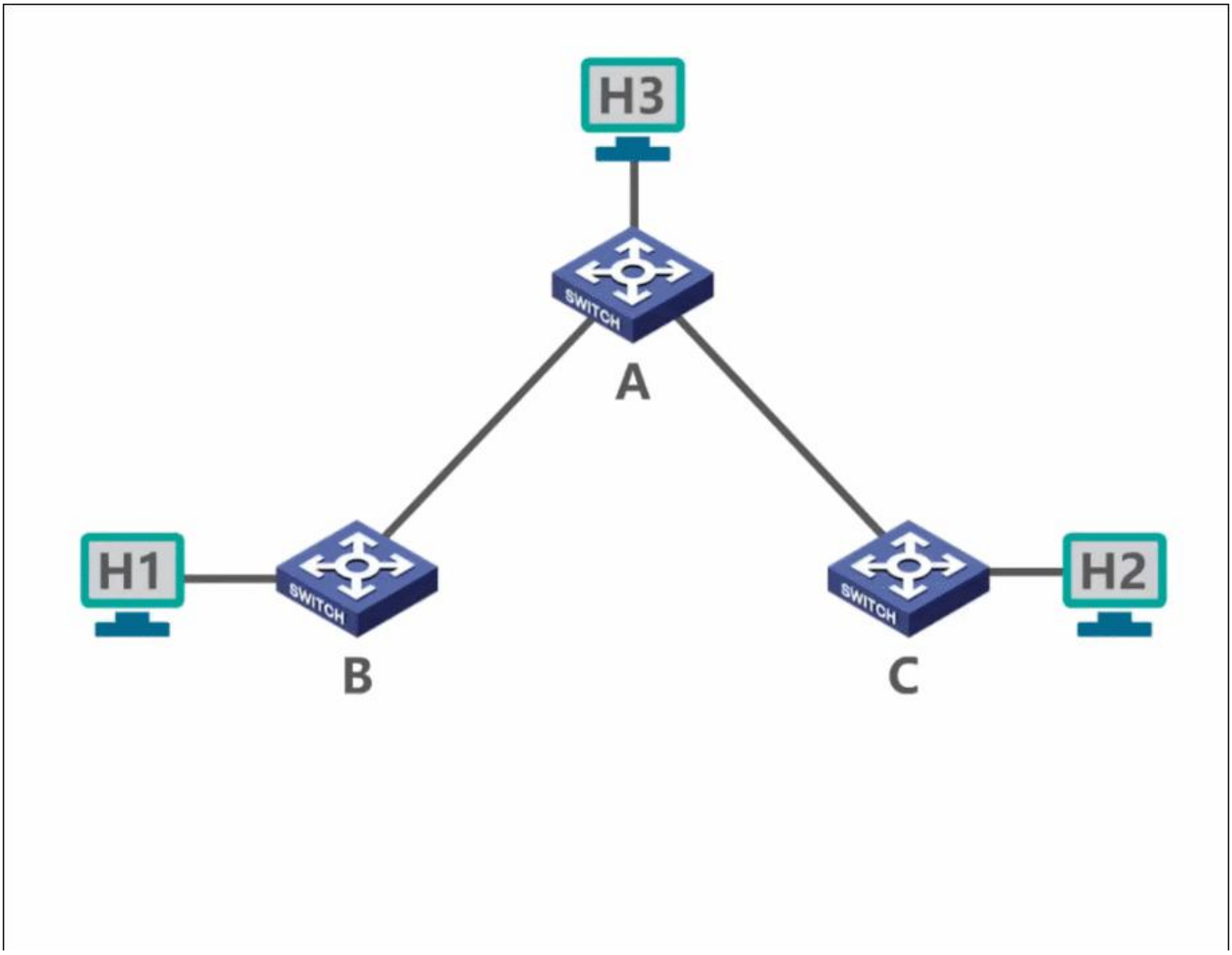
1. B → C　2. D → A　3. G → D　4. E → H　5. C → B　6. F → G

请给出以太网交换机1，2，3的自学习过程以及各自最终的帧交换表的内容。

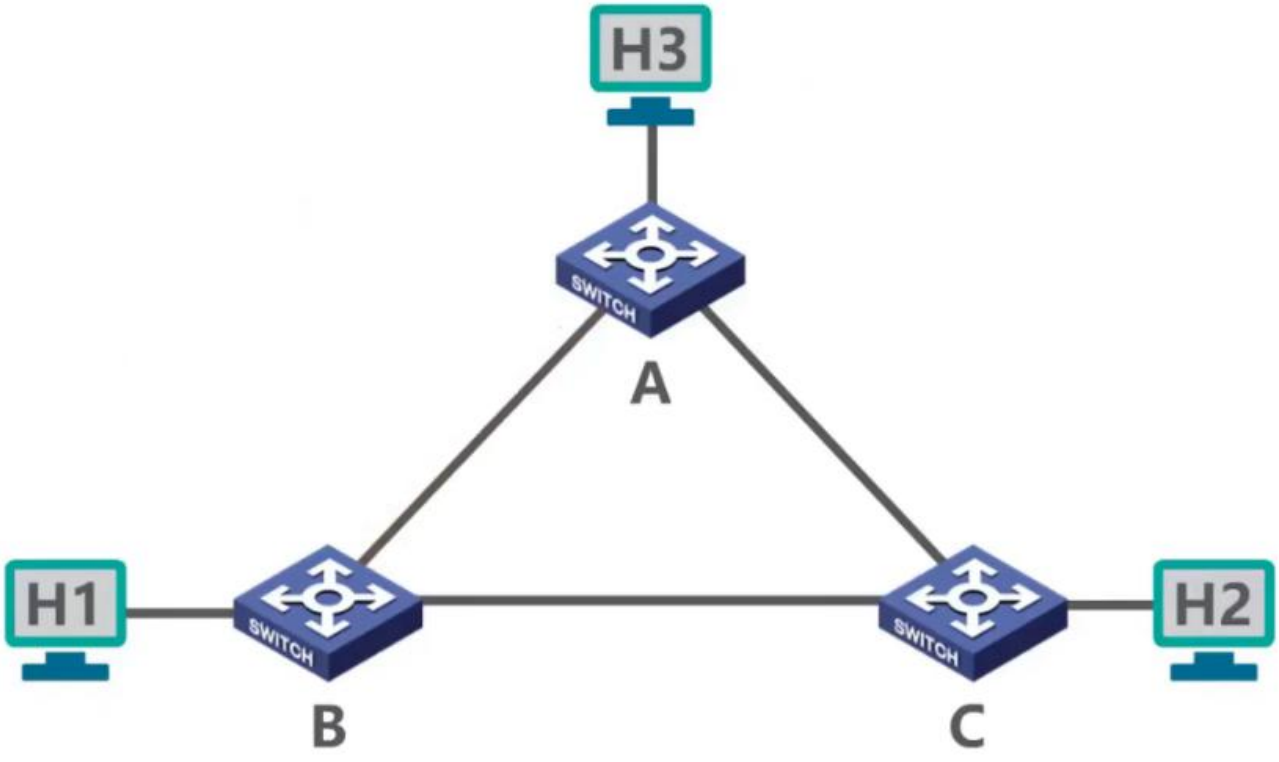


3、以太网交换机的生成树协议STP

- 思考：如何提高以太网的可靠性？假如在网络中有三台交换机A、B、C，他们之间的连接方式如下图所示，假如他们之间的链路出现了故障则会影响各个交换机之间的通信

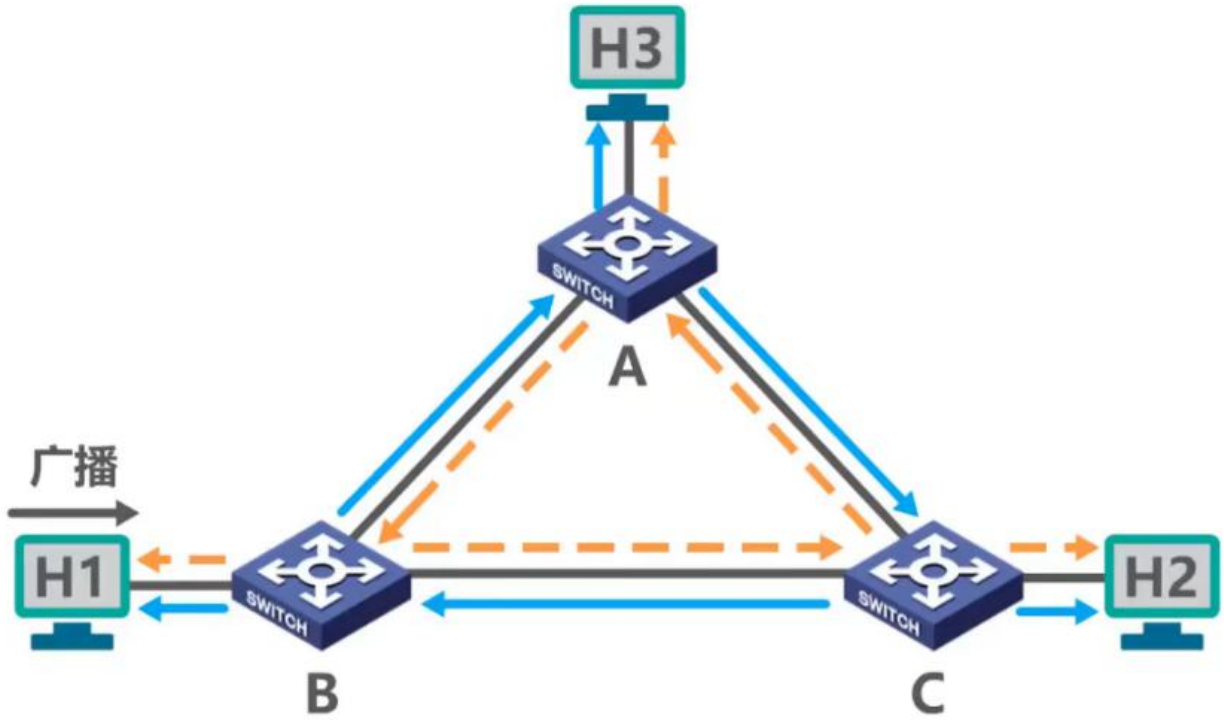


- 添加**冗余链路**可以提高以太网的可靠性：在交换机B和C之间添加冗余链路。但是冗余链路也会带来负面效应----**形成网络环路。**



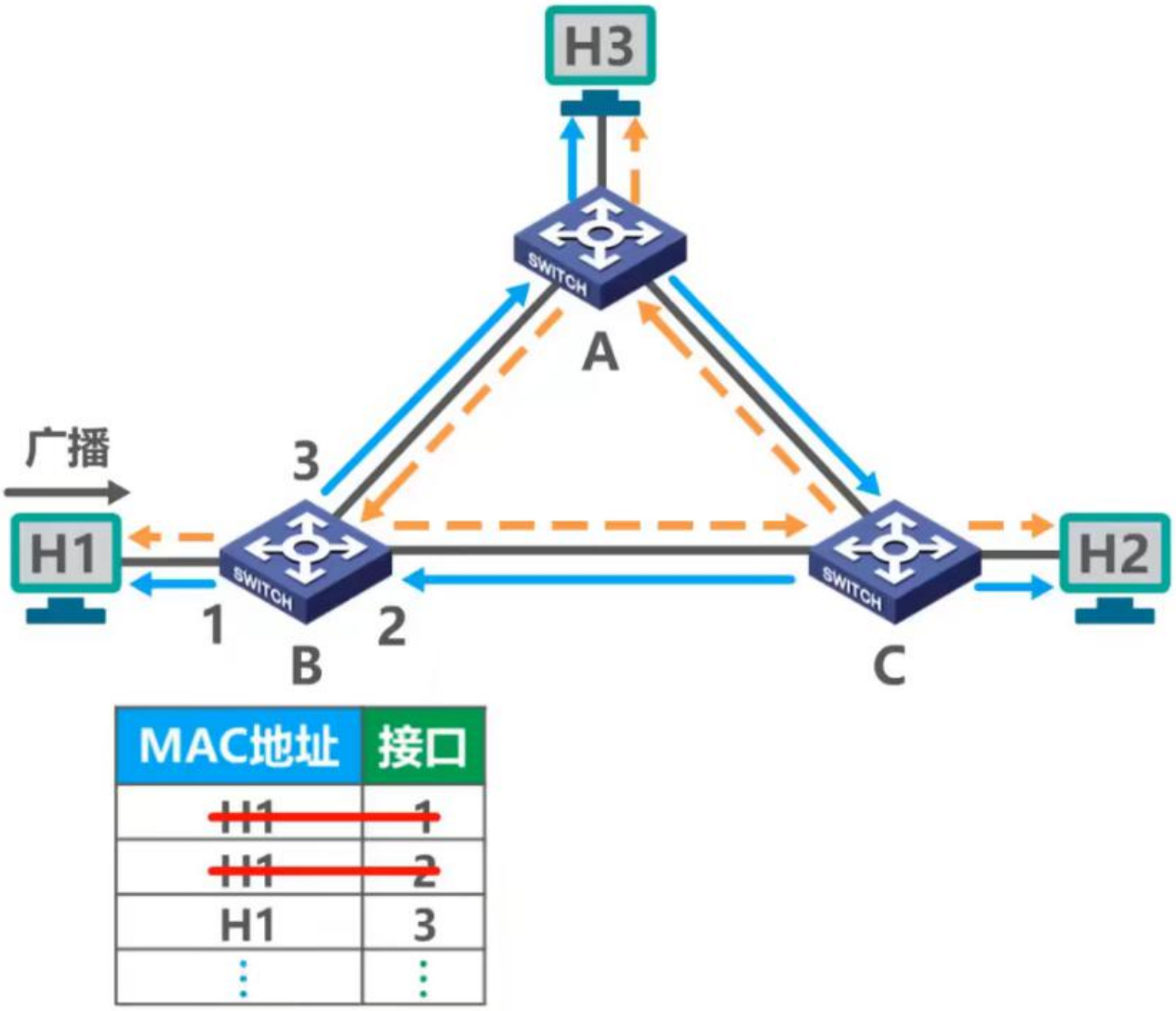
- 网络环路带来的问题

- **广播风暴**：广播帧在各个交换机之间反复转发，分别按顺时针和逆时针方向同时兜圈。广播风暴会大量消耗网络资源，使得网络无法正常转发其他数据帧。



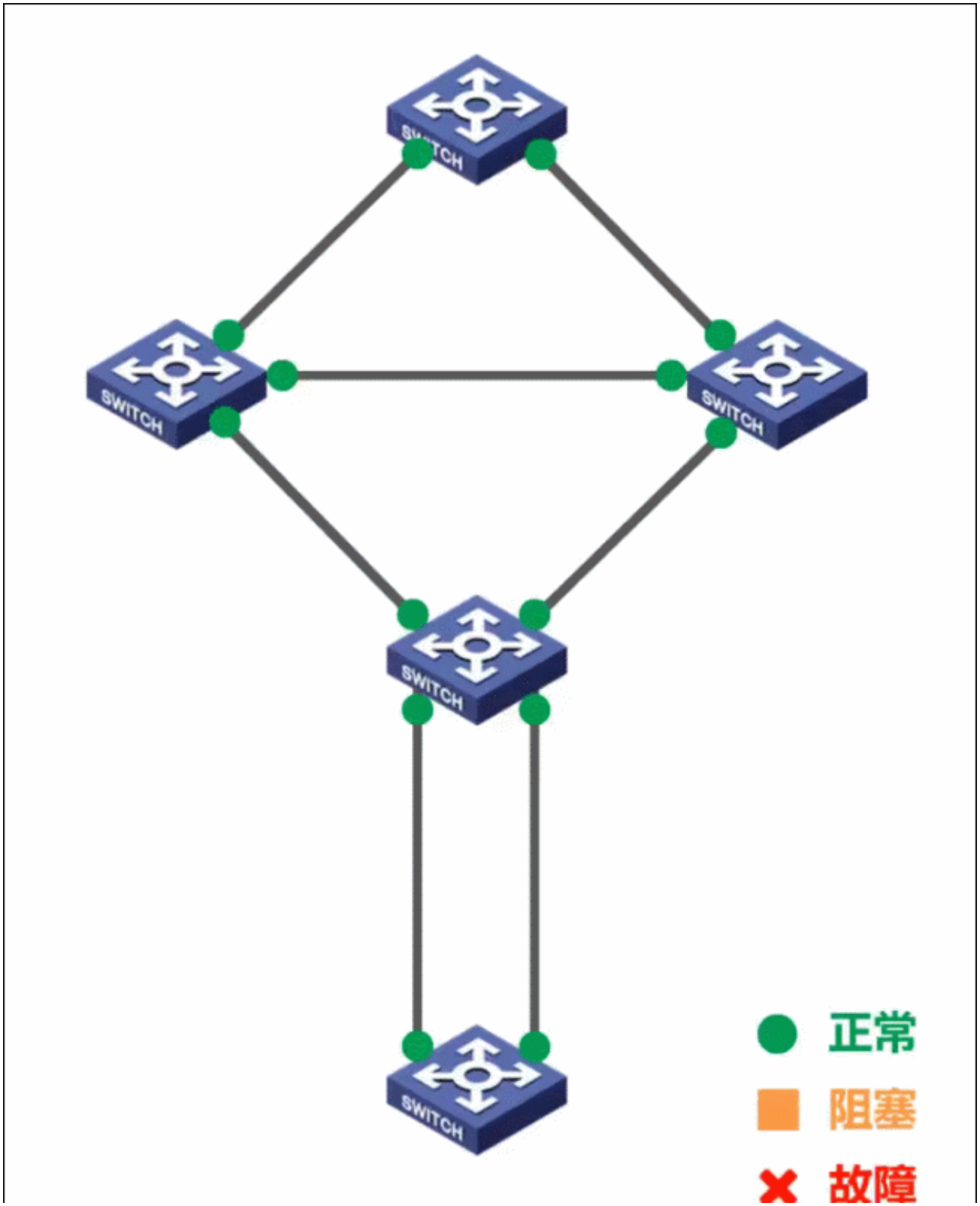
- **主机收到反复的广播帧**，会大量消耗主机的资源。

- **交换机的帧交换表震荡**：同一个MAC地址的记录在其他错误记录之间反复震荡。

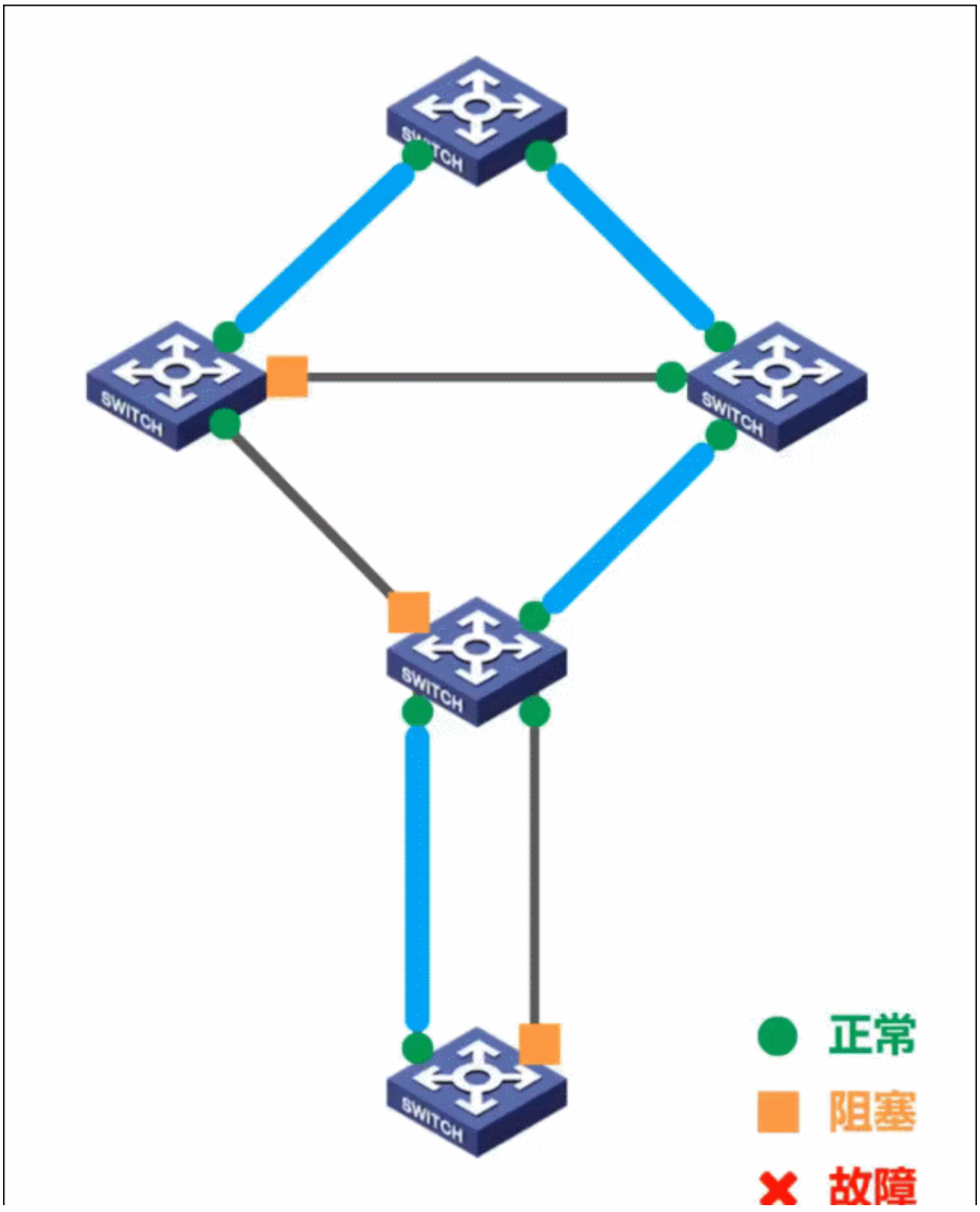


- 以太网交换机使用**生成树协议STP**(Spanning Tree Protocol)可以在增加冗余链路来提高网络可靠性的同时又**避免网络环路带来的各种问题。**
  - 不论交换机之间采用怎样的物理连接，交换机都能够**自动计算并构建一个逻辑上没有环路的网络**，其逻辑拓扑结构必须是**树型**的（无逻辑环路)。最终生成的树型逻辑拓扑要**确保连**

通整个网络；



- 当首次连接交换机或网络**物理拓扑发生变化**时(有可能是人为改变或故障)，交换机都将进行**生成树的重新计算。**

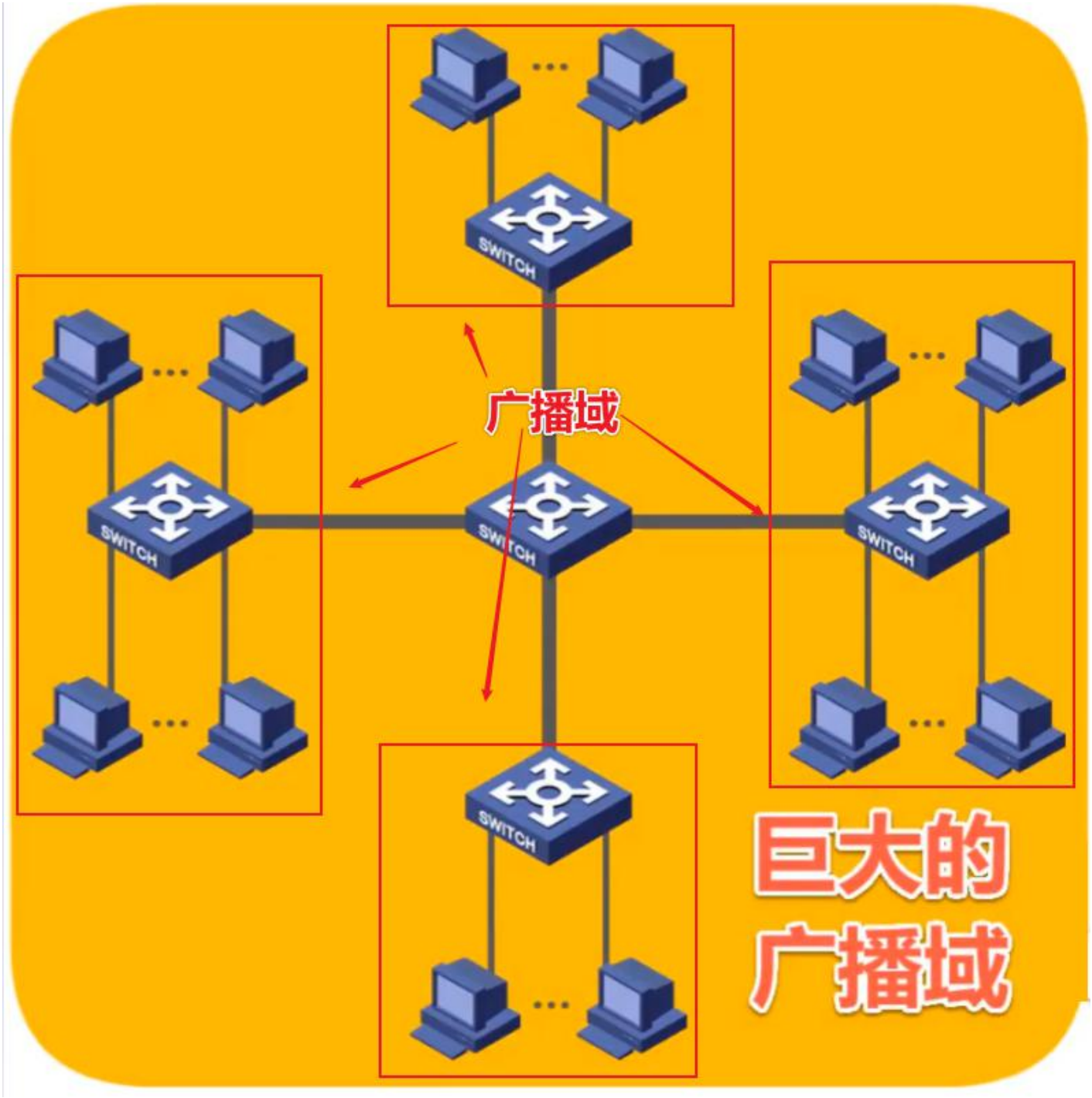


- **说明：生成树算法STA需要大家自行了解。**

2.3.7 虚拟局域网VLAN

1、广播域

使用一个或多个以太网交换机互连起来的交换式以太网，其所有站点都属于**同一个广播域。**随着交换式以太网规模的扩大，广播域相应扩大，从而形成一个巨大的广播域。

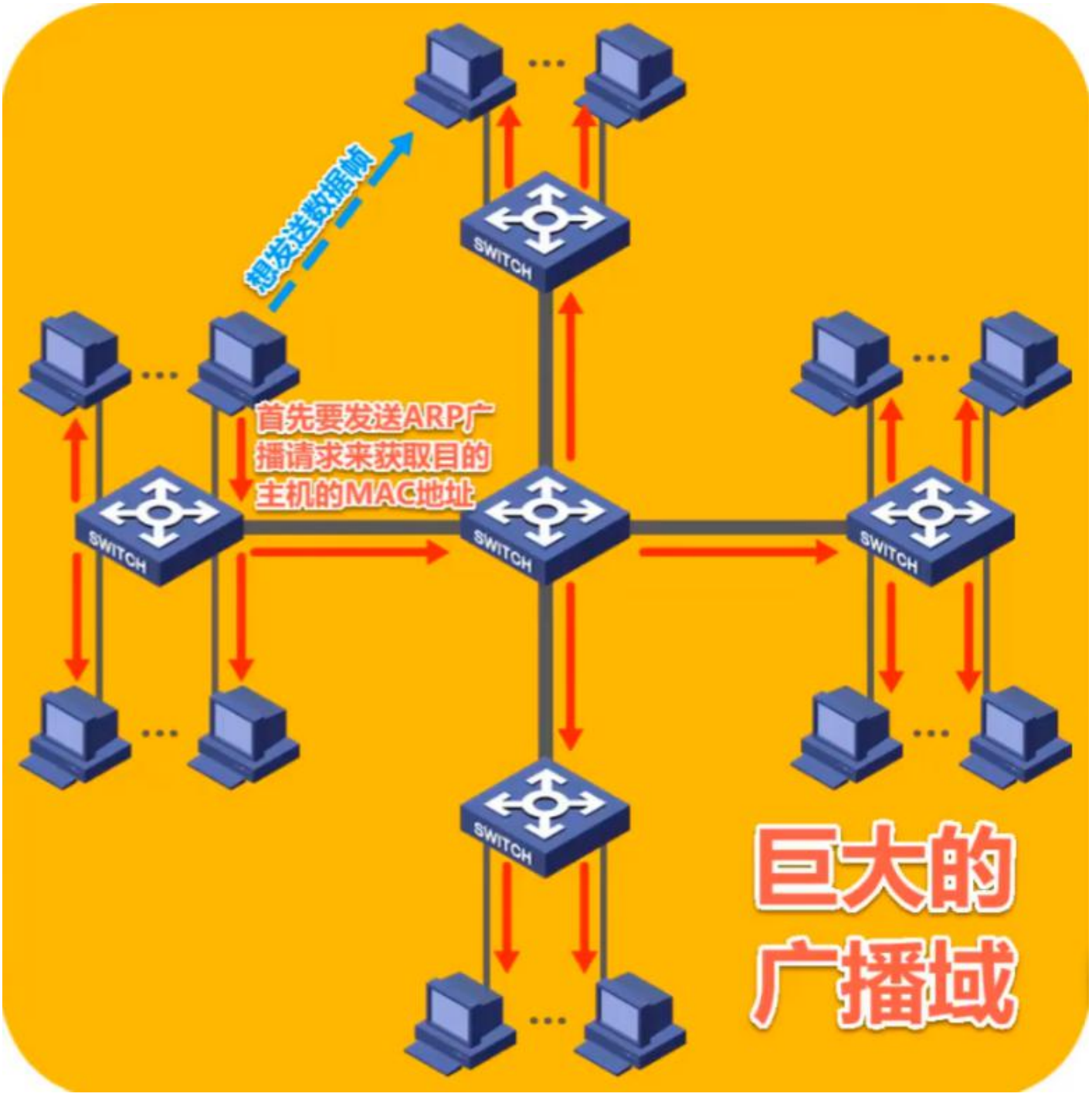


但是巨大的广播域会带来很多的弊端：

- 广播风暴
- 难以管理和维护
- 潜在的安全问题

例如，网络中的某台主机A需要向另外一台主机B发送数据，此时主机A只有主机B的IP地址，但是没有主机B的MAC地址，主机A需要发送ARP（ARP属于体系结构的网络层，在后续的课程中我们会介绍）广播请求来获取主机B的MAC地址，该ARP请求会传遍整个网络，网络中的其他所有主机都能

够收到广播。

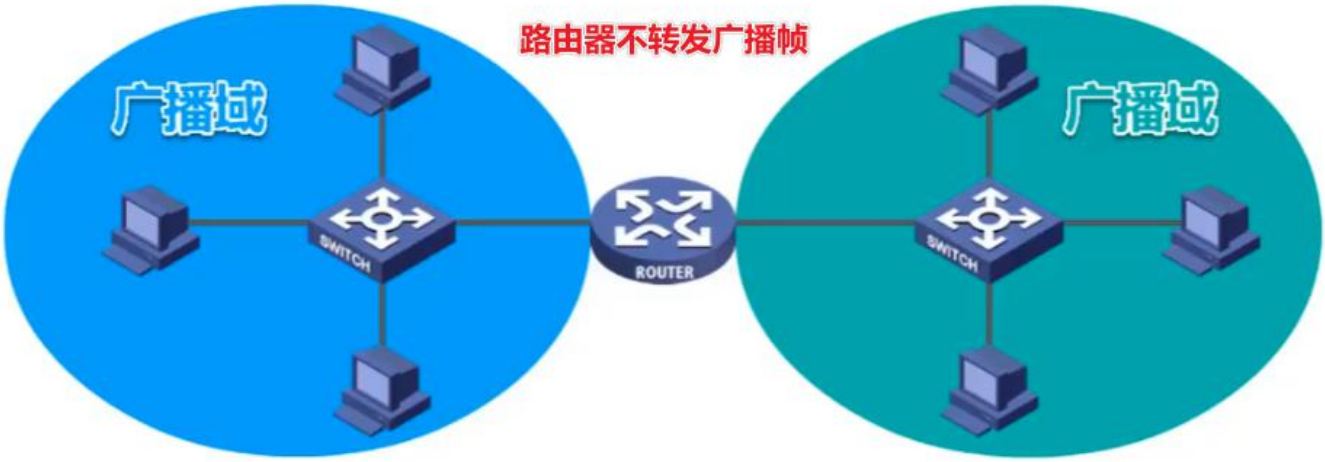


广播风暴会浪费网络资源和各个主机的CPU资源，但是在实际应用中网络中会频繁出现广播信息，TCP/IP协议栈中的很多协议都会使用广播：

- **地址解析协议ARP（已知IP地址，找出对应的MAC地址）**
- **动态主机配置协议DHCP（用于自动配置IP地址）**

我们可以使用**分割广播域**的方法对广播域进行**隔离**：

- **使用路由器隔离广播域**：但是路由器的成本较高，局域网中如果全部用路由器隔离广播域不大现实



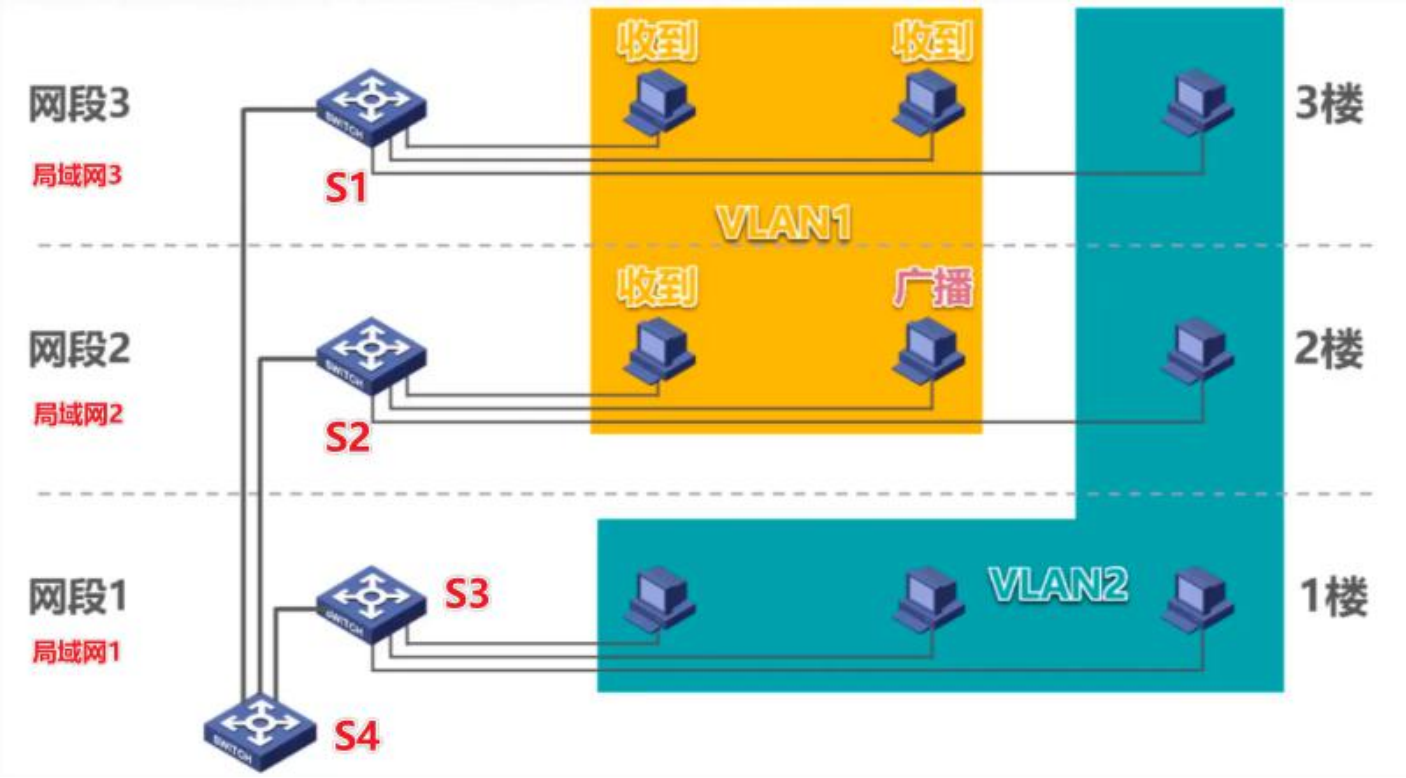
- **使用虚拟局域网VLAN**

2、虚拟局域网VLAN概述

虚拟局域网VLAN(Virtual Local Area Network)是一种将局域网内的**设备划分成与物理位置无关的逻辑组的技术**，这些逻辑组具有某些共同的需求。

假如在网络中有三个局域网，分别叫做局域网1、局域网2、局域网3，我们可以使用一个**交换机S4**将三个局域网互连成一个更大的局域网。原来每个局域网就成为新的大的局域网中的各个**网段**。我们可以在S4上将整个局域网划分成两个VLAN：VLAN1和VLAN2，这样两个**不同的VLAN之间的广播数据包不会互相传输**，当然如果是在同一个VLAN中的主机依然可以进行广播通信。

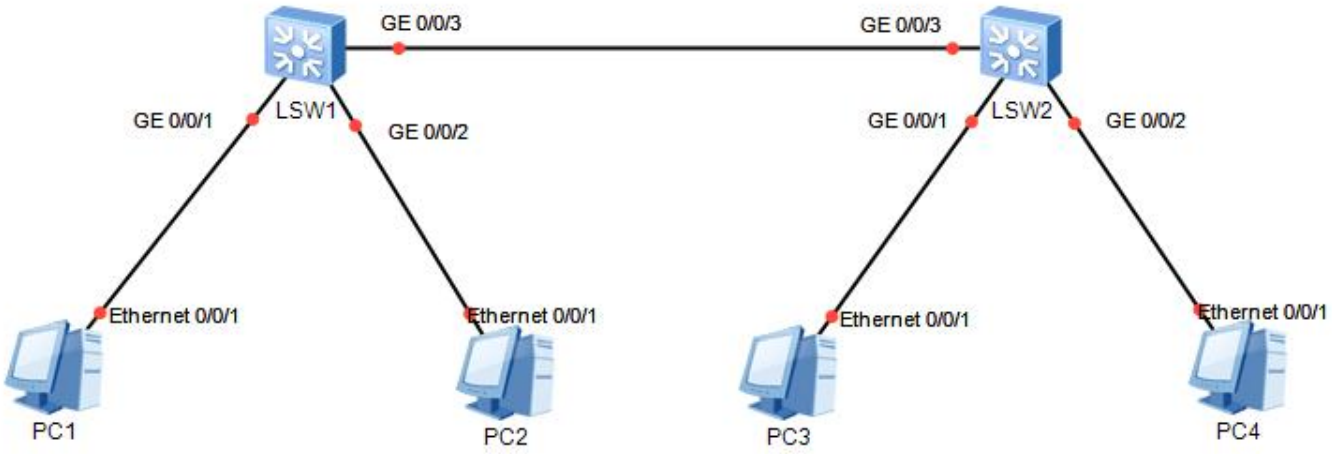
> **注意：不是所有的交换机都能够进行VLAN划分，一般需要企业级交换机才可以。**



2.3.8 华为eNSP VLAN实验

下面我们使用华为的eNSP进行VLAN的实验。

1、新建一个拓扑，然后在拓扑中加入两台S5700的交换机LSW1和LSW2，LSW1和LSW2互相连接，并且每台交换机上接入两台PC。



2、如上图所示：LSW1交换机的 GE 0/0/1和GE 0/0/2接口分别接PC1和PC2， LSW2交换机的 GE 0/0/1和GE 0/0/2接口分别接PC3和PC4，LSW1和LSW2的GE 0/0/3接口互连。

3、对PC1、PC2、PC3、PC4进行如下设置：

- IP地址分别设置为192.168.1.101, 192.168.1.102, 192.168.1.103, 192.168.1.104
- 子网掩码全部设置为：255.255.255.0
- 网关全部设置为：192.168.1.0

