

# 网络七层模型

## 引言

网络七层模型&&网络数据包

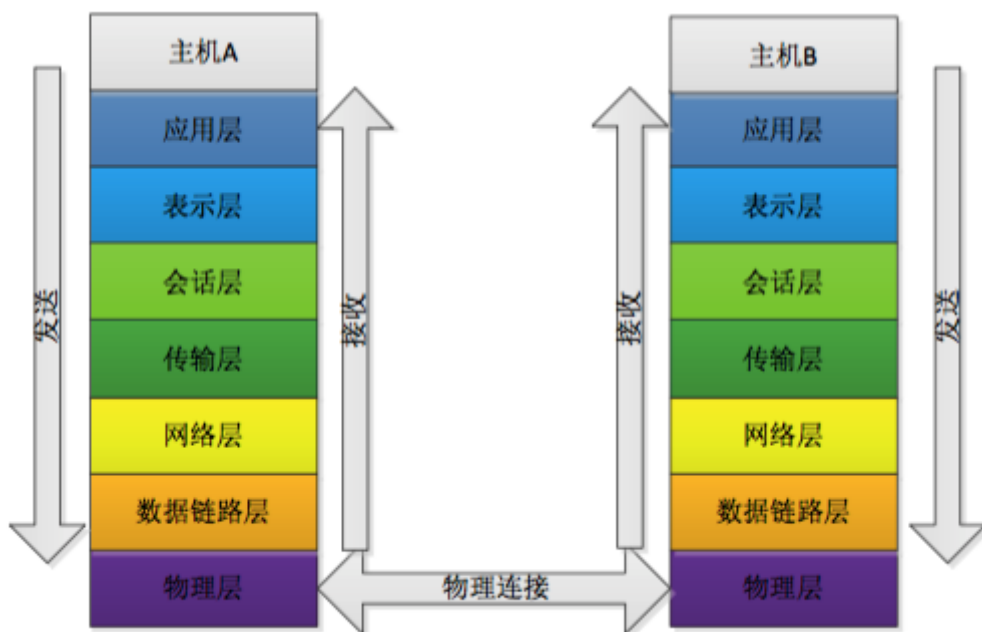
## 网络基本概念

### OSI模型

OSI 模型(Open System Interconnection model)是一个由国际标准化组织提出的概念模型,试图提供一个使各种不同的计算机和网络在世界范围内实现互联的标准框架。

它将计算机网络体系结构划分为七层,每层都可以提供抽象良好的接口。了解 OSI 模型有助于理解实际上互联网络的工业标准——TCP/IP 协议。

OSI 模型各层间关系和通讯时的数据流向如图所示:



