

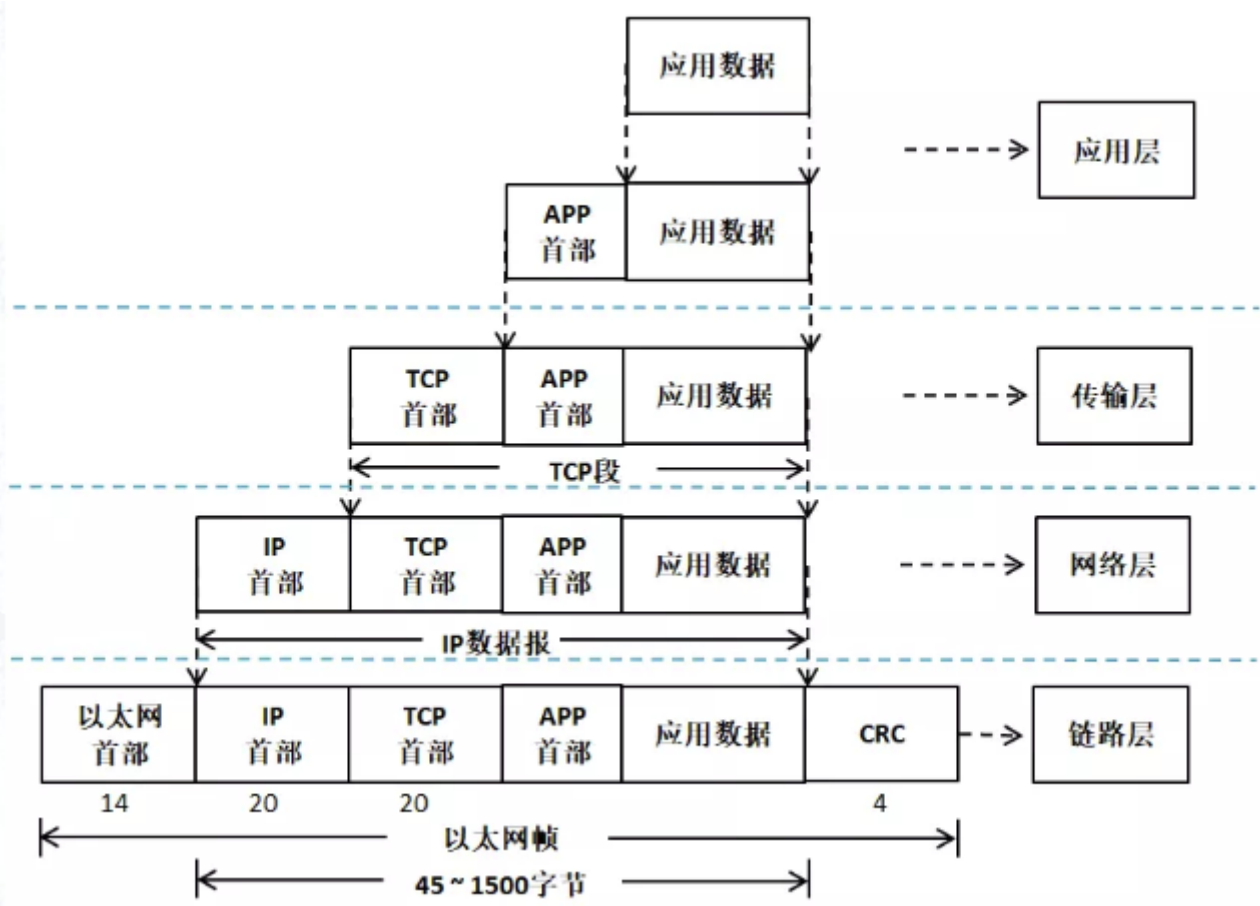
title: 网络底层知识 author: 杰杰 top: false cover: false toc: true mathjax: false date: 2019-10-15 22:56:12 img: coverImg: password: summary: tags: - TCP/IP - 网络 - LwIP categories: TCP/IP