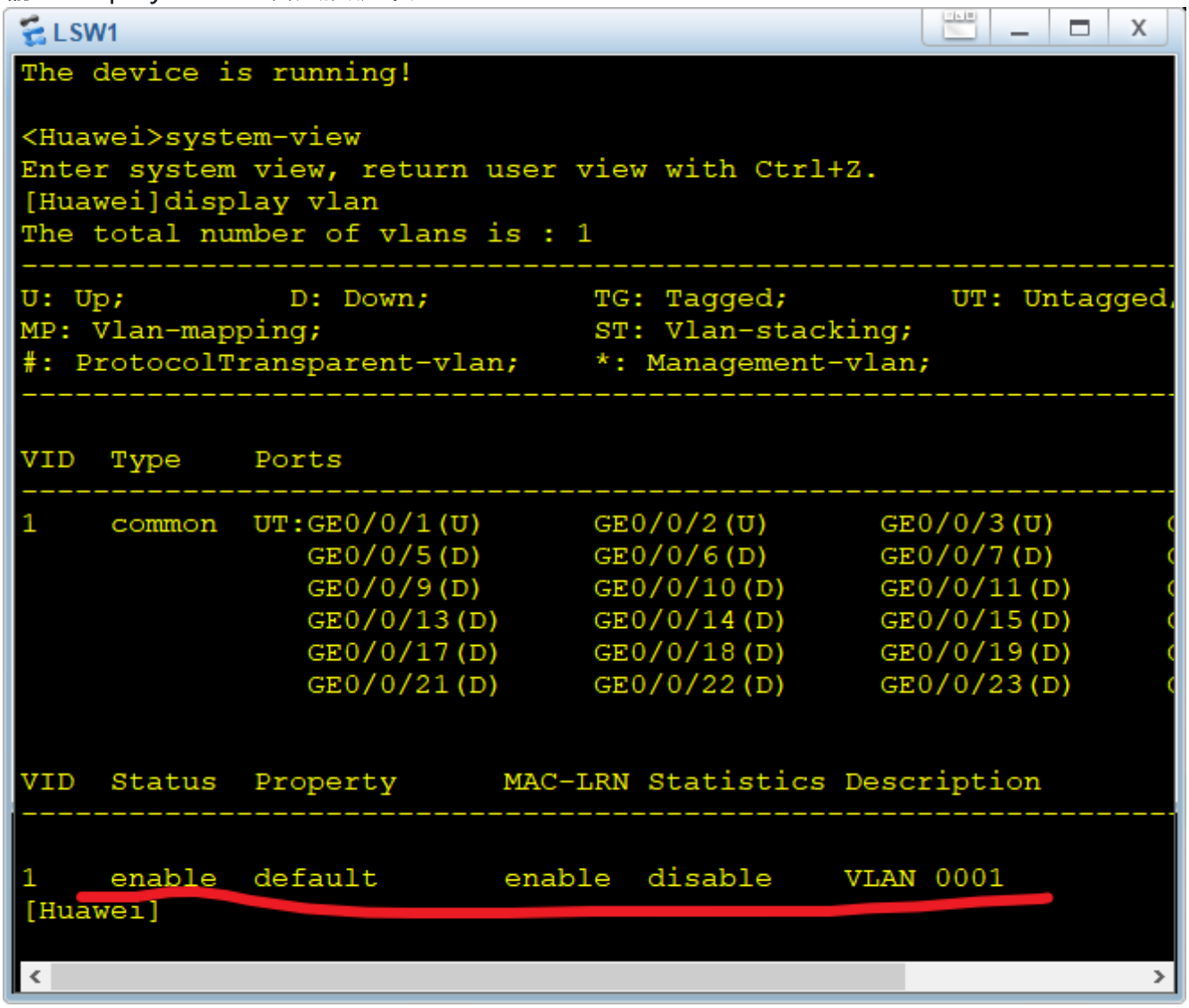
4、设置完成以后，将两台交换机和4台PC启动，然后分别在每台PC的命令行使用ping命令查看能够ping同其他PC。

```
PC>ping 192.168.1.102

Ping 192.168.1.102: 32 data bytes, Press Ctrl_C to break
From 192.168.1.102: bytes=32 seq=1 ttl=128 time=47 ms
From 192.168.1.102: bytes=32 seq=2 ttl=128 time=47 ms
From 192.168.1.102: bytes=32 seq=3 ttl=128 time=15 ms
From 192.168.1.102: bytes=32 seq=4 ttl=128 time=31 ms

--- 192.168.1.102 ping statistics ---
  4 packet(s) transmitted
  4 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 15/35/47 ms

PC>ping 192.168.1.103

Ping 192.168.1.103: 32 data bytes, Press Ctrl_C to break
From 192.168.1.103: bytes=32 seq=1 ttl=128 time=63 ms
From 192.168.1.103: bytes=32 seq=2 ttl=128 time=62 ms
From 192.168.1.103: bytes=32 seq=3 ttl=128 time=63 ms
From 192.168.1.103: bytes=32 seq=4 ttl=128 time=62 ms
From 192.168.1.103: bytes=32 seq=5 ttl=128 time=16 ms

--- 192.168.1.103 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 16/53/63 ms

PC>ping 192.168.1.1.04
host 192.168.1.1.04 unreachable

PC>ping 192.168.1.1.104
host 192.168.1.1.104 unreachable

PC>ping 192.168.1.104

Ping 192.168.1.104: 32 data bytes, Press Ctrl_C to break
From 192.168.1.104: bytes=32 seq=1 ttl=128 time=62 ms
From 192.168.1.104: bytes=32 seq=2 ttl=128 time=63 ms
From 192.168.1.104: bytes=32 seq=3 ttl=128 time=62 ms
From 192.168.1.104: bytes=32 seq=4 ttl=128 time=63 ms
From 192.168.1.104: bytes=32 seq=5 ttl=128 time=78 ms

--- 192.168.1.104 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 62/65/78 ms
```

5、配置交换机LSW1，构建vlan2和vlan3

- 进入命令行输入 system-view命令进入系统试图

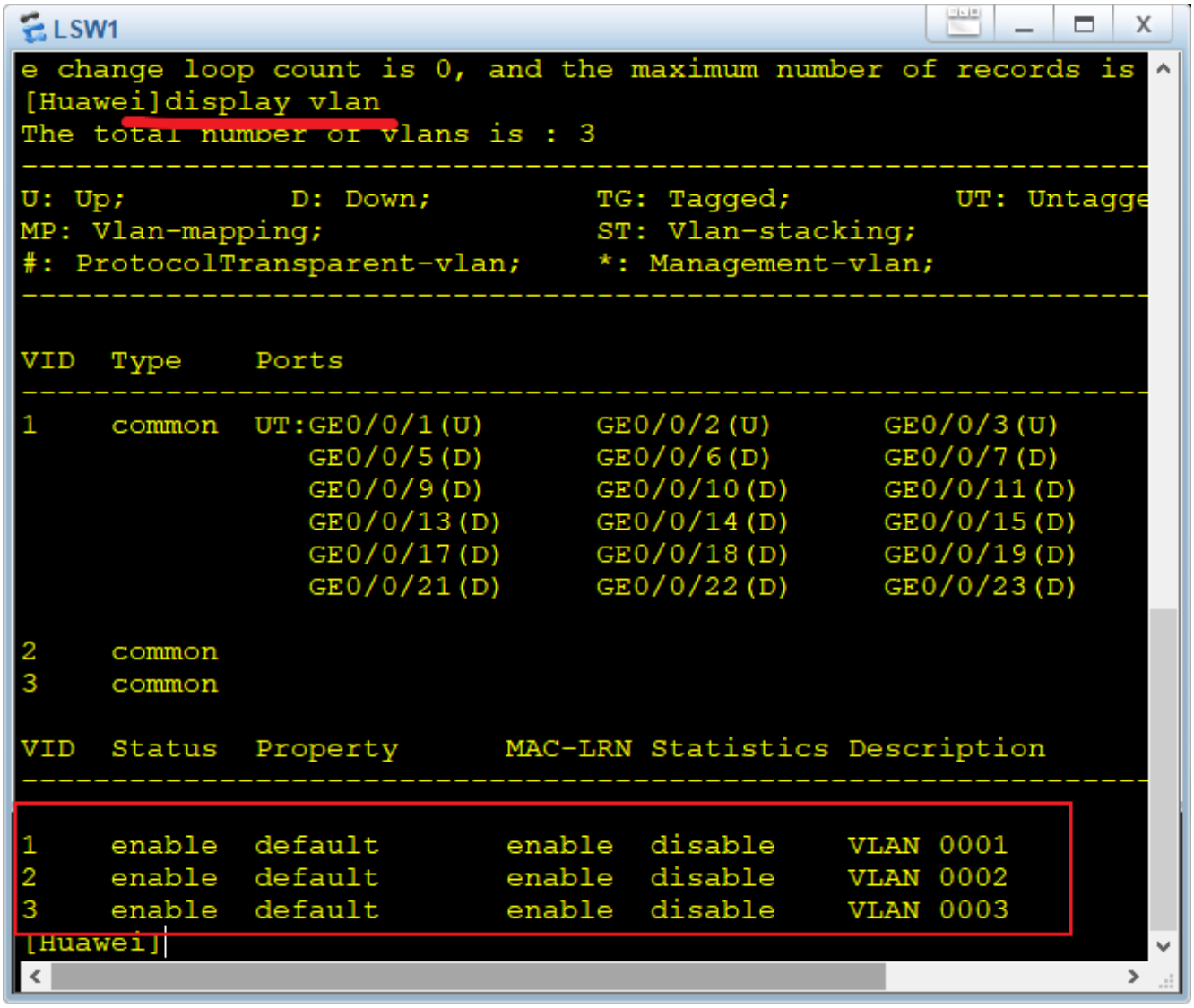
- 输入 display vlan 查看交换机中的vlan



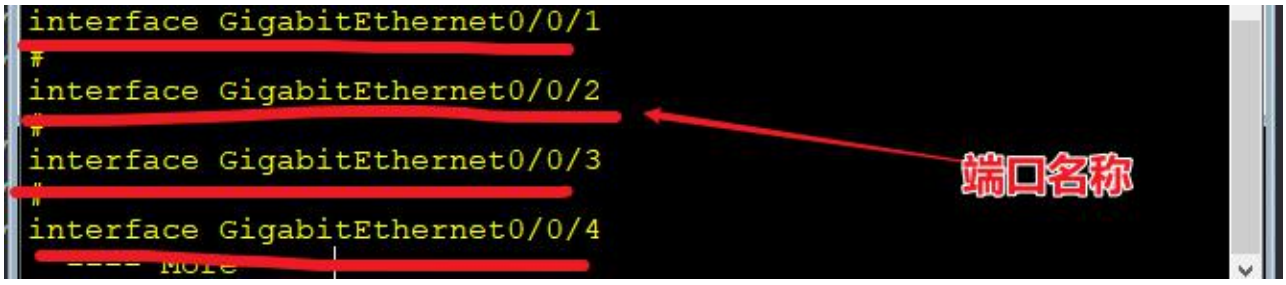
- 通过命令行，创建vlan2

- 通过命令行，创建vlan3

- 输入命令 display vlan 查看交换机中的vlan，发现有三个vlan

- 

- 将0/0/1和0/0/2端口设置为access类型：通过access端口的数据包都是不带VLAN tag的，且只属于一个VLAN，在access端口进方向，交换机接收到数据包后，先判断是否带VLAN tag，有则丢弃数据包，没有则打上该端口已配置的VLAN tag，在access端口出方向，交换机将打了与端口相同VLAN tag的数据包转发出去，并且去掉VLAN tag变成普通数据包，一般连接计算机
  - 输入命令 dis cur 查看端口名称，注意：按回车键可以显示更多信息，按 ctrl + c可以退出该命令



- 设置access类型，并且将端口分配给不同的VLAN

```
[Huawei]interface GigabitEthernet0/0/1                    设置端口0/0/1
[Huawei-GigabitEthernet0/0/1]port link-type access        端口设置为access模式
[Huawei-GigabitEthernet0/0/1]
Dec  9 2021 14:49:51-08:00 Huawei DS/4/DATASYNC_CFGCHANGE:OID 1
.25.191.3.1 configurations have been changed. The current chang
e change loop count is 0, and the maximum number of records is

[Huawei-GigabitEthernet0/0/1]port default vlan 2          端口分配给vlan2
[Huawei-GigabitEthernet0/0/1]
Dec  9 2021 14:50:01-08:00 Huawei DS/4/DATASYNC_CFGCHANGE:OID 1
.25.191.3.1 configurations have been changed. The current chang
e change loop count is 0, and the maximum number of records is
[Huawei-GigabitEthernet0/0/1]quit
[Huawei]interface GigabitEthernet0/0/2                    设置端口0/0/2
[Huawei-GigabitEthernet0/0/2]port link-type access        端口设置为access模式
[Huawei-GigabitEthernet0/0/2]port default vlan 3
[Huawei-GigabitEthernet0/0/2]quit                         端口分配给vlan 3
[Huawei]
```

  - 因为GE 0/0/1端口接PC1，并且GE 0/0/1端口分配给了vlan2，所以PC1就在vlan 2中了，同理，PC3在vlan3中，此时如果我们在PC1的命令行中ping PC2是不能ping通了，因为他们已经不在同一个局域网中了。

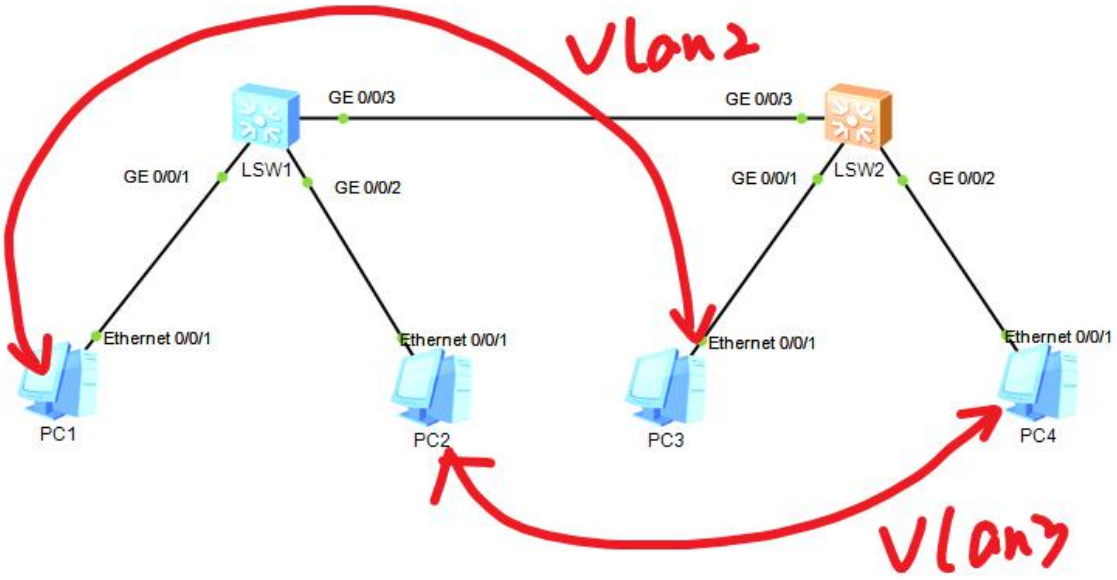
```
PC>ping 192.168.1.102

Ping 192.168.1.102: 32 data bytes, Press Ctrl_C to break
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
```

- 将0/0/3端口设置为trunk类型：端口可以承载多个VLAN，通过trunk端口的数据包都必须带上VLAN tag，在trunk端口进方向，交换机接收到数据包后，先判断是否带VLAN tag，没有则丢弃数据包，有则按照对应VLAN进行转发，在trunk端口出方向，交换机将带VLAN tag的数据包原封不动转发出去，没有带VLAN tag数据包不会从trunk端口转发出去，一般用来接其他交换机

```
[Huawei]interface GigabitEthernet0/0/3                    设置0/0/3端口
[Huawei-GigabitEthernet0/0/3]port link-type trunk         端口设置为trunk模式
[Huawei-GigabitEthernet0/0/3]
Dec  9 2021 15:16:08-08:00 Huawei DS/4/DATASYNC_CFGCHANGE:OID 1
.25.191.3.1 configurations have been changed. The current chang
e change loop count is 0, and the maximum number of records is
llow-pass vlan 2
[Huawei-GigabitEthernet0/0/3]port trunk allow-pass vlan 2  允许vlan 2 的数据通过
[Huawei-GigabitEthernet0/0/3]
Dec  9 2021 15:16:18-08:00 Huawei DS/4/DATASYNC_CFGCHANGE:OID 1
.25.191.3.1 configurations have been changed. The current chang
e change loop count is 0, and the maximum number of records is

[Huawei-GigabitEthernet0/0/3]port trunk allow-pass vlan 3  允许vlan3的数据通过
[Huawei-GigabitEthernet0/0/3]quit
```

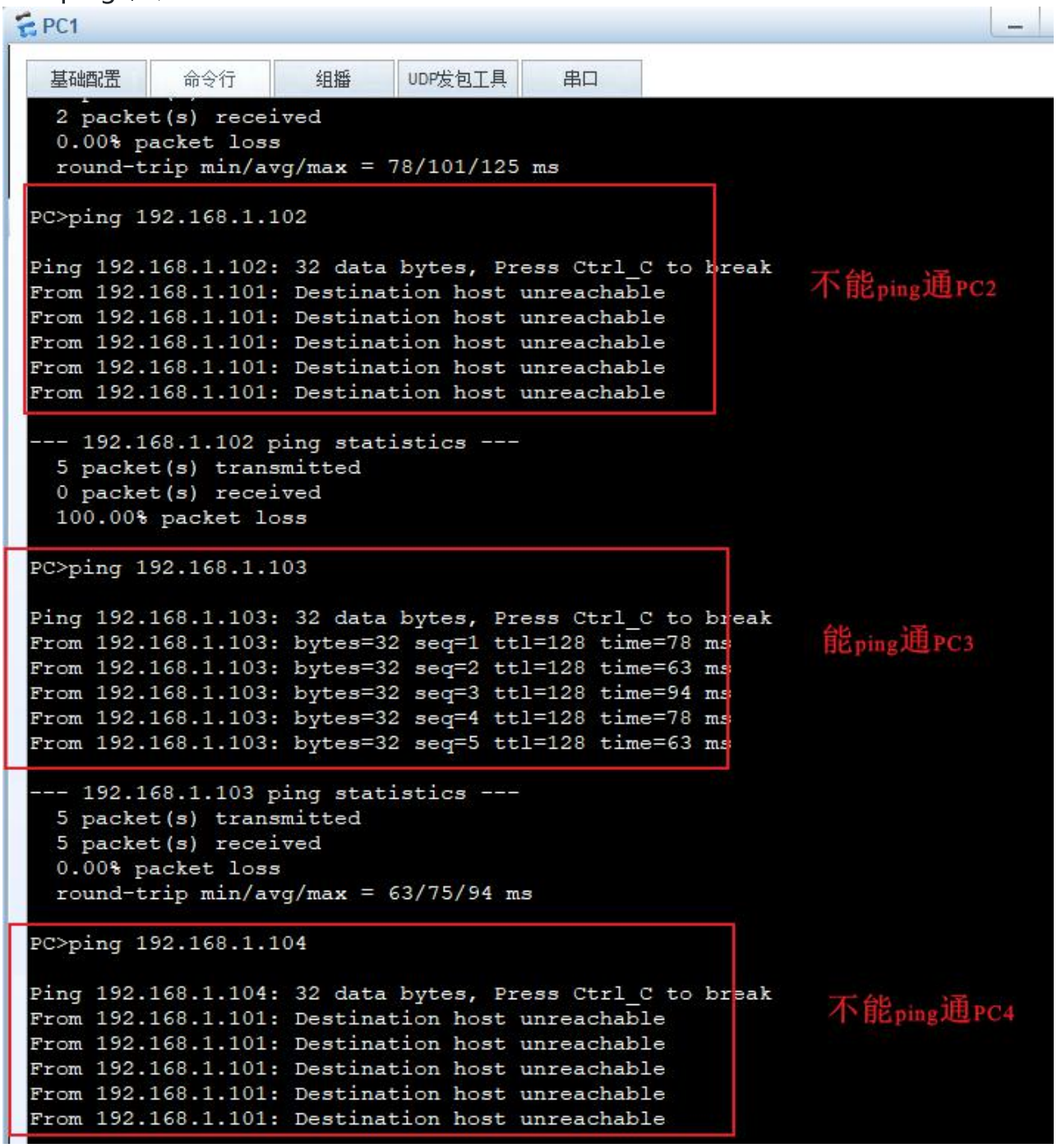
- 设置完交换机LSW1后，用同样的方法设置交换机LSW2，同样的将0/0/1端口分配给vlan2，将0/0/2端口分配给vlan3，0/0/3端口设置为trunk模式。

- 设置完成以后，PC1和PC3属于vlan2， PC2和PC4属于vlan3

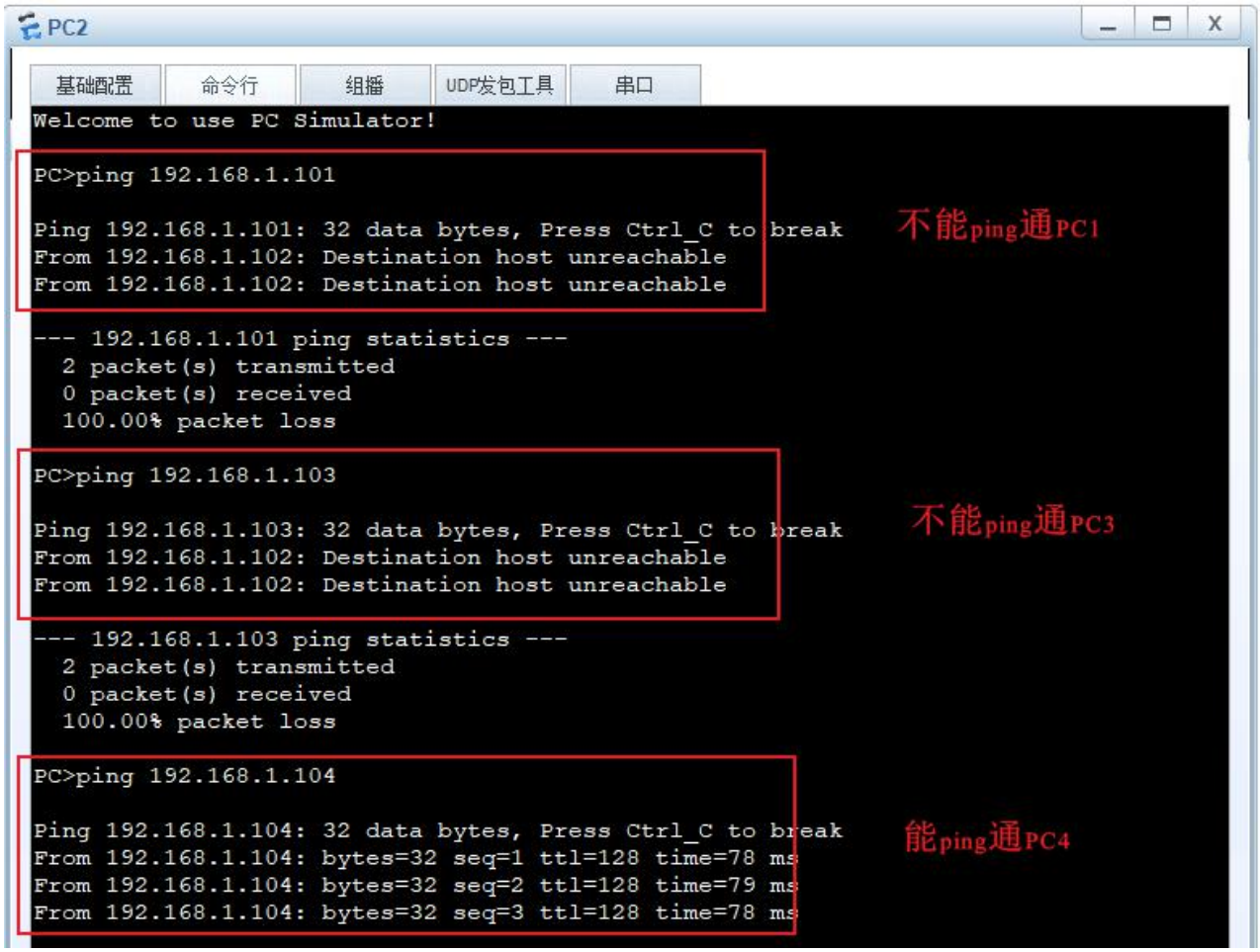


- 我们可以在PC1和PC2上使用ping命令验证

- PC1 ping命令


PC1
基础配置
命令行
组播
UDP发包工具
串口
2 packet(s) received
0.00% packet loss
round-trip min/avg/max = 78/101/125 ms
PC>ping 192.168.1.102
Ping 192.168.1.102: 32 data bytes, Press Ctrl_C to break
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
--- 192.168.1.102 ping statistics ---
5 packet(s) transmitted
0 packet(s) received
100.00% packet loss
不能ping通PC2
PC>ping 192.168.1.103
Ping 192.168.1.103: 32 data bytes, Press Ctrl_C to break
From 192.168.1.103: bytes=32 seq=1 ttl=128 time=78 ms
From 192.168.1.103: bytes=32 seq=2 ttl=128 time=63 ms
From 192.168.1.103: bytes=32 seq=3 ttl=128 time=94 ms
From 192.168.1.103: bytes=32 seq=4 ttl=128 time=78 ms
From 192.168.1.103: bytes=32 seq=5 ttl=128 time=63 ms
--- 192.168.1.103 ping statistics ---
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 63/75/94 ms
能ping通PC3
PC>ping 192.168.1.104
Ping 192.168.1.104: 32 data bytes, Press Ctrl_C to break
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
From 192.168.1.101: Destination host unreachable
不能ping通PC4

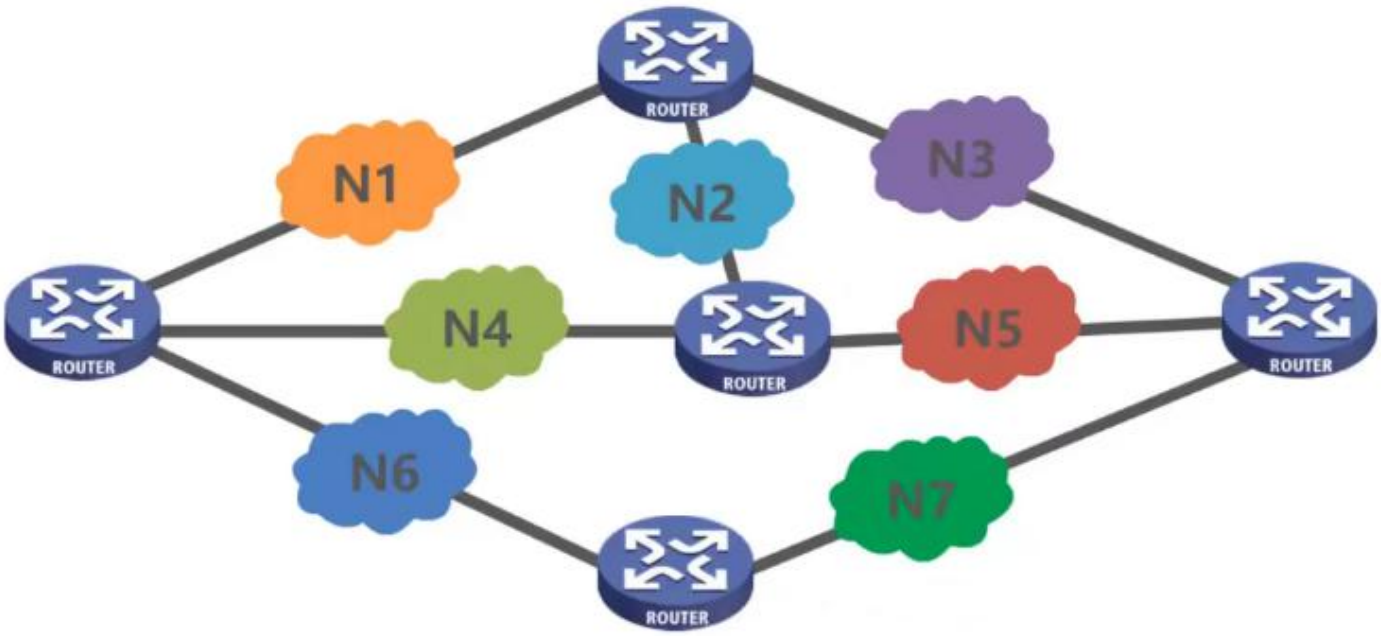
- PC2 ping 命令



2.4 网络层

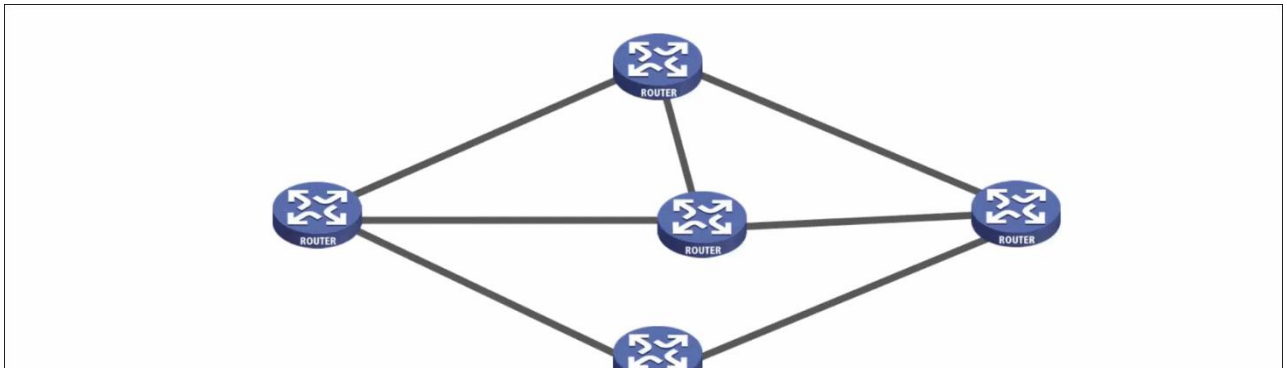
2.4.1 网络层概述

网络层的主要任务是实现**网络互连**，进而**实现数据包在各网络之间的传输。**

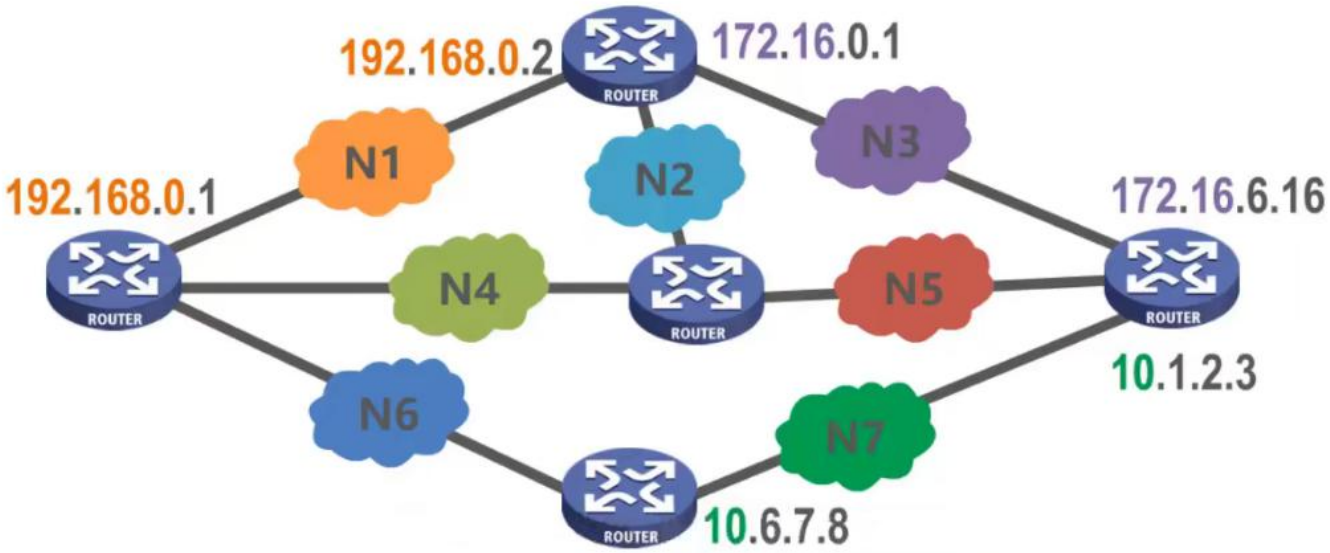


要实现网络层任务，需要解决以下主要问题：

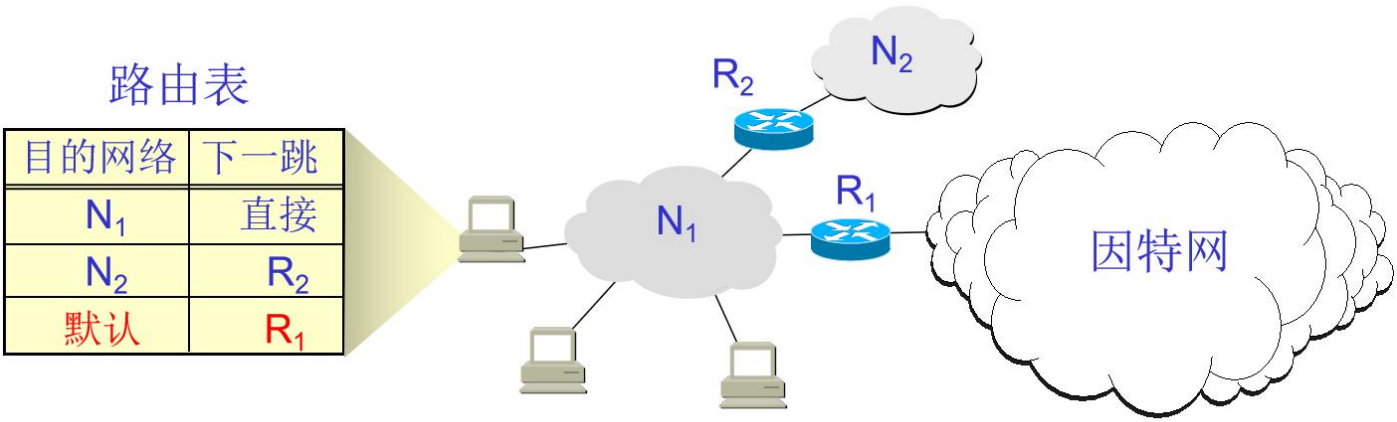
- **网络层向运输层提供怎样的服务（“可靠传输”还是“不可靠传输”）**
  - TCP/IP协议体系结构的网际层提供的是无连接的、不可靠的数据包服务
  - ATM、帧中继和X.25的网络层提供的是面向连接的、可靠的虚电路服务



- **网络层寻址问题**



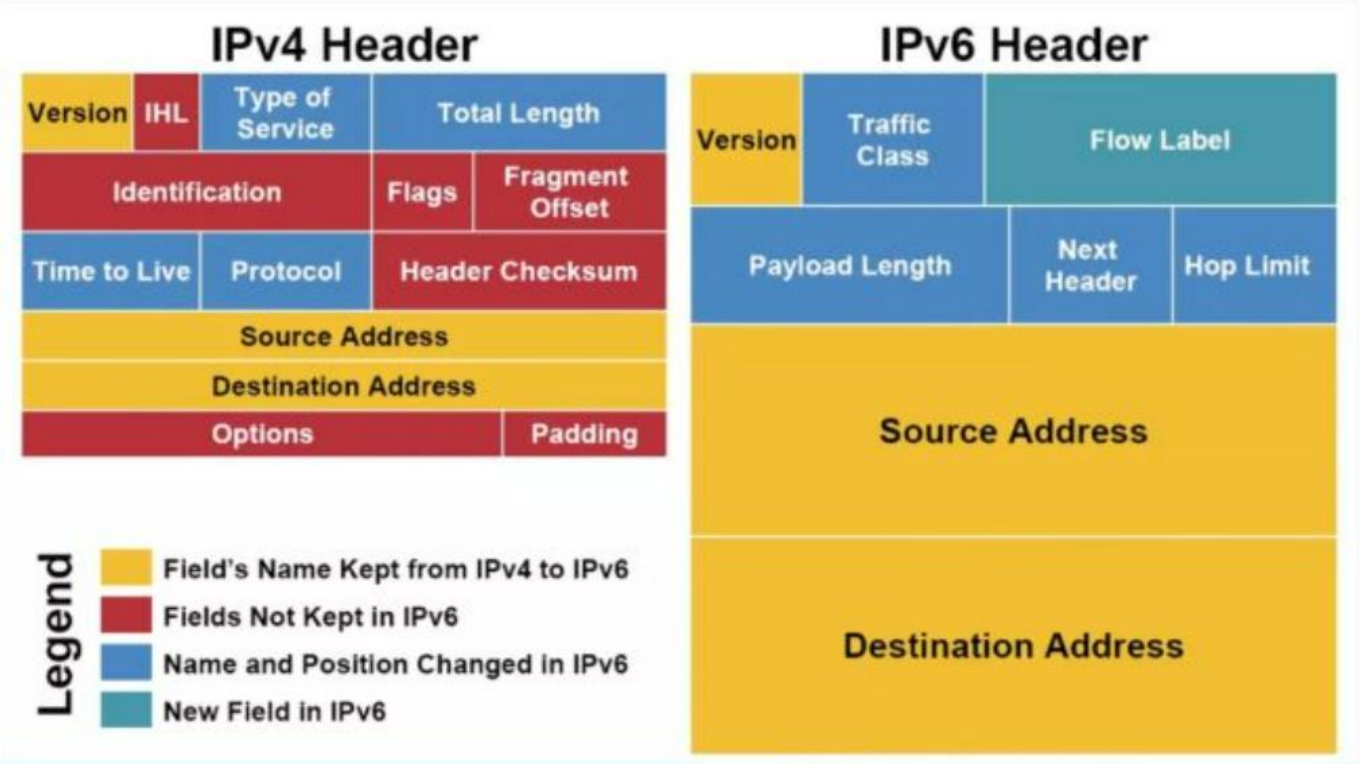
- **路由选择问题**：路由器根据路由表选择下一跳



因特网(Internet)是目前全世界用户数量最多的互联网，它使用TCP/IP协议栈。由于TCP/IP协议栈的网络层使用**网际协议IP**，它是整个协议栈的核心协议，因此在TCP/IP协议栈中网络层常称为**网际层。**本章节我们通过学习TCP/IP协议栈的网际层来学习网络层的理论知识和实践技术。

2.4.2 IPv4地址概述

1、在因特网中，为了实现计算机之间的相互通信，通常需要为每台计算机分配一个IP地址。在互联网的发展过程中主要有两个版本的互联网协议，分别是**IPv4**（Internet Protocol version 4）和**IPv6**（Internet Protocol version 6）。



> **注意：在本课程中我们不对IPv4和IPv6的协议内容进行对比，只对比IPv4的IP地址和IPv6的IP地址。**

2、IPv4 的IP地址就是给因特网(Internet)上的每一台主机(或路由器)）的每一个接口分配一个在全世界范围内是唯一的**32比特**的标识符，地址总数为**2^32**个。

3、IPv6 的 IP地址采用**128比特**地址长度，地址总数为**2^128**个。

4、IP地址由**因特网名字和数字分配机构ICANN**(Internet Corporation for Assigned Names and Numbers)进行分配。我国用户可向**亚太网络信息中心APNIC**(Asia Pacific Network Information Center)申请IP地址，需要**缴费**。

5、因为当下互联网中的绝大多数设备采用的是IPv4的IP地址，所以本课程我们只学习IPv4的IP地址的相关知识。

6、由于32比特的IPv4地址不方便阅读、记录以及输入等，因此IPv4地址采用**点分十进制**表示方法以方便用户使用，例如：192.168.1.100

【练习】请将以下这些32比特的IPv4地址转换为点分十进制形式。

(1) 00001010 11111110 00001111 11110000

(2) 10101100 00010000 10111111 11110111

(3) 11000000 10101000 10100101 00000111

【解析】

(1) 10.254.15.240

(2) 172.16.191.247

(3) 192.168.165.7

7、八位无符号二进制整数转十进制：二进制表示中从右至左分别表示低比特位和高比特位。

$$(b_7b_6b_5b_4b_3b_2b_1b_0)_2 = (b_7\times2^7+b_6\times2^6+b_5\times2^5+b_4\times2^4+b_3\times2^3+b_2\times2^2+b_1\times2^1+b_0\times2^0)_{10}$$

8位二进制数的每个位的权值： 128 64 32 16 8 4 2 1

【举例】

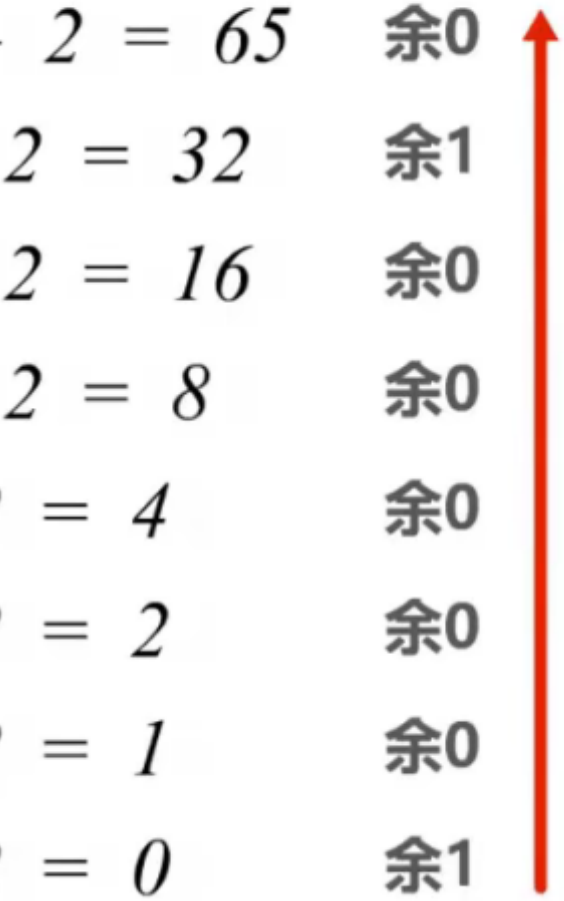
$$\begin{aligned}(10101010)_2 &= (1\times2^7 + 0\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0)_{10}\\ &= (1\times128 + 0\times64 + 1\times32 + 0\times16 + 1\times8 + 0\times4 + 1\times2 + 0\times1)_{10}\\ &= (170)_{10}\end{aligned}$$

8、十进制正整数转8位无符号二进制数的方法：

- **方法1：除2取余法，直到最后商为0**

$$(130)_{10} = (10000010)_2$$

$$130 \div 2 = 65 \quad 余0$$
$$65 \div 2 = 32 \quad 余1$$
$$32 \div 2 = 16 \quad 余0$$
$$16 \div 2 = 8 \quad 余0$$
$$8 \div 2 = 4 \quad 余0$$
$$4 \div 2 = 2 \quad 余0$$
$$2 \div 2 = 1 \quad 余0$$
$$1 \div 2 = 0 \quad 余1$$



- **方法2：凑值法**(必须熟记8位二进制数各位的权值**128 64 32 16 8 4 2 1**)

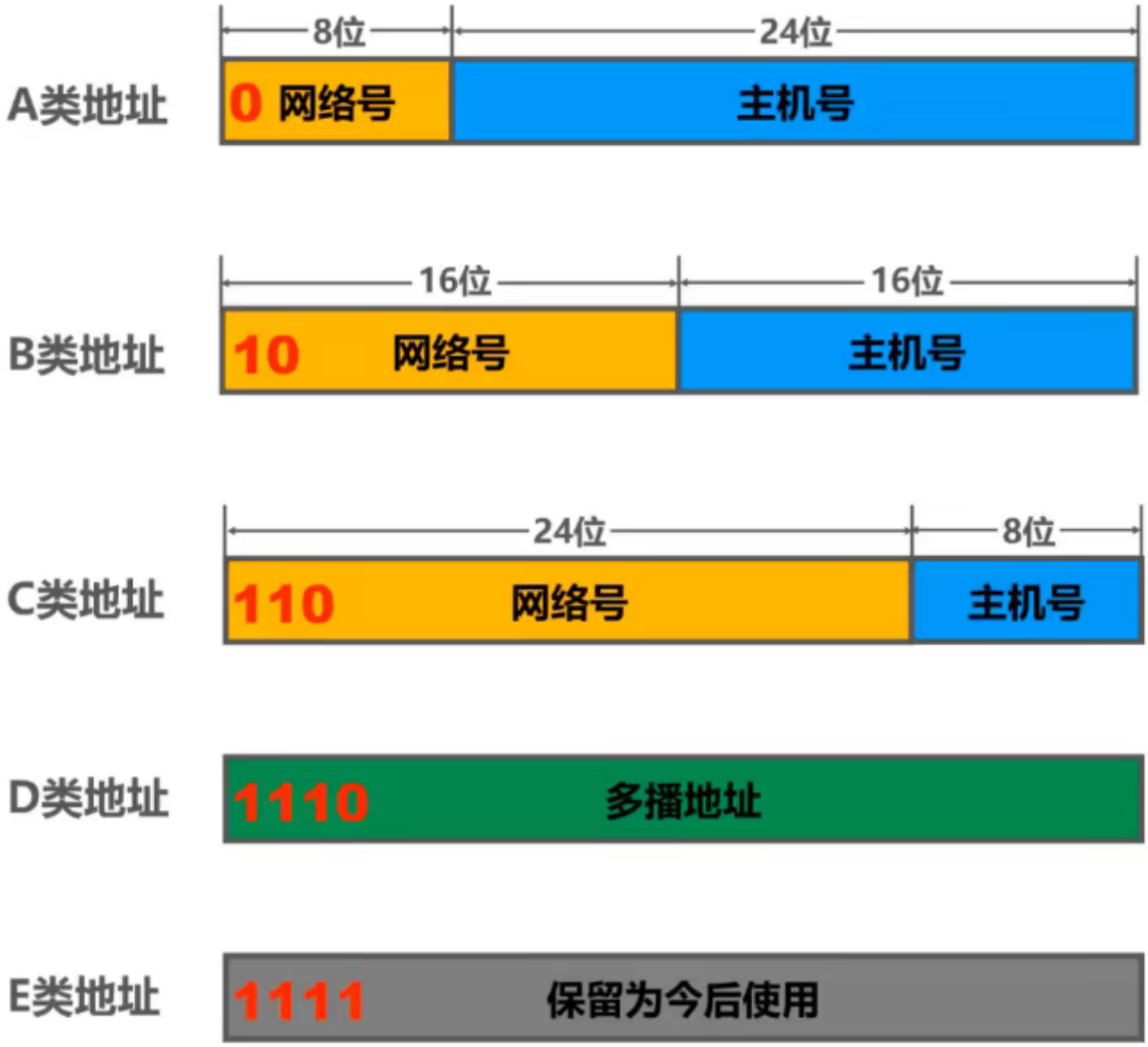
$$(171)_{10} = (10101011)_2$$

$$= (1\times128 + 0\times64 + 1\times32 + 0\times16 + 1\times8 + 0\times4 + 1\times2 + 1\times1)_{10}$$