OSI 模型

显然、如果一个东西想包罗万象、一般时不可能的；在实际的开发应用中一般时在此模型的基础上进行裁剪、整合！

## 七层模型介绍

- 物理层：

物理层负责最后将信息编码成电流脉冲或其它信号用于网上传输；

eg：RJ45等将数据转化成0和1；

- 数据链路层：

数据链路层通过物理网络链路□供数据传输。不同的数据链路层定义了不同的网络和协议特征,其中包括物理编址、网络拓扑结构、错误校验、数据帧序列以及流控；

可以简单的理解为：规定了0和1的分包形式，确定了网络数据包的形式；

- 网络层

网络层负责在源和终点之间建立连接；

可以理解为，此处需要确定计算机的位置，怎么确定？IPv4，IPv6！

- 传输层

传输层向高层□提供可靠的端到端的网络数据流服务。

可以理解为：每一个应用程序都会在网卡注册一个端口号，该层就是端口与端口的通信！常用的（TCP / IP）协议；

- 会话层

会话层建立、管理和终止表示层与实体之间的通信会话；

建立一个连接（自动的手机信息、自动的网络寻址）；

- 表示层：

表示层□供多种功能用于应用层数据编码和转化,以确保以一个系统应用层发送的信息 可以被另一个系统应用层识别；

可以理解为：解决不同系统之间的通信，eg：Linux下的QQ和Windows下的QQ可以通信；

- 应用层：

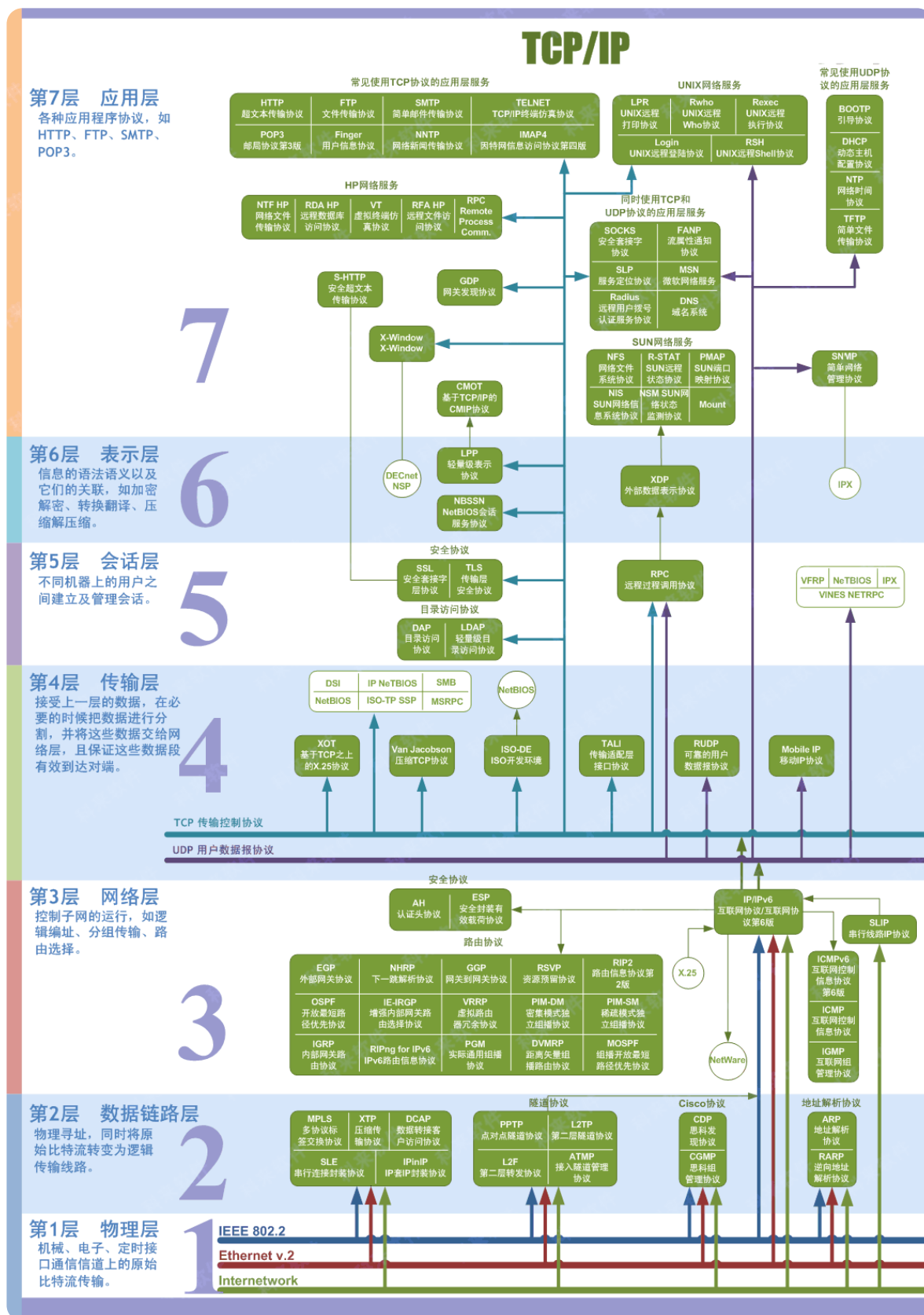
OSI 的应用层协议包括文件的传输、访问及管理协议(FTAM) ,以及文件虚拟终端协议(VIP)和公用管理系统信息(CMIP)等；

规定数据的传输协议；

常见的应用层协议：

协议	端口	说明
HTTP	80	超文本传输协议
HTTPS	443	HTTP+SSL,HTTP的安全版
FTP	20,21,990	文件传输协议
POP3	110	邮局协议
SMTP	25	简单邮件传输协议
telnet	23	远程终端协议

常见的应用层协议



互联网分层结构的好处: 上层的变动完全不影响下层的结构。

OSI 七层模型通过七个层次化的结构模型使不同的系统不同的网络之间实现可靠的通讯，因此其最主要的功能就是帮助不同类型的主机实现数据传输。

完成中继功能的节点通常称为中继系统。在OSI七层模型中，处于不同层的中继系统具有不同的名称。

一个设备工作在哪一层，关键看它工作时利用哪一层的数据头部信息。网桥工作时，是以MAC头部来决定转发端口的，因此显然它是数据链路层的设备。

具体说：

物理层：网卡，网线，集线器，中继器，调制解调器

数据链路层：网桥，交换机

网络层：路由器

网关工作在第四层传输层及其以上

集线器是物理层设备,采用广播的形式来传输信息。

交换机就是用来进行报文交换的机器。多为链路层设备(二层交换机)，能够进行地址学习，采用存储转发的形式来交换报文。

路由器的一个作用是连通不同的网络，另一个作用是选择信息传送的线路。选择通畅快捷的近路，能大大提高通信速度，减轻网络系统通信负荷，节约网络系统资源，提高网络系统畅通率。

交换机和路由器的区别

交换机拥有一条很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总线上，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若不存在则广播到所有的端口，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MAC地址表中。

使用交换机也可以把网络“分段”，通过对照MAC地址表，交换机只允许必要的网络流量通过交换机。通过交换机的过滤和转发，可以有效的隔离广播风暴，减少误包和错包的出现，避免共享冲突。

交换机在同一时刻可进行多个端口对之间的数据传输。每一端口都可视为独立的网段，连接在其上的网络设备独自享有全部的带宽，无须同其他设备竞争使用。当节点A向节点D发送数据时，节点B可同时向节点C发送数据，而且这两个传输都享有网络的全部带宽，都有着自己的虚拟连接。假使这里使用的是10Mbps的以太网交换机，那么该交换机这时的总流量就等于 $2 \times 10\text{Mbps} = 20\text{Mbps}$ ，而使用10Mbps的共享式HUB时，一个HUB的

