

前面说过，网络层的作用就是互连网络，实现各个子网之间的互联，经过前面几篇的铺垫以及数据链路层的学习，我们自然而然地就会问一个问题：**网络层中的数据报文是啥样子的？跟以太网协议中的帧结构一样吗？如果不一样，那么跟帧是什么关系？**

学习了网络层中的报文结构，我们再去学习报文是如何进行路由的，才是正确的学习方向。

## 一、IP协议概述

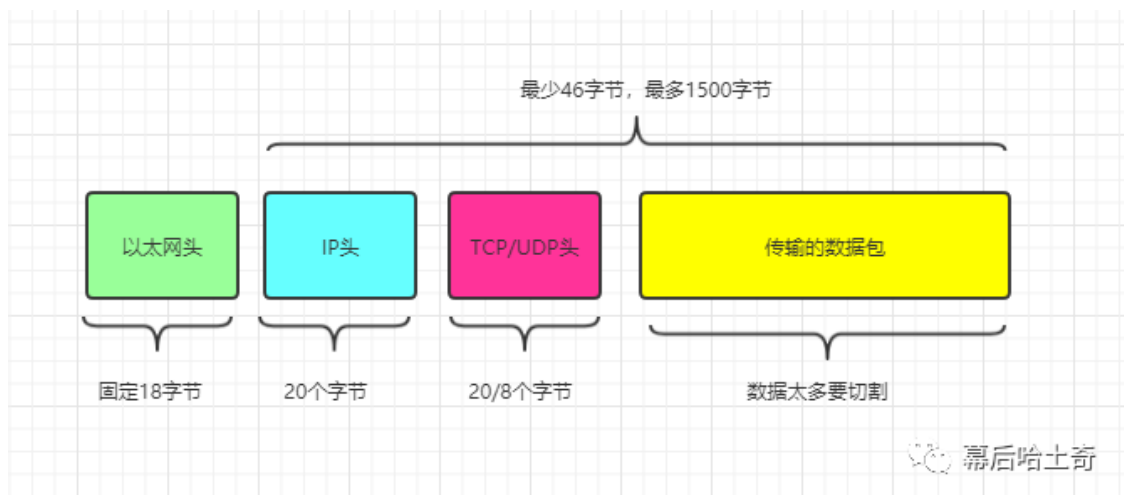
我们之前说过，数据链路层最重要的协议是以太网协议，那么网络层也有一个最重要的协议，那就是伟大的IP协议（IP是Internet Protocol（网际互连协议）的缩写，是TCP/IP体系中的网络层协议）。

**一句话就是：OSI 第 3 层（网络层）的最主要协议就是 IP 协议。**

我们之前学习了以太网协议，涉及以太网帧的结构定义，之前说过，以太网帧除了要发送的消息和CRC校验外，还有三个重要信息：目的MAC地址、源MAC地址、IP协议版本。