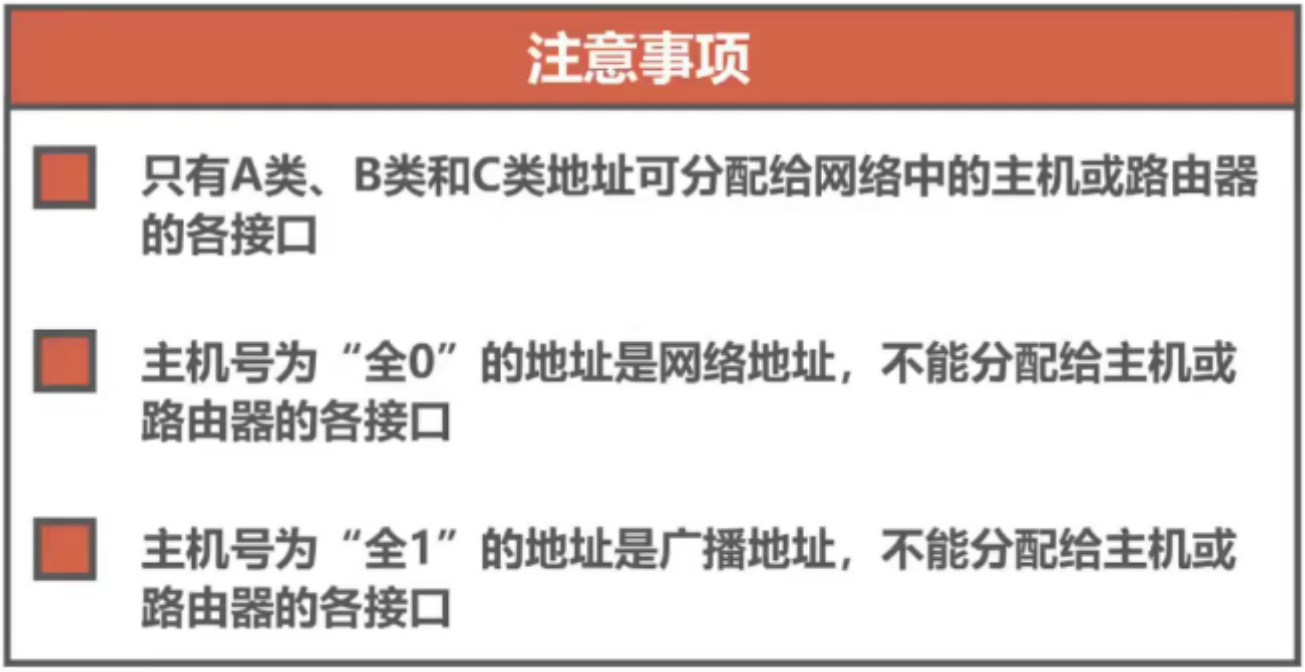
$$b_7 \quad b_6 \quad b_5 \quad b_4 \quad b_3 \quad b_2 \quad b_1 \quad b_0$$

2.4.3 IPv4 地址的分类

1、IPv4 的IP地址由**网络号**（指定主机所属的网络）和 **主机号**（指定被寻址的子网中的某个节点）组成，IP地址可以分为A、B、C、D、E五类。


8位
24位
A类地址
0 网络号
主机号
16位
16位
B类地址
10
网络号
主机号
24位
8位
C类地址
110
网络号
主机号
D类地址
1110
多播地址
E类地址
1111
保留为今后使用

2、注意事项：


注意事项
只有A类、B类和C类地址可分配给网络中的主机或路由器的各接口
主机号为“全0”的地址是网络地址，不能分配给主机或路由器的各接口
主机号为“全1”的地址是广播地址，不能分配给主机或路由器的各接口

3、各类IP地址的细节

- **A类地址**
  - A类地址的取值范围：

  - A类地址可指派的网络数量：因为网络号占8个bit，并且最高位固定为0，网络号有2^7=128中组合，但是因为最小网络号0和最大网络号127不能指派，所以最终A类IP地址能够指派的网络数量为：

$$2^{(8-1)} - 2 = 126$$ **（减2的原因是除去最小网络号0和最大网络号127）**

  - A类地址每个网络中可以分配的地址数量：因为主机号站24个bit，主机号有2^24种组合，因为全0的网络号和全1的广播地址不能指派，所以需要减2

$$2^{24} - 2 = 16777214$$ **（减2的原因是除去主机号为全0的网络地址和全1的广播地址）**

- **B类地址**
  - B类地址的取值范围：

  - B类地址可指派的网络数量：因为网络号占16个bit，并且最高位固定为10，网络号有2^(16-2)种组合

$$2^{(16-2)} = 16384$$

  - B类地址每个网络中可以分配的地址数量：因为主机号站16个bit，主机号有2^16种组合，因为全0的网络号和全1的广播地址不能指派，所以需要减2

$$2^{16} - 2 = 65534$$ **（减2的原因是除去主机号为全0的网络地址和全1的广播地址）**

- **C类地址**

- C类地址的取值范围

最小网络号也是第一个可指派的网络号192.0.0
网络地址为192.0.0.0

最大网络号也是最后一个可指派的网络号223.255.255
网络地址为223.255.255.0

- C类地址可指派的网络数量：因为网络号占24个bit，并且最高位固定为110，网络号有

$$2^{(24-3)} = 2097152$$

2^(24-3)

- C类地址每个网络中可以分配的地址数量：因为主机号站8个bit，主机号有2^8种组合，因为全0的网络号和全1的广播地址不能指派，所以需要减2

$$2^8 - 2 = 254$$ **（减2的原因是除去主机号为全0的网络地址和全1的广播地址）**

- **D类、E类地址**

网络类别	作 用	第一个地址	最后一个地址	地址数量	占总地址空间
D	多播地址	224.0.0.0	239.255.255.255	268435456 (2^{28})	1/16 ($2^{(32-4)}$ / 2^{32})
E	保留为今后使用	240.0.0.0	255.255.255.255	268435456 (2^{28})	1/16 ($2^{(32-4)}$ / 2^{32})

- 练习1

【练习】请填写以下两个表格的内容。

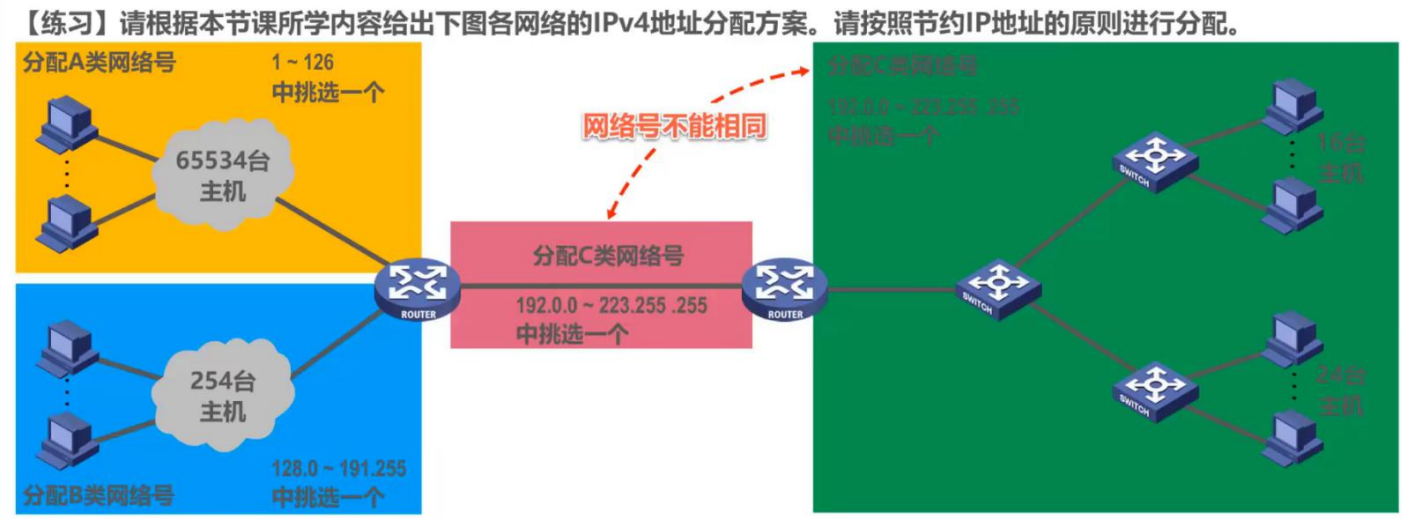
分类的IP地址	类别	是否可以指派给主机
0.1.2.3	A	否（保留的网络号）
1.2.3.4	A	是
126.255.255.255	A	否（广播地址）
127.0.0.1	A	否（本地环回测试地址）
128.0.255.255	B	否（广播地址）
166.16.18.255	B	是

分类的IP地址	类别	是否可以指派给主机
172.18.255.255	B	否（广播地址）
191.255.255.252	B	是
192.0.0.255	C	否（广播地址）
196.2.3.8	C	是
218.75.230.30	C	是
223.255.255.252	C	是

【解析】

1. 根据地址左起第一个十进制数的值，可以判断出网络类别（小于127的为A类，128~191的为B类，192~223的为C类）；
2. 根据网络类别，就可找出地址中的网络号部分和主机号部分（A类地址网络号为左起第一个字节，B类地址网络号为左起前两个字节，C类地址网络号为左起前三个字节）；
3. 以下三种情况的地址不能指派给主机或路由器接口：
   （1）A类网络号0和127
   （2）主机号为“全0”，这是网络地址
   （3）主机号为“全1”，这是广播地址

- 练习2



2.4.4 子网的划分

1、我们先来看一道笔试题：用一根网线直接相连的两台主机的IP地址分别为192.168.1.100、192.168.2.100，子网掩码都为 255.255.255.0，请问使用什么方法可以让这两台主机能够进行正常的数据通信?

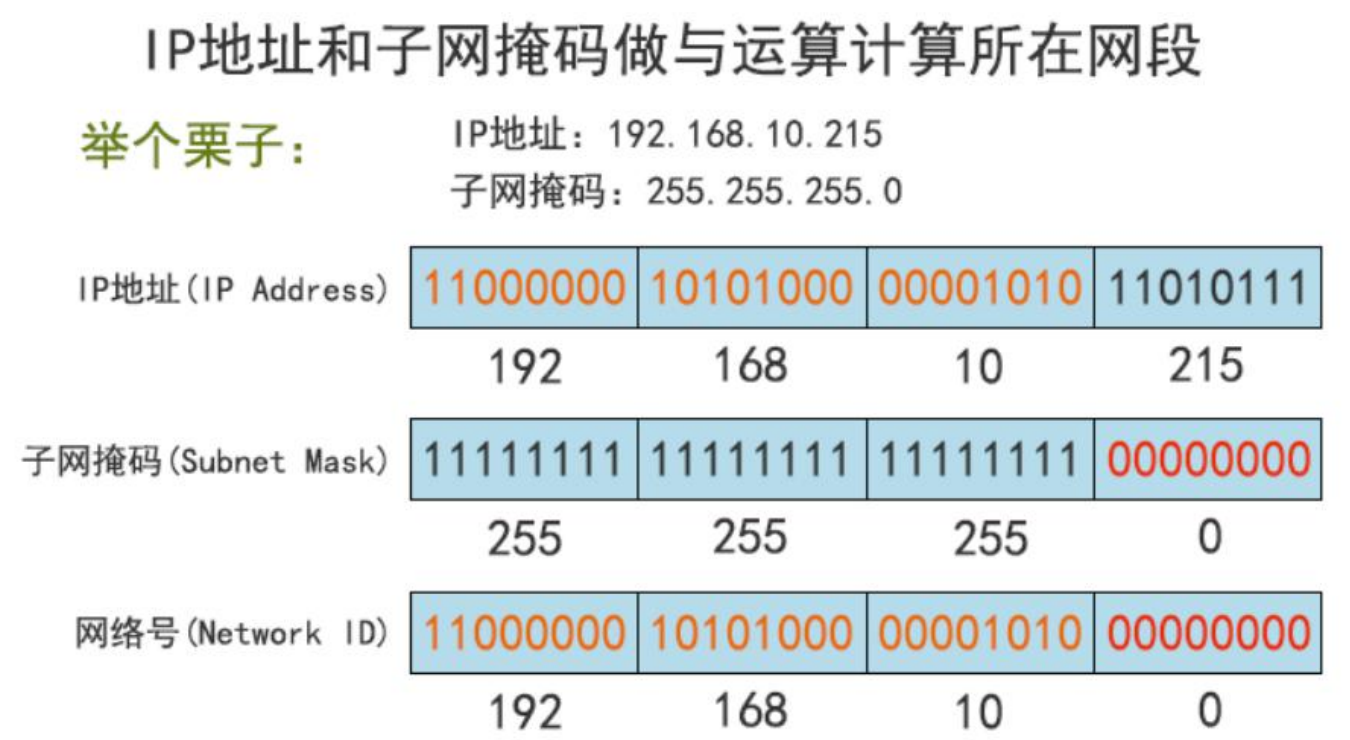
答案：**将两台主机的子网掩码设置为255.255.0.0**

2、子网掩码

- 子网掩码(subnet mask)是一个**32位**地址，又叫网络掩码、地址掩码，它用来**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。**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**将某个IP地址划分成网络地址和主机地址两部分。**
- 左边是网络位，用二进制数字“1”表示，1的数目等于网络位的长度；右边是主机位，用二进制数字“0”表示，0的数目等于主机位的长度。这样做的目的是为了让掩码与IP地址做按位与运算时用0遮住原主机数，而不改变原网络段数字，而且很容易通过0的位数确定子网的主机数。

类别	子网掩码的二进制数值	子网掩码的十进制数值
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

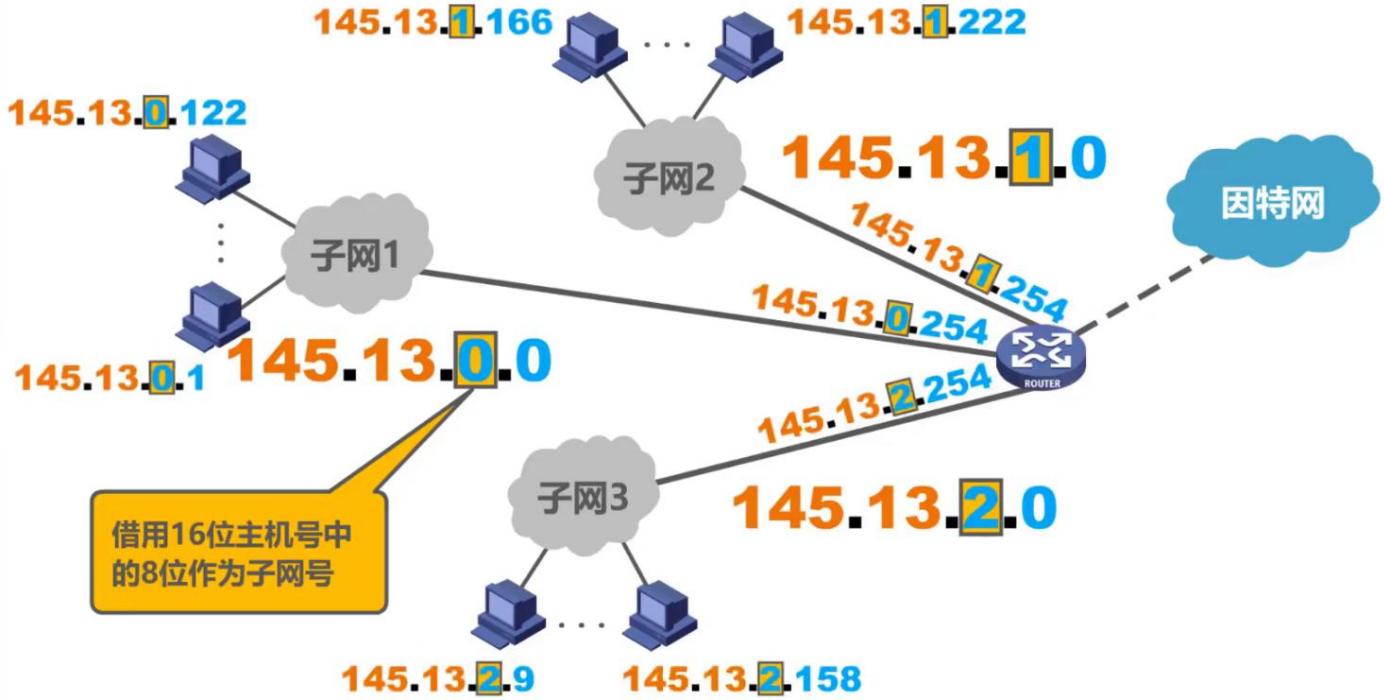
- 子网掩码的两个功能：
  - **屏蔽IP地址的一部分以区别网络标识和主机标识**
  - **将一个大的IP网络划分为若干小的子网络**
- 子网掩码的工作流程：将32位的子网掩码与IP地址进行二进制形式的按位逻辑“与”运算，得到该IPv4地址所在的网络地址



3、子网的划分

子网掩码是在IPv4地址资源紧缺的背景下为了解决IP地址分配而产生的**虚拟IP技术**，通过子网掩码将A、B、C三类地址划分为**若干子网**，从而显著**提高了IP地址的分配效率**，有效解决了IP地址资源紧张的局面。另一方面，在企业内网中为了更好地管理网络，网管人员也利用子网掩码的作用，人为地**将一个较大的企业内部网络划分为更多个小规模的子网**，再利用三层交换机的路由功能实现子

网互联，从而有效**解决了网络广播风暴和网络病毒**等诸多网络管理方面的问题。



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。

下面我们举例说明划分子网的细节： 已知某个网络的地址为218.75.230.0，使用子网掩码255.255.255.128对其进行子网划分，请给出划分细节。

解析流程：

- 从IP地址坐起第一个十进制为218，所以该IP地址为C类地址，C类地址的子网掩码默认为：255.255.255.0，而此时给出的子网掩码为255.255.255.128，则说明子网掩码向主机号借用了一个bit作为子网号

- 因为子网掩码向主机号借用了一个bit作为子网号，所以可划分出来的子网数量为2^1 = 2 个。每个子网可分配的地址数量为：2^(8-1) - 2 = 126个。因为主机号的最高位被子网掩码借用了，主机号只有7位了，所以最多可分配2^(8-1)=128个地址，但是还需要去掉主机号为全0的

网络地址和主机号为全1的广播地址所以还需要减2。

划分出的子网数量 $2^1 = 2$

每个子网可分配的地址数量 $2^{(8-1)} - 2 = 126$

（减2是要去掉主机号为“全0”的网络地址和“全1”的广播地址）

- 详细的子网划分

4、练习题：我们通过一道考研真题对子网的划分方法进行巩固

【2012年 题39】某主机的IP地址为180.80.77.55，子网掩码为255.255.252.0，如该主机向其所在子网发送广播分组，则目的地址可以是 D

A. 180.80.76.0　　B. 180.80.76.255　　C. 180.80.77.255　　D. 180.80.79.255

【解析】

5、子网掩码的**CIDR斜线记法**

格式：IP地址/n

> **说明：n表示子网掩码所使用的bit数，表示子网掩码中网络号的长度，通过n的个数确定子网的主机数=2^(32-n)-2**

例1：192.168.1.100/24，其子网掩码表示为255.255.255.0

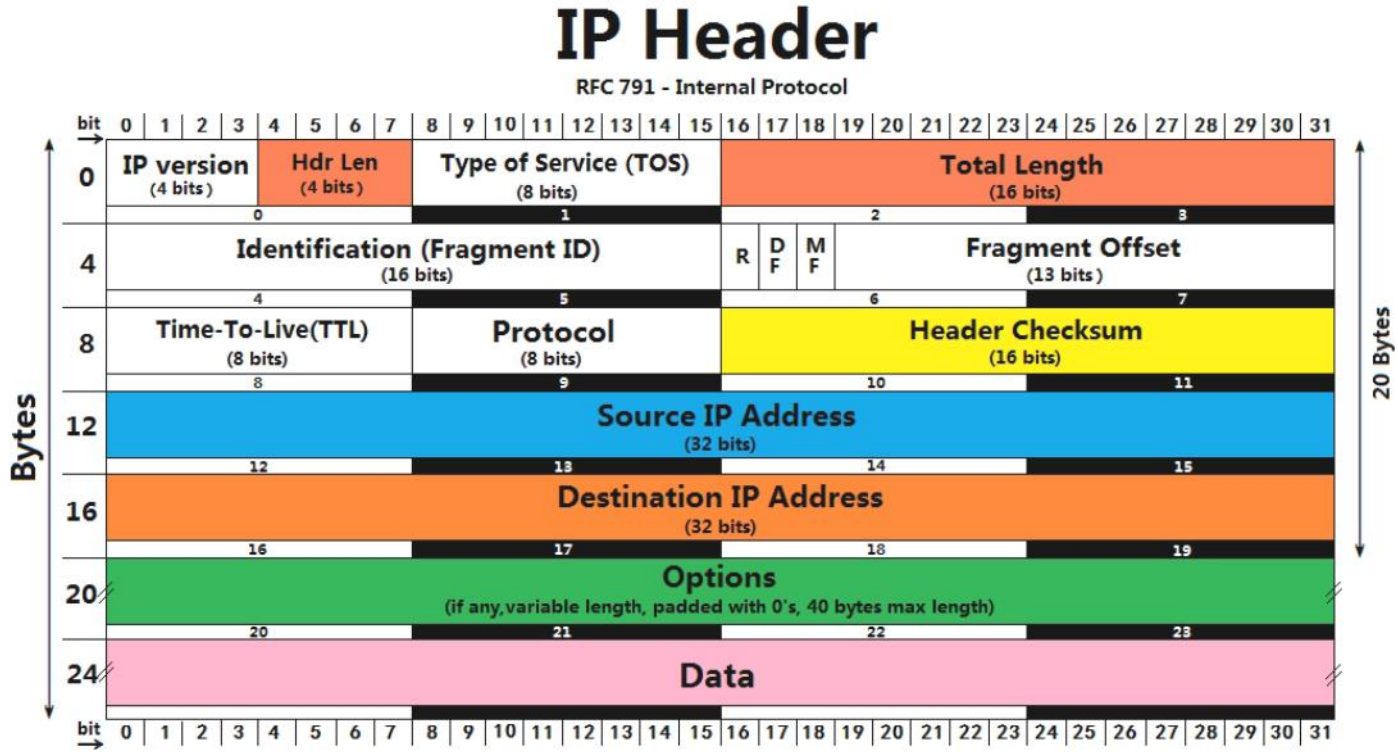
例2：172.16.198.12/20，其子网掩码表示为255.255.240.0

2.4.5 IP 协议

1、用户数据在网络层（网际层）中使用**IP协议**进行封装，然后交付给数据链路层。IP协议提供**不可靠无连接**的**数据报传输服务。**

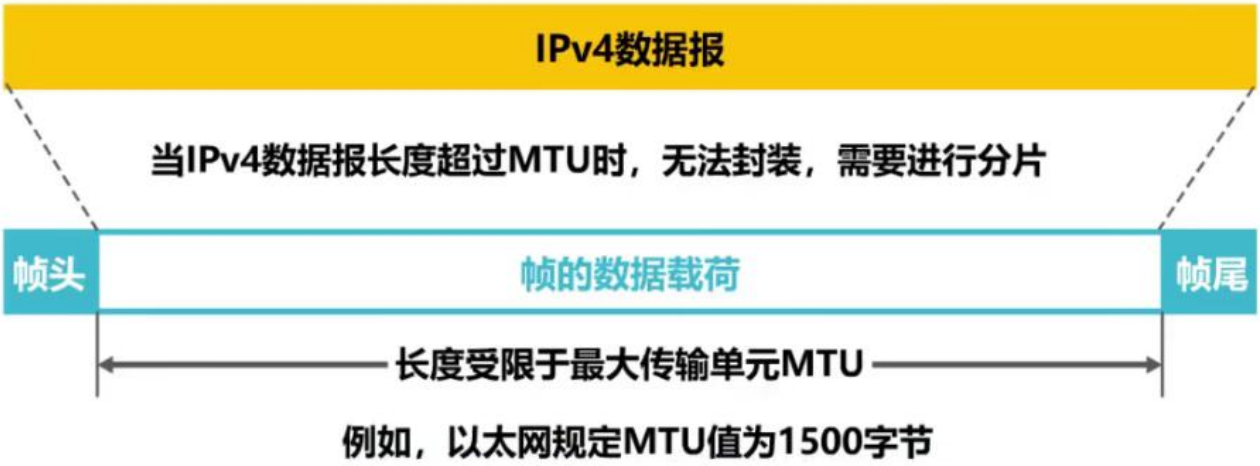
2、使用IP协议封装的数据我们称之为**IP数据报。**

3、IP数据报的首部：IP数据报的首部由20个字节组成，IP数据报的首部常以32个比特为单位进行描述，下图中的每一行都由32个比特（也就是4个字节）构成，每个小格子称为字段或者域。

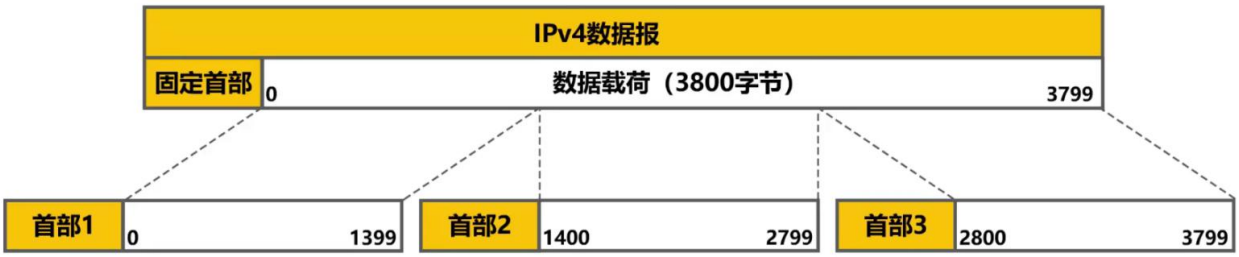


- 版本：占4比特，表示IP协议的版本
- 首部长度：占4比特，表示IP数据报首部的长度。该字段的取值以4字节为单位。最小十进制取值为5，表示IP数据报首部只有20字节固定部分，最大十进制取值为15，表示IP数据报首部包含20字节固定部分和最大40字节可变部分
- 区分服务：8比特，一般不使用
- 总长度：占16比特，表示IP数据报的总长度（首部+数据载荷)。最大取值为十进制的65535，以字节为单位（实际使用中不会传输这么长的IP数据报）。
- 标识、标志、片偏移：这三个字段共同用于IP数据报分片，以太网中数据链路层规定MTU的值为1500字节，如果某个IP数据报总长度超过MTU，需要将原IP数据报分片成多个小的数据报，

再将每个小的数据报封装成帧。



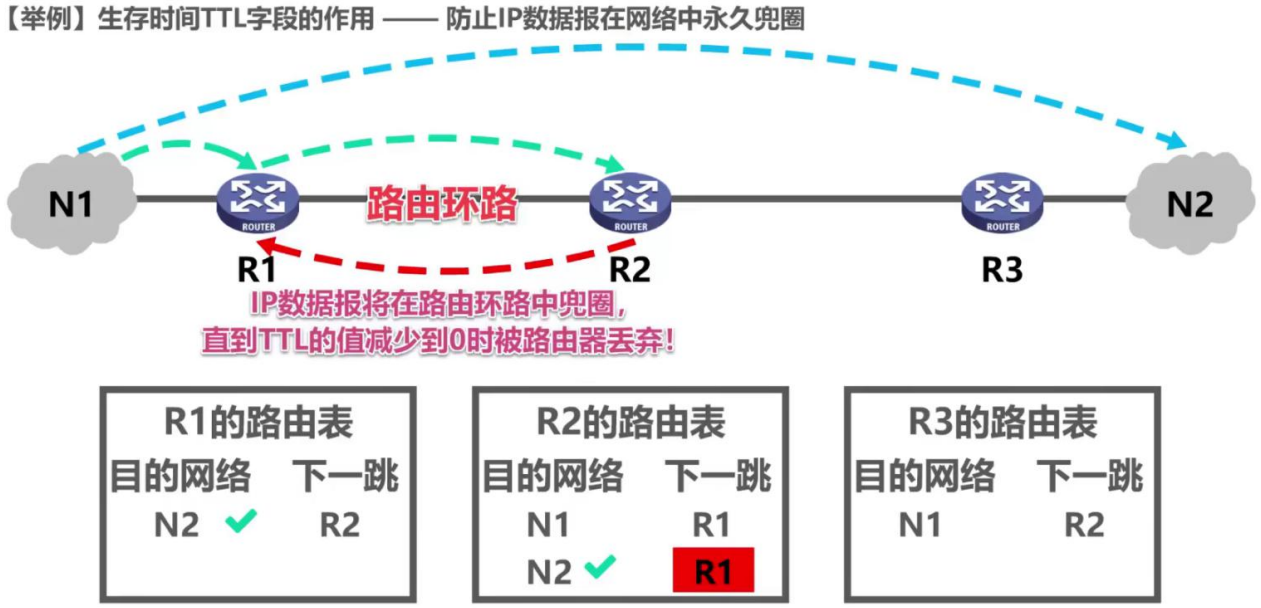
- 标识：占16比特，属于同一个数据报的各分片数据报应该具有相同的标识。IP软件维持一个计数器，每产生一个数据报，计数器值加1，并将此值赋给标识字段。
- 标志：占3比特，各比特含义如下：
  - DF位：1表示不允许分片,0表示允许分片
  - MF位：1表示“后面还有分片”，0表示“这是最后一个分片”
  - 保留位：必须为0
- 片偏移：占13比特，指出分片数据报的数据载荷部分偏移其在原数据报的位置有多少个单位，片偏移以8个字节为单位。
- IP数据报分片举例



	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的值减去IP数据报在本路由器上所耗费的时间，若不为0就转发，否则就丢弃。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的值减1，若不为0就转发，否则就丢弃。

- TTL作用可以防止IP数据报再网络中永久兜圈



- 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。

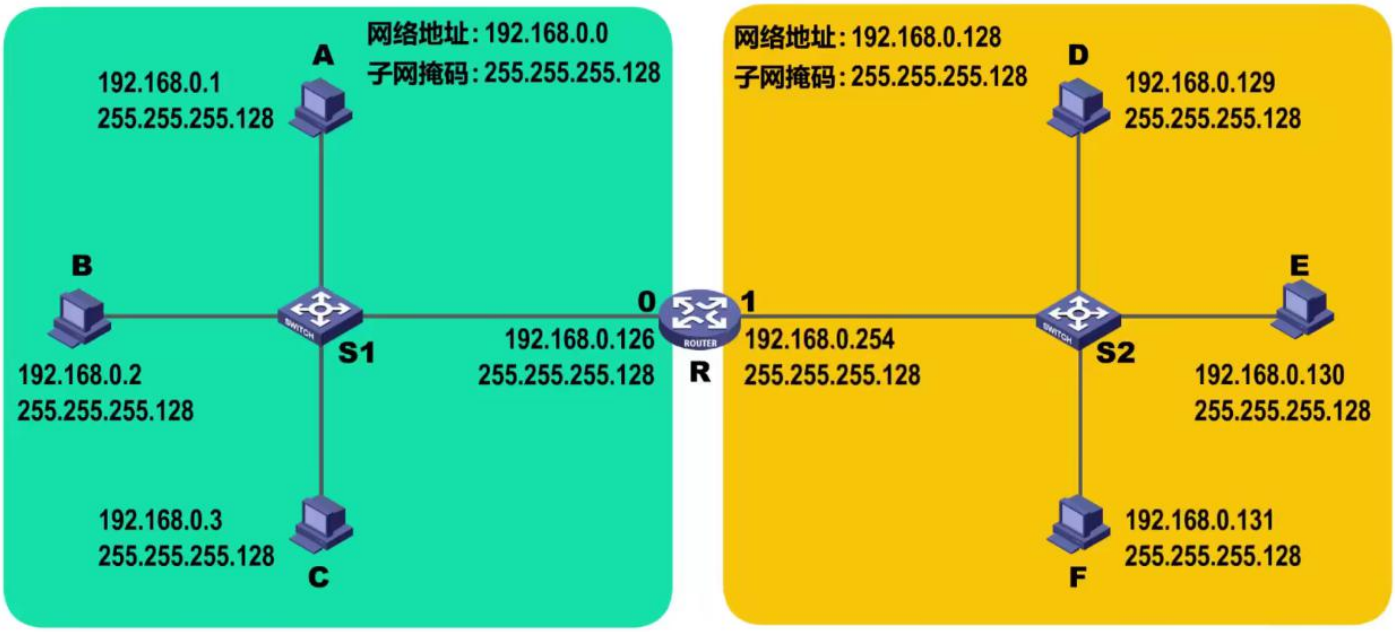
协议名称	ICMP	IGMP	TCP	UDP	IPv6	OSPF
协议字段值	1	2	6	17	41	89

- 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。
- 源IP地址：占32比特，表示发送主机的IP地址。
- 目的IP地址：占32比特，表示目的主机的IP地址。

2.4.6 IP数据报的发送和转发过程

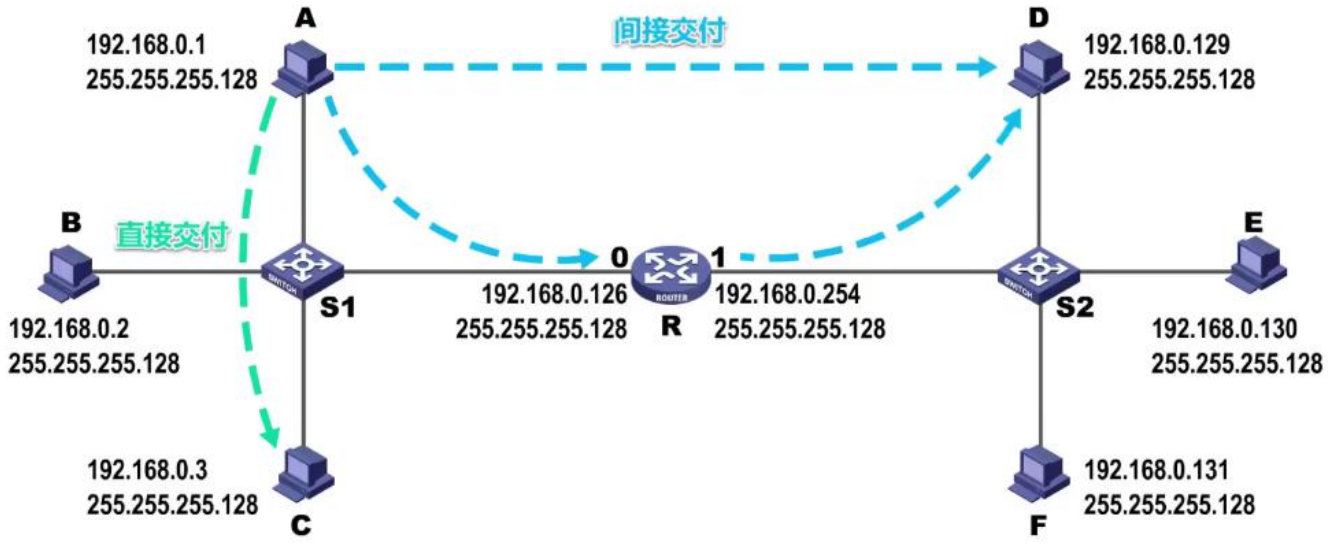
IP数据报的发送和转发过程包含两部分：**主机发送IP数据报**、**路由器转发数据报**。

假设在网络中由一台路由器连接了两个局域网，每个局域网的网络地址、子网掩码，以及每台主机的IP地址分配如下图所示：



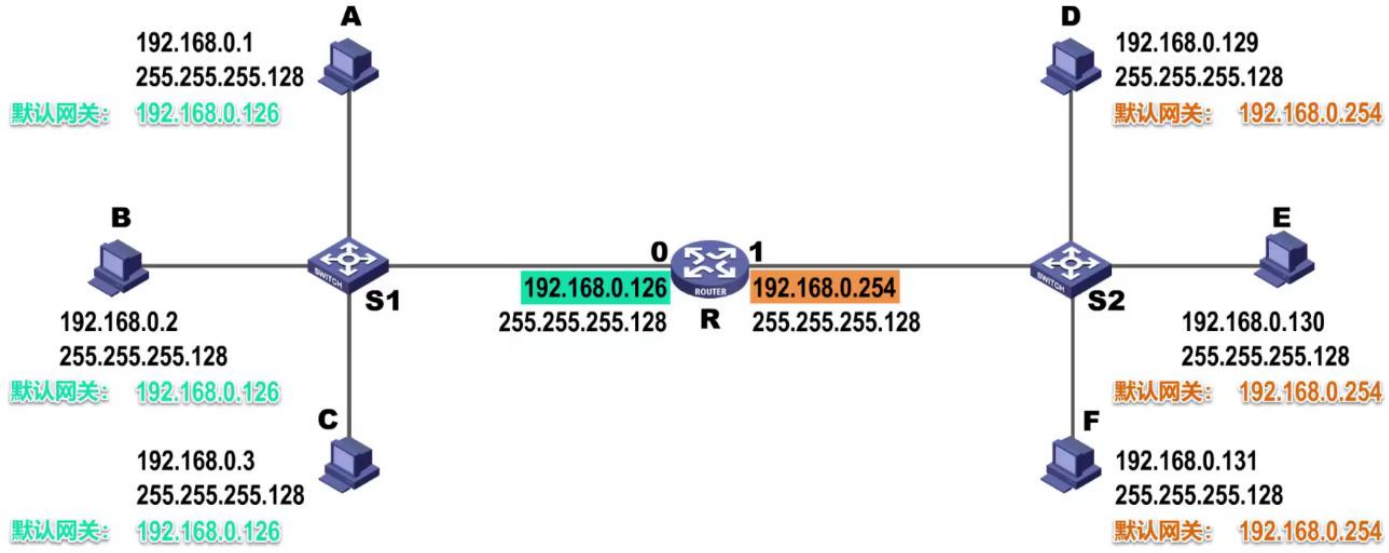
请同学们思考：为什么左边网络的网络地址是：192.168.0.0，右边网络的网络地址是192.168.0.128?