总流量也不会超出10Mbps。

总之，交换机是一种基于MAC地址识别，能完成封装转发数据包功能的网络设备。交换机可以“学习”MAC地址，并将其存放在内部地址表中，通过在数据帧的始发者和目标接收者之间建立临时的交换路径，使数据帧直接由源地址到达目的地。

从过滤网络流量的角度来看，路由器的作用与交换机和网桥非常相似。但是与工作在网络物理层，从物理上划分网段的交换机不同，路由器使用专门的软件协议从逻辑上对整个网络进行划分。例如，一台支持IP协议的路由器可以把网络划分成多个子网段，只有指向特殊IP地址的网络流量才可以通过路由器。对于每一个接收到的数据包，路由器都会重新计算其校验值，并写入新的物理地址。因此，使用路由器转发和过滤数据的速度往往要比只查看数据包物理地址的交换机慢。但是，对于那些结构复杂的网络，使用路由器可以提高网络的整体效率。路由器的另外一个明显优势就是可以自动过滤网络广播。

集线器与路由器在功能上有什么不同？

首先说HUB,也就是集线器。它的作用可以简单的理解为将一些机器连接起来组成一个局域网。而交换机（又名交换式集线器）作用与集线器大体相同。但是两者在性能上有区别：集线器采用的是共享带宽的工作方式，而交换机是独享带宽。这样在机器很多或数据量很大时，两者将会有比较明显的区别。而路由器与以上两者有明显区别，它的作用在于连接不同的网段并且找到网络中数据传输最合适的路径。路由器是产生于交换机之后，就像交换机产生于集线器之后，所以路由器与交换机也有一定联系，不是完全独立的两种设备。路由器主要克服了交换机不能路由转发数据包的不足。

总的来说，路由器与交换机的主要区别体现在以下几个方面：

#### （1）工作层次不同

最初的交换机是工作在数据链路层，而路由器一开始就设计工作在网络层。由于交换机工作在数据链路层，所以它的工作原理比较简单，而路由器工作在网络层，可以得到更多的协议信息，路由器可以做出更加智能的转发决策。

#### （2）数据转发所依据的对象不同

交换机是利用物理地址或者说MAC地址来确定转发数据的目的地址。而路由器则是利用IP地址来确定数据转发的地址。IP地址是在软件中实现的，描述的是设备所在的网络。MAC地址通常是硬件自带的，由网卡生产商来

分配的，而且已经固化到了网卡中去，一般来说是不可更改的。而IP地址则通常由网络管理员或系统自动分配。

(3) 传统的交换机只能分割冲突域，不能分割广播域；而路由器可以分割广播域

由交换机连接的网段仍属于同一个广播域，广播数据包会在交换机连接的所有网段上传播，在某些情况下会导致通信拥挤和安全漏洞。连接到路由器上的网段会被分配成不同的广播域，广播数据不会穿过路由器。虽然第三层以上交换机具有VLAN功能，也可以分割广播域，但是各子广播域之间是不能通信交流的，它们之间的交流仍然需要路由器。

(4) 路由器提供了防火墙的服务

路由器仅仅转发特定地址的数据包，不传送不支持路由协议的数据包传送和未知目标网络数据包的传送，从而可以防止广播风暴。

## 物理层

在OSI参考模型中，物理层（Physical Layer）是参考模型的最低层，也是OSI模型的第一层。

物理层的主要功能是：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。

物理层的作用是实现相邻计算机节点之间比特流的透明传送，尽可能屏蔽掉具体传输介质和物理设备的差异。使其上面的数据链路层不必考虑网络的具体传输介质是什么。“透明传送比特流”表示经实际电路传送后的比特流没有发生变化，对传送的比特流来说，这个电路好像是看不见的。

## 数据链路层

数据链路层（Data Link Layer）是OSI模型的第二层，负责建立和管理节点间的链路。该层的主要功能是：通过各种控制协议，将有差错的物理信道变为无差错的、能可靠传输数据帧的数据链路。

在计算机网络中由于各种干扰的存在，物理链路是不可靠的。因此，这一层的主要功能是在物理层提供的比特流的基础上，通过差错控制、流量控制方法，使有差错的物理线路变为无差错的数据链路，即提供可靠的通过物理介质传输数据的方法。

该层通常又被分为介质访问控制（MAC）和逻辑链路控制（LLC）两个子层。

MAC子层的主要任务是解决共享型网络中多用户对信道竞争的问题，完成网络介质的访问控制；

LLC子层的主要任务是建立和维护网络连接，执行差错校验、流量控制和链路控制。

数据链路层的具体工作是接收来自物理层的位流形式的数据，并封装成帧，传送到上一层；同样，也将来自上层的数据帧，拆装为位流形式的数据转发到物理层；并且，还负责处理接收端发回的确认帧的信息，以便提供可靠的数据传输。

## 网络层

网络层（Network Layer）是OSI模型的第三层，它是OSI参考模型中最复杂的一层，也是通信子网的最高一层。它在下两层的基础上向资源子网提供服务。其主要任务是：通过路由选择算法，为报文或分组通过通信子网选择最适当的路径。该层控制数据链路层与传输层之间的信息转发，建立、维持和终止网络的连接。具体地说，数据链路层的数据在这一层被转换为数据包，然后通过路径选择、分段组合、顺序、进/出路由等控制，将信息从一个网络设备传送到另一个网络设备。

