TCP/IP overview

杨劲松 yjs@oldhand.org

2012.03.24

参考资料

- W.Richard Stevens 《UNIX网络编程》(第1卷)
- W.Richard Stevens《TCP/IP详解》(第1卷)
- W.Richard stevens《UNIX环境高级编程》
- Eric S.Raymond 《UNIX编程艺术》

学习方法(建议)

- 参考W. Richard Stevens《TCP/IP详解》(卷一)
- 阅读相关的RFC文档
- 动手实践
  - 使用tcpdump分析数据包
  - 使用wireshark分析数据包

1. 网络基础

- Internet的历史
- 中国互联网发展大事记
- TCP/IP起源
- 网络体系结构
- ISO/OSI参考模型
- TCP/IP协议
- 网络标准化
- 未来

1.1 Internet的历史

- Internet—“冷战”的产物
  - 1957年10月和11月，前苏联先后有两颗“Sputnik”卫星上天
  - 1958年美国总统艾森豪威尔向美国国会提出建立DARPA (Defense Advanced Research Project Agency)，即国防部高级研究计划署，简称ARPA
  - 1968年6月DARPA提出“资源共享计算机网络” (Resource Sharing Computer Networks)，目的在于让DARPA的所有电脑互连起来，这个网络就叫做ARPAnet，即“阿帕网”，是Internet的最早雏形
- 最初，ARPANet主要用于军事研究目的，它有五大特点
  - 支持资源共享
  - 用分布式控制技术
  - 用分组交换技术
  - 用通信控制处理机
  - 用分层的网络通信协议

1.1 Internet的历史-时间表

- 在1950年代，通信研究者认识到需要允许在不同计算机用户和通信网络之间进行常规的通信，这促使了分散网络、排队论和封包交换的研究
- 1960年美国国防部国防前沿研究项目署（ARPA）出于冷战考虑建立的ARPANet引发了技术进步并使其成为互联网发展的中心
- 1973年ARPANet扩展成互联网，第一批接入的有英国和挪威计算机
- 1974年ARPA的鲍勃·凯恩和斯坦福的温登·泽夫提出TCP/IP协议，定义了在电脑网络之间传送报文的方法
- 1983年1月1日，ARPA网将其网络内核协议由NCP改变为TCP/IP协议
- 1983年保罗·莫卡派乔斯（Paul Mockapetris）发明DNS
  - 原始的技术规范在882号因特网标准草案（RFC 882）中发布。
  - 1987年发布的第1034和1035号草案修正了DNS技术规范，并废除了之前的第882和883号草案。在此之后对因特网标准草案的修改基本上没有涉及到DNS技术规范部分的改动。
- 1986年，美国国家科学基金会（National Science Foundation，NSF）建立了大学之间互联的骨干网络NSFNet，这是互联网历史上重要的一步。
  - 1994年，NSFNet转为商业运营。
  - 1995年随着网络开放予商业. 互联网中成功接入的比较重要的其他网络包括Usenet、Bitnet和多种商用X.25网络。
- 1990年代，整个网络向公众开放。
  - 在1991年8月，在蒂姆·伯纳斯-李（Tim Berners-Lee）在瑞士创立HTML、HTTP和欧洲粒子物理研究所（CERN）的最初几个网页之后两年，他开始宣扬其万维网（World Wide Web）项目。
  - 在1993年，Mosaic网页浏览器版本1.0被放出了，在1994年晚期，公共利益在前学术和技术的互联网上稳步增长。
  - 1996年，“Internet”（互联网）一词被广泛的流传，不过是指几乎整个的万维网。

1.2 中国互联网发展大事记

- 1987年9月，在德国卡尔斯鲁厄大学（Karlsruhe University）维纳<措恩（Werner Zorn）教授带领的科研小组的帮助下，王运丰教授和李澄炯博士等在北京计算机应用技术研究所（ICA）建成一个电子邮件节点，并于9月20日向德国成功发出了一封电子邮件，邮件内容为"Across the Great Wall we can reach every corner in the world.（越过长城，走向世界）"。
- 1988年初，中国第一个X.25分组交换网CNPAC建成，当时覆盖北京、上海、广州、沈阳、西安、武汉、成都、南京、深圳等城市。
- 1990年11月28日，在王运丰教授和维纳<措恩（Werner Zorn）教授的努力下，中国的顶级域名.CN完成注册，钱天白任行政联络员。从此在国际互联网上中国有了自己的身份标识。由于当时中国尚未实现与国际互联网的全功能联接，中国CN顶级域名服务器暂时设在德国卡尔斯鲁厄大学。
- 1992年12月底，清华大学校园网(TUNET)建成并投入使用，是中国第一个采用TCP/IP体系结构的校园网，主干网首次成功采用FDDI技术，在网络规模、技术水平以及网络应用等方面处于国内领先水平。
- 1994年5月15日，中国科学院高能物理研究所设立了国内第一个WEB服务器，推出中国第一套网页，内容除介绍中国高科技发展外，还有一个栏目叫“Tour in China”。此后，该栏目开始提供包括新闻、经济、文化、商贸等更为广泛的图文并茂的信息，并改名为《中国之》。
- 1994年5月21日，在钱天白教授和德国卡尔斯鲁厄大学的协助下，中国科学院计算机网络信息中心完成了中国国家顶级域名(CN)服务器的设置，改变了中国的CN顶级域名服务器一直放在国外的历史。由钱天白、钱华林分别担任中国CN域名的行政联络员和技术联络员。
- 1994年5月，国家智能计算机研究开发中心开通曙光BBS站，这是中国大陆的第一个BBS站。
- 1996年3月，清华大学提交的适应不同国家和地区中文编码的汉字统一传输标准被IETF通过为RFC1922，成为中国国内第一个被认可为RFC文件的提交协议。
- 1996年11月15日，实华开公司在北京首都体育馆旁边开设了实华开网络咖啡屋，这是中国第一家网络咖啡屋。
- 1997年2月，瀛海威全国大网开通，3个月内在北京、上海、广州、福州、深圳、西安、沈阳、哈尔滨8个城市开通，成为中国最早、也是最大的民营ISP、ICP。
- 1997年5月30日，国务院信息化工作领导小组办公室发布《中国互联网络域名注册暂行管理办法》，授权中国科学院组建和管理中国互联网络信息中心(CNNIC)，授权中国教育和科研计算机网网络中心与CNNIC签约并管理二级域名.edu.cn。
- 1998年6月，CERNET正式参加下一代IP协议(IPv6)试验网6BONE。
- 1999年5月，在清华大学网络工程研究中心成立了中国第一个安全事件应急响应组织CCERT(CERNET Computer Emergency Response Team)。
- 1999年7月12日，中华网在纳斯达克首发上市，这是在美国纳斯达克第一个上市的中国概念网络公司股。
- 2000年以后：http://www.cnnic.net.cn/html/Dir/2003/10/22/1001.htm

1.3.1 TCP/IP起源

- TCP/IP起源于1960年代末美国政府资助的一个分组交换网络研究项目，到1990年代已发展成为计算机之间最常应用的组网形式
- 它是一个真正的开放系统，因为协议族的定义及其多种实现可以不用花钱或花很少的钱就可以公开地得到
- 它成为被称作“全球互联网”或“因特网(Internet)”的基础，该广域网（WAN）已包含超过100万台遍布世界各地的计算机

1.3.2 TCP/IP演进过程

- 早期的ARPANet使用网络控制协议(Network Control Protocol，NCP)，不能互联不同类型的计算机和不同类型的操作系统，没有纠错功能
- 1973年由Kahn和Vinton Cerf两人合作为ARPAnet开发了新的互联协议
- 1974年12月两人正式发表第一份TCP协议详细说明，但此协议有信包丢失时不能得到有效的纠正
- TCP协议分成了两个不同的协议：
  - 用来检测网络传输中差错的传输控制协议TCP
  - 专门负责对不同网络进行互联的互联网协议IP
- 1983年ARPAnet上停止使用NCP，互联网上的主机全部使用TCP/IP协议，TCP/IP协议成为Internet中的“世界语”

1.4 网络的体系结构

- 网络采用分而治之的方法设计，将网络的功能划分为不同的模块，以分层的形式有机组合在一起。
- 每层实现不同的功能，其内部实现方法对外部其他层次来说透明，每层向上层提供服务，也可以使用下层提供的服务
- 网络体系结构即指网络的层次结构和每层所使用协议的集合
- 两类非常重要的体系结构：OSI RM与TCP/IP

1.4.1 网络体系结构

- 网络体系结构定义
  - 是指通信系统的整体设计，它为网络硬件、软件、协议、存取控制和拓扑提供标准。它广泛采用的是国际标准化组织（ISO）在1979年提出的开放系统互连（OSI-Open System Interconnection)的参考模型。
- 计算机网络的体系结构的形成
  - 计算机网络是一个非常复杂的系统，需要解决的问题很多并且性质各不相同。所以，在ARPANET设计时，就提出了“分层”的思想，即将庞大而复杂的问题分为若干较小的易于处理的局部问题。
  - 1974年美国IBM公司按照分层的方法制定了系统网络体系结构SNA（System Network Architecture）。现在SNA已成为世界上较广泛使用的一种网络体系结构。
    - 常见的计算机网络体系结构有DEC公司的DNA（数字网络体系结构）、IBM公司的SNA（系统网络体系结构）等
  - 为解决异种计算机系统、异种操作系统、异种网络之间的通信，国际标准化组织（ISO）以国际上其他的一些标准化团体，在各厂家提出的计算机网络体系结构的基础上，提出了开放系统互联参考模型（OSI/RM）。
    - 一开始，各个公司都有自己的网络体系结构，就使得各公司自己生产的各种设备容易互联成网，有助于该公司垄断自己的产品。但是，随着社会的发展，不同网络体系结构的用户迫切要求能互相交换信息。为了使不同体系结构的计算机网络都能互联，国际标准化组织ISO于1997年成立专门机构研究这个问题。1978年ISO提出了“异种机连网标准”的框架结构，这就是著名的开放系统互联参考模型OSI。
    - OSI得到了国际上的承认，成为其他各种计算机网络体系结构依照的标准，大大地推动了计算机网络的发展。20世纪70年代末到80年代初，出现了利用人造通信卫星进行中继的国际通信网络。网络互联技术不断成熟和完善，局域网和网络互联开始商品化。
    - OSI参考模型用物理层、数据链路层、网络层、传送层、对话层、表示层和应用层七个层次描述网络的结构，它的规范对所有的厂商是开放的，具有知道国际网络结构和开放系统走向的作用。它直接影响总线、接口和网络的性能。目前常见的网络体系结构有FDDI、以太网、令牌环网和快速以太网等。从网络互连的角度看，网络体系结构的关键要素是协议和拓扑。

1.4.2 网络协议

- 网络中的计算机与计算机之间要想正确的传送信息和数据，必须在数据传输的顺序、数据的格式及内容等方面有一个约定或规则，这种约定或规则称为协议
  - 所谓计算机网络协议，就是通信双方事先约定的通信规则的集合
- 一个网络协议主要包含以下三个要素：
  - 语法(Syntax)：即数据与控制信息的结构和格式，包括数据格式、编码及信号电平等。
  - 语义(Semantics)：是用于协调和差错处理的控制信息。如需要发出何种控制信息完成何种动作以及做出何种应答等。
  - 时序(Timing)：即对有关事件实现顺序的详细说明，如速度匹配、排序等
- 网络协议大多妥协而产生

1.5 ISO/OSI参考模型

- OSI模型，即开放式通信系统互联参考模型(Open System Interconnection Reference Model)，是国际标准化组织（ISO）提出的一个试图使各种计算机在世界范围内互连为网络的标准框架，简称OSI。
- OSI将计算机网络体系结构（architecture）划分为以下七层
  - 物理层(Physical Layer)是参考模型的最低层。该层是网络通信的数据传输介质，由连接不同结点的电缆与设备共同构成。主要功能是：利用传输介质为数据链路层提供物理连接，负责处理数据传输并监控数据出错率，以便数据流的透明传输。
  - 数据链路层(Data Link Layer)是参考模型的第2层。主要功能是：在物理层提供的服务基础上，在通信的实体间建立数据链路连接，传输以帧为单位的数据包，并采用差错控制与流量控制方法，使有差错的物理线路变成无差错的数据链路。
  - 网络层(Network Layer)是参考模型的第3层。主要功能是：为数据在结点之间传输创建逻辑链路，通过路由选择算法为分组通过通信子网选择最适当的路径，以及实现拥塞控制、网络互联等功能。
  - 传输层(Transport Layer)是参考模型的第4层。主要功能是向用户提供可靠的端到端(End-to-End)服务，处理数据包错误、数据包次序，以及其他一些关键传输问题。传输层向高层屏蔽了下层数据通信的细节，因此，它是计算机通信体系结构中关键的一层。
  - 会话层(Session Layer)是参考模型的第5层。主要功能是：负责维护两个结点之间的传输链接，以便确保点到点传输不中断，以及管理数据交换等功能。
  - 表示层(Presentation Layer)是参考模型的第6层。主要功能是：用于处理在两个通信系统中交换信息的表示方式，主要包括数据格式变换、数据加密与解密、数据压缩与恢复等功能。
  - 应用层(Application Layer)是参考模型的最高层。主要功能是：面向用户为应用软件提供了很多服务，例如文件服务器、数据库服务、电子邮件与其他网络软件服务。

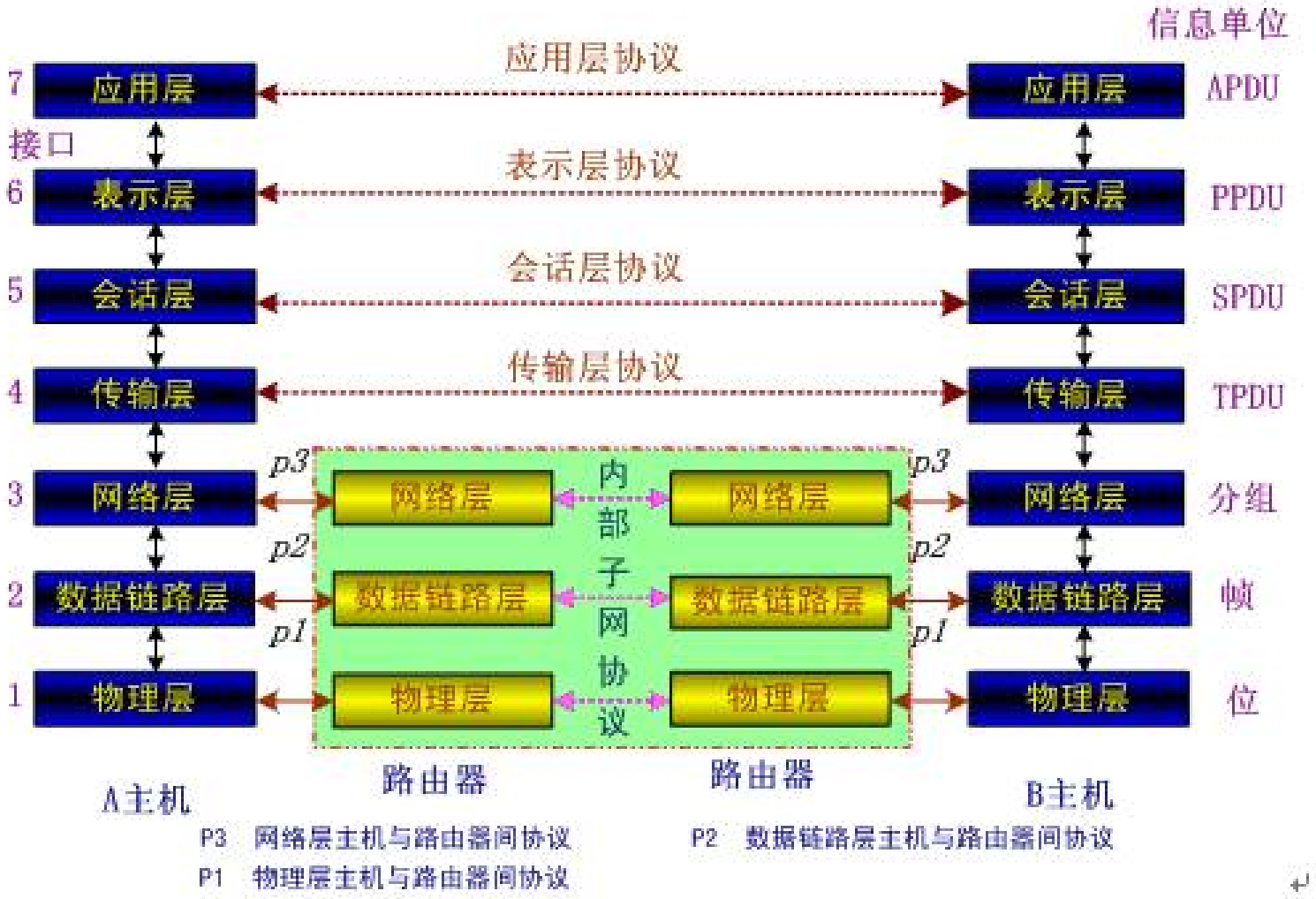
物理层、数据链路层、IP层关系

- 物理层-相当于外部总线
  - 地址：无，通过介质直接相连(无线通过相同信道)
  - 传输单元：bit
  - 提供通道：物理线路
- 数据链路层-对外部总线如何使用进行控制
  - 地址：MAC地址，物理地址，硬编码
  - 传输单元：frame
  - 提供通道：数据链路(由物理线路增加访问控制机制实现)
- IP层-构建逻辑通道
  - 地址：IP地址，逻辑地址，软编码
  - 传输单元：packet
  - 提供通道：逻辑链路(由多个数据链路拼接而成)

常见术语

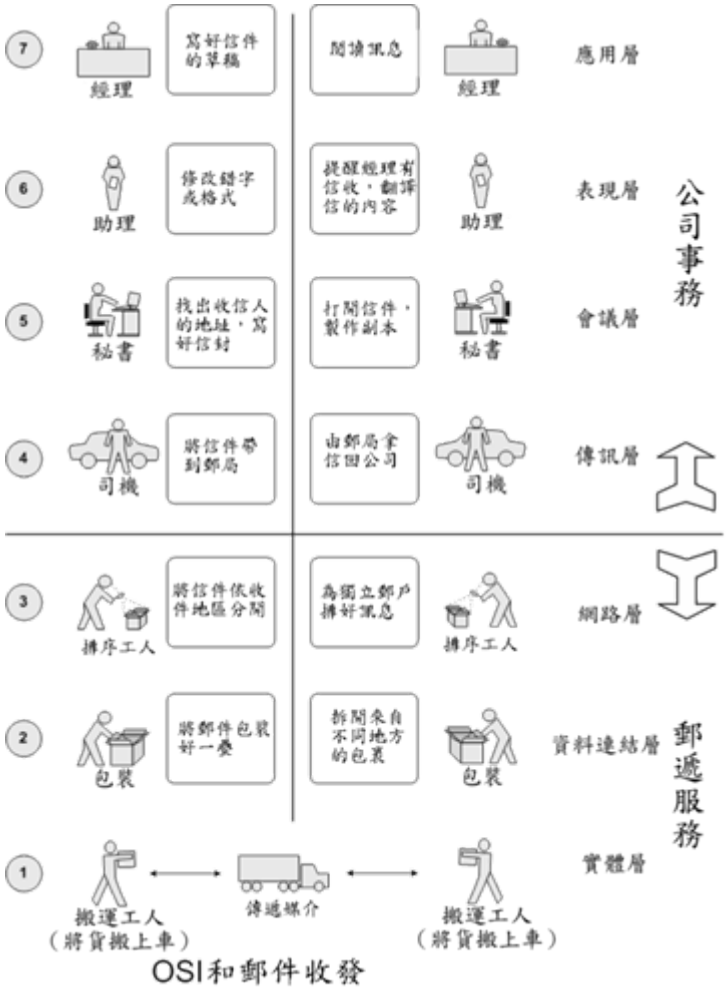
- Host - Host
  - 数据包从源主机到目的主机
  - 由IP层实现
- End - End
  - 数据包从发送进程到目的进程
  - 由传输层实现
  - Endpoint - Endpoint
  - Socket - Socket
  - Process - Process，理解bind的必要性
- Peer - Peer
  - 应用层面的概念，是一种网络服务方式，如BT、eMule等
- Point to point
  - 通讯层面的概念，如PPP协议
  - 有时也将host-host当做point-point

1.5.1 ISO/OSI参考模型-分层



1.5.2 ISO/OSI参考模型 – 比喻

- 7 应用层：老板
- 6 表示层：相当于公司中简报老板、替老板写信的助理
- 5 会话层：相当于公司中收寄信、写信封与拆信封的秘书
- 4 传输层：相当于公司中跑邮局的送信职员
- 3 网络层：相当于邮局中的排序工人
- 2 数据链路层：相当于邮局中的装拆箱工人
- 1 物理层：相当于邮局中的搬运工人



1.5.3 ISO/OSI参考模型 – 地位

- OSI是一个定义良好的协议规范集，并有许多可选部分完成类似的任务。
- OSI定义了开放系统的层次结构、层次之间的相互关系以及各层所包括的可能的任务。
- OSI是作为一个框架来协调和组织各层所提供的服务。
- OSI参考模型并没有提供一个可以实现的方法，而是描述了一些概念，用来协调进程间通信标准的制定。即OSI参考模型并不是一个标准，而是一个在制定标准时所使用的概念性框架。

1.6 TCP/IP协议

- TCP/IP协议
  - 传输控制/网际协议(Transfer Control Protocol/Internet Protocol) 又称作网络通讯协议
  - Internet国际互联网络的基础
  - RFC791定义IP
  - RFC792定义ICMP
  - RFC793定义TCP
  - 一组协议，通常称它为TCP/IP协议族
  - 四个层次：网络接口层、网际层、传输层、应用层

1.6.1 TCP/IP协议族

- 常用协议
  - IP(Internetworking Protocol)网间网协议
  - TCP(Transport Control Protocol)传输控制协议
  - UDP(User Datagram Protocol)用户数据报协议
  - ICMP(Internet Control Message Protocol)互联网控制信息协议
  - SMTP(Simple Mail Transfer Protocol)简单邮件传输协议
  - SNMP(Simple Network manage Protocol)简单网络管理协议
  - HTTP(Hypertext Transfer Protocol) 超文本传输协议
  - FTP(File Transfer Protocol)文件传输协议
  - ARP(Address Resolation Protocol)地址解析协议

1.6.2 TCP/IP协议与OSI参考模型

OSI中的层	功能	TCP/IP协议族
应用层	文件传输，电子邮件，文件服务，虚拟终端	TFTP，HTTP，SNMP FTP，SMTP，DNS，Telnet
表示层	数据格式化，代码转换，数据加密	没有协议
会话层	解除或建立与别的接点的联系	没有协议
传输层	提供端对端的接口	TCP，UDP
网络层	为数据包选择路由	IP，ICMP，RIP，OSPF，BGP，IGMP
数据链路层	传输有地址的帧以及错误检测功能	SLIP，CSLIP，PPP，ARP，RARP
物理层	以二进制数据形式在物理媒体上传输数据	ISO2110，IEEE802.1,IEEE802.2

1.7 网络标准化

- 究竟是谁控制着TCP/IP协议族，又是谁在定义新的标准以及其他类似的事情？事实上，有四个小组在负责Internet技术。
  - Internet协会（ISOC，Internet Society）是一个推动、支持和促进Internet不断增长和发展的专业组织，它把Internet作为全球研究通信的基础设施。
  - Internet体系结构委员会（IAB，Internet Architecture Board）是一个技术监督和协调的机构。它由国际上来自不同专业的15个志愿者组成，其职能是负责Internet标准的最后编辑和技术审核。IAB隶属于ISOC。
  - Internet工程专门小组（IETF，Internet Engineering Task Force）是一个面向近期标准的组织，它分为9个领域（应用、寻径和寻址、安全等等）。IETF开发成为Internet标准的规范。为帮助IETF主席，又成立了Internet工程指导小组（IESG,Internet Engineering Steering Group）。
  - Internet研究专门小组（IRTF，Internet Research Task Force）主要对长远的项目进行研究。IRTF和IETF都隶属于IAB。文献[Crocker1993]提供了关于Internet内部标准化进程更为详细的信息，同时还介绍了它的早期历史。

1.7 网络相关的标准

- RFC
  - 网络通讯相关的协议，如IP/TCP/UDP…
  - 网络服务/应用相关的协议，如HTTP/FTP/SMTP…
  - 辅助协议，如ASN…
- IEEE
  - 链路层以下，802.11…
- ITU
  - 电信业务相关，H.323/H.264/V.92…
- ISO
  - 标准的超集

1.7.1 RFC

- Request For Comments（RFC），是一系列以编号排定的文件。文件收集了有关互联网相关信息，以及UNIX和互联网社区的软件文件。目前RFC文件是由Internet Society（ISOC）赞助发行。
- 基本的互联网通信协议都有在RFC文件内详细说明。RFC文件还额外加入许多的论题在标准内，例如对于互联网新开发的协议及发展中所有的记录。因此几乎所有的互联网标准都有收录在RFC文件之中。
- RFC的历史
  - RFC文件格式最初作为ARPA网计划的基础起源于1969年。如今，它已经成为IETF、Internet Architecture Board (IAB)还有其他一些主要的公共网络研究社区的正式出版物发布途径。
  - 在RFC诞生之时，互联网还不存在，只有4大研究中心的4台计算机连接成的原始网络：UCLA，斯坦福研究所，加州圣塔芭芭拉分校，和盐湖城的犹他大学。[1]最初的RFC作者使用打字机撰写文档，并在美国国防部国防前沿研究项目署（ARPA）研究成员之间传阅。1969年12月，他们开始通过ARPANET途径来发布新的RFC文档。第一份RFC文档由洛杉矶加利福尼亚大学（UCLA）的Steve Crocker撰写，在1969年4月7日公开发表的RFC 1。当初Crocker为了避免打扰他的室友，是在浴室里完成这篇文档的。
  - 在1970年代，很多后来的RFC文档同样来自UCLA，这不仅得益于UCLA的学术质量，同时也因为UCLA是ARPANET第一批Interface Message Processors (IMPs)成员之一。
  - 由Douglas Engelbart领导的，位于Stanford Research Institute的Augmentation Research Center (ARC)是四个最初的ARPANET结点之一，也是最初的Network Information Centre，同时被社会学家Thierry Bardini记录为早期大量RFC文档的发源地。
  - 从1969年到1998年，Jon Postel一直担任RFC文档的编辑职务。随着美国政府赞助合同的到期，Internet Society（代表IETF），和南加州大学（USC）Information Sciences Institute的网络部门合作，（在IAB领导下）负责RFT文档的起草和发布工作。Jon Postel继续担任RFC编辑直到去世。随后，由Bob Braden接任整个项目的领导职务，同时Joyce Reynolds继续在团队中的担任职务。
  - 庆祝RFC的30周年的RFC文件是RFC 2555。
- RFC文件的架构
  - RFC文件只有新增，不会有取消或中途停止发行的情形。但是对于同一主题而言，新的RFC文件可以声明取代旧的RFC文件。RFC文件是纯ASCII文字档格式，可由计算机程序自动转档成其他文件格式。RFC文件有封面、目录及页眉页脚和页码。RFC的章节是数字标示，但数字的小数点后不补零，例如4.9的顺序就在4.10前面，但9的前面并不补零。RFC1000这份文件就是RFC的指南。
- RFC文件的产生
  - RFC文件是由Internet Society审核后给定编号并发行。虽然经过审核，但RFC也并非全部严肃而生硬的技术文件，偶有恶搞之作出现，尤其是4月1日愚人节所发行的，例如RFC 1606: A Historical Perspective On The Usage Of IP Version 9（参见IPv9）、RFC 2324：“超文字咖啡壶控制协议”（Hyper Text Coffee Pot Control Protocol，乍有其事的写了HTCPCP这样看起来很专业的术语缩写字）。以及如前面所提到纪念RFC的30周年庆的RFC文件。

1.7.1 RFC

- 所有关于Internet的正式标准都以RFC文档出版。另外，大量的RFC并不是正式的标准，出版的目的只是为了提供信息。RFC的篇幅从1页到200页不等。每一项都用一个数字来标识，如RFC1122，数字越大说明RFC的内容越新。
- RFC获取方式
  - 通过WEB获取RFC:
    - http://www.ietf.org/rfc.html
    - http://www.rfc-editor.org/
    - http://www.faqs.org/rfcs/
  - FTP
    - ftp://ftp.funet.fi/pub/doc/rfc/rfc<number>.txt
  - E-mail

1.7.1 一些重要的RFC文档

- 赋值RFC（Assigned Numbers RFC）
  - 列出了所有Internet协议中使用的数字和常数
  - RFC1700是最后一个Assigned numbers RFC
  - RFC3232 - Assigned Numbers: RFC 1700 is Replaced by an On-line Database
- Internet正式协议标准(Internet Official Protocol Standards)
  - 目前是RFC5000
  - 描述了各种Internet协议的标准化现状
    - 每种协议都处于下面几种标准化状态之一：标准、草案标准、提议标准、实验标准、信息标准和历史标准
    - 对每种协议都有一个要求的层次、必需的、建议的、可选择的、限制使用的或者不推荐的
  - 定期更新：http://www.rfc-editor.org/rfcxx00.html
- 主机需求RFC(Requirements for Internet Hosts)，RFC1122和RFC1123
  - RFC1122针对链路层、网络层和运输层
  - RFC1123针对应用层
  - 这两个RFC对早期重要的RFC文档作了大量的纠正和解释。如果要查看有关协议更详细的细节内容，它们通常是一个入口点。它们列出了协议中关于“必须”、“应该”、“可以”、“不应该”或者“不能”等特性及其实现细节。
  - 文献[Borman1993b]提供了有关这两个RFC的实用内容。RFC1127[Braden1989c]对工作小组开发主机需求RFC过程中的讨论内容和结论进行了非正式的总结。
- 路由器需求RFC(Requirements for Internet gateways)
  - 目前正式版是RFC1009
  - 与主机需求RFC类似，但是只单独描述了路由器的需求

1.8 未来

- Globalism（全球化）
  - The future of the Internet global distribution of information and knowledge at lower and lower cost will continue to lift the world community for generations to come. People will have access to any information they wish, get smarter sooner, and be more aware of the world outside their local environment. A better informed humanity will make better macro-level decisions, and an increasingly integrated world will drive international relations towards a global focus. Attachments to countries will marginally decrease, and attachments to the Earth as a shared resource will significantly increase.
- Communities（社区化）
  - The future of the Internet communications revolution is ongoing, now uniting communities as it recently united networks. Not everything about the Internet is global; an interconnected world is also locally interconnected. The Internet will increasingly be used for communications within communities as much as across countries. Local communities will organize in virtual space and take increasing advantage of group communication tools such as mailing lists, newsgroups, and websites, and towns and cities will become more organized and empowered at the neighborhood level.
  - At the same time, communities will be as profoundly affected by the capabilities the Internet is bringing to individual communications, providing individuals in the once isolating city the ability to easily establish relationships with others in their local area by first meeting in cyberspace. From hobby clubs to political organizations to social networking, Internet applications will change expectations of geographically oriented community organizations, and provide increasingly wide choices to individuals who wish to participate in local communities that share their interests.