同一个网络中的主机是可以直接通信的， 这叫做**直接交付**，而不同网络中间的主机是不能直接通信的，需要通过路由器进行中转，这属于**间接交付**。那**源主机是如何判断目的主机和自己是否在同一个网络中呢?**



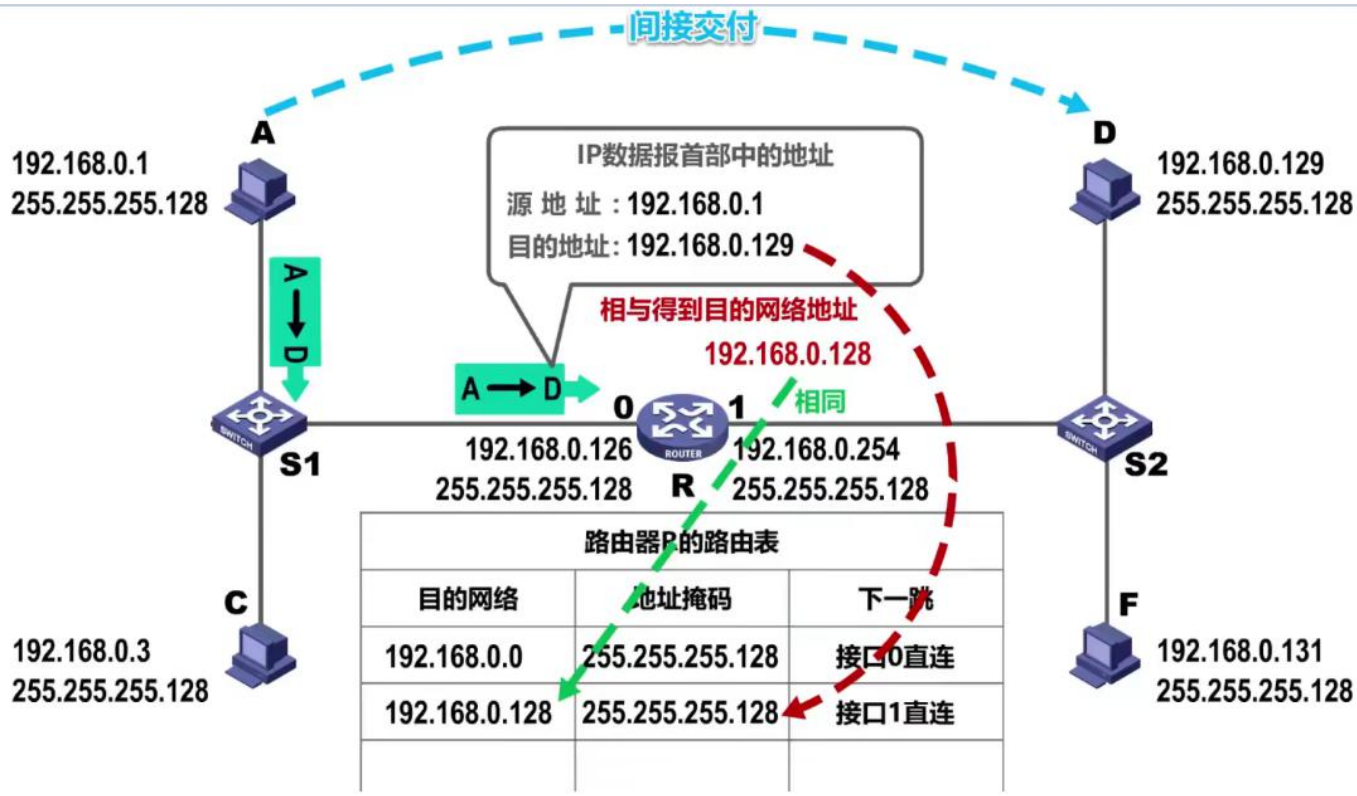
主机C将自己的IP地址与子网掩码进行按位与操作得到主机C的网络地址：192.168.0.0，然后将目的主机的IP地址与自己的子网掩码进行按位与操作得到目的网络地址：192.168.0.128，发现两个网路并不相等，说明主机C和主机F不在同一个网络中，不能进行直接交付，主机C将数据发送给路由器R，由路由器R再转发给主机F，这样就实现了主机F和主机F之间的数据通信（间接交付）。

但是问题又来了：**主机C又是如何知道要将数据发送给路由器R呢？我们需要将路由器的接口地址192.168.0.126作为主机C的默认网关**，这样当主机C发现目的地址与自己不在同一网络中时则将数据发送给设置好的默认网关（路由器）。因此我们可以将左边网络中所有主机的默认网关填写为192.168.0.126，将右边网络所有主机的默认网关填写为192.168.0.254。



那么路由器又是如何对数据进行转发的呢？假设主机A需要将数据发送给主机D，因为主机A与主机D不在同一个网络，所以主机A将数据发送给路由器R，路由器会进行如下操作：

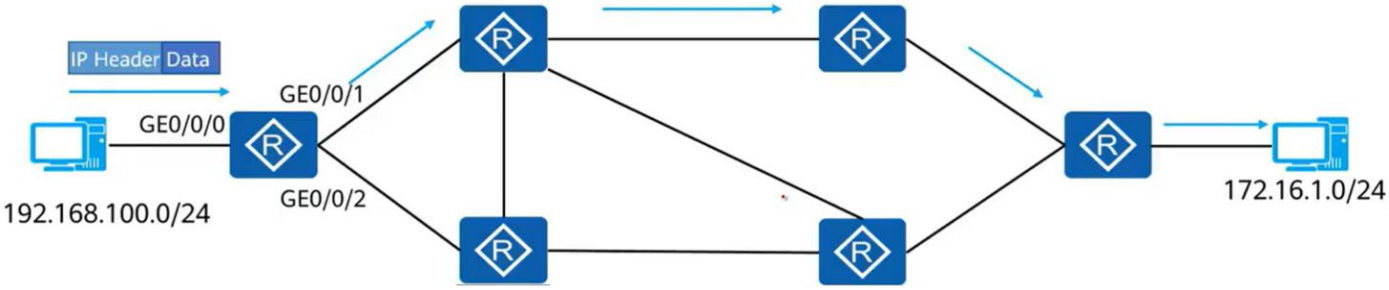
- 首先**检查IP数据报首部是否出错**：若出错，则直接丢弃该IP数据报并通告源主机，若没有出错，则进行转发
- 然后**根据IP数据报的目的地址在路由表中查找匹配的条目**：若找到匹配的条目，则转发给条目中指示的下一跳若找不到，则丢弃该IP数据报并通告源主机



2.4.7 路由表

1、什么是路由?

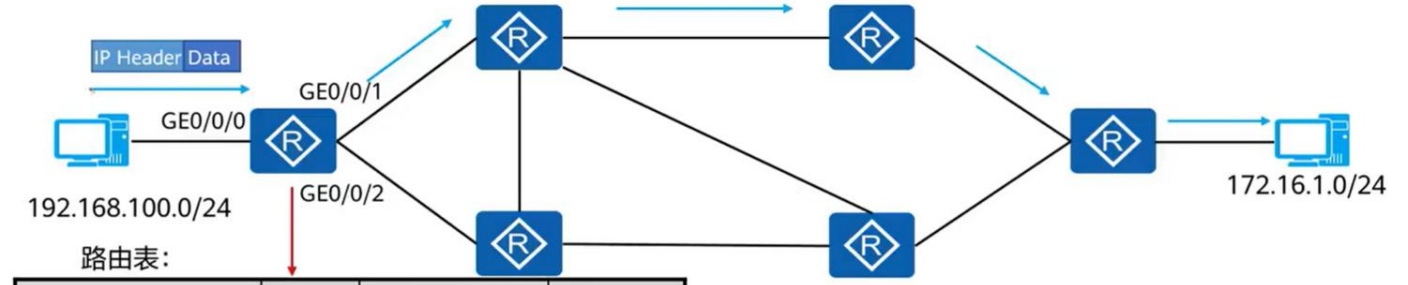
路由就是报文从源端到目的端的路径。当报文从路由器到目的网段有多条路由可达时，路由器可以根据路由表中最佳路由进行转发。



2、什么是路由表?

在计算机网络中，路由表（routing table）是一个**存储在路由器或者联网计算机中的电子表格（文件）或类数据库。路由表存储着指向特定网络地址的路径。路由表建立的主要目标是为了实现路由协议和静态路由选择。路由表就相当于一张地图告诉数据报该如何去往目的地。**

路由器会根据路由表进行数据包的转发：



目的网络/掩码	协议	下一跳	出接口
192.168.100.0/24	Direct	192.168.100.254	GE0/0/0
192.168.12.0/24	Direct	192.168.12.1	GE0/0/1
192.168.13.0/24	Direct	192.168.13.1	GE0/0/2
172.16.1.0/24	static	192.168.12.2	GE0/0/1

在ubuntu系统下我们可以使用**route -n** 命令查看系统的路由表：

```
msb@msb:~$ route -n
内核 IP 路由表
目标            网关            子网掩码        标志  跃点   引用  使用 接口
0.0.0.0         192.168.33.2    0.0.0.0         UG    100    0        0 ens33
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 ens33
192.168.33.0    0.0.0.0         255.255.255.0   U     100    0        0 ens33
```

在windows下可以使用 **route print** 命令查看系统的路由表：

```
IPv4 路由表
===========================================================================
活动路由:
网络目标        网络掩码          网关       接口   跃点数
          0.0.0.0          0.0.0.0      192.168.1.1     192.168.1.4     35
        127.0.0.0        255.0.0.0         在链路上         127.0.0.1    331
        127.0.0.1  255.255.255.255         在链路上         127.0.0.1    331
  127.255.255.255  255.255.255.255         在链路上         127.0.0.1    331
      192.168.1.0    255.255.255.0         在链路上       192.168.1.4    291
      192.168.1.4  255.255.255.255         在链路上       192.168.1.4    291
    192.168.1.255  255.255.255.255         在链路上       192.168.1.4    291
     192.168.33.0    255.255.255.0         在链路上      192.168.33.1    291
     192.168.33.1  255.255.255.255         在链路上      192.168.33.1    291
   192.168.33.255  255.255.255.255         在链路上      192.168.33.1    291
     192.168.40.0    255.255.255.0         在链路上      192.168.40.1    291
     192.168.40.1  255.255.255.255         在链路上      192.168.40.1    291
   192.168.40.255  255.255.255.255         在链路上      192.168.40.1    291
        224.0.0.0        240.0.0.0         在链路上         127.0.0.1    331
        224.0.0.0        240.0.0.0         在链路上      192.168.40.1    291
        224.0.0.0        240.0.0.0         在链路上      192.168.33.1    291
        224.0.0.0        240.0.0.0         在链路上       192.168.1.4    291
  255.255.255.255  255.255.255.255         在链路上         127.0.0.1    331
  255.255.255.255  255.255.255.255         在链路上      192.168.40.1    291
  255.255.255.255  255.255.255.255         在链路上      192.168.33.1    291
  255.255.255.255  255.255.255.255         在链路上       192.168.1.4    291
```

我们发现ubuntu和windows系统的路由表中都有一条特殊的路由：目标地址为全网地址0.0.0.0，子网掩码为0.0.0.0，对应下一跳的地址为网关地址。那么这条路由的作用是什么呢？

我们先了解一下，计算机或者路由器是如何将一条数据通过网络发送出去的。首先将目的IP地址与自己的子网掩码进行按位与运算得到目的主机的主机号，然后拿该主机号与路由表中的路由进行匹

配，如果匹配到了则将数据转发给指定的下一跳，如果没有匹配到，则转发给目标地址为0.0.0.0所对应的下一跳。

因此目标地址为0.0.0.0所对应的那条路由的作用就是：这条路由就是我们所说的默认路由，是对IP数据包中的目的地址找不到存在的其他路由时，路由器所选择的路由。

> 注意1：当我们发现主机（尤其是一些嵌入式或者物联网的设备）不能够上网或者不能ping通外网的时候，假如检查了IP地址、DNS等设置都没问题后依然不能解决，我们可以查看系统的路由表中是否没有添加默认路由。

> 注意2：不同的路由器查看路由表的命令不一样
>
> **思科路由器查看路由表命令为：show ip route**
>
> **华为路由器查看路由表命令为：display ip routing-table**

3、路由表的类型

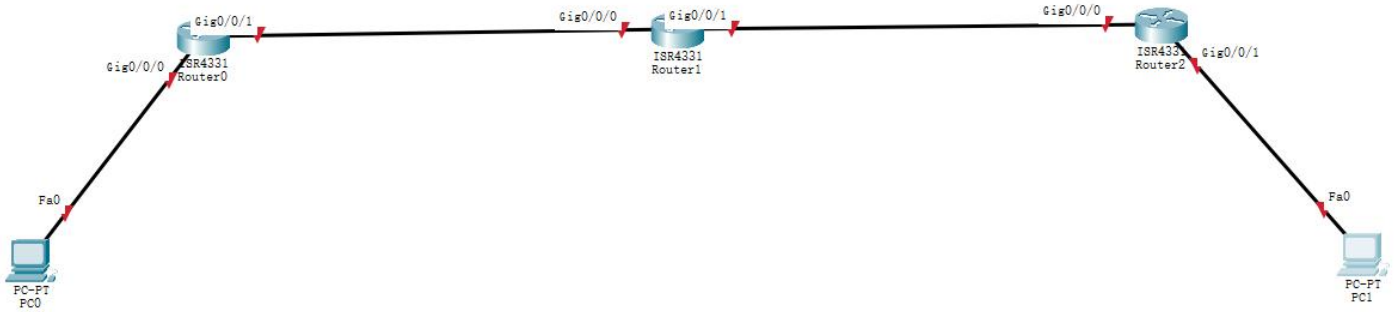
路由表可以分为：静态路由和动态路由。

- **动态路由**：动态路由是指动态路由协议(如RIP)自动建立路由，并且能够根据实际情况的变化适时地进行调整。
  - 动态路由之所以能根据网络的情况自动计算路由、选择转发路径，是由于当网络发生变化时，路由器之间彼此交换的路由信息会告知对方网络的这种变化，通过信息扩散使所有路由器都能得知网络变化。
  - 路由器根据某种路由算法（不同的动态路由协议算法不同）把收集到的路由信息加工成路由表，供路由器在转发IP报文时查阅
  - 常见的动态路由协议有：RIP、OSPF、IS-IS、BGP、IGRP/EIGRP
- **静态路由**：由网络管理员手动配置的路由。
  - Linux系统静态路由操作：
    - 添加路由：**route -n add -net 目标地址/子网掩码位数 gw 下一跳**，例如：route -n add -net 192.168.1.0/24 192.168.33.2
    - 添加默认路由：**route add default gw IP**
    - 删除路由：**route delete -net 目标地址/子网掩码位数**，例如：route delete -net 192.168.1.0/24
    - **注意：参数-net 表示操作某个网段，如果操作的数具体某台主机的地址则使用-host选项**
  - 路由器静态路由操作：
    - 添加路由：**ip route 目标地址 子网掩码 下一跳地址**，例如：ip route 192.168.3.0 255.255.255.0 192.168.1.1
    - 删除路由：**no ip route 目标地址 子网掩码 下一跳地址**

4、路由表实验

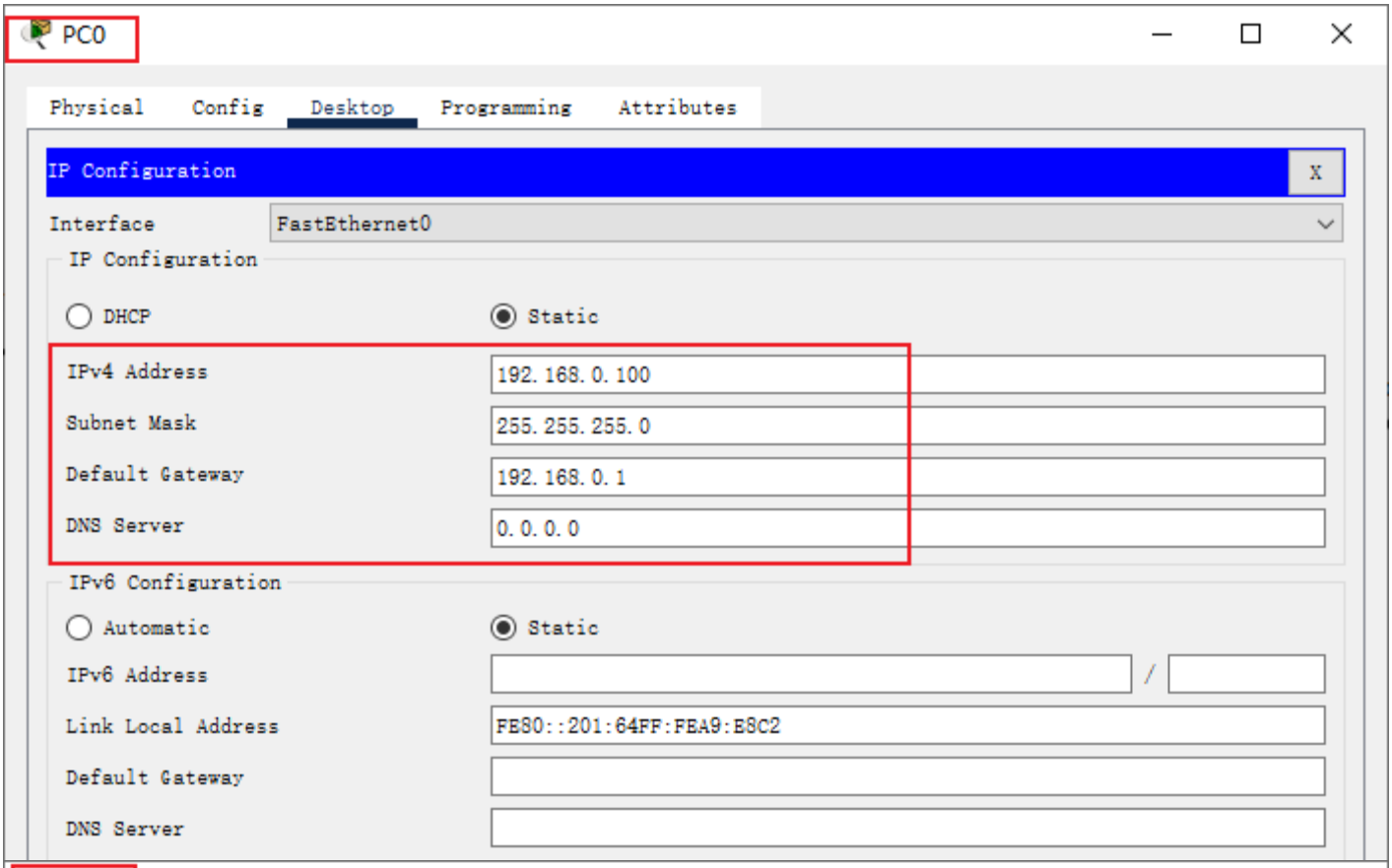
下面我们使用思科的模拟器Cisco Packet Tracer进行路由表的实验

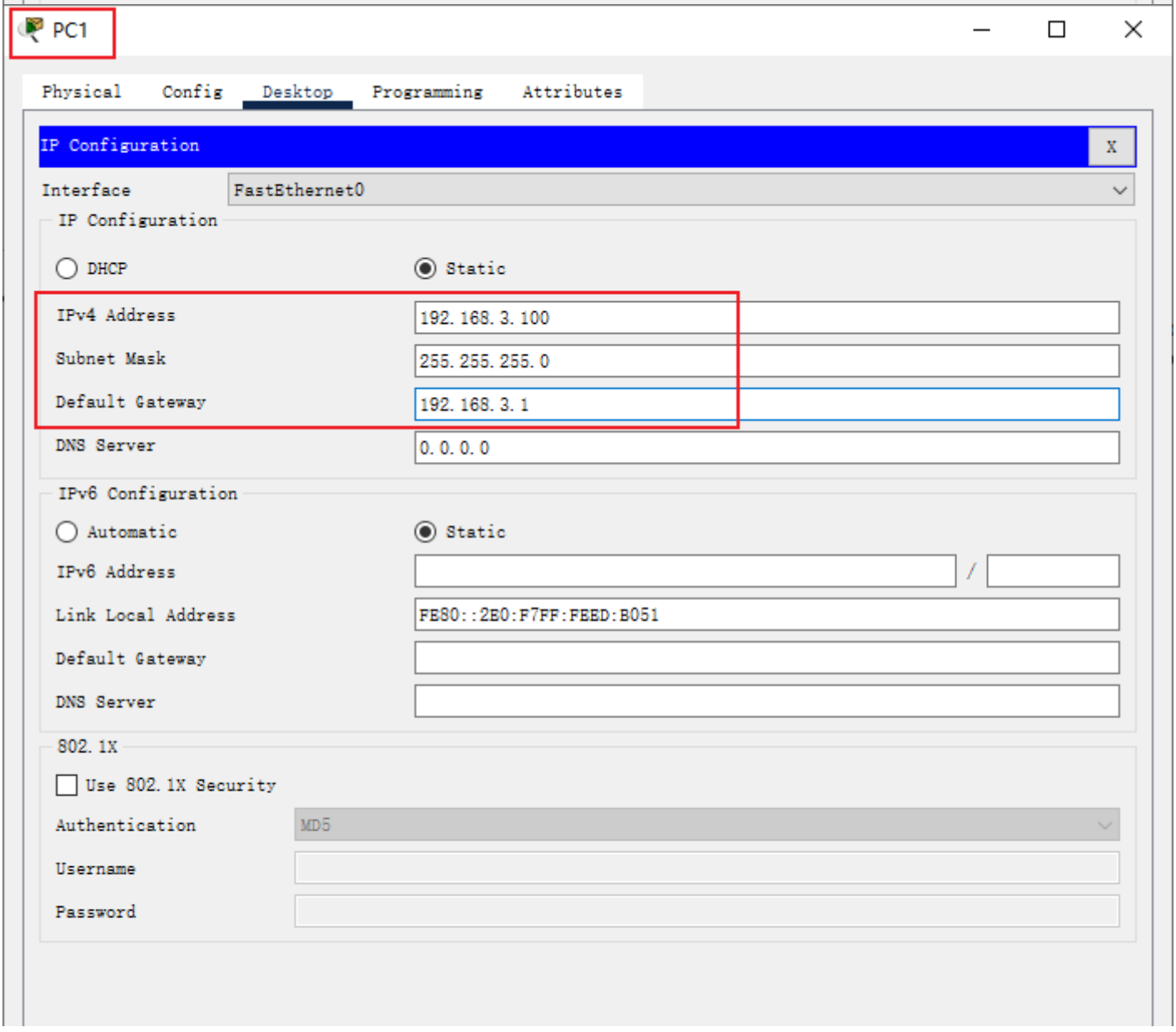
1）首先我们先搭建基础的网络，如下图所示：

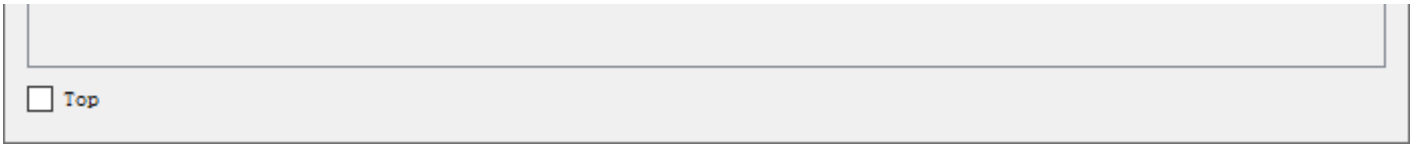


2）在此次实验中我们通过设置三个路由器的路由表实现主机PC0与主机PC1之间的通信

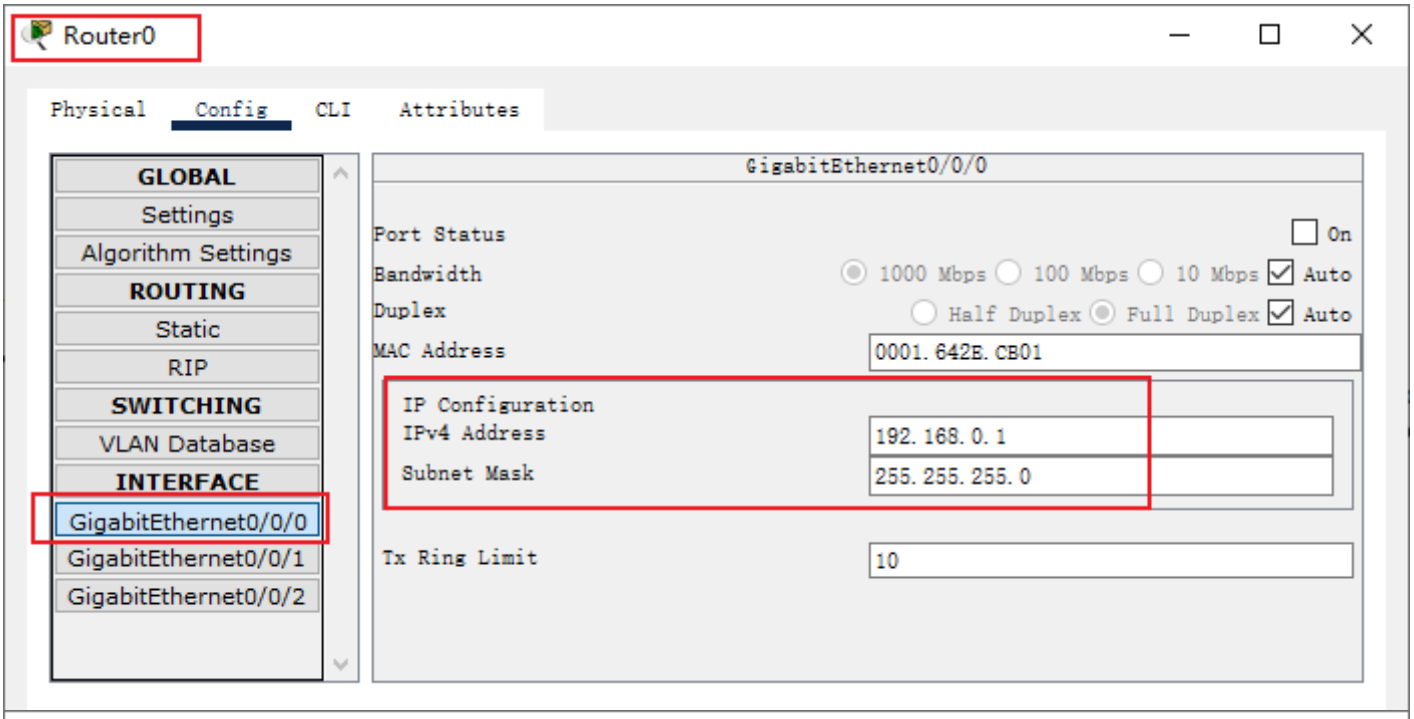
3）分别设置主机PC0与主机PC1的ip地址

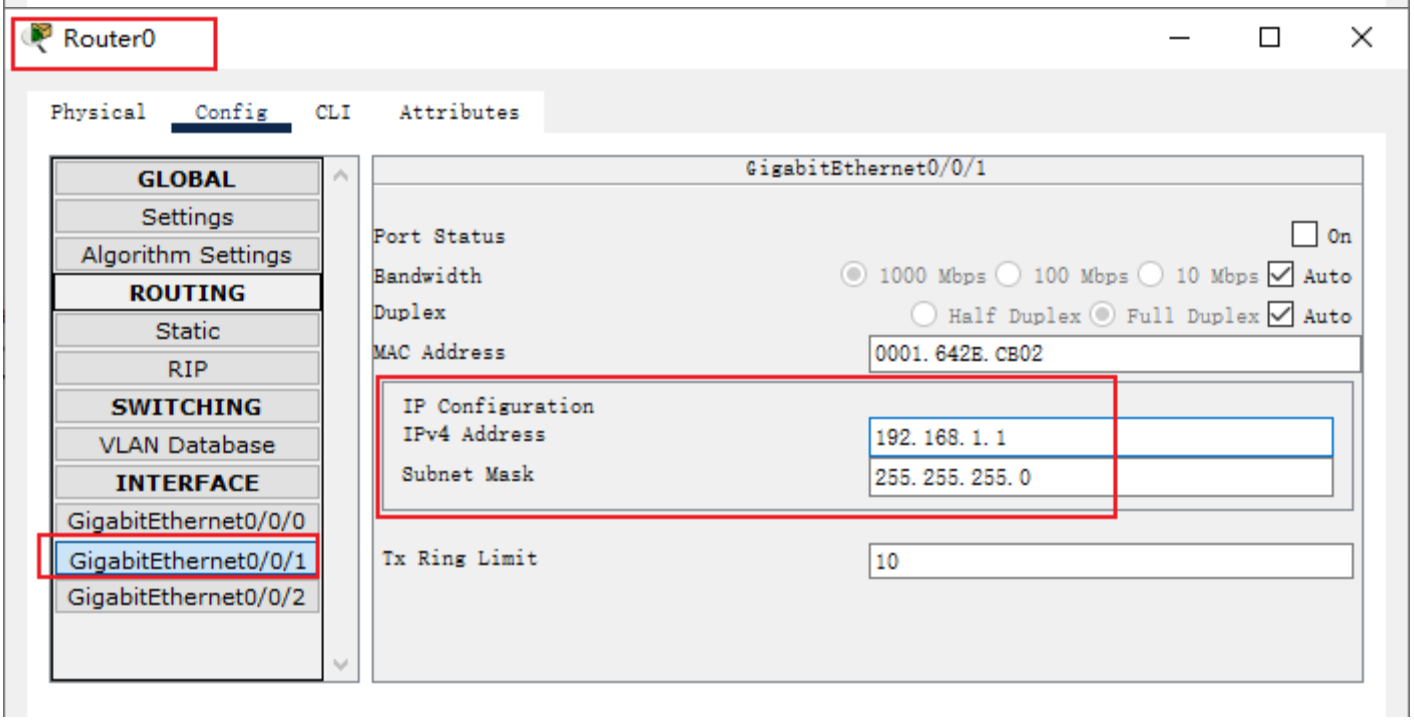

PC0
Physical
Config
Desktop
Programming
Attributes
IP Configuration
X
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
192.168.0.100
Subnet Mask
255.255.255.0
Default Gateway
192.168.0.1
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
/
Link Local Address
FE80::201:64FF:FEA9:B8C2
Default Gateway
DNS Server

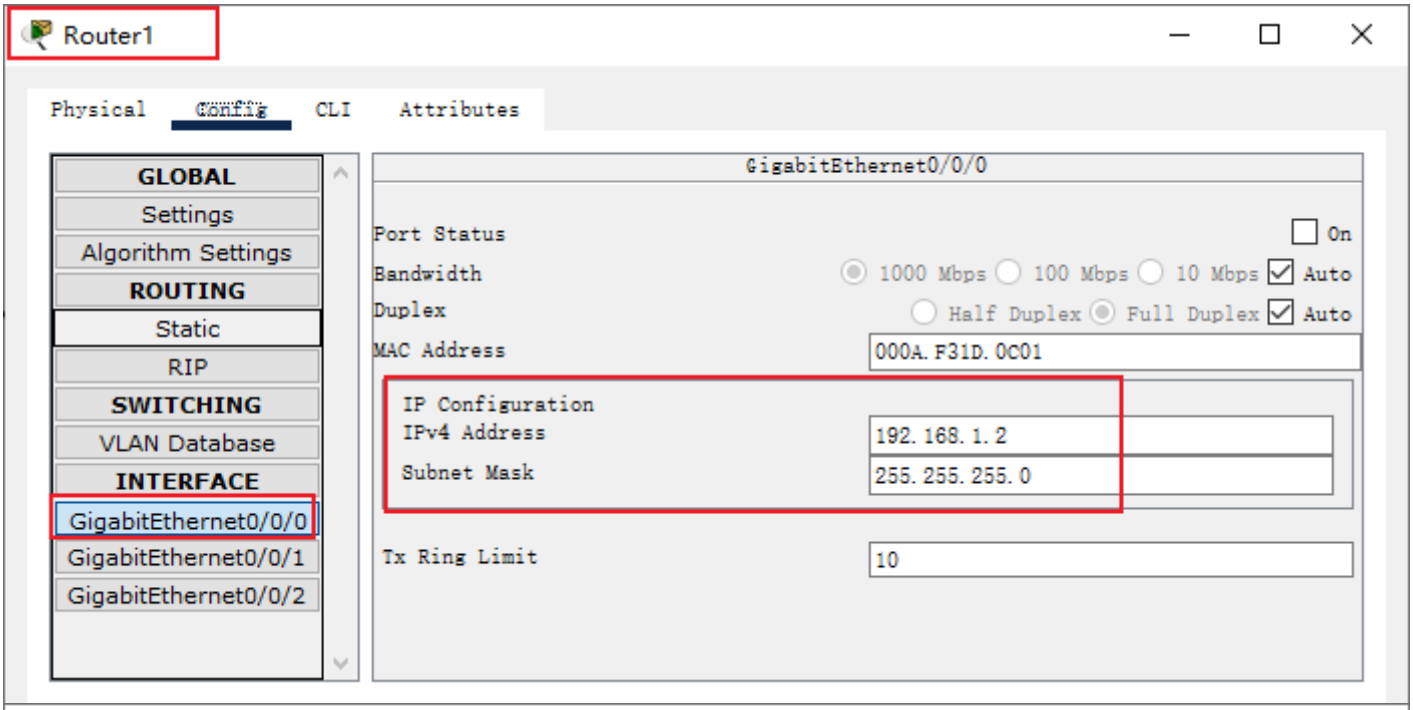

PC1
Physical
Config
Desktop
Programming
Attributes
IP Configuration
X
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
192.168.3.100
Subnet Mask
255.255.255.0
Default Gateway
192.168.3.1
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
/
Link Local Address
FE80::2E0:F7FF:FEED:B051
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Username
Password

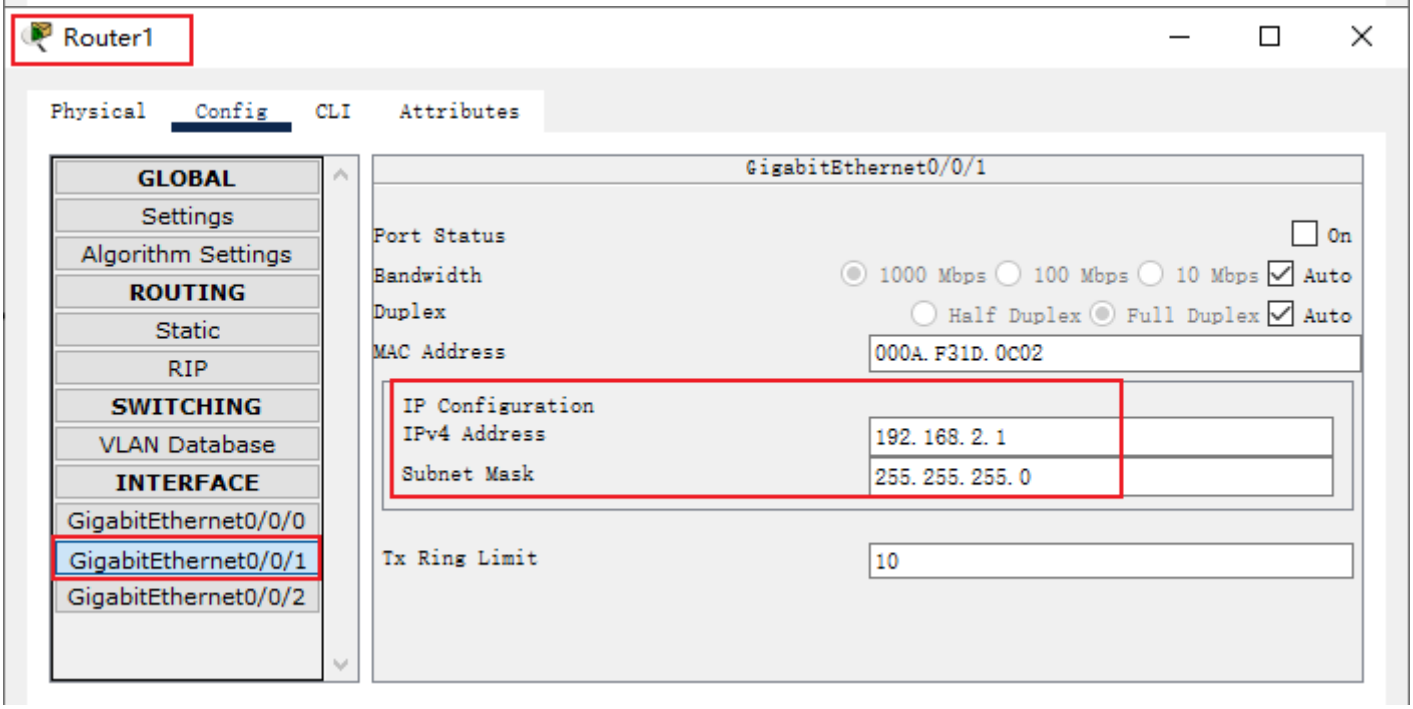


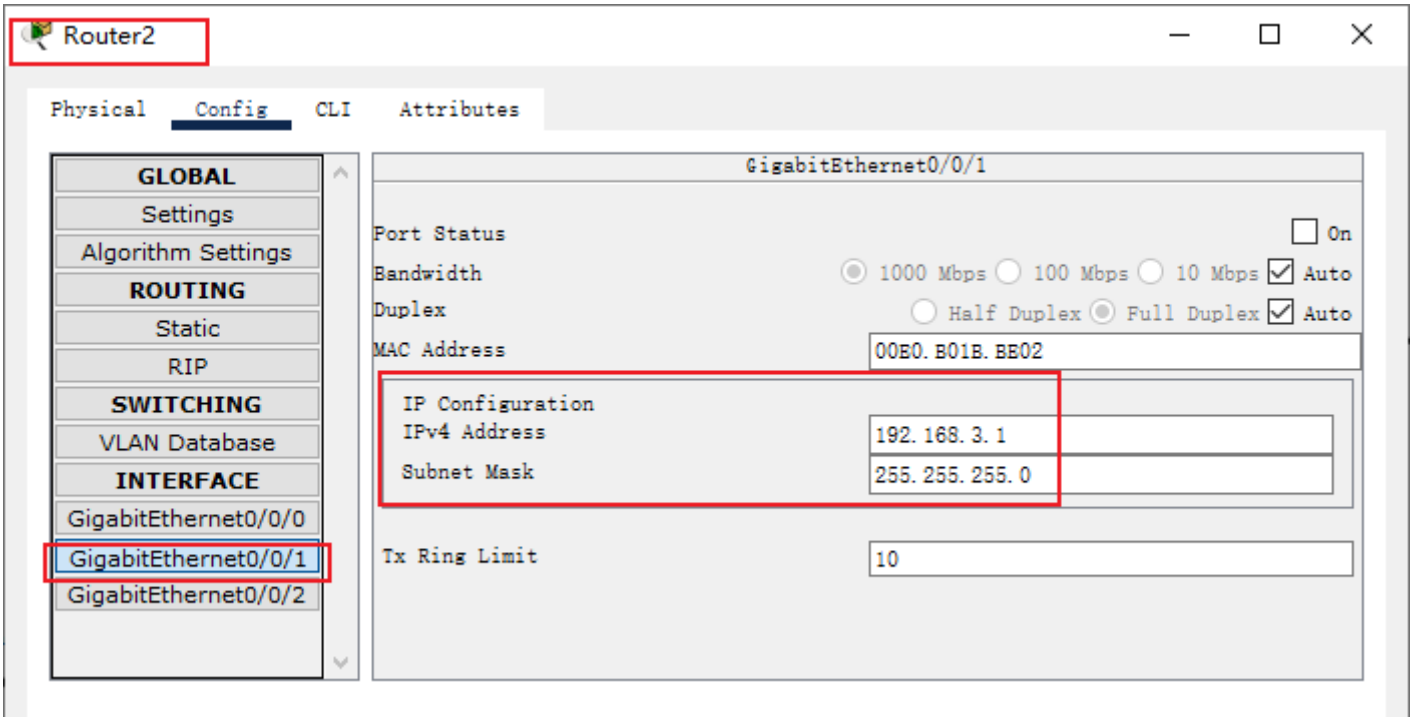
4）分别设置三台路由器的IP地址

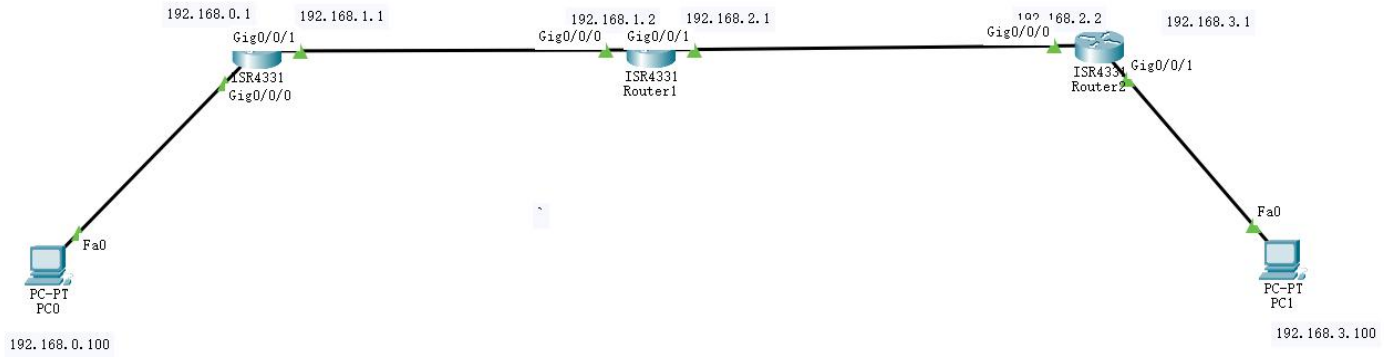





Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0/0
GigabitEthernet0/0/1
GigabitEthernet0/0/2
GigabitEthernet0/0/0
Port Status
On
Bandwidth
1000 Mbps
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
000A.F31D.0C01
IP Configuration
IPv4 Address
192.168.1.2
Subnet Mask
255.255.255.0
Tx Ring Limit
10

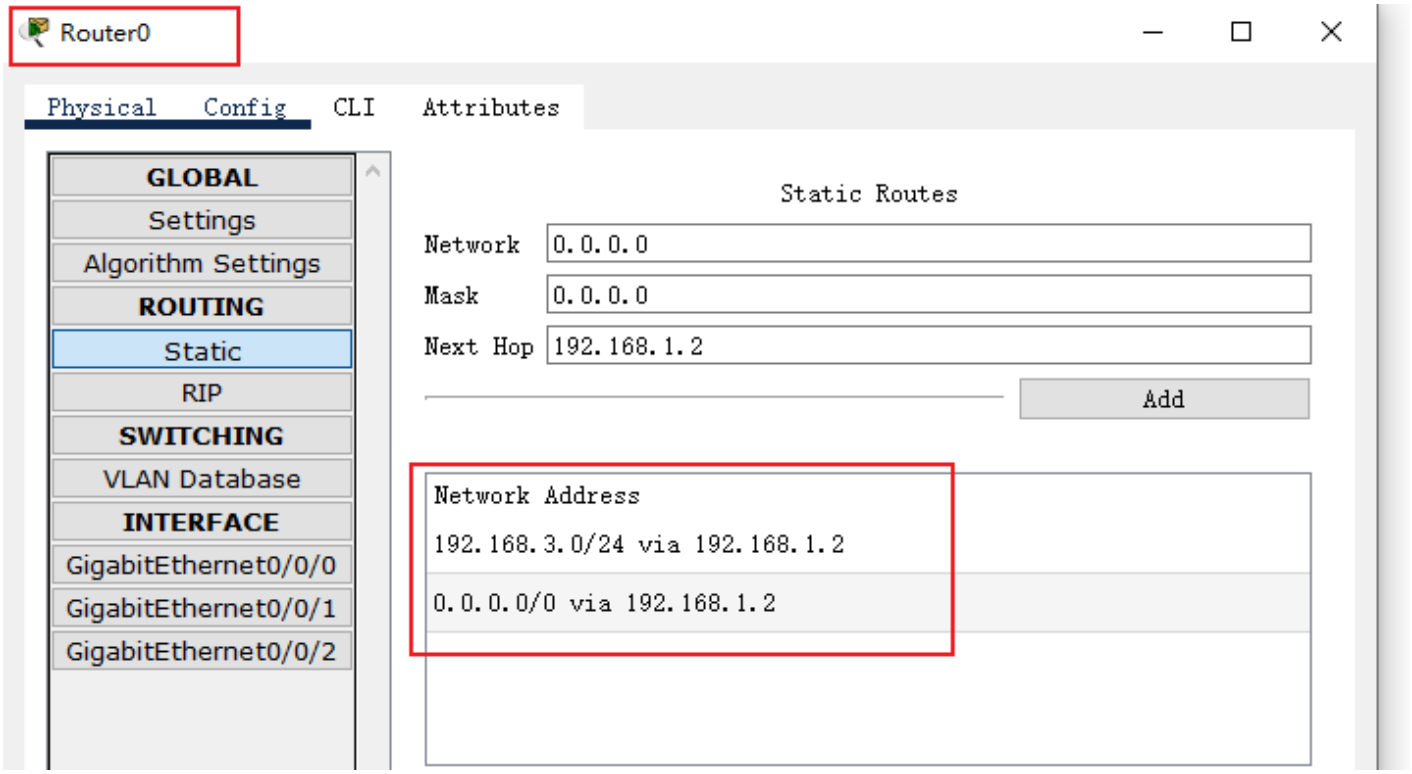

Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0/0
GigabitEthernet0/0/1
GigabitEthernet0/0/2
GigabitEthernet0/0/1
Port Status
On
Bandwidth
1000 Mbps
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
000A.F31D.0C02
IP Configuration
IPv4 Address
192.168.2.1
Subnet Mask
255.255.255.0
Tx Ring Limit
10