一般地，数据链路层是解决同一网络内节点之间的通信，而网络层主要解决不同子网间的通信。例如在广域网之间通信时，必然会遇到路由（即两节点间可能有多条路径）选择问题。

在实现网络层功能时，需要解决的主要问题如下：

寻址：数据链路层中使用的物理地址（如MAC地址）仅解决网络内部的寻址问题。在不同子网之间通信时，为了识别和找到网络中的设备，每一子网中的设备都会被分配一个唯一的地址。由于各子网使用的物理技术可能不同，因此这个地址应当是逻辑地址（如IP地址）。

交换：规定不同的信息交换方式。常见的交换技术有：线路交换技术和存储转发技术，后者又包括报文交换技术和分组交换技术。

路由算法：当源节点和目的节点之间存在多条路径时，本层可以根据路由算法，通过网络为数据分组选择最佳路径，并将信息从最合适的路径由发送端传送到接收端。

连接服务：与数据链路层流量控制不同的是，前者控制的是网络相邻节点间的流量，后者控制的是从源节点到目的节点间的流量。其目的在于防止阻塞，并进行差错检测。

## 传输层

OSI下3层的主要任务是数据通信，上3层的任务是数据处理。而传输层（Transport Layer）是OSI模型的第4层。因此该层是通信子网和资源子网的接口和桥梁，起到承上启下的作用。

该层的主要任务是：向用户提供可靠的端到端的差错和流量控制，保证报



文的正确传输。传输层的作用是向高层屏蔽下层数据通信的细节，即向用户透明地传送报文。该层常见的协议：TCP/IP中的TCP协议、Novell网络中的SPX协议和微软的NetBIOS/NetBEUI协议。

传输层提供会话层和网络层之间的传输服务，这种服务从会话层获得数据，并在必要时，对数据进行分割。然后，传输层将数据传递到网络层，并确保数据能正确无误地传送到网络层。因此，传输层负责提供两节点之间数据的可靠传送，当两节点的联系确定之后，传输层则负责监督工作。

综上，传输层的主要功能如下：

传输连接管理：提供建立、维护和拆除传输连接的功能。传输层在网络层的基础上为高层提供“面向连接”和“面向无连接”的两种服务。

处理传输差错：提供可靠的“面向连接”和不太可靠的“面向无连接”的数据传输服务、差错控制和流量控制。在提供“面向连接”服务时，通过这一层传输的数据将由目标设备确认，如果在指定的时间内未收到确认信息，数据将被重发。

监控服务质量。

## 会话层

会话层（Session Layer）是OSI模型的第5层，是用户应用程序和网络之间的接口，主要任务是：向两个实体的表示层提供建立和使用连接的方法。将不同实体之间的表示层的连接称为会话。因此会话层的任务就是组织和协调两个会话进程之间的通信，并对数据交换进行管理。

用户可以按照半双工、单工和全双工的方式建立会话。当建立会话时，用户必须提供他们想要连接的远程地址。而这些地址与MAC（介质访问控制子层）地址或网络层的逻辑地址不同，它们是为用户专门设计的，更便于用户记忆。域名（DN）就是一种网络上使用的远程地址例如：www.3721.com就是一个域名。会话层的具体功能如下：

会话管理：允许用户在两个实体设备之间建立、维持和终止会话，并支持它们之间的数据交换。例如提供单方向会话或双向同时会话，并管理会话中的发送顺序，以及会话所占用时间的长短。

会话流量控制：提供会话流量控制和交叉会话功能。

寻址：使用远程地址建立会话连接。I

出错控制：从逻辑上讲会话层主要负责数据交换的建立、保持和终止，但实际的工作却是接收来自传输层的数据，并负责纠正错误。会话控制和远程过程调用均属于这一层的功能。但应注意，此层检查的错误不是通信介质的错误，而是磁盘空间、打印机缺纸等类型的高级错误。

## 表示层

表示层（Presentation Layer）是OSI模型的第六层，它对来自应用层的命令和数据进行解释，对各种语法赋予相应的含义，并按照一定的格式传送

给会话层。其主要功能是“处理用户信息的表示问题，如编码、数据格式转换和加密解密”等。表示层的具体功能如下：

数据格式处理：协商和建立数据交换的格式，解决各应用程序之间在数据格式表示上的差异。

数据的编码：处理字符集和数字的转换。例如由于用户程序中的数据类型（整型或实型、有符号或无符号等）、用户标识等都可以有不同的表示方式，因此，在设备之间需要具有在不同字符集或格式之间转换的功能。

压缩和解压缩：为了减少数据的传输量，这一层还负责数据的压缩与恢复。

数据的加密和解密：可以提高网络的安全性。

## 应用层

应用层（Application Layer）是OSI参考模型的最高层，它是计算机用户，以及各种应用程序和网络之间的接口，其功能是直接向用户提供服务，完成用户希望在网络上完成的各种工作。它在其他6层工作的基础上，负责完成网络中应用程序与网络操作系统之间的联系，建立与结束使用者之间的联系，并完成网络用户提出的各种网络服务及应用所需的监督、管理和服务等各种协议。此外，该层还负责协调各个应用程序间的工作。

应用层为用户提供的服务和协议有：文件服务、目录服务、文件传输服务（FTP）、远程登录服务（Telnet）、电子邮件服务（E-mail）、打印服务、安全服务、网络管理服务、数据库服务等。上述的各种网络服务由该层的不同应用协议和程序完成，不同的网络操作系统之间在功能、界面、实现技术、对硬件的支持、安全可靠性以及具有的各种应用程序接口等各个方面的差异是很大的。应用层的主要功能如下：