1.8 未来(cont.)

- Virtual reality（虚拟现实）
  - The future of the Internet technological revolution will continue to be made in man's image. Experiments with wide area voice and video communications on the Internet began to be held in the early 1990's. Voice over IP (VOIP) began to be used regularly for long distance voice communications in 2002. Internet video phones won't be far behind. With the continued doubling of computer capability every couple of years, the ability of technology to process the complex analog environment that humans live in -- "reality" -- will continue to increase, and will be increasingly integrated with the Internet.
  - Three dimensional graphics will become more sophisticated, and virtual reality interfaces such as viewers and tactile feedback systems will become more realistic. The technology will be applied to innovative ways to navigate the Internet's information universe, for hyper-realistic gaming, and for group communications. There will come a day when you will be able to have dinner with a group of friends each in a different city, almost as though you were in the same room, although you will all have to bring your own food.
  - Virtual reality applications will not only better and better reflect the natural world, they will also have the fluidity, flexibility, and speed of the digital world, layered on the Internet, and so will be used to create apparently magical environments of types we can only now begin to imagine. These increasingly sophisticated virtual experiences will continue to change how we understand the nature of reality, experience, art, and human relations.
- Bandwidth（带宽）
  - The future of the Internet growth in bandwidth availability shows little sign of flattening. Large increases of bandwidth in the 10 Mbps range and up will continue to be deployed to home users through cable, phone, and wireless networks. Cable modems and telephone-based DSL modems will continue to spread high speed Internet throughout populated areas. High resolution audio, video, and virtual reality will be increasingly available online and on demand, and the cost of all kinds of Internet connections will continue to drop.

1.8 未来(cont.)

- Wireless（无线）
  - The future of Internet wireless communications is the end-game. Wireless frequencies has two great advantages: (a) there are no infrastructure start-up or maintenance costs other than the base stations, and (b) it frees users to become mobile, taking Internet use from one dimension to three. Wireless Internet networks will offer increasingly faster services at vastly lower costs over wider distances, eventually pushing out physical transmission systems.
  - The Internet's open TCP/IP design was originally inspired by use of radio communications networks in the 1970's. The wireless technologies experimented with in the 1990's were continually improved. By the early 2000's, several technologies provided reliable, secure, high bandwidth networking that worked in crowded city centers and on the move, providing nearly the same mobility for Internet communications as for the cellular phone.
- Grids（网格化）
  - The future of the Internet grid movement is as inevitable as the spread of the Internet seems now. The connection of thousands of computers on the Internet together to solve problems, often called grid computing, will continue to evolve and change many areas of human endevour. In a large scale example of the connected Internet fostering technological cooperation, un-used computer cycles from home users across the world will be harnessed together to provide enormous reservoirs of computer power for all sorts of purposes. Increasingly used for scientific and engineering research, grids can create processing powerhouses far larger than any one organization by itself.
- Integration（融合）
  - The future of the Internet integration with an increasing number of other technologies is as natural as a musician's experimentation with notes. The Internet will become increasingly integrated with phones, televisions, home appliances, portable digital assistants, and a range of other small hardware devices, providing an unprecedented, nearly uniform level of integrated data communications. Users will be able to access, status, and control this connected infrastructure from anywhere on the Internet.

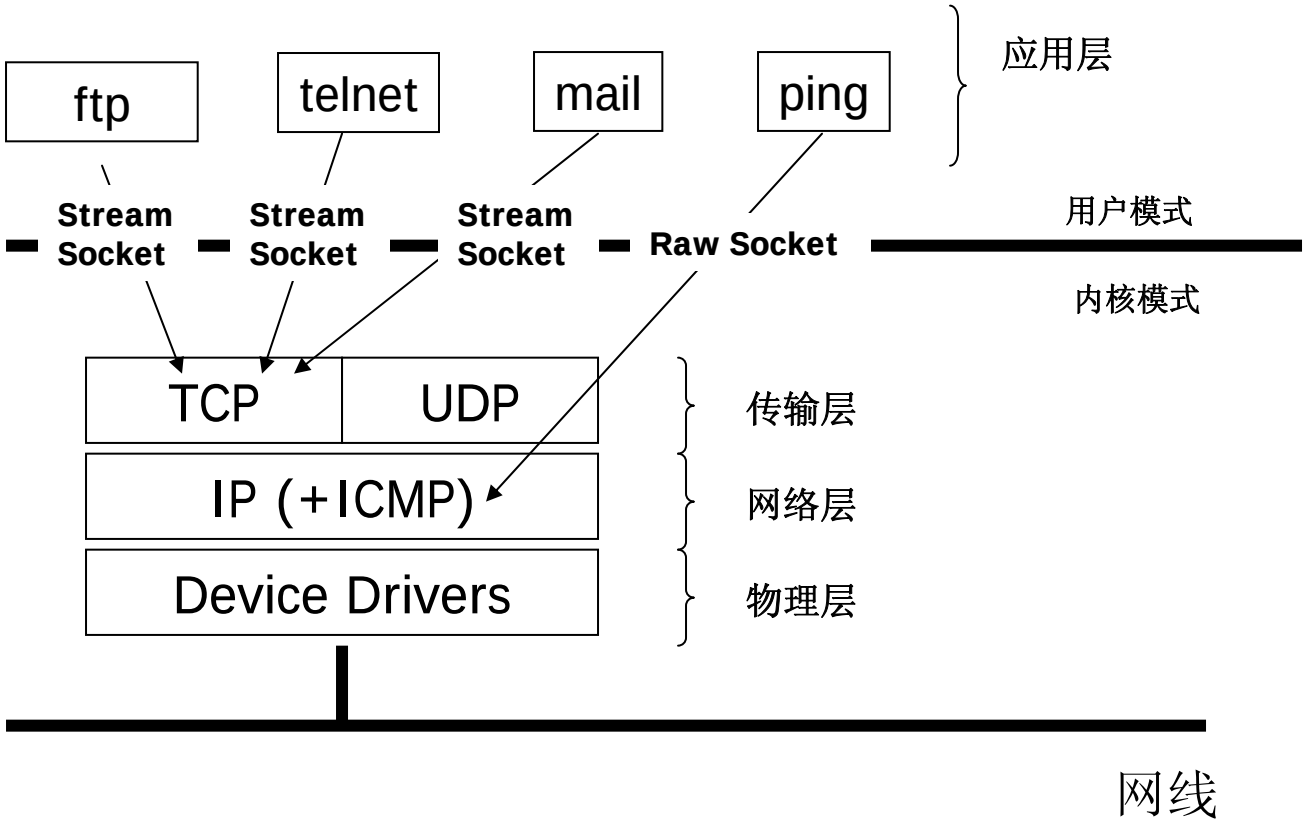
2. TCP/IP协议

- TCP/IP协议
- TCP/IP协议族
- 分层
- 封装
- 分用
- 链路层
- IP协议
- ARP协议
- ICMP协议
- UDP协议
- TCP协议
- TCP v.s. UDP

2.1 TCP/IP协议

- TCP/IP协议
  - 传输控制/网际协议(Transfer Control Protocol/Internet Protocol) 又称作网络通讯协议
  - Internet国际互联网络的基础
  - RFC791定义IP
  - RFC792定义ICMP
  - RFC793定义TCP
  - 一组协议，通常称它为TCP/IP协议族
  - 四个层次：网络接口层、网际层、传输层、应用层
- TCP的语境问题
  - 当和IP一起提时，此时TCP是广义的概念，代表传输层协议，即狭义的TCP和UDP
  - 当和UDP一起提时，此时TCP是狭义的传输控制协议的概念

TCP/IP结构

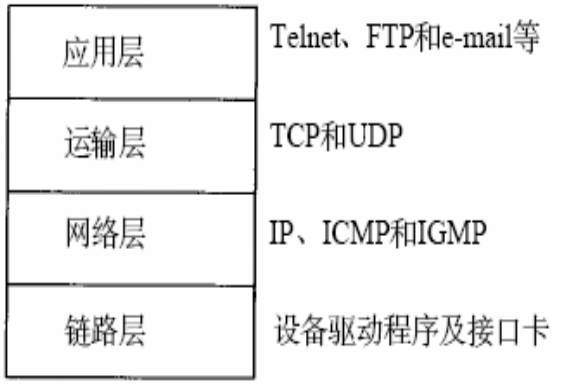

ftp
telnet
mail
ping
应用层
Stream Socket
Stream Socket
Stream Socket
Raw Socket
用户模式
内核模式
TCP
UDP
传输层
IP (+ICMP)
网络层
Device Drivers
物理层
网线

2.2 TCP/IP协议族

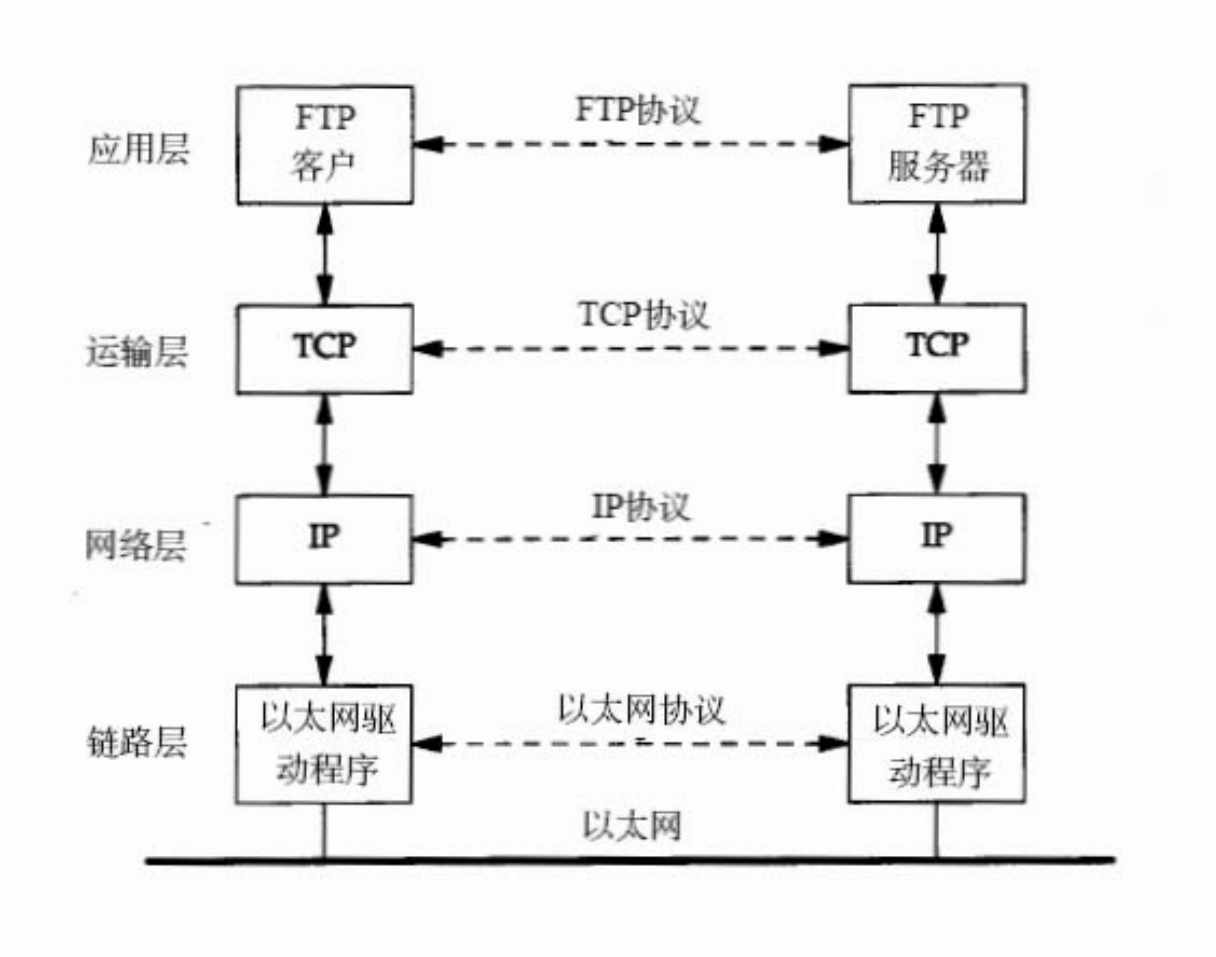
- 常用协议
  - TCP(Transport Control Protocol)传输控制协议
  - IP(Internetworking Protocol)网间网协议
  - UDP(User Datagram Protocol)用户数据报协议
  - ICMP(Internet Control Message Protocol)互联网控制信息协议
  - SMTP(Simple Mail Transfer Protocol)简单邮件传输协议
  - SNMP(Simple Network manage Protocol)简单网络管理协议
  - HTTP(Hypertext Transfer Protocol) 超文本传输协议
  - FTP(File Transfer Protocol)文件传输协议
  - ARP(Address Resolation Protocol)地址解析协议

2.3 分层

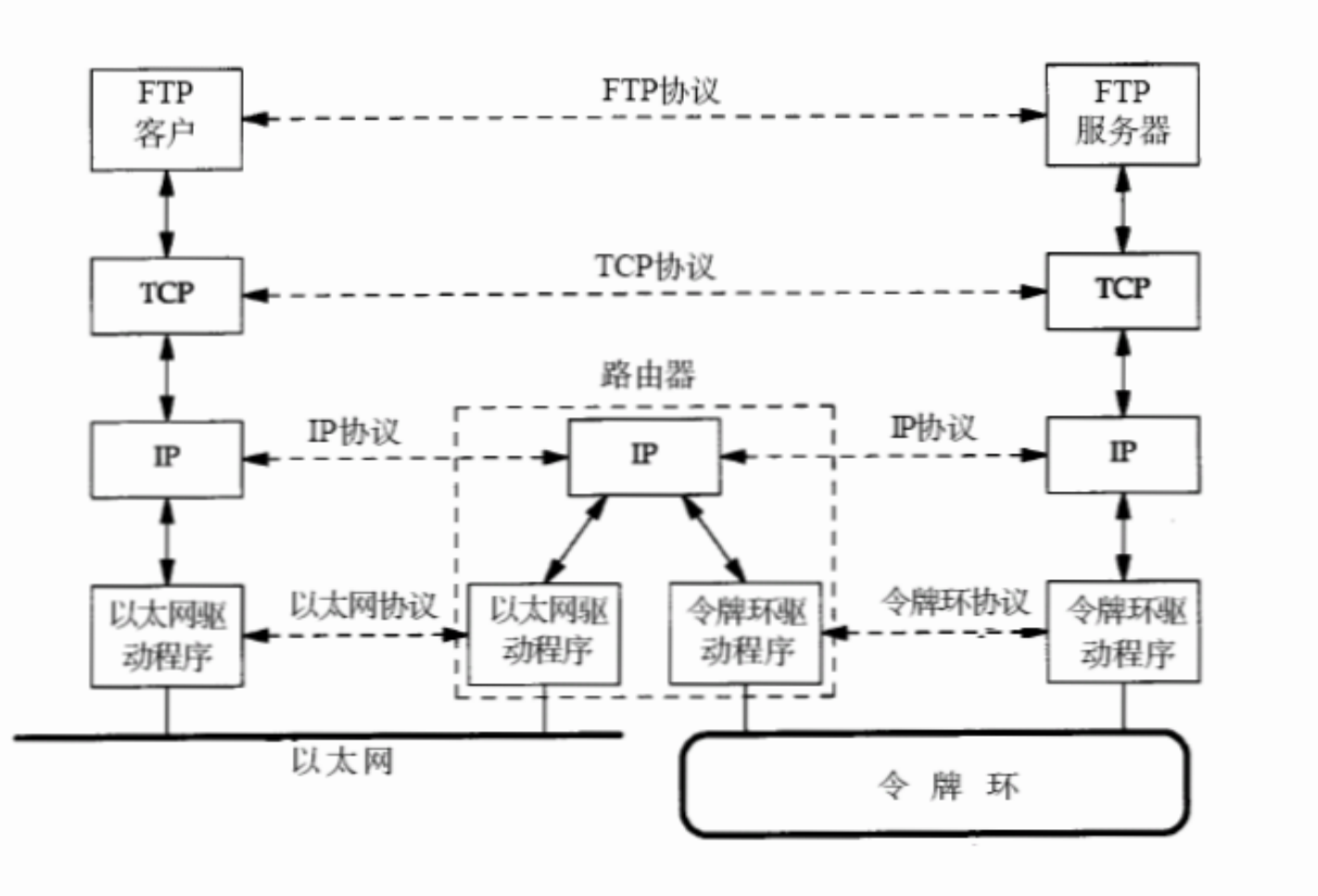
- TCP/IP通常被认为是一个四层协议系统，如右图所示，每一层负责不同的功能：
  - 链路层(有时也称作数据链路层或网络接口层)，通常包括操作系统中的设备驱动程序和计算机中对应的网络接口卡。它们一起处理与电缆（或其他任何传输媒介）的物理接口细节。
  - 网络层（有时也称作互联网层）处理分组在网络中的活动，例如分组的选路。
    - 在TCP/IP协议族中网络层协议包括
      - IP协议（网际协议），
      - ICMP协议（Internet互联网控制报文协议），
      - 以及IGMP协议（Internet组管理协议）。
  - 传输层主要为两台主机上的应用程序提供端到端的通信。
    - 在TCP/IP协议族中，有两个互不相同的传输协议：TCP（传输控制协议）和UDP（用户数据报协议）。
      - TCP为两台主机提供高可靠性的数据通信。它所做的工作包括把应用程序交给它的数据分成合适的小块交给下面的网络层，确认接收到的分组，设置发送最后确认分组的超时时钟等。由于运输层提供了高可靠性的端到端的通信，因此应用层可以忽略所有这些细节。
      - 而另一方面，UDP则为应用层提供一种非常简单的服务。它只是把称作数据报的分组从一台主机发送到另一台主机，但并不保证该数据报能到达另一端。任何必需的可靠性必须由应用层来提供。
  - 应用层负责处理特定的应用程序细节



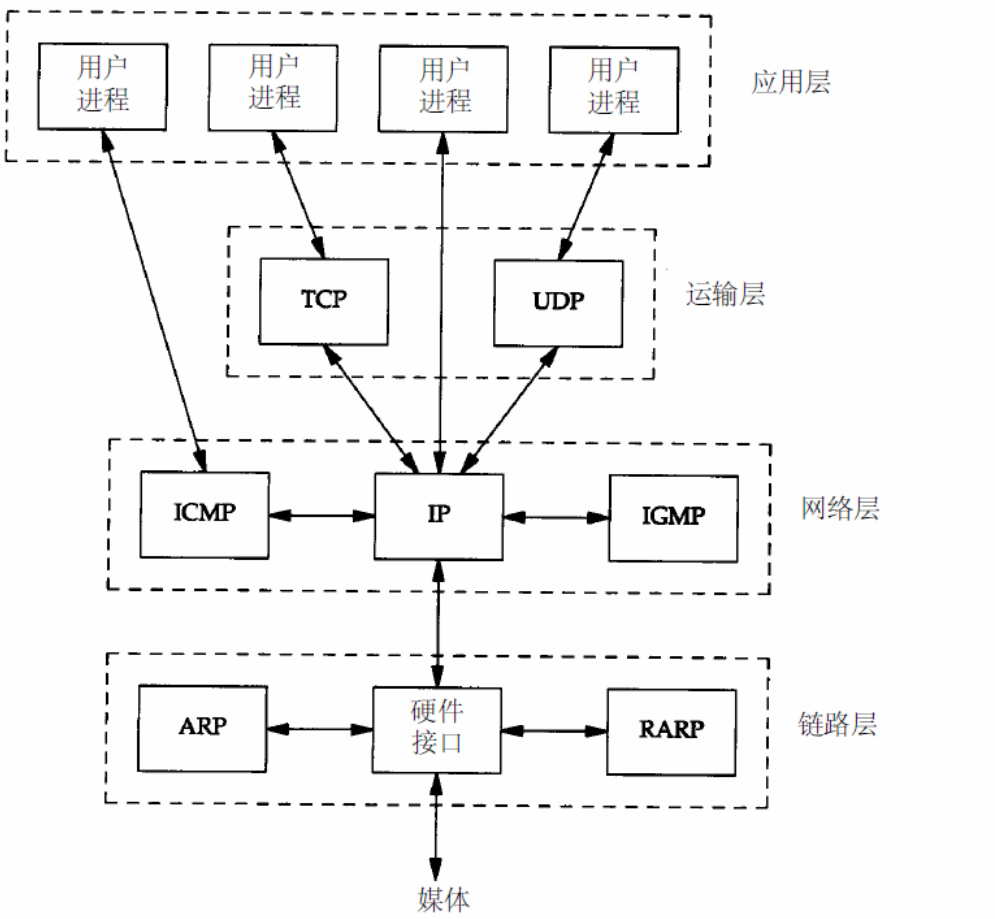
2.3.1 局域网上运行FTP的两台主机



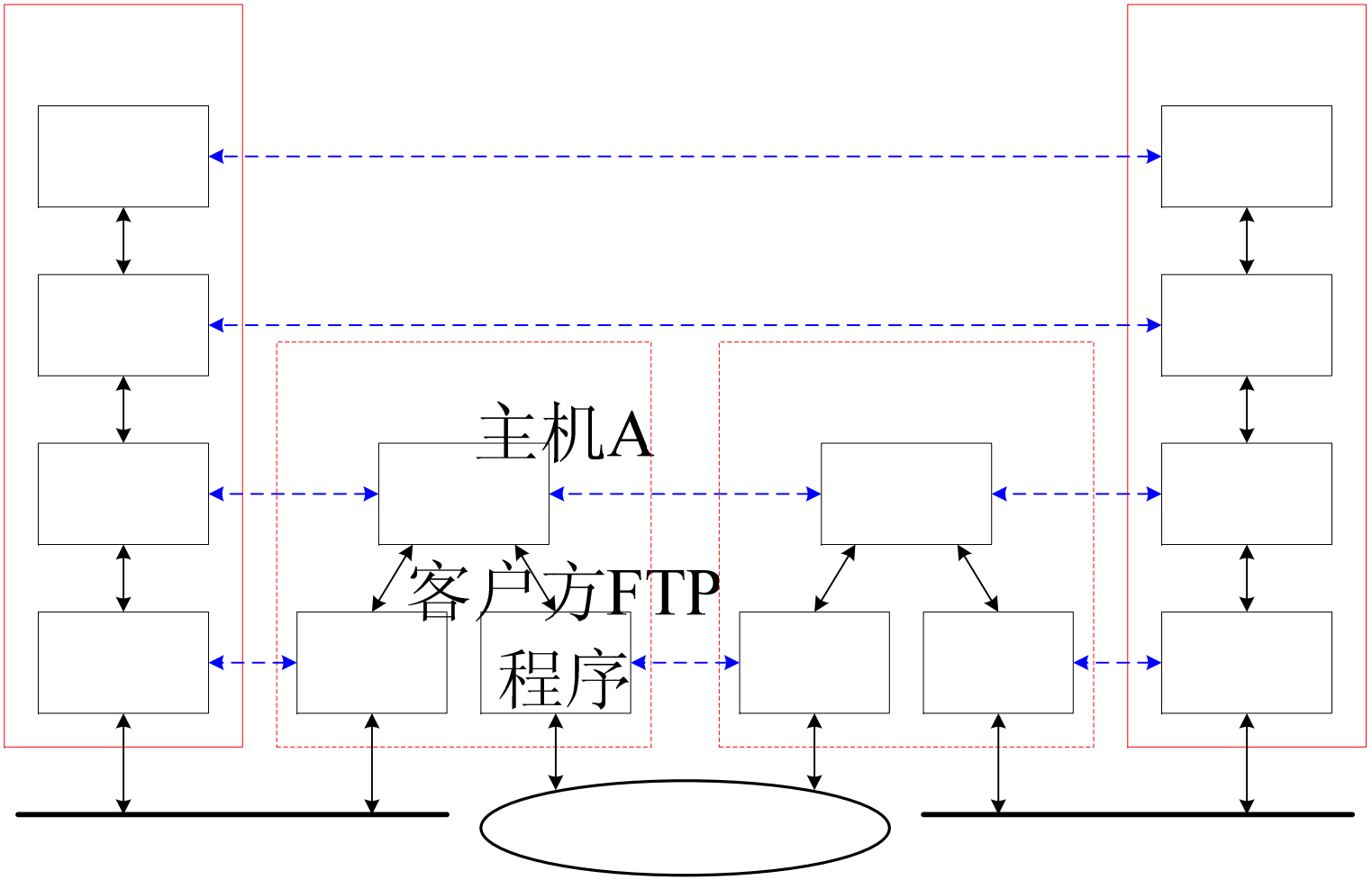
2.3.2 通过路由器连接的两个网络



2.3.3 TCP/IP协议族中不同层次的协议



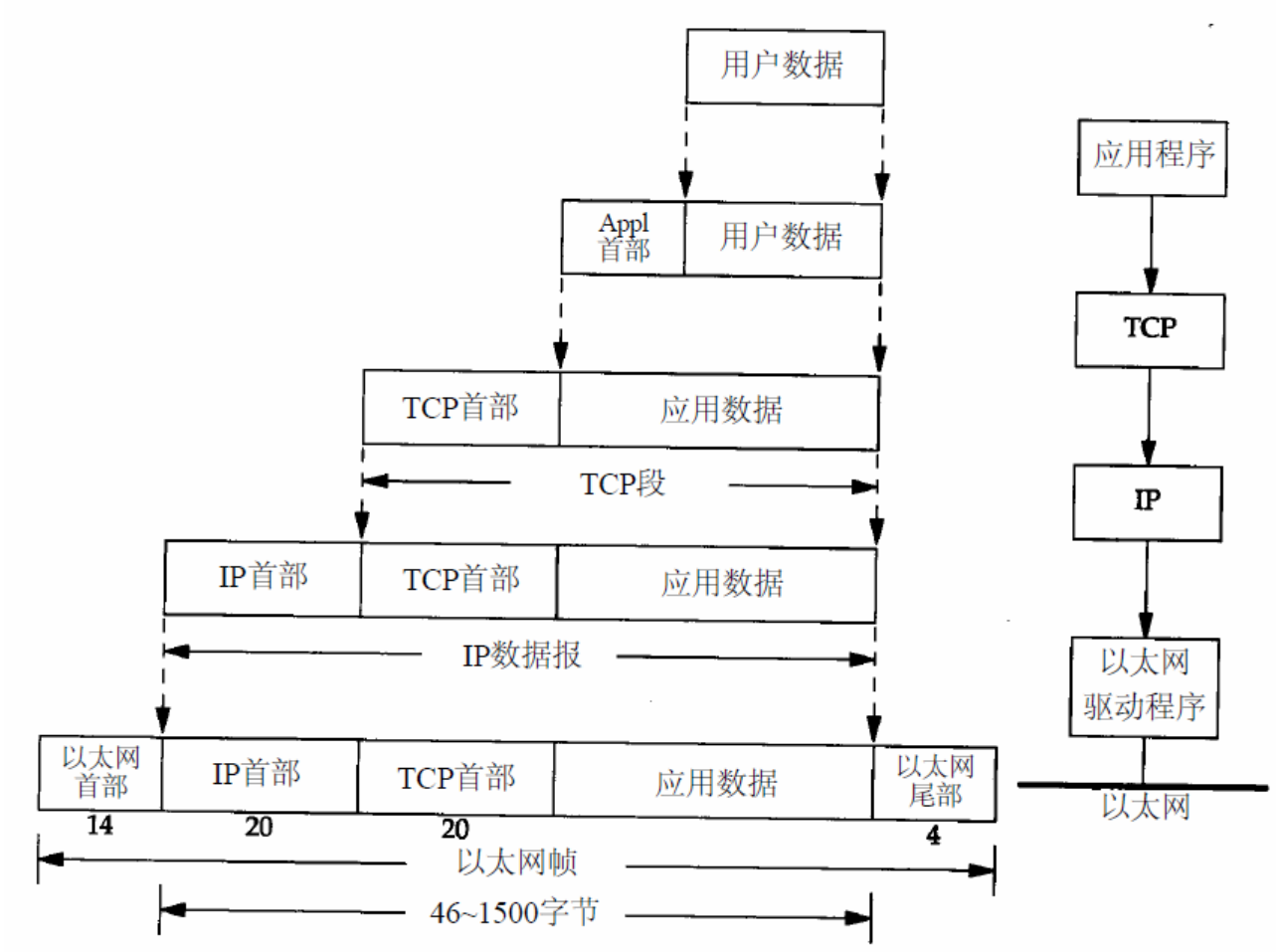
2.3.4 TCP/IP协议通信模型



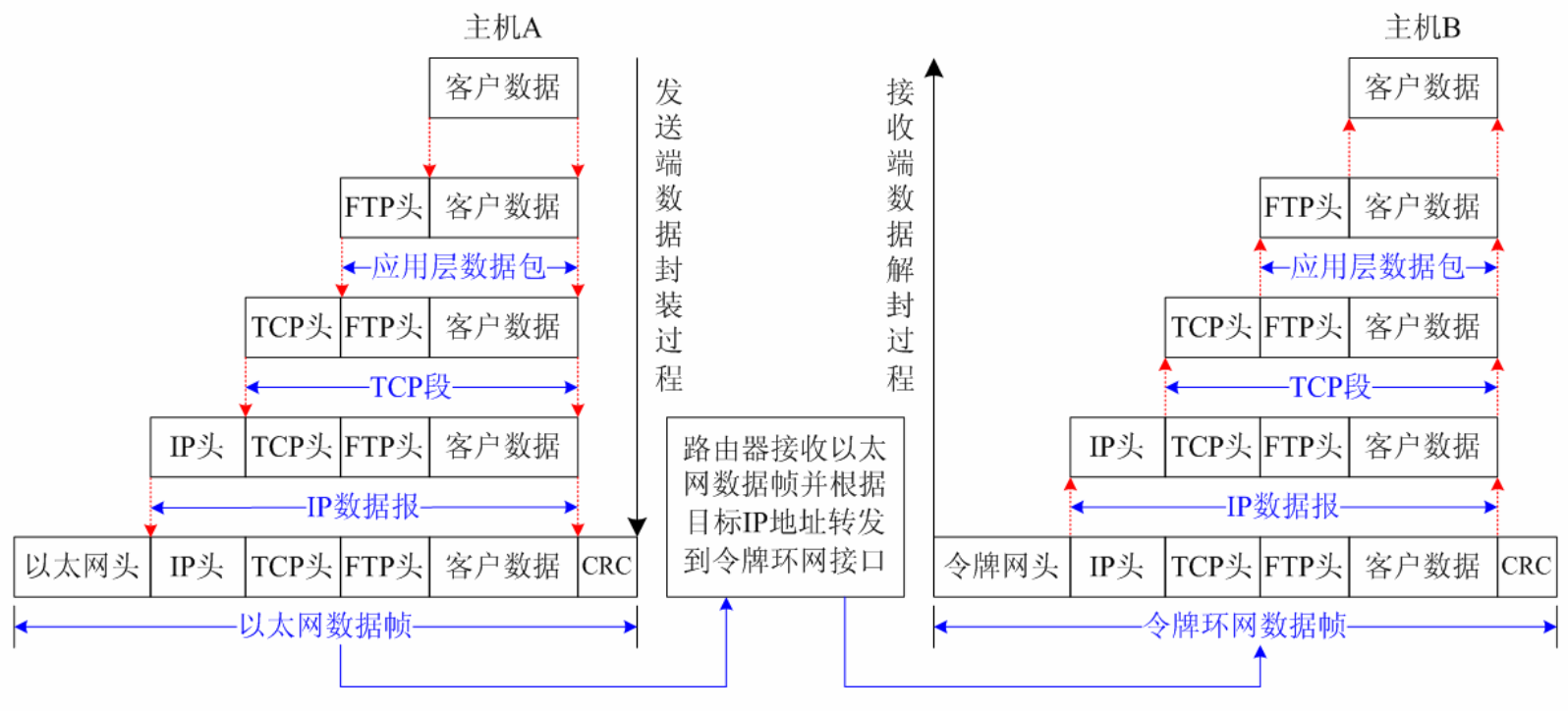
2.4 封装

- 当应用程序用TCP传送数据时，数据被送入协议栈中，然后逐个通过每一层直到被当作一串比特流送入网络。
  - 每一层对收到的数据都要增加一些首部信息（有时还要增加尾部信息）
  - TCP传给IP的数据单元称作TCP报文段或简称为TCP段（TCP segment）
  - IP传给网络接口层的数据单元称作IP数据报(IP datagram)
  - 通过以太网传输的比特流称作帧(Frame)
- UDP数据与TCP数据基本一致
  - 唯一的不同是UDP传给IP的信息单元称作UDP数据报（UDP datagram），而且UDP的首部长为8字节