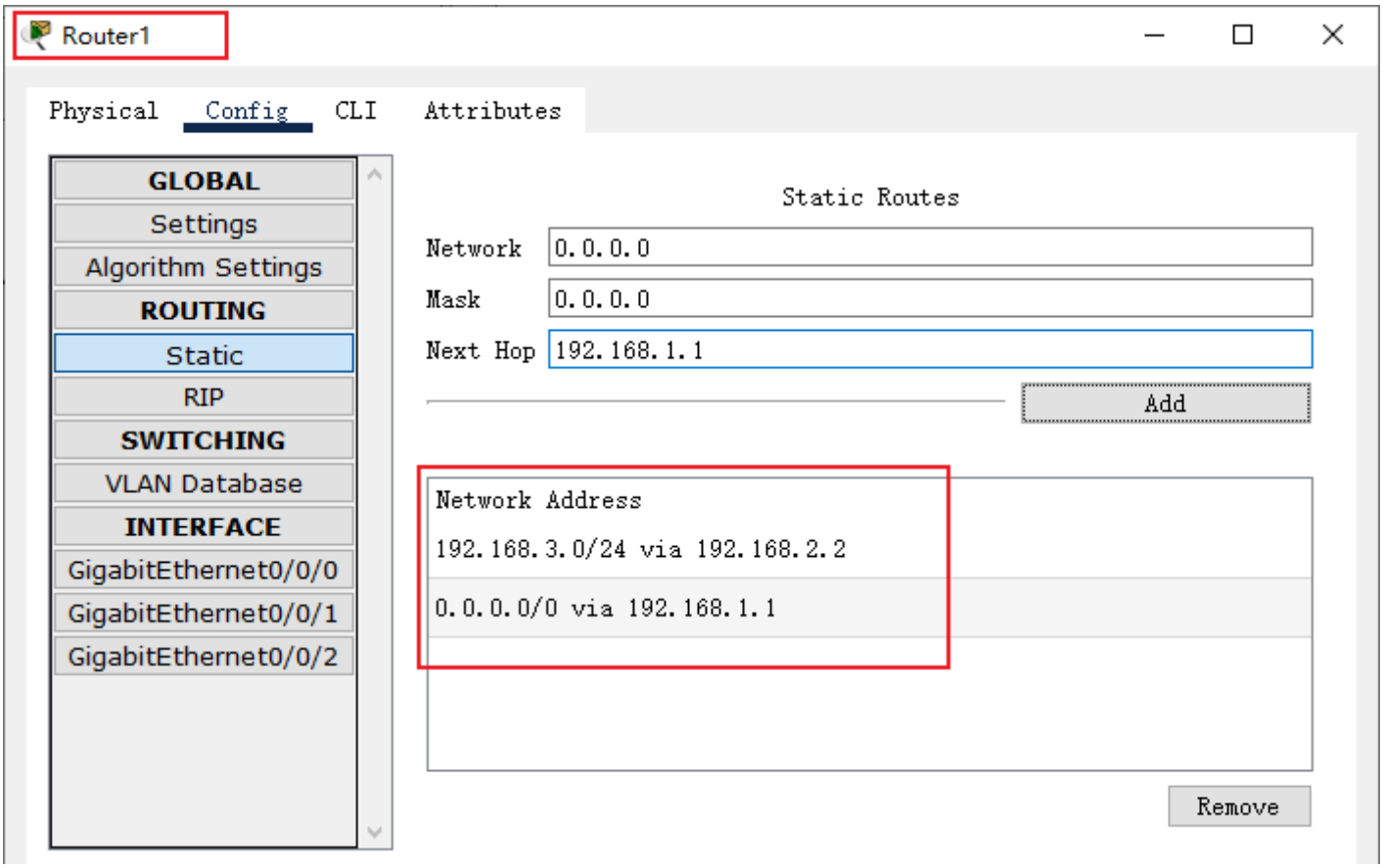


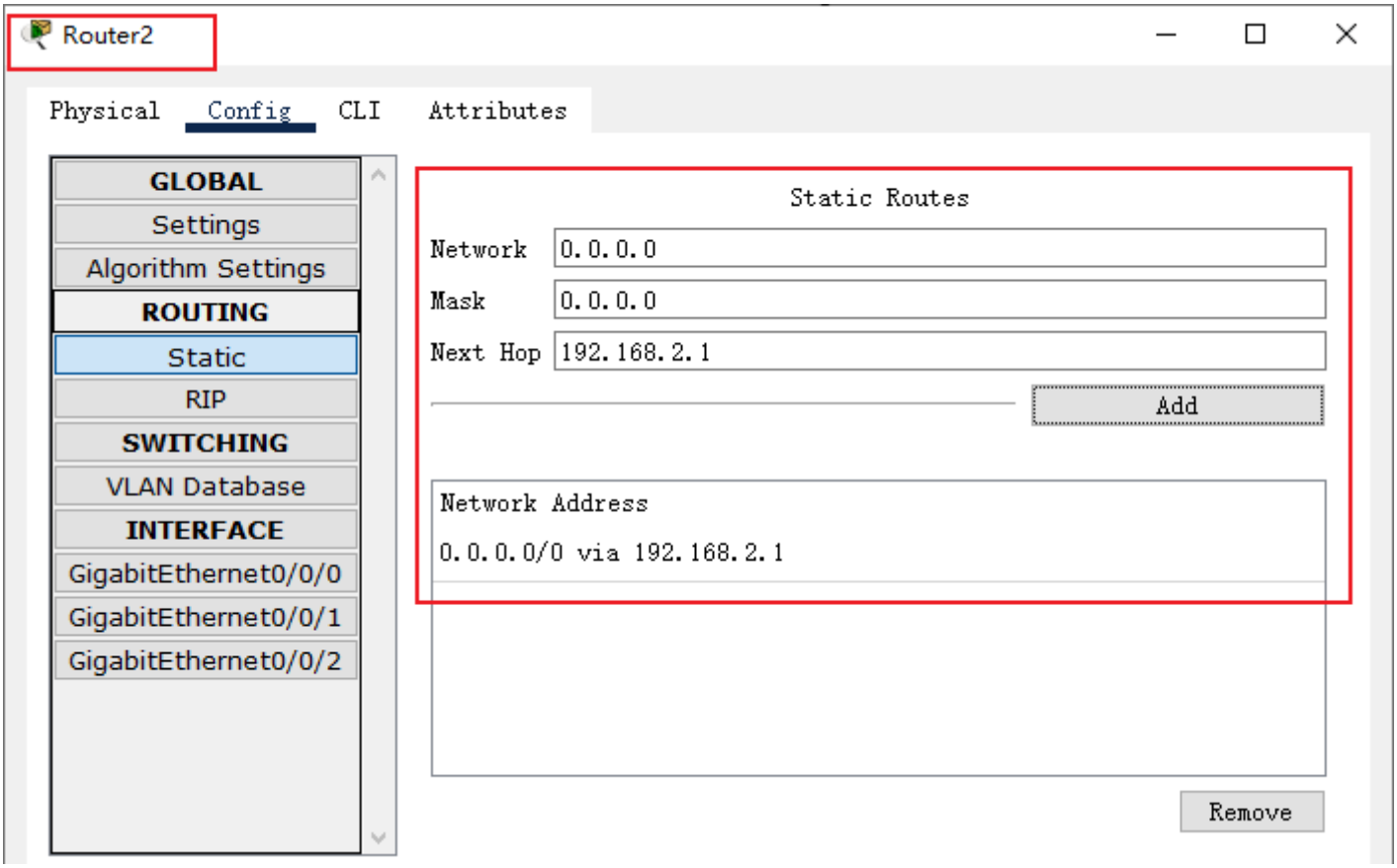


5）分别为三个路由器设置静态路由

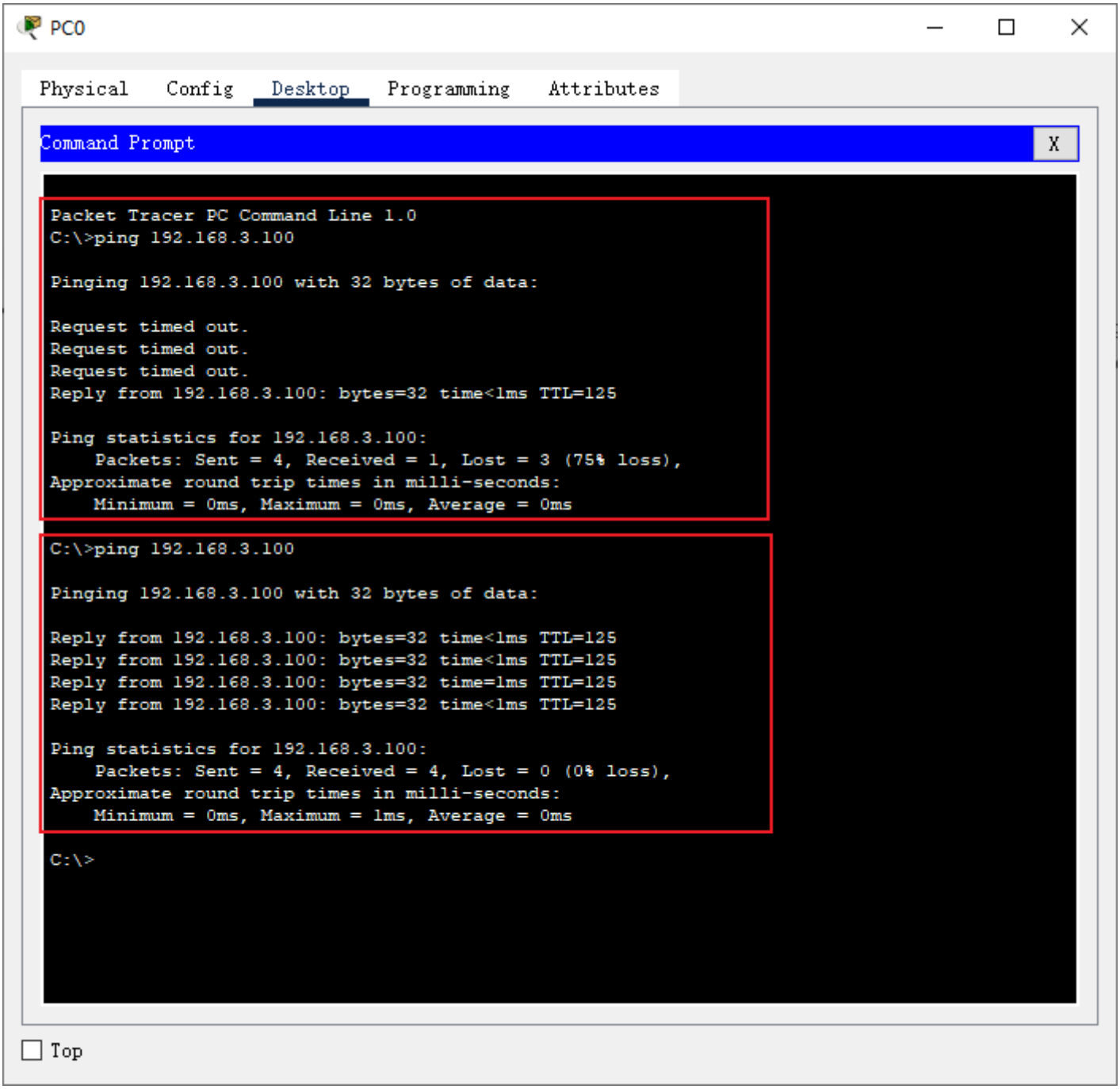
```
Route0:
ip  route  192.168.3.0  255.255.255.0  192.168.1.2
ip  route  0.0.0.0  0.0.0.0  192.168.1.2

Router1:
ip  route  192.168.3.0  255.255.255.0  192.168.2.2
ip  route  0.0.0.0  0.0.0.0  192.168.1.1

Router2:
ip  route  0.0.0.0  0.0.0.0  192.168.2.1
```


Router0
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0/0
GigabitEthernet0/0/1
GigabitEthernet0/0/2
Static Routes
Network 0.0.0.0
Mask 0.0.0.0
Next Hop 192.168.1.2
Add
Network Address
192.168.3.0/24 via 192.168.1.2
0.0.0.0/0 via 192.168.1.2


Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0/0
GigabitEthernet0/0/1
GigabitEthernet0/0/2
Static Routes
Network 0.0.0.0
Mask 0.0.0.0
Next Hop 192.168.1.1
Add
Network Address
192.168.3.0/24 via 192.168.2.2
0.0.0.0/0 via 192.168.1.1
Remove



6）在PC0尝试ping PC1

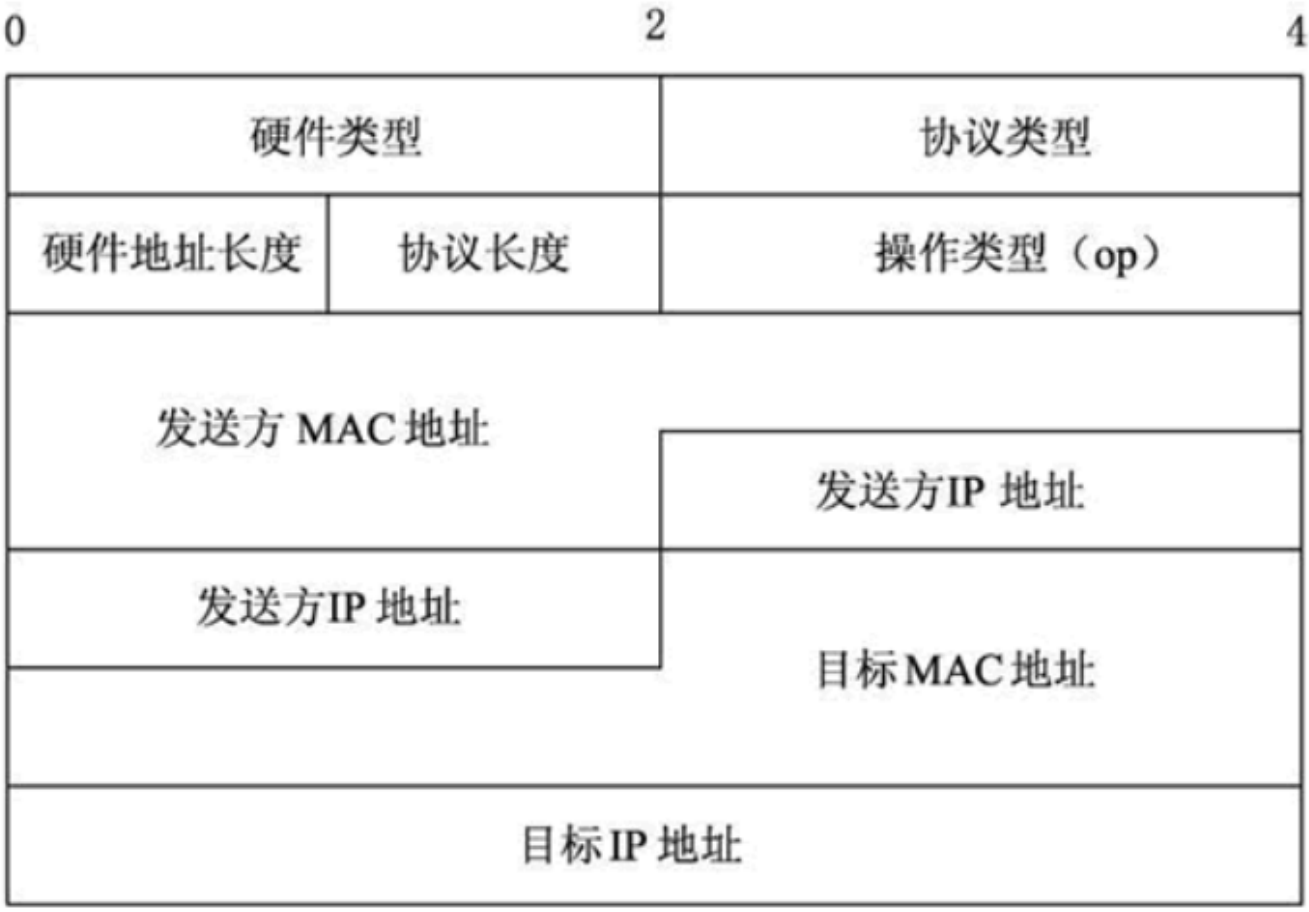


2.4.8 ARP高速缓存表

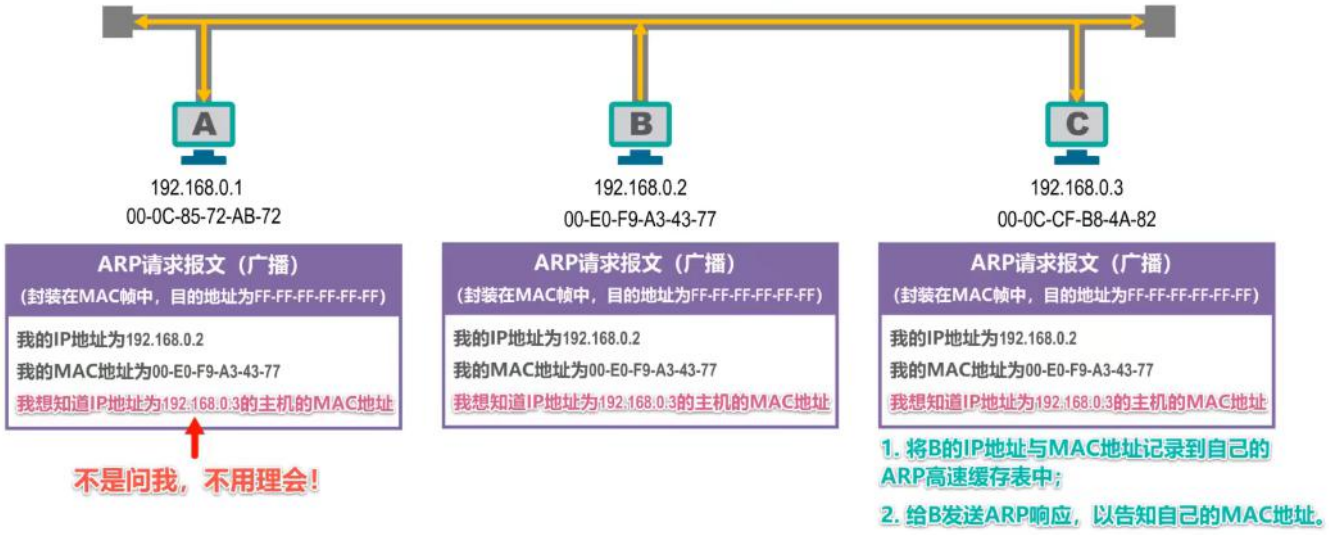
1、地址解析协议ARP

- 通过前面的学习，我们知道：如果网络中的主机需要互相通信，那么源主机必须要知道目的主机的IP地址和MAC地址，因为在数据链路层封装的MAC帧中需要封装目的地址和源地址。当已知目的主机的IP地址时，可以**通过ARP协议获得目的主机的MAC地址。**

- ARP请求报文封装在MAC帧中，目的地址为FF-FF-FF-FF，并且以广播的形式发送的。



- ARP协议工作流程



2、ARP高速缓存表

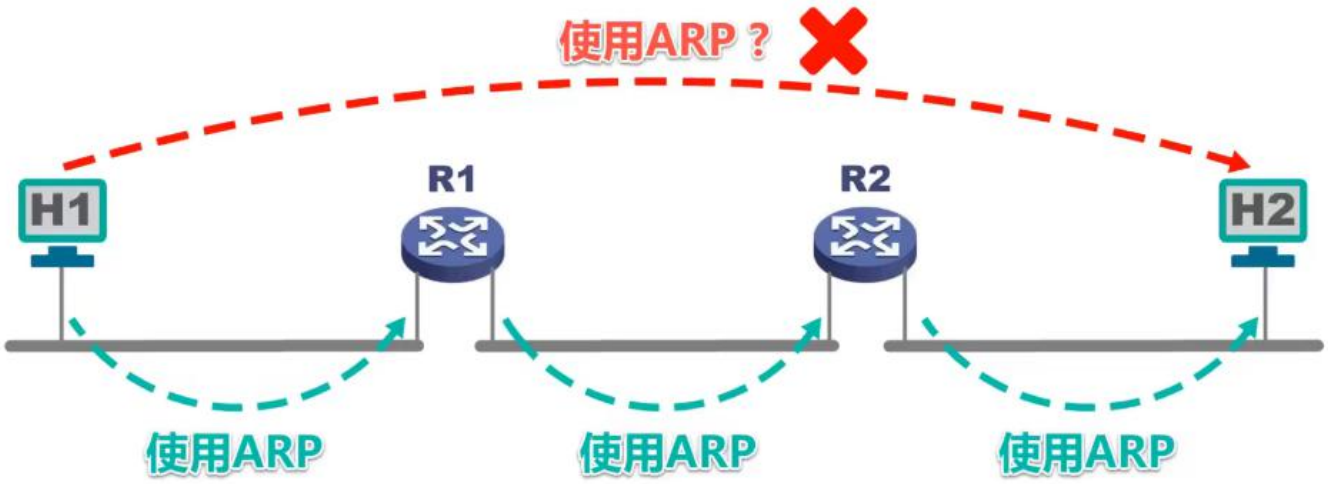
ARP高速缓存(ARP cache)，由最近的ARP项组成的一张临时表，表中记录了**主机IP地址和MAC地址的对应关系。**

主机B的ARP高速缓存		
IP地址	MAC地址	类型
192.168.0.1	00-0C-85-72-AB-72	动态
192.168.0.4	00-01-C7-D3-B2-B5	静态
192.168.0.3	00-0C-CF-B8-4A-82	动态
.	.	.
.	.	.

动态：自动获取，生命周期默认为两分钟；

静态：手工设置，不同操作系统下的生命周期不同，例如系统重启后不存在或系统重启后依然有效。

注意：ARP报文只能在发送主机所在的广播域中被使用！



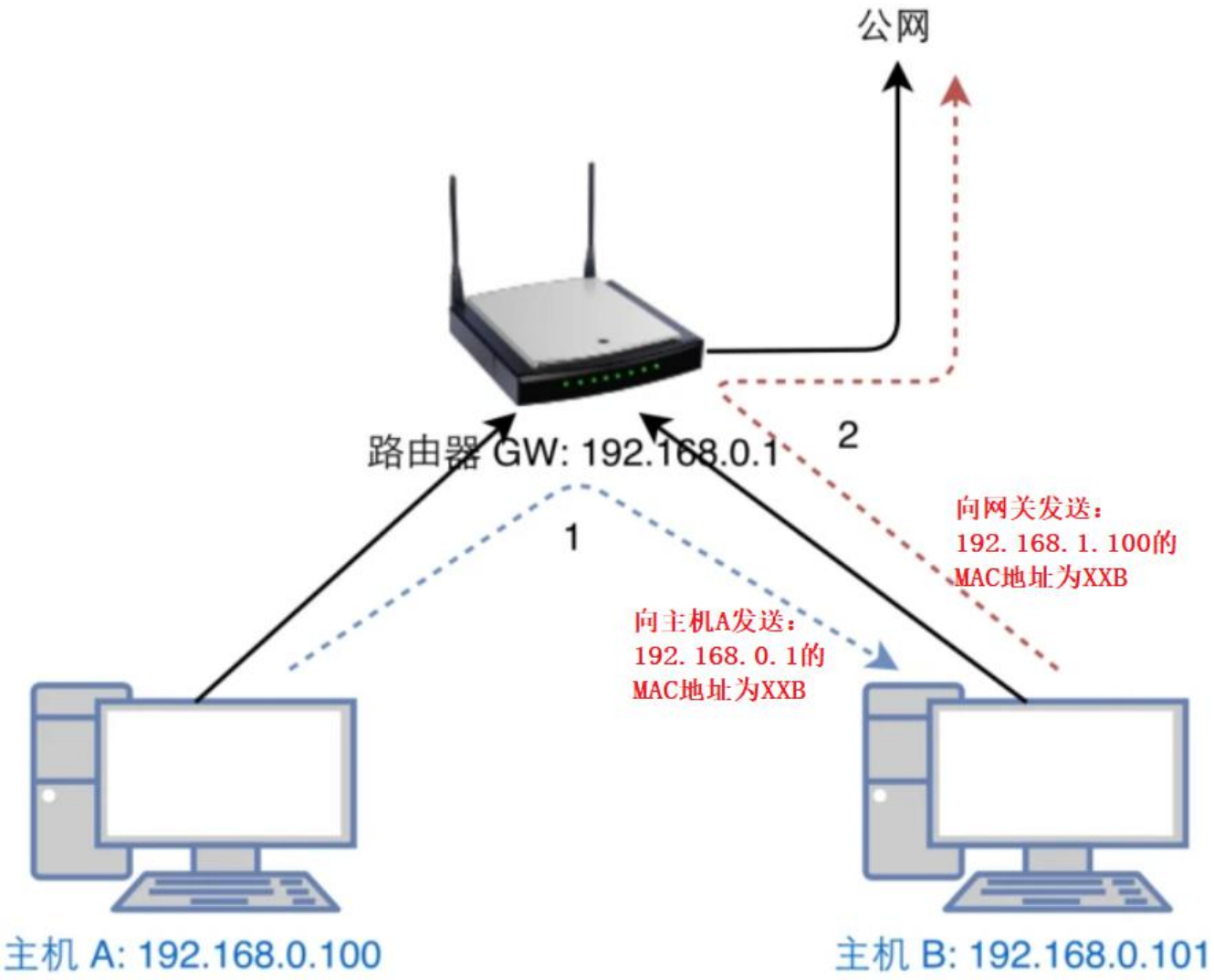
3、ARP欺骗

ARP 欺骗是一种以 **ARP 地址解析协议为基础**的一种网络攻击方式。

ARP欺骗的原理：利用**ARP协议没有安全认证机制**，攻击者发送假的ARP数据包到网上，尤其是送到网关上。其目的是要让送至特定的IP地址的流量被错误送到攻击者所取代的地方。因此攻击者可将这些流量另行转送到真正的网关（被动式数据包嗅探，passive sniffing）或是篡改后再转送（中间人攻击，man-in-the-middle attack）。

ARP欺骗分为两种：

- **主机型欺骗**：攻击者向其他主机发送ARP响应包，攻击者在响应包中填入网关的IP地址和自己MAC地址，这样主机A应该发送给网关的数据包会被网关转发给主机B
- **网关型欺骗**：攻击者向网关发送ARP响应包，攻击者在响应包中填入被攻击主机的IP地址和自己MAC地址，这样网关接收到互联网中发给主机A的数据时也会转发给主机B



ARP欺骗的危害：

- 信息安全收到威胁
- 网络延迟增加
- 网络通信中断

ARP欺骗防范方法：

- 安装补丁
- 静态绑定：网内的主机与网关做IP和MAC绑定
- 使用ARP防火墙
- 不要点开通讯工具中的一些可疑链接、图片、文件等。

2.4.9 特殊IP地址

1、127.0.0.1

回环地址，该地址还有 一个别名叫 “localhost” ，无论是哪个程序，一旦使用该地址发送数据，协议软件会立即返回，不进行任何网络传输，除非出错，包含该网络号的分组是不能够出现 在任何网络上的。

2、10.*.*.*，*172.16.*.*—172.31.*.*，192.168.*.

上面三个网段是私有地址，可以用于自己组网使用，这些地址主要用于企业内部网络中，但不能够在Internet网上使用，Internet网没有这些地址 的路由，而使用这三个网段的计算机要上网必须要通过网络地址转换(NAT)，将私有地址翻译成公用合法的IP地址。

3、255.255.255.255

受限制的广播地址，对本机来说，这个地址指本网段内(同一个广播域)的所有主机，该地址用于主机配置过程中IP数据包的目的地址，这时主机可能还不知道它 所在网络的网络掩码，甚至连它的IP地址也还不知道。在任何情况下，路由器都会禁止转发目的地址为受限的广播地址的数据包，这样的数据包仅会出现在本地网络中。

4、224.0.0.0—239.255.255.255

组播地址

5、169.254.*.*

如果你的主机使用了DHCP功能自动获得一个ip地址，那么当你的DHCP服务器发生故障或响应时间太长而超出系统规定的一个时间，windows系统会为你分配这样一个地址。如果你发现你的主机ip地址是个诸如此类的地址，很不幸，十有八九是你的网络不能正常运行了。

6、114.114.114.114

中国电信DNS

7、8.8.8.8

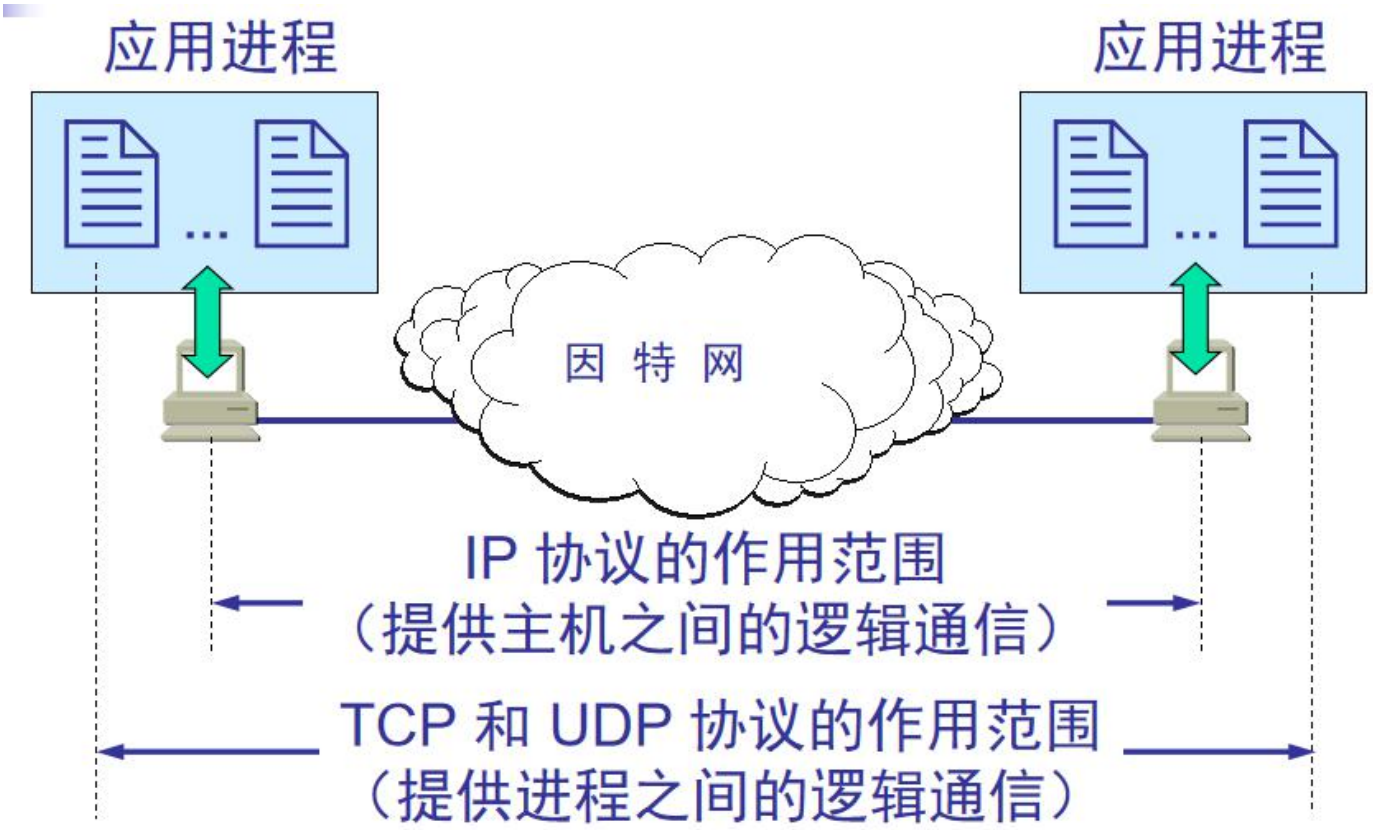
谷歌DNS

2.5 运输层

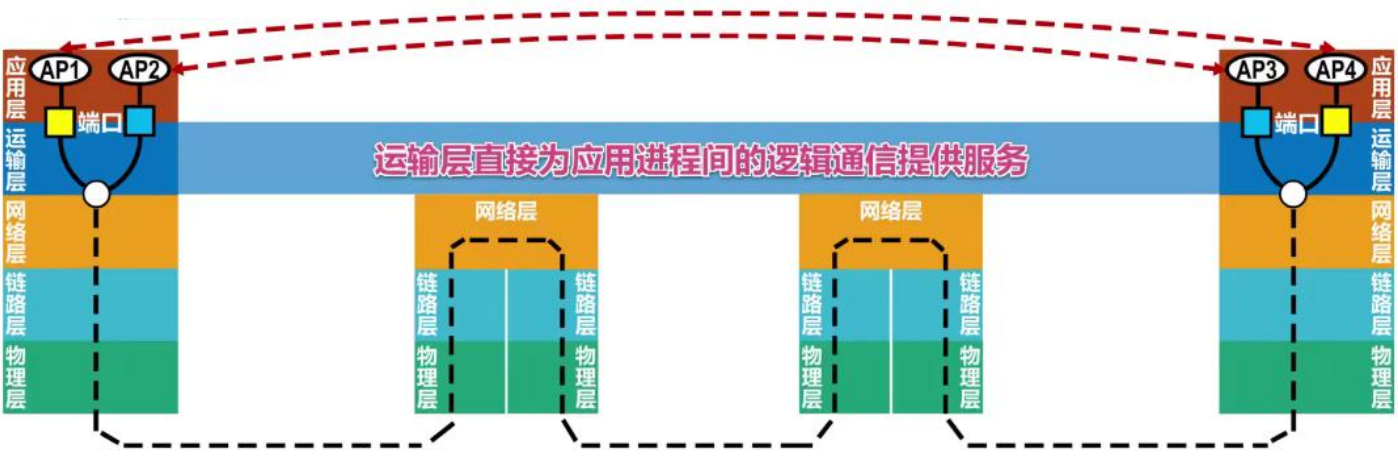
2.5.1 运输层概述

之前课程所介绍的计算机网络体系结构中的物理层、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。两个主机进行通信实际上就是两个主机中的**应用进程互相通信**，应用进程之间的通信又称为**端到端的通信**。

当网络的边缘部分中的两个主机使用网络的核心部分的功能进行端到端的通信时，只有位于**网络边缘部分的主机**的协议栈才有运输层，而网络核心部分中的路由器在转发分组时都只用到下三层的功能。**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是运输层的任务。**



运输层提供的是应用进程间的**逻辑通信**，“逻辑通信”的意思是：运输层之间的通信好像是沿水平方向传送数据。但事实上这两个运输层之间并没有一条水平方向的物理连接。



根据应用需求的不同，因特网的运输层为应用层提供了两种不同的运输协议，即**面向连接的TCP**和**无连接的UDP**。

2.5.2 端口号

1、思考

主机A使用QQ向PC发送了一组数据，数据在网络中经过传输后到达主机B，主机B对数据进行解包以后又是如何知道该数据是发送给哪个进程的呢?

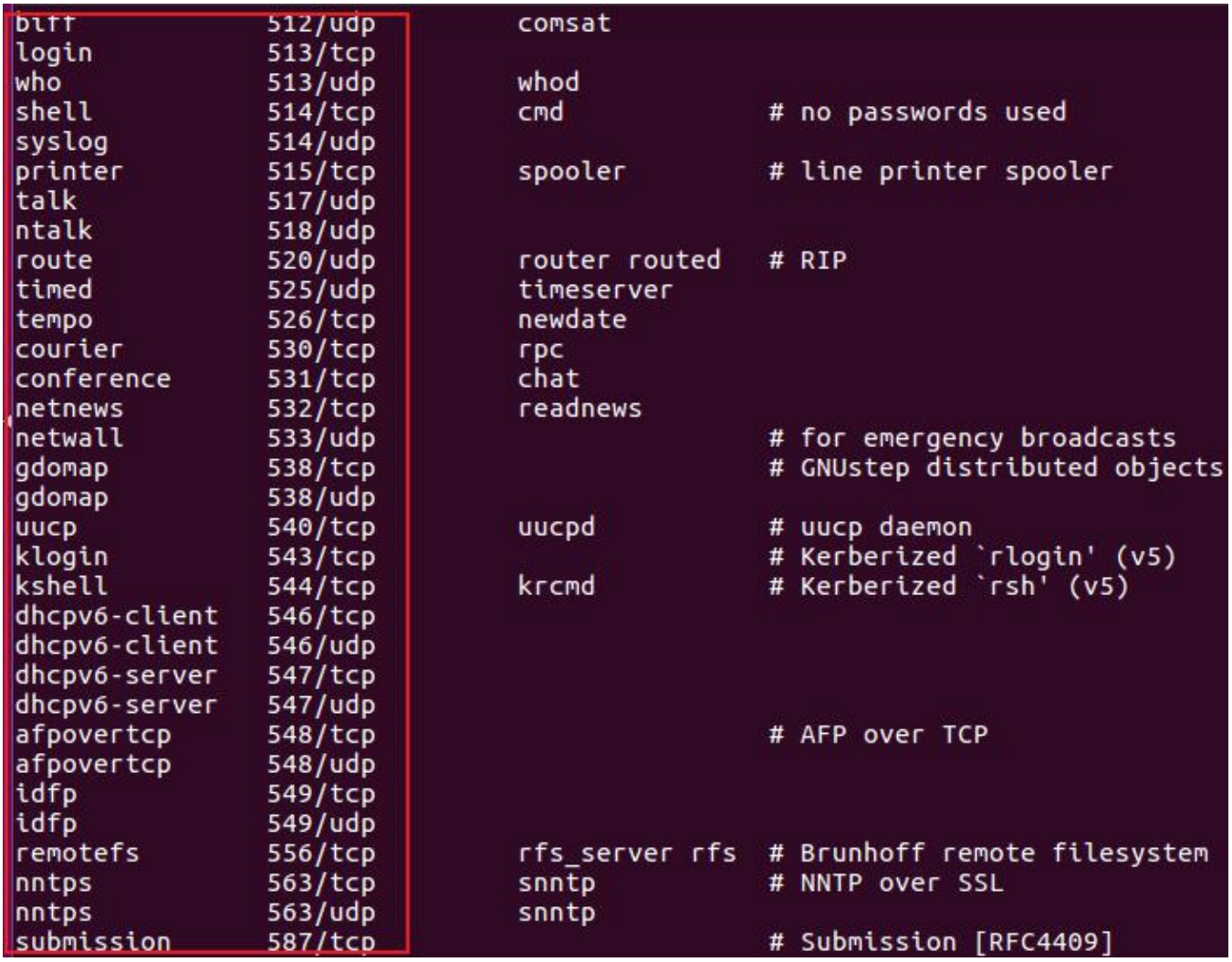
> TCP/IP体系的运输层使用**端口号**来区分应用层的不同应用进程。

2、端口号的值

- 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，例如：**FTP使用21/20，HTTP使用80，DNS使用53。**
- 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。例如：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。
- 常用端口号及对应的网络应用程序表格如下：

应用程序	默认端口号
HTTP	80/8080/3128/8081/9098
SOCKS	1080
FTP（文件传输）协议	21
Telnet（远程登录）	21
HTTP服务器	80/tcp（木马Executor开放此端口）
HTTPS（securely transferring web pages）服务器	443/tcp 443/udp
Telnet（不安全的文本传送）	23/tcp（木马Tiny Telnet Server所开放的端口）
TFTP（Trivial File Transfer Protocol）	69/udp
FTP	21/tcp（木马Doly Trojan、Fore、Invisible FTP、WebEx、WinCrash和Blade Runner所开放的端口）
SSH（安全登录）、SCP（文件传输）、端口号重定向	22/tcp
SMTP Simple Mail Transfer Protocol（E-mail）	25/tcp（木马Antigen、Email Password Sender、Haebu Coceda、Shtrilitz Stealth、WinPC、WinSpy都开放这个端口）
POP3 Post Office Protocol（E-mail）	110/tcp
Webshpere应用程序	9080
webshpere管理工具	9090
JBOSS	**8080**
TOMCAT	**8080**（这里没有出错，建议百度一下）
WIN2003远程登录	3389
Symantec AV/Filter for MSE	8081
Oracle 数据库	1521
ORACLE EMCTL	1158
Oracle XDB（XML 数据库）	8080
Oracle XDB FTP服务	2100
MS SQL*SERVER数据库server	1433/tcp 1433/udp
MS SQL*SERVER数据库monitor	1434/tcp 1434/udp

- 在Linux中/etc/service文件中保存了知名的服务器端口号



```
biff            512/udp         comsat
login           513/tcp
who             513/udp         whod
shell           514/tcp         cmd             # no passwords used
syslog          514/udp
printer         515/tcp         spooler         # line printer spooler
talk            517/udp
ntalk           518/udp
route           520/udp         router routed   # RIP
timed           525/udp         timeserver
tempo           526/tcp         newdate
courier         530/tcp         rpc
conference      531/tcp         chat
netnews         532/tcp         readnews
netwall         533/udp                         # for emergency broadcasts
gdomap          538/tcp                         # GNUstep distributed objects
gdomap          538/udp
uucp            540/tcp         uucpd           # uucp daemon
klogin          543/tcp                         # Kerberized `rlogin' (v5)
kshell          544/tcp         krcmd           # Kerberized `rsh' (v5)
dhcpv6-client   546/tcp
dhcpv6-client   546/udp
dhcpv6-server   547/tcp
dhcpv6-server   547/udp
afpovertcp      548/tcp                         # AFP over TCP
afpovertcp      548/udp
idfp            549/tcp
idfp            549/udp
remotefs        556/tcp         rfs_server rfs  # Brunhoff remote filesystem
nntps           563/tcp         snntp           # NNTP over SSL
nntps           563/udp         snntp
submission      587/tcp                         # Submission [RFC4409]
```