同样地，对于 OSI 第 3 层，我们也得确定传输的消息单元中需要包含哪些信息，以及这些信息排列的顺序。

在一头扎进IP报文前，有必要理下完整数据报文的结构，防止混乱：



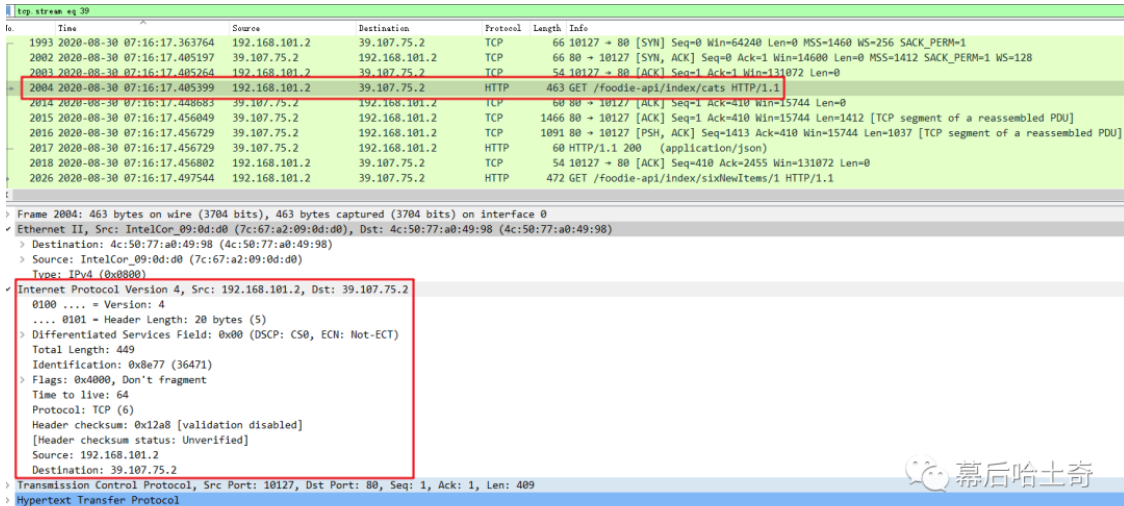
这里就是OSI的封装的概念，一条数据从上到下经过OSI模型传递的时候，会在实际的数据上不停地添加HTTP头、TCP/UDP头、IP头以及以太网的头。我们现在已经学习到IP这一层，里面所谓的数据其实包含了HTTP头、TCP/UDP头、以及实际的数据。

这是为了方便学习，并且我们的网络层其实不关心里面的TCP头是啥，因此等学习到传输层的时候我们再慢慢分解出来。就好像我们学习以太网协议的时候，只关心了以太网的头，具体数据一带而过。**那么现在来到了网络层，我们就需要把以太网协议中的数据再做一次细化，着重把我们网络层感兴趣的IP头拿出来分析。**

好了，下面的文章实际上关心的就是上图说的IP头部分里面有啥。

## 二、小试牛刀

我们来实际抓个包看看，我找了一个HTTP的网站，用 Wireshark 抓了一个包，我点开了一个HTTP协议的报文，Wireshark是常用的抓包软件，后续将单独开辟一篇来介绍如何使用，抓包结果如图所示：



No.	Time	Source	Destination	Protocol	Length	Info
1993	2020-08-30 07:16:17.363764	192.168.101.2	39.107.75.2	TCP	66	10127 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2002	2020-08-30 07:16:17.405197	39.107.75.2	192.168.101.2	TCP	66	80 → 10127 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1412 SACK_PERM=1 WS=128
2003	2020-08-30 07:16:17.405264	192.168.101.2	39.107.75.2	TCP	54	10127 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
2004	2020-08-30 07:16:17.405399	192.168.101.2	39.107.75.2	HTTP	463	GET /foodie-api/index/cats HTTP/1.1
2014	2020-08-30 07:16:17.448883	39.107.75.2	192.168.101.2	TCP	60	80 → 10127 [ACK] Seq=1 Ack=410 Win=15744 Len=0
2015	2020-08-30 07:16:17.456049	39.107.75.2	192.168.101.2	TCP	1466	80 → 10127 [ACK] Seq=1 Ack=410 Win=15744 Len=1412 [TCP segment of a reassembled PDU]
2016	2020-08-30 07:16:17.456729	39.107.75.2	192.168.101.2	TCP	1091	80 → 10127 [PSH, ACK] Seq=1413 Ack=410 Win=15744 Len=1037 [TCP segment of a reassembled PDU]
2017	2020-08-30 07:16:17.456729	39.107.75.2	192.168.101.2	HTTP	60	HTTP/1.1 200 (application/json)
2018	2020-08-30 07:16:17.456802	192.168.101.2	39.107.75.2	TCP	54	10127 → 80 [ACK] Seq=410 Ack=2455 Win=131072 Len=0
2026	2020-08-30 07:16:17.497544	192.168.101.2	39.107.75.2	HTTP	472	GET /foodie-api/index/sixNewItems/1 HTTP/1.1

