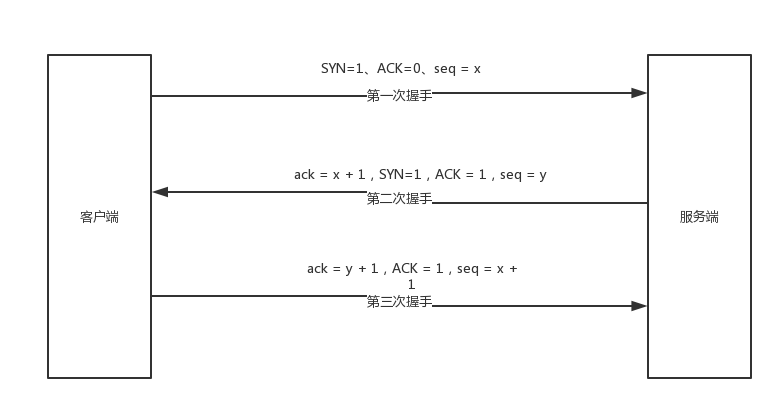
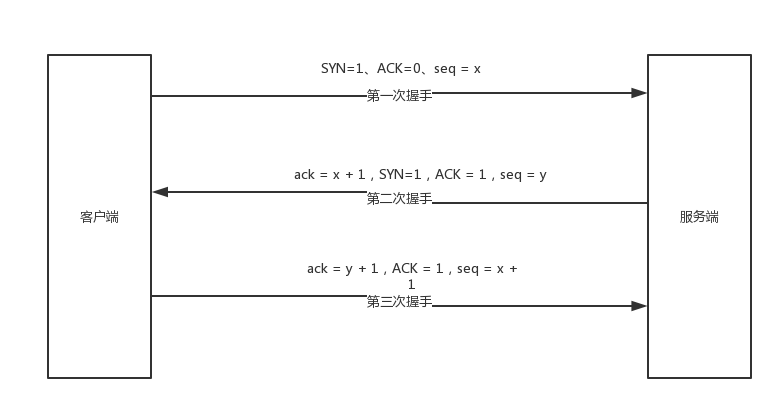
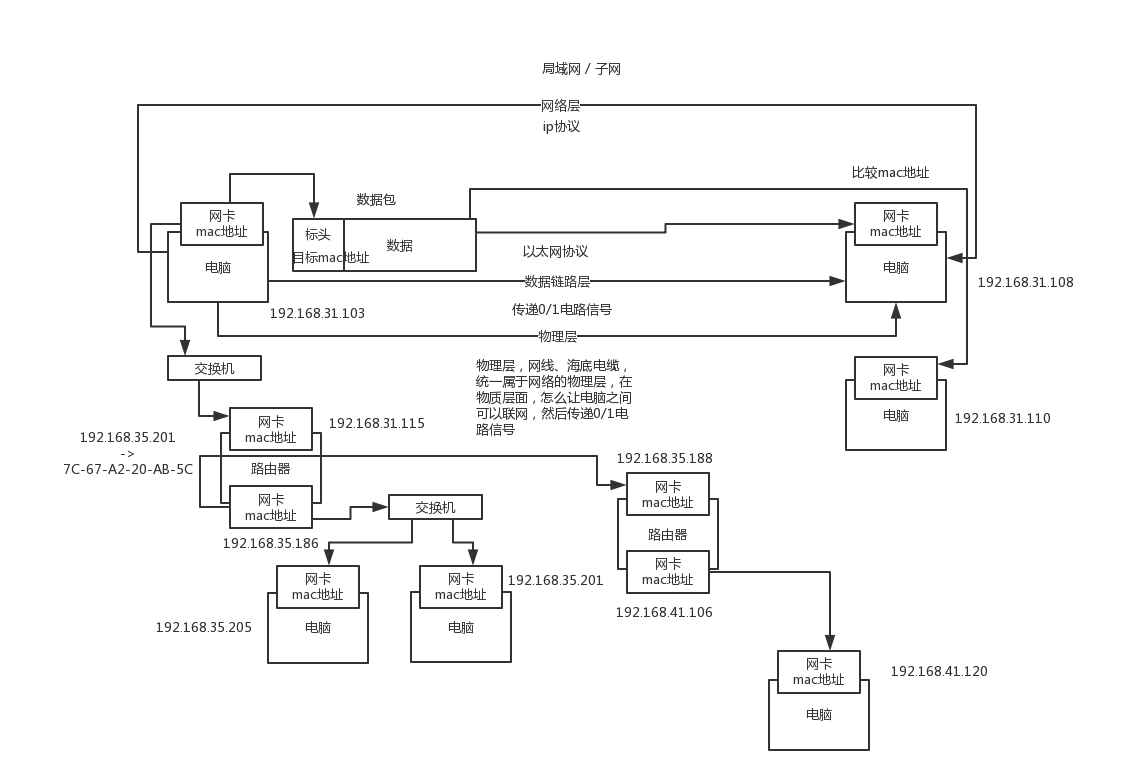
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/008\_一道经典的面试题：你能聊聊TCP三次握手建立连接的过程吗？/01\_tcp三次握手.png**



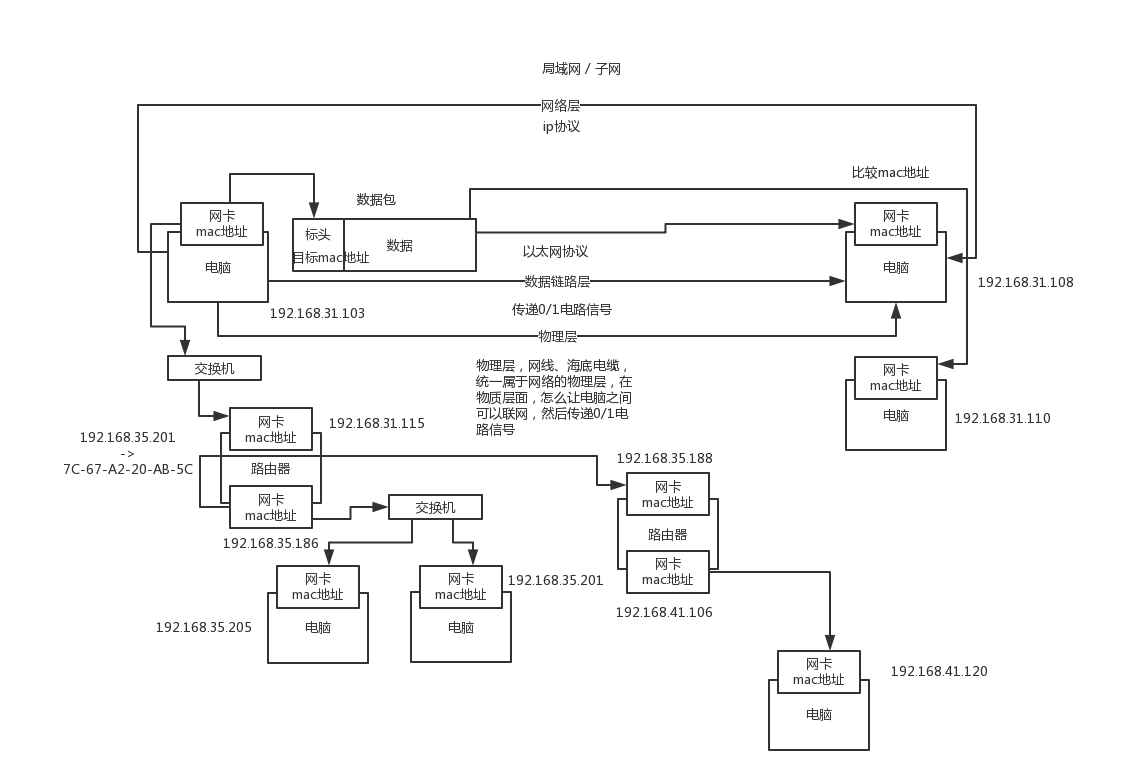
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/009\_一道经典的面试题：如果TCP建立连接就两次握手有什么问题吗？/01\_tcp三次握手.png**



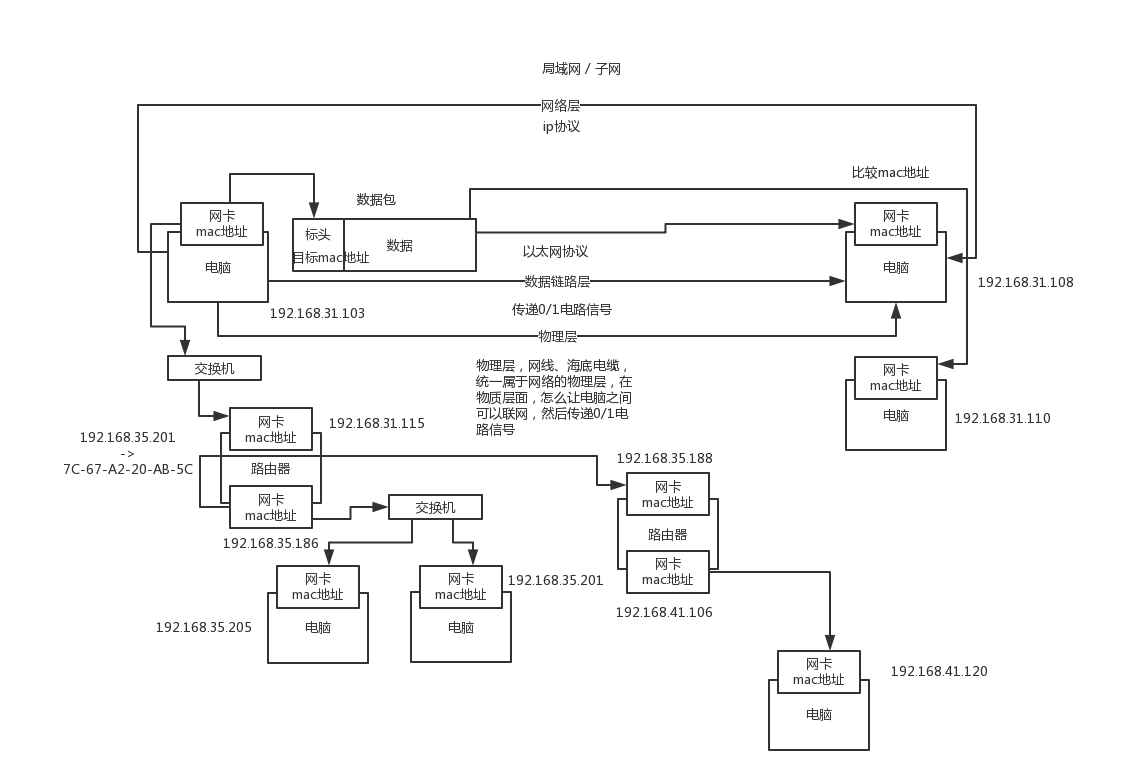
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/001\_Java工程师：你为什么必须要了解一些网络知识？/01\_从底向上学习TCP\_IP四层模型.png**



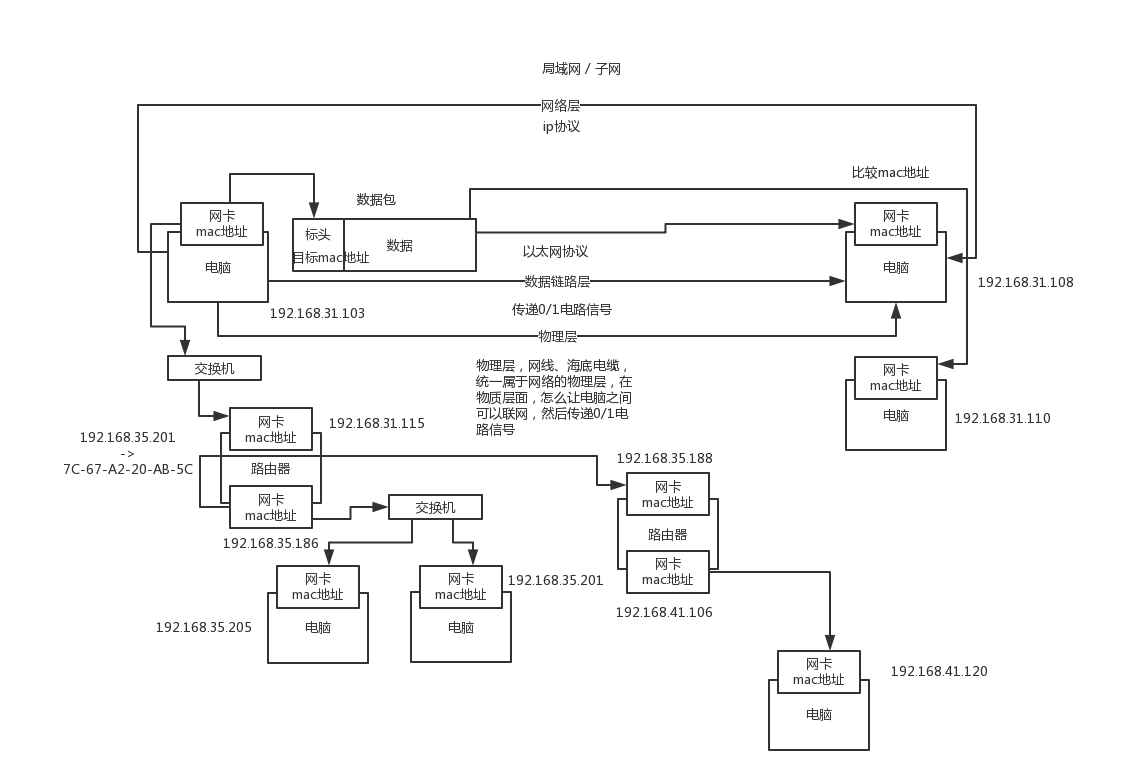
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/002\_数据链路层：以太网协议、mac地址、网卡以及路由器/01\_从底向上学习TCP\_IP四层模型.png**



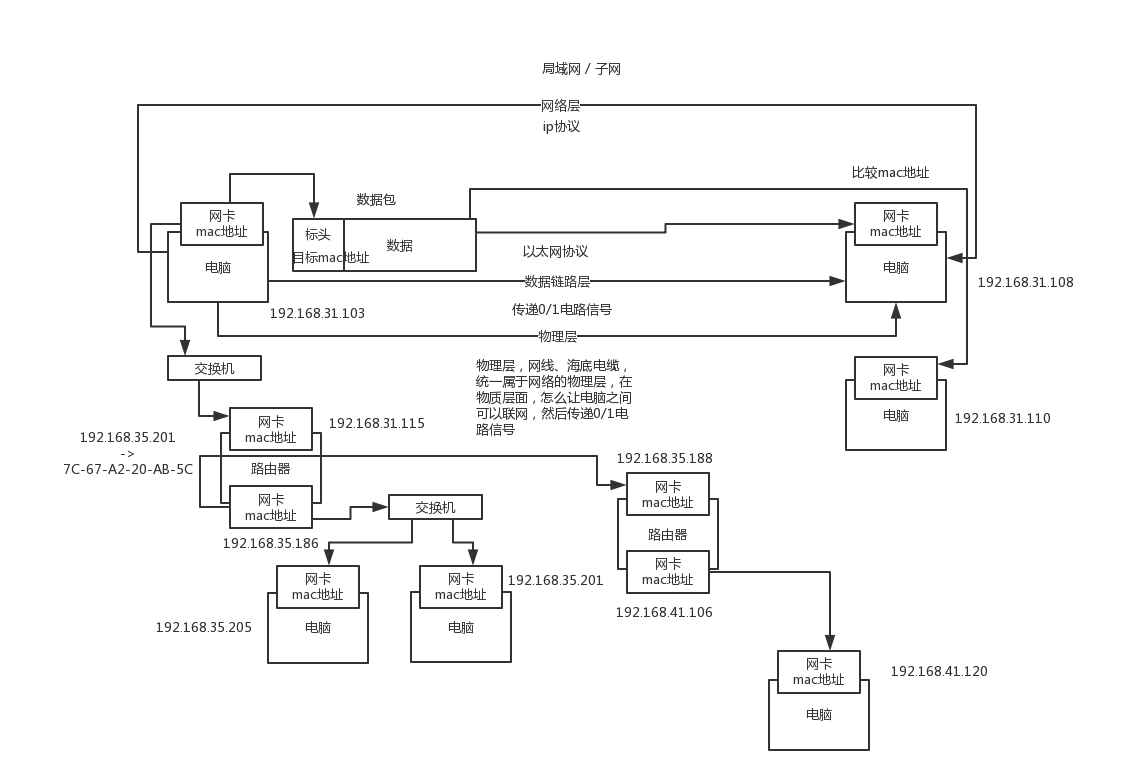
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/003\_网络层：IP协议、子网划分以及子网掩码是什么（上）/01\_从底向上学习TCP\_IP四层模型.png**



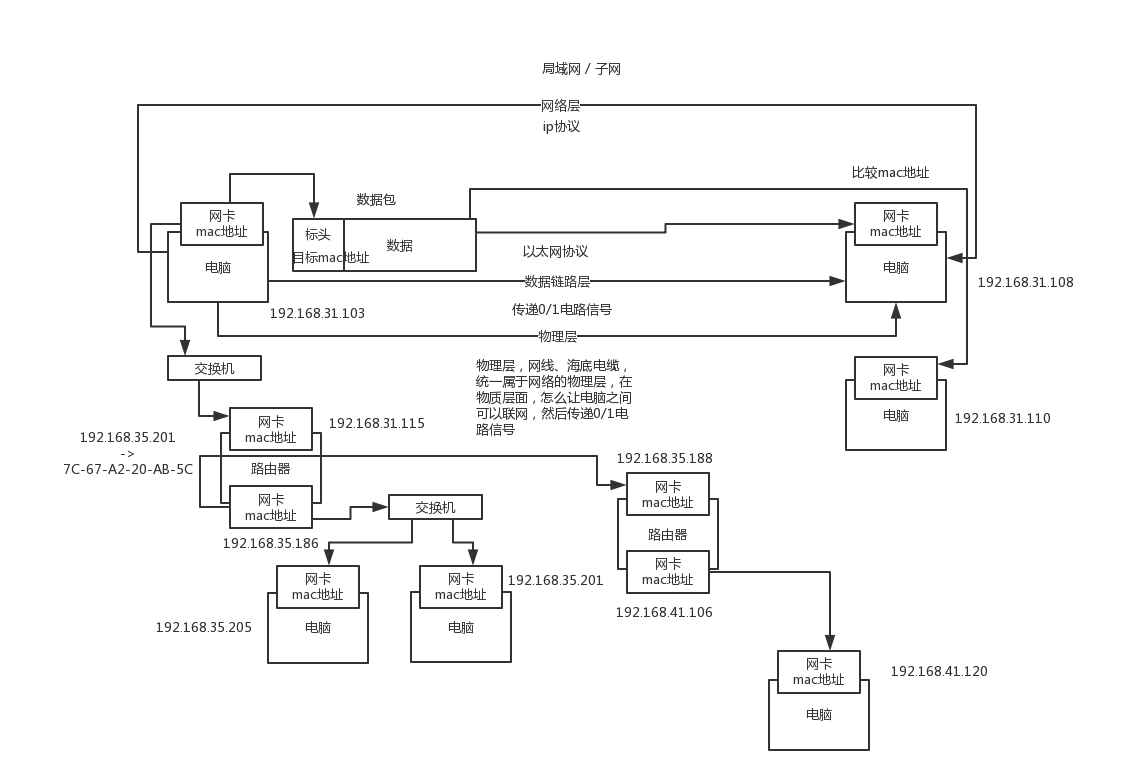
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/005\_传输层：TCP协议、Socket编程以及端口号是什么/01\_从底向上学习TCP\_IP四层模型.png**