前言

从一篇文章，我们也知道大概的网络相关知识，网络这个系列文章就从底层往顶层讲解，本篇文章就来接触一下网络底层的相关知识。

概述网络中的数据递交

当用户发送数据时，将数据向下交给传输层，这是处于应用层的操作，应用层可以通过调用传输层的接口来编写特定的应用程序，并且TCP/IP协议栈一般也会包含一些简单的应用协议如Telnet 远程登录、FTP 文件传输、SMTP 邮件传输协议等，这些协议有自己的首部——APP首部。传输层会在接收到上层协议的数据前面加上传输层首部（此处以TCP协议为例，图的传输层首部为TCP首部。当然传输层协议还有UDP协议），传输层会向下交给网络层。同样地，网络层会在数据前面加上网络层首部（如IP首部），然后网络层再将数据向下交给链路层，数据链路层会对数据进行最后一次封装，即在数据前面加上数据链路层首部（此处使用以太网接口为例），然后将数据交给网卡。最后，网卡将数据转换成物理链路上的电平信号，数据就这样被发送到了网络中。数据的发送过程，可以概括为TCP/IP的各层协议对数据进行封装的过程；当设备的网卡接收到某个数据包后，它会将其放置在网卡的接收缓存中，并告知TCP/IP协议栈内核。然后TCP/IP协议栈内核就开始工作了，它会将数据包从接收缓存中取出，并逐层解析数据包中的协议首部信息，并最终将数据交给某个应用程序。数据的接收过程与发送过程正好相反，可以概括为TCP/IP的各层协议对数据进行解析的过程。



链路层

以太网

关于以太网的理论我就不介绍了，因为这些理论太多了，有兴趣可以自己去网上查看。

链路层与数据链路

我们在这里要明确一下，“链路”和“数据链路”并不是一回事。所谓**链路**就是从 一个结点到相邻结点的一段物理线路（有线或无线）这是实实在在看得见的，比如我们的网线，而中间没有任何其他的交换结点。在进行数据通信时，两台计算机之间的通信路径往往要经过许多段这样的链路。

数据链路则是另一个概念。这是因为当需要在一条线路上传送数据时，除了**必须**有一条物理线路外，还必须有一些必要的协议来控制这些数据的传输，这就是我们**TCP/IP**协议栈要做的事情。把这些**TCP/IP**协议栈加到链路上，就构成了数据链路。一般我们所说的**链路层**指的都是**数据链路**。下文均采用**链路层**表示。

LwIP中的“链路层”

我们知道，网络接口（如以太网接口）是硬件接口，（提示：网络接口又可以称之为网卡，为了统一，下文均采用网卡表示网络接口），LwIP是软件，那么**怎么让硬件与软件无缝连接起来呢**？而且，网卡又有多种多样，怎么能让**LwIP**使用同样的软件程序能兼容不同的硬件呢？

因此**LwIP**使用一个数据结构——**netif**来描述一个网卡，由于网卡是直接和硬件打交道的，硬件不同则处理基本是不同的，比如网卡的有各种各样的芯片，所以必须由**我们自己**提供最底层接口函数，**LwIP**提供统一的接口，比如网卡的初始化，网卡的收发数据，当**LwIP**底层得到了网络的数据之后，才会经过层层传入内核中去处理；同理，当应用层需要发送一个数据包的时候，也是讲数据包层层往下递交，然后也需要调用网卡的**发送函数**，将数据发送到网络中，这样子才能把数据从硬件接口到软件内核无缝连接起来。因此**LwIP**移植的关键是底层的移植，而很多人就卡在这里，如果底层的处理不好，那么就会导致各种各样的问题，比如网卡无缝接收太多数据，导致网速过慢，出现挂掉的现象，时常丢包，或者运行一段时间就导致内存泄漏等等等乱七八糟的问题。

简单来说，**netif**是**LwIP**抽象出来的网卡描述符，它拥有描述设备中的不同网卡，一个设备至少与一个网卡，才能与网络进行通讯，**LwIP**协议栈可以使用多个不同的接口，而源码中的**ethernetif.c**文件则提供了**netif**访问各种不同的网卡，每个网卡有不同的实现方式，我们呢一般只需要修改**ethernetif.c**文件就行了，但是本篇文章不讲移植。（此处用**LwIP**协议栈讲解，本系列教程均基于**LwIP**）