Frame 2004: 463 bytes on wire (3704 bits), 463 bytes captured (3704 bits) on interface 0

Ethernet II, Src: IntelCor\_09:0d:d0 (7c:67:a2:09:0d:d0), Dst: 4c:50:77:a0:49:98 (4c:50:77:a0:49:98)

Destination: 4c:50:77:a0:49:98 (4c:50:77:a0:49:98)

Source: IntelCor\_09:0d:d0 (7c:67:a2:09:0d:d0)

Type: IPv4 (0x800)

Internet Protocol Version 4, Src: 192.168.101.2, Dst: 39.107.75.2

.... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 449

Identification: 0x8e77 (36471)

Flags: 0x0000, Don't fragment

Time to live: 64

Protocol: TCP (6)

Header checksum: 0x12a8 [validation disabled]

[Header checksum status: Unverified]

Source: 192.168.101.2

Destination: 39.107.75.2

Transmission Control Protocol, Src Port: 10127, Dst Port: 80, Seq: 1, Ack: 1, Len: 409

Hypertext Transfer Protocol

我们在下面可以看到各层的信息，比如第一层物理层：

1. Frame 2004，表示帧的编号是2004
2. 该帧的大小为464字节，即3704个bits

第二层的数据链路层中包含：

1. 目标MAC地址
2. 源MAC地址
3. IP协议

可以看到第三层包含的内容比以第二层多多了，OSI第二层消息单元被称为以太网帧或者简称帧，有固定的格式；我们的第三层也是如此，也有固定的格式，不过消息单元被称为数据报（datagram）/数据包（packet）。

我们先挑个简单的来看看，最后两个是源IP和目的地IP。这个跟以太网的MAC地址有点像，很好理解。

思考一个小问题：为了向一台机器发送消息，是否需要知道它的子网掩码？

答案是不需要的，因为我们获取到了对方的IP之后，结合我们自己的子网掩码就很容易地知道它跟我们这台机器在不在同一个网络中，在的话直接用第2层协议来通信，不在则需要借助第3层来通信。

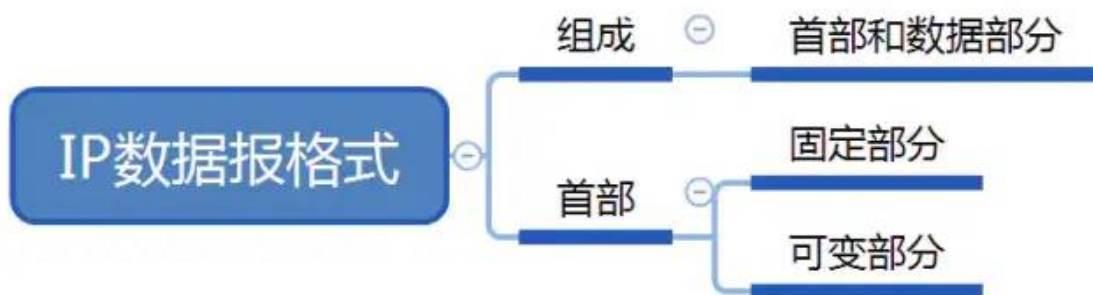
第二个小问题：源IP在前面，目的IP在后面，这顺序跟第二层相反，有搞错吗？换句话说，这里需不需要将目的IP放在前面？

不需要，因为当我们数据传送到对方网络后，OSI第二层就可以根据MAC地址判断谁是接收方电脑了，所以第2层将目的MAC放在前面。那么既然已经确定了谁是接收者，那么IP就不是首要确定的信息了，所以OSI第3层考虑把其他更重要的信息放在了前面，而把IP的信息放在了末尾。