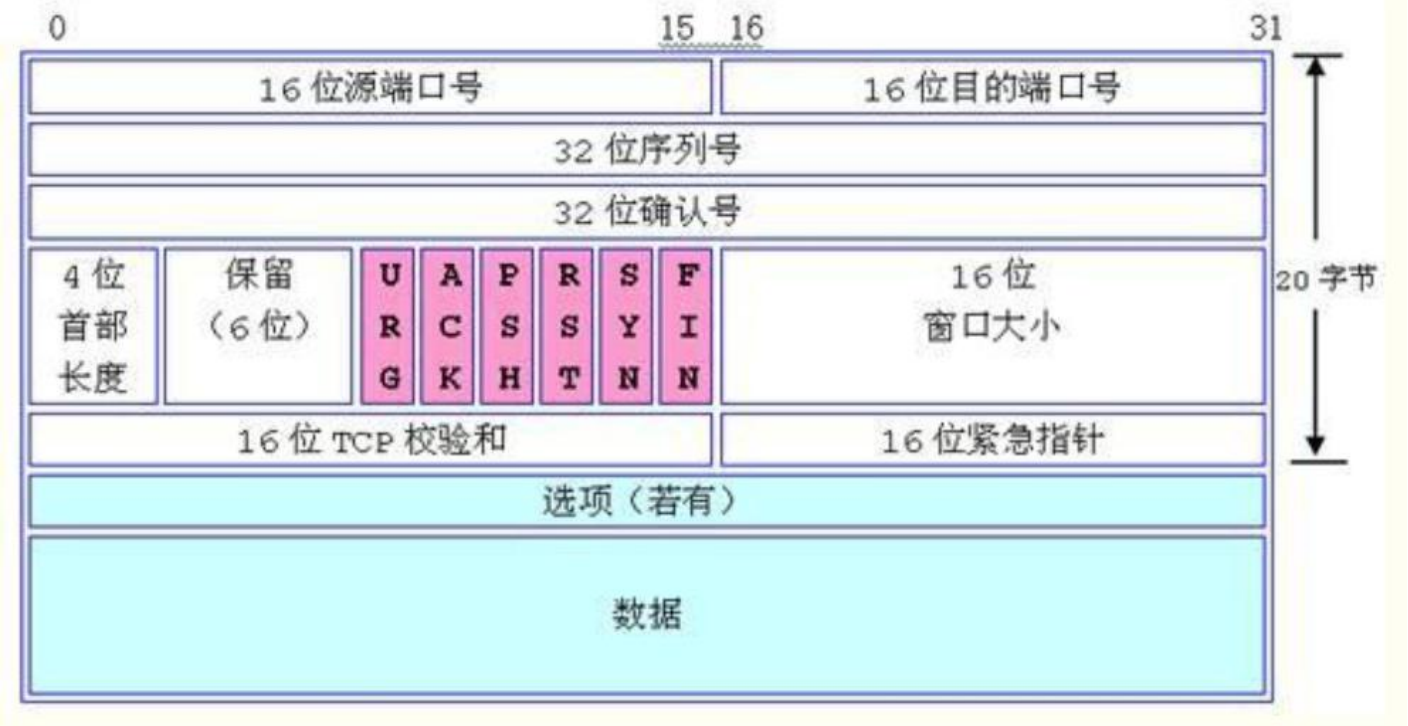
- **端口号只具有本地意义**，即端口号只是为了标识本计算机应用层中的各进程，在因特网中,不同计算机中的相同端口号是没有联系的。

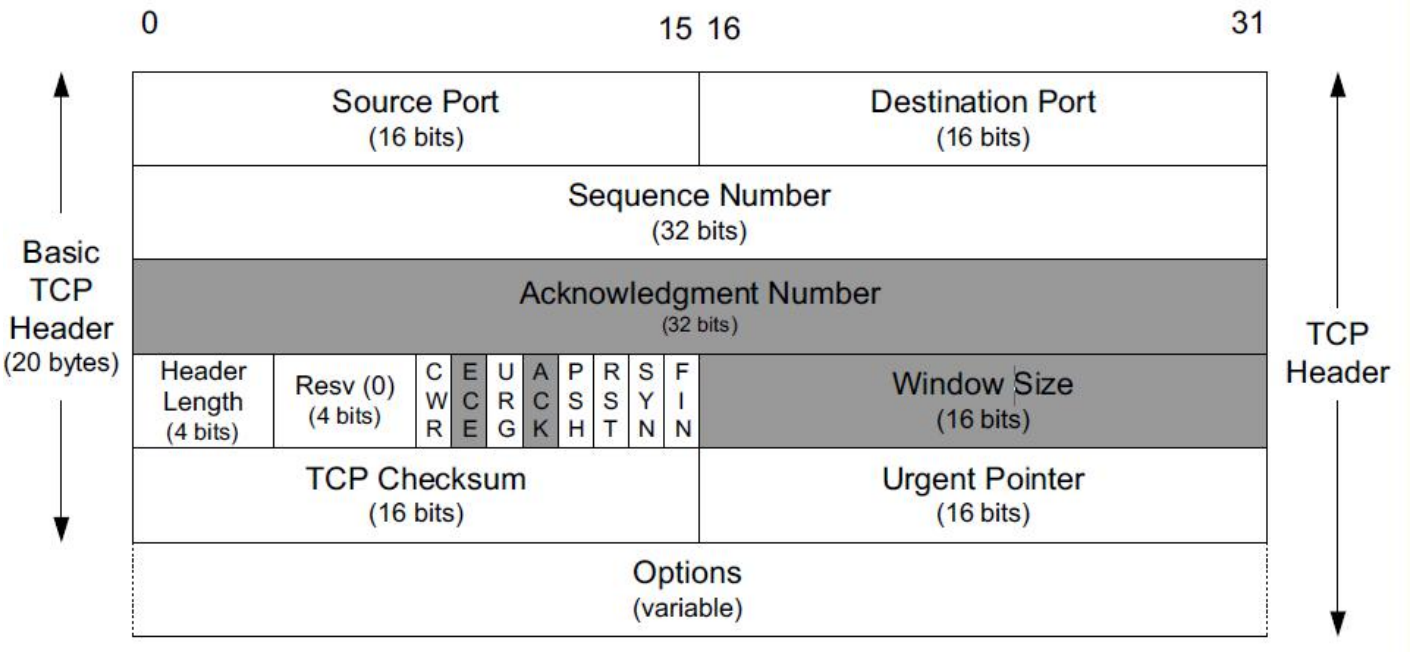
2.5.3 传输控制协议TCP

传输控制协议 TCP (Transmission Control Protocol)一种**基于连接的可靠的稳定的无重复的传输协议。**

1、TCP头部信息

TCP协议头部信息如下：





- **16位源端口号（Source Port）**：发送主机中进程的端口号
- **16位目的端口号（Destination Port）**：接收主机中进程的端口号
- **32位序列号（Sequence Number）**：每一个包中都包含序列号，序列号被系统初始化为某个随机值ISN。后续的TCP报文段中序号加上该报文段所携带数据的第一个字节在整个字节流中的偏移。例如，某个TCP报文段传送的数据是字节流中的第1025～2048字节，那么该报文段的序号值就是ISN+1025
- **32位确认号（Acknowledgment Number）**：目的主机返回确认号，使源主机知道某个或几个报文段已被接收
- **四位首部长度（Header Length）**：由于TCP首部包含一个长度可变的选项部分，所以需要这么一个值来指定这个TCP报文段到底有多长
- **URG标志**：表示紧急指针（urgent pointer）是否有效
- **ACK标志**：表示确认号是否有效。我们称携带ACK标识的TCP报文段为确认报文段

- **PSH标志**：提示接收端应用程序应该立即从TCP接收缓冲区中读走数据，为接收后续数据腾出空间（如果应用程序不将接收到的数据读走，它们就会一直停留在TCP接收缓冲区中）
- **RST标志**：表示要求对方重新建立连接。我们称携带RST标志的TCP报文段为复位报文段
- **SYN标志**：表示请求建立一个连接。我们称携带SYN标志的TCP报文段为同步报文段
- **FIN标志**：表示通知对方本端要关闭连接了。我们称携带FIN标志的TCP报文段为结束报文段
- **16位窗口大小（window size）**：是TCP流量控制的一个手段。这里说的窗口，指的是接收通告窗口（Receiver Window，RWND）。它告诉对方本端的TCP接收缓冲区还能容纳多少字节的数据，这样对方就可以控制发送数据的速度
- **16位校验和（TCP check sum）**：由发送端填充，接收端对TCP报文段执行CRC算法以检验TCP报文段在传输过程中是否损坏。注意，这个校验不仅包括TCP头部，也包括数据部分。这也是TCP可靠传输的一个重要保障。
- **16位紧急指针（urgent pointer）**：是一个正的偏移量。它和序号字段的值相加表示最后一个紧急数据的下一字节的序号。因此，确切地说，这个字段是紧急指针相对当前序号的偏移，不妨称之为紧急偏移。TCP的紧急指针是发送端向接收端发送紧急数据的方法。
- **TCP头部选项**：TCP头部的最后一个选项字段（options）是可变长的可选信息。这部分最多包含40字节

2、TCP运输连接的阶段

- 建立TCP连接
- 数据传输
- 释放TCP连接

> **注意：三报文握手我们也称之为三次握手或者三路握手，四报文挥手我们也称之为四次挥手或者四路挥手**

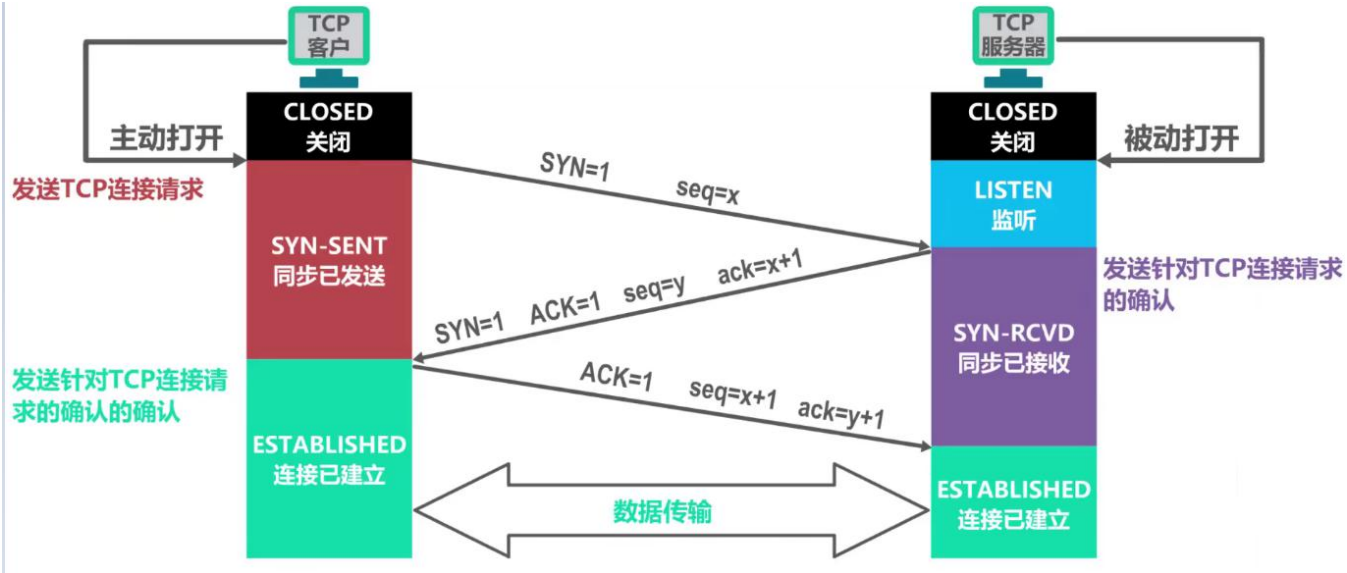
3、TCP连接的建立

TCP连接的建立需要解决以下三个问题：

- 使TCP双方能够确知对方的存在
- 使TCP双方能够协商一些参数（如最大窗口值、是否使用窗口扩大选项和时间戳选项以及服务质量等)
- 使TCP双方能够对运输实体资源（如缓存大小、连接表中的项目等)进行分配

三次握手过程：

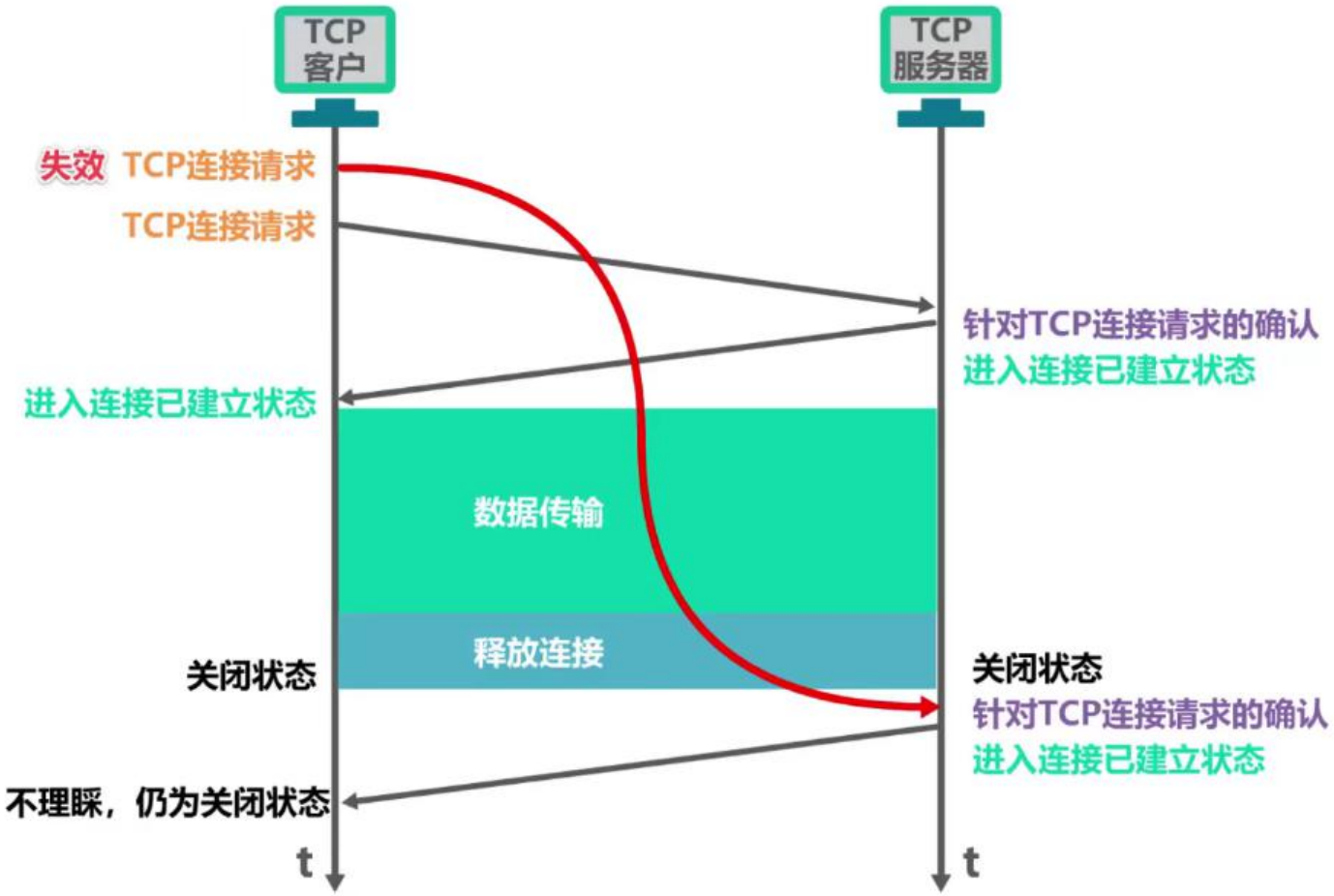
- 由客户端的某个进程主动发起TCP连接建立，最初两端的TCP进程都处于**关闭状态**
- TCP服务器被动等待客户进程的TCP请求，所以TCP服务器进入**监听状态**
- TCP客户进程向TCP服务器进程发送TCP建接请求报文段，并且进入**同步已发送状态**
- TCP连接请求报文段首部中的同步位**SYN**被设置为1，表明这是一个TCP连接请求报文段，32位序列号字段**seq**被设置了一个初始值x，作为TCP客户进程所选择的初始序号。
- TCP服务器进程接收到TCP连接请求报文段后，如果同意建立连接，则向TCP客户进程发送TCP连接请求确认报文段，并且进入**同步已接收状态**，该报文段首部中的同步位**SYN和确认位ACK都设置为1**，表明这是一个TCP连接请求确认报文段。**序号字段seq被设置了一个初始值y**，作为TCP服务器进进程选择的初始序号，确认号字段**ack的值被设置成x+1**，表明接收到了TCP客户进程序号为x的报文。
- TCP客户进程收到TCP连接请求确认报文段后，还要向TCP服务器进程发送一个普通的TCP确认报文段，并且进入连接已建立状态，报文段首部中的**确认位ACK设置为1**，表明这是一个普通的TCP确认报文段。序列号字段**seq**设置为x+1，确认号字段**ack的值被设置成y+1**，表明接收到了TCP服务进程序号为y的报文，TCP服务器进程收到该确认报文段后也进入**连接已建立状态。**



- 思考：假如TCP连接的建立使用两次握手而不是三次握手可以吗？
  - 假设TCP客户进程发送了一个TCP请求报文段，但是该报文段在网络结点中被长时间滞留了，TCP客户采用超时重传机制重发TCP请求报文段并且被TCP服务进程接收，TCP服务进

程发送一个TCP连接请求确认报文段，然后TCP服务进程和客户进程可以进行数据的传输，数据传输完成以后双方都处于关闭状态。

- 随后滞留在网络结点中的那个失效的TCP请求报文段被TCP服务进程接收，TCP服务进程又发送一个TCP连接请求确认报文段，并且进入连接已建立状态，由于TCP客户进程并没有发起新的TCP连接请求，并且已经处于关闭状态了，因此不会理会TCP服务器发送的报文段，但是TCP服务器进程已经进入了连接已建立状态，他认为新的TCP连接已经建立好了，就会一直等待TCP客户进程发来数据，将会浪费TCP服务器主机的很多资源。



- **因此：TCP连接的建立必须采用三次握手。**
- 练习：

  【2011年 题39】 主机甲向主机乙发送一个（SYN=1，seq=11220）的TCP段，期望与主机乙建立TCP连接，若主机乙接受该连接请求，则主机乙向主机甲发送的正确的TCP段可能是

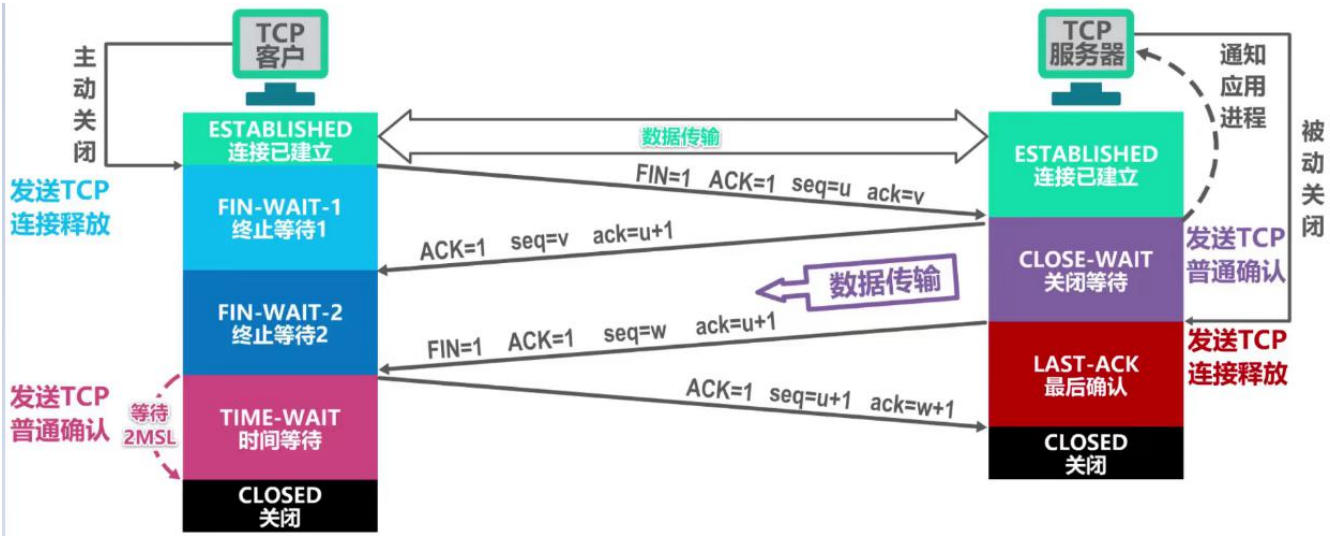
  A. (SYN=0 , ACK=0, seq=11221, ack=11221)　　B. (SYN=1 , ACK=1, seq=11220, ack=11220)

  C. (SYN=1 , ACK=1, seq=11221, ack=11221)　　D. (SYN=0 , ACK=0, seq=11220, ack=11220)

四次挥手过程：TCP通过四次挥手来释放连接，数据通信结束后，TCP双方都可以释放连接

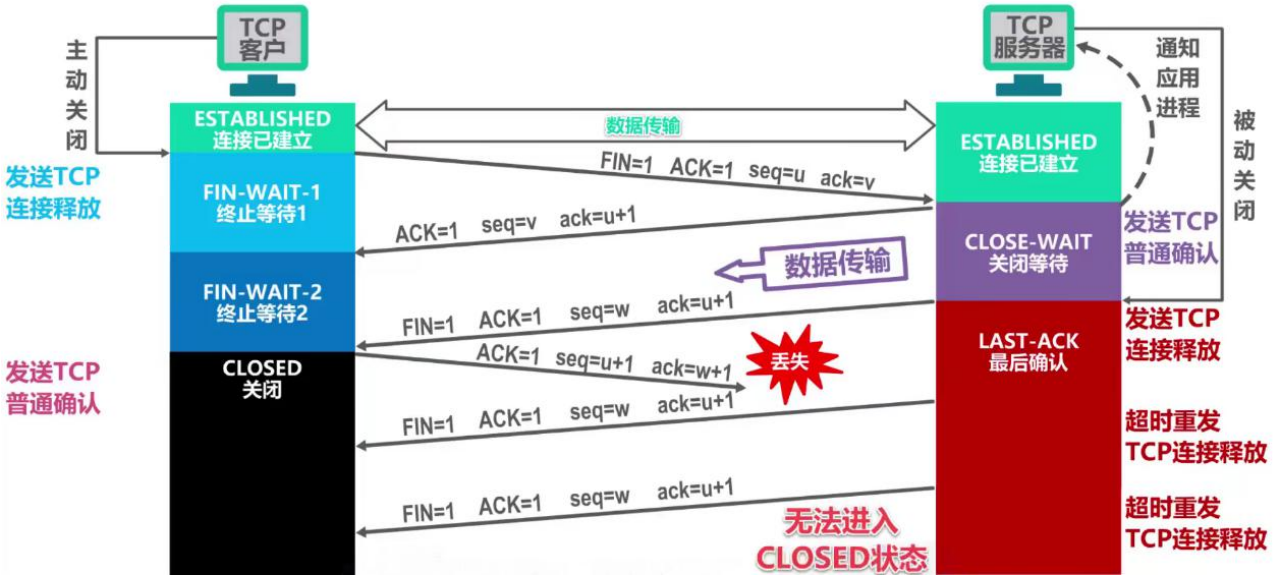
- 假设由客户进程主动关闭TCP连接
- **客户进程发送连接释放报文段**，并且进入**终止等待1状态**，该报文段中的首部终止位FIN设置位1，ACK设置为1，表明这是一个连接释放报文段，同时也对之前接收打报文段进行确认，seq=u表示TCP客户进程之前已发送过的数据的最后一个字节的序号+1，ack=v表示客户进程之前已收到的数据的最后一个字节的序号+1

- 服务器进程接收到TCP连接释放报文段后，会**发送一个普通的TCP确认报文段并且进入关闭等待状态**，此时TCP客户进程到服务器进程这个方向的连接就释放了，这时的TCP连接属于**半关闭状态**，也就是TCP客户进程已经没有数据要发送了，但是TCP服务器进程如果还有数据需要发送，客户进程则还需要接收，这个状态可能会持续一段时间，直到TCP服务器进程将数据发送完毕。
- **TCP客户进程**收到TCP确认报文段后就进入**终止等待2状态**，等待TCP服务器进程发出的TCP连接释放报文段。
- 当TCP服务器进程的没有数据要发送了后，释放连接（被动关闭），**TCP服务器进程发送TCP连接释放报文段并进入最后确认状态。**FIN=1，ACK=1表明这是一个连接释放报文段，seq=w，ack=u+1同时也对之前收到的报文段进行确认。思考：为什么seq=w，而不是seq=v+1呢?
- TCP客户进程收到TCP连接释放报文段后，必须针对该报文段发送普通的TCP确认报文段，之后进入时间等待状态
- TCP服务器进程收到该报文段后就进入关闭状态
- 而TCP客户进程还需要经过**2MSL**后才能进入关闭状态。MSL(Maximum Segment Lifetime)意思是**最长报文段寿命**，RFC793建议为**2分钟**。



- 思考：为什么TCP客户进程还需要经过**2MSL**后才能进入关闭状态?
  - 假设TCP客户进程收到TCP连接释放报文段，并且针对该报文段发送普通的TCP确认报文段后马上进入关闭状态，但是该确认报文段丢失了，TCP服务器进程无法收到该确认报文段，TCP服务器程序就会以为之前发送的TCP连接释放报文段TCP客户进程没有收到，然后TCP服务器进程会对之前所发送的TCP连接释放报文段超时重传，并仍处于最后确认状态。由于TCP客户进程已经处于关闭状态了，不会处理TCP服务器进程发送的连接释放报文段，这样就会造成TCP服务器反复发送TCP连接释放报文段，并且一直处于最后确认状态而无法

进入关闭状态。因此客户端等待2MSL可以确保服务器进程收到最后一个TCP确认报文。



2.5.4 TCP可靠传输的实现

1、分段传输：应用数据根据MSS（Maxitum Segment Size 最大分段大小，这个值TCP协议在实现的时候往往用MTU值代替（需要减去IP数据包包头的大小20Bytes和TCP数据段的包头20Bytes）所以往往MSS为1460 ）值被分割成TCP认为最适合发送的数据段。

> 注意：在运输层的分段传输（分组传输）和网络层的IP数据报分片传输的区别：分段传输只有TCP协议才有的，并且分段的依据为MSS，分片传输的依据为MTU。

2、超时重传：当TCP发出一个段后，它启动一个定时器，等待目的端确认收到这个报文段。如果不能及时收到一个确认，将重发这个报文段。

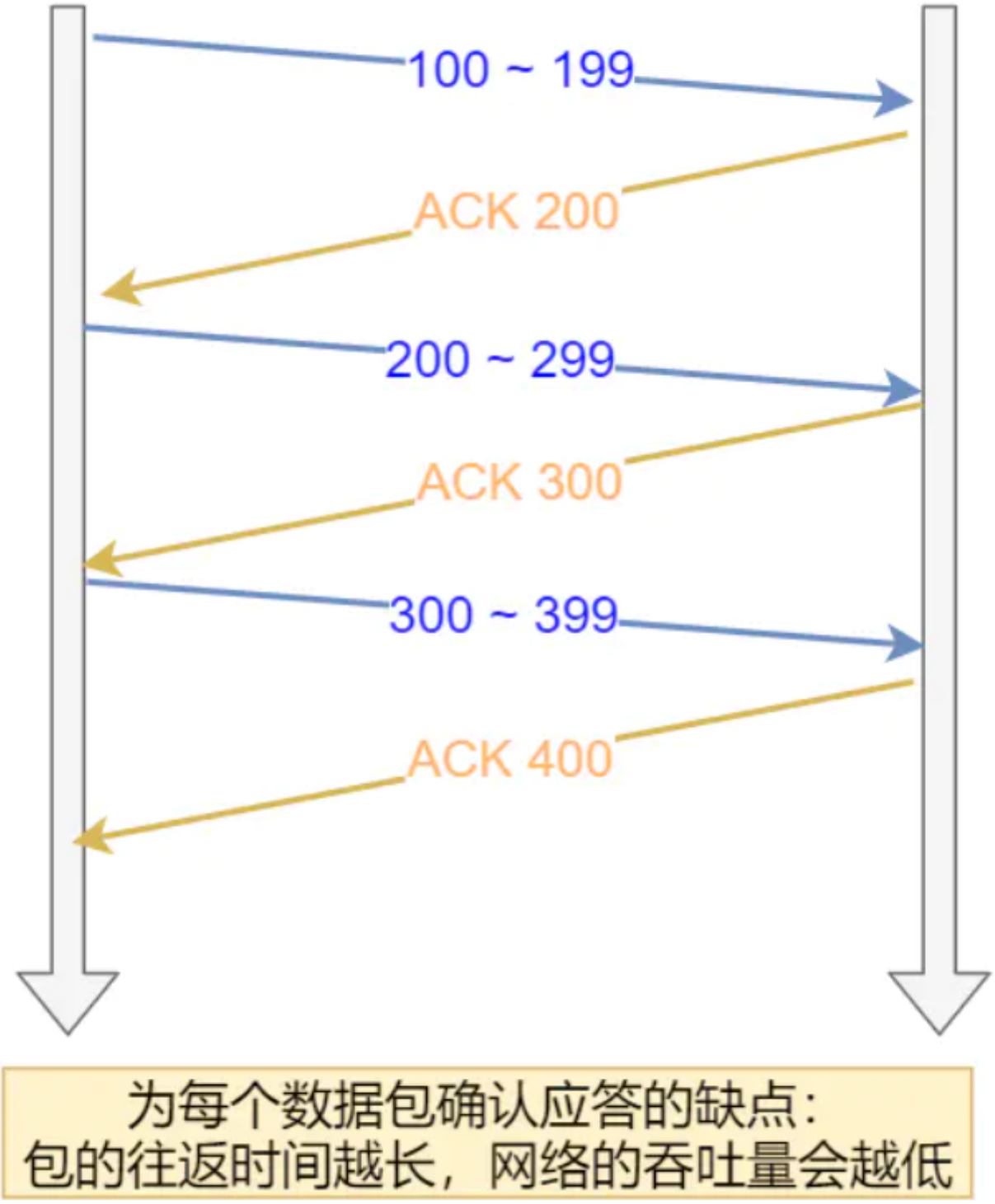
3、CRC校验和：TCP将保持它首部和数据的检验和如果收到段的检验和有差错，TCP将丢弃这个报文段和不确认收到此报文段。

4、流量控制：TCP 提供一种机制可以让「发送方」根据「接收方」的实际接收能力控制发送的数据量，这就是所谓的流量控制。TCP连接的每一方都有固定大小的缓冲空间，TCP的接收端只允许发送端发送接收端缓冲区能接纳的数据。当接收方来不及处理发送方的数据，能提示发送方降低发送的速率，防止包丢失。TCP使用的流量控制协议是**可变大小的滑动窗口协议**。接收方有即时窗口（滑动窗口），随ACK报文发送。（**TCP 利用滑动窗口实现流量控制）**

5、滑动窗口：

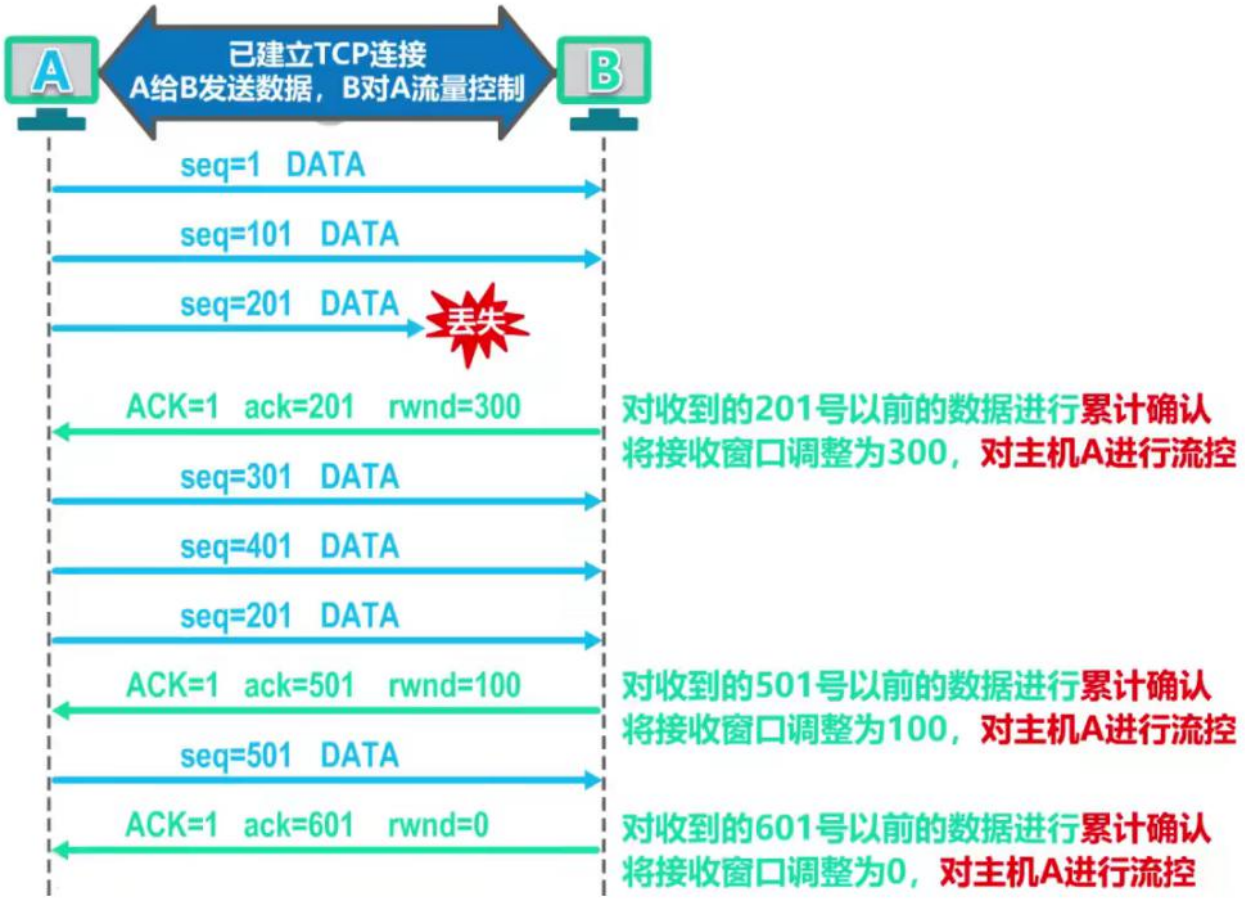
- **应答机制**：假如TCP 每发送一个数据报，都要进行一次确认应答。当上一个数据包收到了应答了， 再发送下一个，这个模式就有点像我和你面对面聊天，你一句我一句。这样的传输方式有一个缺点：**数据包的往返时间越长，通信的效率就越低。**为解决这个问题，我们可以使用累计

应答。



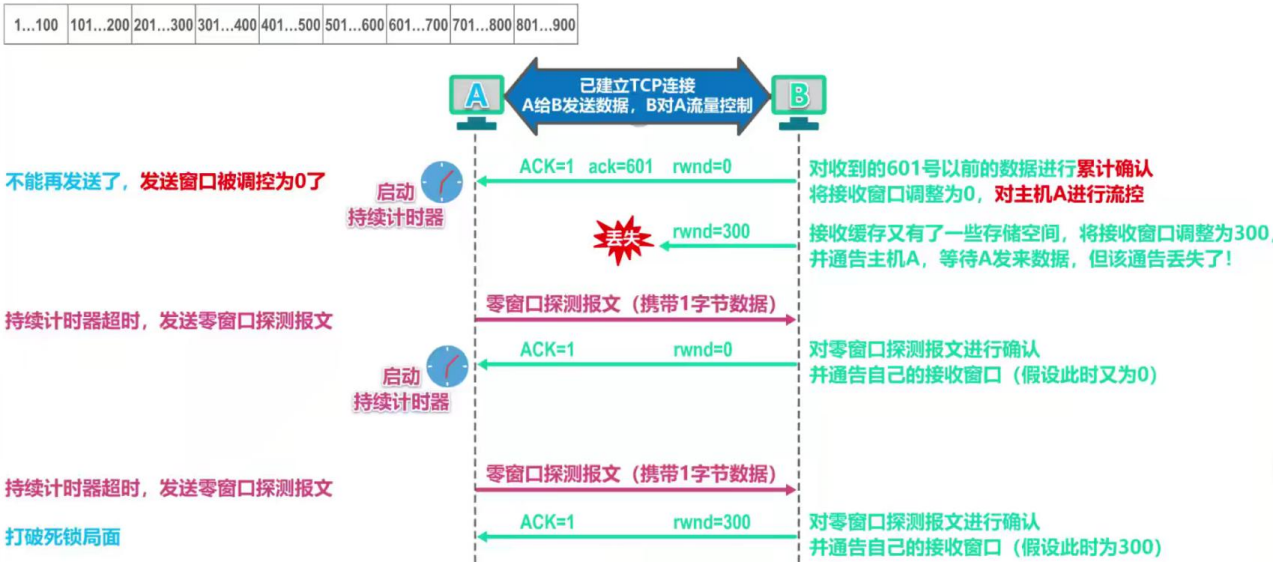
- **累计应答**：接收方在接收到多个数据包后再根据接收到的数据包进行应答，也称为**累计确认**。
- **窗口**：窗口的实现实际上是操作系统开辟的一个**缓存空间**，接收方根据实际情况在应答数据包中告知自己的接收窗口大小。**窗口大小就是指无需等待确认应答，而可以继续发送数据的最大值（以字节为单位）**。发送方主机在等到确认应答返回之前，必须在缓冲区（**发送窗口**）中保留已发送的数据。如果按期收到确认应答，此时数据就可以从缓存区清除。
- **滑动窗口**：如果发送窗口左部的字节已经发送并且收到了确认，那么就将发送窗口向右滑动一定距离，直到左部第一个字节不是已发送并且已确认的状态；接收窗口的滑动类似，接收窗口左部字节已经发送确认并交付主机，就向右滑动接收窗口。
  - 假设主机A发送数据给主机B，在建立TCP连接时，主机B在确认报文中将自己的接收窗口rwnd告知主机A。（假设接收窗口大小为400）

- 主机A根据主机B的接收窗口大小创建自己的发送窗口（在内存上开辟一块空间缓存一个接收窗口大小的数据），并且假设每个数据包中载荷数据为100Bytes
- 主机A分别将第一组数据（为了方便讲解假设seq=1，实际上应该为seq=ISN+1），第二组数据(seq=101)、第三组数据(seq=201)、第四组数据(seq=301)发送给主机B，中间不需要等待主机B的应答数据报（累计应答）
- 假设第三组数据（seq=201）在输出过程中被丢失了，尽管主机B接收到了第四组数据，但是因为累计应答时只应答最大连续报文，所以应答数据包中ack=201表示序号201之前的所有数据全部正确接收。假设主机B将接收窗口大小调整为300，在应答报文中rwnd=300
- 主机A接收到应答数据报后，将自己发送窗口中的序号1~200的数据删除，发送窗口往前（向右）移动并且将大小重新设置为300（开辟接收窗口大小的缓存序号为201-500的数据）
- 主机B将序号为201，301，401的数据报发送给主机B
- 假设以上三组数据报没有丢失，主机B在接收到所有数据后发送应答数据报，ack=501，并且将窗口调整为100，rwnd=100A
- 主机A接收到应答数据报后，将自己发送窗口中的序号201~500的数据删除，发送窗口往前（向右）移动并且将大小重新设置为100（开辟接收窗口大小的缓存序号为501-600的数据）
- 主机A将序号位600的数据报发送给主机B ，按照以上逻辑知道数据发送完毕。



- **接收窗口为0的处理**：当发送窗口被调整为0后，发送方就不能再发送数据了，假如接收方的接收窗口调整为大于0了，如果不采取特殊措施发送方是不知道的，因为接收方不会主动

告知发送方自己接收窗口的大小。这时就需要**持续计数器**了，当发送方接收到接收窗口为0的应答报文时马上启动一个持续计时器，当定时达到时主动向接收方发送一个**零窗口探测报文**，该报文只携带一个字节的数据，然后这种逻辑直到接收方回复的接收窗口大于0。



2.5.5 用户数据报协议UDP

1、UDP概述

UDP 是User Datagram Protocol的简称， 中文名是用户数据包协议。

UDP是一种无连接的不可靠的传输协议（不需要进行三次握手和四次挥手）。

UDP不提供数据包分组、组装和不能对数据包进行排序的缺点，也就是说，当报文发送之后，是无法得知其是否安全完整到达的，因此适合实时数据传输，例如：**IP电话、网络视频会议**等实时应

