

应用层

应用层是直接为用户打交道的。

应用程序体系结构

1. C/S体系结构: 如web服务器响应浏览器的请求。
 - 服务器的IP地址一般是固定的, 服务器是一直运行的。
2. P2P体系结构: 如文件共享 (BitTorrent), 对等方协助下载加速器 (如迅雷), 因特网电话 (Skype) 和 IPTV (迅雷看看, PPstream)。
 - 对数据中心的专用服务器有着最小 (或没有) 依赖, 区别于C/S体系结构
 - 端对端直接通信, 一般不通过中继的服务器, 因此被称为对等方。
3. 混合体系结构: 如即时通信, 服务器用于跟踪用户的IP地址, 但用户到用户的报文在用户主机之间直接发送 (无需通过中间服务器)

进程通信

端系统中进行通信的, 按照操作系统术语来说, 是**进程**, 每个进程占用一个端口进行通信, 当然具体通信规则每个端系统是不同的。

同一个主机的不同进程也需要通信。但不属于网络通信范畴内。

应用层的数据称之为**报文 (message)**。进程就是通过计算机网络交换报文进行通信。

客户和服务进程

Web应用程序中, 一个客户浏览器进程与一台web服务器进程交换报文。在P2P文件共享系统中, 文件从一个对等方的进程传输到另一个对等方的进程。

1. 每队通信进程, 分别标识为**客户和服务**:
 - web而言, 浏览器为客户进程, web服务器为服务器进程。
 - p2p文件共享, 下载文件的对等方为客户, 上载的为服务器。
2. 进程与计算机网络之间, 通过被称之为**套接字 (socket)** 软件接口进行发送报文和接收报文。
 - 套接字也称之为应用程序和网络之间的API, 应用程序可以控制套接字在应用层端的一切, 但是对其传输层的行为没有控制权限。应用程序有以下控制权:
 1. 选择传输层协议 (TCP还是UDP等)
 2. 设置某些传输层参数, 传递给传输层, 如最大缓存, 最大报文段 (传输层的数据单位) 长度等。
3. 进程寻址: 一台主机的进程和另一个主机的进程通信, 有以下信息标识各自的地址,
 1. 主机的地址 (IP地址标识)
 2. 进程在主机中的标识符 (如双方约定好的端口号)

可供应用程序使用的运输服务

运输层提供不同的服务可供应用程序使用。类似于两个城市之间可以坐火车或坐飞机等交通工具。

有如下服务：

1. 可靠数据服务：该服务确保应用层接收到的是完全无差错的数据。发送进程只要将其数据传递给套接字，就可以完全相信该数据可以完全无差错的到达接收进程。
 - 一些应用可以容忍一定的丢包等数据丢失，如音频/视频等。可以不必采用可靠数据服务。
2. 吞吐量：两个进程之间的数据比特发送和接收的速率。
 - 运输层协议可以确保一个最低吞吐量，例如因特网电话程序对语音以32kbps的速率编码，那么运输层必须以不低于这个速率向网络传输数据，如果不能提供这种吞吐量，该应用程序或以较低速率进行编码，或可能放弃本次发送。这是因为因特网电话而言，接收所需吞吐量的一半几乎没有任何用处。此类对吞吐量有要求的应用称之为**带宽敏感的应用**。
3. 定时：运输层可以提供定时服务，常用于实时应用。
4. 安全性：如是否加密，是否检测完整性等等。

TCP/UDP

TCP服务：

- 面向连接的服务，即端到端之间先建立连接
- 可靠的数据传送服务，即无差错，按适当顺序（不是乱序）交付发送的数据，具体为数据（字节流）传递给套接字，运输层TCP将相同的字节流交付给接收方的套接字，没有字节丢失和冗余。
- 拥塞控制服务

UDP服务：无连接，无保障的服务，不提供拥塞控制

然而因特网协议即TCP/UDP协议不提供定时和吞吐量服务，应用层在TCP的基础上可以使用SSL，以提供安全性。今天的因特网通常可以为保障时间敏感应用提供满意的服务，但不能提供任何定时或带宽要求。

应用层协议

SMTP 电子邮件 TCP

Telnet/SSH 远程终端访问 TCP

HTTP web服务 TCP

应用层协议定义了（指明不同端系统的应用程序如何相互传递报文和理解报文）：

- 交换的报文类型，如请求报文和响应报文
- 报文的语法，格式等，如报文中的各个字段如何描述
- 各个字段的语义，即各个字段代表什么含义
- 进程何时发送，如何发送，如何相应等规则