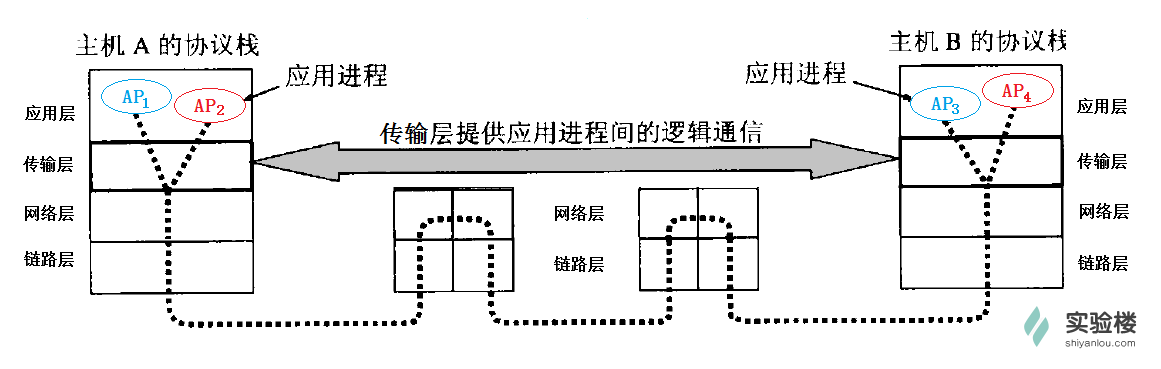
# 传输层：UDP 协议

## 一、传输层协议

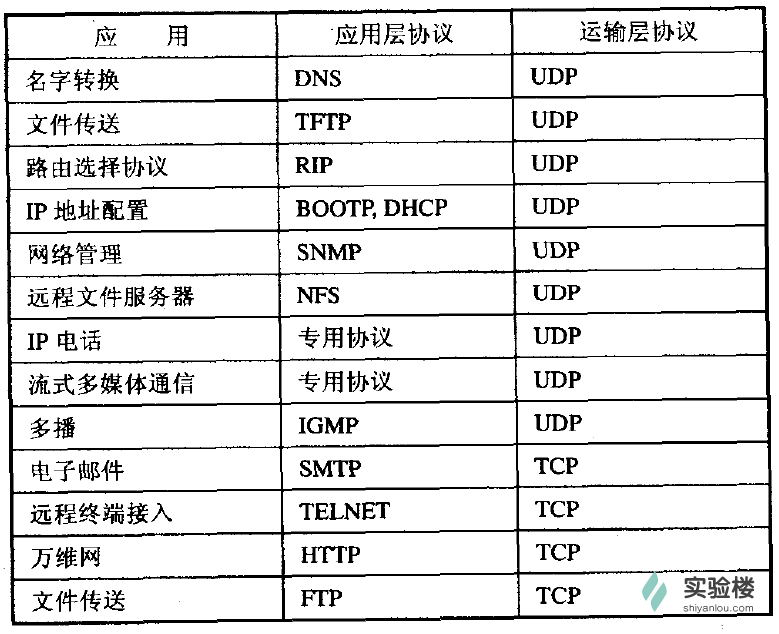
从之前介绍的网络层协议来看，通信的两端是两台主机，IP 数据报首部就标明了这两台主机的 IP 地址。但是从传输层来看，是发送方主机中的一个**进程**与接收方主机中的一个**进程**在交换数据，因此，严格地讲，通信双方不是主机，而是主机中的**进程**。

主机中常常有多个应用进程同时在与外部通信(比如你的浏览器和 QQ 在同时运行)，下图中，A 主机的 AP1 进程在于 B 主机的 AP3 进程通信，同时主机 A 的 AP2 进程也在与 B 主机的 AP4 进程通信。

两个主机的传输层之间有一个灰色双向箭头，写着“传输层提供应用进程间的逻辑通信”。 **逻辑通信**：看起来是数据似乎是沿着双向箭头在传输层水平传输的，但实际上是沿图中的虚线经多个协议层次而传输。



TCP/IP 协议栈传输层有两个重要协议——UDP 和 TCP，不同的应用进程在传输层使用 TCP 或 UDP 之一：



这一节先介绍比较简单的 UDP，比较复杂的 TCP 将在下一节讨论。

## 二、端口

在第一节我们已经了解过端口的概念，端口的作用体现在传输层。

刚才的图中，AP1 与 AP3 的通信与 AP2 与 AP4 的通信可以使用同一个传输层协议来传输(TCP 或 UDP)，根据 IP 地址或 MAC 地址都只能是把数据传到正确的主机，但具体需要传到哪一个进程，是通过端口来辨认的。比如同时使用浏览器和 QQ，浏览器占用 80 端口，而 QQ 占用 4000 端口，那么发送过来的 QQ 消息便会通过 4000 端口显示在 QQ 客户端，而不会错误地显示在浏览器上。

端口号有 0～65535 的编号，其中：

* 编号 0～1023 为 **系统端口号** ，这些端口号可以在网址 www.iana.org 查询到，它们被指派给了 TCP/IP 最重要的一些应用程序，以下是一些常见的系统端口号：

| **应用层协议：** | **FTP** | **TELNET** | **SMTP** | **DNS** | **TFTP** | **HTTP** | **SNMP** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 系统端口号： | 21 | 23 | 25 | 53 | 69 | 80 | 161 |