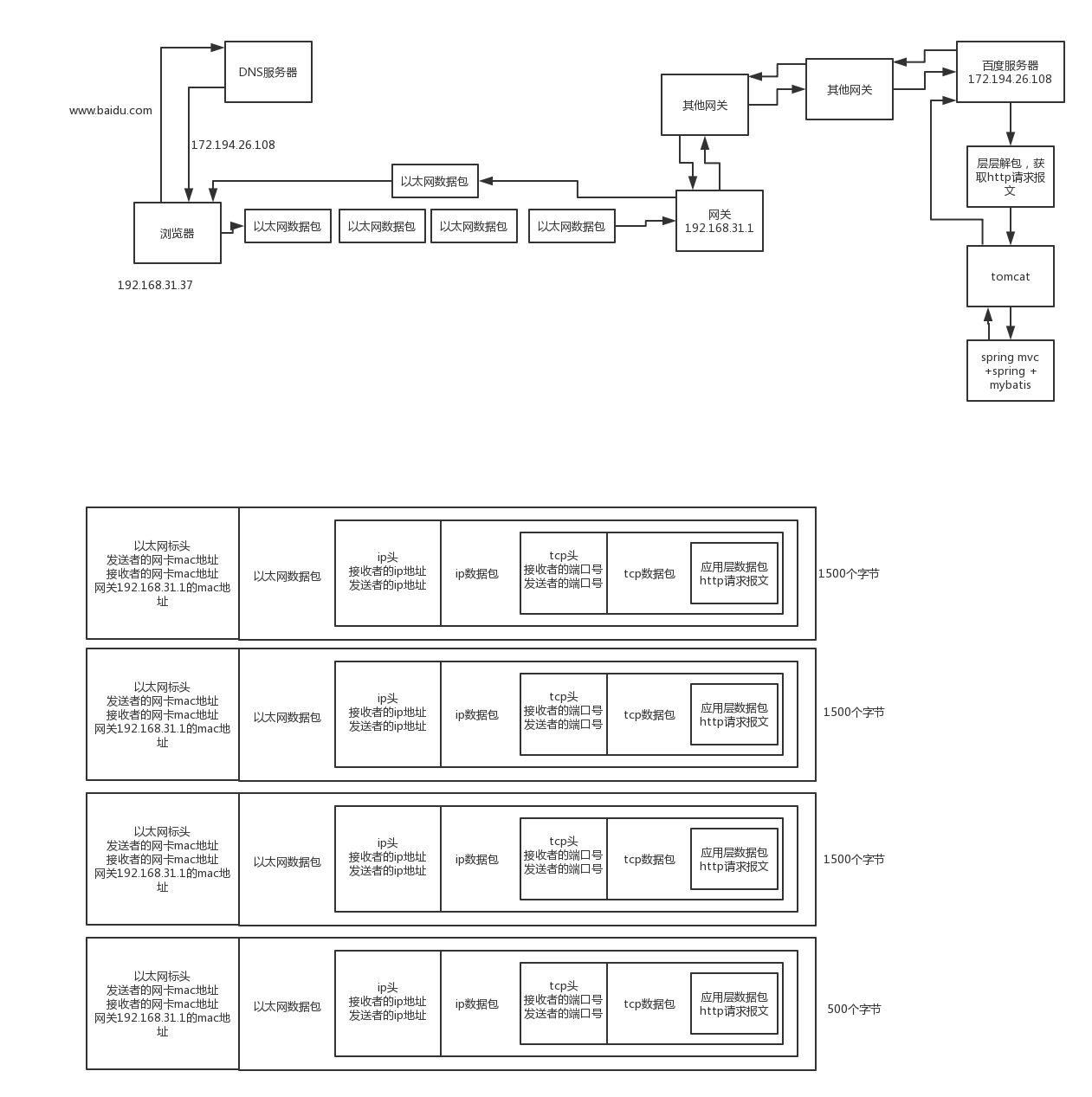
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/006\_应用层：HTTP协议又是什么以及七层网络模型的总结/01\_从底向上学习TCP\_IP四层模型.png**



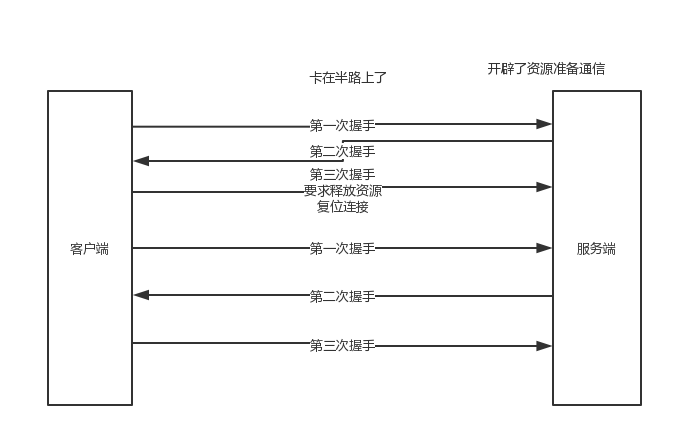
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/004\_网络层：IP协议、子网划分以及子网掩码是什么（下）/01\_从底向上学习TCP\_IP四层模型.png**



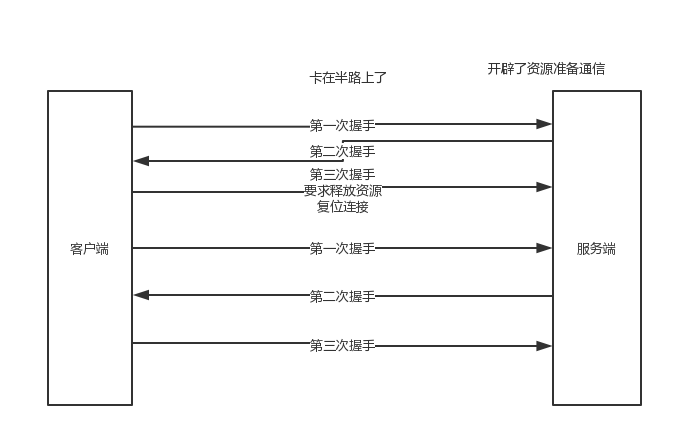
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/007\_一道经典的面试题：通过浏览器请求谷歌的时候会发生哪些事情？/01\_浏览器发送一次请求的过程.png**



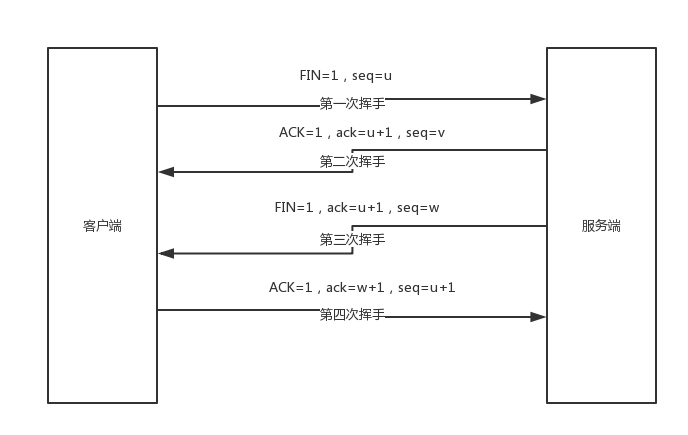
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/008\_一道经典的面试题：你能聊聊TCP三次握手建立连接的过程吗？/02\_tcp连接为什么是3次握手.png**



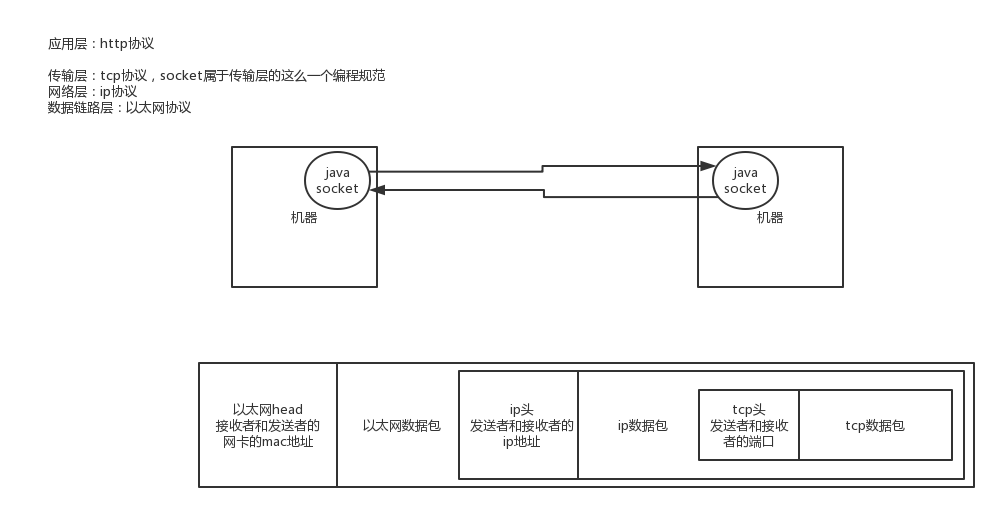
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/009\_一道经典的面试题：如果TCP建立连接就两次握手有什么问题吗？/02\_tcp连接为什么是3次握手.png**



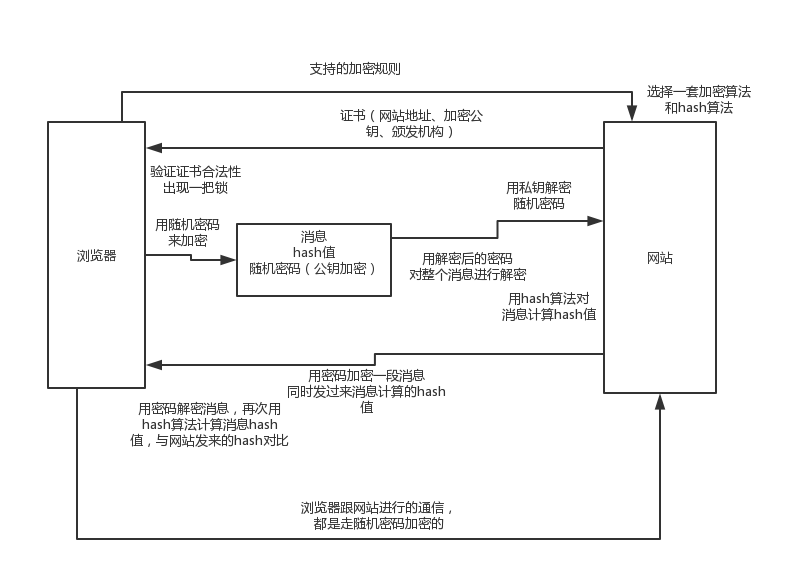
# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/010\_一道经典的面试题：我再追问你一下TCP断开连接四次挥手的过程/03\_tcp的4次挥手断开连接.png**



# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/011\_一道经典的面试题：Socket网络编程跟TCP IP协议的关系是什么/05\_01\_socket工作原理.png**



# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/013\_一道经典的面试题：HTTPS协议加密通信的实现原理是什么/07\_01\_https原理.png**



# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/018\_深入细节：无线局域网和以太网之间又是什么关系？/笔记.doc**

现在我们家里用的路由器其实都是集成了集线器、交换机和路由器的功能的，他是一个综合的东西，之前我们说过，所有电脑发出去的网络请求，其实都是按照几层网络模型来封装数据包

比如说按照tcp协议封装一个包，再封装到ip包里去，最后封装到以太网包里去，这个以太网层面都是有mac地址的，会写在包头里，表明了目标设备的mac地址，接着这个以太网包就会通过电脑的网卡发送出去

这个网卡就会负责把我们的数据包转换为电路信号，就是0101之类的电信号，通过正负电极来做的，这个电路信号也是按照以太网的协议来进行分割，这样电路信号就可以表示出完整的以太网数据包的内容了

然后电信号会通过无线网传输到路由器，路由器里面有一个集线器的功能，这个集线器就是负责把这个数据包广播给子网内的所有设备，这里就是指路由器本身包含的交换机功能，交换机收到这个数据包之后

他会根据mac地址表做一个转发，就是根据你的目标mac地址转发到对应的端口上去，你大概就认为是转发给路由器吧，路由器会根据目标地址和自己本身的mac地址判断一下，发现是自己的包，就会接收这个包

这里大家要注意的一点是，路由器里的交换机的这个功能，实际上是基于mac地址实现的，不知道大家发现了没有，也就是基于以太网协议工作的，属于子网内的通信方式，他是通过集线器功能广播出去数据包

各个设备（比如交换机，路由器）都是根据目标mac地址判断是不是应该自己接收的，所以一定要注意这一点

而且这里要注意的一点，比如你是通过以太网网卡和网线连接路由器的，那么此时就是封装以太网包然后通过以太网网卡发包出去到集线器，然后广播给路由器，路由器网卡接收电路信号，转换为数字信号，根据mac地质判断是自己的包

如果你的笔记本电脑走的是无线网呢？其实是类似的，只不过tcp包封在ip包里，然后ip包就不是封在以太网包里了，是封在无线局域网协议规定格式的包里，然后这个包通过无线网卡发出去，发到路由器上的同样是无线网卡上

所以无论底层是什么样的联网方式，其实本质都是类似的

假设如果我们家里的电脑是通过网线跟路由器连接的，那么可以做以太网协议，按照以太网协议封装好以太网的包，在包头里写上mac地址，发送到交换机上去，转发到路由器上，之间是通过网线连接的

如果是家里走的是无线局域网呢？笔记本电脑跟路由器之间是不走网线的

无线局域网跟以太网是一个层次的协议，都是数据链路层的协议，走的是不同的底层网络数据传输的方式，你的电脑就会具备一个无线网网卡，可以支持不同的上网方式的。家用电脑，支持的就是两种，以太网网卡，无线局域网网卡，走两种不同的协议发网络数据包出去，到路由器上

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/022\_深入细节：为什么说交换机工作在以太网层而路由器工作在IP层？/笔记.doc**

路由器通过一个端口接收到了以太网包之后，会根据mac地址判断是否是转发给自己的包，如果是的话，他就会拆开以太网包，看里面的IP包的ip地址，根据ip地址来转发给自己连接的其他子网的路由器

所以这里非常关键的一点，就是路由器是工作在IP协议层的，他是基于IP协议来的，跟交换机不一样，交换机是工作在以太网协议层的，是基于以太网协议工作的

然后根据下一个路由器的ip地址查找对应的mac地址

路由器之间传输数据，其实也是很简单的，假设也是通过以太网方式来转发数据，那么就是给数据包加上下一个目标路由器的mac地址，接着发送到下一个路由器即可，这个时候会把数据包再次转换为电路信号发送出去

下一个路由器你可以想想肯定也是交换机先接收到数据包的，因为通过以太网协议和mac地址，让下一个路由器的交换机部分收到数据包，根据mac地址转发给下一个路由器，下一个路由器比对一下mac地址发现是给自己的

接着各个路由器之间都按照一样的思路不停的转发，最后一直转发到目标ip地址所在的子网的路由器上去，那个子网路由器再转发给目标的web服务器

这里的话要注意几个要点，一个是这里路由器再发送网络包的时候，其实是会自动拆包的，就是可能会把大的数据包再次拆分为小的包去发送，这是一个，大家需要注意一下

然后这里还有一个问题，假如说你的家用路由器跟下一个路由器之间是通过无线网连接的，或者是ADSL线路连接的，不是以太网连接的，那怎么办呢？其实很简单，你的路由器接收到数据之后，肯定会从里面拿出来ip地址，因为路由器的工作都是依托ip地址来的

接着假如说你是走以太网发送数据到下一个路由器，那就根据以太网协议，查询下一个路由器的ip地址对应的mac地址，然后通过以太网协议发送数据出去，这里会走一个广播的方式，广播到下一个路由器去

下一个路由器的交换机接收到了再转发给路由器，根据mac地址判断是自己的包，这都是走以太网协议的，对吧

那么如果是无线局域网或者是ADSL线路呢？那路由器就根据ip地址找到下一个路由器的ip地址，然后通过无线局域网的规范，或者是网络线路的规范，把数据转换为指定的格式，按照协议封装数据，接着转换为电路信号发送出去就可以了

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/016\_深入细节：如何使用操作系统的Socket库中的DNS解析器查询IP/笔记.doc**

查ip地址，是通过你的电脑上的什么东西去查呢？

就是操作系统的层面提供了Socket库，就可以进行网络相关的一些常见的操作，比如说发送请求到DNS服务器去查询ip地址

DNS服务器是在哪里知道的呢？你的电脑又是怎么知道DNS服务器自己本身的IP地址呢？每台电脑可以自己手动设置一个DNS服务器的地址，当然也可以让电脑自动获取，比如电脑自己预先设置好的DNS服务器地址，这个都没什么问题

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/021\_研究中间件源码不得不回的网络知识：粘包和拆包是什么意思？/笔记.doc**

netty，粘包和拆包是什么意思，是一种常见的网络问题

就是说是这样子的，本来两个包应该是属于独立的两块的数据，但是此时服务器一下子收到了两个包，把里面的数据认为是一块数据，处理的时候，合并起来处理，两个包粘在一起了，数据出现错误

拆包，服务器先收到了一个包的一半的数据，然后又收到了这个包的另外一半的数据，如果说刚收到前一半的数据就开始处理，肯定是不对的，拆包，这也是一个常见的网络问题，netty的时候

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/015\_从Java网络编程的角度来看看HTTP、TCP、以太网的关系/笔记.doc**

看看http、tcp、以太网的关系，硬件设备

（1）浏览器发出了一个请求

他一定会通过底层的操作系统，跟对方的服务器进行TCP三次握手，建立TCP连接

接下来，浏览器就会把自己封装好的http请求报文，你可以认为就是一段http协议规范的请求数据，通过底层的tcp流的方式，如果请求报文比较大的话，此时就会拆包，把一段大的数据拆分放到多个TCP包里去

服务器端其实返回http响应，也是通过底层的TCP的连接，以IO流的形式在发送数据，一段很大的HTML页面，这个就肯定会拆分为多个TCP包，每个包里包含一段数据，不可能通过一个大的包来发送

电脑收到多个TCP包，组装起来，成为一个完整的HTML数据，接着就可以在浏览器里渲染显示这个html页面了

如果此时浏览器被关闭，就会断开TCP连接，也就是走四次挥手的过程，断开这个连接

（2）我们自己手工基于Java的Socket进行网络编程，基于TCP协议进行数据传输

连接连接：TCP三次握手

你通过IO流的形式发送数据过去，底层拆分为多个TCP包过去，人家通过IO流的形式给你返回数据回来，底层拆分为多个TCP回来

断开连接：TCP四次挥手

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/024\_课外有兴趣可以拓展的知识：极客时间《趣谈网络协议》专栏推荐/笔记.doc**

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/017\_深入细节：全世界几万台DNS服务器如何大接力完成IP地址的查询/笔记.doc**

域名，ip地址，怎么可能都在一台DNS服务器上呢？所以去世界有数万台DNS服务器，很多很多，提供DNS的查询服务，在查找ip地址的时候，几万台DNS服务器如何大接力呢？这个说一下

全世界有13台根域DNS服务器

（1）任何一台DNS服务器上都有根域DNS服务器的地址

（2）从根域DNS服务器开始多层级顺藤摸瓜往下级DNS服务器去找

（3）每台DNS服务器都有缓存的功能

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/020\_研究中间件源码不得不会的网络知识：全双工与半双工指的是什么？/笔记.doc**

全双工和半双工，主要发生在路由器的交换机的这块功能里，如果是全双工，就是说发送数据和接收数据可以同时进行；如果是半双工，那么要么这一时刻只能发送数据，要么这一时刻只能接收数据