好了，简单的搞定了，意味着热身运动结束了，下面就开始看看报文其他元素的含义吧。

### 三、IP数据报首部结构整体说明

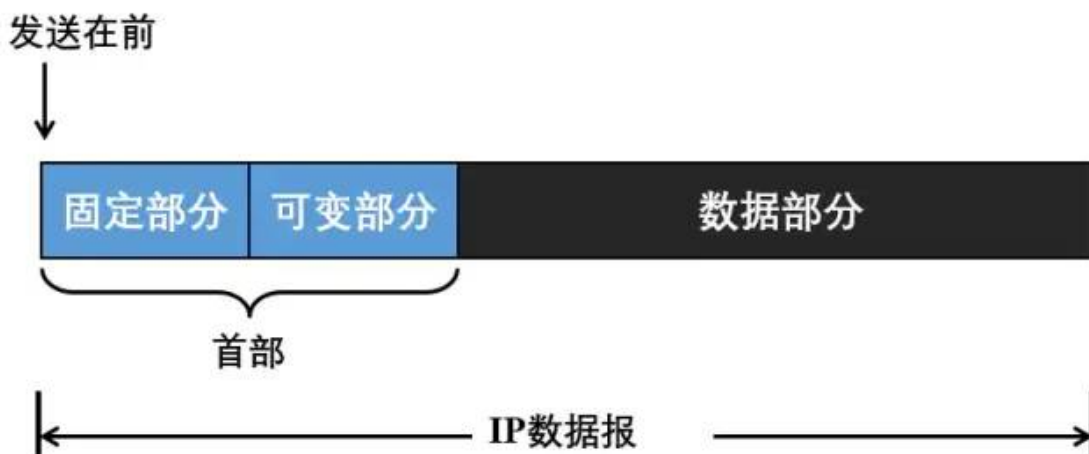
跟以太网有18个字节的帧头一样，IP报文也有它自己的头，称为首部，以及数据部分：



一个IP数据报由首部和数据两个部分组成。

其中首部的第一个部分是固定部分，长度固定共20字节，这是所有IP数据报必须具有的。后一部分是可变部分，其长度是可变的，不是必须的，最小是0字节，最大是40字节。

所以IP的报文格式是这样：



## 四、IP数据报首部固定部分

那么下面重点当然就落在了首部的格式上了，并且重点是固定部分：



(1) **版本**：占4位，指IP协议的版本。目前广泛使用的IP协议版本有两种IPv4和IPv6。

(2) **首部长度**：占4位，该字段的取值以4字节为单位。所以首部长度必须是4B的整数倍。如首部长度字段的4个二进制位分别是1111（对应十进制是15），则IP协议首部的长度是 $15 \times 4B = 60B$ （字节），表示IP数据报首部包含20字节的固定部分和最大40字节可变部分。由于IP数据报首部的固定部分长度固定是20，所以首部字段最小从0101开始。

(3) **区分服务**：占8位，一般情况下不使用该字段。只有使用区分服务时，这个字段才起作用，如要求当前的数据报设置高优先级优先发送。

(4) **总长度**：占16位，表示首部和数据部分长度之和，单位是字节。

(5) **标识、标志、片偏移**是关于IP数据报分片的，下面详细说下。

(6) **生存时间**：占8位，表示数据报在网络中的寿命。由发送数据报的源点设置这个字段，其目的是为了防止那些无法交付的数据报无限制的在互联网中兜圈子（例如从路由器R1转发到R2，再转发到R3，然后又转发到R1），因而白白浪费网络资源。数据报每经过一个路由器，这个值就会减1，当减至0时，就丢弃该数据报。window系统默认为128。发送 **ICMP** 回显应答时经常把 **TTL** 设为最大值 255。


(7) **协议**：占8位，协议字段是指出次数据报所携带的数据是使用的协议。这里记两个协议字段的值：6表示TCP协议，17表示UDP协议。

(8) **首部校验和**：占16位，只校验数据报的首部，不检验数据部分。数据报每经过一个路由器都要重新计算一下首部校验和（一些字段，如生存时间、标志、片偏移可能发生了变化）。