举个例子：我们可以理解将整个网络的数据传输理解为物流，那么网卡就是不同的运输工具，我们可以选择汽车、飞机、轮船等运输工具，不同的运输工具速度是不一样的，但是对于一个物流公司而言，可能同时存在很多种运输的工具，这就需要物流公司去记录这些运输工具，当有一个包裹需要通过飞机运输出去，那么物流公司就会将这个包裹通过飞机发送出去，这就好比我们的网卡，需要哪个网卡发送或者接收网络数据的时候，就会让对应的网卡去工作。（可能不太贴切）

MAC地址简介

MAC地址（英语：Media Access Control Address），直译为媒体访问控制地址，也称为局域网地址（LAN Address），以太网地址（Ethernet Address）或物理地址（Physical Address），它是一个用来确认网络设备位置的地址。在OSI模型中，第三层网络层负责IP地址，第二层数据链路层则负责MAC地

址。MAC地址用于在网络中唯一标示一个网卡，一台设备若有一或多个网卡，则每个网卡都需要并会有一个唯一的MAC地址。（引用来自维基百科）

可能我们做计算机相关专业的人，多多少少都会知道IP地址、MAC地址（也称之为链路层地址）。在前一篇文章说了，想要不同主机之间进行通讯，那么就必须有IP地址，那现在为什么我们在链路层都需要地址呢？事实上，并不是主机或路由器具有链路层地址，而是它们的网卡具有链路层地址。因此，一个主机拥有多少个网卡，那么它就有多少个MAC地址。MAC地址实际上就是网卡地址或网卡标识符。当某台计算机使用某块网卡后，网卡上的标识符就成为该计算机的MAC地址。MAC地址长度为6字节（48bit），其前3个字节表示组织唯一标志符（Organizationally Unique Identifier，即OUI），由IEEE的注册管理机构给不同厂家分配的代码，以区分不同的厂家，后3个字节由厂家自行分配，称为扩展标识符。同一个厂家生产的网卡中MAC地址后24位是不同的。

IP地址与MAC地址的关系

TCP/IP协议有自己的IP地址，IP地址（IPv4）是一个32位的IP地址，网络层发送数据包只需要知道目标主机IP地址即可，而以网卡发送数据则必须知道对方的硬件MAC地址，同时IP地址的分配与硬件MAC地址是没有关系的，为了让网络层只需要知道IP地址就可以完成通信工作，那就需要有一个协议将IP地址映射成为对应的MAC地址，此外还需要考虑IP地址可能是动态的，非常灵活，使用静态的映射方法是行不通的，所以ARP协议就提供优质的地址动态解析服务，让32位的IP地址能映射成为48位的MAC地址，让上层应用与底层完全分离开，这样子在上层应用就能灵活使用IP地址作为标识，进行通信。

简单来说，就是计算机中会维护一个ARP缓存表，这个表记录着IP地址与MAC地址的映射关系，我们可以通过在电脑的控制台通过arp -a指令查看一下我们自己计算机的ARP缓存表：

```
C:\Users\Administrator>arp -a

接口: 192.168.0.181 --- 0x4
Internet 地址      物理地址      类型
192.168.0.1        a0-c5-f2-b1-5c-7a 动态
192.168.0.119      94-77-2b-86-6a-78 动态
192.168.0.120      c4-36-55-de-3c-86 动态
192.168.0.122      02-00-00-00-00-00 动态
192.168.0.125      00-e0-4c-0d-88-10 动态
192.168.0.149      e0-d5-5e-4a-db-2d 动态
192.168.0.165      94-77-2b-66-6a-78 动态
192.168.0.168      94-77-2b-56-6a-7a 动态
192.168.0.173      e0-3f-49-44-3f-0b 动态
192.168.0.175      00-11-32-9f-d5-83 动态
192.168.0.177      e0-d5-5e-b3-97-c9 动态
192.168.0.182      e0-d5-5e-4a-dc-1c 动态
192.168.0.187      50-8f-4c-ff-4e-2e 动态
192.168.0.211      94-77-2b-66-6a-78 动态
192.168.0.223      90-2b-34-af-ae-3d 动态
192.168.0.225      94-77-2b-76-6a-78 动态
192.168.0.255      ff-ff-ff-ff-ff-ff 静态
224.0.0.2          01-00-5e-00-00-02 静态
224.0.0.22         01-00-5e-00-00-16 静态
224.0.0.251        01-00-5e-00-00-fb 静态
224.0.0.252        01-00-5e-00-00-fc 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态
255.255.255.255    ff-ff-ff-ff-ff-ff 静态

C:\Users\Administrator>a_
```