用户接口：应用层是用户与网络，以及应用程序与网络间的直接接口，使得用户能够与网络进行交互式联系。

实现各种服务：该层具有的各种应用程序可以完成和实现用户请求的各种服务。

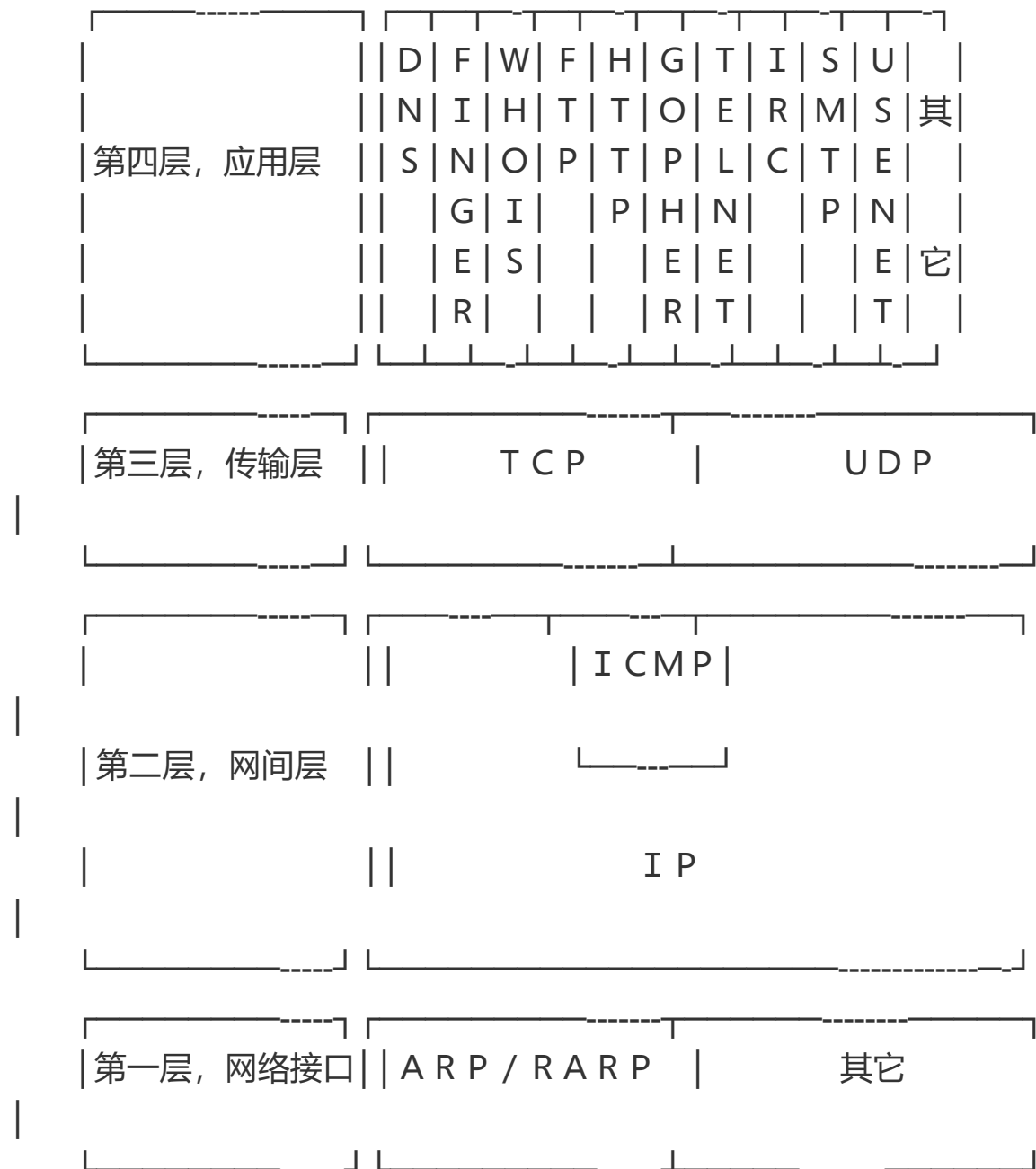
## OSI7层模型的小结

由于OSI是一个理想的模型，因此一般网络系统只涉及其中的几层，很少有系统能够具有所有的7层，并完全遵循它的规定。

在7层模型中，每一层都提供一个特殊的网络功能。从网络功能的角度观察：下面4层（物理层、数据链路层、网络层和传输层）主要提供数据传输和交换功能，即以节点到节点之间的通信为主；第4层作为上下两部分的桥梁，是整个网络体系结构中最关键的部分；而上3层（会话层、表示层和应用层）则以提供用户与应用程序之间的信息和数据处理功能为主。简言

之，下4层主要完成通信子网的功能，上3层主要完成资源子网的功能。

以下是TCP/IP分层模型



TCP/IP四层参考模型

TCP/IP协议被组织成四个概念层，其中有三层对应于ISO参考模型中的相应层。ICP/IP协议族并不包含物理层和数据链路层，因此它不能独立完成整个计算机网络系统的功能，必须与许多其他的协议协同工作。