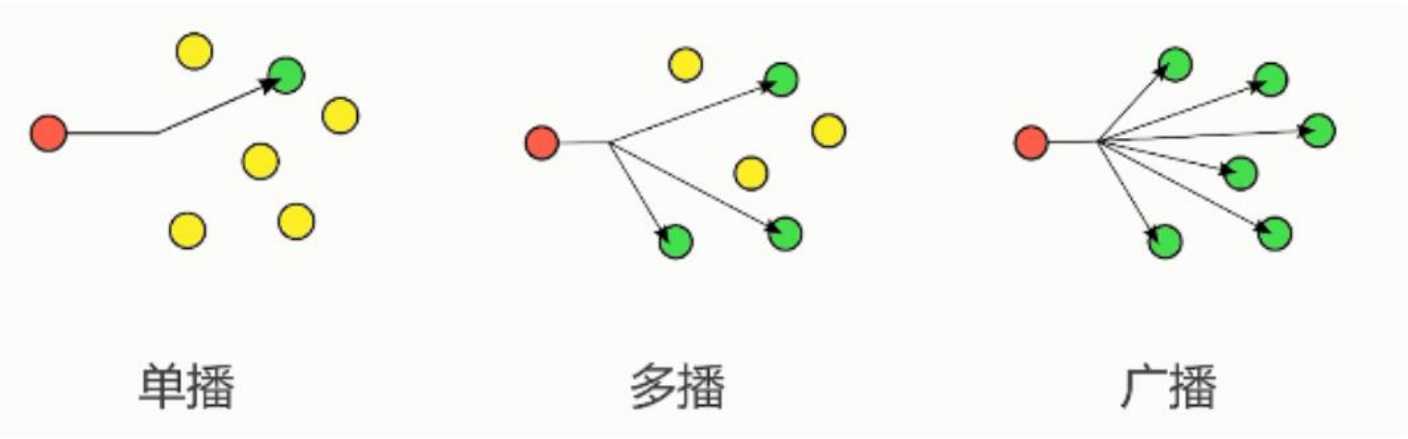
用。

UDP向上层提供无连接不可靠传输服务
（适用于IP电话、视频会议等实时应用）

2、UDP的头部

3、UDP的应用

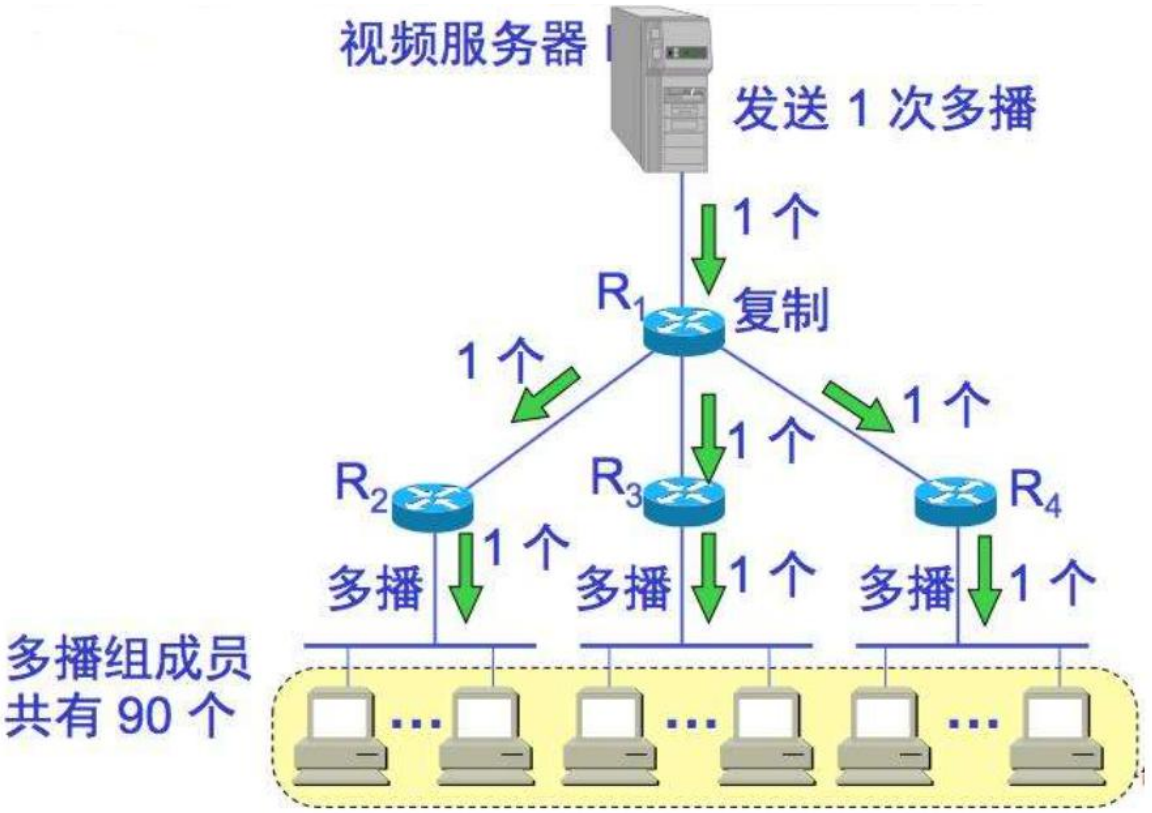
使用UDP运输协议可以进行单播、多播和广播。



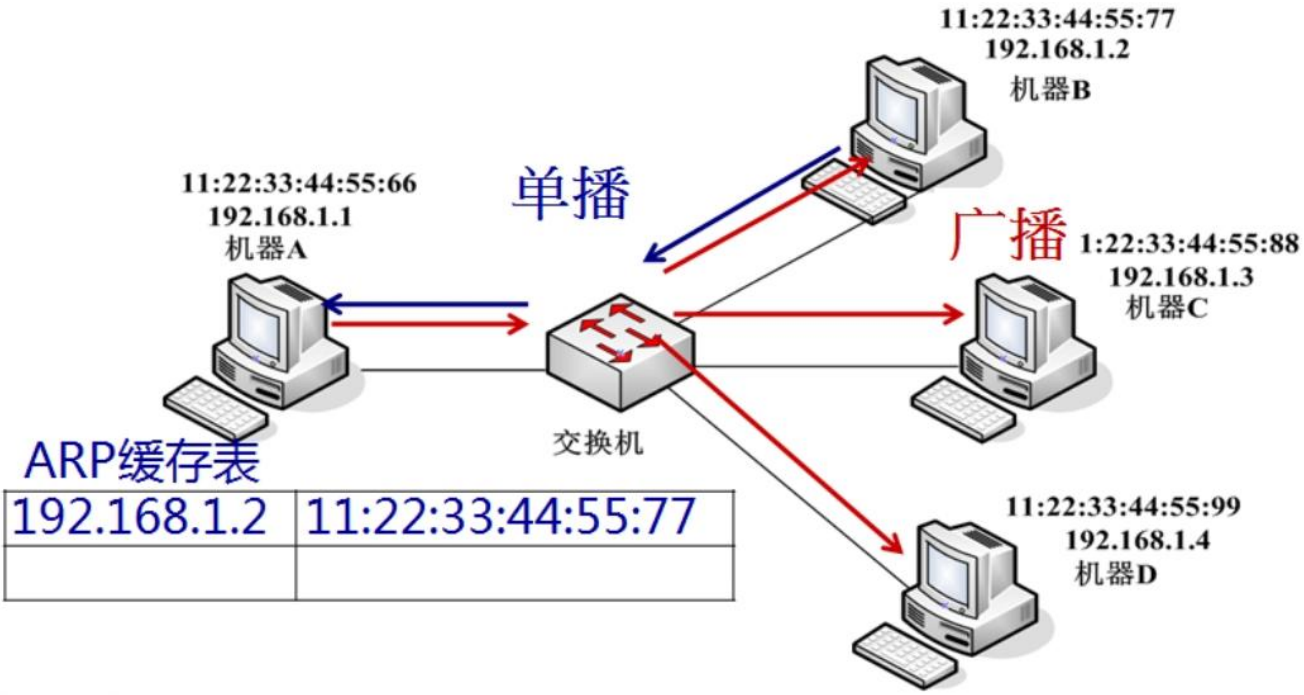
- **单播应用：DNS域名解析**，域名系统是因特网上作为域名和IP(Internet Protocol Address)地址相互映射的一个分布式数据库。
    - 浏览器如何通过域名去查询URL对应的IP（对应服务器地址）呢?
        - 浏览器缓存：浏览器会按照一定的频率缓存DNS记录。
        - 操作系统缓存：如果浏览器缓存中找不到需要的DNS记录，那就去操作系统中找。
        - 路由缓存：路由器也有DNS缓存。
        - ISP的DNS服务器：ISP是互联网服务提供商(Internet Service Provider)的简称，ISP有专门的DNS服务器应对DNS查询请求
        - 根服务器：ISP的DNS服务器还找不到的话，它就会向根服务器发出请求，进行递归查询

    - **当两个网络应用进程间需要快速传输大文件（音视频文件、图片文件等）时也可以使用UDP单播。**
- **多播应用：网络视频会议、教学，视频监控等。**IP多播（也称多址广播或组播）技术，是一种允许一台或多台主机（多播源）发送单一数据包到多台主机（一次的，同时的）的TCP/IP网络技术。IP多播通信必须依赖于IP多播地址，在IPv4中它是一个D类IP地址，范围从**224.0.0.0**到

239.255.255.255。



- **广播的应用：ARP数据报广播**



> 注意：在进行UDP编程时，UDP包的大小可以达到64k，但实际上MTU大小只有1k多，如果直接发一个超过MTU大小的包，就会在网络层被分片，这样的问题是，**如果只要有一个分片在传输中出错了即校验不正确（这是较容易发生的），整个传输的udp包就被丢弃。**注意是整个而不是单个分片。这就是为什么**发送UDP包通常也是1k多**大小的原因。

2.5.6 TCP和UDP的区别

1、对比

	UDP	TCP
是否连接	无连接	面向连接
是否可靠	不可靠传输，不使用流量控制和拥塞控制	可靠传输，使用流量控制和拥塞控制
连接对象个数	支持一对一，一对多，多对一和多对多交互通信	只能是一对一通信
传输方式	面向报文	面向字节流
首部开销	首部开销小，仅8字节	首部最小20字节，最大60字节
适用场景	适用于实时应用（IP电话、视频会议、直播等）	适用于要求可靠传输的应用，例如文件传输

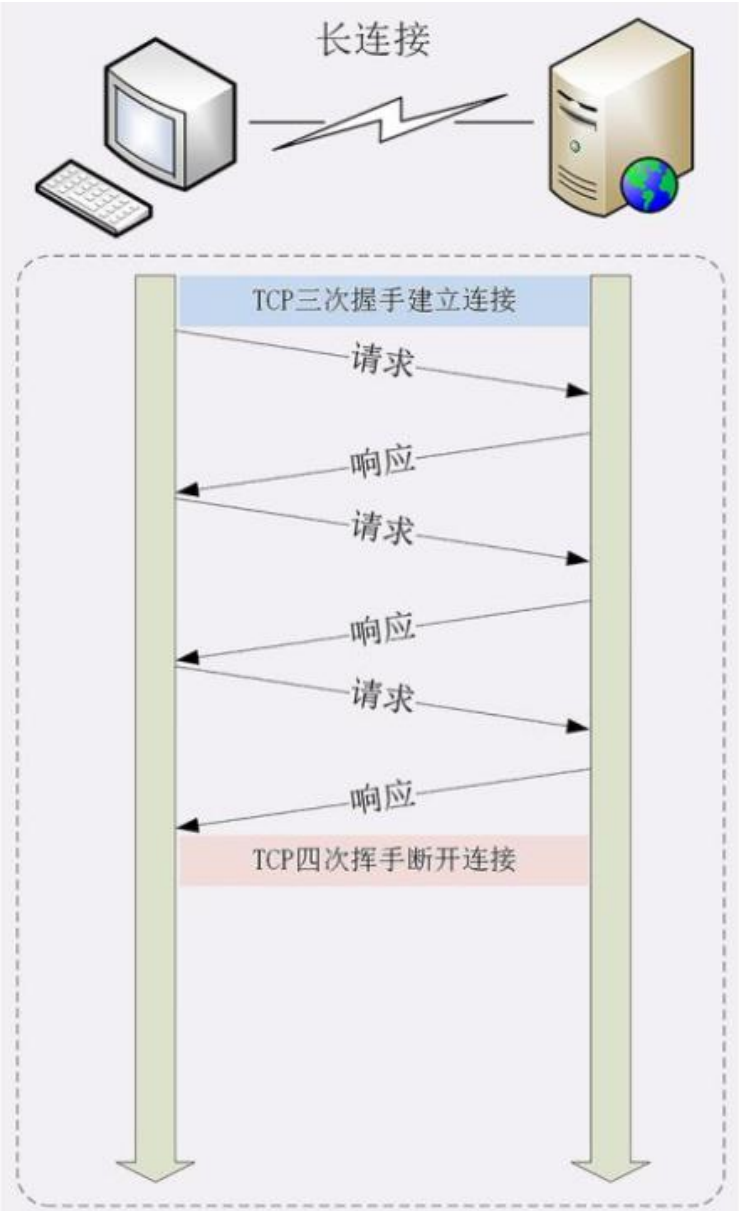
除了上图所展示的区别外，使用TCP传输协议时，一旦建立好TCP连接后，系统需要实时的维护该连接，所以TCP所消耗的系统资源比UDP要多。

另外因为TCP的可靠性传输机制导致TCP传输数据时比UDP要慢的多。

2、TCP的长连接和短链接

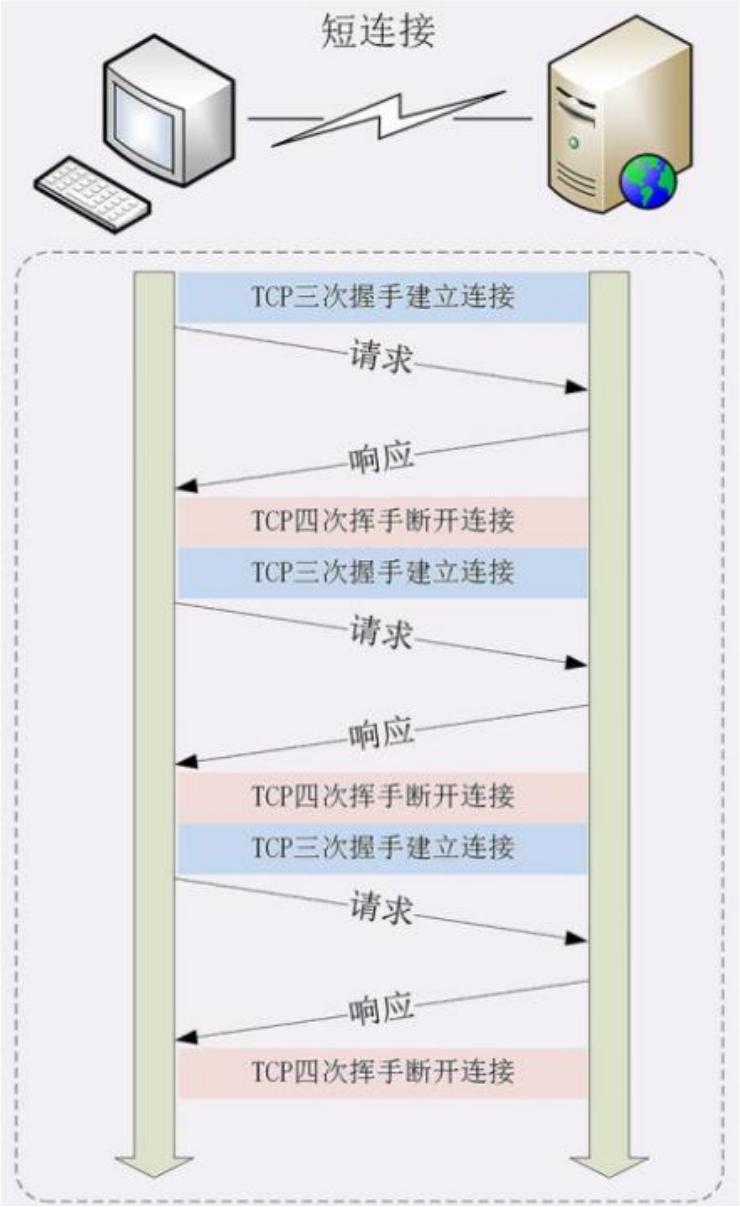
- **长连接**：TCP通信双方在建立好连接后，在较长一段时间内保持连接，直至某一方主动关闭连接。**长连接多用于操作频繁，点对点的通讯**，例如在物联网开发中某下位机需要定时地频繁向

服务发送数据等。



- **短链接**：通信双方有数据交互时，就建立一个TCP连接，数据发送完成后，则断开此TCP连接。短连接多用于操作不频繁，点对点的通讯，例如：在HTTP/1.0中默认使用短连接。也就是说，

客户端和服务器每进行一次HTTP操作，就建立一次连接，任务结束就中断连接。



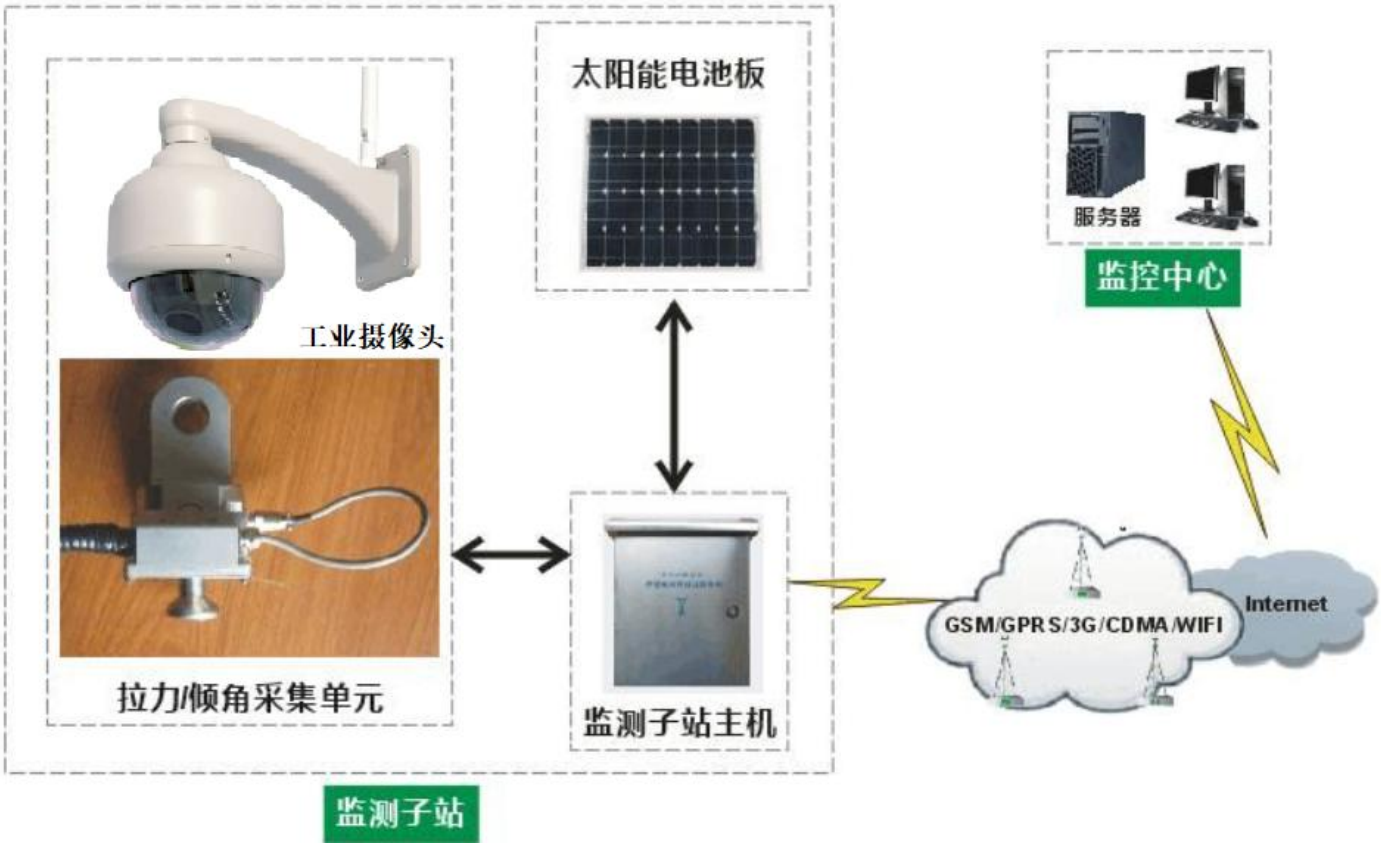
- **长连接短链接对比**

连接	优点	缺点
长连接	传输速度快，server可以主动发送数据给client	保持的连接会占用和多系统资源， 后台设计相对要复杂
短连接	不需要占用系统的太多的资源使得server可以处理更多client的connect	发送小数据划不来，比较耗时。server无法主动发送数据到client

3、UDP的灵活应用

应用项目：高压线覆冰监测系统。

监控中心为了较快显示工业摄像头采集到的高清照片，监控子站主机需要使用UDP与监控中心主机进行图像数据的传输，但是因为使用了UDP传输协议可能会导致某些数据报丢失。为了保证图像数据的完整性，我们需要在应用层设计私有协议保障丢失的数据报能够进行重传。

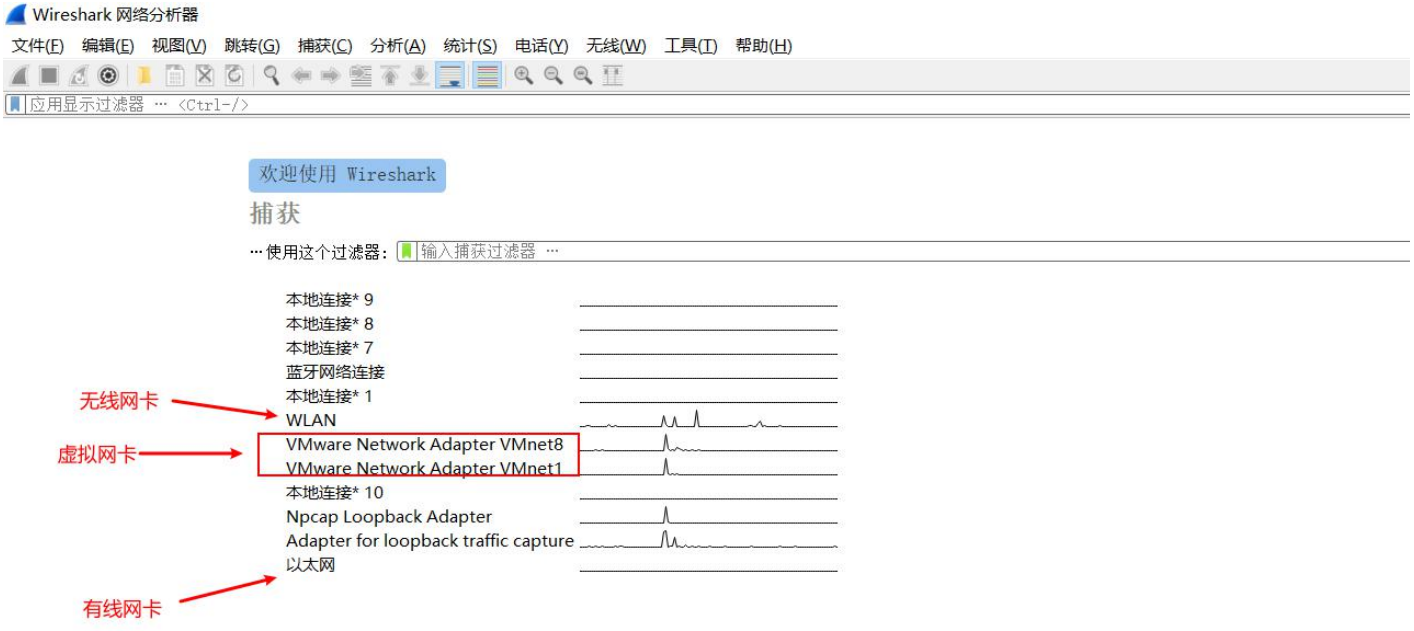


2.6 wireshark

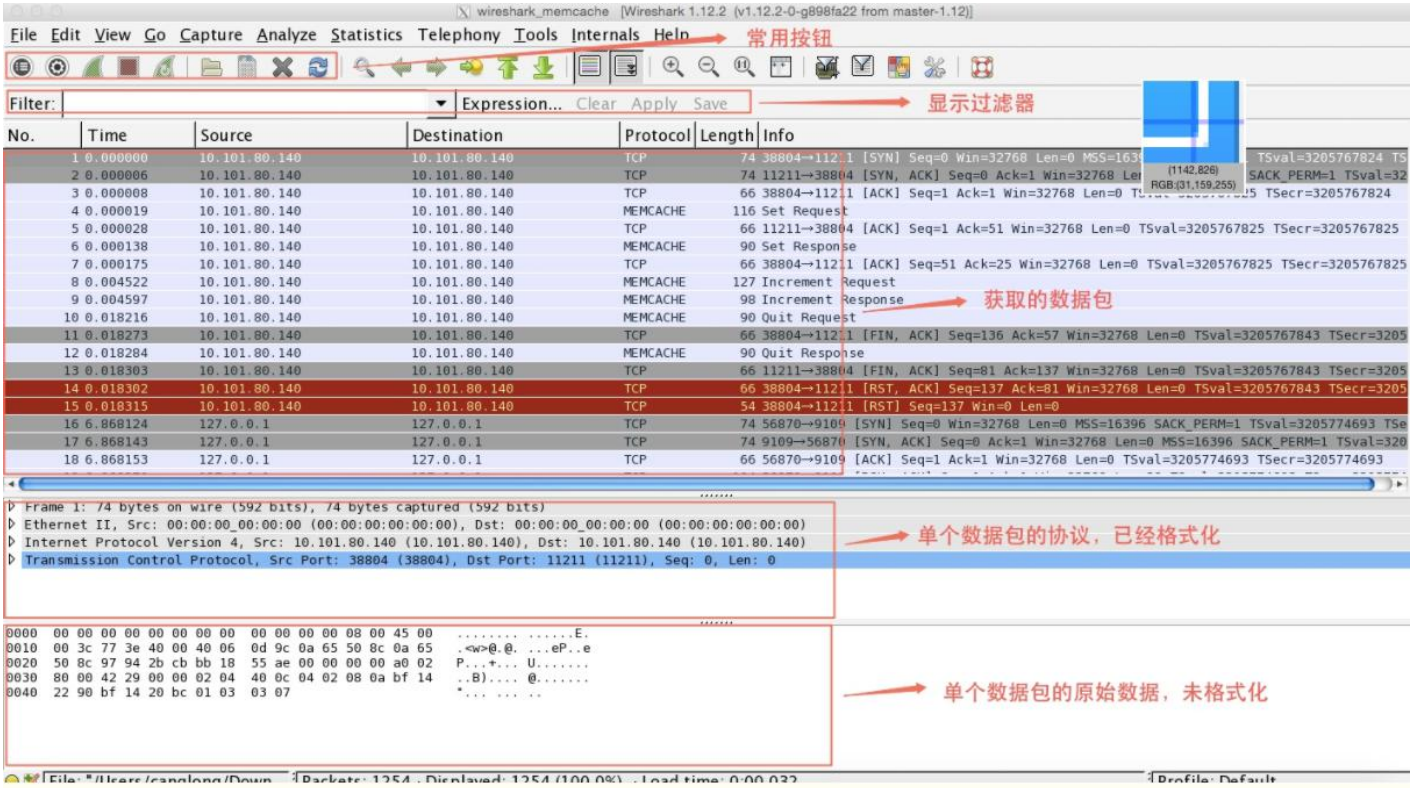
2.6.1 wireshark的安装

2.6.2 界面介绍

1、网卡选择



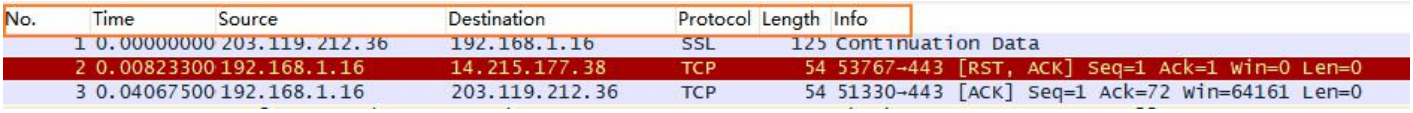
2、首页



常用按钮从左到右的功能依次是:

- 列出可用接口。
- 抓包时需要设置的一些选项。一般会保留最后一次的设置结果。
- 开始新的一次抓包。
- 暂停抓包。
- 继续进行本次抓包。
- 打开抓包文件。可以打开之前抓包保存后的文件。不仅可以打开wireshark软件保存的文件，也可以打开tcpdump使用-w参数保存的文件。
- 保存文件。把本次抓包或者分析的结果进行保存。
- 关闭打开的文件。文件被关闭后，就会切换到初始界面。
- 重载抓包文件。

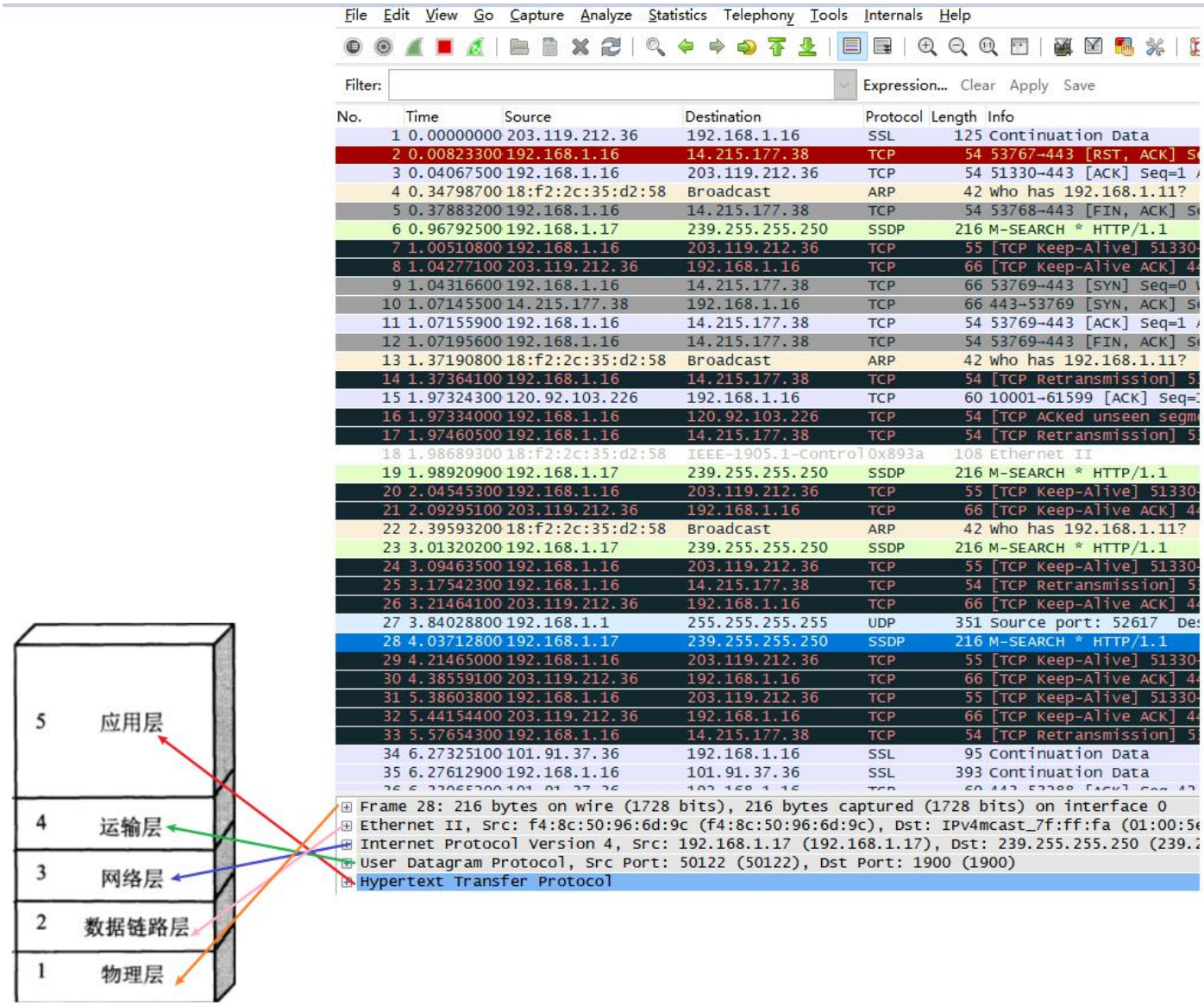
3、数据包列表， 显示捕获到的数据包，每个**数据包编号，时间戳，源地址，目标地址，协议，长度，以及数据包信息。**



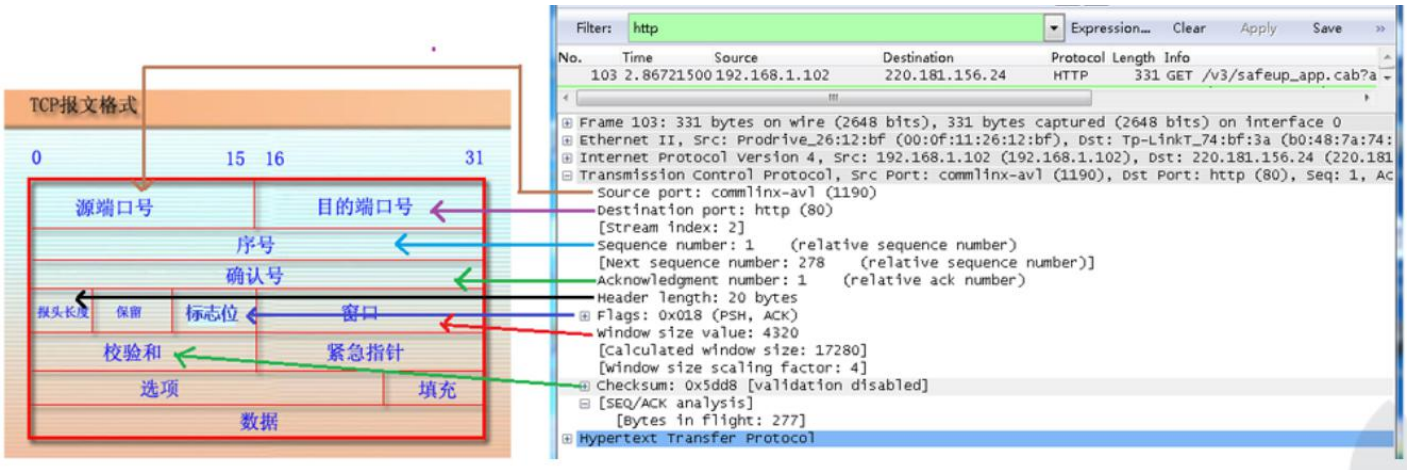
4、数据包详细信息，在数据包列表中选择指定数据包，在数据包详细信息中会显示数据包的所有详细信息内容。数据包详细信息面板是最重要的，用来查看协议中的每一个字段。各行信息分别为

- Frame: 物理层的数据帧概况
- Ethernet II: 数据链路层以太网帧头部信息
- Internet Protocol Version 4: 互联网层IP包头部信息
- User Datagram Protocol: 传输层的数据段头部信息，此处是UDP

- Hypertext Transfer Protocol: 应用层的信息，此处是HTTP协议



5、我们可以在过滤器窗口筛选出http协议的相关数据包，并且查看TCP包的每个字段



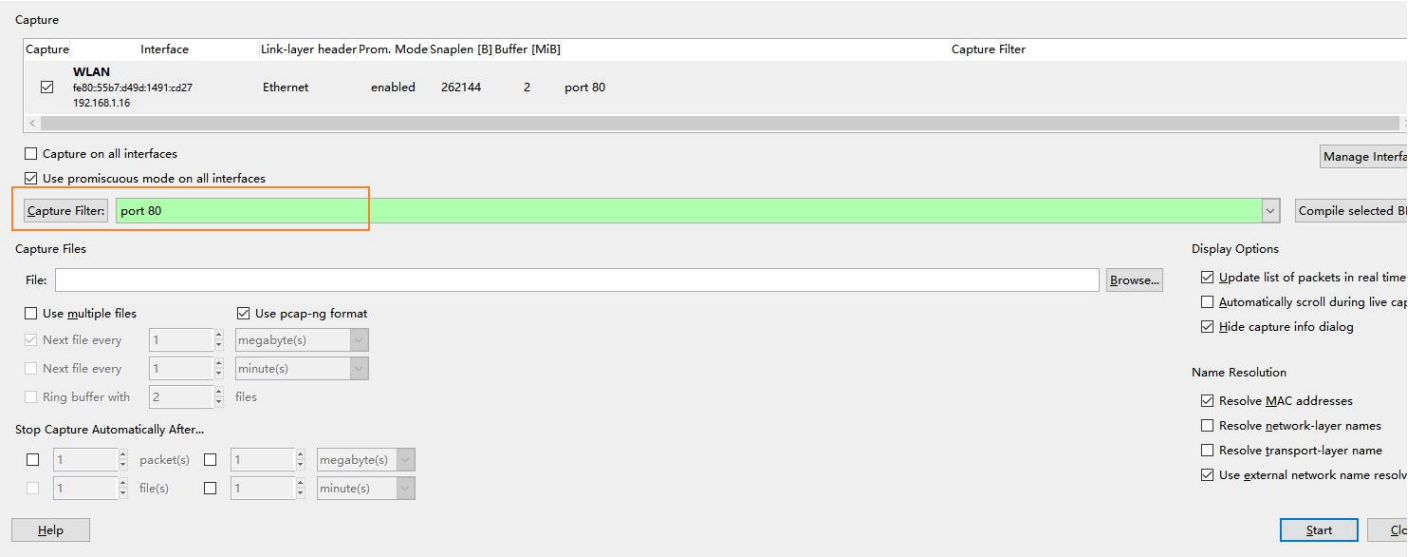
2.6.3 wireshark过滤器

初学者使用wireshark时，将会得到大量的冗余数据包列表，以至于很难找到自己自己抓取的数据包部分。wireshar工具中自带了两种类型的过滤器，学会使用这两种过滤器会帮助我们在大量的数据

中迅速找到我们需要的信息。

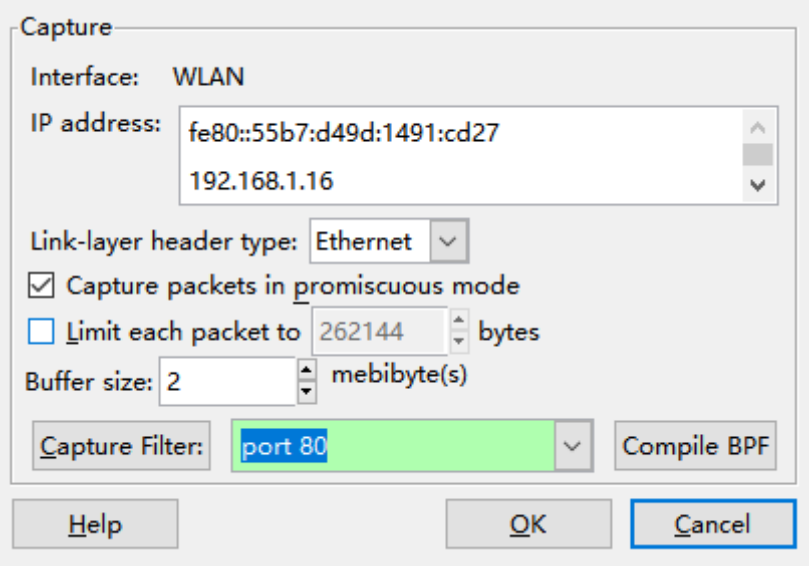
1、设置数据抓取选项

点击常用按钮中的设置按钮，就会弹出设置选项对话框。在这个对话框中我们可以选中需要监听的接口，设置混杂模式，设置抓取数据包的过滤条件。如下图：



首先，选中需要监听获取数据包的接口。接口列表区列出了所有可以使用的接口。如果接口前面的复选框被选中，说明对这个接口监听捕获数据包。

其次，设置混杂模式。设置混杂模式的作用是将网卡设置到混杂模式。如果不设置混杂模式，你的计算机只能获取数据包发往的目标是你计算机和从你计算机出去的数据包。如果设置了混杂模式，你就可以捕获局域网中所有的数据包。如果窗口中的 "Use promiscuous mode on all interfaces"前面的复选框被选中，说明对所有的接口使用混杂模式。如果想单独设置，可以双击接口列表中的接口，会弹出如下的对话框。然后选中或者去掉“Capture packets in promiscuous mode”前面复选框。然后点ok按钮。

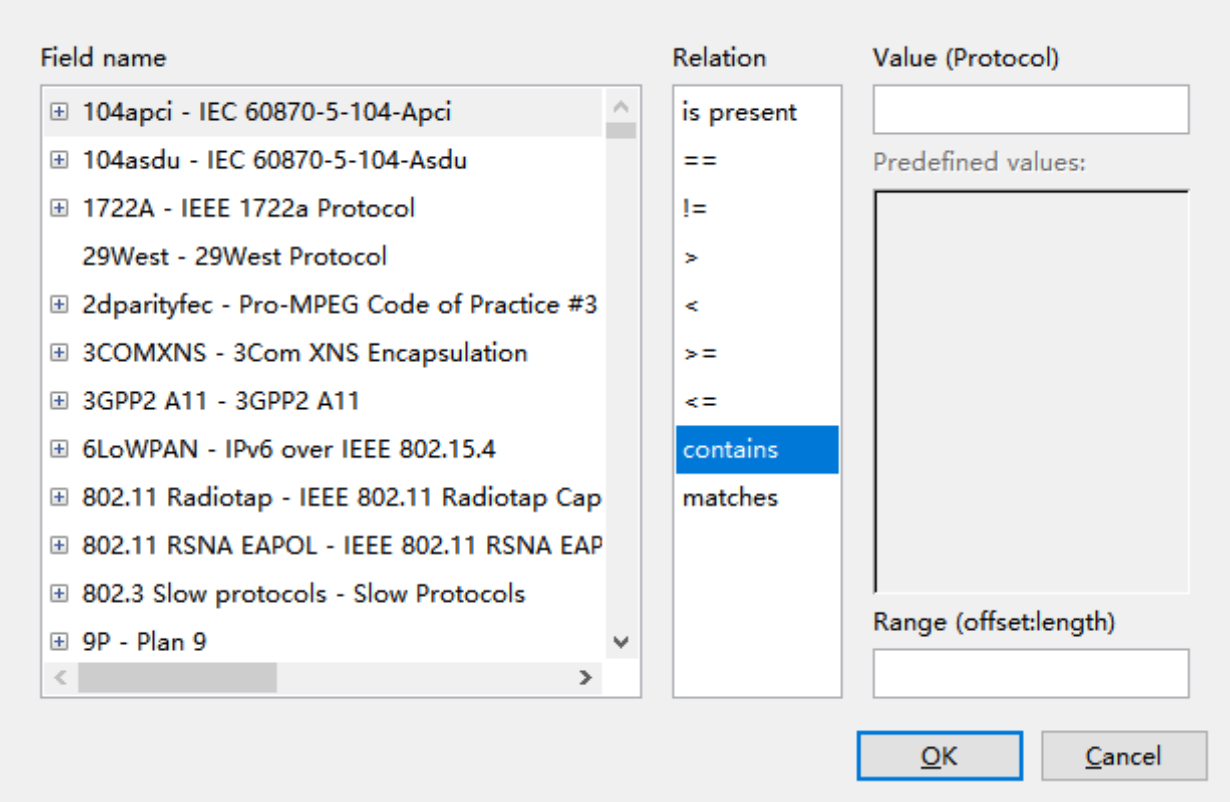


2、显示过滤器

显示过滤器应用于捕获文件，用来告诉wireshark只显示那些符合过滤条件的数据包。显示过滤器比捕获过滤器更常用。他可以用来过滤不想看到的数据包，但是不会把数据删除。如果想恢复原状，

只要把过滤条件删除即可。

过滤器表达式对话框，是的wireshark的可以很简单的设置过滤表达式。点击“Expression”按钮就可以打开这个对话框。如下图：



对话框分左中右三部分。左边为可以使用的所有协议域（过滤项）。右边为和协议域相关的条件值（过滤值）。中间为协议域与条件值之间的关系（过滤关系）。

一条基本的表达式由过滤项、过滤关系、过滤值三项组成。

比如：http contains baidu.com， http为过滤项， contains 为过滤关系， baidu.com 为过滤值，表示显示http协议包中包含关键词“baidu.com”的所有数据包

比如：ip.addr == 192.168.1.1，ip.addr是过滤项、==是过滤关系，192.168.1.1是过滤值（整条表达示的意思是找出所有ip协议中源或目标ip、等于、192.168.1.1的数据包）

3、过滤关系

过滤关系就是大于、小于、等于等几种等式关系，我们可以直接看官方表格，注意其中有“English”和“C-like”两个字段，这个意思是说“English”和“C-like”这两种写法在wireshark中是等价的、都是可用的。

Table 6.4. Display Filter comparison operators

English	C-like	Description and example
eq	==	Equal. `ip.src==10.0.0.5`
ne	!=	Not equal. `ip.src!=10.0.0.5`
gt	>	Greater than. `frame.len > 10`
lt	<	Less than. `frame.len < 128`
ge	>=	Greater than or equal to. `frame.len ge 0x100`
le	<=	Less than or equal to. `frame.len <= 0x20`
contains		Protocol, field or slice contains a value. `sip.To contains "a1762"`
matches	~	Protocol or text field match Perl regualar expression. `http.host matches "acme\.(org|com|net)"`
bitwise_and	&	Compare bit field value. `tcp.flags & 0x02`

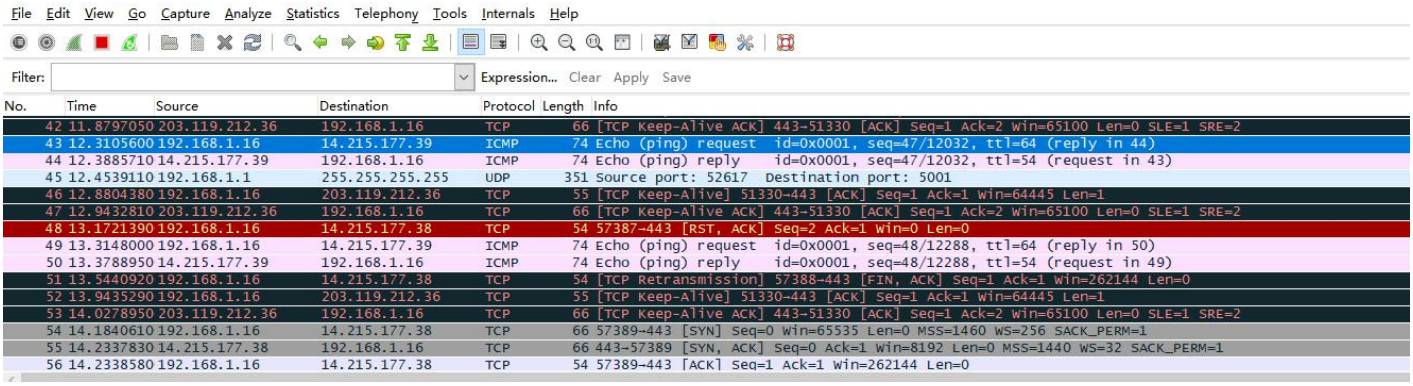
4、复合过滤表达式

所谓复合过滤表达式，就是指由多条基本过滤表达式组合而成的表达示。基本过滤表达式的写法还是不变的，复合过滤表达示多出来的东西就只是基本过滤表达示的“连接词”。我们依然直接参照官方表格，同样“English”和“C-like”这两个字段还是说明这两种写法在wireshark中是等价的、都是可用的。

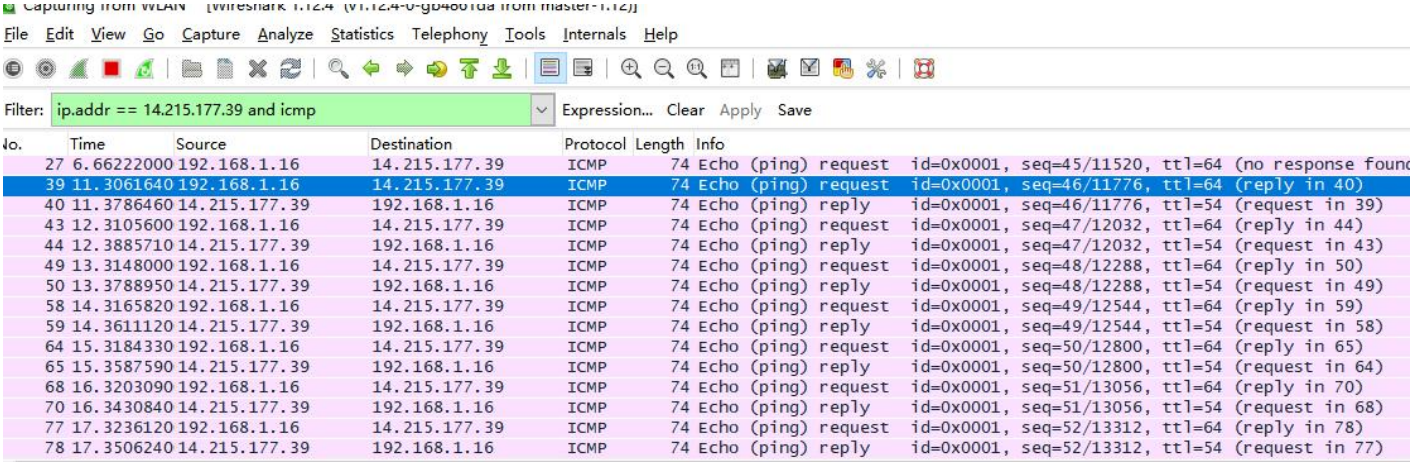
Table 6.5. Display Filter Logical Operations

English	C-like	Description and example
and	&&	Logical AND. `ip.src==10.0.0.5 and tcp.flags.fin`
or	||	Logical OR. `ip.scr==10.0.0.5 or ip.src==192.1.1.1`
xor	^^	Logical XOR. `tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29`
not	!	Logical NOT. `not llc`
[...]		See "Slice Operator" below.
in		See "Membership Operator" below.