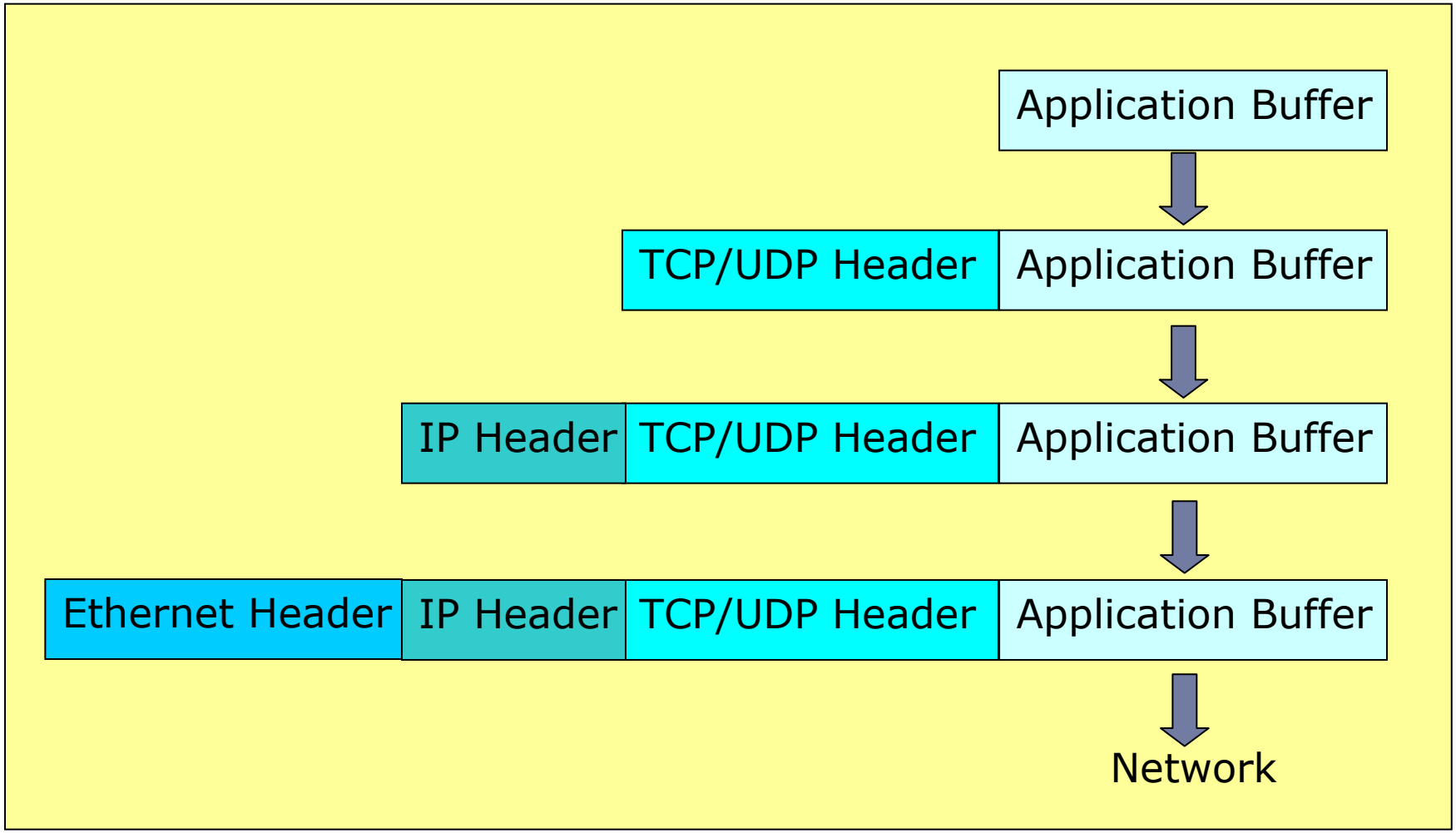
2.4.1 数据进入协议栈时的封装过程



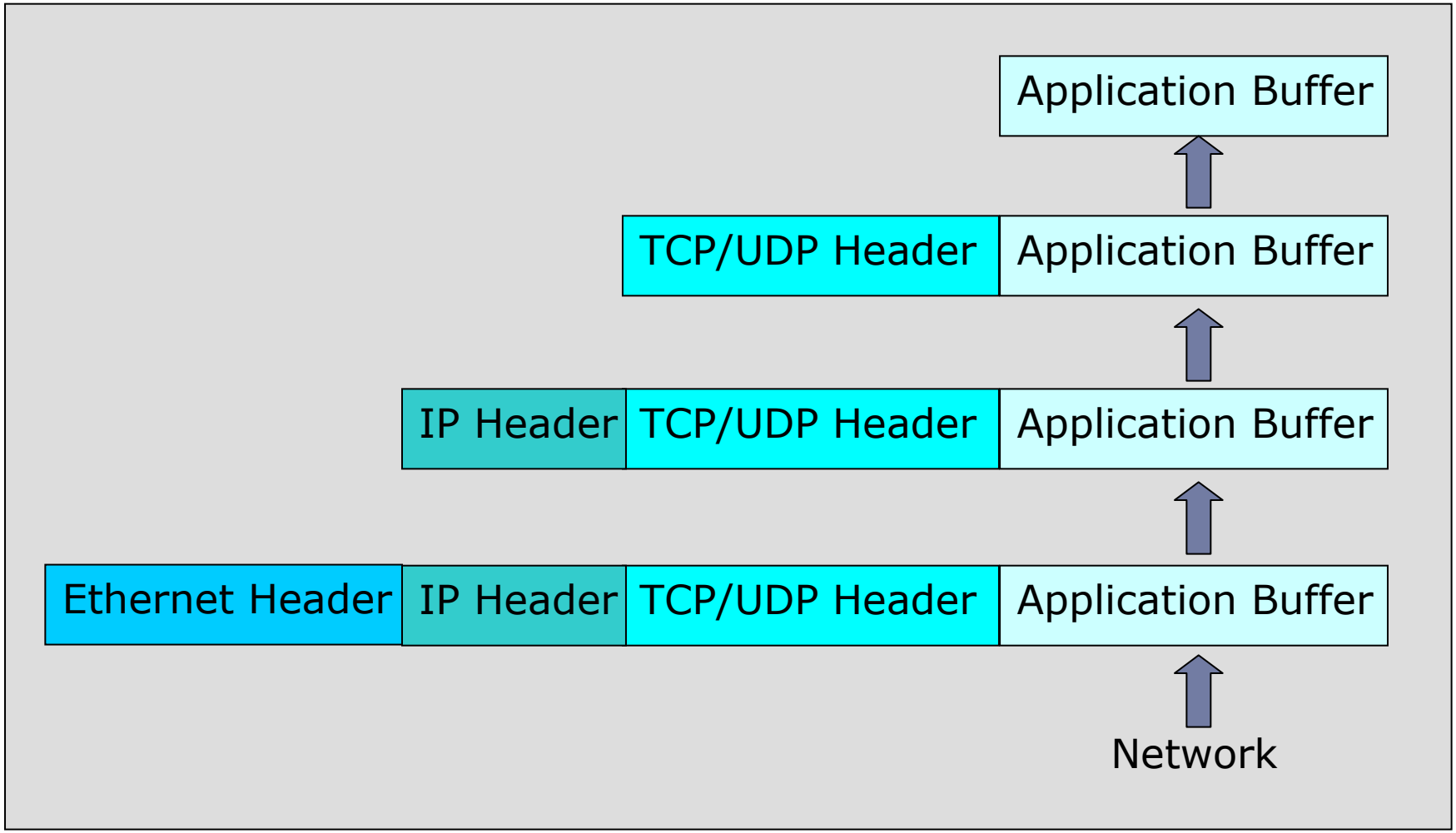
2.4.2 数据的封装与传递过程



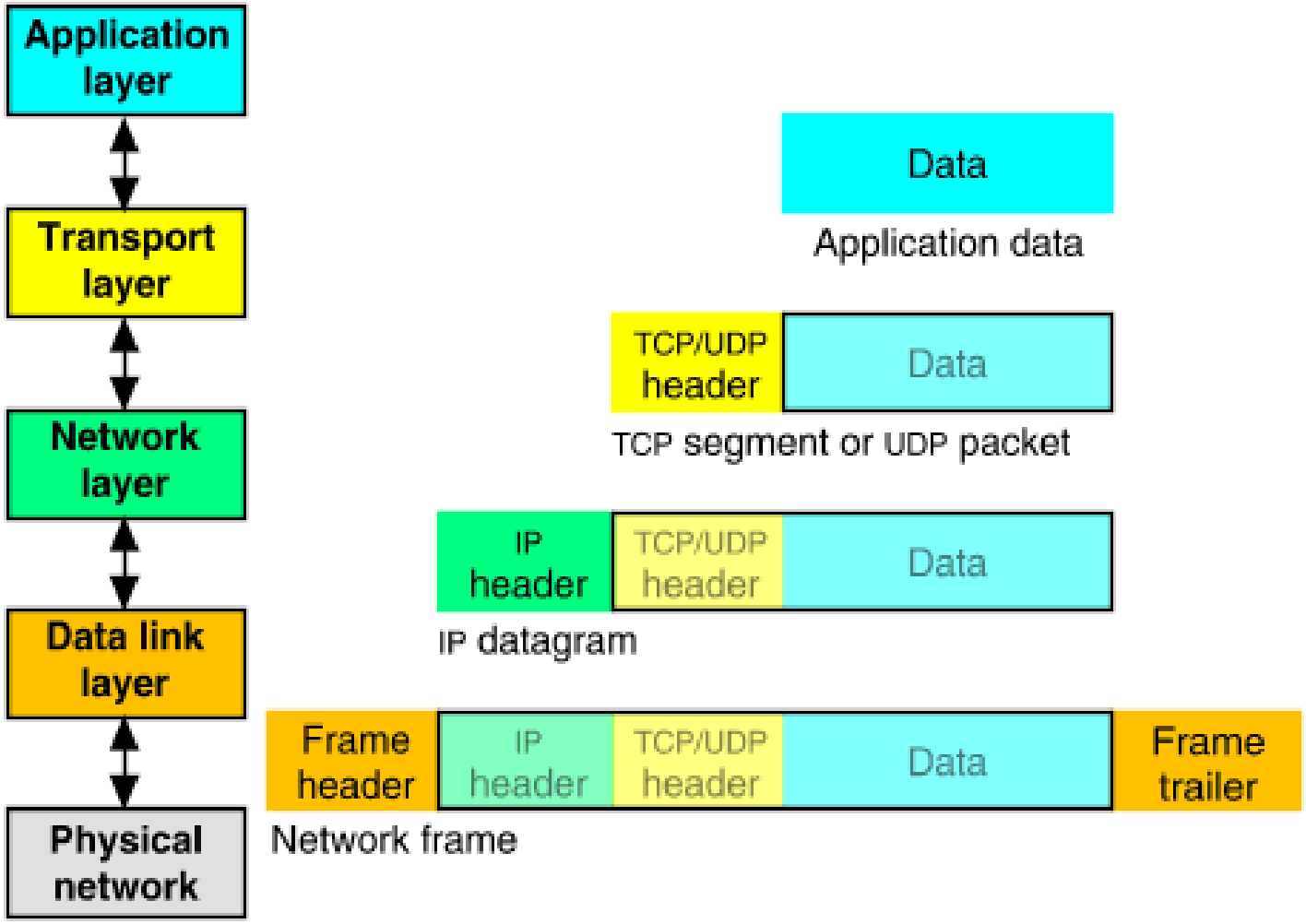
2.4.2.1 发送端-数据打包



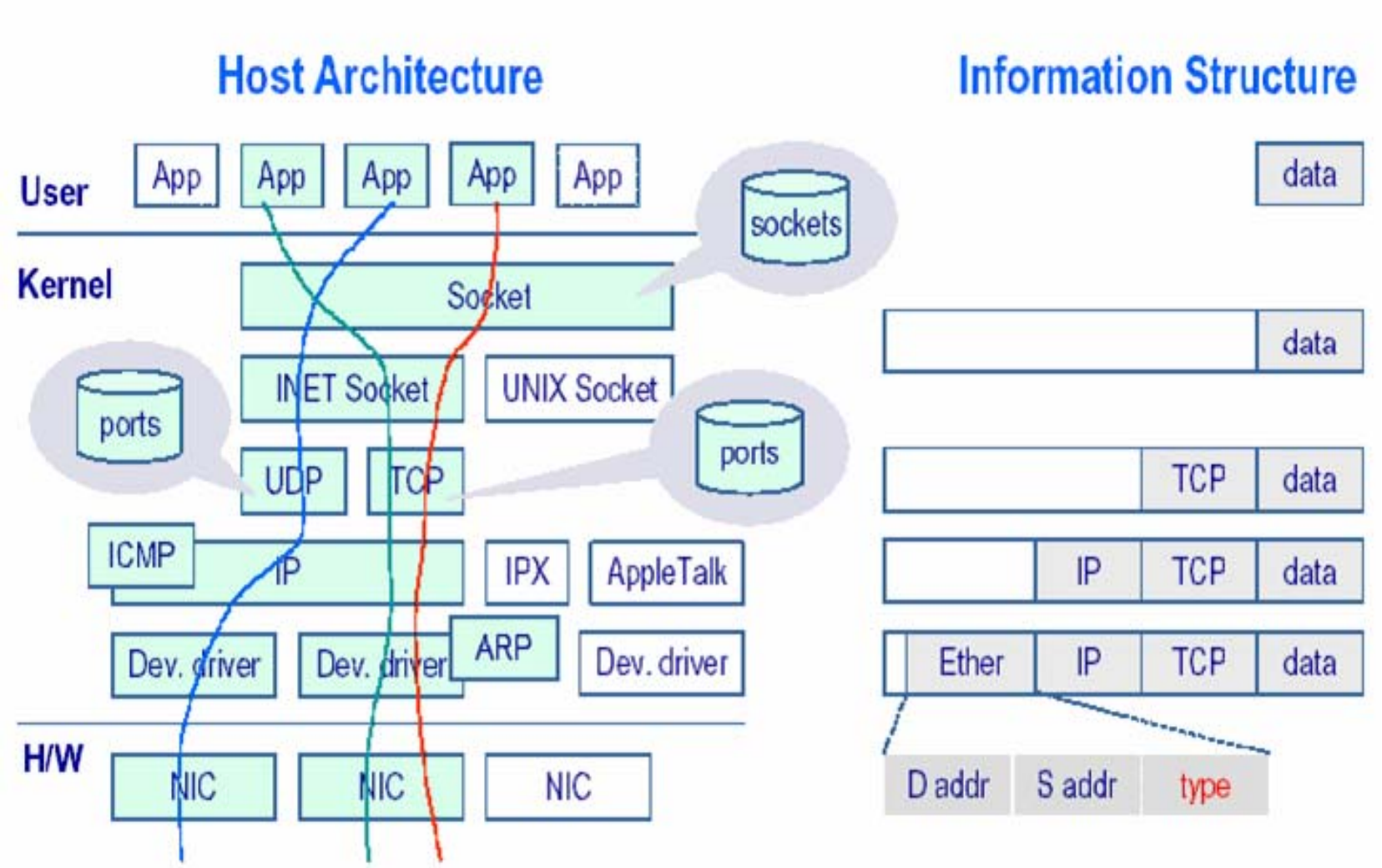
2.4.2.2 接收端-数据解包



A day in the life of Network Packet

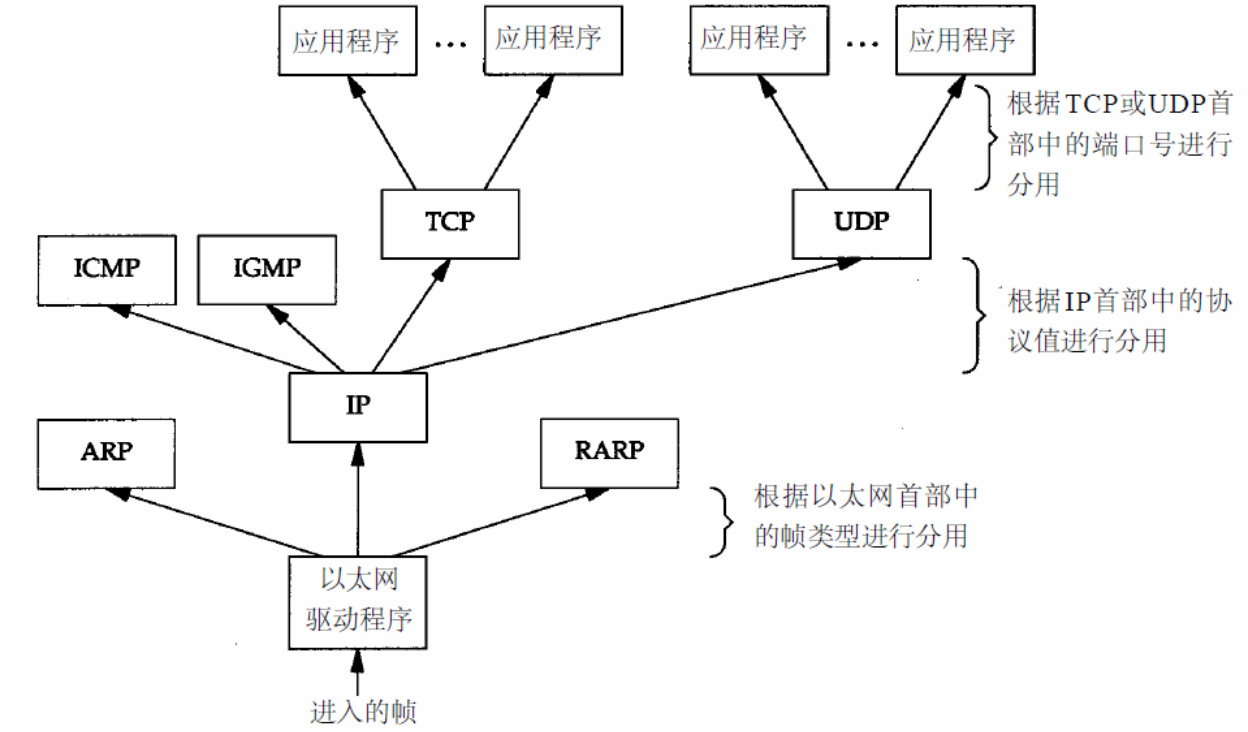


The gory details …..



2.5 分用(Demultiplexing)

- 当目的主机收到一个以太网数据帧时，数据就开始从协议栈中由底向上升，同时去掉各层协议加上的报文首部。每层协议盒都要去检查报文首部中的协议标识，以确定接收数据的上层协议。这个过程称作分用（ Demultiplexing），下图显示了该过程是如何发生的。



- 在分用的同时还有数据包的过滤过程

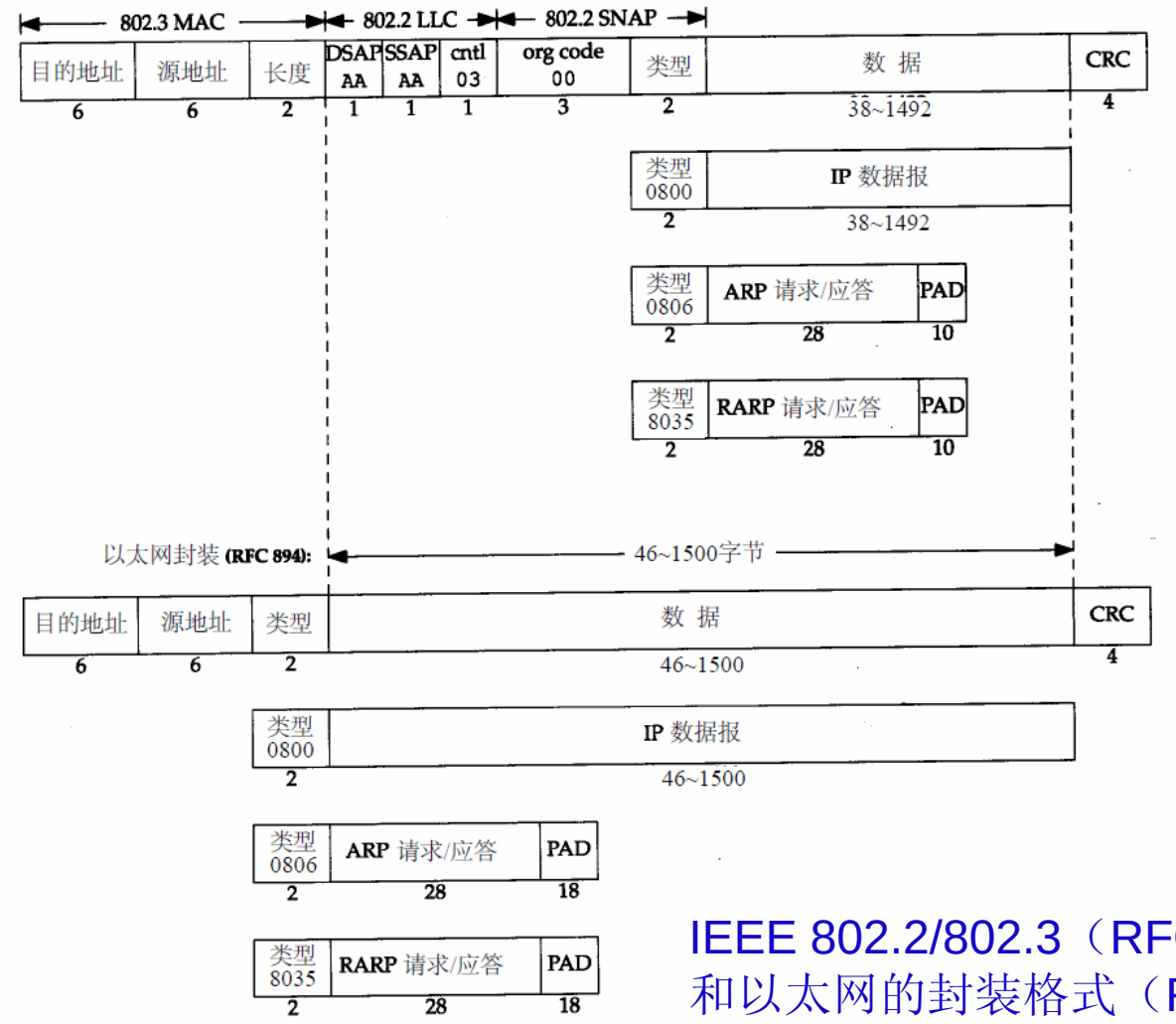
2.6 链路层

- 链路层主要有三个目的：
  - 为IP模块发送和接收IP数据报
  - 为ARP模块发送ARP请求和接收ARP应答
  - 为RARP发送RARP请求和接收RARP应答
- TCP/IP支持多种不同的链路层协议，这取决于网络所使用的硬件
  - 以太网
  - 令牌环网
  - 令牌总线
  - FDDI（光纤分布式数据接口）
  - RS-232串行线路等
  - 无线局域网(WLAN)
  - ……

2.6.1 以太网和IEEE 802封装

- 以太网这个术语一般是指数字设备公司（Digital Equipment Corp.）、英特尔公司（Intel Corp.）和Xerox公司在1982年联合公布的一个标准。
  - 它是当今TCP/IP采用的主要的局域网技术
  - 它采用一种称作CSMA/CD的媒体接入方法，其意思是带冲突检测的载波侦听多路接入（Carrier Sense, Multiple Access with Collision Detection）。
  - 它的速率为10Mb/s，地址为48bit。
- 几年后，IEEE（电子电气工程师协会）802委员会公布了一个稍有不同的标准集
  - 802.3针对整个CSMA/CD网络
  - 802.4针对令牌总线网络
  - 802.5针对令牌环网络
  - 这三者的共同特性由802.2标准来定义，那就是802网络共有的逻辑链路控制（LLC）
    - 不幸的是，802.2和802.3定义了一个与以太网不同的帧格式。文献[Stallings1987]对所有的IEEE802标准进行了详细的介绍。
- 在TCP/IP世界中，以太网IP数据报的封装是在RFC894[Hornig1984]中定义的，IEEE802网络的IP数据报封装是在RFC1042[PostelandReynolds1988]中定义的。主机需求RFC要求每台Internet主机都与一个10Mb/s的以太网电缆相连接：
  - 必须能发送和接收采用RFC894（以太网）封装格式的分组。
  - 应该能接收与RFC894混合的RFC1042（IEEE802）封装格式的分组。
  - 也许能够发送采用RFC1042格式封装的分组。如果主机能同时发送两种类型的分组数据，那么发送的分组必须是可以设置的，而且默认条件下必须是RFC894分组。

2.6.2 以太网封装格式



IEEE 802.2/802.3（RFC 1042）
和以太网的封装格式（RFC 894）

2.6.3 最大传输单元MTU

- 以太网和802.3对数据帧的长度都有一个限制，其最大值分别是1500和1492字节。链路层的这个特性称作MTU（最大传输单元），不同类型的网络大多数都有一个上限。
- 如果IP层有一个数据报要传，而且数据的长度比链路层的MTU还大，那么IP层就需要进行分片（fragmentation），把数据报分成若干片，这样每一片都小于MTU。
- 下图列出了一些典型的MTU值，它们摘自RFC1191[Mogul and Deering 1990]。
- 使用netstat命令可以打印出网络接口的MTU。

网　　络	MTU字节
超通道	65535
16 Mb/s令牌环(IBM)	17914
4 Mb/s令牌环(IEEE 802.5)	4464
FDDI	4352
以太网	1500
IEEE 802.3/802.2	1492
X.25	576
点对点(低时延)	296

2.6.4 路径MTU

- 当在同一个网络上的两台主机互相进行通信时，该网络的MTU是非常重要的
- 如果两台主机之间的通信要通过多个网络，那么每个网络的链路层就可能有不同的MTU，重要的不是两台主机所在网络的MTU的值，重要的是两台通信主机路径中的最小MTU，它被称作路径MTU
- 两台主机之间的路径MTU不一定是个常数，它取决于当时所选择的路由，而选路不一定是对称的（从A到B的路由可能与从B到A的路由不同），因此路径MTU在两个方向上不一定是一致的。
- RFC1191[Mogul and Deering 1990]描述了路径MTU的发现机制，即在任何时候确定路径MTU的方法
  - ICMP的不可到达错误
  - traceroute

2.6.5 CSMA/CD

- 总线型LAN中，所有的节点对信道的访问是以多路访问方式进行的。
- 任一节点都可以将数据帧发送到总线上，所有连接在信道上的节点都能检测到该帧。
- 当目的节点检测到该数据帧的目的地址（MAC地址）为本节点地址时，就继续接收该帧中包含的数据，同时给源节点返回一个响应。
- 当有两个或更多的节点在同一时间都发送了数据，在信道上就造成了帧的重叠，导致冲突出现。
- 为了克服这种冲突，在总线LAN中常采用CSMA/CD协议，即带有冲突检测的载波侦听多路访问协议，它是一种随机争用型的介质访问控制方法。