TCP/IP分层模型的四个协议层分别完成以下的功能：

第一层:网络接口层

包括用于协作IP数据在已有网络介质上传输的协议。实际上TCP/IP标准并不定义与ISO数据链路层和物理层相对应的功能。相反，它定义像地址解析协议(Address Resolution Protocol,ARP)这样的协议，提供TCP/IP协议的数据结构和实际物理硬件之间的接口。

### 第二层:网间层

对应于OSI七层参考模型的网络层。本层包含IP协议、RIP协议(Routing Information Protocol，路由信息协议)，负责数据的包装、寻址和路由。同时还包含网间控制报文协议(Internet Control Message Protocol,ICMP)用来提供网络诊断信息。

### 第三层:传输层

对应于OSI七层参考模型的传输层，它提供两种端到端的通信服务。其中TCP协议(Transmission Control Protocol)提供可靠的数据流运输服务，UDP协议(Use Datagram Protocol)提供不可靠的用户数据报服务。

### 第四层:应用层

对应于OSI七层参考模型的应用层和表达层。因特网的应用层协议包括Finger、Whois、FTP(文件传输协议)、Gopher、HTTP(超文本传输协议)、Telnet(远程终端协议)、SMTP(简单邮件传送协议)、IRC(因特网中继会话)、NNTP(网络新闻传输协议)等，这也是本书将要讨论的重点。

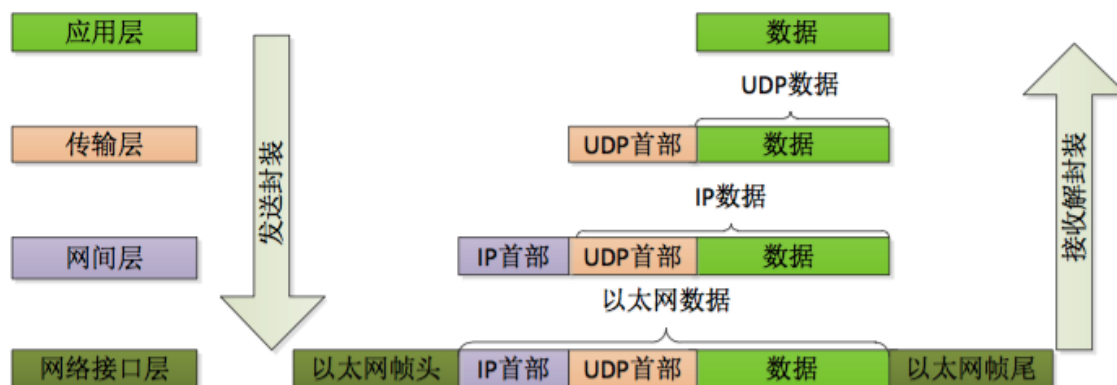
## TCP/IP 协议基本概念

---

OSI 模型所分的七层,在实际应用中,往往有一些层被整合,或者功能分散到其他层去。TCP/IP 没有照搬 OSI 模型,也没有一个公认的 TCP/IP 层级模型,一般划分为三层到五层模型来描述 TCP/IP 协议。

- 在此描述用一个通用的四层模型来描述,每一层都和 OSI 模型有较强的相关性但是又可能会有交叉。
- TCP/IP 的设计,是吸取了分层模型的精华思想——封装。每层对上一层提供服务的时候,上一层的数据结构是黑盒,直接作为本层的数据,而不需要关心上一层协议的任何细节。

TCP/IP 分层模型的分层以以太网上传输 UDP 数据包如图所示;



UDP 数据包.png

### 数据包

宽泛意义的数据包:每一个数据包都包含"标头"和"数据"两个部分."标头"包含本数据包的一些说明."数据"则是本数据包的内容.

细分数据包:

- 应用程序数据包: 标头部分规定应用程序的数据格式.数据部分传输具体的数据内容.\*\*\* ——对应上图中的数据! \*\*\*
- TCP/UDP数据包:标头部分包含双方的发出端口和接收端口. UDP数据包:'标头'长度:8个字节,"数据包"总长度最大为65535字节,正好放进一个IP数据包. TCP数据包:理论上没有长度限制,但是,为了保证网络传输效率,通常不会超过IP数据长度,确保单个包不会被分割. \*\*\* ——对应上图中的UDP数据! \*\*\*
- IP数据包: 标头部分包含通信双方的IP地址,协议版本,长度等信息. '标头'长度:20~60字节,"数据包"总长度最大为65535字节. \*\*\* ——对应上图中的IP数据\*\*\*
- 以太网数据包: 最基础的数据包.标头部分包含了通信双方的MAC地址,数据类型等. '标头'长度:18字节,'数据'部分长度:46~1500字节. \*\*\* ——对应上图中的以太网数据\*\*\*

### 四层模型

#### 1. 网络接口层

网络接口层包括用于协作IP数据在已有网络介质上传输的协议。

它定义像地址解析协议(Address Resolution Protocol,ARP)这样的协

议,□供 TCP/IP 协议的数据结构和实际物理硬件之间的接口。

可以理解为: 确定了网络数据包的形式。

## 2. 网间层

网间层对应于 OSI 七层参考模型的网络层, 本层包含 IP 协议、RIP 协议(Routing Information Protocol,路由信息协议),负责数据的包装、寻址和路由。同时还包含网间控制报文协议 (Internet Control Message Protocol,ICMP)用来□供网络诊断信息;