5、执行ping www.baidu.com获取的数据包列表如下



观察上述获取的数据包列表，含有大量的无效数据。这时可以通过设置显示器过滤条件进行提取分析信息。ip.addr == 14.215.177.39 and icmp。并进行过滤

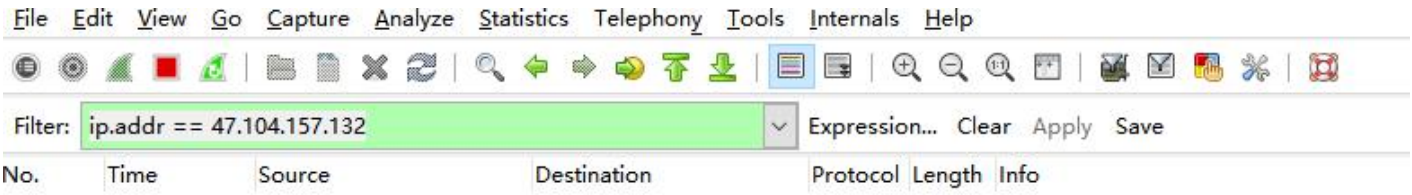


6、常见用显示过滤需求及其对应表达式

- 数据链路层：
  - 筛选mac地址为04:f9:38:ad:13:26的数据包：eth.src == 04:f9:38:ad:13:26
  - 筛选源mac地址为04:f9:38:ad:13:26的数据包：eth.src == 04:f9:38:ad:13:26
- 网络层：
  - 筛选ip地址为192.168.1.1的数据包：ip.addr == 192.168.1.1
  - 筛选192.168.1.0网段的数据： ip contains 192.168.1
  - 筛选192.168.1.1和192.168.1.2之间的数据包：ip.addr == 192.168.1.1 && ip.addr == 192.168.1.2
  - 筛选从192.168.1.1到192.168.1.2的数据包：ip.src == 192.168.1.1 && ip.dst == 192.168.1.2
- 传输层：
  - 筛选tcp协议的数据包：tcp
  - 筛选除tcp协议以外的数据包:!tcp
  - 筛选端口为80的数据包：tcp.port == 80
  - 筛选12345端口和80端口之间的数据包：tcp.port == 12345 && tcp.port == 80
  - 筛选从12345端口到80端口的数据包：tcp.srcport == 12345 && tcp.dstport == 80
- 应用层：
  - 特别说明：http中http.request表示请求头中的第一行（如GET index.jsp HTTP/1.1），http.response表示响应头中的第一行（如HTTP/1.1 200 OK），其他头部都用http.header_name形式。
  - 筛选url中包含.php的http数据包：http.request.uri contains .php
  - 筛选内容包含username的http数据包：http contains username

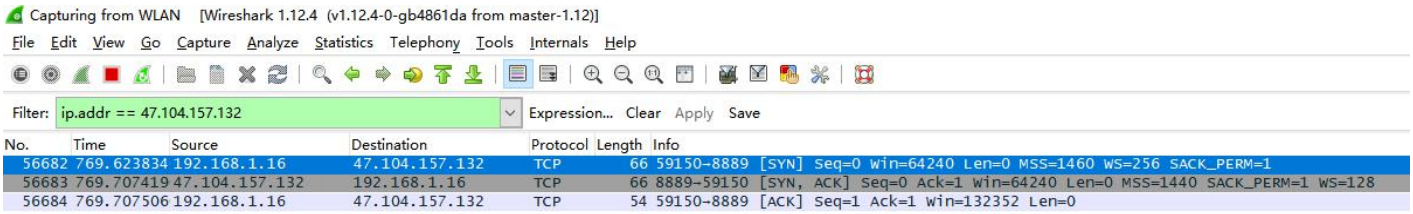
2.6.4 使用wireshark分析TCP三次握手

1、设置过滤阿里云服务器的IP地址：ip.addr == 47.104.157.132

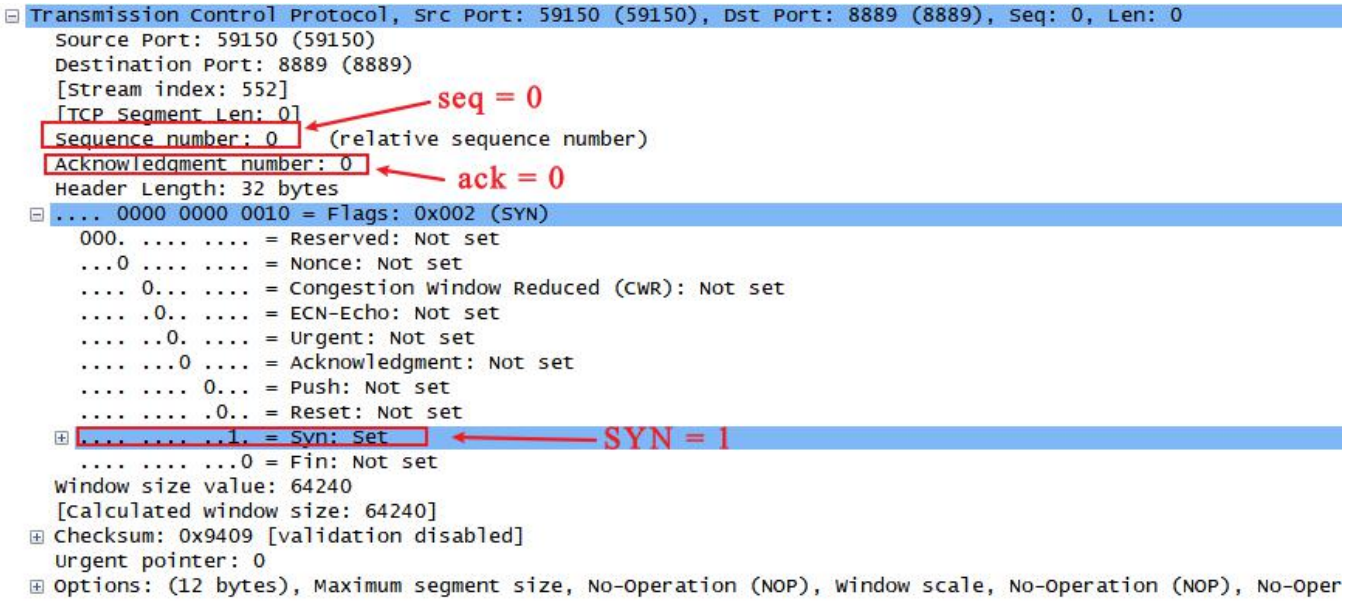


2、使用TCP调试助手连接阿里云服务器上的TCP服务端

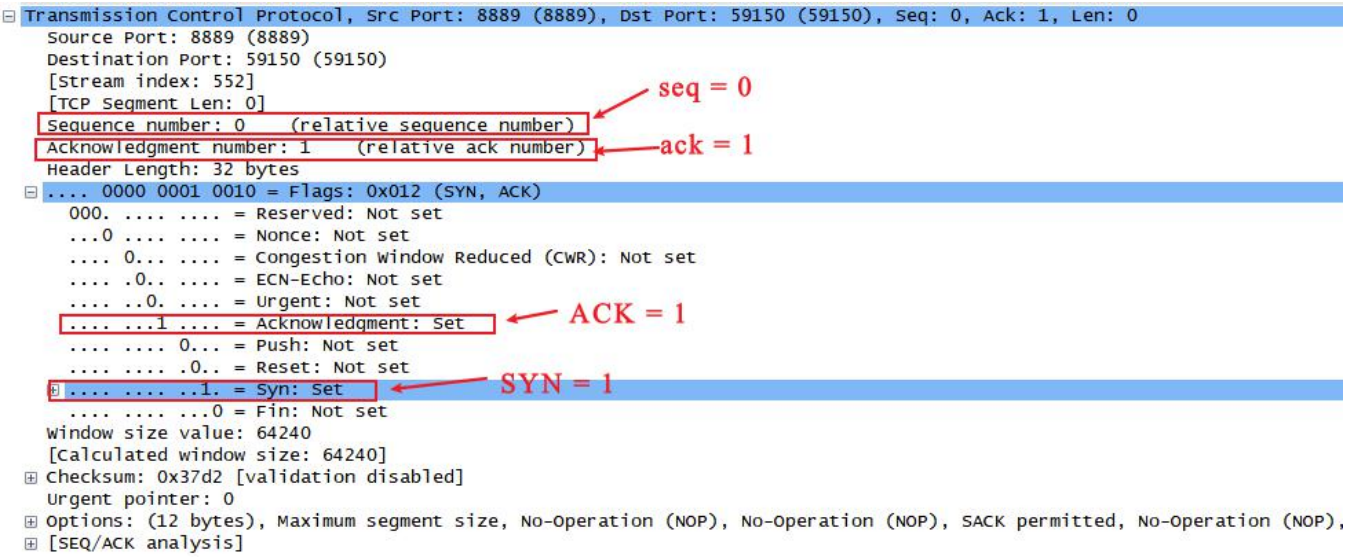
3、我们发现wireshark捕获到了3条TCP数据



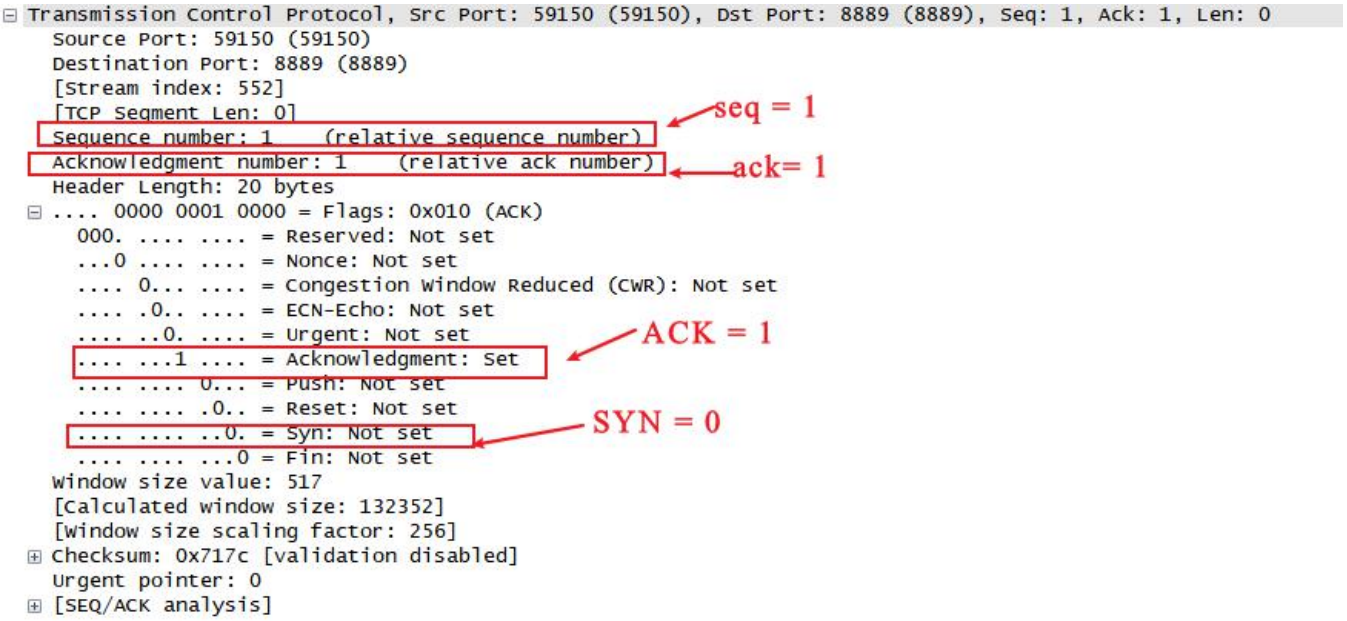
- 第一次握手数据包：客户端发送一个TCP，标志位为SYN，序列号为0， 代表客户端请求建立连接。 如下图



- 第二次握手：服务器发回确认包, 标志位为 SYN,ACK. 将确认序号(Acknowledgement Number)设置为客户的seq加1以.即0+1=1, 如下图

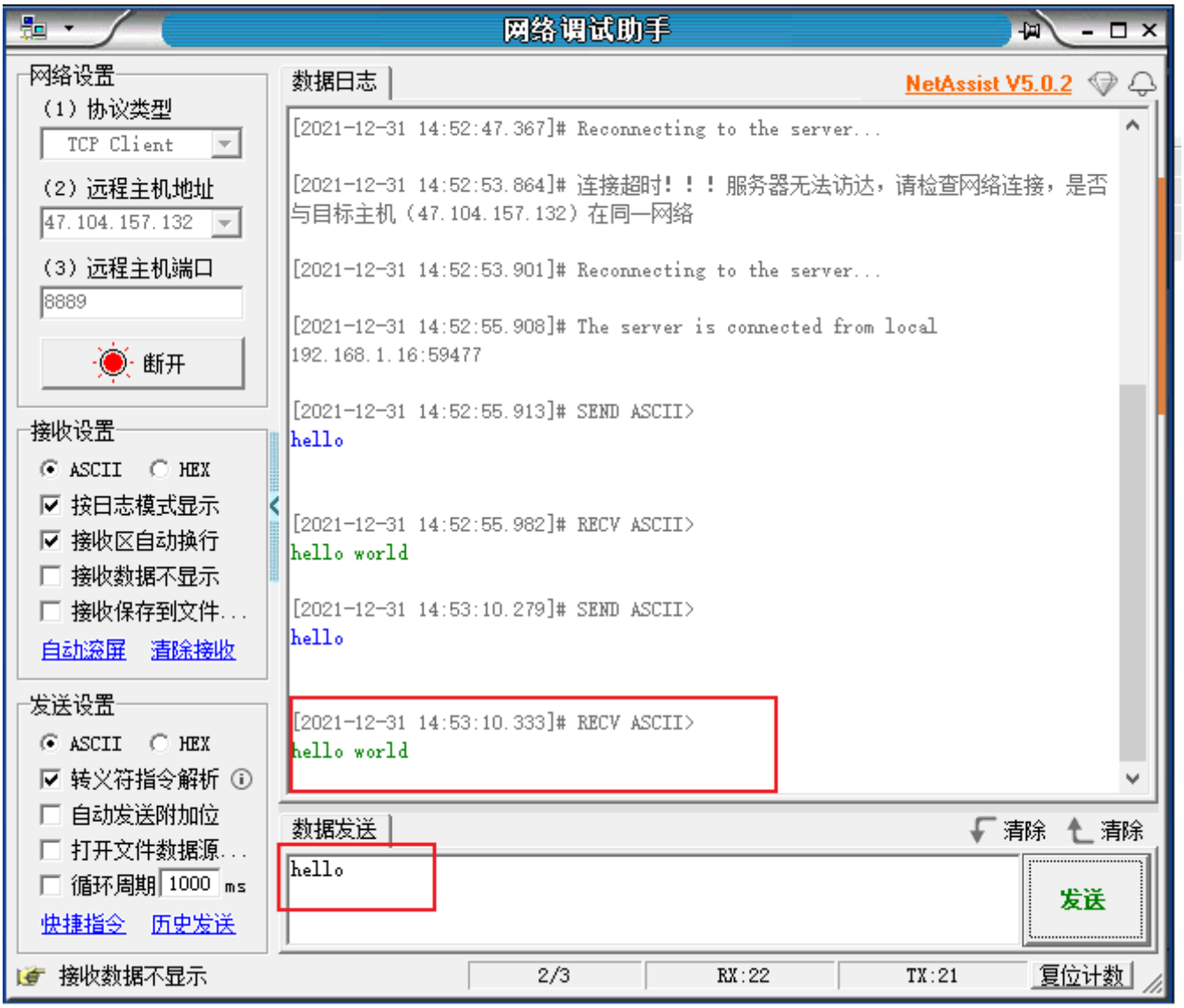


- 第三次握手：客户端再次发送确认包(ACK) SYN标志位为0,ACK标志位为1，如下图:

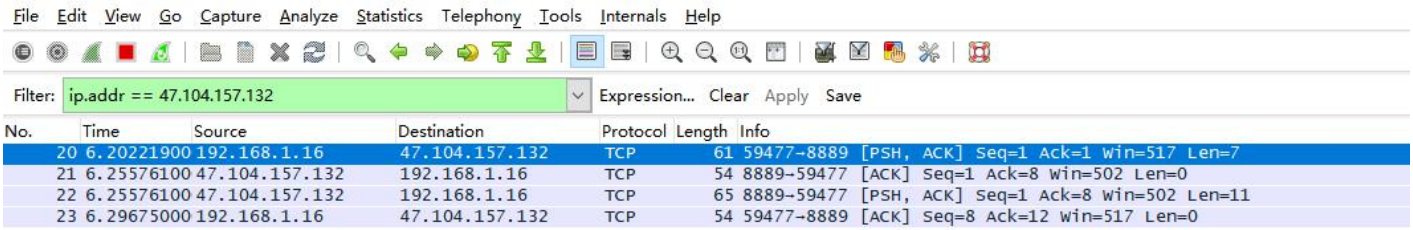


2.6.5 使用wireshark分析TCP数据包

1、使用tcp调试助手向阿里云服务器TCP服务端发送hello，并且接收到服务器返回的hello world



2、我们发现wireshark捕获到了四条TCP数据



2.7 应用层

2.7.1 应用层概述

应用层是网络体系结构中的最上层，所有的网络应用程序和服务都工作在应用层。

常见的网络应用程序：浏览器、手游、即时通信软件、网站的后台程序、网游的主程等。

常见的网络服务：telnet、FTP、ssh等。

2.7.2 万维网

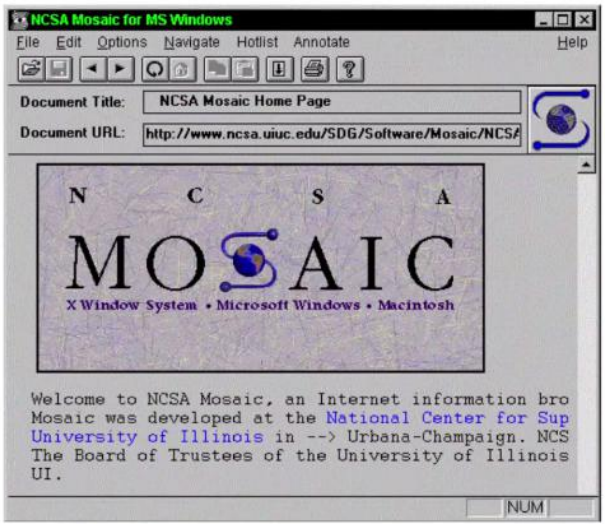
1、万维网概述

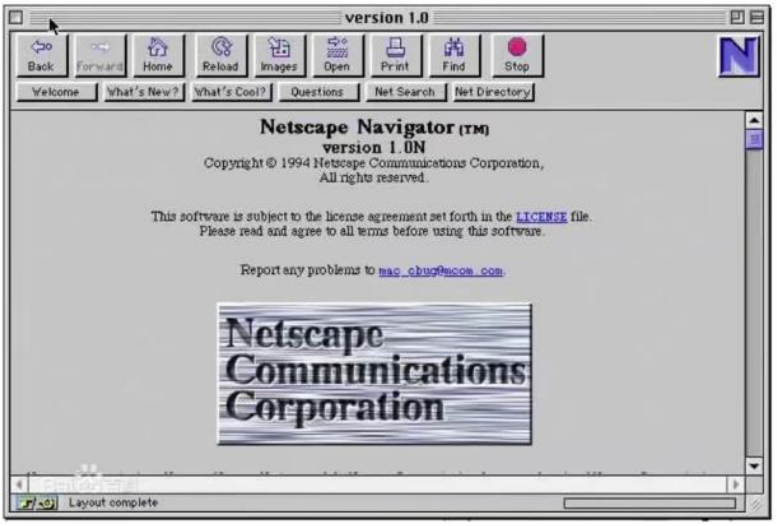
万维网WWW (World Wide Web）**并非某种特殊的计算机网络。**它是一个大规模的、联机式的信息储藏所，是**运行在因特网上的一个分布式应用。**

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。

万维网是欧洲粒子物理实验室的Tim Berners-Lee（蒂姆伯纳斯李）最初于1989年3月提出的。

1993年2月，第一个图形界面的浏览器Mosaic诞生，1995年著名的Netscape Navigator（网景）浏览器上市





目前流行的浏览器有以下几种：

2、统一资源定位符URL

当我们访问某个网页时，在浏览器的地址栏会出现一长串的字符，这一长串的字符我们**称之为统一资源定位符URL。**

```
https://news.sina.com.cn/c/xl/2021-12-14/doc-ikyakumx4140998.shtml
```

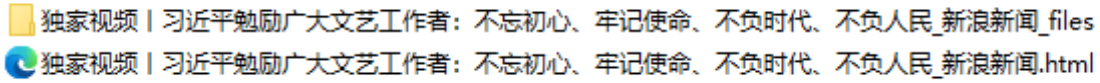
我们所访问到的网页上的内容其实是服务器上的某个资源，每个资源在服务器上都对应着一个位置。

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。

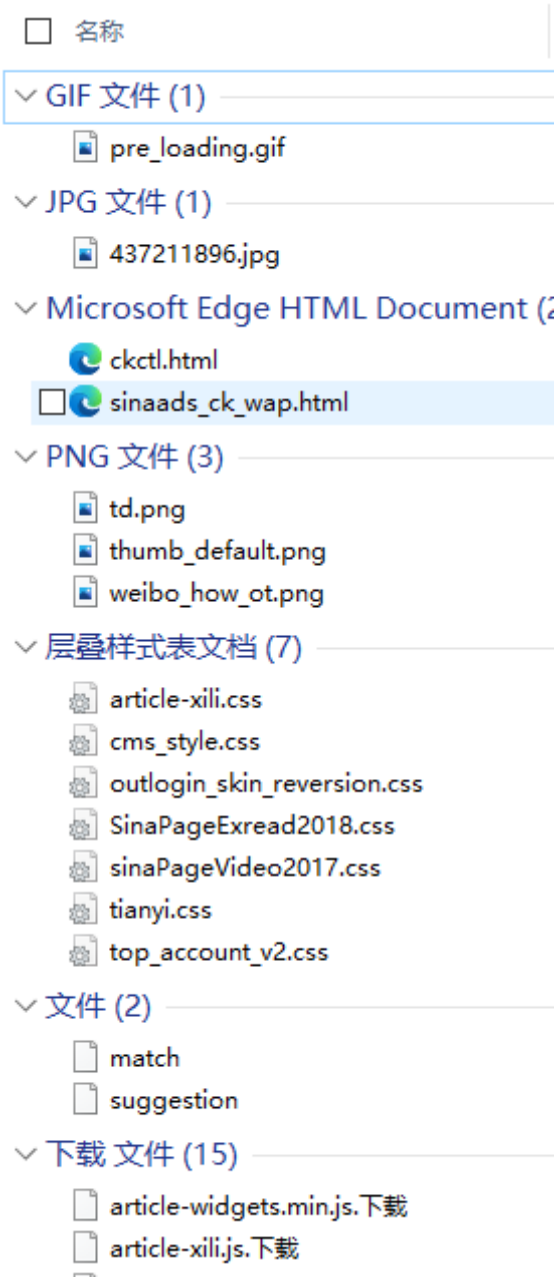
URL的一般形式由以下四个部分组成：

<协议>://<主机>:<端口>/<路径>

我们可以将某个网页另存为文件，存储后有一个html文件和一个文件夹。



文件夹中的内容如下：



万维网文档：由HTMLCSS、JavaSaript编写的方维网文档

超文本标记语言HTML(HyperText Markup Language)

使用多种“标签”来描述网页的结构和内容

层叠样式表CSS(Cascading Style Sheets)

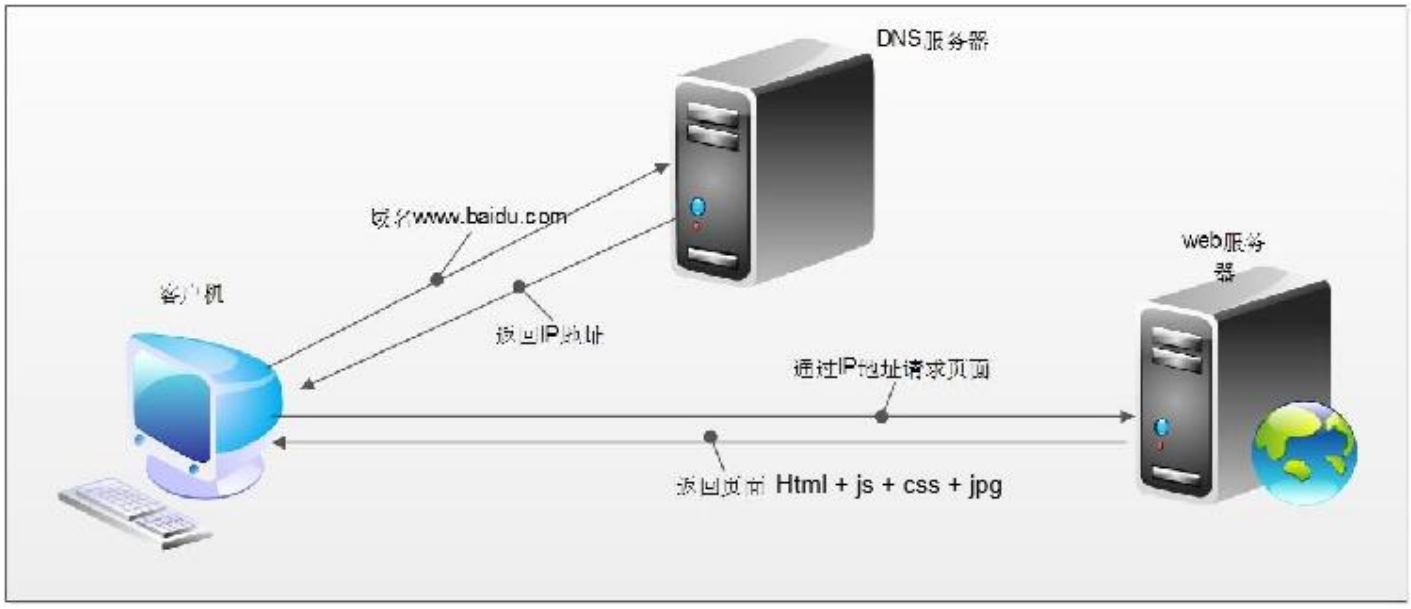
从审美的角度来描述网页的样式

一种脚本语言（和Java没有任何关系）

控制网页的行为

2.7.3 HTTP协议

1、浏览器发送HTTP请求的过程



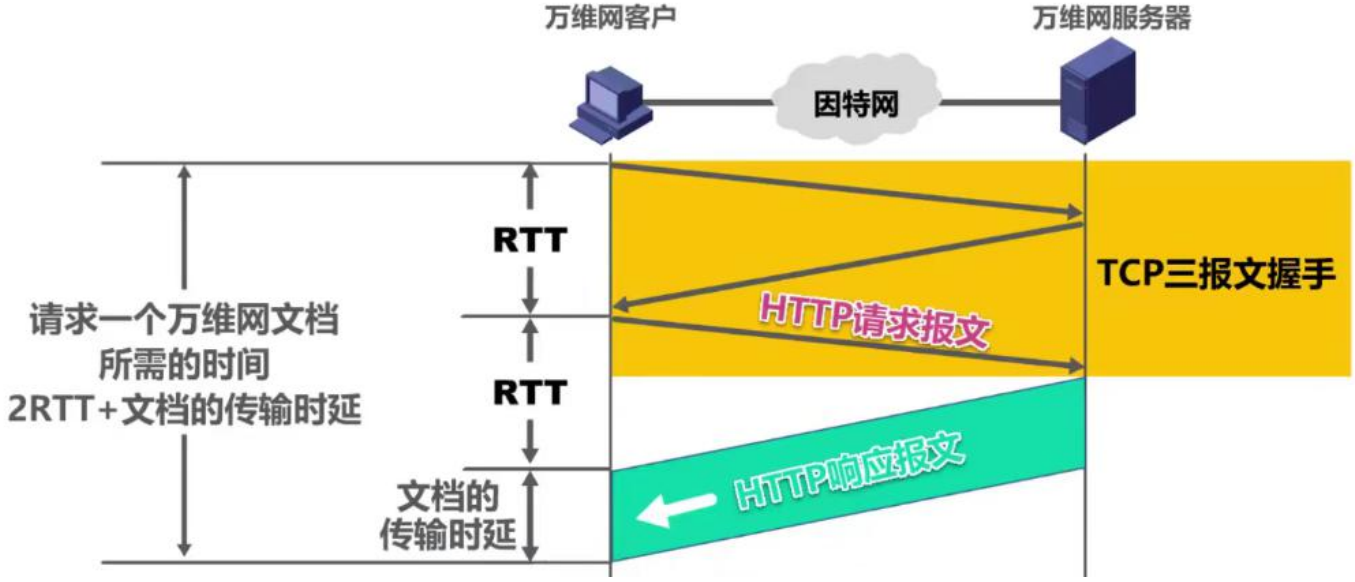
2、HTTP协议概述

超文本传输协议HTTP(HyperText Transfer Protocol)定义了浏览器(即万维网客户进程)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。是处理**客户端**和**服务端**之间的通信

3、HTTP/1.0

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，这会使其负担很重。



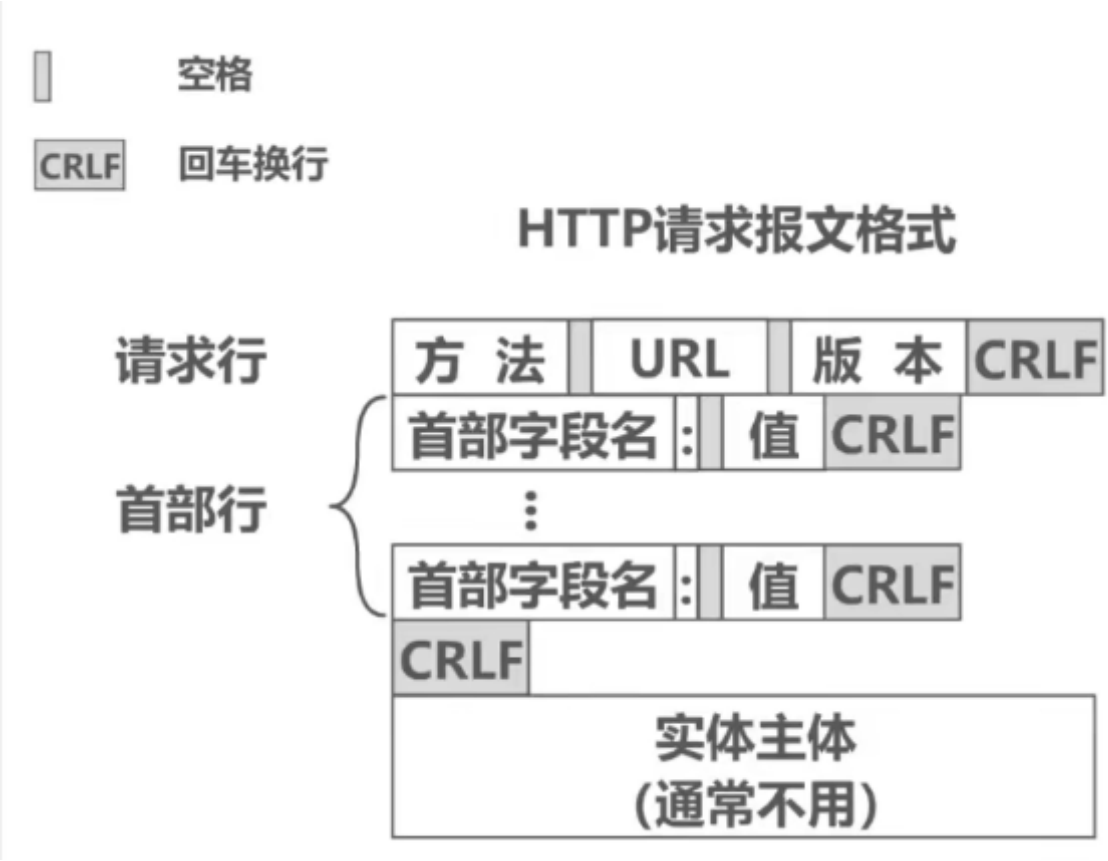
4、HTTP/1.1

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。

5、HTTP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。



我们可以利用浏览器的开发者工具查看某个HTTP请求的报文：

```
GET / HTTP/1.1
Host: www.baidu.com
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="95", "Chromium";v="95", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Cookie: BIDUPSID=5BDFB000AC259DC7C017DDA5A53536A8; PSTM=1629443792; __yjs_duid=1_4e745ab8ddb383a425ecf2e8df34bf411629444396777; BAIDUID=7F46D575E5E518
9951E24CA0643E5F45:FG=1; MCITY=-158%3A; BAIDUID_BFESS=7F46D575E5E5189951E24CA0643E5F45:FG=1; baikeVisitId=50212b31-81de-448a-a2bc-01335e928cfb; COOKIE
_SESSION=60_2_7_9_4_18_0_2_6_7_0_2_52_5_3_0_1638422268_1638422212_1638422265%7C9%238723808_50_1638422205%7C8
```

HTTP协议支持的请求方法：

方 法	描 述
GET	请求URL标志的文档
HEAD	请求URL标志的文档的首部
POST	向服务器发送数据
PUT	在指明的URL下存储一个文档
DELETE	删除URL标志的文档
CONNECT	用于代理服务器
OPTIONS	请求一些选项信息
TRACE	用来进行环回测试
PATCH	对PUT方法的补充，用来对已知资源进行局部更新

HTTP常见的请求头：

1. Host (主机和端口号)
2. Connection (链接类型)
3. Upgrade-Insecure-Requests (升级为HTTPS请求)
4. User-Agent (浏览器名称)
5. Accept (传输文件类型)
6. Referer (页面跳转处)
7. Accept-Encoding（文件编解码格式）
8. Cookie （Cookie）
9. x-requested-with :XMLHttpRequest （是Ajax 异步请求）

Connection：决定HTTP连接（不是TCP连接）是否在当前事务完成后关闭。

- Http1.0 默认是 close
- Http1.1 后默认是 keep-alive

> **Keep-Alive：多次请求复用一个TCP连接。**
>
> Keep-Alive: timeout=5, max=1000

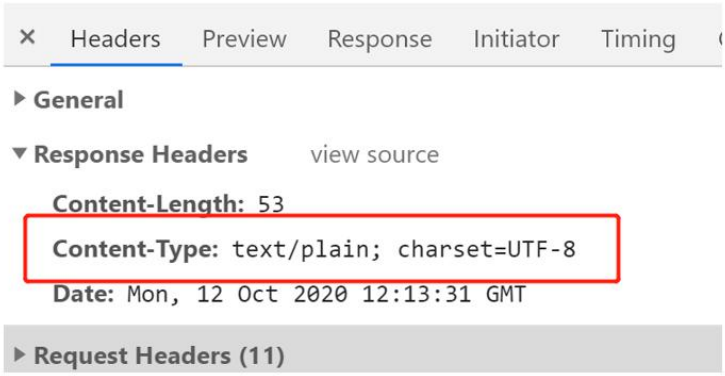
User-Agent：这个字段可以帮助统计客户端用了什么浏览器、操作系统等

通用标记符号　　操作系统　　引擎版本

```
Sec-Fetch-Site: same-origin
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.3
6 (KHTML, like Gecko) Chrome/84.0.4147.89 Safari/537.36
```

浏览器版本

Content-Type：请求的时候，告知服务端数据的媒体类（MediaType/MIME Type)。返回的时候告知客户端，数据的媒体类型。



- text/html：HTML格式
- text/css：css文本
- application/json： JSON数据格式
- image/jpeg：jpg图片格式
- text/plain：纯文本格式

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。**

> HTTP是无状态的：协议对于事务处理没有记忆能力，对同一个url请求没有上下文关系，每次的请求都是独立的，它的执行情况和结果与前面的请求和之后的请求是无直接关系的，它不会受前面的请求应答情况直接影响，也不会直接影响后面的请求应答情况

使用Cookie在服务器上记录用户信息：

状态码：web服务器告诉客户端请求的状态

常见的状态码为以下四类：

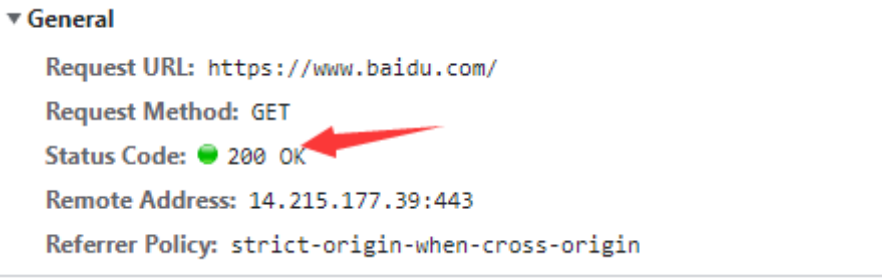
2xx：成功

3xx：重定向

4xx：客户端错误

5xx：服务端错误

- **200状态码**



- 3xx状态码

  300 – Multiple Choices 用户请求了多个选项的资源（返回选项列表）

  301 – Moved Permanently 永久转移

  302 – Found 资源被找到（以前是临时转移）

  303 – See Other 可以使用GET方法在另一个URL找到资源

  304 – Not Modified 没有修改（缓存部分特别说明）

  305 – Use Proxy 需要代理

  307 – Temporary Redirect 临时重定向

  308 – Permanent Redirect 永久重定向

- 4xx状态码

  400 – Bad Request 请求格式错误

  401 – Unauthorized 没有授权

  402 – Payment Required 请先付费

  403 – Forbidden 禁止访问

  404 – Not Found 没有找到

  405 – Method Not Allowed 方法不被允许

  406 – Not Acceptable 服务端可以提供的内容和客户端期待的不一样

- 5xx状态码

  500 – Internal Server Error(内部服务器错误)

  501 – Not Implemented（没有实现)

  502 – Bad Gateway(网关错误)

  503 – Service Unavailable(服务不可用)

  504 – Gateway Timeout(网关超时)

  505 – HTTP Version Not Supported（版本不支持)