如果是工作在半双工的情况下，那么需要先判断是否有数据接入进来，避免发生信号碰撞的问题，要等一个方向的数据线过去了，自己才能把数据发出去；如果是全双工的话，不需要判断信号碰撞的问题，直接把数据数据发送出去好了，因为可以同时接收数据和发送数据，在全双工模式下

以太网网络层面处于半双工的工作模式

全双工的意思：同时在网络硬件设备的层面，有人可以发包过去，还有人可以传包回来

这就是他们的区别

在dubbo、netty等底层中间件里，都有全双工和半双工这种网络知识的涉及，所以这就是为什么要深入研究技术，必须先搞定底层网络知识

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/023\_深入细节：家里的调制解调器（Modem）是怎么接入运营商网络的？/笔记.doc**

一般我们家里上网，都是找某个运营商，比如联通，移动，电信之类的，然后上门装个Modem也就是调制解调器，这个大家应该都有印象吧，一般国内我们管那个东西叫做“猫”，这个就是我们家里的路由器可以联网的关键

之前说到路由器收到数据包以后会再次转化为电路信号发送出去，那么发送给谁呢？其实就是这个Modem，只要在数据包上写Modem设备的mac地址，就可以通过以太网协议给发过去

这个Modem收到数据包之后，其实会干一个事儿，就是把数据包拆成一个一个很小的单位，叫做信元，接着就是把这个信元再次转换为电路信号，然后就会通过Modem直接连的网线，比如有的地方不知道大家见过没有，网线是在电线杆上的

当然现在很多地方这个网线都埋到地下去了，接着会通过网线发送到运营商的电话局里去，然后就是进入互联网了，也就是转发到下一个路由器，这个一般是运营商的路由器，然后路由器之间都是类似的转发操作在里面

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/019\_深入细节：ARP广播获取路由器mac地址以及ARP缓存机制/笔记.doc**

比如说我们一般是知道，在自己的笔记本电脑上都会配置一个网关的ip地址，实际上来说就是路由器的网卡的ip地址，同时还有一个mac地址，所以说，我们现在假设仅仅知道他的ip地址，那如何知道路由器网卡的mac地址呢？

ARP机制，就是在电脑上，ARP功能，会在子网内广播一下，找各个设备，问他们呢，你们的ip地址和mac地址的对应的关系能否告知我一下？路由器就会反过来告诉ARP机制，说我的ip地址和mac地址对应关系

ARP机制会自动缓存ip和mac的对应关系

电脑就知道路由网关的网卡ip地址和mac地址之间的关系了

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/001\_Java工程师：你为什么必须要了解一些网络知识？/笔记.docx**

1、面试题

说一下TCP/IP的四层网络模型？那么七层网络模型呢？

2、面试官心里分析

结构化面试：不是瞎面，cs基础（computer science），java基础，j2ee，mysql，linux，tomcat，分布式，高并发，高可用，系统设计，数据结构和算法，软素质，工程素质

坦白讲，一些大的公司，计算机基础必面，尤其是针对薪资30k以内的工程师，因为薪资30k以内，你还是要干活儿的吧，还没上升到就设计架构就可以的程度吧，你还没到那个高度吧。所以只要你干活儿，你就不可避免要跟机器、网络、cpu、磁盘、内存，成天打交道。而线上系统，计算基础的一些东西，网络、cpu、磁盘、内存，都是关联很大的，比如说你线上系统会不会因为网络故障导致一些问题？cpu负载达到100%了咋办？磁盘读写很慢快满了咋办？内存使用率过高咋办？

你起码得有一套自己的计算机功底去支撑你玩儿线上系统吧。所以很多人呢，都说计算机基础没啥用，那这个话呢，也对，也不对。对就在于，你如果毕业出来干简单的crud，这些东西你确实不需要；不对就在于，你如果当个高工，带几个小弟干高并发有压力的线上系统，机器负载很高，很容易出问题，结果你连机器都不敢摸，或者也不知道怎么摸，那不是尴尬了么。

所以说，计算机基础，网络、磁盘、cpu、内存，还是得会一点儿基础的

作为一个大公司的面试官，一定会考察你这些东西

3、面试题剖析

（1）简单介绍一下

四层模型和七层模型可以一块儿聊

为啥需要协议？没有协议，各个电脑厂商，比如IBM、苹果啥的，都弄自己的协议，结果就苹果电脑和苹果电脑自己可以通信，和IBM电脑就不可以通信，这不是尴尬么。所以搞一个国际通行的协议，大家都按照这个来，所有电脑都可以通信，不是很好么。

OSI七层模型，就是搞一个标准的网络模型出来，大家都按照这个来走，那么大家都有统一的规范。OSI七层模型，是应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。TCP/IP四层模型，数据链路层、网络层、传输层、应用层。

（2）从底向上的网络分层

1）物理层

物理层，物理层干啥的，就是电脑之间要联网，一般咋弄？类似于说，你有台电脑，现在要联网，咋联？以前N年前，大家记不记得都是在电脑上插根线是吧，然后才能上网，结果现在就是联个wifi就行了，还有中国美国之前联网靠的是海底的光缆。所以物理层就指的这个，就是怎么把各个电脑给联结起来，形成一个网络，这就是物理层的含义，物理层负责传输0和1的电路信号。学过一些计算机的同学，计算机的最最底层，就是0/1，电信号。

2）数据链路层

数据链路层，物理层给各个电脑连接起来了，还传输最底层的0和1电路信号，关键不行啊，你得定义清楚哪些0和1分为一组，这些信号啥意思？这才能进行通信。所以数据链路层就干这事儿，定义一下电路信号咋分组。

00000011（从电脑1出发，要到电脑2去）

00101（从电脑1出发，要到电脑3去）

0101（从电脑2触发，要到电脑4去）

01（从电脑3出发，要到电脑5去）

很多年前，每个公司都定义自己的电路信号分组方式，但是后来出来了以太网协议，以太网。一组电信号是一个数据包，叫一个帧（frame），每个帧分成两个部分，标头（head）和数据（data），标头包含一些说明性的东西，比如说发送者、接收者和数据类型之类的。

每台电脑要往另外一台电脑发送数据，一堆0/1电路信号，封装成数据包，包含头和数据，头里包含了从哪儿来到哪儿去，必须从一台电脑的一个网卡，发送到另外一个电脑的一个网卡，所以以太网发送的数据包必须得指定，目标电脑的网卡的mac地址。

以太网规定了，每个网卡必须得包含一个mac地址，mac地址就是这个网卡的唯一标识，

以太网协议规定了，接入网络里的所有设备，都得有个网卡，以太网协议里的那个数据包，在数据链路层传输的数据包，必须从一个电脑的网卡传输到另外一个电脑的网卡，而这个网卡地址就叫做所谓的mac地址。每块网卡出厂的时候，就有一个唯一的mac地址，48位的二进制，但是一般用12个16进制数字表示，前6个16进制是厂商编号，后6个16进制是网卡流水号。

windows上，ipconfig /all，看看物理地址，就是mac地址，7C-67-A2-20-AB-5C

所以在以太网里传输数据包的时候，必须指定接收者的mac地址才能传输数据。

但是以太网的数据包怎么从一个mac地址发送到另一个mac地址？这个不是精准推送的，以太网里面，如果一个电脑发个数据包出去，会广播给局域网内的所有电脑设备的网卡，然后每台电脑都从数据包里获取接收者的mac地址，跟自己的mac地址对比一下，如果一样，就说明这是发给自己的数据包。

但是上面这种广播的方式，仅仅针对一个子网（局域网）内的电脑，会广播，否则一个电脑不能广播数据包给全世界所有的其他电脑吧，是仅仅广播给一个子网里面的电脑的。

3）网络层

上面说到，子网内的电脑，通过以太网发个数据包，对局域网内的电脑，是广播出去的。那么怎么知道哪些电脑在一个子网内呢？这就得靠网络层了，这里就有一套IP地址，IP地址就可以让我们区分哪些电脑是一个子网的。

网络层里有IP协议，IP协议定义的地址就叫做IP地址。IP地址有IPv4和IPv6两个版本，目前广泛使用的是IPv4，是32个二进制数字组成的，但是一般用4个十进制数字表示，范围从0.0.0.0到255.255.255.255之间。

每台计算机，都会分配一个ip地址，ip地址的前24位（就是前面3个十进制数字），代表了网络，后8位（就是最后1个十进制数字），代表了主机。如果几台电脑是一个子网的，那么前面的3个十进制数字一定是一样的。举个例子，大家平时做实验，玩儿虚拟机吧，自己win上开几个linux虚拟机，你会发现，win上的ip地址可能是192.168.0.103，然后几个虚拟机的ip地址是192.168.0.182，192.168.0.125，192.168.0.106，类似这样的。这个win机器和几个虚拟机，前面3个十进制数字都是192.168.0，就代表大家是一个子网内的，最后那个数字是这个子网的不同主机的编号。

但是实际上上面就是举个例子，其实单单从ip地址是看不出来哪些机器是一个子网的，因为从10进制是判断不出来的。需要通过ip地址的二进制来判断，结合一个概念来判断，叫做子网掩码。比如说ip地址是192.168.56.1，子网掩码是255.255.255.0。知道了子网掩码之后，如果要判断两个ip地址是不是一个子网的，就分别把两个ip地址和自己的子网掩码进行二进制的与运算，与运算之后，比较一下代表网络的那部分。

192.168.56.1和192.168.32.7，判断是不是一个子网的，拿子网掩码255.255.255.0，跟两个ip地址的二进制做与运算

11000000.10101000.00111000.00000001

11111111.11111111.11111111.00000000

子网掩码的二进制是：11111111.11111111.11111111.00000000，然后就跟ip地址的二进制做与好了，通过二进制来比较网络部分的地址是不是一模一样的。

有了网络层的ip地址之后，两台在子网内的电脑终于可以通过广播+mac地址判断来传输数据包进行通信了。

但是如果发现要接受数据包的计算机不在子网内，那么就不能通过广播来发送数据包，需要通过路由来发送数据包。看到路由，就想到了路由器了，对了，路由器大家都熟悉吧，自己平时也会去买对吧，比如小米的路由器啥的，家里上网一般都会弄个路由器对吧，ok。路由器负责将多个子网进行连接，因为比如你在自己家里，其实你就只是你自己的一个子网，你要是访问网站啥的，是跟那个网站机器所在的子网进行通信。

每个电脑都可以搞多个网卡的，不是只有一个网卡，一般笔记本电脑都有以太网网卡和wifi网卡，发送数据包的时候要决定走哪个网卡。路由器，其实就是配置了多个网卡的一个专用设备，可以通过不同的网卡接入不同的网络。

网关其实是就是路由器的一种，运作在网络层，这个概念不多解释了，大家可以就把路由器上的ip地址认为是网关，路由器上每个网卡都有mac地址和对应的ip地址。路由器虽然有mac地址，但是不能通过mac地址寻址的，必须通过ip地址寻址，所以路由器其实是工作在网络层的设备。

网络交换机，也是一种设备，是工作在数据链路层的，路由器是工作在网路层的。网络交换机是通过mac地址来寻址和传输数据包的；但是路由器是通过ip地址寻址和传输数据包的。网络交换机主要用在局域网的通信，一般你架设一个局域网，里面的电脑通信是通过数据链路层发送数据包，通过mac地址来广播的，广播的时候就是通过网络交换机这个设备来把数据广播到局域网内的其他机器上去的；路由器一般用来让你连入英特网。

LAN，就是local area network，就是局域网；WAN，就是wide area network，就是广域网。WLAN是wireless local area network，就是无线局域网，也就是wifi，在局域网内，直接通过wifi无线联网。

家里的路由器是包含了交换机和路由的两个功能的，如果是连接到局域网内的设备就把线插LAN那儿；如果是连接到英特网，就把线插在WAN那儿。

这儿给大家举个例子，就是两个局域网之间，如果要是通过一个路由器进行通信的话，是怎么弄的。

大概过程就是，路由器配置了两块网卡，每个网卡可以连到一个局域网内。

局域网1内的电脑，要发送数据包到局域网2内的电脑，在数据包里写上自己的ip地址和对方的ip地址。但是他们俩不在一个局域网内，于是局域网1内的电脑，先通过交换机将数据包发送给路由器，这个过程需要将路由器的一块网卡的ip地址对应的mac地址写到数据包的头部，然后才能通过交换机广播出去，路由器接收到之后比较自己一块网卡的mac地址，就知道是来找自己的。

接着路由器接收到数据包之后，就会在局域网2内，将目标机器的ip地址对应的mac地址写入头部，接着再次通过交换机发送广播通知，发送给局域网2内的电脑。

一个局域网内的每台机器都有自己的ARP cache，这个ARP就是用来在一个局域网内让各个设备都知道每个设备的ip地址和mac地址的对应关系的，一般就是某个机器发送广播通知自己的ip地址和mac地址的对应关系，然后每个机器给他一个回应。以此类推，大家都互相这样广播一把，ip地址和mac地址的对应关系，大家不就都知道了。

一个ip地址对应着一个mac地址

所以大家在上面可以看到，一个子网内的机器之间通信，就是在数据包里写上对方的mac地址，然后交换机广播出去ok了；但是如果是跨子网的通信，就是写上对方的ip地址，然后先通过mac地址广播到路由器，让路由器再根据另外一个子网的ip地址转换为mac地址，通过另外一个子网的交换机广播过去。就这个意思。

4）传输层

上面我们大概明白了通过网络层的ip地址怎么划分出来一个一个的子网，然后在子网内部怎么通过mac地址广播通信；跨子网的时候，怎么通过ip地址 -> mac地址 -> 交换机 -> 路由器 -> ip地址 -> mac地址 -> 交换机的方式来通过路由器进行通信。

但是这里还有一个问题，就是一台机器上，是很多个程序用一个网卡进行网络通信的，比如说浏览器、QQ、视频直播，这些软件都用了一个网卡往外面发送数据，然后从网卡接收数据，对吧。

所以还需要一个端口号的概念，就是你得发送数据包到某个机器的一个网卡的某个端口上去，然后那个机器上监听那个端口的程序，就可以提取发送到这个端口的数据，知道是自己的数据。端口号是0~65536的范围内，0~1023被系统占用了，别的应用程序就用1024以上的端口就ok了。

电脑1，是在端口48362监听的，通过网卡发送了一条数据 -> 电脑2的ip地址的20386这个端口 -> 电脑2的上面的某个QQ，监听着20386的端口 -> 电脑2的网卡接收到一条数据之后，发现人家找的是20386这个端口，就去找谁哪个哥儿们在监听20386端口，QQ在监听，我就把这个网卡过来的数据，传递给QQ，通过端口知道，哪条数据是给你的

所以其实大家会发现一点，网络层，是基于ip协议，进行主机和主机间的寻址和通信的，然后传输层，其实是建立某个主机的某个端口，到另外一个主机的某个端口的连接和通信的。这个通信，就是通过socket来实现的，通过socket就可以基于tcp/ip协议完成刚才上面说的一系列的比如基于ip地址和mac地址转换和寻址啊，通过路由器通信啊之类的，而且会建立一个端口到另外一个端口的连接。

udp和tcp都是传输层的协议，作用就是在数据包里加入端口号，可以通过端口号进行点对点的通信了。udp协议是不可靠的，发出去人家收到没有就不知道了；tcp协议是可靠的，要求三次握手，而且要求人家接收到数据必须回复你。

传输层的tcp协议，仅仅只是规定了一套基于端口的点对点的通信协议，包括如何建立连接，如何发送和读取消息，但是实际上如果你要基于tcp协议来开发，你一般是用socket，java socket网络编程

5）应用层

通过传输层的tcp协议可以传输数据，但是人家收到数据之后，怎么来解释？比如说收到个邮件你怎么处理？收到个网页你怎么处理？类似这个意思，所以针对各种不同的应用，邮件、网页之类的，都是定义不同的应用层协议的。这个应用层，我们就假设综合了会话层、表示层和应用层了，3层合成1层。

电脑1走tcp协议发送了一段东西过来，发送到电脑2的20386端口

GET http://localhost:8080/ http/1.1

key:valuel

key:value

电脑2走tcp协议读取到了属于自己这个20386端口 的一段数据

GET http://localhost:8080/ http/1.1

key:valuel

key:value

发送了一段响应

200

key;value

key:value

又通过底层的tcp发了出去，电脑1的30987端口，ip

电脑1，网卡，走以太网协议收到一个数据包

200

key;value

key:value

比如最常见的，应用层的协议就是http协议，进行网络通信。

4层：数据链路层（以太网协议），网络层（ip协议），传输层（tcp协议），应用层（http协议）

7层：物理层（网线，海底光缆，传递0/1电路信号），会话层、表示层、应用层 -> 应用层

然后我们看下自己的网络设置，一般包含了ip地址、子网掩码、网关地址、DNS地址。前面3个我们其实都知道啥意思了。ip地址和子网掩码用来划分子网的，判断哪些ip地址在一个子网内。同时你的ip地址和mac地址关联起来的，唯一定位了你的网卡。网关地址，你就认为是路由器上的那个网卡的ip地址吧，路由器的网卡也有mac地址，mac地址对应了一个ip地址。

DNS地址是啥呢？Domain Name System。因为我们一般定位是通过ip地址+mac地址+端口号来定位一个通信目标的，但是如果在浏览器上输入一个www.baidu.com，咋整？这个时候是先把www.baidu.com发给DNS服务器，然后DNS服务器告诉你www.baidu.com对应的ip地址的。

8.8.8.8

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/002\_数据链路层：以太网协议、mac地址、网卡以及路由器/笔记.docx**

1、面试题

说一下TCP/IP的四层网络模型？那么七层网络模型呢？

2、面试官心里分析

结构化面试：不是瞎面，cs基础（computer science），java基础，j2ee，mysql，linux，tomcat，分布式，高并发，高可用，系统设计，数据结构和算法，软素质，工程素质

坦白讲，一些大的公司，计算机基础必面，尤其是针对薪资30k以内的工程师，因为薪资30k以内，你还是要干活儿的吧，还没上升到就设计架构就可以的程度吧，你还没到那个高度吧。所以只要你干活儿，你就不可避免要跟机器、网络、cpu、磁盘、内存，成天打交道。而线上系统，计算基础的一些东西，网络、cpu、磁盘、内存，都是关联很大的，比如说你线上系统会不会因为网络故障导致一些问题？cpu负载达到100%了咋办？磁盘读写很慢快满了咋办？内存使用率过高咋办？

你起码得有一套自己的计算机功底去支撑你玩儿线上系统吧。所以很多人呢，都说计算机基础没啥用，那这个话呢，也对，也不对。对就在于，你如果毕业出来干简单的crud，这些东西你确实不需要；不对就在于，你如果当个高工，带几个小弟干高并发有压力的线上系统，机器负载很高，很容易出问题，结果你连机器都不敢摸，或者也不知道怎么摸，那不是尴尬了么。

所以说，计算机基础，网络、磁盘、cpu、内存，还是得会一点儿基础的

作为一个大公司的面试官，一定会考察你这些东西

3、面试题剖析

（1）简单介绍一下

四层模型和七层模型可以一块儿聊

为啥需要协议？没有协议，各个电脑厂商，比如IBM、苹果啥的，都弄自己的协议，结果就苹果电脑和苹果电脑自己可以通信，和IBM电脑就不可以通信，这不是尴尬么。所以搞一个国际通行的协议，大家都按照这个来，所有电脑都可以通信，不是很好么。

OSI七层模型，就是搞一个标准的网络模型出来，大家都按照这个来走，那么大家都有统一的规范。OSI七层模型，是应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。TCP/IP四层模型，数据链路层、网络层、传输层、应用层。