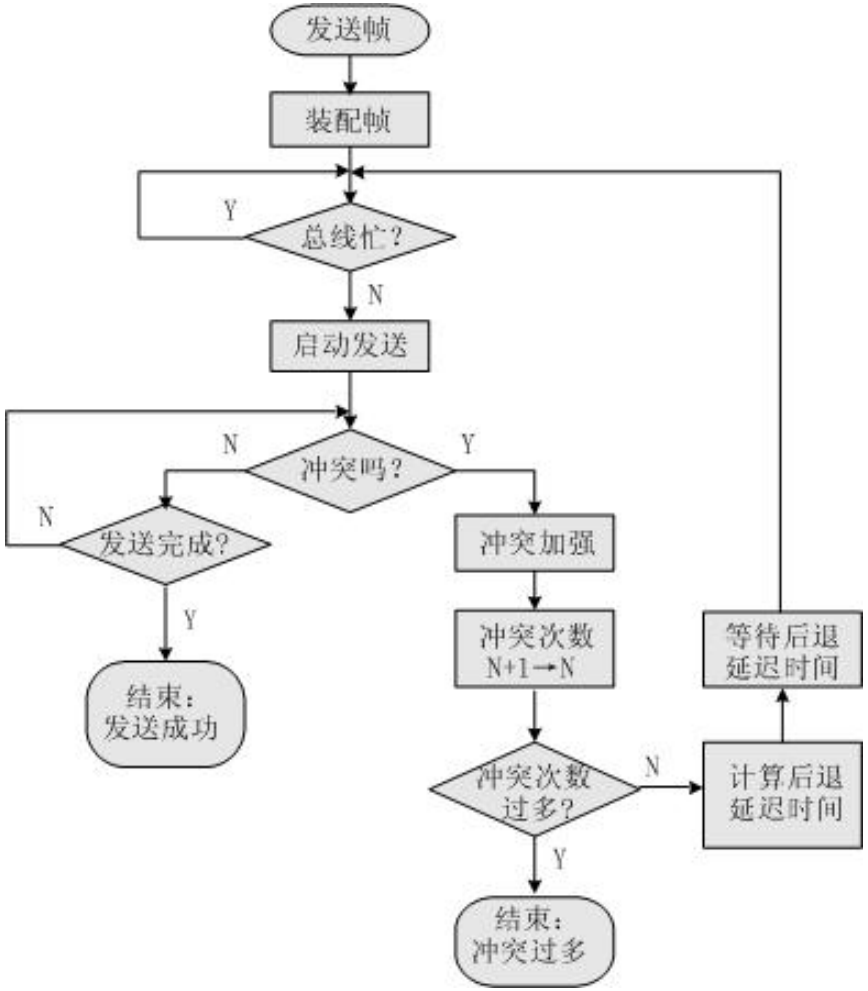
2.6.5 CSMA/CD协议的特点

- 在采用CSMA/CD协议的总线LAN中，各节点通过竞争的方法强占对媒体的访问权利，出现冲突后，必须延迟重发。因此，节点从准备发送数据到成功发送数据的时间是不能确定的，它不适合传输对时延要求较高的实时性数据。
- 结构简单、网络维护方便、增删节点容易，网络在轻负载（节点数较少）的情况下效率较高。
- 随着网络中节点数量的增加，传递信息量增大，即在重负载时，冲突概率增加，总线LAN的性能就会明显下降。

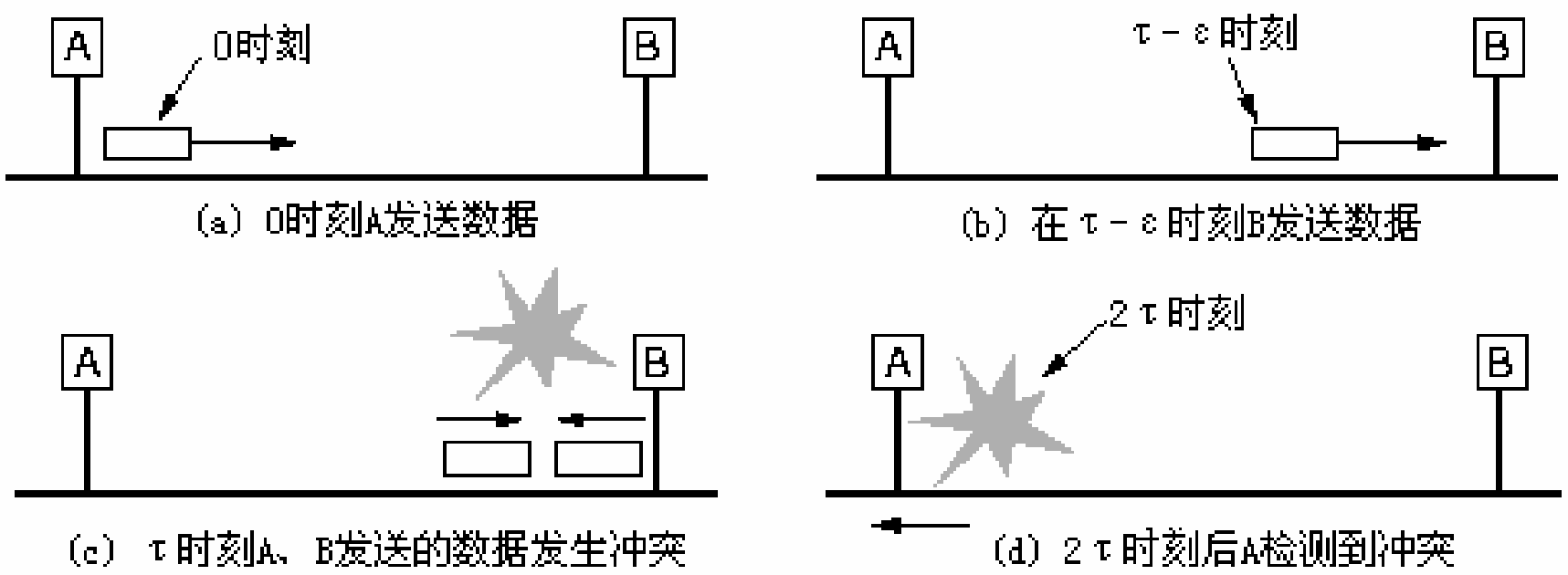
2.6.5 CSMA/CD发送

- 某站点想要发送数据，它必须
  1. 首先侦听信道；
  2. 如果信道空闲，立即发送数据并进行冲突检测；
  3. 如果信道忙，继续侦听信道，直到信道变为空闲，立即发送数据并进行冲突检测；
  4. 如果在发送数据过程中检测到冲突，立即停止发送数据，并等待一随机长的时间，重新回到1

2.6.5 CSMA/CD发送流程



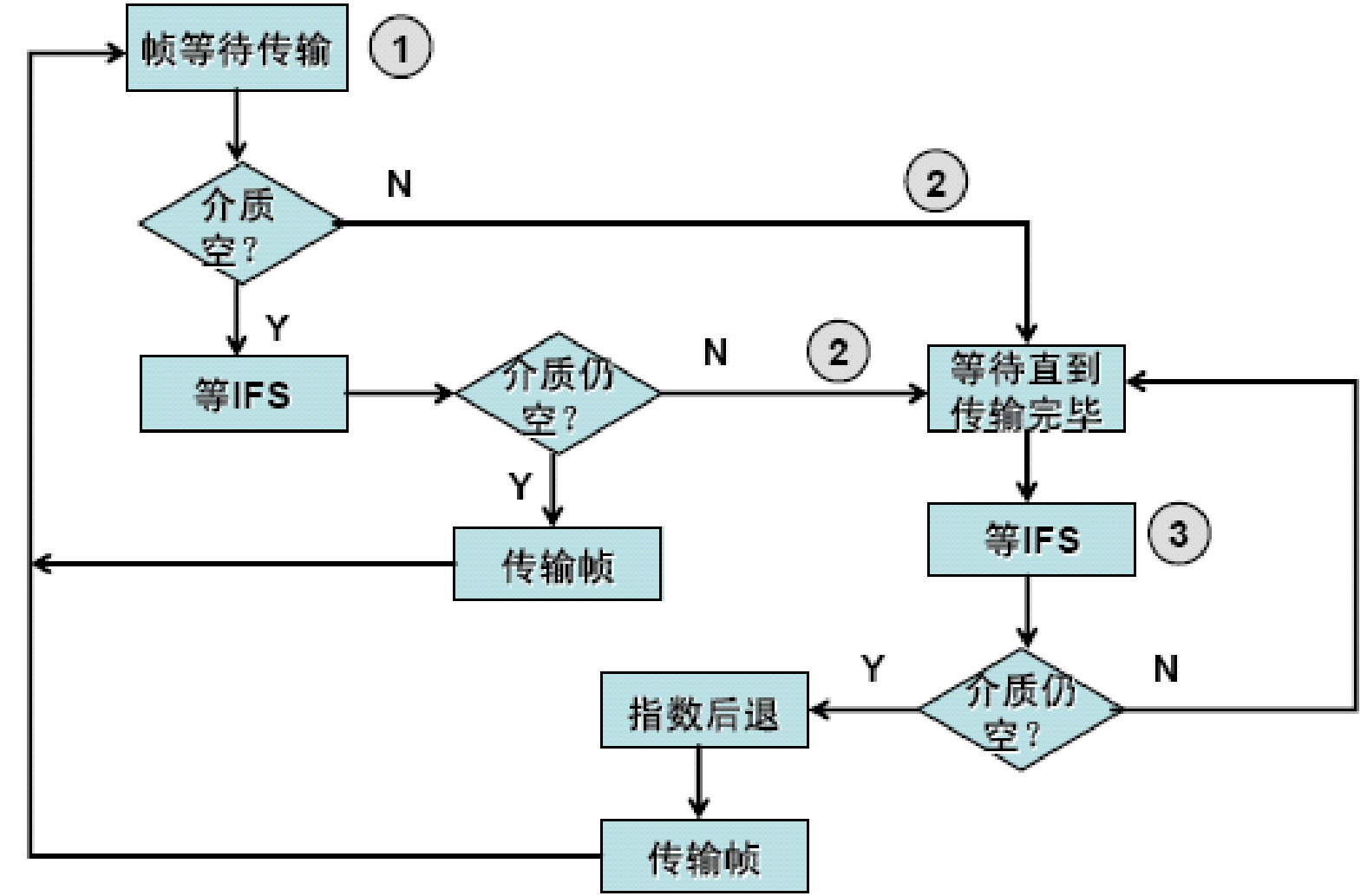
2.6.5 CSMA/CD冲突窗口



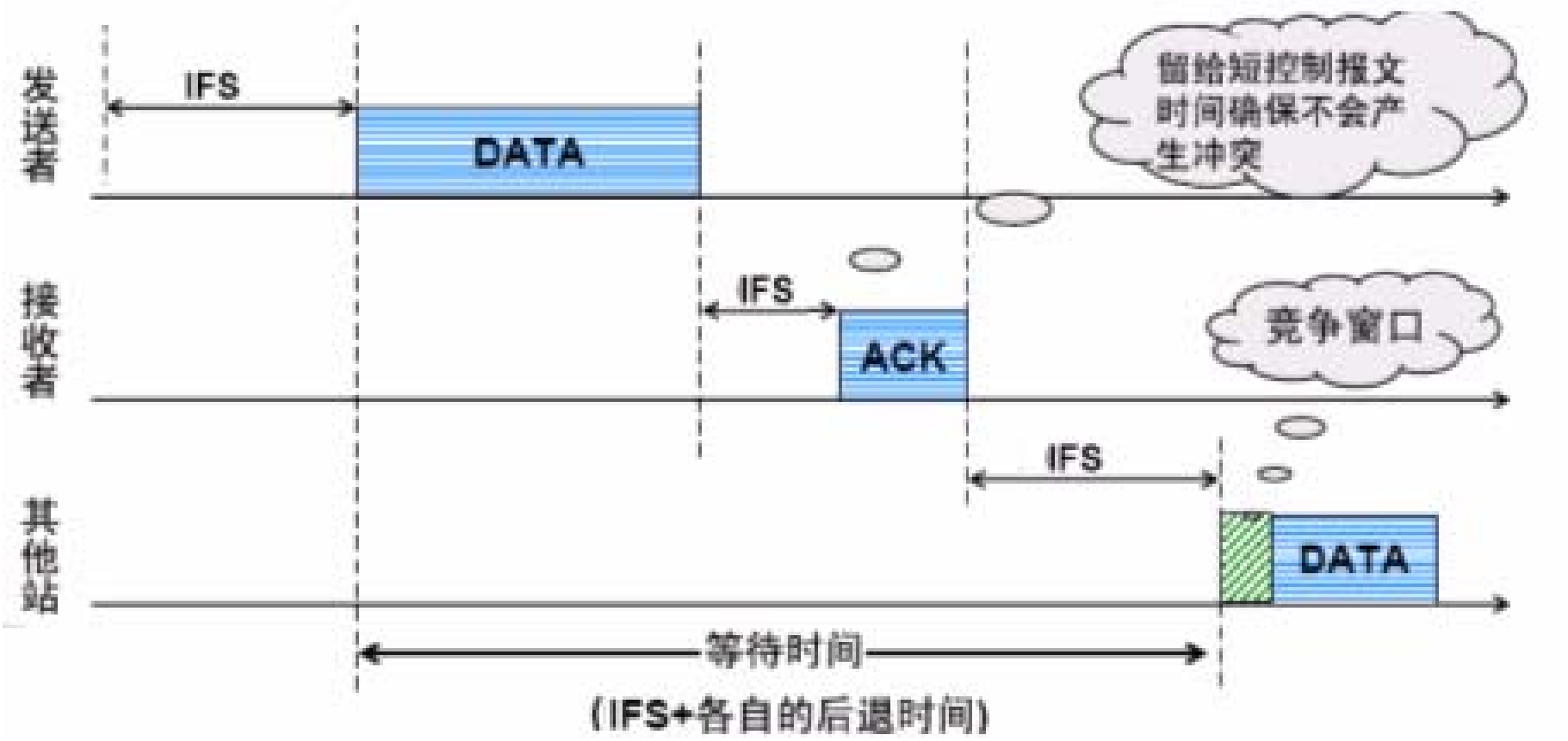
2.6.6 CSMA/CA

- CSMA/CA的工作过程如下：
  - 如果某站点有数据要发送，它首先侦听信道：
    1. 如果信道空闲，继续等待IFS（Inter Frame Space）时间，然后再侦听信道如果仍然空闲，立即发送数据。
    2. 如果信道忙，该站点继续侦听信道直到当前传输完全结束。
    3. 一旦当前传输结束，站点继续等待IFS时间，然后再侦听信道，如果信道仍然保持空闲，节点按照指数后退一个随机长的时间后，发送数据。

2.6.6 CSMA/CA算法



2.6.6 发送DATA帧以及接收ACK帧

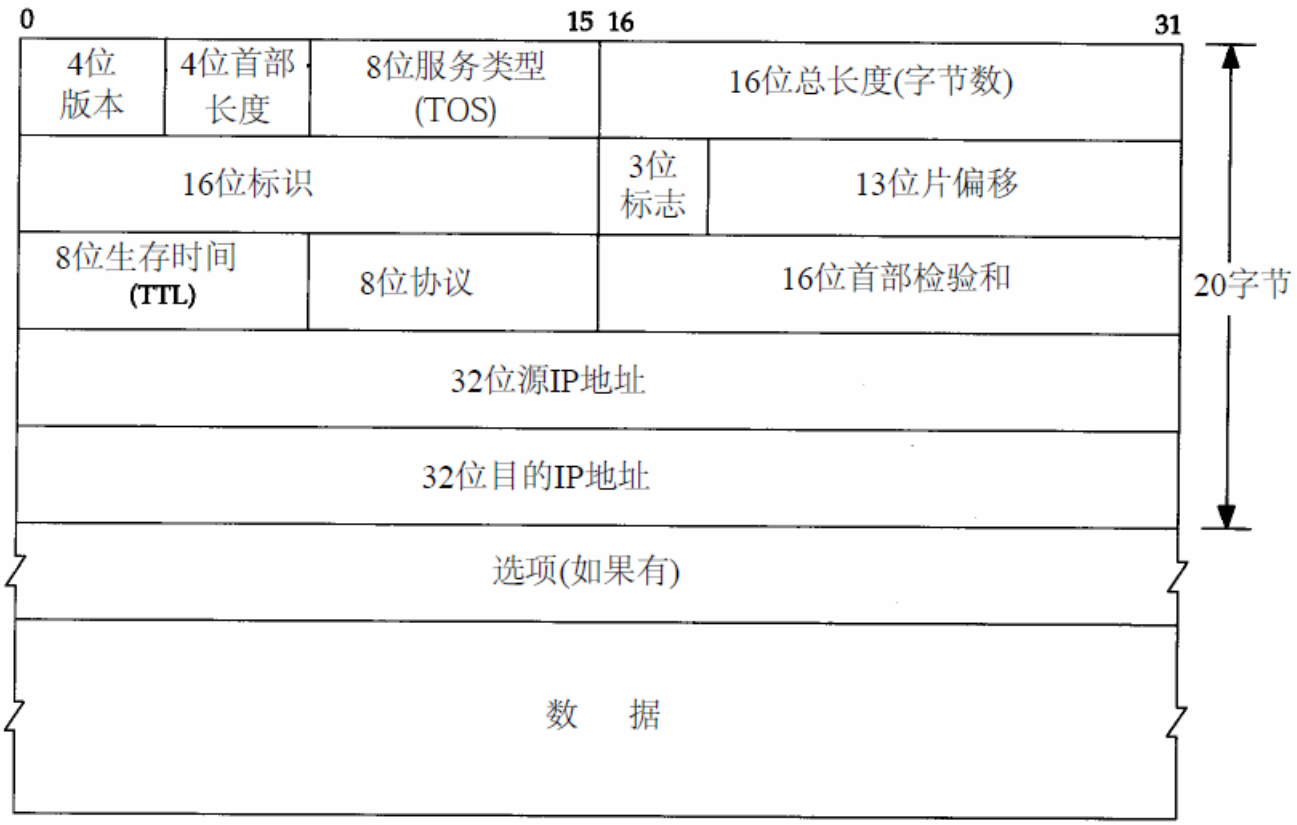


2.7 IP协议

- IP是TCP/IP协议族中最为核心的协议，所有的TCP、UDP、ICMP及IGMP数据都以IP数据报格式传输
- IP提供不可靠、无连接的数据报传送服务
  - 不可靠（unreliable）的意思是它不能保证IP数据报能成功地到达目的地
    - IP仅提供最好的传输服务，如果发生某种错误时，如某个路由器暂时用完了缓冲区，IP有一个简单的错误处理算法：丢弃该数据报，然后发送ICMP消息报给信源端
    - 任何要求的可靠性必须由上层来提供（如TCP）
  - 无连接（connectionless）这个术语的意思是IP并不维护任何关于后续数据报的状态信息，每个数据报的处理是相互独立的
    - IP数据报可以不按发送顺序接收如果一信源向相同的信宿发送两个连续的数据报（先是A，然后是B），每个数据报都是独立地进行路由选择，可能选择不同的路线，因此B可能在A到达之前先到达

2.7.1 IP头

- 普通的IP首部长为20个字节，除非含有选项字段， IP数据报的格式下图所示

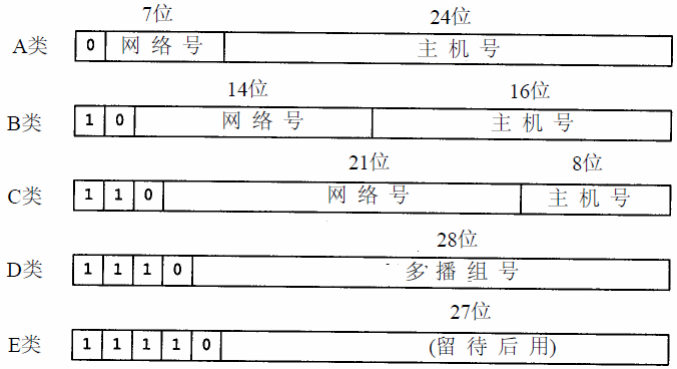


2.7.1 IP头

- 版本号，IPv4取值4
- 首部长度，4个字节为单位，取值范围5～15
- 服务类型（TOS），指定传输的优先级、传输速度、可靠性和吞吐量等
- 报文总长度，最大长度为65535字节
- 报文标识，唯一标识一个数据报，如果数据报分段，则每个分段的标识都一样
- 标志，最高位未使用，定义为0，其余两位为DF（不分段）和MF（更多分段）
- 段偏移量，以8个字节为单位，指出该分段的第一个数据字在原始数据报中的偏移位置
- 生存时间（ TTL, time-to-live），取值0～255，每经过一个路由节点减1，为0时被丢弃
- 协议，指明该数据报的协议类型，如1为ICMP，4为IP，6为TCP，17为UDP等
- 首部校验和，每通过一次网关都要重新计算该值，用于保证IP首部的完整性
- 选项，长度可变，提供某些场合下需要的控制功能，IP首部的长度必须是4个字节的整数倍，如果选项长度不是4的整数倍，必须填充数据

2.7.2 IP地址

- 互联网上的每个接口必须有一个唯一的Internet地址（也称作IP地址），IPv4地址长32bit，IPv6地址长为128bit
- 传统上，IPv4地址通过分类来控制子网规模，五类不同的互联网地址格式及地址范围如下图所示



类型	范围
A	0.0.0.0 到 127.255.255.255
B	128.0.0.0 到 191.255.255.255
C	192.0.0.0 到 223.255.255.255
D	224.0.0.0 到 239.255.255.255
E	240.0.0.0 到 247.255.255.255

- CIDR是更为合理的地址分配方式
  - 更精确的控制子网规模
  - 减小路由表的大小

地址表示方法

- IPv4，采用十进制点分法表示
  - 192.168.1.108
  - 有些时候，也采用十六进制表示，这种方法不常见
  - 在程序里，使用二进制表示
- IPv6，采用十六进制，以冒号分割的方式，也有时候采用点分割，点分割的方式不常见
  - fe80::20d:60ff:fecb:86e5
  - fe80:0000:0000:0000:020d:60ff:fecb:86e5
  - 有两种表示方式，长方式和短方式
    - 长方式，为0的位置也要写出来
    - 短方式，连续为0的位置，通过两个连续的冒号表示

2.7.2.1 IP地址分配

- 互联网络信息中心（Internet Network Information Centre，称作InterNIC）负责为接入互联网的网络分配IP地址
  - InterNIC只分配网络号，主机号的分配由系统管理员来负责。

2.7.2.2 IP地址分类

- 单播地址（unicast）
  - 目的为单个主机
- 广播地址（broadcast）
  - 目的端为给定网络上的所有主机
- 多播地址（multicast）
  - 目的端为同一组内的所有主机

2.7.2.3 特殊的IP地址

- 全部比特为0的地址(0.0.0.0)
  - 不能作为目的IP地址
  - 但是bind调用时，可以传递该地址，做为绑定所有接口的参数
- 全部比特为1的地址（255.255.255.255）
  - 表示绝对广播地址(也称为受限广播地址)，在不知道子网参数的时候使用，比如DHCP、ARP
- 子网号确定的情况下
  - 主机号全部为1的地址，叫子网广播地址，在子网参数确定的情况下，使用该地址进行广播通讯
  - 主机号全部为0的地址，代表一个子网，通常在配置防火墙等网络参数时候使用。

2.7.2.4 私有IP地址

- 用于私有网络，不在Internet上出现，见RFC1918、RFC4193、RFC5735
- IPv4私有地址（RFC1918）

Ranges for private IPv4 addresses.

Number of addresses	Prefix	Range
16,777,216	10/8	10.0.0.0 to 10.255.255.255
1,048,576	172.16/12	172.16.0.0 to 172.31.255.255
65,536	192.168/16	192.168.0.0 to 192.168.255.255

- IPv6的私有地址：fc00::/7
- Local-link地址(*Zero configuration networking*)：
  - IPv4: 169.254.0.0/16
  - IPv6: fe80::/10

私有IP地址

- 私有IP地址
  - A类：10.0.0.0~10.255.255.255
  - B类：172.16.0.0~172.31.255.255
  - C类：192.168.0.0~192.168.255.255
  - Local-link address: 169.254.0.0/16
- 私有IP地址的数据不能在Internet上路由
- 私有网络的数据包要想通过Internet传输，需要做NAT(网络地址转换)
  - 数据包从一个私有网络离开时(A->B)，边界的NAT设备将该数据包的目的地址不变，源地址由A变为NAT设备的外出接口IP地址，结果是NAT->B
  - 当B收到数据包时，如果需要返回，这种情况下B->NAT，则数据包到达私有网络时，NAT设备将数据包的目的IP地址，替换为私有网络内的主机IP地址，比如给A的，替换为A，最终A收到的为B->A
  - NAT设备需要记录状态，A->B，在数据包返回时，需要翻译为B->A
  - B如果也是一个私有网络的地址的时候，如何处理？
- NAT设备记录转换状态的方式不同，NAT的类型也不同
- 做P2P类应用时，需要处理NAT穿越的问题。

NAT穿越

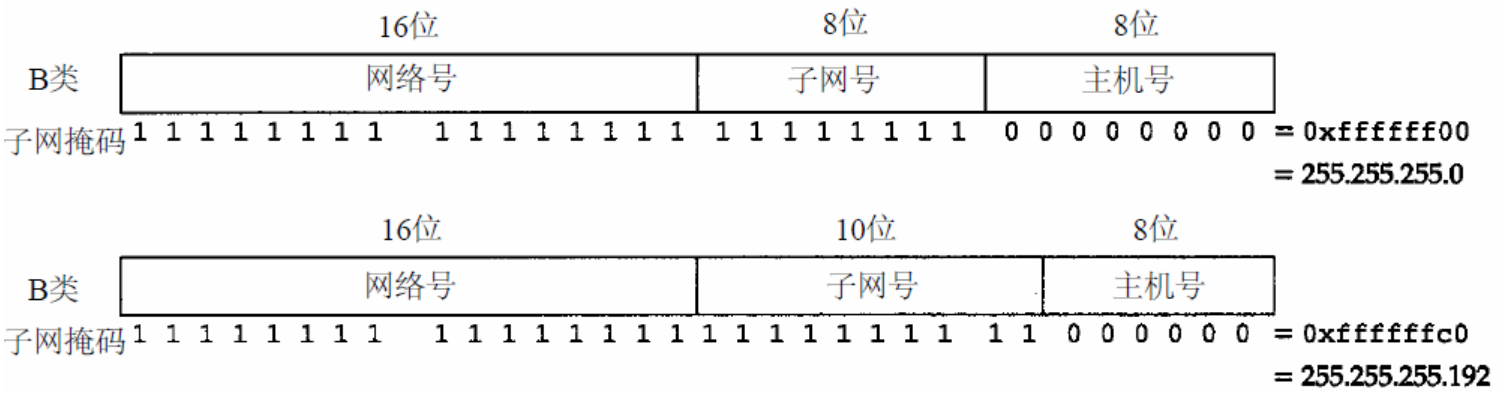
- NAT原理、类型等介绍
  - http://en.wikipedia.org/wiki/Network_address_translation
- STUN
  - http://en.wikipedia.org/wiki/STUN
- NAT穿越
  - http://www.brynosaurus.com/pub/net/p2pnat/

2.7.2.5 CIDR

- 无类别域间路由（Classless Inter-Domain Routing、CIDR）是一个用于给用户分配IP地址以及在互联网上有效地路由IP数据包的对IP地址进行归类的方法。
  - 为了解决传统按类分配IP地址空间不足问题，互联网工程工作小组在1993年发布了一新系列的标准RFC 1518和RFC 1519，以定义新的分配IP地址块和路由IPv4数据包的方法。
- 一个IP地址包含两部分：标识网络的前缀和紧接着的在这个网络内的主机地址。
  - 在传统的分类网络中，IP地址的分配把IP地址的32位按每8位为一段分开。这使得前缀必须为8，16或者24位。因此，可分配的最小的地址块有256（24位前缀，8位主机地址，28=256）个地址，而这对大多数企业来说太少了。大一点的地址块包含65536（16位前缀，16位主机，216=65536）个地址，而这对大公司来说都太多了。这导致不能充分使用IP地址和在路由上的不便，因为大量的需要单独路由的小型网络（C类网络）因在地域上分得很开而很难进行聚合路由，于是给路由设备增加了很多负担。
  - CIDR是基于可变长子网掩码（VLSM)来进行任意长度的前缀的分配的。在RFC 950（1985）中有关于可变长子网掩码的说明。
- CIDR包括：
  - 指定任意长度的前缀的可变长子网掩码技术。遵从CIDR规则的地址有一个后缀说明前缀的位数，例如192.168.0.0/16。这使得对日益缺乏的IPv4地址的使用更加有效。
  - 将多个连续的前缀聚合成超网，在互联网中，只要有可能，就显示为一个聚合的网络，因此在总体上可以减少路由表的表项数目。聚合使得互联网的路由表不用分为多级，并通过VLSM逆转“划分子网”的过程。
  - 根据机构的实际需要和短期预期需要而不是分类网络中所限定的过大或过小的地址块来管理IP地址的分配的过程。
- IPv6使用CIDR

2.7.3 子网掩码

- 除了IP地址以外，主机还需要知道有多少比特用于子网号及多少比特用于主机号。这是在引导过程中通过子网掩码来确定的。这个掩码是一个32 bit的值，其中值为1的比特留给网络号和子网号，为0的比特留给主机号
- 尽管IP地址一般以点分十进制方法表示，但是子网掩码却经常用十六进制来表示，特别是当界限不是一个字节时，因为子网掩码是一个比特掩码。
- 给定IP地址和子网掩码以后，主机就可以确定IP数据报的目的是：
  - 本子网上的主机
  - 本网络中其他子网中的主机
  - 其他网络上的主机
- 如果知道本机的IP地址，那么就知道它是否为A类、B类或C类地址(从IP地址的高位可以得知)，也就知道网络号和子网号之间的分界线。而根据子网掩码就可知道子网号与主机号之间的分界线。



子网掩码

- 通过子网掩码可以判断目标主机是否跟本主机在同一子网内
  - IP与子网掩码做位与运算，得到网络号，如果网络号相同，则跟自己在同一子网内
  - 实际上就是做路由选择的算法
- 计算网络号
  - IP & netmask
- 计算主机号
  - IP & ~netmask
- 计算子网广播地址
  - (IP & netmask) | (255.255.255.255 & ~netmask)
- 某网络IP分配为172.16.1.x，子网掩码为255.255.252.0，请计算该网络的广播地址
  - 172.16.3.255

2.7.4.1 IP路由

- 路由（名词）：
  - 数据报从源地址到目的地址所经过的道路，由一系列路由节点的地址构成。
- 路由（动词）：
  - 某个路由节点为数据报选择投递方向的选路过程。
- 路由节点：
  - 一个具有路由能力的主机或路由器，其内维护一张路由表，该表中标明了去往某个网络，应该经由的接口。
- 接口：
  - 路由节点连往某个网络的一个硬件端口。
- 路由表：
  - 由很多路由条目组成。其中最后一条是缺省路由条目。
- 条目：
  - 路由表中的一行，每个条目主要由4部分组成：网络地址，掩码，下一跳节点，接口。
- 缺省路由条目：
  - 路由表中的最后一行，该条目主要由2部分组成：下一跳节点，接口。