编号 1024～49151 为 **登记端口号** ，为没有系统端口号的应用程序使用，使用这类端口号必须在 IANA 按规定手续登记，以防止重复。

编号 49152～65535 为 **短暂端口号** ，是留给客户进程选择暂时使用的，使用结束后，这类端口号会被放开以供其它程序使用。

## 三、UDP 概述

UDP(User Datagram Protocol)用户数据报协议，它只在 IP 数据报服务之上增加了很少一点功能，它的主要特点有：

(1).UDP 是**无连接**的，发送数据之前不需要建立连接(而 TCP 需要)，减少了开销和时延。

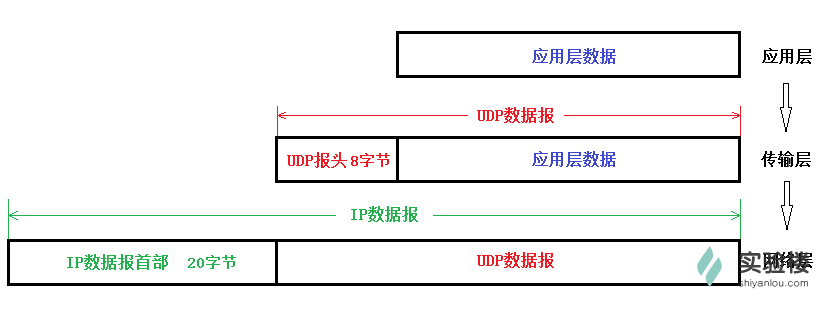
(2).UDP**尽最大努力交付**，不保证交付可靠性。

(3).UDP 是面向报文的，对于从网络层交付下来的 IP 数据报，只做很简单的封装(8 字节 UDP 报头)，首部开销小。

(4).UDP 没有拥塞控制，出现网络拥塞时发送方也不会降低发送速率。这种特性对某些实时应用是很重要的，比如 IP 电话，视频会议等，它们允许拥塞时丢失一些数据，因为如果不抛弃这些数据，极可能造成时延的累积。

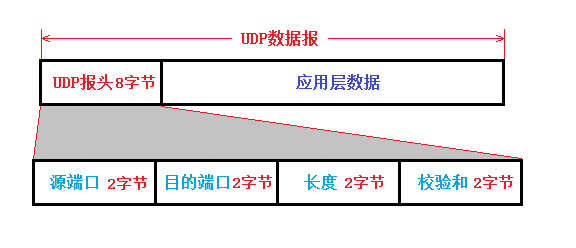
(5).UDP 支持一对一、一对多、多对一和多对多的交互通信。

从应用层到传输层，再到网络层的各层次封装：



## 四、UDP 报文

UDP 数据报可分为两部分：UDP 报头和数据部分。其中数据部分是应用层交付下来的数据。UDP 报头总共 8 字节，而这 8 字节又分为 4 个字段：



(1)源端口 2 字节 在对方需要回信时可用，不需要时可以全 0；

(2)目的端口 2 字节 必须，也是最重要的字段；

(3)长度 2 字节 长度值包括报头和数据部分；

(4)校验和 2 字节 用于检验 UDP 数据报在传输过程中是否有出错，有错就丢弃。

## 五、tcpdump 抓取 UDP 报文

现在我们动手实践，尝试抓取一个 UDP 数据报，并解读其内容。

我们需要一个小程序，用于向 **指定 IP 地址** 的 **指定端口** 发送一个 **指定内容** 的 UDP 数据报，这个程序已经编写好，依次输入以下命令，使用 github 把它下载下来，并编译：

cd Desktop

git clone http://git.shiyanlou.com/shiyanlou/tcp\_ip\_5

cd tcp\_ip\_5

gcc -o test test.c

这个 C 程序会向 IP 地址 **192.168.1.1** 的 **7777 端口** 发送一条 "hello" 消息。你可以用编辑器修改程序，向不同的 IP 不同的 IP 发送不同的内容。

编译完成后先别运行，我们还需要使用一个知名的抓包工具 **tcpdump** ，依次输入以下命令安装，并运行 tcpdump：

sudo apt-get update

sudo apt-get install tcpdump

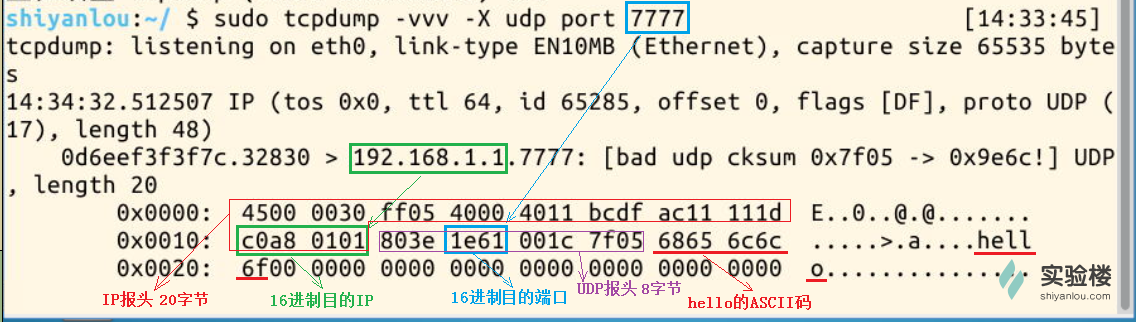
sudo tcpdump -vvv -X udp port 7777

现在最小化当前终端，另开启一个终端，输入以下命令运行刚才编译好的 C 程序 test：

cd Desktop/tcp\_ip\_5

./test

test 程序运行结束，返回刚才运行 tcpdump 的终端查看抓包结果：



## 六、作业

1、使用 **tcpdump** 抓取 UDP 数据报，解读报文，并截图。

2、修改 C 程序，向不同的 IP，不同的端口发送不同的内容，截图抓包结果。