（2）从底向上的网络分层

1）物理层

物理层，物理层干啥的，就是电脑之间要联网，一般咋弄？类似于说，你有台电脑，现在要联网，咋联？以前N年前，大家记不记得都是在电脑上插根线是吧，然后才能上网，结果现在就是联个wifi就行了，还有中国美国之前联网靠的是海底的光缆。所以物理层就指的这个，就是怎么把各个电脑给联结起来，形成一个网络，这就是物理层的含义，物理层负责传输0和1的电路信号。学过一些计算机的同学，计算机的最最底层，就是0/1，电信号。

2）数据链路层

数据链路层，物理层给各个电脑连接起来了，还传输最底层的0和1电路信号，关键不行啊，你得定义清楚哪些0和1分为一组，这些信号啥意思？这才能进行通信。所以数据链路层就干这事儿，定义一下电路信号咋分组。

00000011（从电脑1出发，要到电脑2去）

00101（从电脑1出发，要到电脑3去）

0101（从电脑2触发，要到电脑4去）

01（从电脑3出发，要到电脑5去）

很多年前，每个公司都定义自己的电路信号分组方式，但是后来出来了以太网协议，以太网。一组电信号是一个数据包，叫一个帧（frame），每个帧分成两个部分，标头（head）和数据（data），标头包含一些说明性的东西，比如说发送者、接收者和数据类型之类的。

每台电脑要往另外一台电脑发送数据，一堆0/1电路信号，封装成数据包，包含头和数据，头里包含了从哪儿来到哪儿去，必须从一台电脑的一个网卡，发送到另外一个电脑的一个网卡，所以以太网发送的数据包必须得指定，目标电脑的网卡的mac地址。

以太网规定了，每个网卡必须得包含一个mac地址，mac地址就是这个网卡的唯一标识，

以太网协议规定了，接入网络里的所有设备，都得有个网卡，以太网协议里的那个数据包，在数据链路层传输的数据包，必须从一个电脑的网卡传输到另外一个电脑的网卡，而这个网卡地址就叫做所谓的mac地址。每块网卡出厂的时候，就有一个唯一的mac地址，48位的二进制，但是一般用12个16进制数字表示，前6个16进制是厂商编号，后6个16进制是网卡流水号。

windows上，ipconfig /all，看看物理地址，就是mac地址，7C-67-A2-20-AB-5C

所以在以太网里传输数据包的时候，必须指定接收者的mac地址才能传输数据。

但是以太网的数据包怎么从一个mac地址发送到另一个mac地址？这个不是精准推送的，以太网里面，如果一个电脑发个数据包出去，会广播给局域网内的所有电脑设备的网卡，然后每台电脑都从数据包里获取接收者的mac地址，跟自己的mac地址对比一下，如果一样，就说明这是发给自己的数据包。

但是上面这种广播的方式，仅仅针对一个子网（局域网）内的电脑，会广播，否则一个电脑不能广播数据包给全世界所有的其他电脑吧，是仅仅广播给一个子网里面的电脑的。

3）网络层

上面说到，子网内的电脑，通过以太网发个数据包，对局域网内的电脑，是广播出去的。那么怎么知道哪些电脑在一个子网内呢？这就得靠网络层了，这里就有一套IP地址，IP地址就可以让我们区分哪些电脑是一个子网的。

网络层里有IP协议，IP协议定义的地址就叫做IP地址。IP地址有IPv4和IPv6两个版本，目前广泛使用的是IPv4，是32个二进制数字组成的，但是一般用4个十进制数字表示，范围从0.0.0.0到255.255.255.255之间。

每台计算机，都会分配一个ip地址，ip地址的前24位（就是前面3个十进制数字），代表了网络，后8位（就是最后1个十进制数字），代表了主机。如果几台电脑是一个子网的，那么前面的3个十进制数字一定是一样的。举个例子，大家平时做实验，玩儿虚拟机吧，自己win上开几个linux虚拟机，你会发现，win上的ip地址可能是192.168.0.103，然后几个虚拟机的ip地址是192.168.0.182，192.168.0.125，192.168.0.106，类似这样的。这个win机器和几个虚拟机，前面3个十进制数字都是192.168.0，就代表大家是一个子网内的，最后那个数字是这个子网的不同主机的编号。

但是实际上上面就是举个例子，其实单单从ip地址是看不出来哪些机器是一个子网的，因为从10进制是判断不出来的。需要通过ip地址的二进制来判断，结合一个概念来判断，叫做子网掩码。比如说ip地址是192.168.56.1，子网掩码是255.255.255.0。知道了子网掩码之后，如果要判断两个ip地址是不是一个子网的，就分别把两个ip地址和自己的子网掩码进行二进制的与运算，与运算之后，比较一下代表网络的那部分。

192.168.56.1和192.168.32.7，判断是不是一个子网的，拿子网掩码255.255.255.0，跟两个ip地址的二进制做与运算

11000000.10101000.00111000.00000001

11111111.11111111.11111111.00000000

子网掩码的二进制是：11111111.11111111.11111111.00000000，然后就跟ip地址的二进制做与好了，通过二进制来比较网络部分的地址是不是一模一样的。

有了网络层的ip地址之后，两台在子网内的电脑终于可以通过广播+mac地址判断来传输数据包进行通信了。

但是如果发现要接受数据包的计算机不在子网内，那么就不能通过广播来发送数据包，需要通过路由来发送数据包。看到路由，就想到了路由器了，对了，路由器大家都熟悉吧，自己平时也会去买对吧，比如小米的路由器啥的，家里上网一般都会弄个路由器对吧，ok。路由器负责将多个子网进行连接，因为比如你在自己家里，其实你就只是你自己的一个子网，你要是访问网站啥的，是跟那个网站机器所在的子网进行通信。

每个电脑都可以搞多个网卡的，不是只有一个网卡，一般笔记本电脑都有以太网网卡和wifi网卡，发送数据包的时候要决定走哪个网卡。路由器，其实就是配置了多个网卡的一个专用设备，可以通过不同的网卡接入不同的网络。

网关其实是就是路由器的一种，运作在网络层，这个概念不多解释了，大家可以就把路由器上的ip地址认为是网关，路由器上每个网卡都有mac地址和对应的ip地址。路由器虽然有mac地址，但是不能通过mac地址寻址的，必须通过ip地址寻址，所以路由器其实是工作在网络层的设备。

网络交换机，也是一种设备，是工作在数据链路层的，路由器是工作在网路层的。网络交换机是通过mac地址来寻址和传输数据包的；但是路由器是通过ip地址寻址和传输数据包的。网络交换机主要用在局域网的通信，一般你架设一个局域网，里面的电脑通信是通过数据链路层发送数据包，通过mac地址来广播的，广播的时候就是通过网络交换机这个设备来把数据广播到局域网内的其他机器上去的；路由器一般用来让你连入英特网。

LAN，就是local area network，就是局域网；WAN，就是wide area network，就是广域网。WLAN是wireless local area network，就是无线局域网，也就是wifi，在局域网内，直接通过wifi无线联网。

家里的路由器是包含了交换机和路由的两个功能的，如果是连接到局域网内的设备就把线插LAN那儿；如果是连接到英特网，就把线插在WAN那儿。

这儿给大家举个例子，就是两个局域网之间，如果要是通过一个路由器进行通信的话，是怎么弄的。

大概过程就是，路由器配置了两块网卡，每个网卡可以连到一个局域网内。

局域网1内的电脑，要发送数据包到局域网2内的电脑，在数据包里写上自己的ip地址和对方的ip地址。但是他们俩不在一个局域网内，于是局域网1内的电脑，先通过交换机将数据包发送给路由器，这个过程需要将路由器的一块网卡的ip地址对应的mac地址写到数据包的头部，然后才能通过交换机广播出去，路由器接收到之后比较自己一块网卡的mac地址，就知道是来找自己的。

接着路由器接收到数据包之后，就会在局域网2内，将目标机器的ip地址对应的mac地址写入头部，接着再次通过交换机发送广播通知，发送给局域网2内的电脑。

一个局域网内的每台机器都有自己的ARP cache，这个ARP就是用来在一个局域网内让各个设备都知道每个设备的ip地址和mac地址的对应关系的，一般就是某个机器发送广播通知自己的ip地址和mac地址的对应关系，然后每个机器给他一个回应。以此类推，大家都互相这样广播一把，ip地址和mac地址的对应关系，大家不就都知道了。

一个ip地址对应着一个mac地址

所以大家在上面可以看到，一个子网内的机器之间通信，就是在数据包里写上对方的mac地址，然后交换机广播出去ok了；但是如果是跨子网的通信，就是写上对方的ip地址，然后先通过mac地址广播到路由器，让路由器再根据另外一个子网的ip地址转换为mac地址，通过另外一个子网的交换机广播过去。就这个意思。

4）传输层

上面我们大概明白了通过网络层的ip地址怎么划分出来一个一个的子网，然后在子网内部怎么通过mac地址广播通信；跨子网的时候，怎么通过ip地址 -> mac地址 -> 交换机 -> 路由器 -> ip地址 -> mac地址 -> 交换机的方式来通过路由器进行通信。

但是这里还有一个问题，就是一台机器上，是很多个程序用一个网卡进行网络通信的，比如说浏览器、QQ、视频直播，这些软件都用了一个网卡往外面发送数据，然后从网卡接收数据，对吧。

所以还需要一个端口号的概念，就是你得发送数据包到某个机器的一个网卡的某个端口上去，然后那个机器上监听那个端口的程序，就可以提取发送到这个端口的数据，知道是自己的数据。端口号是0~65536的范围内，0~1023被系统占用了，别的应用程序就用1024以上的端口就ok了。

电脑1，是在端口48362监听的，通过网卡发送了一条数据 -> 电脑2的ip地址的20386这个端口 -> 电脑2的上面的某个QQ，监听着20386的端口 -> 电脑2的网卡接收到一条数据之后，发现人家找的是20386这个端口，就去找谁哪个哥儿们在监听20386端口，QQ在监听，我就把这个网卡过来的数据，传递给QQ，通过端口知道，哪条数据是给你的

所以其实大家会发现一点，网络层，是基于ip协议，进行主机和主机间的寻址和通信的，然后传输层，其实是建立某个主机的某个端口，到另外一个主机的某个端口的连接和通信的。这个通信，就是通过socket来实现的，通过socket就可以基于tcp/ip协议完成刚才上面说的一系列的比如基于ip地址和mac地址转换和寻址啊，通过路由器通信啊之类的，而且会建立一个端口到另外一个端口的连接。

udp和tcp都是传输层的协议，作用就是在数据包里加入端口号，可以通过端口号进行点对点的通信了。udp协议是不可靠的，发出去人家收到没有就不知道了；tcp协议是可靠的，要求三次握手，而且要求人家接收到数据必须回复你。

传输层的tcp协议，仅仅只是规定了一套基于端口的点对点的通信协议，包括如何建立连接，如何发送和读取消息，但是实际上如果你要基于tcp协议来开发，你一般是用socket，java socket网络编程

5）应用层

通过传输层的tcp协议可以传输数据，但是人家收到数据之后，怎么来解释？比如说收到个邮件你怎么处理？收到个网页你怎么处理？类似这个意思，所以针对各种不同的应用，邮件、网页之类的，都是定义不同的应用层协议的。这个应用层，我们就假设综合了会话层、表示层和应用层了，3层合成1层。

电脑1走tcp协议发送了一段东西过来，发送到电脑2的20386端口

GET http://localhost:8080/ http/1.1

key:valuel

key:value

电脑2走tcp协议读取到了属于自己这个20386端口 的一段数据

GET http://localhost:8080/ http/1.1

key:valuel

key:value

发送了一段响应

200

key;value

key:value

又通过底层的tcp发了出去，电脑1的30987端口，ip

电脑1，网卡，走以太网协议收到一个数据包

200

key;value

key:value

比如最常见的，应用层的协议就是http协议，进行网络通信。

4层：数据链路层（以太网协议），网络层（ip协议），传输层（tcp协议），应用层（http协议）

7层：物理层（网线，海底光缆，传递0/1电路信号），会话层、表示层、应用层 -> 应用层

然后我们看下自己的网络设置，一般包含了ip地址、子网掩码、网关地址、DNS地址。前面3个我们其实都知道啥意思了。ip地址和子网掩码用来划分子网的，判断哪些ip地址在一个子网内。同时你的ip地址和mac地址关联起来的，唯一定位了你的网卡。网关地址，你就认为是路由器上的那个网卡的ip地址吧，路由器的网卡也有mac地址，mac地址对应了一个ip地址。

DNS地址是啥呢？Domain Name System。因为我们一般定位是通过ip地址+mac地址+端口号来定位一个通信目标的，但是如果在浏览器上输入一个www.baidu.com，咋整？这个时候是先把www.baidu.com发给DNS服务器，然后DNS服务器告诉你www.baidu.com对应的ip地址的。

8.8.8.8

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/007\_一道经典的面试题：通过浏览器请求谷歌的时候会发生哪些事情？/笔记.docx**

1、面试题

用浏览器请求一个链接的时候，经历了哪些过程（DNS解析过程）？

2、面试官心里分析

这个问题，其实就是跟之前的那个一样，他就是考察考察你的基本功，看你对基本的网络通信知识有没有了解。而且话说回来，考察一个人的基本功，就这个问题应该是相当经典和直接的一个问题，你理解清楚了，那么网络通信这块的一些基本概念，你基本都了解了。否则有的人连个网卡是啥都不知道，mac地址更是懵逼。

3、面试题剖析

现在我们知道网络七层模型大概都是怎么回事了，然后四层模型其实就是会话层、表示层和应用层，合并为了一个应用层，同时没把物理层算在内。就这样。

而且我们也大概知道每一层的协议和作用，网络通信的时候都是怎么回事了，现在我们来看看假设通过浏览器发送一个请求，你访问到那个网站对应的机器，然后人家再给你一个响应的全过程。

现在我们先假设，我们给电脑设置了几个东西：

ip地址：192.168.31.37

子网掩码：255.255.255.0

网关地址：192.168.31.1

DNS地址：8.8.8.8

完了，我们打开一个浏览器，请求www.baidu.com地址，这个时候找DNS服务器，DNS服务器解析域名之后，返回一个ip地址，比如172.194.26.108。接着会判断两个ip地址是不是一个子网的，用子网掩码255.255.255.0，对两个ip地址做与运算，拿到192.168.31.0和172.194.26.0，明显不是一个子网的。

那就得发送一个数据包给网关，其实你就认为是我们的路由器吧，就是192.168.31.1，而且我们是可以拿到网关ip地址的mac地址的，现在我们从应用层出发，通过浏览器访问一个网站，是走应用层的http协议的。

既然要把浏览器发出的请求打包成数据包，要把哪些东西给放到数据包中去呢？应该是把http请求给打包到数据包中去。

http协议分为几个部分：

请求方法 + URL地址 + http版本：比如GET http://172.194.26.108/test HTTP/1.1，类似这种

请求头，类似下面这种：

Host: upload.jiangsu.io

Proxy-Connection: keep-alive

User-Agent: Mozilla/5.0

空行

请求体，比如常见的可以放一个json

这就构成了一个http请求报文

浏览器请求一个地址，先按照应用层的http协议，封装一个应用层数据包，数据包里就放了http请求报文

这个时候会将这个http请求报文打包成一个数据包，仅仅只是数据包的数据部分，此时是数据包是没有头的。上面根据http协议搞一个http请求报文，然后搞一个数据包出来，就是网络模型中到的应用层干的事儿了。

接着就是跑传输层来了，这个层是tcp协议，这个tcp协议会让你设置端口，发送方的端口随机选一个，接收方的端口一般是默认的80端口。这个时候，会把应用层数据包给封装到tcp数据包中去，而且会加一个tcp头，这个tcp数据包是对应一个tcp头的，这个tcp头里就放了端口号信息。

接着跑到网络层来了，走ip协议，这个时候会把tcp头和tcp数据包，放到ip数据包里去，然后再搞一个ip头，ip头里本机和目标机器的ip地址。这里本机ip地址是192.168.31.37，目标机器是172.194.26.108。

应为，通过ip协议，可以判断说，两个ip地址不是在一个子网内的，所以此时只能将数据包先通过以太网协议广播到网关上去，通过网关再给他发送出去

接着是数据链路层，这块走以太网协议，这里是把ip头和ip数据包封到以太网数据包里去，然后再加一个以太网数据包的头，头里放了本机网卡mac地址，和网关的mac地址。但是以太网数据包的限制是1500个字节，但是假设这个时候ip数据包都5000个字节了，那么需要将ip数据包切割一下。

这个时候一个以太网数据包要切割为4个数据包，每个数据包包含了以太网头、ip头和切割后的ip数据包，4个数据包的大小分别是1500，1500,1500，560。ip头里包含了每个数据包的序号。

这4个以太网数据包都会通过交换机发到你的网关上，然后你的路由器是可以联通别的子网的，这个是时候你的路由器就会转发到别的子网的可能也是某个路由器里去，然后以此类推吧，N多个路由器或者你叫网关也行，N多个网关转发之后，就会跑到百度的某台服务器，接收到4个以太网数据包。

百度服务器接收到4个以太网数据包以后，根据ip头的序号，把4个以太网数据包里的ip数据包给拼起来，就还原成一个完整的ip数据包了。接着就从ip数据包里面拿出来tcp数据包，再从tcp数据包里取出来http数据包，读取出来http数据包里的各种协议内容，接着就是做一些处理，然后再把响应结果封装成htp响应报文，封装在http数据包里，再一样的过程，封装tcp数据包，封装ip数据包，封装以太网数据包，接着通过网关给发回去。

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/003\_网络层：IP协议、子网划分以及子网掩码是什么（上）/笔记.docx**

1、面试题

说一下TCP/IP的四层网络模型？那么七层网络模型呢？

2、面试官心里分析

结构化面试：不是瞎面，cs基础（computer science），java基础，j2ee，mysql，linux，tomcat，分布式，高并发，高可用，系统设计，数据结构和算法，软素质，工程素质

坦白讲，一些大的公司，计算机基础必面，尤其是针对薪资30k以内的工程师，因为薪资30k以内，你还是要干活儿的吧，还没上升到就设计架构就可以的程度吧，你还没到那个高度吧。所以只要你干活儿，你就不可避免要跟机器、网络、cpu、磁盘、内存，成天打交道。而线上系统，计算基础的一些东西，网络、cpu、磁盘、内存，都是关联很大的，比如说你线上系统会不会因为网络故障导致一些问题？cpu负载达到100%了咋办？磁盘读写很慢快满了咋办？内存使用率过高咋办？

你起码得有一套自己的计算机功底去支撑你玩儿线上系统吧。所以很多人呢，都说计算机基础没啥用，那这个话呢，也对，也不对。对就在于，你如果毕业出来干简单的crud，这些东西你确实不需要；不对就在于，你如果当个高工，带几个小弟干高并发有压力的线上系统，机器负载很高，很容易出问题，结果你连机器都不敢摸，或者也不知道怎么摸，那不是尴尬了么。

所以说，计算机基础，网络、磁盘、cpu、内存，还是得会一点儿基础的

作为一个大公司的面试官，一定会考察你这些东西

3、面试题剖析

（1）简单介绍一下

四层模型和七层模型可以一块儿聊

为啥需要协议？没有协议，各个电脑厂商，比如IBM、苹果啥的，都弄自己的协议，结果就苹果电脑和苹果电脑自己可以通信，和IBM电脑就不可以通信，这不是尴尬么。所以搞一个国际通行的协议，大家都按照这个来，所有电脑都可以通信，不是很好么。

OSI七层模型，就是搞一个标准的网络模型出来，大家都按照这个来走，那么大家都有统一的规范。OSI七层模型，是应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。TCP/IP四层模型，数据链路层、网络层、传输层、应用层。