2.7.4.2 选路过程

- 路由节点提取出数据报的目的地址
- 查找路由表中的每一个条目，将目的地址与路由条目中的掩码做“与”的计算，如果等于该条目中的网络地址，就将该数据报从该条目所标明的接口投递给下一跳节点（若下一跳为空或者就是自身，表明目的地址在与路由节点直接相连的网络上，直接发往目的地址即可）
- 如果找不到，就按缺省路由条目所指示的方向投递

2.7.4.3 ifconfig

```
[root@nec vick]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:C2:8D:7E
          inet addr:192.168.10.223  Bcast:192.168.10.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:420 (420.0 b)
          Interrupt:10 Base address:0x10a0

eth1      Link encap:Ethernet  HWaddr 00:0C:29:C2:8D:88
          inet addr:192.168.56.136  Bcast:192.168.56.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:603 errors:0 dropped:0 overruns:0 frame:0
          TX packets:110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:55551 (54.2 Kb)  TX bytes:7601 (7.4 Kb)
          Interrupt:9 Base address:0x10c0

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:37 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3020 (2.9 Kb)  TX bytes:3020 (2.9 Kb)
```

2.7.4.4 route

- route(1)命令可以打印出当前的路由表

```
[root@nec vick]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.10.0    *               255.255.255.0   U     0      0        0 eth0
192.168.56.0    *               255.255.255.0   U     0      0        0 eth1
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         192.168.10.1    0.0.0.0         UG    0      0        0 eth0
[root@nec vick]#
```

- 对于一个给定的路由器，可以打印出五种不同的标志（flag）：
  - U 该路由可以使用
  - G 该路由是到一个网关（路由器）。如果没有设置该标志，说明目的地是直接相连的。
  - H 该路由是到一个主机，也就是说，目的地址是一个完整的主机地址。如果没有设置该标志，说明该路由是到一个网络，而目的地址是一个网络地址：一个网络号，或者网络号与子网号的组合。
  - D 该路由是由重定向报文创建的
  - M 该路由已被重定向报文修改

2.7.4.5 netstat

- netstat(1)命令也提供系统上的接口信息。-i参数将打印出接口信息，-n参数则打印出IP地址，而不是主机名字。
- 下面命令打印出每个接口的MTU、输入分组数、输入错误、输出分组数、输出错误、冲突以及当前的输出队列长度。

```
[yjs@amigaone ~]$ netstat -in
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0       1500 0    254282      0      0 0        201850      0      0      0 BMRU
lo        16436 0     99451      0      0 0         99451      0      0      0 LRU
```

- netstat –nr可以打印路由表

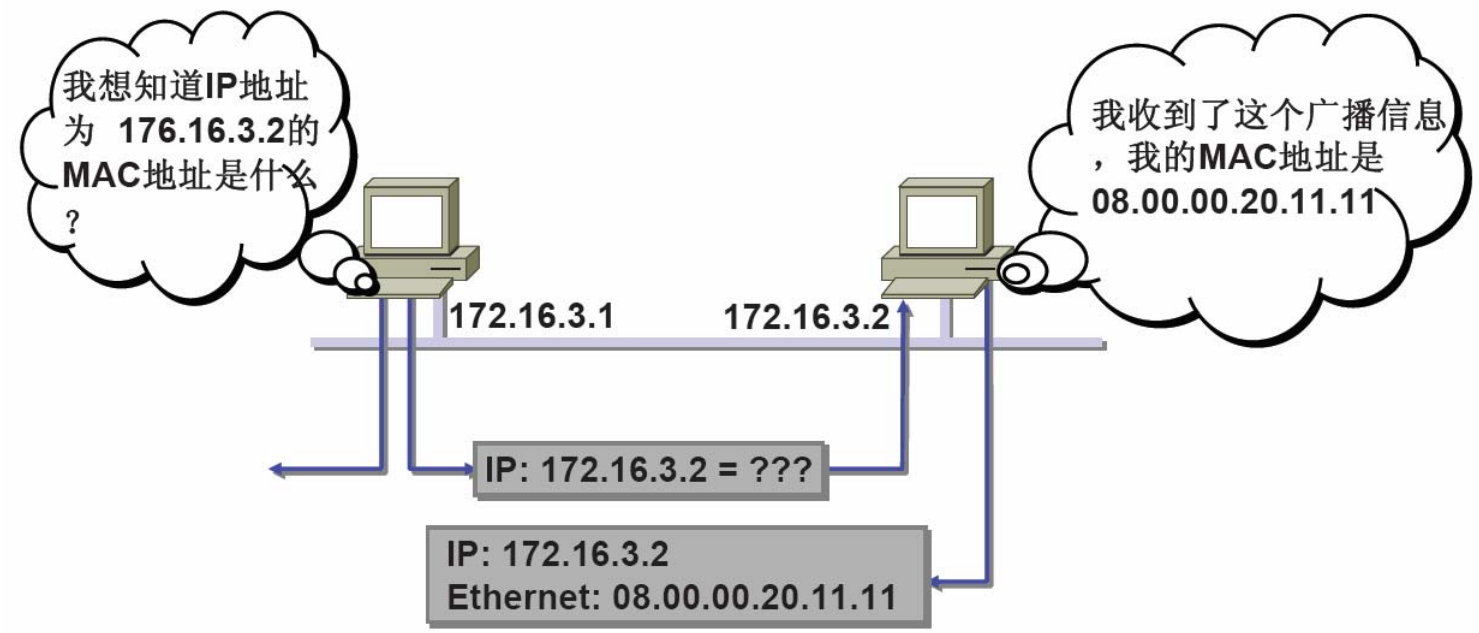
```
[yjs@amigaone ~]$ netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
208.67.219.231  172.16.0.6      255.255.255.255 UGH       0 0          0 eth0
172.16.0.0      0.0.0.0         255.255.255.0   U         0 0          0 eth0
0.0.0.0         172.16.0.1      0.0.0.0         UG        0 0          0 eth0
```

2.7.5 IPv6

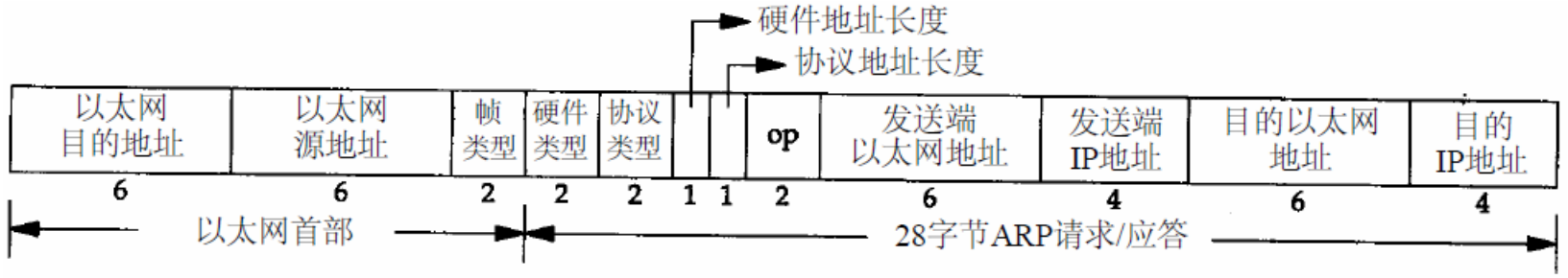
- IPv6具有更大的地址空间。IPv4中规定IP地址长度为32，即有2^{32}-1个地址；而IPv6中IP地址的长度为128，即有2^{128}-1个地址。
- IPv6使用更小的路由表。IPv6的地址分配一开始就遵循聚类（Aggregation）的原则，这使得路由器能在路由表中用一条记录（Entry）表示一片子网，大大减小了路由器中路由表的表目项数量，提高了路由器转发数据包的速度。
- IPv6增加了增强的组播（Multicast）支持以及对流的支持（Flow Control），这使得网络上的多媒体应用有了长足发展的机会，为服务质量（QOS，Quality of Service）控制提供了良好的网络平台。
- IPv6加入了对自动配置（Auto Configuration）的支持。这是对DHCP协议的改进和扩展，使得网络（尤其是局域网）的管理更加方便和快捷。
- IPv6具有更高的安全性。在使用IPv6网络中用户可以对网络层的数据进行加密并对IP报文进行校验，极大的增强了网络的安全性。

2.8 ARP协议

- ARP为IP地址到对应的硬件地址之间提供动态映射，这个过程是自动完成的，一般应用程序用户或系统管理员不必关心



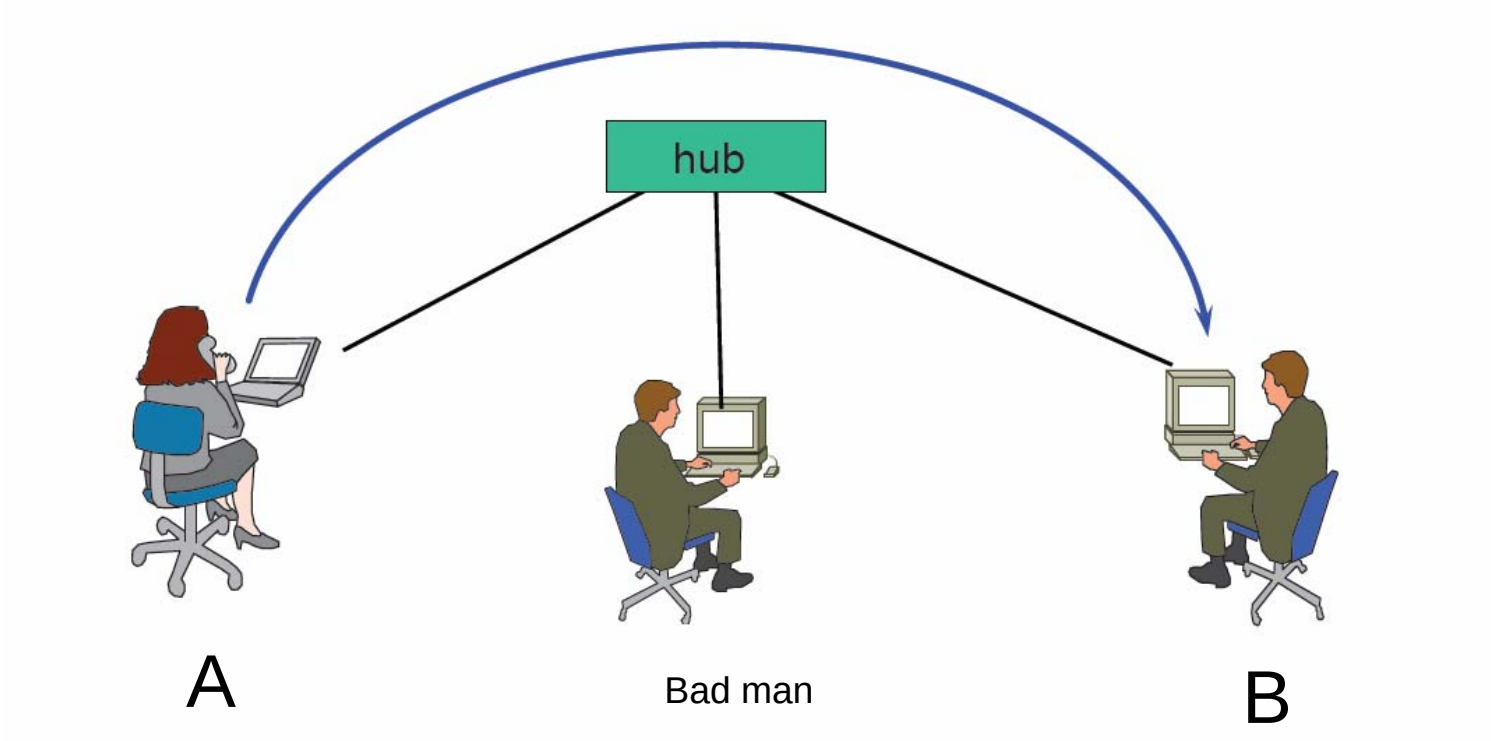
2.8.1 ARP分组格式



- 以太网报头中的前两个字段是以太网的源地址和目的地址。目的地址为全1的特殊地址是广播地址。电缆上的所有以太网接口都要接收广播的数据帧。
- 两个字节长的以太网帧类型表示后面数据的类型，对于ARP请求或应答来说，该字段的值为0x0806。
- hardware(硬件)和protocol(协议)用来描述ARP分组中的各个字段。例如，一个ARP请求分组询问协议地址（这里是IP地址）对应的硬件地址（这里是以太网地址）。
- 硬件类型字段表示硬件地址的类型。它的值为1即表示以太网地址。
- 协议类型字段表示要映射的协议地址类型。它的值为0x0800即表示IP地址。它的值与包含IP数据报的以太网数据帧中的类型字段的值相同，这是有意设计的
- 接下来的两个1字节的字段，硬件地址长度和协议地址长度分别指出硬件地址和协议地址的长度，以字节为单位。对于以太网上IP地址的ARP请求或应答来说，它们的值分别为6和4。
- 操作字段指出四种操作类型，它们是ARP请求（值为1）、ARP应答（值为2）、RARP请求（值为3）和RARP应答（值为4），这个字段必需的，因为ARP请求和ARP应答的帧类型字段值是相同的。
- 接下来的四个字段是发送端的硬件地址、发送端的协议地址（IP地址）、目的端的硬件地址和目的端的协议地址。注意，这里有一些重复信息：在以太网的数据帧报头中和ARP请求数据帧中都有发送端的硬件地址。

2.8.2 ARP协议的安全问题

- 如下图所示的环境中，Bad man可以冒充A或者B发送ARP应答给对方，进而实现中间人攻击

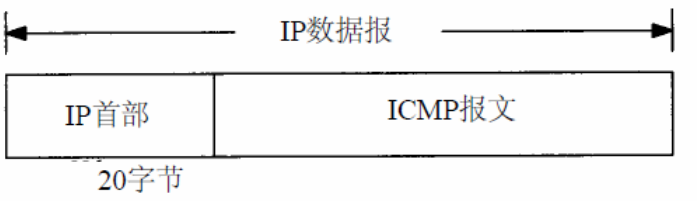


ARP缓存机制

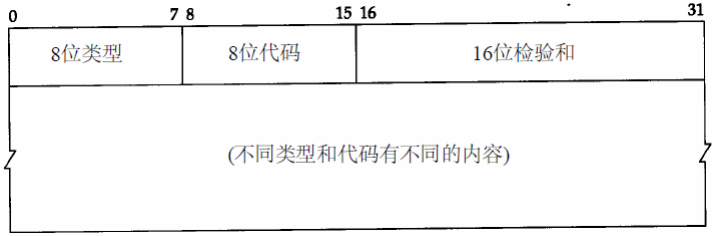
- 链路层会维护ARP缓存，通过如下的命令行可以查询：
  - $ arp -a
- 用户空间可以通过arp命令来操作ARP缓存，
  - $ arp -a
  - # arp -d 删除
  - # arp -s 设定静态的ARP
- 缓存是有超时机制的，当超时时间到时，则清掉超时的缓存条目
- 缓存自动更新机制，当上层(IP层)有数据包要发送时，会查询ARP缓存
  - 如果在缓存中找到对应的条目，并且该条目未超时，则直接使用MAC地址
  - 如果在缓存中未找到对应的条目，则链路层自动发起一个ARP查询请求，等待对方的ARP应答，得到应答后，更新ARP缓存。

2.9 ICMP

- ICMP是IP层的一个组成部分，它传递差错报文以及其他需要注意的信息
- ICMP报文通常被IP层或更高层协议（TCP或UDP）使用，一些ICMP报文把差错报文返回给用户进程。
- ICMP报文是在IP数据报内部被传输的
- ICMP的正式规范参见RFC792[Posterl1981b]
- ICMP报文的格式如下图所示，所有报文的前4个字节都是一样的，但是剩下的其他字节则互不相同
- 类型字段可以有15个不同的值，以描述特定类型的ICMP报文，某些ICMP报文还使用代码字段的值来进一步描述不同的条件。
- 检验和字段覆盖整个ICMP报文，使用的算法与IP首部检验和算法相同，ICMP的检验和是必需的



ICMP封装在IP数据报内部



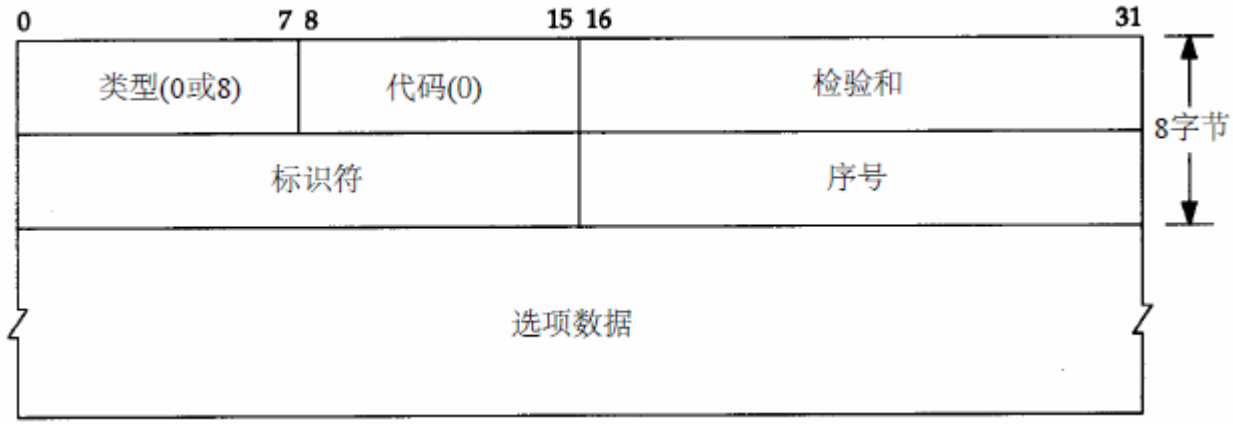
ICMP报文

2.9.1 ICMP报文类型

- ICMP差错报文
  - 目的不可达报文（类型3）
  - 超时报文（类型11）
  - 参数出错报文（类型12）
- ICMP控制报文
  - 报源抑制报文（类型4）
  - 重定向（类型5）
- ICMP请求/应答报文
  - 回送请求和响应报文（类型0和8）
  - 时间戳请求和响应报文（类型13和14）
  - 地址掩码请求和响应报文（类型17和18）
- 其他

类型	代码	描述	查询	差错
0	0	回显应答(Ping应答，第7章)	•	
3		目的不可达:		
	0	网络不可达（9.3节）		•
	1	主机不可达（9.3节）		•
	2	协议不可达		•
	3	端口不可达（6.5节）		•
	4	需要进行分片但设置了不分片比特（11.6节）		•
	5	源站选路失败（8.5节）		•
	6	目的网络不认识		•
	7	目的主机不认识		•
	8	源主机被隔离（作废不用）		•
	9	目的网络被强制禁止		•
	10	目的主机被强制禁止		•
	11	由于服务类型TOS，网络不可达（9.3节）		•
	12	由于服务类型TOS，主机不可达（9.3节）		•
	13	由于过滤，通信被强制禁止		•
	14	主机越权		•
	15	优先权中止生效		•
4	0	源端被关闭（基本流控制，11.11节）		•
5		重定向（9.5节）:		•
	0	对网络重定向		•
	1	对主机重定向		•
	2	对服务类型和网络重定向		•
	3	对服务类型和主机重定向		•
8	0	请求回显（Ping请求，第7章）	•	
9	0	路由器通告（9.6节）	•	
10	0	路由器请求（9.6节）	•	
11		超时:		
	0	传输期间生存时间为0（Traceroute，第8章）		•
	1	在数据报组装期间生存时间为0（11.5节）		•
12		参数问题:		
	0	坏的IP首部（包括各种差错）		•
	1	缺少必需的选项		•
13	0	时间戳请求（6.4节）	•	
14	0	时间戳应答（6.4节）	•	
15	0	信息请求（作废不用）	•	
16	0	信息应答（作废不用）	•	
17	0	地址掩码请求（6.3节）	•	
18	0	地址掩码应答（6.3节）	•	

2.9.2 ICMP echo/reply报文



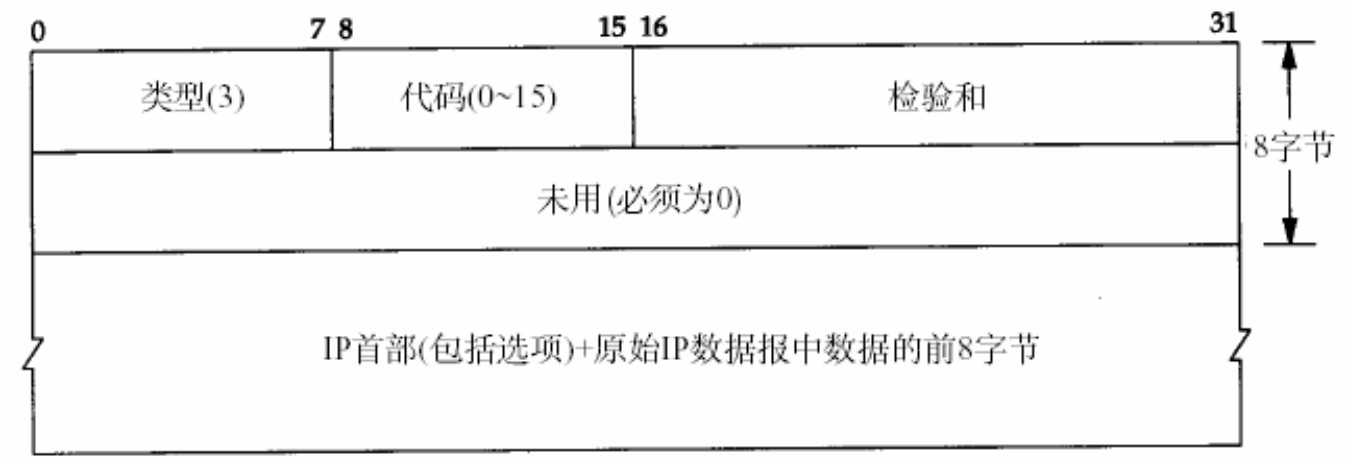
- ICMP回显请求和回显应答报文
  - 类型：0表示Echo Reply，8表示Echo
  - 代码：0
  - 标识符：标识一个会话，例如，用进程ID
  - 序号：例如每个请求增1
  - 选项数据：回显