(9) **源地址和目的地址**：各占32位。这个上面已解释。

这样，就可以理解实际抓包中网络层各行的含义了，请读者仔细对比看一遍，这里再将抓包图贴一下。

tcp_stream_03_30						
Time	Source	Destination	Protocol	Length	Info	
1993 2020-08-30 07:16:17.363764	192.168.101.2	39.107.75.2	TCP	66	10127 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1	
2002 2020-08-30 07:16:17.405197	39.107.75.2	192.168.101.2	TCP	66	80 → 10127 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1412 SACK_PERM=1 WS=128	
2003 2020-08-30 07:16:17.405264	192.168.101.2	39.107.75.2	TCP	54	10127 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0	
2004 2020-08-30 07:16:17.405399	192.168.101.2	39.107.75.2	HTTP	463	GET /foodie-api/index/cats HTTP/1.1	
2014 2020-08-30 07:16:17.448683	39.107.75.2	192.168.101.2	TCP	60	80 → 10127 [ACK] Seq=1 Ack=410 Win=15744 Len=0	
2015 2020-08-30 07:16:17.456049	39.107.75.2	192.168.101.2	TCP	1466	80 → 10127 [ACK] Seq=1 Ack=410 Win=15744 Len=1412 [TCP segment of a reassembled PDU]	
2016 2020-08-30 07:16:17.456729	39.107.75.2	192.168.101.2	TCP	1091	80 → 10127 [PSH, ACK] Seq=1413 Ack=410 Win=15744 Len=1037 [TCP segment of a reassembled PDU]	
2017 2020-08-30 07:16:17.456729	39.107.75.2	192.168.101.2	HTTP	60	HTTP/1.1 200 (application/json)	
2018 2020-08-30 07:16:17.456802	192.168.101.2	39.107.75.2	TCP	54	10127 → 80 [ACK] Seq=410 Ack=2455 Win=131072 Len=0	
2026 2020-08-30 07:16:17.497544	192.168.101.2	39.107.75.2	HTTP	472	GET /foodie-api/index/sixNewItems/1 HTTP/1.1	
Frame 2004: 463 bytes on wire (3704 bits), 463 bytes captured (3704 bits) on interface 0						
Ethernet II, Src: IntelCor_09:0d:d0 (7c:67:a2:09:0d:d0), Dst: 4c:50:77:a0:49:98 (4c:50:77:a0:49:98)						
Destination: 4c:50:77:a0:49:98 (4c:50:77:a0:49:98)						
Source: IntelCor_09:0d:d0 (7c:67:a2:09:0d:d0)						
Type: IPv4 (0x0800)						
Internet Protocol Version 4, Src: 192.168.101.2, Dst: 39.107.75.2						
0100 .... = Version: 4						
..... 0101 = Header Length: 20 bytes (5)						
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)						
Total Length: 449						
Identification: 0x8e77 (36471)						
Flags: 0x4000, Don't fragment						
Time to live: 64						
Protocol: TCP (6)						
Header checksum: 0x12a8 [validation disabled]						
[Header checksum status: Unverified]						
Source: 192.168.101.2						
Destination: 39.107.75.2						
Transmission Control Protocol, Src Port: 10127, Dst Port: 80, Seq: 1, Ack: 1, Len: 409						
Hypertext Transfer Protocol						

 幕后哈士奇

## 五、IP数据报首部可变部分

(1) **可选字段**：长度可变，从1字节~40字节。可变部分是为了增加IP数据报的功能，如用来支持排错、测量以及安全等措施。

(2) **填充**：IP数据报的首部长度必须是4B的整数倍，所以如果首部长度不满足4B整数倍时，就使用填充字段将首部填充到4B的整数倍。

本篇文章遗留一个问题：标识、标志、片偏移是关于IP数据报分片的三个字段，它们到底是什么意思呢？我们下篇文章来解读。