（2）从底向上的网络分层

1）物理层

物理层，物理层干啥的，就是电脑之间要联网，一般咋弄？类似于说，你有台电脑，现在要联网，咋联？以前N年前，大家记不记得都是在电脑上插根线是吧，然后才能上网，结果现在就是联个wifi就行了，还有中国美国之前联网靠的是海底的光缆。所以物理层就指的这个，就是怎么把各个电脑给联结起来，形成一个网络，这就是物理层的含义，物理层负责传输0和1的电路信号。学过一些计算机的同学，计算机的最最底层，就是0/1，电信号。

2）数据链路层

数据链路层，物理层给各个电脑连接起来了，还传输最底层的0和1电路信号，关键不行啊，你得定义清楚哪些0和1分为一组，这些信号啥意思？这才能进行通信。所以数据链路层就干这事儿，定义一下电路信号咋分组。

00000011（从电脑1出发，要到电脑2去）

00101（从电脑1出发，要到电脑3去）

0101（从电脑2触发，要到电脑4去）

01（从电脑3出发，要到电脑5去）

很多年前，每个公司都定义自己的电路信号分组方式，但是后来出来了以太网协议，以太网。一组电信号是一个数据包，叫一个帧（frame），每个帧分成两个部分，标头（head）和数据（data），标头包含一些说明性的东西，比如说发送者、接收者和数据类型之类的。

每台电脑要往另外一台电脑发送数据，一堆0/1电路信号，封装成数据包，包含头和数据，头里包含了从哪儿来到哪儿去，必须从一台电脑的一个网卡，发送到另外一个电脑的一个网卡，所以以太网发送的数据包必须得指定，目标电脑的网卡的mac地址。

以太网规定了，每个网卡必须得包含一个mac地址，mac地址就是这个网卡的唯一标识，

以太网协议规定了，接入网络里的所有设备，都得有个网卡，以太网协议里的那个数据包，在数据链路层传输的数据包，必须从一个电脑的网卡传输到另外一个电脑的网卡，而这个网卡地址就叫做所谓的mac地址。每块网卡出厂的时候，就有一个唯一的mac地址，48位的二进制，但是一般用12个16进制数字表示，前6个16进制是厂商编号，后6个16进制是网卡流水号。

windows上，ipconfig /all，看看物理地址，就是mac地址，7C-67-A2-20-AB-5C

所以在以太网里传输数据包的时候，必须指定接收者的mac地址才能传输数据。

但是以太网的数据包怎么从一个mac地址发送到另一个mac地址？这个不是精准推送的，以太网里面，如果一个电脑发个数据包出去，会广播给局域网内的所有电脑设备的网卡，然后每台电脑都从数据包里获取接收者的mac地址，跟自己的mac地址对比一下，如果一样，就说明这是发给自己的数据包。

但是上面这种广播的方式，仅仅针对一个子网（局域网）内的电脑，会广播，否则一个电脑不能广播数据包给全世界所有的其他电脑吧，是仅仅广播给一个子网里面的电脑的。

3）网络层

上面说到，子网内的电脑，通过以太网发个数据包，对局域网内的电脑，是广播出去的。那么怎么知道哪些电脑在一个子网内呢？这就得靠网络层了，这里就有一套IP地址，IP地址就可以让我们区分哪些电脑是一个子网的。

网络层里有IP协议，IP协议定义的地址就叫做IP地址。IP地址有IPv4和IPv6两个版本，目前广泛使用的是IPv4，是32个二进制数字组成的，但是一般用4个十进制数字表示，范围从0.0.0.0到255.255.255.255之间。

每台计算机，都会分配一个ip地址，ip地址的前24位（就是前面3个十进制数字），代表了网络，后8位（就是最后1个十进制数字），代表了主机。如果几台电脑是一个子网的，那么前面的3个十进制数字一定是一样的。举个例子，大家平时做实验，玩儿虚拟机吧，自己win上开几个linux虚拟机，你会发现，win上的ip地址可能是192.168.0.103，然后几个虚拟机的ip地址是192.168.0.182，192.168.0.125，192.168.0.106，类似这样的。这个win机器和几个虚拟机，前面3个十进制数字都是192.168.0，就代表大家是一个子网内的，最后那个数字是这个子网的不同主机的编号。

但是实际上上面就是举个例子，其实单单从ip地址是看不出来哪些机器是一个子网的，因为从10进制是判断不出来的。需要通过ip地址的二进制来判断，结合一个概念来判断，叫做子网掩码。比如说ip地址是192.168.56.1，子网掩码是255.255.255.0。知道了子网掩码之后，如果要判断两个ip地址是不是一个子网的，就分别把两个ip地址和自己的子网掩码进行二进制的与运算，与运算之后，比较一下代表网络的那部分。

192.168.56.1和192.168.32.7，判断是不是一个子网的，拿子网掩码255.255.255.0，跟两个ip地址的二进制做与运算

11000000.10101000.00111000.00000001

11111111.11111111.11111111.00000000

子网掩码的二进制是：11111111.11111111.11111111.00000000，然后就跟ip地址的二进制做与好了，通过二进制来比较网络部分的地址是不是一模一样的。

有了网络层的ip地址之后，两台在子网内的电脑终于可以通过广播+mac地址判断来传输数据包进行通信了。

但是如果发现要接受数据包的计算机不在子网内，那么就不能通过广播来发送数据包，需要通过路由来发送数据包。看到路由，就想到了路由器了，对了，路由器大家都熟悉吧，自己平时也会去买对吧，比如小米的路由器啥的，家里上网一般都会弄个路由器对吧，ok。路由器负责将多个子网进行连接，因为比如你在自己家里，其实你就只是你自己的一个子网，你要是访问网站啥的，是跟那个网站机器所在的子网进行通信。

每个电脑都可以搞多个网卡的，不是只有一个网卡，一般笔记本电脑都有以太网网卡和wifi网卡，发送数据包的时候要决定走哪个网卡。路由器，其实就是配置了多个网卡的一个专用设备，可以通过不同的网卡接入不同的网络。

网关其实是就是路由器的一种，运作在网络层，这个概念不多解释了，大家可以就把路由器上的ip地址认为是网关，路由器上每个网卡都有mac地址和对应的ip地址。路由器虽然有mac地址，但是不能通过mac地址寻址的，必须通过ip地址寻址，所以路由器其实是工作在网络层的设备。

网络交换机，也是一种设备，是工作在数据链路层的，路由器是工作在网路层的。网络交换机是通过mac地址来寻址和传输数据包的；但是路由器是通过ip地址寻址和传输数据包的。网络交换机主要用在局域网的通信，一般你架设一个局域网，里面的电脑通信是通过数据链路层发送数据包，通过mac地址来广播的，广播的时候就是通过网络交换机这个设备来把数据广播到局域网内的其他机器上去的；路由器一般用来让你连入英特网。

LAN，就是local area network，就是局域网；WAN，就是wide area network，就是广域网。WLAN是wireless local area network，就是无线局域网，也就是wifi，在局域网内，直接通过wifi无线联网。

家里的路由器是包含了交换机和路由的两个功能的，如果是连接到局域网内的设备就把线插LAN那儿；如果是连接到英特网，就把线插在WAN那儿。

这儿给大家举个例子，就是两个局域网之间，如果要是通过一个路由器进行通信的话，是怎么弄的。

大概过程就是，路由器配置了两块网卡，每个网卡可以连到一个局域网内。

局域网1内的电脑，要发送数据包到局域网2内的电脑，在数据包里写上自己的ip地址和对方的ip地址。但是他们俩不在一个局域网内，于是局域网1内的电脑，先通过交换机将数据包发送给路由器，这个过程需要将路由器的一块网卡的ip地址对应的mac地址写到数据包的头部，然后才能通过交换机广播出去，路由器接收到之后比较自己一块网卡的mac地址，就知道是来找自己的。

接着路由器接收到数据包之后，就会在局域网2内，将目标机器的ip地址对应的mac地址写入头部，接着再次通过交换机发送广播通知，发送给局域网2内的电脑。

一个局域网内的每台机器都有自己的ARP cache，这个ARP就是用来在一个局域网内让各个设备都知道每个设备的ip地址和mac地址的对应关系的，一般就是某个机器发送广播通知自己的ip地址和mac地址的对应关系，然后每个机器给他一个回应。以此类推，大家都互相这样广播一把，ip地址和mac地址的对应关系，大家不就都知道了。

一个ip地址对应着一个mac地址

所以大家在上面可以看到，一个子网内的机器之间通信，就是在数据包里写上对方的mac地址，然后交换机广播出去ok了；但是如果是跨子网的通信，就是写上对方的ip地址，然后先通过mac地址广播到路由器，让路由器再根据另外一个子网的ip地址转换为mac地址，通过另外一个子网的交换机广播过去。就这个意思。

4）传输层

上面我们大概明白了通过网络层的ip地址怎么划分出来一个一个的子网，然后在子网内部怎么通过mac地址广播通信；跨子网的时候，怎么通过ip地址 -> mac地址 -> 交换机 -> 路由器 -> ip地址 -> mac地址 -> 交换机的方式来通过路由器进行通信。

但是这里还有一个问题，就是一台机器上，是很多个程序用一个网卡进行网络通信的，比如说浏览器、QQ、视频直播，这些软件都用了一个网卡往外面发送数据，然后从网卡接收数据，对吧。

所以还需要一个端口号的概念，就是你得发送数据包到某个机器的一个网卡的某个端口上去，然后那个机器上监听那个端口的程序，就可以提取发送到这个端口的数据，知道是自己的数据。端口号是0~65536的范围内，0~1023被系统占用了，别的应用程序就用1024以上的端口就ok了。

电脑1，是在端口48362监听的，通过网卡发送了一条数据 -> 电脑2的ip地址的20386这个端口 -> 电脑2的上面的某个QQ，监听着20386的端口 -> 电脑2的网卡接收到一条数据之后，发现人家找的是20386这个端口，就去找谁哪个哥儿们在监听20386端口，QQ在监听，我就把这个网卡过来的数据，传递给QQ，通过端口知道，哪条数据是给你的

所以其实大家会发现一点，网络层，是基于ip协议，进行主机和主机间的寻址和通信的，然后传输层，其实是建立某个主机的某个端口，到另外一个主机的某个端口的连接和通信的。这个通信，就是通过socket来实现的，通过socket就可以基于tcp/ip协议完成刚才上面说的一系列的比如基于ip地址和mac地址转换和寻址啊，通过路由器通信啊之类的，而且会建立一个端口到另外一个端口的连接。

udp和tcp都是传输层的协议，作用就是在数据包里加入端口号，可以通过端口号进行点对点的通信了。udp协议是不可靠的，发出去人家收到没有就不知道了；tcp协议是可靠的，要求三次握手，而且要求人家接收到数据必须回复你。

传输层的tcp协议，仅仅只是规定了一套基于端口的点对点的通信协议，包括如何建立连接，如何发送和读取消息，但是实际上如果你要基于tcp协议来开发，你一般是用socket，java socket网络编程

5）应用层

通过传输层的tcp协议可以传输数据，但是人家收到数据之后，怎么来解释？比如说收到个邮件你怎么处理？收到个网页你怎么处理？类似这个意思，所以针对各种不同的应用，邮件、网页之类的，都是定义不同的应用层协议的。这个应用层，我们就假设综合了会话层、表示层和应用层了，3层合成1层。

电脑1走tcp协议发送了一段东西过来，发送到电脑2的20386端口

GET http://localhost:8080/ http/1.1

key:valuel

key:value

电脑2走tcp协议读取到了属于自己这个20386端口 的一段数据

GET http://localhost:8080/ http/1.1

key:valuel

key:value

发送了一段响应

200

key;value

key:value

又通过底层的tcp发了出去，电脑1的30987端口，ip

电脑1，网卡，走以太网协议收到一个数据包

200

key;value

key:value

比如最常见的，应用层的协议就是http协议，进行网络通信。

4层：数据链路层（以太网协议），网络层（ip协议），传输层（tcp协议），应用层（http协议）

7层：物理层（网线，海底光缆，传递0/1电路信号），会话层、表示层、应用层 -> 应用层

然后我们看下自己的网络设置，一般包含了ip地址、子网掩码、网关地址、DNS地址。前面3个我们其实都知道啥意思了。ip地址和子网掩码用来划分子网的，判断哪些ip地址在一个子网内。同时你的ip地址和mac地址关联起来的，唯一定位了你的网卡。网关地址，你就认为是路由器上的那个网卡的ip地址吧，路由器的网卡也有mac地址，mac地址对应了一个ip地址。

DNS地址是啥呢？Domain Name System。因为我们一般定位是通过ip地址+mac地址+端口号来定位一个通信目标的，但是如果在浏览器上输入一个www.baidu.com，咋整？这个时候是先把www.baidu.com发给DNS服务器，然后DNS服务器告诉你www.baidu.com对应的ip地址的。

8.8.8.8

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/005\_传输层：TCP协议、Socket编程以及端口号是什么/笔记.docx**

1、面试题

说一下TCP/IP的四层网络模型？那么七层网络模型呢？

2、面试官心里分析

结构化面试：不是瞎面，cs基础（computer science），java基础，j2ee，mysql，linux，tomcat，分布式，高并发，高可用，系统设计，数据结构和算法，软素质，工程素质

坦白讲，一些大的公司，计算机基础必面，尤其是针对薪资30k以内的工程师，因为薪资30k以内，你还是要干活儿的吧，还没上升到就设计架构就可以的程度吧，你还没到那个高度吧。所以只要你干活儿，你就不可避免要跟机器、网络、cpu、磁盘、内存，成天打交道。而线上系统，计算基础的一些东西，网络、cpu、磁盘、内存，都是关联很大的，比如说你线上系统会不会因为网络故障导致一些问题？cpu负载达到100%了咋办？磁盘读写很慢快满了咋办？内存使用率过高咋办？

你起码得有一套自己的计算机功底去支撑你玩儿线上系统吧。所以很多人呢，都说计算机基础没啥用，那这个话呢，也对，也不对。对就在于，你如果毕业出来干简单的crud，这些东西你确实不需要；不对就在于，你如果当个高工，带几个小弟干高并发有压力的线上系统，机器负载很高，很容易出问题，结果你连机器都不敢摸，或者也不知道怎么摸，那不是尴尬了么。

所以说，计算机基础，网络、磁盘、cpu、内存，还是得会一点儿基础的

作为一个大公司的面试官，一定会考察你这些东西

3、面试题剖析

（1）简单介绍一下

四层模型和七层模型可以一块儿聊

为啥需要协议？没有协议，各个电脑厂商，比如IBM、苹果啥的，都弄自己的协议，结果就苹果电脑和苹果电脑自己可以通信，和IBM电脑就不可以通信，这不是尴尬么。所以搞一个国际通行的协议，大家都按照这个来，所有电脑都可以通信，不是很好么。

OSI七层模型，就是搞一个标准的网络模型出来，大家都按照这个来走，那么大家都有统一的规范。OSI七层模型，是应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。TCP/IP四层模型，数据链路层、网络层、传输层、应用层。

（2）从底向上的网络分层

1）物理层

物理层，物理层干啥的，就是电脑之间要联网，一般咋弄？类似于说，你有台电脑，现在要联网，咋联？以前N年前，大家记不记得都是在电脑上插根线是吧，然后才能上网，结果现在就是联个wifi就行了，还有中国美国之前联网靠的是海底的光缆。所以物理层就指的这个，就是怎么把各个电脑给联结起来，形成一个网络，这就是物理层的含义，物理层负责传输0和1的电路信号。学过一些计算机的同学，计算机的最最底层，就是0/1，电信号。

2）数据链路层

数据链路层，物理层给各个电脑连接起来了，还传输最底层的0和1电路信号，关键不行啊，你得定义清楚哪些0和1分为一组，这些信号啥意思？这才能进行通信。所以数据链路层就干这事儿，定义一下电路信号咋分组。

00000011（从电脑1出发，要到电脑2去）

00101（从电脑1出发，要到电脑3去）

0101（从电脑2触发，要到电脑4去）

01（从电脑3出发，要到电脑5去）

很多年前，每个公司都定义自己的电路信号分组方式，但是后来出来了以太网协议，以太网。一组电信号是一个数据包，叫一个帧（frame），每个帧分成两个部分，标头（head）和数据（data），标头包含一些说明性的东西，比如说发送者、接收者和数据类型之类的。

每台电脑要往另外一台电脑发送数据，一堆0/1电路信号，封装成数据包，包含头和数据，头里包含了从哪儿来到哪儿去，必须从一台电脑的一个网卡，发送到另外一个电脑的一个网卡，所以以太网发送的数据包必须得指定，目标电脑的网卡的mac地址。

以太网规定了，每个网卡必须得包含一个mac地址，mac地址就是这个网卡的唯一标识，

以太网协议规定了，接入网络里的所有设备，都得有个网卡，以太网协议里的那个数据包，在数据链路层传输的数据包，必须从一个电脑的网卡传输到另外一个电脑的网卡，而这个网卡地址就叫做所谓的mac地址。每块网卡出厂的时候，就有一个唯一的mac地址，48位的二进制，但是一般用12个16进制数字表示，前6个16进制是厂商编号，后6个16进制是网卡流水号。

windows上，ipconfig /all，看看物理地址，就是mac地址，7C-67-A2-20-AB-5C

所以在以太网里传输数据包的时候，必须指定接收者的mac地址才能传输数据。

但是以太网的数据包怎么从一个mac地址发送到另一个mac地址？这个不是精准推送的，以太网里面，如果一个电脑发个数据包出去，会广播给局域网内的所有电脑设备的网卡，然后每台电脑都从数据包里获取接收者的mac地址，跟自己的mac地址对比一下，如果一样，就说明这是发给自己的数据包。

但是上面这种广播的方式，仅仅针对一个子网（局域网）内的电脑，会广播，否则一个电脑不能广播数据包给全世界所有的其他电脑吧，是仅仅广播给一个子网里面的电脑的。

3）网络层

上面说到，子网内的电脑，通过以太网发个数据包，对局域网内的电脑，是广播出去的。那么怎么知道哪些电脑在一个子网内呢？这就得靠网络层了，这里就有一套IP地址，IP地址就可以让我们区分哪些电脑是一个子网的。

网络层里有IP协议，IP协议定义的地址就叫做IP地址。IP地址有IPv4和IPv6两个版本，目前广泛使用的是IPv4，是32个二进制数字组成的，但是一般用4个十进制数字表示，范围从0.0.0.0到255.255.255.255之间。

每台计算机，都会分配一个ip地址，ip地址的前24位（就是前面3个十进制数字），代表了网络，后8位（就是最后1个十进制数字），代表了主机。如果几台电脑是一个子网的，那么前面的3个十进制数字一定是一样的。举个例子，大家平时做实验，玩儿虚拟机吧，自己win上开几个linux虚拟机，你会发现，win上的ip地址可能是192.168.0.103，然后几个虚拟机的ip地址是192.168.0.182，192.168.0.125，192.168.0.106，类似这样的。这个win机器和几个虚拟机，前面3个十进制数字都是192.168.0，就代表大家是一个子网内的，最后那个数字是这个子网的不同主机的编号。

但是实际上上面就是举个例子，其实单单从ip地址是看不出来哪些机器是一个子网的，因为从10进制是判断不出来的。需要通过ip地址的二进制来判断，结合一个概念来判断，叫做子网掩码。比如说ip地址是192.168.56.1，子网掩码是255.255.255.0。知道了子网掩码之后，如果要判断两个ip地址是不是一个子网的，就分别把两个ip地址和自己的子网掩码进行二进制的与运算，与运算之后，比较一下代表网络的那部分。