2.9.2.1 example

- 通过tcpdump捕获ICMP包
  - 以下是ping www.google.com时tcpdump捕获的数据包

```
[root@boss ~]# tcpdump -nnvvvXS icmp -i wlan0 -c 2 -s 1024
tcpdump: listening on wlan0, link-type EN10MB (Ethernet), capture size 1024 bytes
22:20:29.605812 IP (tos 0x0, ttl  64, id 0, offset 0, flags [DF], proto: ICMP (1), length: 84)
 172.16.0.6 > 74.125.128.104: ICMP echo request, id 4923, seq 1, length 64
        0x0000:  4500 0054 0000 4000 4001 c3ad ac10 0006  E..T..@.@.......
        0x0010:  4a7d 8068 0800 84db 133b 0001 ad57 5f4f  J}.h.....;...W_O
        0x0020:  5f3e 0900 0809 0a0b 0c0d 0e0f 1011 1213  _>..............
        0x0030:  1415 1617 1819 1a1b 1c1d 1e1f 2021 2223  .............!"#
        0x0040:  2425 2627 2829 2a2b 2c2d 2e2f 3031 3233  $%&'()*+,-./0123
        0x0050:  3435 3637                                4567
22:20:29.697149 IP (tos 0x0, ttl  46, id 45728, offset 0, flags [none], proto: ICMP (1), lengt
h: 84) 74.125.128.104 > 172.16.0.6: ICMP echo reply, id 4923, seq 1, length 64
        0x0000:  4500 0054 b2a0 0000 2e01 630d 4a7d 8068  E..T......c.J}.h
        0x0010:  ac10 0006 0000 8cdb 133b 0001 ad57 5f4f  .........;...W_O
        0x0020:  5f3e 0900 0809 0a0b 0c0d 0e0f 1011 1213  _>..............
        0x0030:  1415 1617 1819 1a1b 1c1d 1e1f 2021 2223  .............!"#
        0x0040:  2425 2627 2829 2a2b 2c2d 2e2f 3031 3233  $%&'()*+,-./0123
        0x0050:  3435 3637                                4567
2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

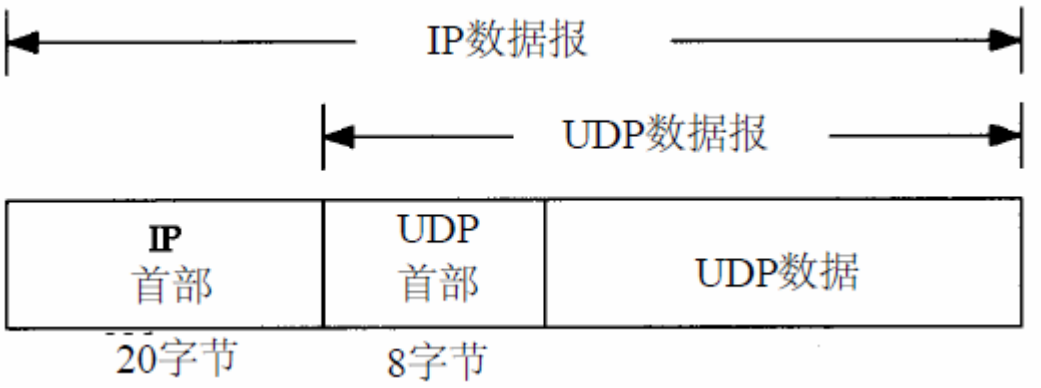
2.9.3 ICMP目的不可达报文



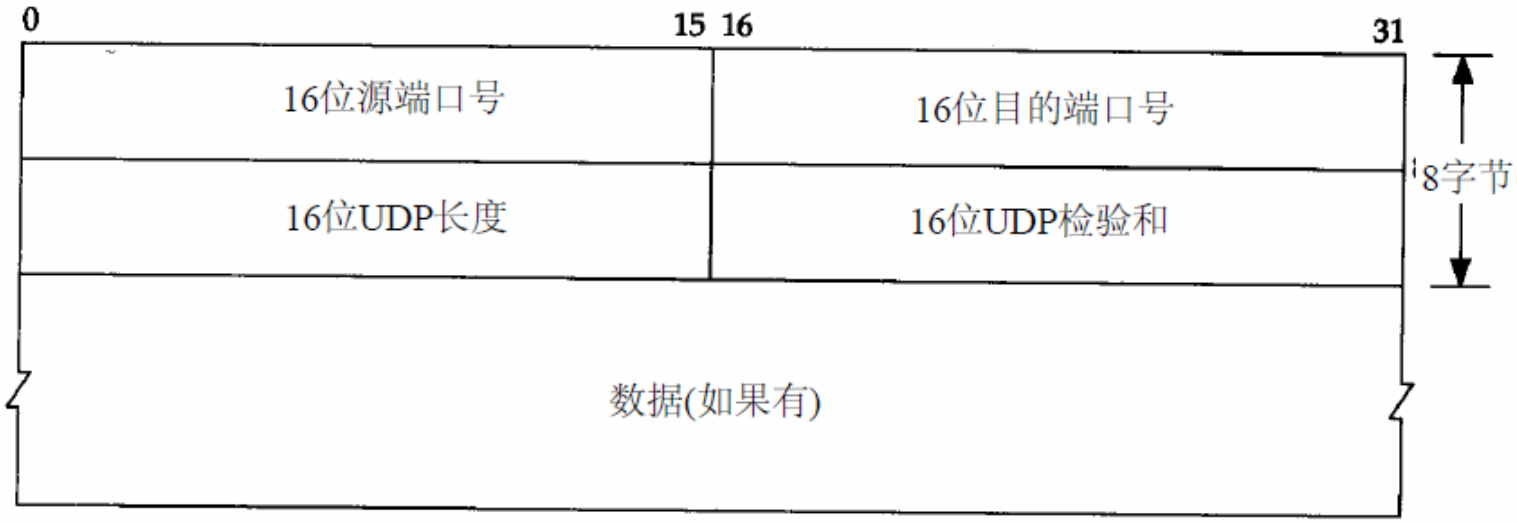
- 类型：3
- 代码：
  - 0表示网络不可达，
  - 1表示主机不可达；
  - 2表示协议不可达；
  - 3表示端口不可达
  - ……
- 出错的IP包的IP首部+原始IP数据包中前8个字节

2.10 UDP协议

- UDP是一个简单的面向数据报的传输层协议
  - 进程的每个输出操作都正好产生一个UDP数据报，并组装成一份待发送的IP数据报。
  - 与面向流字符的协议不同，如TCP（采用面向流的协议，其应用程序产生的全体数据与真正发送的单个IP数据报可能没有什么联系）
- UDP数据报封装成一份IP数据报的格式如下图所示
- RFC768[Postel 1980]是UDP的正式规范。
- UDP不提供可靠性，它把应用程序传给IP层的数据发送出去，但是并不保证它们能到达目的地
- 应用程序必须关心IP数据报的长度
  - 如果它超过网络的MTU，那么就要对IP数据报进行分片。
  - 如果需要，源端到目的端之间的每个网络都要进行分片，并不只是发送端主机连接第一个网络才这样做



2.10.1 UDP头



- 端口号表示发送进程和接收进程
- UDP长度字段指的是UDP首部和UDP数据的字节长度。
  - 该字段的最小值为8字节（可以发送0字节的UDP数据报）
  - 这个UDP长度是有冗余的
  - IP数据报长度指的是数据报全长，因此UDP数据报长度是全长减去IP首部的长度

2.11 TCP协议

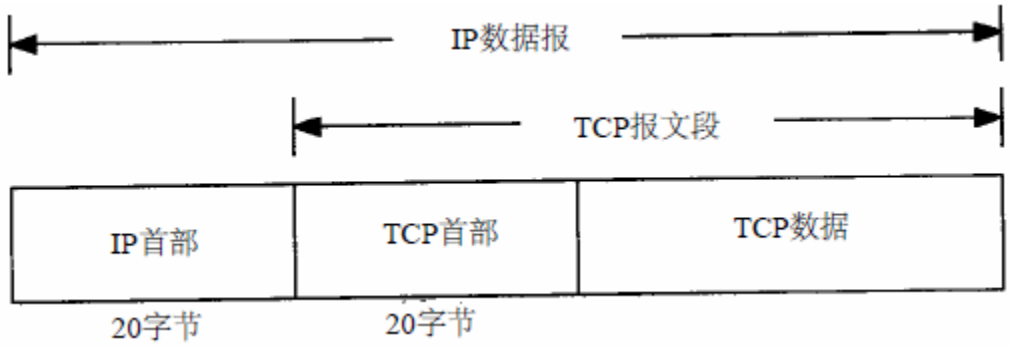
- TCP提供一种面向连接的、可靠的字节流服务
- 面向连接意味着两个使用TCP的应用（通常是一个客户和一个服务器）在彼此交换数据之前必须先建立一个TCP连接。
  - 这一过程与打电话很相似，先拨号振铃，等待对方摘机说“喂”，然后才说明是谁
- 需要有连接建立的过程
  - 通常靠几次握手来实现
  - 通常也需要协商一些通讯的参数
- 需要有连接维持的过程
  - 有可能通过数据收发来维持连接
  - 也有可能有专门的维持连接的指令来维持连接
- 需要有连接拆除的过程
  - 参与通讯的双方要释放资源，恢复状态
- 有可能需要提供连接恢复功能
- 在连接建立之后，可以进入数据收发阶段

TCP协议的可靠性

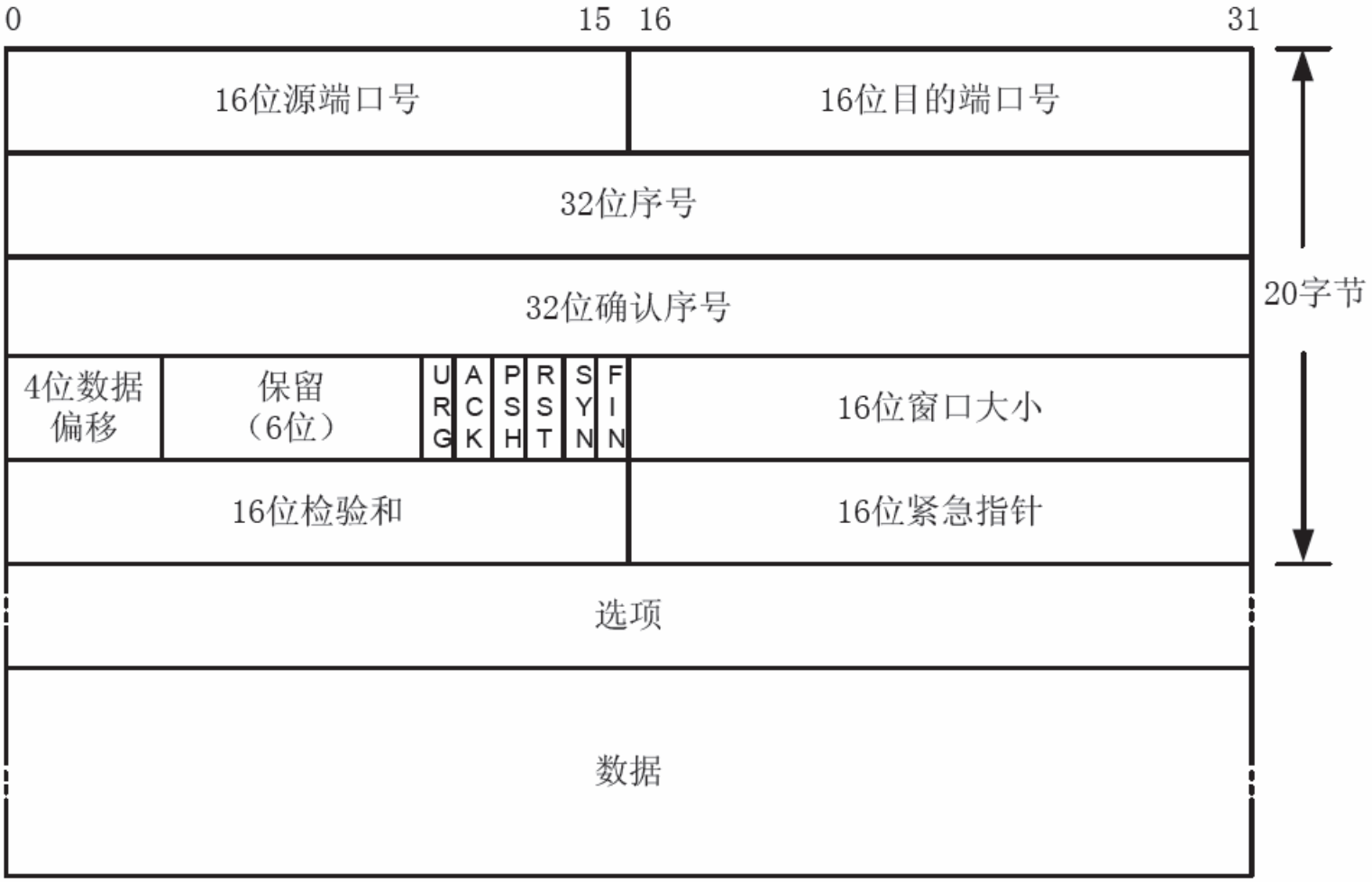
- 应用数据被分割成TCP认为最适合发送的数据块
  - 由TCP传递给IP的信息单位称为报文段或段（segment）
- 当TCP发出一个段后，它启动一个定时器，等待目的端确认收到这个报文段，如果不能及时收到一个确认，将重发这个报文段。
- 当TCP收到发自TCP连接另一端的数据，它将发送一个确认
  - 这个确认不是立即发送，通常将推迟几分之一秒
- TCP将保持它首部和数据的检验和
  - 这是一个端到端的检验和，目的是检测数据在传输过程中的任何变化。
  - 如果收到段的检验和有差错，TCP将丢弃这个报文段和不确认收到此报文段（希望发端超时并重发）。
- 如果必要，TCP将对收到的数据进行重新排序，将收到的数据以正确的顺序交给应用层。
  - TCP报文段作为IP数据报来传输，而IP数据报的到达可能会失序，因此TCP报文段的到达也可能会失序
- TCP的接收端必须丢弃重复的数据
  - 既然IP数据报会发生重复
- TCP还能提供流量控制
  - TCP连接的每一方都有固定大小的缓冲空间，TCP的接收端只允许另一端发送接收端缓冲区所能接纳的数据，这将防止较快主机致使较慢主机的缓冲区溢出。

2.11.1 TCP数据封装

- TCP数据被封装在一个IP数据报中



2.11.2 TCP头

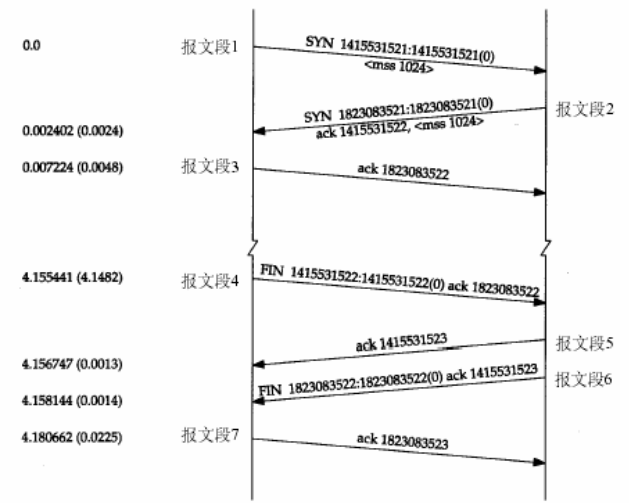

0
15
16
31
16位源端口号
16位目的端口号
32位序号
32位确认序号
4位数据偏移
保留（6位）
URG
ACK
PSH
RST
SYN
FIN
16位窗口大小
16位检验和
16位紧急指针
选项
数据
20字节

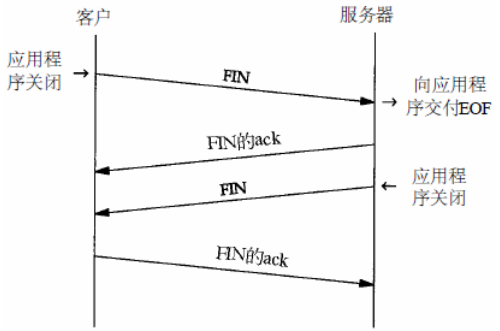
2.11.2 TCP头

- 源端口号和目的端口号：源和目的主机的IP地址加上端口号构成一个TCP连接
- 序号和确认号：序号为该TCP数据包的第一个数据字在所发送的数据流中的偏移量；确认号为希望接收的下一个数据字的序号；
- 首部长度，以4个字节为单位，通常为20个字节
- 6个标志位：
  - URG：如果使用了紧急指针，URG置1，紧急指针为当前序号到紧急数据位置的偏移量
  - ACK：为1表示确认号有效，为0表示该TCP数据包不包含确认信息
  - PSH：表示是带有PUSH标志的数据，接收到数据后不必等缓冲区满再发送
  - RST：用于连接复位，也可用于拒绝非法的数据或拒绝连接请求
  - SYN：用于建立连接，连接请求时SYN＝1，ACK=0；响应连接请求时SYN=1，ACK=1
  - FIN：用于释放连接，表示发送方已经没有供发送的数据
- 窗口大小：表示在确认字节后还可以发送字节数，用于流量控制
- 校验和：覆盖了整个数据包，包括对数据包的首部和数据
- 选项：常见的选项是MSS(Maximum Segment Size)

2.11.3 TCP连接建立与终止

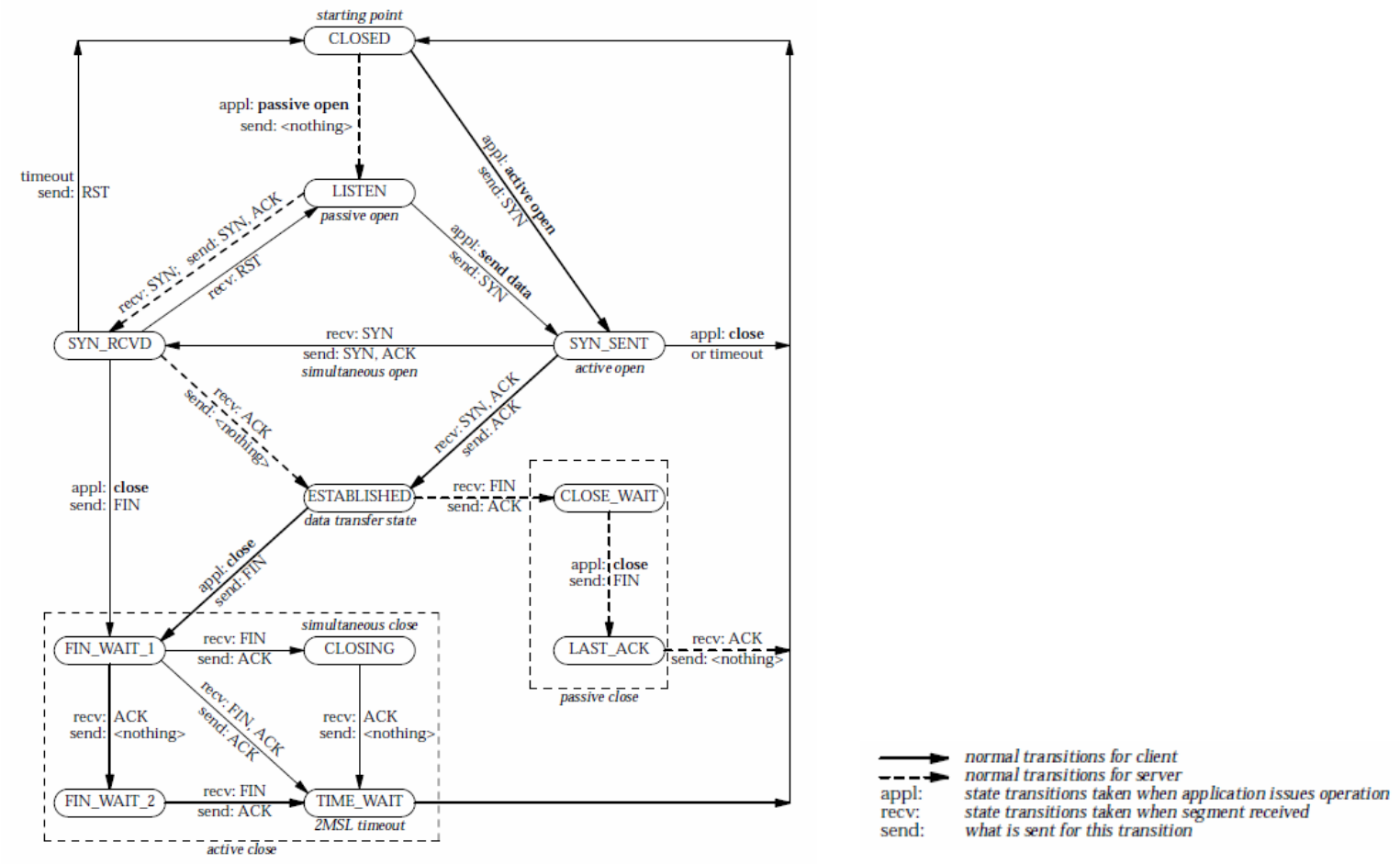
- TCP通过三次握手（three-way handshake）来建立连接
  - 请求端（通常称为客户端）发送一个SYN段指明客户打算连接的服务器的端口，以及初始序号（ISN，在这个例子中为1415531521）。这个SYN段为报文段1。
  - 服务器发回包含服务器的初始序号的SYN报文段（报文段2）作为应答。同时，将确认序号设置为客户的ISN加1以对客户的SYN报文段进行确认。一个SYN将占用一个序号。
  - 客户必须将确认序号设置为服务器的ISN加1以对服务器的SYN报文段进行确认（报文段3）。



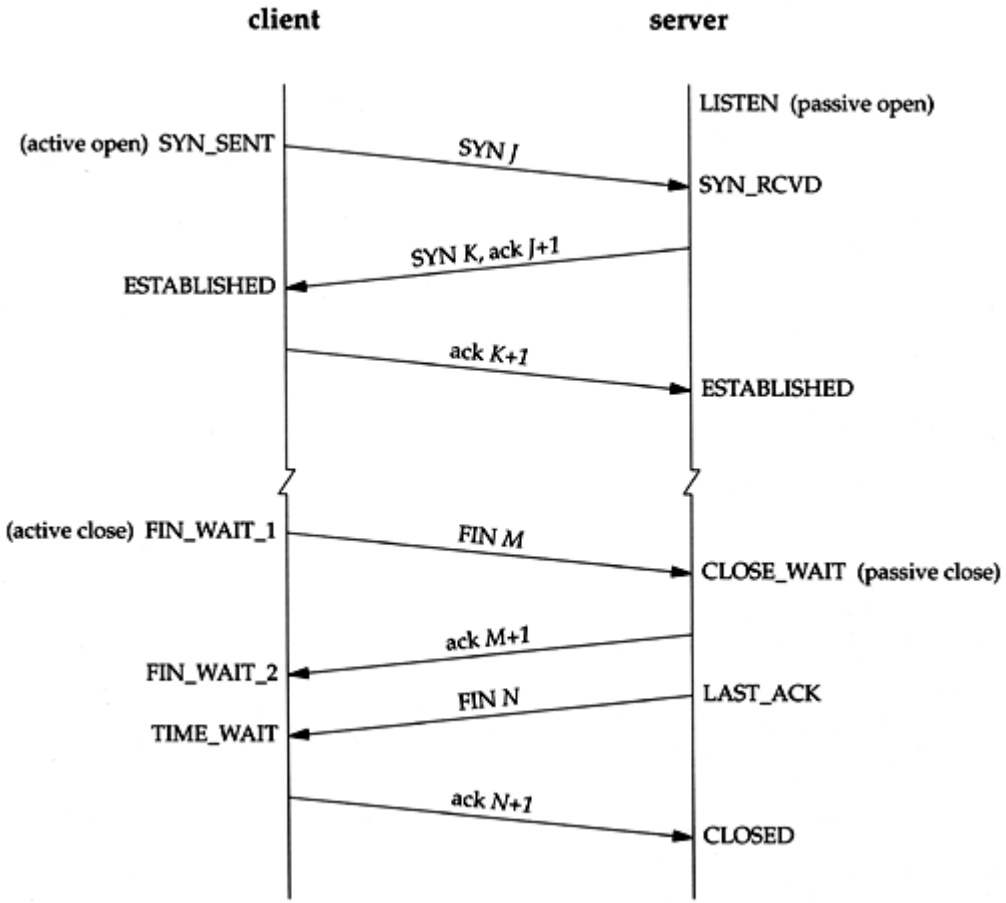


- 终止连接需要四次握手
  - 解决TCP的半关闭（half-close）的问题

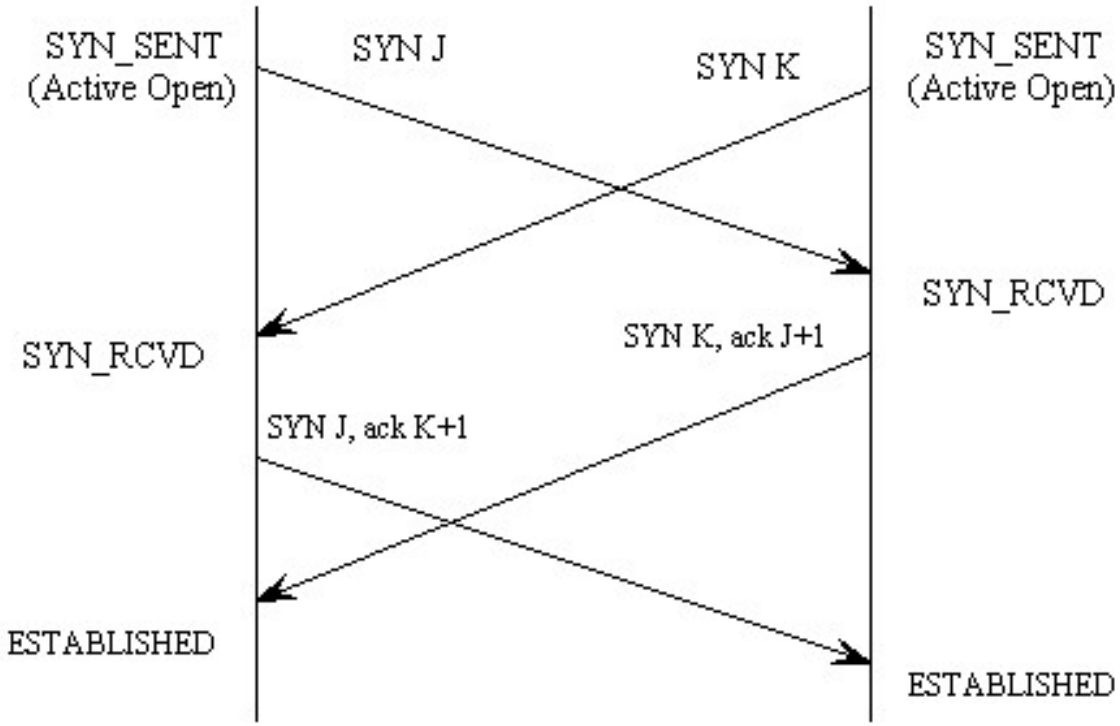
2.11.4 TCP状态变迁图



2.11.4 正常建立连接和断开连接过程

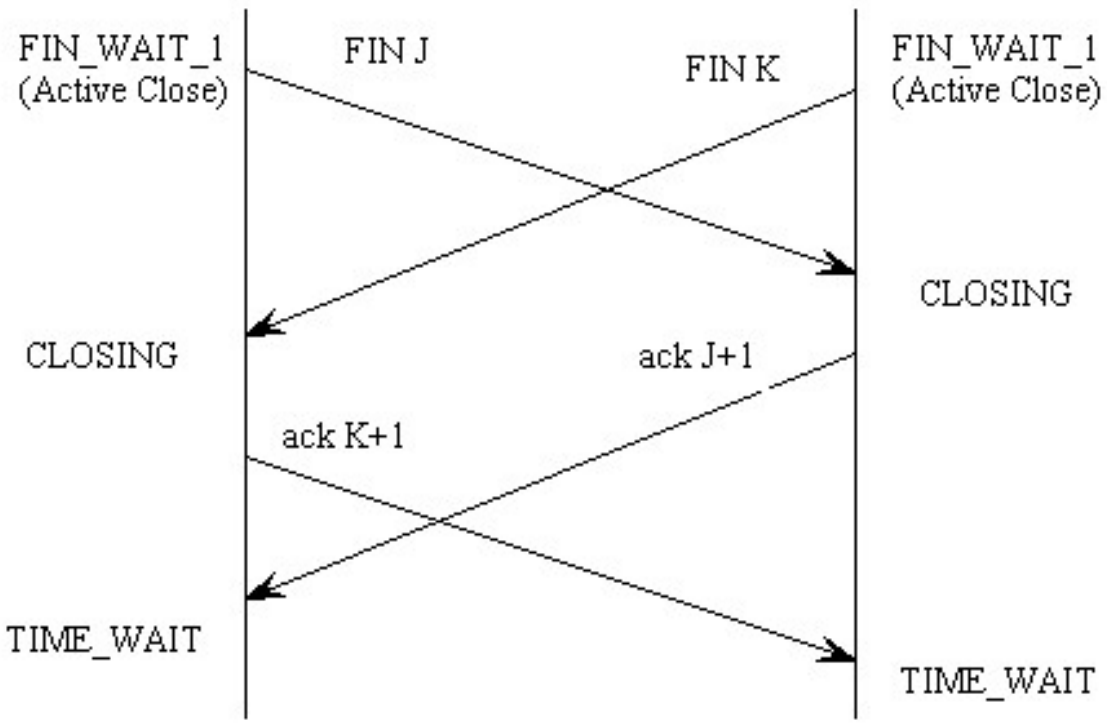


2.11.4 同时打开的交互过程



state transitions in simultaneous open

2.11.4 同时关闭的交互过程



state transitions in simultaneous close

2.12 TCP vs UDP

- 共同点：
  - 同为传输层协议
- 不同点：
  - TCP：有连接，可靠，传输的是字节流
  - UDP：无连接，不可靠，传输的是数据报
- 适用性？
  - MSN/QQ使用的传输层协议？

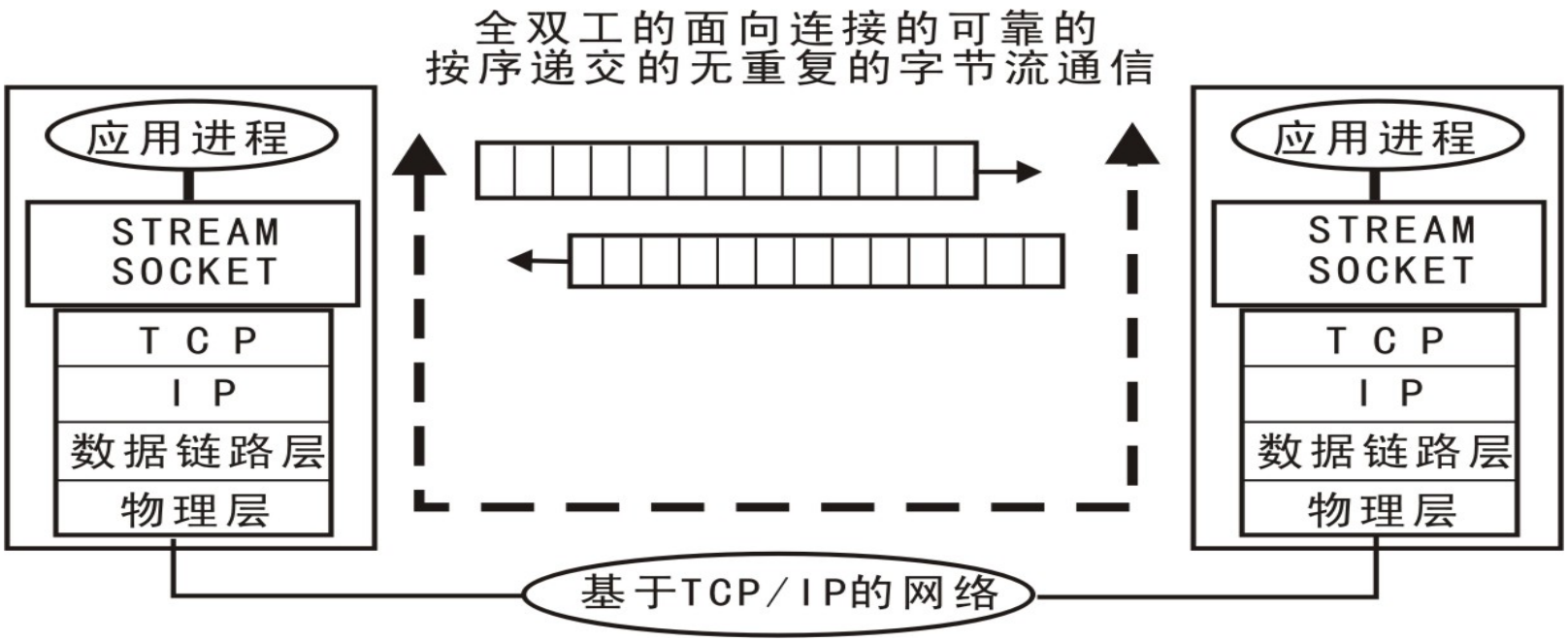
字节流和数据报的特点

- 字节流特点
  - 有明确的方向，数据会发送给建立连接的对方，耦合度高
  - 无记录边界
  - 字节流数据无意义(无格式)，处理时需要把字节流切分，变成记录块，这样才有意义(有格式)
  - 管道、FIFO 、TCP传输的是字节流
- 数据报特点
  - 没有明确的方向，耦合度低
  - 有记录边界
  - 只要记录块完整，数据报就有意义（有格式）
  - 消息队列可以认为传输的是数据报，UDP传输的是数据报

2.12.1 TCP协议特点

- TCP（即传输控制协议）：是一种面向连接的传输层协议，它能提供高可靠性通信(即数据无误、数据无丢失、数据无失序、数据无重复到达的通信。)
- 适用情况：
  - 适合于对传输质量要求较高，以及传输大量数据的通信。
  - 在需要可靠数据传输的场合，通常使用TCP协议
- MSN/QQ等即时通讯软件的用户登录账户管理相关的功能通常采用TCP协议

2.12.1 TCP传输


全双工的面向连接的可靠的
按序递交的无重复的字节流通信
应用进程
STREAM SOCKET
T C P
I P
数据链路层
物理层
应用进程
STREAM SOCKET
T C P
I P
数据链路层
物理层
基于TCP/IP的网络

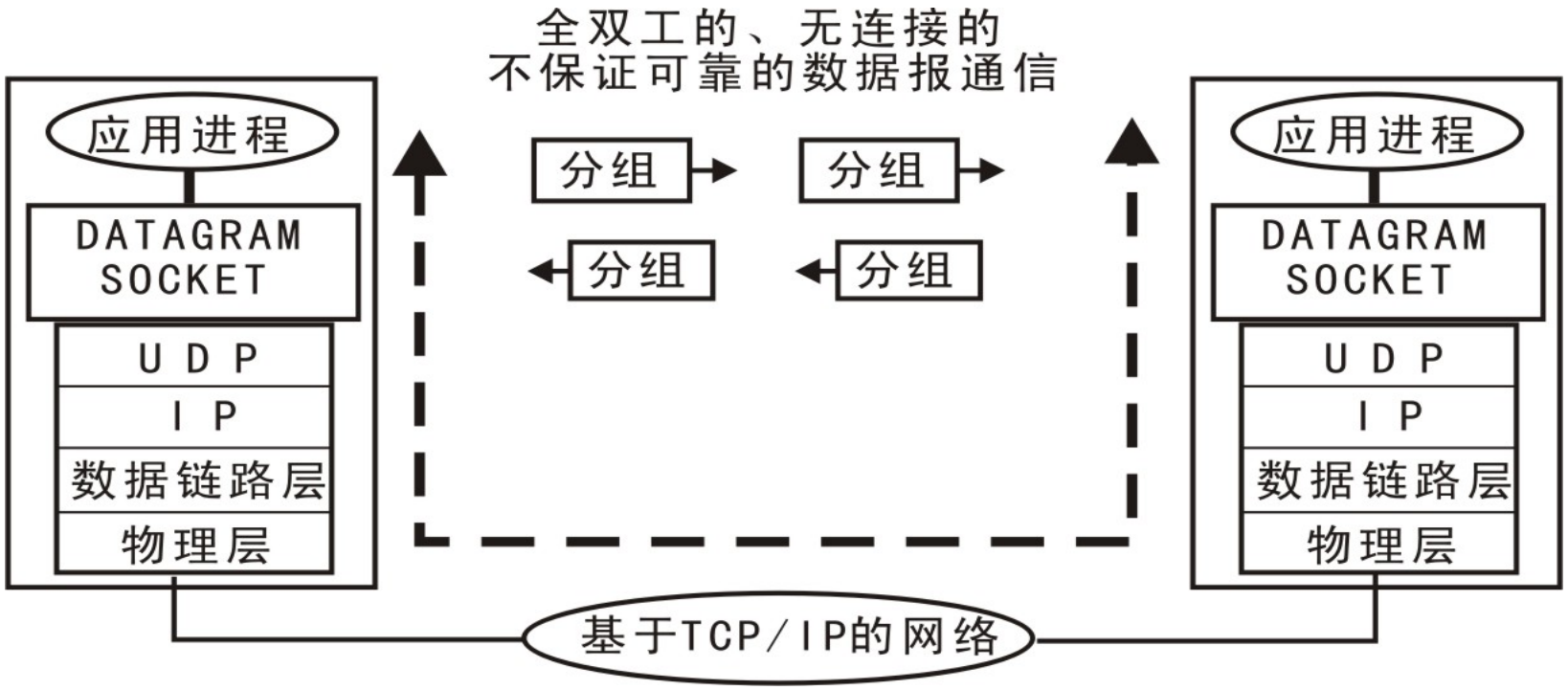
使用TCP通讯的条件

- 双方网络通
  - ping
  - traceroute
- 防火墙允许连接
  - 在开发阶段，可以考虑关闭防火墙
    - # service iptables stop
  - 部署时，可以配置防火墙的规则
- 双方的协议栈相同
  - 同为IPv4或者IPv6
- Socket类型相同
  - TCP的socket类型为：SOCK_STREAM

2.12.2 UDP协议特点

- UDP（用户数据报协议），是不可靠的无连接的协议，在数据发送前，因为不需要进行连接，所以可以进行高效率的数据传输。
  - 与TCP协议相比，具有传输速度高的优点。
- 适用情况：
  - 发送小尺寸数据（如：对DNS服务器进行IP地址查询时，若进行连接之后再进行数据传输就会降低效率，这时就使用UDP。）
  - 在接收到数据，给出应答较困难的网络中使用UDP。（如：无线网络）
  - 适合于广播/组播式通信中。
  - MSN/QQ/Skype等即时通讯软件的点对点文本通讯以及音视频通讯通常采用UDP协议
  - 流媒体、VOD、VoIP、IPTV等网络多媒体服务中通常采用UDP方式进行实时数据传输

2.12.2 UDP传输


全双工的、无连接的
不保证可靠的数据报通信
应用进程
DATAGRAM SOCKET
UDP
IP
数据链路层
物理层
分组
分组
分组
分组
应用进程
DATAGRAM SOCKET
UDP
IP
数据链路层
物理层
基于TCP/IP的网络

使用UDP通讯的条件

- 双方网络通
  - ping
  - traceroute
- 防火墙允许连接
  - 在开发阶段，可以考虑关闭防火墙
    - # service iptables stop
  - 部署时，可以配置防火墙的规则
- 双方的协议栈相同
  - 同为IPv4或者IPv6
- Socket类型相同
  - UDP的socket类型为：SOCK_DGRAM

2.12.2 UDP协议的高级应用

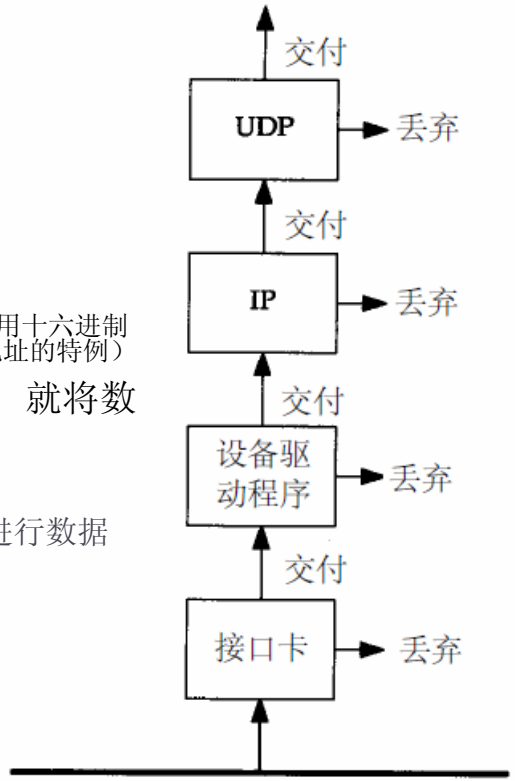
- 广播及多播
- TCP over UDP
  - 具有UDP协议速度快实时性强的特点
  - 具有TCP协议可靠的特点
- P2P技术

2.13 广播和多播

- 广播和多播仅应用于UDP，它们对需将报文同时传往多个接收者的应用来说十分重要
  - TCP是一个面向连接的协议，它意味着分别运行于两主机（由IP地址确定）内的两进程（由端口号确定）间存在一条连接。
- 每个以太网帧包含源主机和目的主机的以太网地址（即MAC地址，48bits）
  - 通常每个以太网帧仅发往单个目的主机，目的地址指明单个接收接口，因而称为单播(unicast)
  - 一个主机要向网上(确切的说应该是子网内)的所有其他主机发送帧，这就是广播(broadcast)
  - 多播(multicast) 处于单播和广播之间：帧仅传送给属于多播组的多个主机(1个或者多个)

2.13.1 数据包过滤的过程

- 网卡查看由信道传送过来的帧，确定是否接收该帧，若接收后就将它传往设备驱动程序
  - 通常网卡仅接收那些目的地址为网卡物理地址或广播地址的帧
  - 如果接口被设置为混杂模式(promiscuous mode)，此时能接收每个帧的一个复制
    - tcpdump等sniffer使用混杂模式
    - 在使用HUB组网的环境下，会接收到帧的复制
    - 在使用交换机组网的环境下，不会接收到帧的复制
- 设备驱动程序做相应的检查和处理，将数据帧传送给下一层
  - 如果帧检验和错，丢弃该帧
  - 随后设备驱动程序将进行另外的帧过滤
    - 首先，帧类型中必须指定要使用的协议（ IP、ARP等等）
      - 当帧类型指定为IP数据报时，就传往IP层
      - 当帧类型指定为ARP/RARP数据报时，交给相应的内核代码进行处理
    - 其次，进行多播过滤来检测该主机是否属于多播地址说明的多播组