我给大家举个例子，IP地址就相当于你家的地址，MAC就是你的身份证，你的身份证是全国唯一的，这没疑问吧，但你的地址是可以跟别人重叠的（你家人也是住在你家）。当你寄快递的时候，快递员要确认这个快递是从你家寄出去的，是你本人亲自寄出去的，同样的，当你收快递的时候，快递员也是先电话找到你，然后必须是你本人才能拿到快递，再拿回家。（可能也不贴切，将就理解一下）

数据帧

链路层的主体部分是网卡中实现的，在发送的数据帧的时候，协议栈取得由高层传递过来的数据报（注意，数据帧与数据报不是一个东西，数据报是形容网络层及其以上的报文，而数据帧一般形容链路层的数据，是一帧一帧的，也是链路层的服务——数据成帧），在链路层中封装该数据报，也就是填写数据帧的各个字段，然后遵循链路接入协议将该帧数据进行传输；在接收端，控制器接收了整个数据帧，抽取出网络层的数据报，往上层传递。下面来看看以太网帧：

前同步码	帧开始符	目标MAC地址	源MAC地址	类型	数据	CRC
7字节	1字节	6字节	6字节	2字节	46~1500字节	4字节

一看这个以太网帧结构，就觉得它有7个字段，但是事实上，前同步码与帧开始符字段不能算是真正的以太网帧数据结构，他们是在网卡发送的时候自动添加进去的，为了数据的准确传输。

以太网帧以一个7字节的前同步码开始。该前同步码的值都是规定为10101010；而后紧接着一个字节的帧开始符，其值是10101011。前同步码字段的作用是实现物理层帧输入输出的同步，而帧开始符表示着以太网数据帧的开始，剩下的5个字段才是真正的以太网数据帧结构。

目标MAC地址（6字节）：这个字段包含目标网卡的MAC地址，当一个网卡收到一个以太网数据帧，如果该数据帧的目标地址是网卡自身的MAC地址或者是MAC广播地址，它都将该帧的数据字段的内容传递给网络层；如果它收到了具有任何其他MAC地址的帧，则将该数据帧丢弃。源MAC地址（6字节）：自身的MAC地址。类型字段（2字节）：类型字段允许以太网复用多种网络层协议。我们只需要记住主机能够使用除了IP协议以外的其他网络层协议。事实上，一台给定的主机可以支持多种网络层协议，以对不同的应用采用不同的协议。因此，当以太网帧到达网卡中，网卡需要知道它应该将数据字段的内容传递给哪个网络层协议。比如如有IP协议、ARP协议等。

注意了：当这个字段的值小于1518时，它表示后面数据字段的数据长度，当大于1518的时候才表示递交给哪个协议。

数据字段（46~1500字节）：这个字段承载了IP数据报。以太网的最大传输单元（MTU）是1500字节。这意味着如果IP数据报超过了1500字节，则主机必须将该数据报分片（IPv4支持数据报分片，而IPv6不支持）。数据字段的最小长度是46字节，这意味着如果IP数据报小于46字节，数据报必须被填充到46字节。当采用填充时，传递到网络层的数据包括IP数据报和填充部分，网络层使用IP数据报首部中的长度字段来去除填充部分。CRC（4字节）：CRC字段包含了以太网的差错校验信息。补充：所有的以太网都向网络层提供不可靠的无连接服务，在接收到不对的以太网帧时，或者CRC校验不通过时，主机将直接丢弃以太网帧，并不会告知发送的一方。

未完待续..... 下一篇进入协议篇

喜欢就关注我吧！



相关代码可以在公众号后台获取。