192.168.56.1和192.168.32.7，判断是不是一个子网的，拿子网掩码255.255.255.0，跟两个ip地址的二进制做与运算

11000000.10101000.00111000.00000001

11111111.11111111.11111111.00000000

子网掩码的二进制是：11111111.11111111.11111111.00000000，然后就跟ip地址的二进制做与好了，通过二进制来比较网络部分的地址是不是一模一样的。

有了网络层的ip地址之后，两台在子网内的电脑终于可以通过广播+mac地址判断来传输数据包进行通信了。

但是如果发现要接受数据包的计算机不在子网内，那么就不能通过广播来发送数据包，需要通过路由来发送数据包。看到路由，就想到了路由器了，对了，路由器大家都熟悉吧，自己平时也会去买对吧，比如小米的路由器啥的，家里上网一般都会弄个路由器对吧，ok。路由器负责将多个子网进行连接，因为比如你在自己家里，其实你就只是你自己的一个子网，你要是访问网站啥的，是跟那个网站机器所在的子网进行通信。

每个电脑都可以搞多个网卡的，不是只有一个网卡，一般笔记本电脑都有以太网网卡和wifi网卡，发送数据包的时候要决定走哪个网卡。路由器，其实就是配置了多个网卡的一个专用设备，可以通过不同的网卡接入不同的网络。

网关其实是就是路由器的一种，运作在网络层，这个概念不多解释了，大家可以就把路由器上的ip地址认为是网关，路由器上每个网卡都有mac地址和对应的ip地址。路由器虽然有mac地址，但是不能通过mac地址寻址的，必须通过ip地址寻址，所以路由器其实是工作在网络层的设备。

网络交换机，也是一种设备，是工作在数据链路层的，路由器是工作在网路层的。网络交换机是通过mac地址来寻址和传输数据包的；但是路由器是通过ip地址寻址和传输数据包的。网络交换机主要用在局域网的通信，一般你架设一个局域网，里面的电脑通信是通过数据链路层发送数据包，通过mac地址来广播的，广播的时候就是通过网络交换机这个设备来把数据广播到局域网内的其他机器上去的；路由器一般用来让你连入英特网。

LAN，就是local area network，就是局域网；WAN，就是wide area network，就是广域网。WLAN是wireless local area network，就是无线局域网，也就是wifi，在局域网内，直接通过wifi无线联网。

家里的路由器是包含了交换机和路由的两个功能的，如果是连接到局域网内的设备就把线插LAN那儿；如果是连接到英特网，就把线插在WAN那儿。

这儿给大家举个例子，就是两个局域网之间，如果要是通过一个路由器进行通信的话，是怎么弄的。

大概过程就是，路由器配置了两块网卡，每个网卡可以连到一个局域网内。

局域网1内的电脑，要发送数据包到局域网2内的电脑，在数据包里写上自己的ip地址和对方的ip地址。但是他们俩不在一个局域网内，于是局域网1内的电脑，先通过交换机将数据包发送给路由器，这个过程需要将路由器的一块网卡的ip地址对应的mac地址写到数据包的头部，然后才能通过交换机广播出去，路由器接收到之后比较自己一块网卡的mac地址，就知道是来找自己的。

接着路由器接收到数据包之后，就会在局域网2内，将目标机器的ip地址对应的mac地址写入头部，接着再次通过交换机发送广播通知，发送给局域网2内的电脑。

一个局域网内的每台机器都有自己的ARP cache，这个ARP就是用来在一个局域网内让各个设备都知道每个设备的ip地址和mac地址的对应关系的，一般就是某个机器发送广播通知自己的ip地址和mac地址的对应关系，然后每个机器给他一个回应。以此类推，大家都互相这样广播一把，ip地址和mac地址的对应关系，大家不就都知道了。

一个ip地址对应着一个mac地址

所以大家在上面可以看到，一个子网内的机器之间通信，就是在数据包里写上对方的mac地址，然后交换机广播出去ok了；但是如果是跨子网的通信，就是写上对方的ip地址，然后先通过mac地址广播到路由器，让路由器再根据另外一个子网的ip地址转换为mac地址，通过另外一个子网的交换机广播过去。就这个意思。

4）传输层

TCP / IP协议，其实工作的时候是放在一起来工作的

上面我们大概明白了通过网络层的ip地址怎么划分出来一个一个的子网，然后在子网内部怎么通过mac地址广播通信；跨子网的时候，怎么通过ip地址 -> mac地址 -> 交换机 -> 路由器 -> ip地址 -> mac地址 -> 交换机的方式来通过路由器进行通信。

但是这里还有一个问题，就是一台机器上，是很多个程序用一个网卡进行网络通信的，比如说浏览器、QQ、视频直播，这些软件都用了一个网卡往外面发送数据，然后从网卡接收数据，对吧。

所以还需要一个端口号的概念，就是你得发送数据包到某个机器的一个网卡的某个端口上去，然后那个机器上监听那个端口的程序，就可以提取发送到这个端口的数据，知道是自己的数据。端口号是0~65536的范围内，0~1023被系统占用了，别的应用程序就用1024以上的端口就ok了。

电脑1，是在端口48362监听的，通过网卡发送了一条数据 -> 电脑2的ip地址的20386这个端口 -> 电脑2的上面的某个QQ，监听着20386的端口 -> 电脑2的网卡接收到一条数据之后，发现人家找的是20386这个端口，就去找谁哪个哥儿们在监听20386端口，QQ在监听，我就把这个网卡过来的数据，传递给QQ，通过端口知道，哪条数据是给你的

所以其实大家会发现一点，网络层，是基于ip协议，进行主机和主机间的寻址和通信的，然后传输层，其实是建立某个主机的某个端口，到另外一个主机的某个端口的连接和通信的。这个通信，就是通过socket来实现的，通过socket就可以基于tcp/ip协议完成刚才上面说的一系列的比如基于ip地址和mac地址转换和寻址啊，通过路由器通信啊之类的，而且会建立一个端口到另外一个端口的连接。

udp和tcp都是传输层的协议，作用就是在数据包里加入端口号，可以通过端口号进行点对点的通信了。udp协议是不可靠的，发出去人家收到没有就不知道了；tcp协议是可靠的，要求三次握手，而且要求人家接收到数据必须回复你。

传输层的tcp协议，仅仅只是规定了一套基于端口的点对点的通信协议，包括如何建立连接，如何发送和读取消息，但是实际上如果你要基于tcp协议来开发，你一般是用socket，java socket网络编程

第一个点：端口号，允许了两个电脑上程序对程序的通信

第二个点：TCP就是一个我们平时主要用于网络编程通信的协议了，上层的协议，平时经常用到的一个协议，三次握手建立连接，流式拆包发送数据，四次握手断开连接

TCP网路编程是什么都不知道的话，百度一下，不可能，主要参加过4个月的java就业培训的同学，都应该知道Socket编程，就是直接使用TCP协议进行程序对程序的网络通信，建立连接，TCP三次握手

两个程序之间互相交换几次数据，程序1先发送一个数据给程序2，程序2返回一个结果，以此类推，反复进行3次，大家互相交换一些数据，交换成功了以后，就认为两个程序之间建立了TCP连接

三次握手，交换了3次数据，底层的网络模型里面，具体在干什么？

电脑上的程序，要先发送一个数据给服务器上的程序，这个数据实际上来说是按照tcp协议的规范封装的一个tcp包，tcp包头和具体的数据

互相发送TCP包，包头里面就包含了TCP三次握手互相之间要设置和交换的一些数据，具体交换哪些数据，可以看后面的一讲，专门讲解TCP三次握手的视频

Socket编程，连接，以IO流的形式，发送数据流，以流的方式从对方服务器再接收一段数据回来，这个过程也是按照TCP协议在进行通信，对应到底层的网络模型，到底是个什么意思呢？

你发送的数据，按照固定的大小，封装成一个一个TCP包，每个TCP包封装在IP包里，封装在以太网包，最后通过网卡出去，走底层的以太网协议 + IP协议，子网内 + 跨子网，把你的一个一个连续的TCP包通过这种方式给传输到对方的服务器上去

对方的服务器接收到N多个TCP包，最后组装起来就是你以IO流的方式发送 过来的一段数据，人家就会反过来，通过IO流的方式返回一个一个TCP包的数据给你，一样的通过底层的IP协议 + 以太网协议，来传输

如果TCP通信结束之后，他实际上来说就会断开TCP连接，四次挥手，互相交换四次数据，就是互相传输4个TCP包，把socket相关的数据给删除掉，连接其实就是断开了

传输TCP包的时候，对应到底层，如果一个TCP包比较大的话，此时是不能直接用一个以太网包来传输，底层的以太网包也是有一个固定的大小的，会在底层把你的TCP包再次拆分为多个以太网包

建立TCP连接，到底是什么意思呢？

就是以TCP的协议的规范，定义的是说一台电脑的一个端口上的程序，有一个socket；跟另外一个服务器上的某个端口上的程序，另外一个socket，之间，通过交换几次TCP包，大家交换了一下数据，最后就在各自的电脑 / 服务器上，建立了对应的socket，就代表说，建立了一个TCP连接

没有TCP三次握手建立连接的话，两个程序之间不知道对方的一些基本情况，没法有效的进行通信，所以通过三次握手建立连接，交换一些数据，就知道对方是什么情况了，此时就可以进行通信了

5）应用层

通过传输层的tcp协议可以传输数据，但是人家收到数据之后，怎么来解释？比如说收到个邮件你怎么处理？收到个网页你怎么处理？类似这个意思，所以针对各种不同的应用，邮件、网页之类的，都是定义不同的应用层协议的。这个应用层，我们就假设综合了会话层、表示层和应用层了，3层合成1层。

电脑1走tcp协议发送了一段东西过来，发送到电脑2的20386端口

GET http://localhost:8080/ http/1.1

key:valuel

key:value

电脑2走tcp协议读取到了属于自己这个20386端口 的一段数据

GET http://localhost:8080/ http/1.1

key:valuel

key:value

发送了一段响应

200

key;value

key:value

又通过底层的tcp发了出去，电脑1的30987端口，ip

电脑1，网卡，走以太网协议收到一个数据包

200

key;value

key:value

比如最常见的，应用层的协议就是http协议，进行网络通信。

4层：数据链路层（以太网协议），网络层（ip协议），传输层（tcp协议），应用层（http协议）

7层：物理层（网线，海底光缆，传递0/1电路信号），会话层、表示层、应用层 -> 应用层

然后我们看下自己的网络设置，一般包含了ip地址、子网掩码、网关地址、DNS地址。前面3个我们其实都知道啥意思了。ip地址和子网掩码用来划分子网的，判断哪些ip地址在一个子网内。同时你的ip地址和mac地址关联起来的，唯一定位了你的网卡。网关地址，你就认为是路由器上的那个网卡的ip地址吧，路由器的网卡也有mac地址，mac地址对应了一个ip地址。

DNS地址是啥呢？Domain Name System。因为我们一般定位是通过ip地址+mac地址+端口号来定位一个通信目标的，但是如果在浏览器上输入一个www.baidu.com，咋整？这个时候是先把www.baidu.com发给DNS服务器，然后DNS服务器告诉你www.baidu.com对应的ip地址的。

8.8.8.8

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/008\_一道经典的面试题：你能聊聊TCP三次握手建立连接的过程吗？/笔记.docx**

1、面试题

TCP三次握手和四次握手的工作流程是什么（画一下流程图）？为什么不是五次握手或者两次握手？

2、面试官心里分析

这个问题相当经典，大家可别以为就是考察应届生的，实际上在普通社招java面试中，一些大公司，很喜欢考察这个问题，尤其是后面第二个追加问题，让你聊聊为啥必须是三次握手，而不是两次呢？

3、面试题剖析

（1）tcp三次握手过程

通过传输层的tcp协议建立网络连接的时候，其实走的是三次握手的过程

建立三次握手的时候，TCP报头用到了下面几个东西，ACK、SYN、FIN。

第一次握手，客户端发送连接请求报文，此时SYN=1、ACK=0，这就是说这是个连接请求，seq = x，接着客户端处于SYN\_SENT状态，等待服务器响应。

第二次握手，服务端收到SYN=1的请求报文，需要返回一个确认报文，ack = x + 1，SYN=1，ACK = 1，seq = y，发送给客户端，自己处于SYN\_RECV状态。

第三次握手，客户端收到了报文，将ack = y + 1，ACK = 1，seq = x + 1

其实三次握手说白了，就是来回来去三次请求，每次请求带上一堆TCP报文头，根据报文头是否正确，就是越好的协议来建立连接。简单说就是这样。

（2）为啥不是2次或者4次握手呢？

假设两次握手就ok了，要是客户端第一次握手过去，结果卡在某个地方了，没到服务端；完了客户端再次重试发送了第一次握手过去，服务端收到了，ok了，大家来回来去，三次握手建立了连接。

结果，尴尬的是，后来那个卡在哪儿的老的第一次握手发到了服务器，服务器直接就返回一个第二次握手，这个时候服务器开辟了资源准备客户端发送数据啥的，结果呢？客户端根本就不会理睬这个发回去的二次握手，因为之前都通信过了。

但是如果是三次握手，那个二次握手发回去，客户端发现根本不对，就会发送个复位的报文过去，让服务器撤销开辟的资源，别等着了。

因为3次握手就够了，不需要4次或者5次浪费资源了。

（3）tcp断开连接的4次挥手

第一次挥手，客户端发送报文，FIN=1，seq=u，此时进入FIN-WAIT-1状态

第二次挥手，服务端收到报文，这时候进入CLOSE\_WATI状态，返回一个报文，ACK=1，ack=u+1，seq=v。客户端收到这个报文之后，直接进入FIN-WAIT-2状态，此时客户端到服务端的连接就释放了。

第三次挥手，服务端发送连接释放报文，FIN=1，ack=u+1，seq=w，服务端进入LAST-ACK状态

第四次挥手，客户端收到连接释放报文之后，发应答报文，ACK=1，ack=w+1，seq=u+1，进入TIME\_WAIT状态，等待一会儿客户端进入CLOSED状态，服务端收到报文之后就进入CLOSED状态。

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/011\_一道经典的面试题：Socket网络编程跟TCP IP协议的关系是什么/笔记.docx**

1、面试题

说说socket通信的原理？

2、面试官心里分析

其实不知道大家发现没有，网络相关的问题，都是围绕着所谓的七层模型，或者是四层模型去走的。聊完四层模型，接着就是一次请求的全过程，紧接着就是聊传输层的tcp的连接，然后就是传输层的tcp协议之上的socket编程，接下来还会聊聊应用层的http协议。

所以说，来吧，这都是最最基础的网络知识。

3、面试题剖析

其实说白了，socket就是在传输层里把tcp/ip协议给封装了一下，我们程序员一般都是面向socket来编程的，比如java原生就支持socket网络编程的。

大体来说这个步骤，就是我们搞一个ServerSocket无限等待别人来连接你，然后某个机器要跟你连接，就在本地创建一个socket去连接你，然后建立连接之后，在服务器上，ServerSocket也会创建出来一个socket的。通过客户端的socket跟服务端的socket进行通信，我给你写数据，你读数据，你给我写数据，我读数据，就这个过程。

当然这个底层，比如建立连接和释放连接，都是基于tcp三次握手和四次挥手的规范来搞的，包括基于tcp协议传输数据，其实就跟我们之前说的一样，都是封装个tcp数据包，里面有tcp报头，整了端口号啥的，然后封装在ip数据包里，最后封在以太网数据包里传递。

就这么回事，大家理解了就好了。

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/014\_一道经典的面试题：HTTP实现长连接的底层工作原理到底是啥？/笔记.docx**

1、面试题

什么是长连接？http长连接是什么？

2、面试官心里分析

一期学员，在外面面试的时候，正好还碰到了，聊到dubbo，dubbo://协议，是走的长连接，你聊聊什么是长连接？什么是http长连接？

3、面试题剖析

http本身没什么所谓的长连接短连接之说，其实说白了都是http下层的tcp连接是长连接还是短连接，tcp连接保持长连接，那么多个http请求和响应都可以通过一个链接来走。其实http 1.1之后，默认都是走长连接了，就是底层都是一个网页一个tcp连接，一个网页的所有图片、css、js的资源加载，都走底层一个tcp连接，来多次http请求即可。

http 1.0的时候，底层的tcp是短连接，一个网页发起的请求，每个请求都是先tcp三次握手，然后发送请求，获取响应，然后tcp四次挥手断开连接；每个请求，都会先连接再断开。短连接，建立连接之后，发送个请求，直接连接就给断开了

http 1.1，tcp长连接，tcp三次握手，建立了连接，无论有多少次请求都是走一个tcp连接的，走了n多次请求之后，然后tcp连接被释放掉了

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/009\_一道经典的面试题：如果TCP建立连接就两次握手有什么问题吗？/笔记.docx**

1、面试题

TCP三次握手和四次握手的工作流程是什么（画一下流程图）？为什么不是五次握手或者两次握手？

2、面试官心里分析

这个问题相当经典，大家可别以为就是考察应届生的，实际上在普通社招java面试中，一些大公司，很喜欢考察这个问题，尤其是后面第二个追加问题，让你聊聊为啥必须是三次握手，而不是两次呢？

3、面试题剖析

（1）tcp三次握手过程

通过传输层的tcp协议建立网络连接的时候，其实走的是三次握手的过程

建立三次握手的时候，TCP报头用到了下面几个东西，ACK、SYN、FIN。

第一次握手，客户端发送连接请求报文，此时SYN=1、ACK=0，这就是说这是个连接请求，seq = x，接着客户端处于SYN\_SENT状态，等待服务器响应。

第二次握手，服务端收到SYN=1的请求报文，需要返回一个确认报文，ack = x + 1，SYN=1，ACK = 1，seq = y，发送给客户端，自己处于SYN\_RECV状态。

第三次握手，客户端收到了报文，将ack = y + 1，ACK = 1，seq = x + 1

其实三次握手说白了，就是来回来去三次请求，每次请求带上一堆TCP报文头，根据报文头是否正确，就是越好的协议来建立连接。简单说就是这样。

（2）为啥不是2次或者4次握手呢？

假设两次握手就ok了，要是客户端第一次握手过去，结果卡在某个地方了，没到服务端；完了客户端再次重试发送了第一次握手过去，服务端收到了，ok了，大家来回来去，三次握手建立了连接。

结果，尴尬的是，后来那个卡在哪儿的老的第一次握手发到了服务器，服务器直接就返回一个第二次握手，这个时候服务器开辟了资源准备客户端发送数据啥的，结果呢？客户端根本就不会理睬这个发回去的二次握手，因为之前都通信过了。

但是如果是三次握手，那个二次握手发回去，客户端发现根本不对，就会发送个复位的报文过去，让服务器撤销开辟的资源，别等着了。

因为3次握手就够了，不需要4次或者5次浪费资源了。

（3）tcp断开连接的4次挥手

第一次挥手，客户端发送报文，FIN=1，seq=u，此时进入FIN-WAIT-1状态

第二次挥手，服务端收到报文，这时候进入CLOSE\_WATI状态，返回一个报文，ACK=1，ack=u+1，seq=v。客户端收到这个报文之后，直接进入FIN-WAIT-2状态，此时客户端到服务端的连接就释放了。

第三次挥手，服务端发送连接释放报文，FIN=1，ack=u+1，seq=w，服务端进入LAST-ACK状态

第四次挥手，客户端收到连接释放报文之后，发应答报文，ACK=1，ack=w+1，seq=u+1，进入TIME\_WAIT状态，等待一会儿客户端进入CLOSED状态，服务端收到报文之后就进入CLOSED状态。

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/013\_一道经典的面试题：HTTPS协议加密通信的实现原理是什么/笔记.docx**