      - 对于以太网，当地址中最高字节的最低位设置为1时表示该地址是一个多播地址，用十六进制可表示为01:00:00:00:00:00（以太网广播地址ff:ff:ff:ff:ff:ff可看作是以太网多播地址的特例）
- IP根据IP地址中的源地址和目的地址进行更多的过滤检测，如果正常，就将数据报传送给下一层（如TCP或UDP）
- TCP/UDP层根据端口号将数据报传送给进程
  - 每次UDP收到由IP传送来的数据报，就根据目的端口号，有时还有源端口号进行数据报过滤。
    - 如果当前没有进程使用该目的端口号，就丢弃该数据报并产生一个ICMP不可达报文
    - 如果U D P数据报存在检验和错，将被丢弃。
  - TCP根据它的端口号作相似的过滤



2.13.2 广播 vs 多播

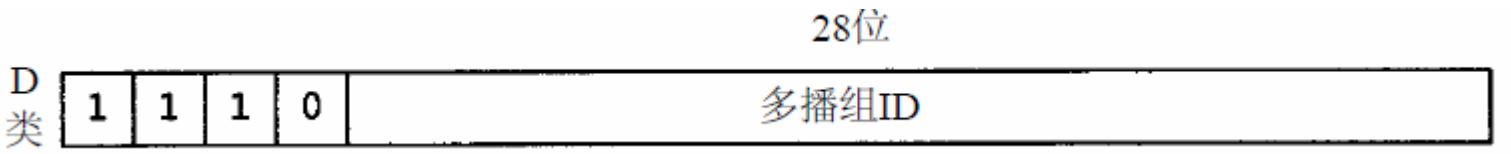
- 使用广播的问题在于它增加了对广播数据不感兴趣主机的处理负荷
  - 拿一个使用UDP广播应用作为例子，如果网内有50个主机，但仅有20个参与该应用，每次这20个主机中的一个发送UDP广播数据时，其余30个主机不得不处理这些广播数据报，一直到UDP层，收到的UDP广播数据报才会被丢弃，这30个主机丢弃UDP广播数据报是因为这些主机没有使用这个目的端口。
- 多播的出现减少了对应用不感兴趣主机的处理负荷。
  - 使用多播，主机可加入一个或多个多播组，这样，网卡将获悉该主机属于哪个多播组，然后仅接收主机所在多播组的那些多播帧。
  - 主机可以离开多播组

2.13.3 广播的种类

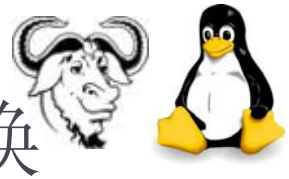
- 受限的广播
  - 受限的广播地址是255.255.255.255，该地址用于主机配置过程中IP数据报的目的地址，此时，主机可能还不知道它所在网络的网络掩码，甚至连它的IP地址也不知道
    - DHCP即使用该地址
  - 在任何情况下，路由器都不转发目的地址为受限的广播地址的数据报，这样的数据报仅出现在本地网络中
- 指向网络的广播
  - 指向网络的广播地址是主机号为全1的地址
    - A类网络广播地址为netid.255.255.255，其中netid为A类网络的网络号
  - 一个路由器必须转发指向网络的广播，但它也必须有一个不进行转发的选择。
- 指向子网的广播
  - 指向子网的广播地址为主机号为全1且有特定子网号的地址。
  - 作为子网直接广播地址的IP地址需要了解子网的掩码。例如，如果路由器收到发往128.1.2.255的数据报，当B类网络128.1的子网掩码为255.255.255.0时，该地址就是指向子网的广播地址；但如果该子网的掩码为255.255.254.0，该地址就不是指向子网的广播地址。
- 指向所有子网的广播
  - 指向所有子网的广播也需要了解目的网络的子网掩码，以便与指向网络的广播地址区分开。指向所有子网的广播地址的子网号及主机号为全1。例如，如果目的子网掩码为255.255.255.0，那么IP地址128.1.255.255是一个指向所有子网的广播地址。然而，如果网络没有划分子网，这就是一个指向网络的广播。
- 在采用CIDR地址分配的情况下，只有两种广播：受限广播和子网广播

2.13.4 多播组地址

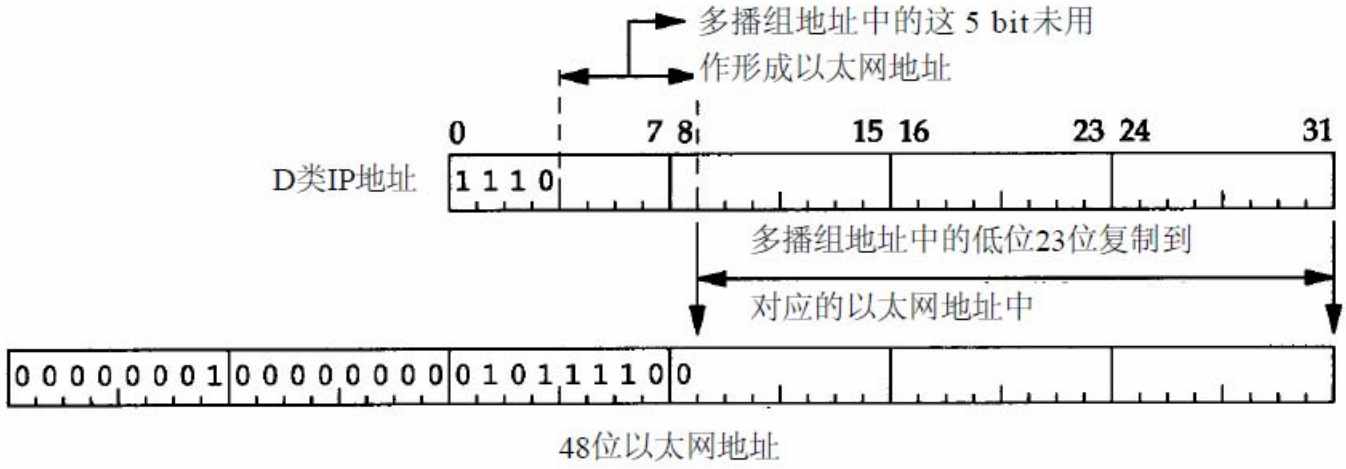
- 多播组地址(D类地址)包括为1110的最高4bit和多播组号。它们通常可表示为点分十进制数，范围从224.0.0.0到239.255.255.255。
- 能够接收发往一个特定多播组地址数据的主机集合称为主机组(host group)
  - 一个主机组可跨越多个网络
  - 主机组中成员可随时加入或离开主机组
  - 主机组中对主机的数量没有限制
  - 不属于某一主机组的主机可以向该组发送信息
- 一些多播组地址被IANA确定为知名地址。它们也被当作永久主机组，这和TCP及UDP中的周知端口相似。同样，这些知名多播地址在RFC最新分配数字中列出。注意这些多播地址所代表的组是永久组，而它们的组成员却不是永久的。
  - 224.0.0.1代表“该子网内的所有系统组”，
  - 224.0.0.2代表“该子网内的所有路由器组”。
  - 多播地址224.0.1.1用作网络时间协议NTP
  - 224.0.0.9用作RIP-2
  - 224.0.1.2用作SGI公司的dogfight应用。



2.13.5 多播组地址到以太网地址的转换



- IANA分配与IP多播相对应的以太网地址范围从01:00:5e:00:00:00到01:00:5e:7f:ff:ff
- 这种地址分配将使以太网多播地址中的23bit与IP多播组号对应起来，通过将多播组号中的低位23bit映射到以太网地址中的低位23bit实现，这个过程如图所示
- 多播组号中的最高5bit在映射过程中被忽略，因此每个以太网多播地址对应的多播组是不唯一的，32个不同的多播组号被映射为一个以太网地址
- 既然地址映射是不唯一的，那么设备驱动程序或IP层就必须对数据报进行过滤
  - 因为网卡可能接收到主机不想接收的多播数据帧
  - 如果网卡不提供足够的多播数据帧过滤功能，设备驱动程序就必须接收所有多播数据帧，然后对它们进行过滤。



2.14 DNS

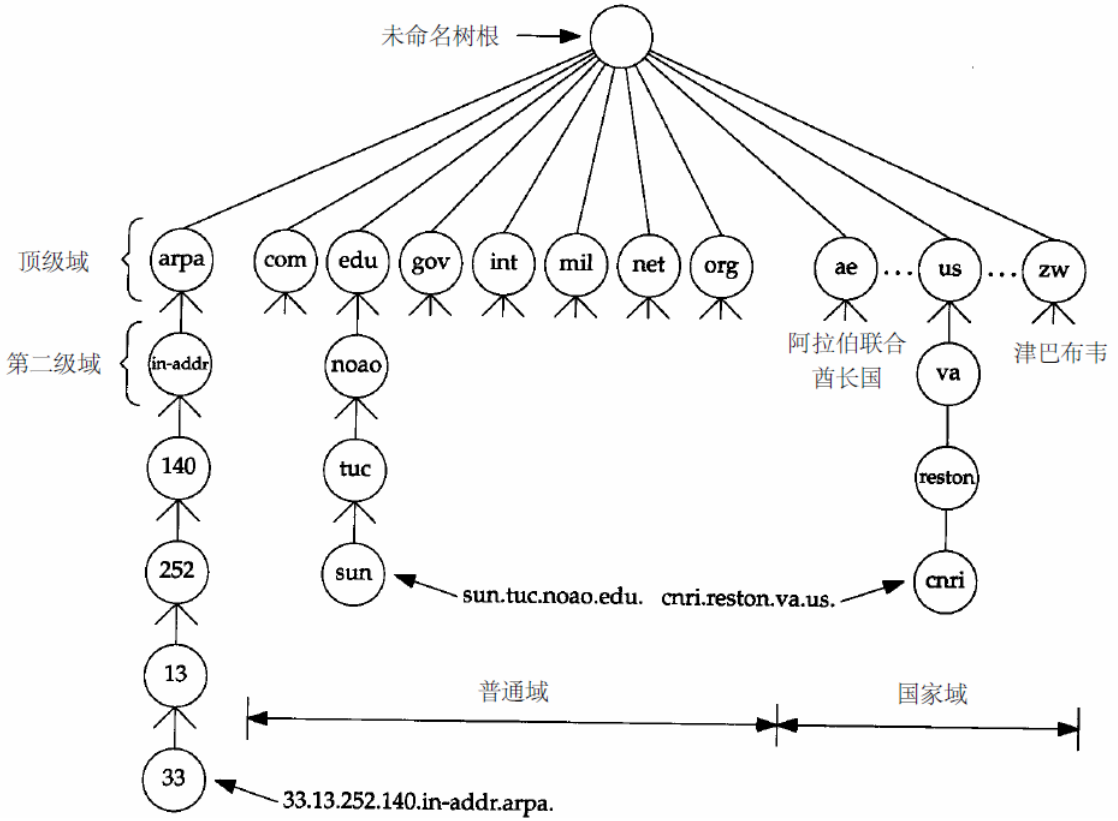
- 域名系统(DNS)是一种用于TCP/IP应用程序的分布式数据库，它提供主机名字和IP地址之间的转换及有关电子邮件的选路信息
  - 所谓的“分布式”是指在Internet上的单个站点不能拥有所有的信息
  - 每个站点（如大学中的系、校园、公司或公司中的部门）保留它自己的信息数据库，并运行一个服务器程序供Internet上的其他系统（客户程序）查询
  - DNS提供了允许服务器和客户程序相互通信的协议
- 从应用的角度上看，对DNS的访问是通过一个地址解析器(resolver)来完成的
  - 在Unix主机中，该解析器主要是通过两个库函数gethostbyname(3)和gethostbyaddr(3)来访问的，它们在编译应用程序时与应用程序连接在一起
    - gethostbyname(3)接收主机名字返回IP地址
    - gethostbyaddr(3)接收IP地址来寻找主机名字
    - getaddrinfo()是比较新的函数
  - 解析器通过一个或多个名字服务器来完成这种相互转换
  - 解析器通常是应用程序的一部分
  - 在一个应用程序请求TCP打开一个连接或使用UDP发送一个数据报之前，心须将一个主机名转换为一个IP地址
  - 操作系统内核中的TCP/IP协议族对于DNS不关心
- RFC1034[Mockapetris 1987a]说明了DNS的概念和功能
- RFC1035[Mockapetris 1987b]详细说明了DNS的规范和实现
- DNS最常用的版本（包括解析器和名字服务器）是BIND—伯克利Internet域名服务器，该服务器称作named。

2.14.1 DNS的历史

- DNS最早于1983年由保罗·莫卡派乔斯（Paul Mockapetris）发明；
  - 原始的技术规范在882号因特网标准草案（RFC 882）中发布。
  - 1987年发布的第1034和1035号草案修正了DNS技术规范，并废除了之前的第882和883号草案。在此之后对因特网标准草案的修改基本上没有涉及到DNS技术规范部分的改动。
- 早期的域名必须以英文句号“.”结尾，当用户访问 www.example.com 的HTTP服务时必须在址栏中输入： http://www.example.com. ，这样DNS才能够进行域名解析
  - 现在DNS服务器已经可以自动补上结尾的点
- 当前，对于域名长度的限制是63个字符，包括www.和.com或者其他的扩展名。
- 域名同时也仅限于ASCII字符的一个子集，这使得很多其他语言无法正确表示他们的名字和单词
  - 基于Punycode码的IDNA系统，可以将Unicode字符串映射为有效的DNS字符集，这已经通过了验证并被一些注册机构作为一种变通的方法所采纳。
  - 近年，包括中国在内的一些非英语国家也在大力的推动非ASCII字符的域名

2.14.2 DNS组织

- DNS的名字空间和Unix的文件系统相似，也具有层次结构
  - 每个结点（右图中的圆圈）有一个至多63个字符长的标识
  - 这颗树的树根是没有任何标识的特殊结点
  - 命名标识中一律不区分大写和小写
  - 命名树上任何一个结点的域名就是将从该结点到最高层的域名串连起来，中间使用一个点“.”分隔这些域名
  - 域名树中的每个结点必须有一个唯一的域名，但域名树中的不同结点可使用相同的标识
  - 以点”.”结尾的域名称为绝对域名或完全合格的域名FQDN(Full Qualified Domain Name)
  - 如果一个域名不以点结尾，则认为该域名是不完全的，如何使域名完整依赖于使用的DNS软件



- 右图的域名：
  - 33.13.252.140.in-addr.arpa.
  - sun.tuc.nono.edu.
  - cnri.reston.va.us.

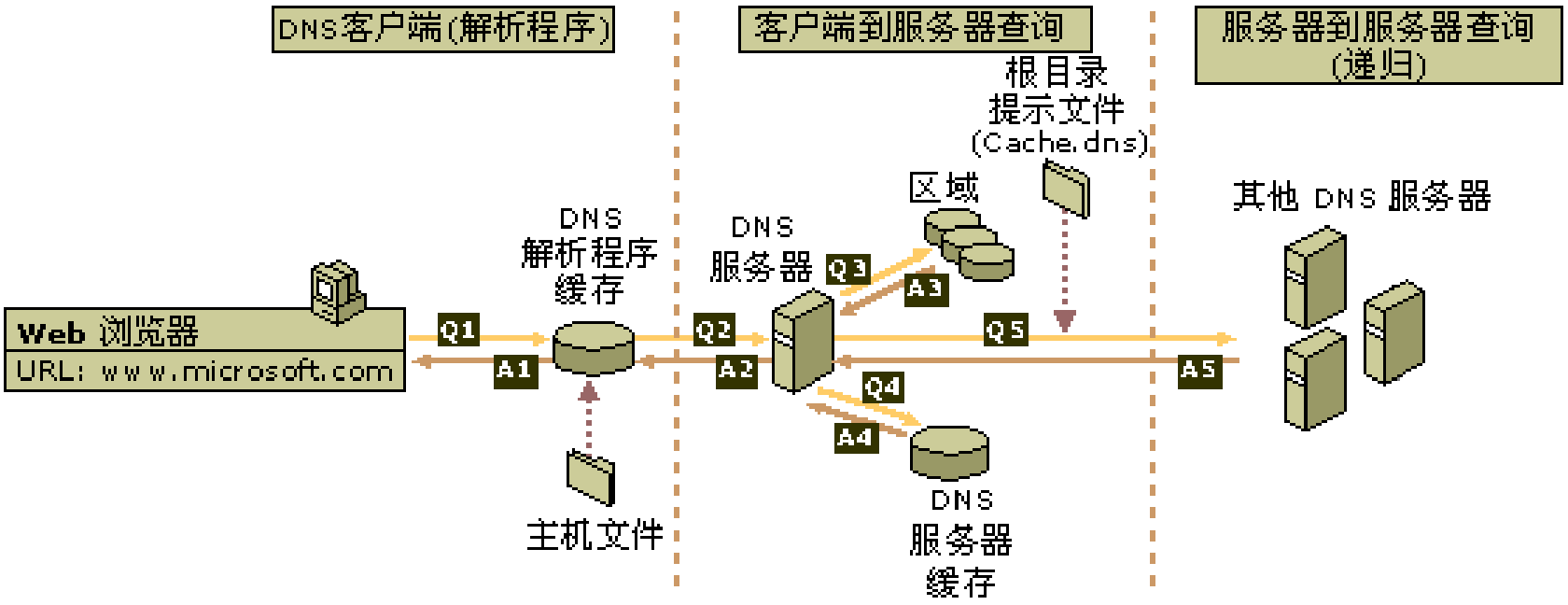
2.14.3 顶级域名

- arpa是一个用作地址到名字转换的特殊域
- 7个3字符长的普通域(组织域)
  - .com – 商业结构
  - .org -其他组织
  - .net –网络
  - .edu -教育机构
  - .gov -其他美国政府部门
  - .mil -美国军事网点
  - .int -国际组织
- 所有2字符长的域均是基于ISO3166中定义的国家代码，这些域被称为国家域(或地理域)
  - 例如：
    - .cn – 中国
    - .us – 美国
    - .hk – 香港
    - .tw – 台湾
    - …
  - 许多国家将它们的二级域组织成类似于普通域的结构，例如:
    - .co.uk是英国商业机构的二级域名
    - .gov.cn中国的政府机构
    - .edu.cn中国的大学

2.14.4 域名管理

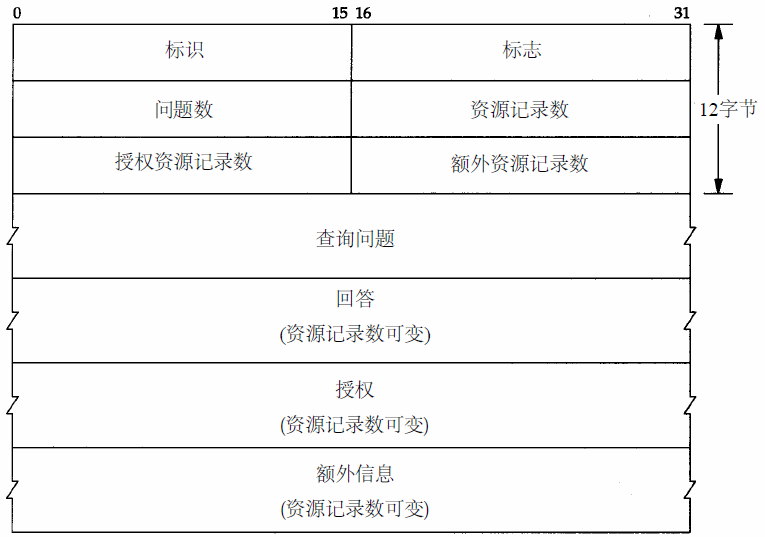
- 网络信息中心NIC负责分配顶级域和委派其他指定地域(zone)的授权机构
  - 一个独立管理的DNS子树称为一个域(zone)
- 一个域(zone)的授权机构被委派后，由它负责向该域提供多个名字服务器
  - 当一个新系统加入到一个区域中时，该区域的DNS管理者为该新系统申请一个域名和一个IP地址，并将它们加到名字服务器的数据库中
  - 一个名字服务器负责一个或多个域
  - 一个域的管理者必须为该域提供一个主名字服务器和至少一个辅助名字服务器
    - 主、辅名字服务器必须是独立和冗余的，以便当某个名字服务器发生故障时不会影响该区域的名字服务
    - 主、辅名字服务器的主要区别在于主名字服务器从磁盘文件中调入该区域的所有信息，而辅名字服务器则从主服务器调入所有信息
      - 我们将辅名字服务器从主服务器调入信息称为区域传送

2.14.5 域名请求处理流程



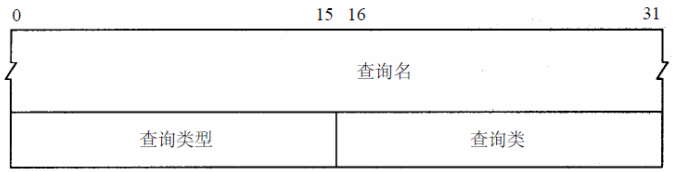
2.14.6 DNS报文

- DNS定义了一个用于查询和响应的报文格式



DNS查询和响应的一般格式

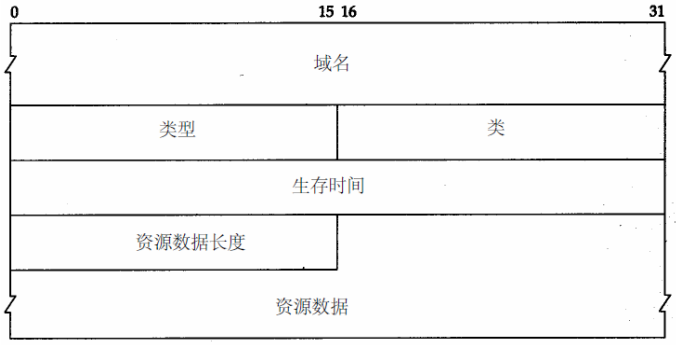
DNS报文首部中的标志字段



DNS查询报文中问题部分的格式

名　字	数　值	描　述	类型?	查询类型
A	1	IP地址	•	•
NS	2	名字服务器	•	•
CNAME	5	规范名称	•	•
PTR	12	指针记录	•	•
HINFO	13	主机信息	•	•
MX	15	邮件交换记录	•	•
AXFR	252	对区域转换的请求		•
* 或ANY	255	对所有记录的请求		•

DNS问题和响应的类型值和查询类型值



DNS响应报文中的资源记录部分

2.14.6 DNS报文解释

- QR是1bit字段，表示报文类型
  - 0表示查询报文
  - 1表示响应报文。
- opcode是一个4bit字段，表示操作码
  - 0 - 标准查询
  - 1 - 反向查询
  - 2 - 服务器状态请求
- AA是1bit标志，表示“授权回答(authoritative answer)”，该名字服务器是授权于该域的
- TC是1bit字段，表示“可截断的(truncated)”。使用UDP时，它表示当应答的总长度超过512字节时，只返回前512个字节
- RD是1bit字段表示“期望递归（recursion desired）”。该比特能在一个查询中设置，并在响应中返回
  - 如果该位为1，则告诉名字服务器必须处理这个查询，这称为一个递归查询
  - 如果该位为0，且被请求的名字服务器没有一个授权回答，它就返回一个能解答该查询的其他名字服务器列表，这称为迭代查询
- RA是1bit字段，表示“可用递归”，如果名字服务器支持递归查询，则在响应中将该比特设置为1
  - 大多数名字服务器都提供递归查询，除了某些根服务器
- RA之后的3bit字段必须为0
- rcode是一个4bit的返回码字段
  - 0 - 没有差错
  - 3 - 名字差错，名字差错只有从一个授权名字服务器上返回，它表示在查询中指定的域名不存在
- 问题数、资源记录数、授权资源记录数、额外资源记录数（分别为16bit字段）说明最后4个变长字段中包含的条目数。
  - 对于查询报文，问题(question)数通常是1，而其他3项则均为0
  - 对于应答报文，回答数至少是1，剩下的两项可以是0或非0

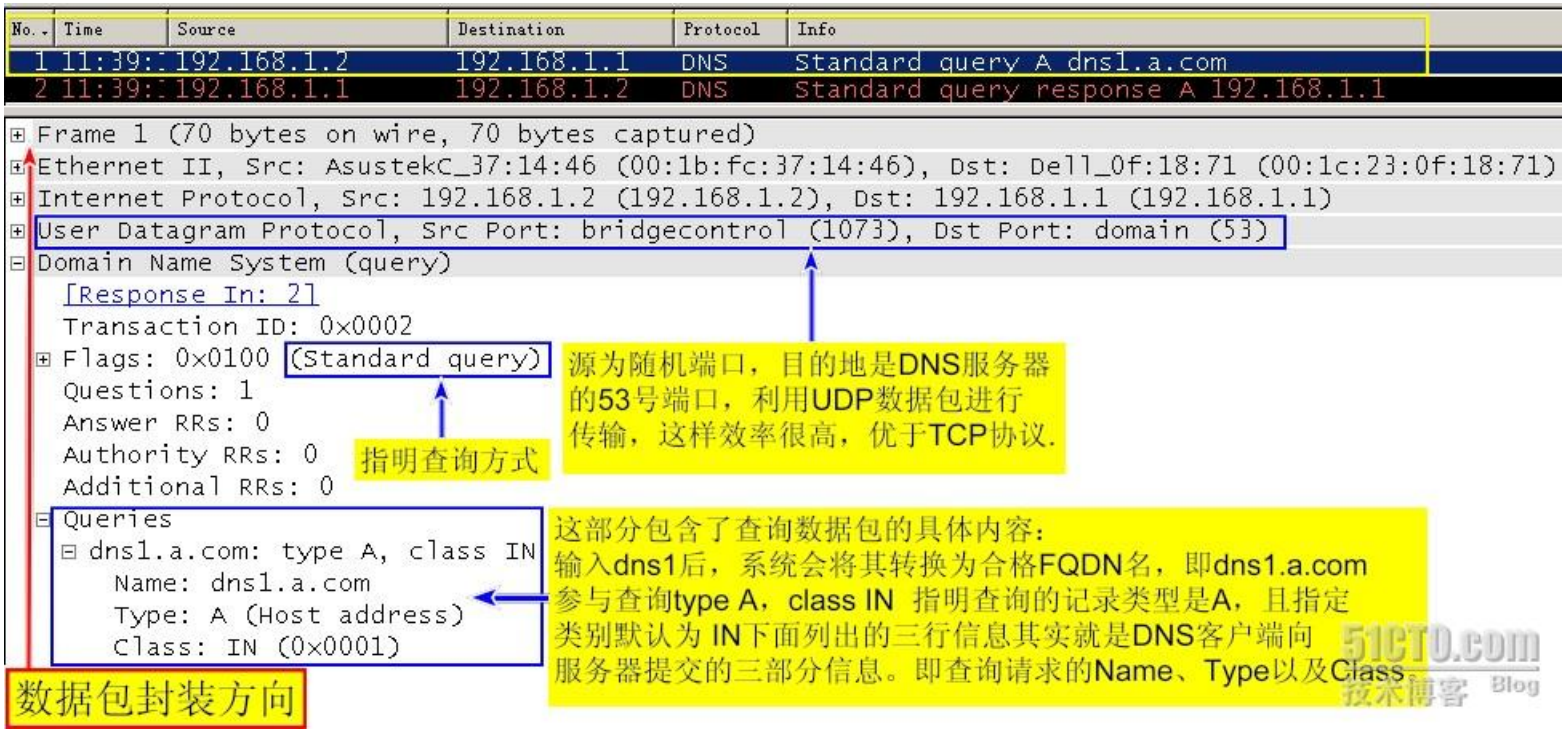
2.14.7 资源记录

- A记录
  - 一个A记录定义了一个IP地址，它存储32bit的二进制数
- PTR指针
  - 用于指针查询。IP地址被看作是in-addr.arpa域下的一个域名（标识符串）。
- CNAME
  - 这表示“规范名字(canonical name)”。它用来表示一个域名（标识符串），而有规范名字的域名通常被称为别名(alias)
- HINFO
  - 表示主机信息：包括说明主机CPU和操作系统的两个字符串
  - 并非所有的站点均提供它们系统的HINFO记录，并且提供的信息也可能不是最新的。
- MX
  - 邮件交换记录，用于以下一些场合
    - 一个没有连到Internet的站点能将一个连到Internet的站点作为它的邮件交换器。这两个站点能够用一种交替的方式交换到达的邮件，而通常使用的协议是UUCP协议。
    - MX记录提供了一种将无法到达其目的主机的邮件传送到一个替代主机的方式。
    - MX记录允许机构提供供他人发送邮件的虚拟主机，如cs.university.edu，即使这样的主机名根本不存在。
    - 防火墙网关能使用MX记录来限制外界与内部系统的连接
- NS
  - 名字服务器记录，说明一个域的授权名字服务器

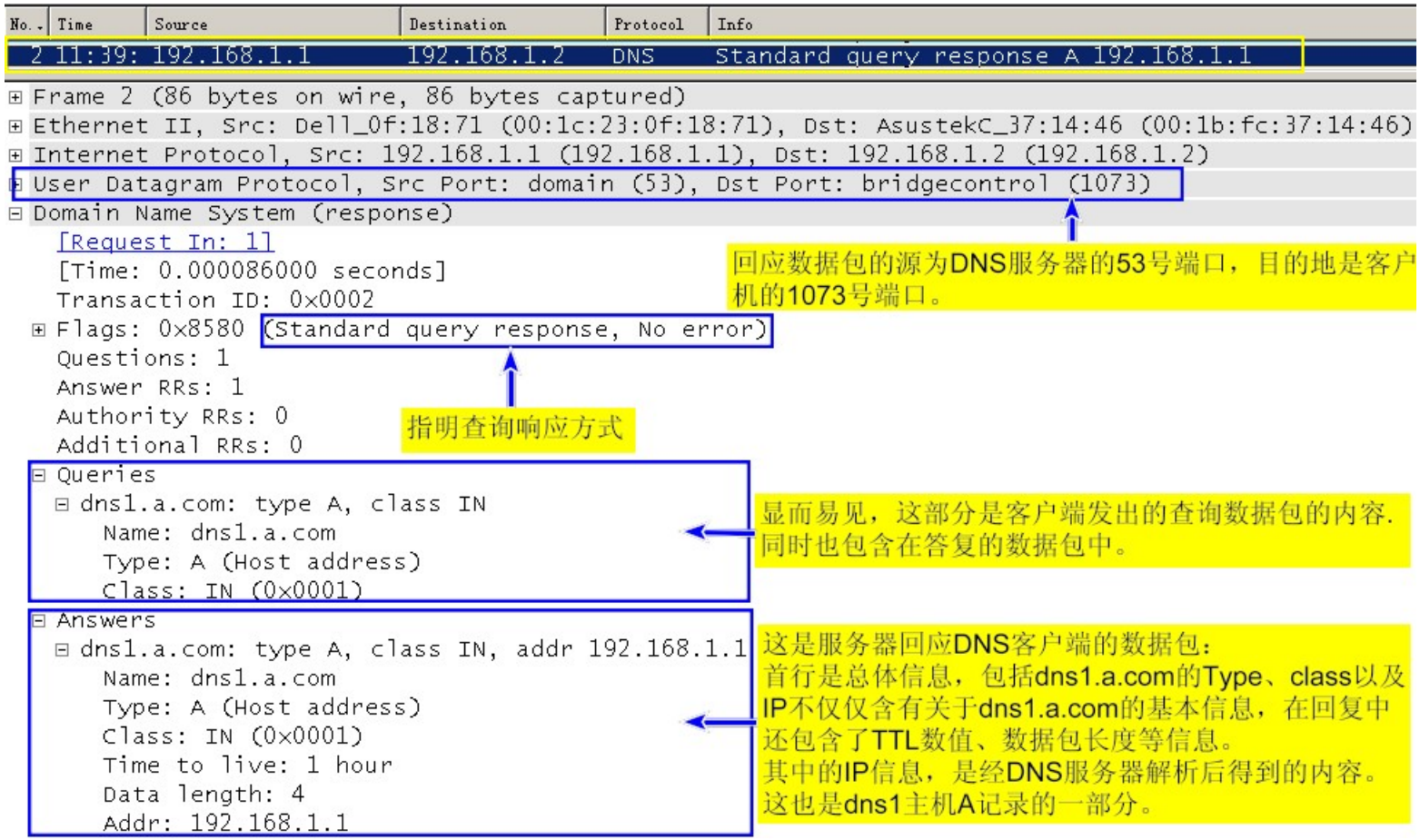
2.14.8 DNS相关的工具

- nslookup
- dig
- whois

2.14.9.1 DNS数据包分析-请求



2.14.2 DNS数据包分析-应答



3. 常用工具

- telnet
  - 可用于测试和调试TCP程序
- lsof
  - 可用于检查程序打开的文件(包括socket)
- netstat
  - Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships
- tcpdump等sniffer，如wireshark
  - 检查网络传输的细节
- nc
  - 可用于测试TCP/UDP客户端、服务器

3.1 telnet

- 使用telnet可以检查TCP服务器是否可连接

- 通过以下的测试，我们知道www.google.com的80端口上的服务可连接，即Google的http服务正常