可以理解为: 该层时确定计算机的位置。

## 3. 传输层

传输层对应于 OSI 七层参考模型的传输层,它□供两种端到端的通信服务。其中 TCP 协议(Transmission Control Protocol)□供可靠的数据流运输服务,UDP 协议(Use Datagram Protocol)□供不可靠的用户数据报服务。

TCP:三次握手、四次挥手;UDP:只发不管别人收不收得到——任性哈

## 4. 应用层

应用层对应于 OSI 七层参考模型的应用层和表达层;

不明白的再看看7层参考模型的描述。

### TCP/IP 协议族常用协议

- 应用层: TFTP, HTTP, SNMP, FTP, SMTP, DNS, Telnet 等等
- 传输层: TCP, UDP
- 网络层: IP, ICMP, OSPF, EIGRP, IGMP
- 数据链路层: SLIP, CSLIP, PPP, MTU

重要的 TCP/IP 协议族协议进行简单介绍:

- IP(Internet Protocol,网际协议)是网间层的主要协议,任务是在源地址和目的地址之间传输数据。IP 协议只是尽最大努力来传输数据包,并不保证所有的包都可以传输 到目的地,也不保证数据包的顺序和唯一。
  - IP 定义了 TCP/IP 的地址,寻址方法,以及路由规则。现在广泛使用的 IP 协议有 IPv4 和 IPv6 两种:IPv4 使用 32 位二进制整数做地址,一般使用点分十进制方式表示,比如 192.168.0.1。
  - IP 地址由两部分组成,即网络号和主机号。故一个完整的 IPv4 地址往往表示为 192.168.0.1/24 或192.168.0.1/255.255.255.0 这种形式。

- IPv6 是为了解决 IPv4 地址耗尽和其它一些问题而研发的最新版本的 IP。使用 128 位 整数表示地址,通常使用冒号分隔的十六进制来表示,并且可以省略其中一串连续的 0,如:fe80::200:1ff:fe00:1。目前使用并不多!
- ICMP(Internet Control Message Protocol, 网络控制消息协议)是 TCP/IP 的核心协议之一,用于在 IP 网络中发送控制消息,□供通信过程中的各种问题反馈。ICMP 直接使用 IP 数据包传输,但 ICMP 并不被视为 IP 协议的子协议。常见的联网状态诊断工具比如依赖于 ICMP 协议;
- TCP(TransmissionControlProtocol,传输控制协议)是一种面向连接的,可靠的, 基于字节流传输的通信协议。TCP 具有端口号的概念,用来标识同一个地址上的不同应用。□述 TCP 的标准文档是 RFC793。
- UDP(UserDatagramProtocol,用户数据报协议)是一个面向数据报的传输层协议。UDP 的传输是不可靠的,简单的说就是发了不管,发送者不会知道目标地址 的数据通路是否发生拥塞,也不知道数据是否到达,是否完整以及是否还是原来的 次序。它同 TCP 一样有用来标识本地应用的端口号。所以应用 UDP 的应用,都能够容忍一定数量的错误和丢包,但是对传输性能敏感的,比如流媒体、DNS 等。
- ECHO(EchoProtocol,回声协议)是一个简单的调试和检测工具。服务器会 原样回发它收到的任何数据,既可以使用 TCP 传输,也可以使用 UDP 传输。使用 端口号 7。
- DHCP(DynamicHostConfigurationProtocol,动态主机配置协议)是用于局域网自动分配 IP 地址和主机配置的协议。可以使局域网的部署更加简单。
- DNS(DomainNameSystem,域名系统)是互联网的一项服务,可以简单的将用“.” 分隔的一般会有意义的域名转换成不易记忆的 IP 地址。一般使用 UDP 协议传输, 也可以使用 TCP,默认服务端口号 53。□
- FTP(FileTransferProtocol,文件传输协议)是用来进行文件传输的标准协议。FTP 基于 TCP 使用端口号 20 来传输数据,21 来传输控制信息。
- TFTP(Trivial File Transfer Protocol,简单文件传输协议)是一个简化的文件传输协议,其设计非常简单,通过少量存储器就能轻松实现,所以一般被用来通 过网络引导计算机过程中传输引导文件等小文件;

- SSH(SecureShell,安全Shell),因为传统的网络服务程序比如TELNET本质上都极不安全,明文传输数据和用户信息包括密码,SSH 被开发出来避免这些问题,它其实是一个协议框架,有大量的扩展冗余能力,并且提供了加密压缩的通道可以为其他协议使用。
- POP(PostOfficeProtocol,邮局协议)是支持通过客户端访问电子邮件的服务,现在版本是 POP3,也有加密的版本 POP3S。协议使用 TCP,端口 110。
- SMTP(Simple Mail Transfer Protocol,简单邮件传输协议)是现在在互联网上发送电子邮件的事实标准。使用 TCP 协议传输,端口号 25。
- HTTP(HyperTextTransferProtocol,超文本传输协议)是现在广为流行的WEB 网络的基础,HTTPS 是 HTTP 的加密安全版本。协议通过 TCP 传输,HTTP 默认使用端口 80,HTTPS 使用 443。

全文完!