1、面试题

http和https的区别是什么？https的原理是什么？

2、面试官心里分析

聊到http了，那肯定会聊聊https

3、面试题剖析

http协议都是明文的，是没有加密的，所以其实现在一般大部分应用都是用https协议的。之前是基于SSL协议对http进行加密，后来又升级到了TSL协议来加密，现在称之为SSL / TSL吧。

https的工作原理大概是这样的：

（1）浏览器把自己支持的加密规则发送给网站

（2）网站从这套加密规则里选出来一套加密算法和hash算法，然后把自己的身份信息用证书的方式发回给浏览器，证书里有网站地址、加密公钥、证书颁发机构

（3）浏览器验证证书的合法性，然后浏览器地址栏上会出现一把小锁；浏览器接着生成一串随机数密码，然后用证书里的公钥进行加密，这块走的非对称加密；用约定好的hash算法生成握手消息的hash值，然后用密码对消息进行加密，然后把所有东西都发给网站，这块走的是对称加密

（4）网站，从消息里面可以取出来公钥加密后的随机密码，用本地的私钥对消息解密取出来密码，然后用密码解密浏览器发来的握手消息，计算消息的hash值，并验证与浏览器发送过来的hash值是否一致，最后用密码加密一段握手消息，发给浏览器

浏览器解密握手消息，然后计算消息的hash值，如果跟网站发来的hash一样，握手就结束，之后所有的数据都会由之前浏览器生成的随机密码，然后用对称加密来进行进行加密。

常用的非对车呢加密是RSA算法，对称加密是AES、RC4等，hash算法就是MD5

就好比，有个人说我加密的时候是用了一个公钥去加密，然后你解密的时候是用私钥去解密；我加密的时候用的算法，跟解密的时候用的算法，是一样的，对称加密

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/006\_应用层：HTTP协议又是什么以及七层网络模型的总结/笔记.docx**

1、面试题

说一下TCP/IP的四层网络模型？那么七层网络模型呢？

2、面试官心里分析

结构化面试：不是瞎面，cs基础（computer science），java基础，j2ee，mysql，linux，tomcat，分布式，高并发，高可用，系统设计，数据结构和算法，软素质，工程素质

坦白讲，一些大的公司，计算机基础必面，尤其是针对薪资30k以内的工程师，因为薪资30k以内，你还是要干活儿的吧，还没上升到就设计架构就可以的程度吧，你还没到那个高度吧。所以只要你干活儿，你就不可避免要跟机器、网络、cpu、磁盘、内存，成天打交道。而线上系统，计算基础的一些东西，网络、cpu、磁盘、内存，都是关联很大的，比如说你线上系统会不会因为网络故障导致一些问题？cpu负载达到100%了咋办？磁盘读写很慢快满了咋办？内存使用率过高咋办？

你起码得有一套自己的计算机功底去支撑你玩儿线上系统吧。所以很多人呢，都说计算机基础没啥用，那这个话呢，也对，也不对。对就在于，你如果毕业出来干简单的crud，这些东西你确实不需要；不对就在于，你如果当个高工，带几个小弟干高并发有压力的线上系统，机器负载很高，很容易出问题，结果你连机器都不敢摸，或者也不知道怎么摸，那不是尴尬了么。

所以说，计算机基础，网络、磁盘、cpu、内存，还是得会一点儿基础的

作为一个大公司的面试官，一定会考察你这些东西

3、面试题剖析

（1）简单介绍一下

四层模型和七层模型可以一块儿聊

为啥需要协议？没有协议，各个电脑厂商，比如IBM、苹果啥的，都弄自己的协议，结果就苹果电脑和苹果电脑自己可以通信，和IBM电脑就不可以通信，这不是尴尬么。所以搞一个国际通行的协议，大家都按照这个来，所有电脑都可以通信，不是很好么。

OSI七层模型，就是搞一个标准的网络模型出来，大家都按照这个来走，那么大家都有统一的规范。OSI七层模型，是应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。TCP/IP四层模型，数据链路层、网络层、传输层、应用层。

（2）从底向上的网络分层

1）物理层

物理层，物理层干啥的，就是电脑之间要联网，一般咋弄？类似于说，你有台电脑，现在要联网，咋联？以前N年前，大家记不记得都是在电脑上插根线是吧，然后才能上网，结果现在就是联个wifi就行了，还有中国美国之前联网靠的是海底的光缆。所以物理层就指的这个，就是怎么把各个电脑给联结起来，形成一个网络，这就是物理层的含义，物理层负责传输0和1的电路信号。学过一些计算机的同学，计算机的最最底层，就是0/1，电信号。

2）数据链路层

数据链路层，物理层给各个电脑连接起来了，还传输最底层的0和1电路信号，关键不行啊，你得定义清楚哪些0和1分为一组，这些信号啥意思？这才能进行通信。所以数据链路层就干这事儿，定义一下电路信号咋分组。

00000011（从电脑1出发，要到电脑2去）

00101（从电脑1出发，要到电脑3去）

0101（从电脑2触发，要到电脑4去）

01（从电脑3出发，要到电脑5去）

很多年前，每个公司都定义自己的电路信号分组方式，但是后来出来了以太网协议，以太网。一组电信号是一个数据包，叫一个帧（frame），每个帧分成两个部分，标头（head）和数据（data），标头包含一些说明性的东西，比如说发送者、接收者和数据类型之类的。

每台电脑要往另外一台电脑发送数据，一堆0/1电路信号，封装成数据包，包含头和数据，头里包含了从哪儿来到哪儿去，必须从一台电脑的一个网卡，发送到另外一个电脑的一个网卡，所以以太网发送的数据包必须得指定，目标电脑的网卡的mac地址。

以太网规定了，每个网卡必须得包含一个mac地址，mac地址就是这个网卡的唯一标识，

以太网协议规定了，接入网络里的所有设备，都得有个网卡，以太网协议里的那个数据包，在数据链路层传输的数据包，必须从一个电脑的网卡传输到另外一个电脑的网卡，而这个网卡地址就叫做所谓的mac地址。每块网卡出厂的时候，就有一个唯一的mac地址，48位的二进制，但是一般用12个16进制数字表示，前6个16进制是厂商编号，后6个16进制是网卡流水号。

windows上，ipconfig /all，看看物理地址，就是mac地址，7C-67-A2-20-AB-5C

所以在以太网里传输数据包的时候，必须指定接收者的mac地址才能传输数据。

但是以太网的数据包怎么从一个mac地址发送到另一个mac地址？这个不是精准推送的，以太网里面，如果一个电脑发个数据包出去，会广播给局域网内的所有电脑设备的网卡，然后每台电脑都从数据包里获取接收者的mac地址，跟自己的mac地址对比一下，如果一样，就说明这是发给自己的数据包。

但是上面这种广播的方式，仅仅针对一个子网（局域网）内的电脑，会广播，否则一个电脑不能广播数据包给全世界所有的其他电脑吧，是仅仅广播给一个子网里面的电脑的。

3）网络层

上面说到，子网内的电脑，通过以太网发个数据包，对局域网内的电脑，是广播出去的。那么怎么知道哪些电脑在一个子网内呢？这就得靠网络层了，这里就有一套IP地址，IP地址就可以让我们区分哪些电脑是一个子网的。

网络层里有IP协议，IP协议定义的地址就叫做IP地址。IP地址有IPv4和IPv6两个版本，目前广泛使用的是IPv4，是32个二进制数字组成的，但是一般用4个十进制数字表示，范围从0.0.0.0到255.255.255.255之间。

每台计算机，都会分配一个ip地址，ip地址的前24位（就是前面3个十进制数字），代表了网络，后8位（就是最后1个十进制数字），代表了主机。如果几台电脑是一个子网的，那么前面的3个十进制数字一定是一样的。举个例子，大家平时做实验，玩儿虚拟机吧，自己win上开几个linux虚拟机，你会发现，win上的ip地址可能是192.168.0.103，然后几个虚拟机的ip地址是192.168.0.182，192.168.0.125，192.168.0.106，类似这样的。这个win机器和几个虚拟机，前面3个十进制数字都是192.168.0，就代表大家是一个子网内的，最后那个数字是这个子网的不同主机的编号。

但是实际上上面就是举个例子，其实单单从ip地址是看不出来哪些机器是一个子网的，因为从10进制是判断不出来的。需要通过ip地址的二进制来判断，结合一个概念来判断，叫做子网掩码。比如说ip地址是192.168.56.1，子网掩码是255.255.255.0。知道了子网掩码之后，如果要判断两个ip地址是不是一个子网的，就分别把两个ip地址和自己的子网掩码进行二进制的与运算，与运算之后，比较一下代表网络的那部分。

192.168.56.1和192.168.32.7，判断是不是一个子网的，拿子网掩码255.255.255.0，跟两个ip地址的二进制做与运算

11000000.10101000.00111000.00000001

11111111.11111111.11111111.00000000

子网掩码的二进制是：11111111.11111111.11111111.00000000，然后就跟ip地址的二进制做与好了，通过二进制来比较网络部分的地址是不是一模一样的。

有了网络层的ip地址之后，两台在子网内的电脑终于可以通过广播+mac地址判断来传输数据包进行通信了。

但是如果发现要接受数据包的计算机不在子网内，那么就不能通过广播来发送数据包，需要通过路由来发送数据包。看到路由，就想到了路由器了，对了，路由器大家都熟悉吧，自己平时也会去买对吧，比如小米的路由器啥的，家里上网一般都会弄个路由器对吧，ok。路由器负责将多个子网进行连接，因为比如你在自己家里，其实你就只是你自己的一个子网，你要是访问网站啥的，是跟那个网站机器所在的子网进行通信。

每个电脑都可以搞多个网卡的，不是只有一个网卡，一般笔记本电脑都有以太网网卡和wifi网卡，发送数据包的时候要决定走哪个网卡。路由器，其实就是配置了多个网卡的一个专用设备，可以通过不同的网卡接入不同的网络。

网关其实是就是路由器的一种，运作在网络层，这个概念不多解释了，大家可以就把路由器上的ip地址认为是网关，路由器上每个网卡都有mac地址和对应的ip地址。路由器虽然有mac地址，但是不能通过mac地址寻址的，必须通过ip地址寻址，所以路由器其实是工作在网络层的设备。

网络交换机，也是一种设备，是工作在数据链路层的，路由器是工作在网路层的。网络交换机是通过mac地址来寻址和传输数据包的；但是路由器是通过ip地址寻址和传输数据包的。网络交换机主要用在局域网的通信，一般你架设一个局域网，里面的电脑通信是通过数据链路层发送数据包，通过mac地址来广播的，广播的时候就是通过网络交换机这个设备来把数据广播到局域网内的其他机器上去的；路由器一般用来让你连入英特网。

LAN，就是local area network，就是局域网；WAN，就是wide area network，就是广域网。WLAN是wireless local area network，就是无线局域网，也就是wifi，在局域网内，直接通过wifi无线联网。

家里的路由器是包含了交换机和路由的两个功能的，如果是连接到局域网内的设备就把线插LAN那儿；如果是连接到英特网，就把线插在WAN那儿。

这儿给大家举个例子，就是两个局域网之间，如果要是通过一个路由器进行通信的话，是怎么弄的。

大概过程就是，路由器配置了两块网卡，每个网卡可以连到一个局域网内。

局域网1内的电脑，要发送数据包到局域网2内的电脑，在数据包里写上自己的ip地址和对方的ip地址。但是他们俩不在一个局域网内，于是局域网1内的电脑，先通过交换机将数据包发送给路由器，这个过程需要将路由器的一块网卡的ip地址对应的mac地址写到数据包的头部，然后才能通过交换机广播出去，路由器接收到之后比较自己一块网卡的mac地址，就知道是来找自己的。

接着路由器接收到数据包之后，就会在局域网2内，将目标机器的ip地址对应的mac地址写入头部，接着再次通过交换机发送广播通知，发送给局域网2内的电脑。

一个局域网内的每台机器都有自己的ARP cache，这个ARP就是用来在一个局域网内让各个设备都知道每个设备的ip地址和mac地址的对应关系的，一般就是某个机器发送广播通知自己的ip地址和mac地址的对应关系，然后每个机器给他一个回应。以此类推，大家都互相这样广播一把，ip地址和mac地址的对应关系，大家不就都知道了。

一个ip地址对应着一个mac地址

所以大家在上面可以看到，一个子网内的机器之间通信，就是在数据包里写上对方的mac地址，然后交换机广播出去ok了；但是如果是跨子网的通信，就是写上对方的ip地址，然后先通过mac地址广播到路由器，让路由器再根据另外一个子网的ip地址转换为mac地址，通过另外一个子网的交换机广播过去。就这个意思。

4）传输层

上面我们大概明白了通过网络层的ip地址怎么划分出来一个一个的子网，然后在子网内部怎么通过mac地址广播通信；跨子网的时候，怎么通过ip地址 -> mac地址 -> 交换机 -> 路由器 -> ip地址 -> mac地址 -> 交换机的方式来通过路由器进行通信。

但是这里还有一个问题，就是一台机器上，是很多个程序用一个网卡进行网络通信的，比如说浏览器、QQ、视频直播，这些软件都用了一个网卡往外面发送数据，然后从网卡接收数据，对吧。

所以还需要一个端口号的概念，就是你得发送数据包到某个机器的一个网卡的某个端口上去，然后那个机器上监听那个端口的程序，就可以提取发送到这个端口的数据，知道是自己的数据。端口号是0~65536的范围内，0~1023被系统占用了，别的应用程序就用1024以上的端口就ok了。

电脑1，是在端口48362监听的，通过网卡发送了一条数据 -> 电脑2的ip地址的20386这个端口 -> 电脑2的上面的某个QQ，监听着20386的端口 -> 电脑2的网卡接收到一条数据之后，发现人家找的是20386这个端口，就去找谁哪个哥儿们在监听20386端口，QQ在监听，我就把这个网卡过来的数据，传递给QQ，通过端口知道，哪条数据是给你的

所以其实大家会发现一点，网络层，是基于ip协议，进行主机和主机间的寻址和通信的，然后传输层，其实是建立某个主机的某个端口，到另外一个主机的某个端口的连接和通信的。这个通信，就是通过socket来实现的，通过socket就可以基于tcp/ip协议完成刚才上面说的一系列的比如基于ip地址和mac地址转换和寻址啊，通过路由器通信啊之类的，而且会建立一个端口到另外一个端口的连接。

udp和tcp都是传输层的协议，作用就是在数据包里加入端口号，可以通过端口号进行点对点的通信了。udp协议是不可靠的，发出去人家收到没有就不知道了；tcp协议是可靠的，要求三次握手，而且要求人家接收到数据必须回复你。

传输层的tcp协议，仅仅只是规定了一套基于端口的点对点的通信协议，包括如何建立连接，如何发送和读取消息，但是实际上如果你要基于tcp协议来开发，你一般是用socket，java socket网络编程

5）应用层

通过传输层的tcp协议可以传输数据，但是人家收到数据之后，怎么来解释？比如说收到个邮件你怎么处理？收到个网页你怎么处理？类似这个意思，所以针对各种不同的应用，邮件、网页之类的，都是定义不同的应用层协议的。这个应用层，我们就假设综合了会话层、表示层和应用层了，3层合成1层。

就是对数据交换的格式的规范和限定，你们里

电脑1走tcp协议发送了一段东西过来，发送到电脑2的20386端口

GET http://localhost:8080/ http/1.1

key:valuel

key:value

电脑2走tcp协议读取到了属于自己这个20386端口 的一段数据

GET http://localhost:8080/ http/1.1

key:valuel

key:value

发送了一段响应

200

key;value

key:value

又通过底层的tcp发了出去，电脑1的30987端口，ip

电脑1，网卡，走以太网协议收到一个数据包

200

key;value

key:value

浏览器自己就会通过操作系统，然后跟对方的服务器进行TCP三次握手，建立TCP连接，接下来就可以发送这个一大段的http数据，就会通过Socket数据流的方式来发送，就会自动被拆分为多个TCP包

服务器返回http响应，也是一段数据，可能拆分为多个TCP包，HTTP协议，规定的数据网络传输的数据的格式

比如最常见的，应用层的协议就是http协议，进行网络通信。

4层：数据链路层（以太网协议），网络层（ip协议），传输层（tcp协议），应用层（http协议）

7层：物理层（网线，海底光缆，传递0/1电路信号），会话层、表示层、应用层 -> 应用层

然后我们看下自己的网络设置，一般包含了ip地址、子网掩码、网关地址、DNS地址。前面3个我们其实都知道啥意思了。ip地址和子网掩码用来划分子网的，判断哪些ip地址在一个子网内。同时你的ip地址和mac地址关联起来的，唯一定位了你的网卡。网关地址，你就认为是路由器上的那个网卡的ip地址吧，路由器的网卡也有mac地址，mac地址对应了一个ip地址。

DNS地址是啥呢？Domain Name System。因为我们一般定位是通过ip地址+mac地址+端口号来定位一个通信目标的，但是如果在浏览器上输入一个www.baidu.com，咋整？这个时候是先把www.baidu.com发给DNS服务器，然后DNS服务器告诉你www.baidu.com对应的ip地址的。

8.8.8.8

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/004\_网络层：IP协议、子网划分以及子网掩码是什么（下）/笔记.docx**

1、面试题

说一下TCP/IP的四层网络模型？那么七层网络模型呢？

2、面试官心里分析

结构化面试：不是瞎面，cs基础（computer science），java基础，j2ee，mysql，linux，tomcat，分布式，高并发，高可用，系统设计，数据结构和算法，软素质，工程素质

坦白讲，一些大的公司，计算机基础必面，尤其是针对薪资30k以内的工程师，因为薪资30k以内，你还是要干活儿的吧，还没上升到就设计架构就可以的程度吧，你还没到那个高度吧。所以只要你干活儿，你就不可避免要跟机器、网络、cpu、磁盘、内存，成天打交道。而线上系统，计算基础的一些东西，网络、cpu、磁盘、内存，都是关联很大的，比如说你线上系统会不会因为网络故障导致一些问题？cpu负载达到100%了咋办？磁盘读写很慢快满了咋办？内存使用率过高咋办？

你起码得有一套自己的计算机功底去支撑你玩儿线上系统吧。所以很多人呢，都说计算机基础没啥用，那这个话呢，也对，也不对。对就在于，你如果毕业出来干简单的crud，这些东西你确实不需要；不对就在于，你如果当个高工，带几个小弟干高并发有压力的线上系统，机器负载很高，很容易出问题，结果你连机器都不敢摸，或者也不知道怎么摸，那不是尴尬了么。

所以说，计算机基础，网络、磁盘、cpu、内存，还是得会一点儿基础的

作为一个大公司的面试官，一定会考察你这些东西

3、面试题剖析

（1）简单介绍一下

四层模型和七层模型可以一块儿聊

为啥需要协议？没有协议，各个电脑厂商，比如IBM、苹果啥的，都弄自己的协议，结果就苹果电脑和苹果电脑自己可以通信，和IBM电脑就不可以通信，这不是尴尬么。所以搞一个国际通行的协议，大家都按照这个来，所有电脑都可以通信，不是很好么。

OSI七层模型，就是搞一个标准的网络模型出来，大家都按照这个来走，那么大家都有统一的规范。OSI七层模型，是应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。TCP/IP四层模型，数据链路层、网络层、传输层、应用层。

（2）从底向上的网络分层

1）物理层

物理层，物理层干啥的，就是电脑之间要联网，一般咋弄？类似于说，你有台电脑，现在要联网，咋联？以前N年前，大家记不记得都是在电脑上插根线是吧，然后才能上网，结果现在就是联个wifi就行了，还有中国美国之前联网靠的是海底的光缆。所以物理层就指的这个，就是怎么把各个电脑给联结起来，形成一个网络，这就是物理层的含义，物理层负责传输0和1的电路信号。学过一些计算机的同学，计算机的最最底层，就是0/1，电信号。