```
[akaedu2akaedu-desktop ~/yjs]$ telnet www.google.com 80
Trying 208.67.219.230...
Connected to google.navigation.opendns.com.
Escape character is '^]'.
q
Connection closed by foreign host.
```

- 通过以下的测试，我们知道192.168.0.28的8000端口上的服务拒绝接

```
[akaedu2akaedu-desktop ~/yjs]$ telnet 192.168.0.28 8000
Trying 192.168.0.28...
telnet: Unable to connect to remote host: Connection refused
```

3.2 lsof

- lsof可以列出所有打开的文件(包括socket)
- 以下命令可以列出ssh打开的所有文件

```
[akaedu2akaedu-desktop ~/yjs]$ lsof | grep "\<ssh "
ssh       6563     akaedu  cwd       DIR        8,7     4096    7868 /home/akaedu
ssh       6563     akaedu  rtd       DIR        8,7     4096       2 /
ssh       6563     akaedu  txt       REG        8,7   287676   19197 /usr/bin/ssh
ssh       6563     akaedu  mem       REG        0,0                0 [heap] (stat: No such file or directory)
ssh       6563     akaedu  mem       REG        8,7     7552   18069 /lib/libnss_mdns4.so.2
ssh       6563     akaedu  mem       REG        8,7    17884   17942 /lib/tls/i686/cmov/libnss_dns-2.5.so
ssh       6563     akaedu  mem       REG        8,7     7084   18070 /lib/libnss_mdns4_minimal.so.2
ssh       6563     akaedu  mem       REG        8,7    38416   17944 /lib/tls/i686/cmov/libnss_files-2.5.so
ssh       6563     akaedu  mem       REG        8,7    34352   17948 /lib/tls/i686/cmov/libnss_nis-2.5.so
ssh       6563     akaedu  mem       REG        8,7    30436   17940 /lib/tls/i686/cmov/libnss_compat-2.5.so
ssh       6563     akaedu  mem       REG        8,7     9684   17933 /lib/tls/i686/cmov/libdl-2.5.so
ssh       6563     akaedu  mem       REG        8,7  1307104   17927 /lib/tls/i686/cmov/libc-2.5.so
ssh       6563     akaedu  mem       REG        8,7    14164   31390 /usr/lib/libkrb5support.so.0.0
ssh       6563     akaedu  mem       REG        8,7     5792   18033 /lib/libcom_err.so.2.1
ssh       6563     akaedu  mem       REG        8,7   151296   24678 /usr/lib/libk5crypto.so.3.0
ssh       6563     akaedu  mem       REG        8,7   512500   37675 /usr/lib/libkrb5.so.3.2
ssh       6563     akaedu  mem       REG        8,7   113800   31543 /usr/lib/libgssapi_krb5.so.2.2
ssh       6563     akaedu  mem       REG        8,7    21908   17931 /lib/tls/i686/cmov/libcrypt-2.5.so
ssh       6563     akaedu  mem       REG        8,7    79596   17938 /lib/tls/i686/cmov/libnsl-2.5.so
ssh       6563     akaedu  mem       REG        8,7    78276   27400 /usr/lib/libz.so.1.2.3
ssh       6563     akaedu  mem       REG        8,7     9696   17961 /lib/tls/i686/cmov/libutil-2.5.so
ssh       6563     akaedu  mem       REG        8,7  1299556   38429 /usr/lib/i686/cmov/libcrypto.so.0.9.8
ssh       6563     akaedu  mem       REG        8,7    67408   17955 /lib/tls/i686/cmov/libresolv-2.5.so
ssh       6563     akaedu  mem       REG        8,7   109268   18006 /lib/ld-2.5.so
ssh       6563     akaedu    0u      CHR      136,5                7 /dev/pts/5
ssh       6563     akaedu    1u      CHR      136,5                7 /dev/pts/5
ssh       6563     akaedu    2u      CHR      136,5                7 /dev/pts/5
ssh       6563     akaedu    3u     IPv4      20118              TCP akaedu-desktop.local:46325->221.221.53.47:8022 (ESTABLISHED)
ssh       6563     akaedu    4u      CHR      136,5                7 /dev/pts/5
ssh       6563     akaedu    5u      CHR      136,5                7 /dev/pts/5
ssh       6563     akaedu    6u      CHR      136,5                7 /dev/pts/5
```

3.3 netstat

- Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

- 以下命令打印当前的连接情况

```
[akaedu2akaedu-desktop ~/yjs]$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:45362         localhost:ipp           ESTABLISHED
tcp        0      0 localhost:6010          localhost:52605         ESTABLISHED
tcp        0      0 localhost:ipp           localhost:45362         ESTABLISHED
tcp        0      0 akaedu-desktop.lo:46325 221.221.53.47:8022      ESTABLISHED
tcp        0   3068 localhost:52605         localhost:6010          ESTABLISHED
tcp6       0    292 ::ffff:192.168.0.28:ssh ::ffff:192.168.0.2:1274 ESTABLISHED
tcp6       0      0 ::ffff:192.168.0.28:ssh ::ffff:192.168.0.2:1268 ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ]         DGRAM                    7837   @/com/ubuntu/upstart
unix  2      [ ]         DGRAM                    8001   @/org/kernel/udev/udevd
unix  2      [ ]         DGRAM                    14428  @/org/freedesktop/hal/udev_event
unix  13     [ ]         DGRAM                    14317  /dev/log
unix  3      [ ]         STREAM     CONNECTED     19689
unix  3      [ ]         STREAM     CONNECTED     19688
unix  3      [ ]         STREAM     CONNECTED     19686  /tmp/.esd-1000/socket
unix  3      [ ]         STREAM     CONNECTED     19685
```

3.3 netstat

- 以下命令打印当前的路由表

```
[akaedu2akaedu-desktop ~/yjs]$ netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.0.0     0.0.0.0         255.255.255.0   U         0 0          0 eth2
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth2
0.0.0.0         192.168.0.1     0.0.0.0         UG        0 0          0 eth2
```

3.3 netstat

- 以下命令打印所有处于listen状态的socket

```
[akaedu2akaedu-desktop ~/yjs]$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:2208          *:*                     LISTEN
tcp        0      0 *:nfs                   *:*                     LISTEN
tcp        0      0 *:35458                 *:*                     LISTEN
tcp        0      0 *:812                   *:*                     LISTEN
tcp        0      0 *:sunrpc                *:*                     LISTEN
tcp        0      0 *:ftp                   *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp        0      0 *:54329                 *:*                     LISTEN
tcp        0      0 localhost:6010          *:*                     LISTEN
tcp        0      0 localhost:2207          *:*                     LISTEN
tcp6       0      0 *:ssh                   *:*                     LISTEN
tcp6       0      0 ip6-localhost:6010      *:*                     LISTEN
udp        0      0 *:32768                 *:*
udp        0      0 *:nfs                   *:*
udp        0      0 *:32770                 *:*
udp        0      0 *:32771                 *:*
udp        0      0 *:934                   *:*
udp        0      0 *:809                   *:*
udp        0      0 *:bootpc                *:*
udp        0      0 *:tftp                  *:*
udp        0      0 *:mdns                  *:*
udp        0      0 *:sunrpc                *:*
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING     15618    /var/run/avahi-daemon/socket
unix  2      [ ACC ]     STREAM     LISTENING     16806    @/tmp/dbus-i11XYwGHiU
unix  2      [ ACC ]     STREAM     LISTENING     14401    /var/run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     15646    @/tmp/dbus-5BzEOMQ2eW
unix  2      [ ACC ]     STREAM     LISTENING     17035    /tmp/scim-socket-frontend-akaedu
unix  2      [ ACC ]     STREAM     LISTENING     17092    /tmp/scim-helper-manager-socket-akaedu
unix  2      [ ACC ]     STREAM     LISTENING     17096    /tmp/scim-panel-socket:0-akaedu
unix  2      [ ACC ]     STREAM     LISTENING     17637    /tmp/mapping-akaedu
unix  2      [ ACC ]     STREAM     LISTENING     14188    /var/run/acpid.socket
```

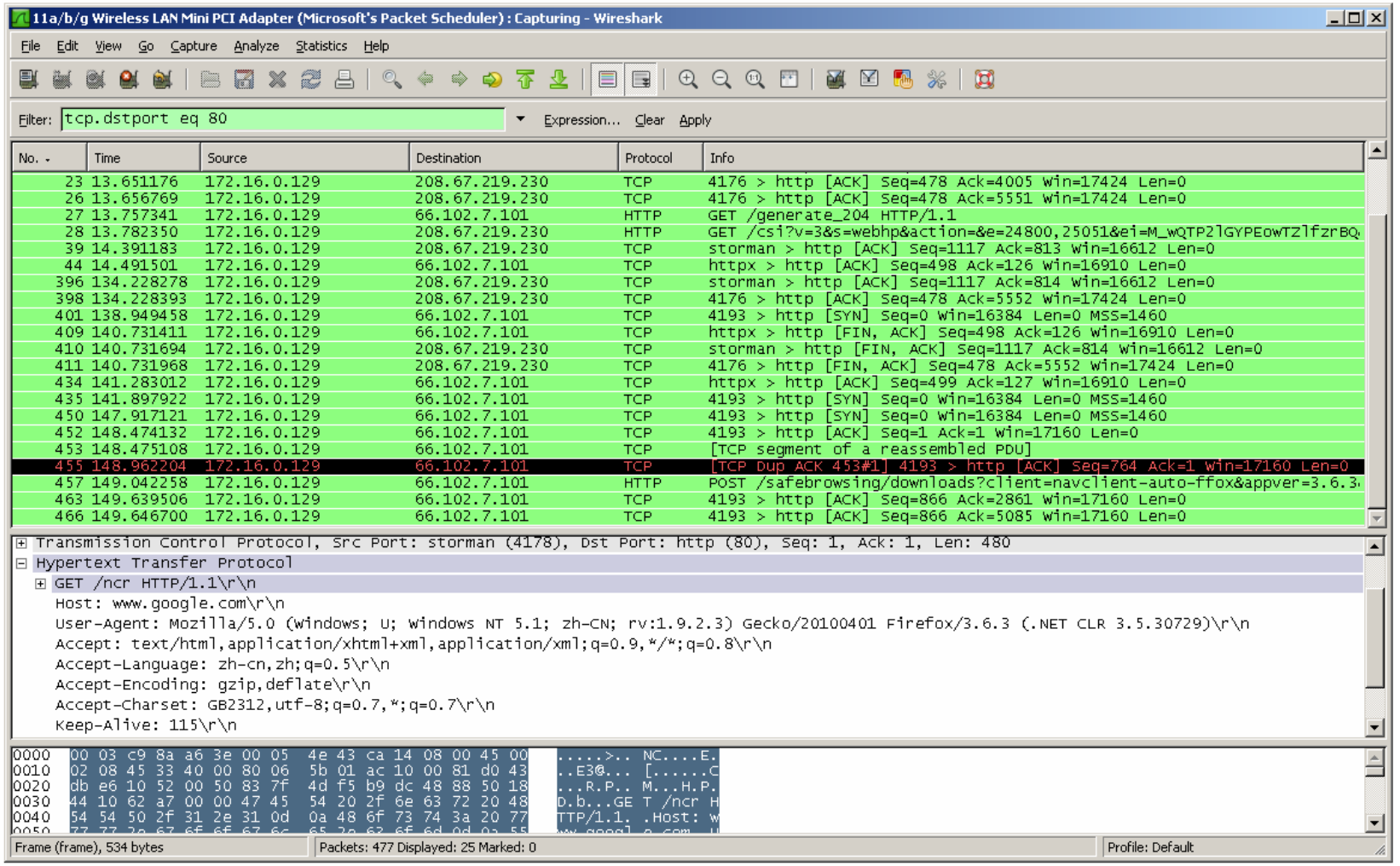
3.4.1 tcpdump

- 以下命令捕获源或者目的为80端口的TCP包

```
[akaedu2akaedu-desktop ~/yjs]$ sudo tcpdump tcp port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 96 bytes
08:59:26.056693 IP akaedu-desktop.local.35253 > google.navigation.opendns.com.www: S 2881298031:2881298031
imestamp 925797 0,nop,wscale 2>
08:59:26.277756 IP google.navigation.opendns.com.www > akaedu-desktop.local.35253: S 2407701970:2407701970
ss 1452,nop,wscale 3,sackOK,timestamp 3049848329 925797>
08:59:26.277778 IP akaedu-desktop.local.35253 > google.navigation.opendns.com.www: . ack 1 win 1460 <nop,n
08:59:31.609391 IP akaedu-desktop.local.35253 > google.navigation.opendns.com.www: P 1:8(7) ack 1 win 1460
848329>
08:59:31.858256 IP google.navigation.opendns.com.www > akaedu-desktop.local.35253: P 1:446(445) ack 8 win
909 927185>
08:59:31.858279 IP akaedu-desktop.local.35253 > google.navigation.opendns.com.www: . ack 446 win 1728 <nop
9>
08:59:31.858520 IP google.navigation.opendns.com.www > akaedu-desktop.local.35253: F 446:446(0) ack 8 win
909 927185>
08:59:31.858591 IP akaedu-desktop.local.35253 > google.navigation.opendns.com.www: F 8:8(0) ack 447 win 17
49853909>
08:59:32.078885 IP google.navigation.opendns.com.www > akaedu-desktop.local.35253: . ack 9 win 8279 <nop,n
```

- Manpage of tcpdump
  - http://www.tcpdump.org/tcpdump_man.html

3.4.2 wireshark

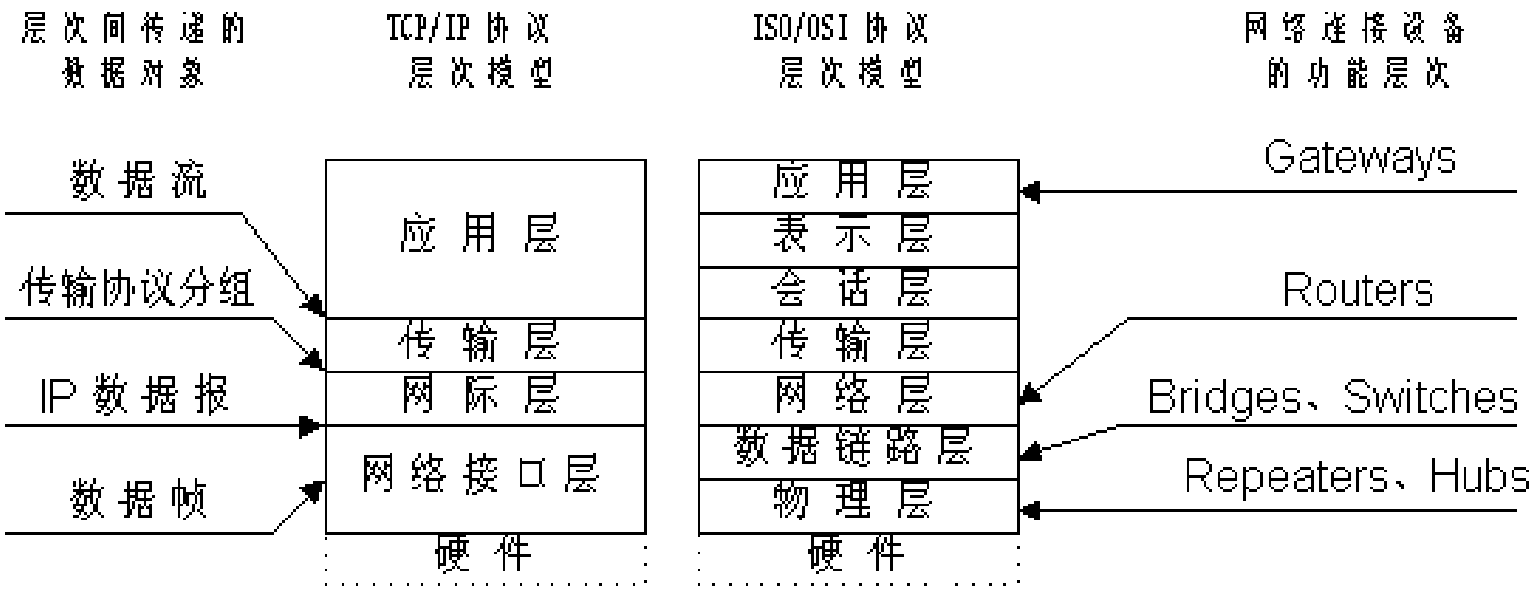


- Wireshark User's Guide: http://www.wireshark.org/docs/wsug_html/

4. 网络设备

- 中继器（Repeator）
- 集线器（Hub）
- 网桥（Bridge）
- 交换机（LAN Switch）
- 路由器（Router）
- 网关（Gateway）

4.1 网络设备所处的分层



4.2 中继器(Repeater)

- 中继器是位于第1层（OSI参考模型的物理层）的网络设备
- 当数据离开源在网络上传送时，它是转换为能够沿着网络介质传输的电脉冲或光脉冲的——这些脉冲称为信号（signal）
- 当信号离开发送工作站时，信号是规划的，而且很容易辨认出来。但是，当信号沿着网络介质进行传送时， 随着经过的线缆越来越长，信号就会变得越来越弱，越来越差。
- 中继器的目的是在比特级别对网络信号进行再生和重定时，从而使得它们能够在网络上传输更长的距离

4.3 网桥(Bridge)

- 网桥是第2层的设备，它设计用来创建两个或多个LAN分段
  - 每一个分段都是一个独立的冲突域
  - 网桥设计用来产生更大可用宽带
- 网桥的目的是过滤LAN的通信流，使得本地的通信流保留在本地，而让那些定向到LAN其他部分（分段）的通信流转发到那里去
  - 每一台网络设备在NIC（网络接口卡）中都有一个惟一的MAC（介质访问控制）地址
  - 网桥会记录它每一边的MAC地址，然后基于这张MAC地址表作出转发决策
- 以下是网桥的一些重要特性：
  - 网桥比集线器更为智能。它只运行在第2层，就是说，它能分析传入的帧，并且能基于寻址信息进行转发或丢弃它们
  - 网桥在两个或多个LAN分段之间收集和转发分组
  - 网桥创建更多的冲突域，使得多台设备能同时无冲突地发送
  - 网桥维持MAC地址表，称为网桥表

4.4 集线器(HUB)

- 集线器的特性与中继器很相似，实际就是一种多端口的中继器(multiport repeater)
- 集线器是网络中各个设备的通用连接点，它通常用于连接LAN的分段
  - 集线器具有多个端口，每一个分组到达某个端口时，都会被复制到其他所有端口，以便所有的LAN分段都能看见所有的分组
  - 集线器并不认识信号、地址或数据中任何信息模式
- 集线器一般有4、8、16、24、32等数量的RJ45接口，通过这些接口，集线器便能为相应数量的电脑完成“中继”功能（将已经衰减得不完整的信号经过整理，重新产生出完整的信号再继续传送）。由于它在网络中处于一种“中心”位置，因此集线器也叫做“Hub”。
- 集线器的工作原理很简单，比如有一个具备8个端口的集线器，共连接了8台电脑。集线器处于网络的“中心”，通过集线器对信号进行转发，8台电脑之间可以互连互通。具体通信过程是这样的：假如计算机1要将一条信息发送给计算机8，当计算机1的网卡将信息通过双绞线送到集线器上时，集线器并不会直接将信息送给计算机8，它会将信息进行“广播”——将信息同时发送给8个端口，当8个端口上的计算机接收到这条广播信息时，会对信息进行检查，如果发现该信息是发给自己的，则接收，否则不予理睬。由于该信息是计算机1发给计算机8的，因此最终计算机8会接收该信息，而其它7台电脑看完信息后，会因为信息不是自己的而不接收该信息。
- 以下是集线器最为重要的特性
  - 放大信号
  - 在整个网络传播信号
  - 无需过滤
  - 无需路径判定或交换
  - 用作网络会集点

4.5 交换机(Switch)

- 交换机也叫交换式集线器，它通过对信息进行重新生成，并经过内部处理后转发至指定端口，具备自动寻址能力和交换作用，由于交换机根据所传递信息包的目的地址，将每一信息包独立地从源端口送至目的端口，避免了和其他端口发生碰撞。
- 广义的交换机就是一种在通信系统中完成信息交换功能的设备。
- 在计算机网络系统中，交换机是针对共享工作模式的弱点而推出的。集线器是采用共享工作模式的代表，如果把集线器比作一个邮递员，那么这个邮递员是个不认识字的“傻瓜”——要他去送信，他不知道直接根据信件上的地址将信件送给收信人，只会拿着信分发给所有的人，然后让接收的人根据地址信息来判断是不是自己的!而交换机则是一个“聪明”的邮递员——交换机拥有一条高带宽的背部总线和内部交换矩阵。交换机的所有的端口都挂接在这条背部总线上，当控制电路收到数据包以后，处理端口会查找内存中的地址对照表以确定目的MAC（网卡的硬件地址）的NIC（网卡）挂接在哪个端口上，通过内部交换矩阵迅速将数据包传送到目的端口。目的MAC若不存在，交换机才广播到所有的端口，接收端口回应后交换机会“学习”新的地址，并把它添加入内部地址表中。
- 可见，交换机在收到某个网卡发过来的“信件”时，会根据上面的地址信息，以及自己掌握的“常住居民户口簿”快速将信件送到收信人的手中。万一收信人的地址不在“户口簿”上，交换机才会像集线器一样将信分发给所有的人，然后从中找到收信人。而找到收信人之后，交换机会立刻将这个人的信息登记到“户口簿”上，这样以后再为该客户服务时，就可以迅速将信件送达了。

4.6 路由器(Router)

- 路由器是网络中进行网间连接的关键设备。作为不同网络之间互相连接的枢纽，路由器系统构成了基于TCP/IP 的国际互连网络Internet 的主体脉络。
- 路由器之所以在互连网络中处于关键地位，是因为它处于网络层，一方面能够跨越不同的物理网络类型（DDN、FDDI、以太网等等），另一方面在逻辑上将整个互连网络分割成逻辑上独立的网络单位，使网络具有一定的逻辑结构。路由器的主要工作就是为经过路由器的每个数据帧寻找一条最佳传输路径，并将该数据有效地传送到目的站点。
- 路由器的基本功能是，把数据（IP 报文）传送到正确的网络，细分则包括：
  - IP 数据报的转发，包括数据报的寻径和传送;
  - 子网隔离，抑制广播风暴;
  - 维护路由表，并与其它路由器交换路由信息，这是 IP 报文转发的基础;
  - IP 数据报的差错处理及简单的拥塞控制;
  - 实现对 IP 数据报的过滤和记帐。
- 路由器构成了Internet 的骨架。它的处理速度是网络通信的主要瓶颈之一，它的可靠性则直接影响着网络互连的质量。因此Internet 研究领域中，路由器技术始终处于核心地位。

4.7 网关(Gateway)

- 网关(Gateway)又称网间连接器、协议转换器
- 网关在传输层上以实现网络互连，是最复杂的网络互连设备，仅用于两个高层协议不同的网络互连
- 网关既可以用于广域网互连，也可以用于局域网互连
- 网关是一种充当转换重任的计算机系统或设备
- 在使用不同的通信协议、数据格式或语言，甚至体系结构完全不同的两种系统之间，网关是一个翻译器
- 与网桥只是简单地传达信息不同，网关对收到的信息要重新打包，以适应目的系统的需求
- 网关也可以提供过滤和安全功能
- 大多数网关运行在OSI 7层协议的顶层--应用层

Any questions?

Contact:

- yjs@oldhand.org
- 133 0122 6268