---



网络七层协议的通俗理解

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，RIP，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	为数据选择路由	IP，ICMP，OSPF，BGP，IGMP，ARP，RARP
数据链路层	传输有地址的帧以及错误检测功能	SLIP，CSLIP，PPP，MTU，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，IEEE802.2

OSI七层模式简单通俗理解

这个模型学了好多次，总是记不住。今天又看了一遍，发现用历史推演的角度去看问题会更有逻辑，更好记。本文不一定严谨，可能有错漏，主要是抛砖引玉，帮助记性不好的人。总体来说，OSI模型是从底层往上层发展出来的。

这个模型推出的最开始，是因为美国人有两台机器之间进行通信的需求。

需求1：

科学家要解决的第一个问题是，两个硬件之间怎么通信。具体就是一台发些比特流，然后另一台能收到。

于是，科学家发明了物理层：

主要定义物理设备标准，如网线的接口类型、光纤的接口类型、各种传输介质的传输速率等。它的主要作用是传输比特流(就是由1、0转化为电流强弱来进行传输，到达目的地后在转化为1、0，也就是我们常说的数模转换与模数转换)。这一层的数据叫做比特。

需求2：

现在通过电线我能发数据流了，但是，我还希望通过无线电波，通过其它介质来传输。然后我还要保证传输过去的比特流是正确的，要有纠错功能。

于是，发明了数据链路层：

定义了如何让格式化数据以进行传输，以及如何让控制对物理介质的访问。这一层通常还提供错误检测和纠正，以确保数据的可靠传输。

需求3：

现在我能发正确的发比特流数据到另一台计算机了，但是当我发大量数据时候，可能需要好长时间，例如一个视频格式的，网络会中断好多次（事实上，即使有了物理层和数据链路层，网络还是经常中断，只是中断的时间是毫秒级别的）。

那么，我还须要保证传输大量文件时的准确性。于是，我要对发出去的数据进行封装。就像发快递一样，一个个地发。

于是，先发明了传输层（传输层在OSI模型中，是在网络层上面）

例如TCP，是用于发大量数据的，我发了1万个包出去，另一台电脑就要告诉我是否接受到了1万个包，如果缺了3个包，就告诉我是第1001，234，8888个包丢了，那我再发一次。这样，就能保证对方把这个视频完整接收了。

例如UDP，是用于发送少量数据的。我发20个包出去，一般不会丢包，所以，我不管你收到多少个。在多人互动游戏，也经常用UDP协议，因为一般都是简单的信息，而且有广播的需求。如果用TCP，效率就很低，因为它会不停地告诉主机我收到了20个包，或者我收到了18个包，再发我两个！如果同时有1万台计算机都这样做，那么用TCP反而会降低效率，还不如用UDP，主机发出去就算了，丢几个包你就卡一下，算了，下次再发包你再更新。

TCP协议是会绑定IP和端口的协议，下面会介绍IP协议。

#### 需求4：

传输层只是解决了打包的问题。但是如果我有多台计算机，怎么找到我要发的那台？或者，A要给F发信息，中间要经过B，C，D,E，但是中间还有好多节点如K.J.Z.Y。我怎么选择最佳路径？这就是路由要做的事。

于是，发明了**网络层**。即路由器，交换价那些具有寻址功能的设备所实现的功能。这一层定义的是IP地址，通过IP地址寻址。所以产生了IP协议。

#### 需求5：

现在我们已经保证给正确的计算机，发送正确的封装过后的信息了。但是用户级别的体验好不好？难道我每次都要调用TCP去打包，然后调用IP协议去找路由，自己去发？当然不行，所以我们要建立一个自动收发包，自动寻址的功能。

于是，发明了**会话层**。会话层的作用就是建立和管理应用程序之间的通信。

#### 需求6：

现在我能保证应用程序自动收发包和寻址了。但是我要用Linux给window发包，两个系统语法不一致，就像安装包一样，exe是不能在linux下用的，shell在window下也是不能直接运行的。于是需要**表示层**（presentation），帮我们解决不同系统之间的通信语法问题。

#### 需求7：

OK，现在所有必要条件都准备好了，我们可以写个android程序，web程序去实现需求把。

#### 补充：

#### Socket：

这不是一个协议，而是一个通信模型。其实它最初是伯克利加州分校软件研究所，简称BSD发明的，主要用来一台电脑的两个进程间通信，然后把它用到了两台电脑的进程间通信。所以，可以把它简单理解为进程间通信，不是什么高级的东西。主要做的事情不就是：

A发包：发请求包给某个已经绑定的端口（所以我们经常会访问这样的地址182.13.15.16:1235，1235就是端口）；收到B的允许；然后正式发送；发送完了，告诉B要断开链接；收到断开允许，马上断开，然后发送已经断开信息给B。

B收包：绑定端口和IP；然后在这个端口监听；接收到A的请求，发允许给A，并做好接收准备，主要就是清理缓存等待接收新数据；然后正式接收；接受到断开请求，允许断开；确认断开后，继续监听其它请求。

可见，Socket其实就是I/O操作。Socket并不仅限于网络通信。在网络通信中，它涵盖了网络层、传输层、会话层、表示层、应用层——其实这都不需要记，因为Socket通信时候用到了IP和端口，仅这两个就表明了它用到了网络层和传输层；而且它无视多台电脑通信的系统差别，所以它涉及了表示层；一般Socket都是基于一个应用程序的，所以会涉及到会话层和应用层。

---