2）数据链路层

数据链路层，物理层给各个电脑连接起来了，还传输最底层的0和1电路信号，关键不行啊，你得定义清楚哪些0和1分为一组，这些信号啥意思？这才能进行通信。所以数据链路层就干这事儿，定义一下电路信号咋分组。

00000011（从电脑1出发，要到电脑2去）

00101（从电脑1出发，要到电脑3去）

0101（从电脑2触发，要到电脑4去）

01（从电脑3出发，要到电脑5去）

很多年前，每个公司都定义自己的电路信号分组方式，但是后来出来了以太网协议，以太网。一组电信号是一个数据包，叫一个帧（frame），每个帧分成两个部分，标头（head）和数据（data），标头包含一些说明性的东西，比如说发送者、接收者和数据类型之类的。

每台电脑要往另外一台电脑发送数据，一堆0/1电路信号，封装成数据包，包含头和数据，头里包含了从哪儿来到哪儿去，必须从一台电脑的一个网卡，发送到另外一个电脑的一个网卡，所以以太网发送的数据包必须得指定，目标电脑的网卡的mac地址。

以太网规定了，每个网卡必须得包含一个mac地址，mac地址就是这个网卡的唯一标识，

以太网协议规定了，接入网络里的所有设备，都得有个网卡，以太网协议里的那个数据包，在数据链路层传输的数据包，必须从一个电脑的网卡传输到另外一个电脑的网卡，而这个网卡地址就叫做所谓的mac地址。每块网卡出厂的时候，就有一个唯一的mac地址，48位的二进制，但是一般用12个16进制数字表示，前6个16进制是厂商编号，后6个16进制是网卡流水号。

windows上，ipconfig /all，看看物理地址，就是mac地址，7C-67-A2-20-AB-5C

所以在以太网里传输数据包的时候，必须指定接收者的mac地址才能传输数据。

但是以太网的数据包怎么从一个mac地址发送到另一个mac地址？这个不是精准推送的，以太网里面，如果一个电脑发个数据包出去，会广播给局域网内的所有电脑设备的网卡，然后每台电脑都从数据包里获取接收者的mac地址，跟自己的mac地址对比一下，如果一样，就说明这是发给自己的数据包。

但是上面这种广播的方式，仅仅针对一个子网（局域网）内的电脑，会广播，否则一个电脑不能广播数据包给全世界所有的其他电脑吧，是仅仅广播给一个子网里面的电脑的。

3）网络层

上面说到，子网内的电脑，通过以太网发个数据包，对局域网内的电脑，是广播出去的。那么怎么知道哪些电脑在一个子网内呢？这就得靠网络层了，这里就有一套IP地址，IP地址就可以让我们区分哪些电脑是一个子网的。

网络层里有IP协议，IP协议定义的地址就叫做IP地址。IP地址有IPv4和IPv6两个版本，目前广泛使用的是IPv4，是32个二进制数字组成的，但是一般用4个十进制数字表示，范围从0.0.0.0到255.255.255.255之间。

每台计算机，都会分配一个ip地址，ip地址的前24位（就是前面3个十进制数字），代表了网络，后8位（就是最后1个十进制数字），代表了主机。如果几台电脑是一个子网的，那么前面的3个十进制数字一定是一样的。举个例子，大家平时做实验，玩儿虚拟机吧，自己win上开几个linux虚拟机，你会发现，win上的ip地址可能是192.168.0.103，然后几个虚拟机的ip地址是192.168.0.182，192.168.0.125，192.168.0.106，类似这样的。这个win机器和几个虚拟机，前面3个十进制数字都是192.168.0，就代表大家是一个子网内的，最后那个数字是这个子网的不同主机的编号。

但是实际上上面就是举个例子，其实单单从ip地址是看不出来哪些机器是一个子网的，因为从10进制是判断不出来的。需要通过ip地址的二进制来判断，结合一个概念来判断，叫做子网掩码。比如说ip地址是192.168.56.1，子网掩码是255.255.255.0。知道了子网掩码之后，如果要判断两个ip地址是不是一个子网的，就分别把两个ip地址和自己的子网掩码进行二进制的与运算，与运算之后，比较一下代表网络的那部分。

192.168.56.1和192.168.32.7，判断是不是一个子网的，拿子网掩码255.255.255.0，跟两个ip地址的二进制做与运算

11000000.10101000.00111000.00000001

11111111.11111111.11111111.00000000

子网掩码的二进制是：11111111.11111111.11111111.00000000，然后就跟ip地址的二进制做与好了，通过二进制来比较网络部分的地址是不是一模一样的。

有了网络层的ip地址之后，两台在子网内的电脑终于可以通过广播+mac地址判断来传输数据包进行通信了。

但是如果发现要接受数据包的计算机不在子网内，那么就不能通过广播来发送数据包，需要通过路由来发送数据包。看到路由，就想到了路由器了，对了，路由器大家都熟悉吧，自己平时也会去买对吧，比如小米的路由器啥的，家里上网一般都会弄个路由器对吧，ok。路由器负责将多个子网进行连接，因为比如你在自己家里，其实你就只是你自己的一个子网，你要是访问网站啥的，是跟那个网站机器所在的子网进行通信。

每个电脑都可以搞多个网卡的，不是只有一个网卡，一般笔记本电脑都有以太网网卡和wifi网卡，发送数据包的时候要决定走哪个网卡。路由器，其实就是配置了多个网卡的一个专用设备，可以通过不同的网卡接入不同的网络。

网关其实是就是路由器的一种，运作在网络层，这个概念不多解释了，大家可以就把路由器上的ip地址认为是网关，路由器上每个网卡都有mac地址和对应的ip地址。路由器虽然有mac地址，但是不能通过mac地址寻址的，必须通过ip地址寻址，所以路由器其实是工作在网络层的设备。

网络交换机，也是一种设备，是工作在数据链路层的，路由器是工作在网路层的。网络交换机是通过mac地址来寻址和传输数据包的；但是路由器是通过ip地址寻址和传输数据包的。网络交换机主要用在局域网的通信，一般你架设一个局域网，里面的电脑通信是通过数据链路层发送数据包，通过mac地址来广播的，广播的时候就是通过网络交换机这个设备来把数据广播到局域网内的其他机器上去的；路由器一般用来让你连入英特网。

LAN，就是local area network，就是局域网；WAN，就是wide area network，就是广域网。WLAN是wireless local area network，就是无线局域网，也就是wifi，在局域网内，直接通过wifi无线联网。

家里的路由器是包含了交换机和路由的两个功能的，如果是连接到局域网内的设备就把线插LAN那儿；如果是连接到英特网，就把线插在WAN那儿。

这儿给大家举个例子，就是两个局域网之间，如果要是通过一个路由器进行通信的话，是怎么弄的。

大概过程就是，路由器配置了两块网卡，每个网卡可以连到一个局域网内。

局域网1内的电脑，要发送数据包到局域网2内的电脑，在数据包里写上自己的ip地址和对方的ip地址。但是他们俩不在一个局域网内，于是局域网1内的电脑，先通过交换机将数据包发送给路由器，这个过程需要将路由器的一块网卡的ip地址对应的mac地址写到数据包的头部，然后才能通过交换机广播出去，路由器接收到之后比较自己一块网卡的mac地址，就知道是来找自己的。

接着路由器接收到数据包之后，就会在局域网2内，将目标机器的ip地址对应的mac地址写入头部，接着再次通过交换机发送广播通知，发送给局域网2内的电脑。

一个局域网内的每台机器都有自己的ARP cache，这个ARP就是用来在一个局域网内让各个设备都知道每个设备的ip地址和mac地址的对应关系的，一般就是某个机器发送广播通知自己的ip地址和mac地址的对应关系，然后每个机器给他一个回应。以此类推，大家都互相这样广播一把，ip地址和mac地址的对应关系，大家不就都知道了。

一个ip地址对应着一个mac地址

所以大家在上面可以看到，一个子网内的机器之间通信，就是在数据包里写上对方的mac地址，然后交换机广播出去ok了；但是如果是跨子网的通信，就是写上对方的ip地址，然后先通过mac地址广播到路由器，让路由器再根据另外一个子网的ip地址转换为mac地址，通过另外一个子网的交换机广播过去。就这个意思。

4）传输层

上面我们大概明白了通过网络层的ip地址怎么划分出来一个一个的子网，然后在子网内部怎么通过mac地址广播通信；跨子网的时候，怎么通过ip地址 -> mac地址 -> 交换机 -> 路由器 -> ip地址 -> mac地址 -> 交换机的方式来通过路由器进行通信。

但是这里还有一个问题，就是一台机器上，是很多个程序用一个网卡进行网络通信的，比如说浏览器、QQ、视频直播，这些软件都用了一个网卡往外面发送数据，然后从网卡接收数据，对吧。

所以还需要一个端口号的概念，就是你得发送数据包到某个机器的一个网卡的某个端口上去，然后那个机器上监听那个端口的程序，就可以提取发送到这个端口的数据，知道是自己的数据。端口号是0~65536的范围内，0~1023被系统占用了，别的应用程序就用1024以上的端口就ok了。

电脑1，是在端口48362监听的，通过网卡发送了一条数据 -> 电脑2的ip地址的20386这个端口 -> 电脑2的上面的某个QQ，监听着20386的端口 -> 电脑2的网卡接收到一条数据之后，发现人家找的是20386这个端口，就去找谁哪个哥儿们在监听20386端口，QQ在监听，我就把这个网卡过来的数据，传递给QQ，通过端口知道，哪条数据是给你的

所以其实大家会发现一点，网络层，是基于ip协议，进行主机和主机间的寻址和通信的，然后传输层，其实是建立某个主机的某个端口，到另外一个主机的某个端口的连接和通信的。这个通信，就是通过socket来实现的，通过socket就可以基于tcp/ip协议完成刚才上面说的一系列的比如基于ip地址和mac地址转换和寻址啊，通过路由器通信啊之类的，而且会建立一个端口到另外一个端口的连接。

udp和tcp都是传输层的协议，作用就是在数据包里加入端口号，可以通过端口号进行点对点的通信了。udp协议是不可靠的，发出去人家收到没有就不知道了；tcp协议是可靠的，要求三次握手，而且要求人家接收到数据必须回复你。

传输层的tcp协议，仅仅只是规定了一套基于端口的点对点的通信协议，包括如何建立连接，如何发送和读取消息，但是实际上如果你要基于tcp协议来开发，你一般是用socket，java socket网络编程

5）应用层

通过传输层的tcp协议可以传输数据，但是人家收到数据之后，怎么来解释？比如说收到个邮件你怎么处理？收到个网页你怎么处理？类似这个意思，所以针对各种不同的应用，邮件、网页之类的，都是定义不同的应用层协议的。这个应用层，我们就假设综合了会话层、表示层和应用层了，3层合成1层。

电脑1走tcp协议发送了一段东西过来，发送到电脑2的20386端口

GET http://localhost:8080/ http/1.1

key:valuel

key:value

电脑2走tcp协议读取到了属于自己这个20386端口 的一段数据

GET http://localhost:8080/ http/1.1

key:valuel

key:value

发送了一段响应

200

key;value

key:value

又通过底层的tcp发了出去，电脑1的30987端口，ip

电脑1，网卡，走以太网协议收到一个数据包

200

key;value

key:value

比如最常见的，应用层的协议就是http协议，进行网络通信。

4层：数据链路层（以太网协议），网络层（ip协议），传输层（tcp协议），应用层（http协议）

7层：物理层（网线，海底光缆，传递0/1电路信号），会话层、表示层、应用层 -> 应用层

然后我们看下自己的网络设置，一般包含了ip地址、子网掩码、网关地址、DNS地址。前面3个我们其实都知道啥意思了。ip地址和子网掩码用来划分子网的，判断哪些ip地址在一个子网内。同时你的ip地址和mac地址关联起来的，唯一定位了你的网卡。网关地址，你就认为是路由器上的那个网卡的ip地址吧，路由器的网卡也有mac地址，mac地址对应了一个ip地址。

DNS地址是啥呢？Domain Name System。因为我们一般定位是通过ip地址+mac地址+端口号来定位一个通信目标的，但是如果在浏览器上输入一个www.baidu.com，咋整？这个时候是先把www.baidu.com发给DNS服务器，然后DNS服务器告诉你www.baidu.com对应的ip地址的。

8.8.8.8

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/012\_一道经典的面试题：HTTP 1.0与HTTP 1.1版本间的区别是什么/笔记.docx**

1、面试题

说一下http的工作流程？http 1.0、http 1.1、http 2.0具体有哪些区别？

2、面试官心里分析

这个就是让你聊聊http，说白了http工作原理，你都知道了，发起个http，底层都是tcp、ip、以太网那块再走，一层一层包裹数据包。所以http的关键就是让你聊聊http请求和http响应的规范。

3、面试题剖析

http发起请求的底层原理，大家其实都知道了，理解了那个原理，就一通百通了。那么来聊下http请求和响应的规范吧。其实请求的报文，就是请求头、请求方法、请求正文，GET/POST啥的，应该都知道；请求头，自己百度一下吧，作为一个工程师必须知道。响应，状态行，响应头，响应正文，状态行，200,400,500，实在不想讲了；响应头，自己查一下。

http请求封装到应用层数据包，封装在tcp数据包，封装在ip数据包，封装在以太网数据包，如果过大，可能会拆成几个包，走以太网协议+交换机 -> 广播 -> 网关 -> 多个网关 -> 目标的机器 -> 一层一层拆包 -> http请求报文 -> 传递给tomcat -> spring mvc -> http响应 -> 一样的路径会去

最最底层，这个数据如何传输？走的是物理层，网线、光缆，所有数据都是0/1电路信号

http协议，其实是每个搞java必须会的基础。

互联网初期，一般一个网页几乎都没什么图片，当时就是挂一些文字，一个网页里就是一大坨的文字。http 1.0版本。

浏览器 -> 网站，互相之间是先要通过tcp三次握手，建立一个连接，浏览器和网站互相都给对方留出一份资源，浏览器发起http请求 -> tcp -> ip -> 以太网，网站上面去，网站返回一个响应，连接关闭，tcp四次挥手。释放掉浏览器和网站各自给对方保持的一份资源。

http 1.0要指定keep-alive来开启持久连接，默认是短连接，就是浏览器每次请求都要重新建立一次tcp连接，完事儿了就释放tcp连接。早期的网页都很low，没啥东西，就一点文字，就用这个没问题。但是现在，一个网页打开之后，还要加载大量的图片、css、js，这就坑爹了，发送多次请求。

早期，2000年之前，那个时候网页，都很low，当时你打开一个网页，就是说现场底层tcp三次握手，跟网站建立一个tcp连接，然后通过这个tcp连接，发送一次http请求，网站返回一个http响应（网页的html，里面有一大段文字），浏览器收到html渲染成网页，浏览器就走tcp四次挥手，跟网站断开连接了

到了后面，发现说2000之后，2010之后更不用说了，网页发展很迅猛，一个网页包含着大量的css、js、图片等资源。比如你请求一个网页，这个网页的html先过来，过来之后，浏览器再次发起大量的请求去加载css、js、图片，打开一个网页可能浏览器要对网站服务器发送几十次请求。

http 1.0，疯了，刚开始请求网页的html，tcp三次握手建立连接 -> 请求/响应 -> tcp四次挥手断开连接，接着再次要加载css、js、图片，要发送30个请求，上面的过程来30次，30次频繁的建立tcp连接以及释放tcp连接。很慢很慢。

其实最慢的不是说发送请求和获取响应，打开和释放连接，这都是很重的过程

http 1.1默认支持长连接，就是说，浏览器打开一个网页之后，底层的tcp连接就保持着，不会立马断开，之后加载css、js之类的请求，都会基于这个tcp连接来走。http 1.1还支持host头，也就可以支持虚拟主机；而且对断点续传有支持。

浏览器，第一次请求去一个网站的一个页面的时候，就会打开一个tcp连接，接着就在一段时间内都不关闭了，然后接下来这个网页加载css、js、图片大量的请求全部走同一个tcp连接，频繁的发送请求获取响应，最后过了一段时间，这些事儿都完了，然后才会去释放那一个tcp连接。大幅度的提升复杂网页的打开的速度，性能。

http 2.0，支持多路复用，基于一个tcp连接并行发送多个请求以及接收响应，解决了http 1.1对同一时间同一个域名的请求有限制的问题。二进制分帧，将传输数据拆分为更小的帧（数据包），frame（数据包，帧），提高了性能，实现低延迟高吞吐。

# **File Path: /Users/xiaotingting/Downloads/学习资料/09\_讲给Java工程师听的大白话网络课程unzip/资料/010\_一道经典的面试题：我再追问你一下TCP断开连接四次挥手的过程/笔记.docx**

1、面试题

TCP三次握手和四次握手的工作流程是什么（画一下流程图）？为什么不是五次握手或者两次握手？

2、面试官心里分析

这个问题相当经典，大家可别以为就是考察应届生的，实际上在普通社招java面试中，一些大公司，很喜欢考察这个问题，尤其是后面第二个追加问题，让你聊聊为啥必须是三次握手，而不是两次呢？

3、面试题剖析

（1）tcp三次握手过程

通过传输层的tcp协议建立网络连接的时候，其实走的是三次握手的过程

建立三次握手的时候，TCP报头用到了下面几个东西，ACK、SYN、FIN。

第一次握手，客户端发送连接请求报文，此时SYN=1、ACK=0，这就是说这是个连接请求，seq = x，接着客户端处于SYN\_SENT状态，等待服务器响应。

第二次握手，服务端收到SYN=1的请求报文，需要返回一个确认报文，ack = x + 1，SYN=1，ACK = 1，seq = y，发送给客户端，自己处于SYN\_RECV状态。

第三次握手，客户端收到了报文，将ack = y + 1，ACK = 1，seq = x + 1

其实三次握手说白了，就是来回来去三次请求，每次请求带上一堆TCP报文头，根据报文头是否正确，就是越好的协议来建立连接。简单说就是这样。

（2）为啥不是2次或者4次握手呢？

假设两次握手就ok了，要是客户端第一次握手过去，结果卡在某个地方了，没到服务端；完了客户端再次重试发送了第一次握手过去，服务端收到了，ok了，大家来回来去，三次握手建立了连接。

结果，尴尬的是，后来那个卡在哪儿的老的第一次握手发到了服务器，服务器直接就返回一个第二次握手，这个时候服务器开辟了资源准备客户端发送数据啥的，结果呢？客户端根本就不会理睬这个发回去的二次握手，因为之前都通信过了。

但是如果是三次握手，那个二次握手发回去，客户端发现根本不对，就会发送个复位的报文过去，让服务器撤销开辟的资源，别等着了。

因为3次握手就够了，不需要4次或者5次浪费资源了。

（3）tcp断开连接的4次挥手

第一次挥手，客户端发送报文，FIN=1，seq=u，此时进入FIN-WAIT-1状态

第二次挥手，服务端收到报文，这时候进入CLOSE\_WATI状态，返回一个报文，ACK=1，ack=u+1，seq=v。客户端收到这个报文之后，直接进入FIN-WAIT-2状态，此时客户端到服务端的连接就释放了。

第三次挥手，服务端发送连接释放报文，FIN=1，ack=u+1，seq=w，服务端进入LAST-ACK状态

第四次挥手，客户端收到连接释放报文之后，发应答报文，ACK=1，ack=w+1，seq=u+1，进入TIME\_WAIT状态，等待一会儿客户端进入CLOSED状态，服务端收到报文之后就进入CLOSED状态。