

# 单元一讨论题讲稿

## 〇、序言

我们组的题目如下：

假设我和爸妈在聊天，那么从我在宿舍里的电脑开始，到家里妈妈的电脑，再到上班路上爸爸的手机，中间涉及到了哪些网络、设备、媒介与技术？

在这份 PPT 里，我会先在总体上描述一遍，信息按时序经过了哪些设备与媒介；然后再挑若干个，我们小组在研讨过程中，发现的比较有趣的技术，给大家做一个分享。

## 一、总体流程

首先，数据从宿舍的电脑开始发送，经过 TCP/IP 模型的层层封装，变成了比特流。现在大家宿舍里基本都是用的 WiFi 无线上网，很少会在笔记本电脑上连根网线，所以这堆信号会以无线电波的形式，传输到宿舍里的路由器。

但路由器的背后就不再是无线的了，它背后是连着光纤的；这些光纤会先在——比如一层或一栋楼有一个——光纤分线盒汇聚，然后通过主干线连向电信的基站机房。真正的远程传输，就从基站开始，一个一个基站地传输。

然后这些数据还不会直接到达你妈那边，它会先去比如腾讯的服务器；腾讯的服务器会进行相关的查找和验证，比如查找你妈的地址，检查你和你妈是不是好友、数据库备份消息等等。这里还有个技术叫长连接（Keepalive），我们待会细说。验证成功后，这些数据再同样通过基站、光纤、WiFi，到达家里妈妈的电脑上。

妈妈给爸爸发消息的话就不太一样了，爸爸在上班路上肯定没有 WiFi，所以数据同样从基站出来之后，是通过蜂窝网络直接到达离他最近的基站，然后直接传到爸爸的手机

上。

整个流程图大概就是这样。然后我们再挑几个重点和大家讲一讲，首先是关于 WiFi 的一些知识点。

---

## 二、WiFi

### 2.1 工作频段

主要频段是 **2.4G/5GHz**。

2.4G 频段非常拥挤，微波炉等家用电器、蓝牙、各种物联网设备、无线 USB 等等，都在这个频段工作。所以日常生活中，2.4G 的 WiFi 干扰非常严重。但是 2.4G 的覆盖范围比 5G 要大，这也就是在家里，可以搜到左右邻居的无线信号的原因。能搜到的信号基本都是 2.4G 信号。

5G 的覆盖范围相对 2.4G 小很多。现在家里或者寝室里一般都用的是 5G 的 WiFi，范围再小也够用了，信号更强，而且使用这个频段的设备也不多。

像我们宿舍就有两个 WiFi，一个 2.4G、一个 5G，5G 的那个就明显要比 2.4G 的那个更好用。

在 2.4G 或者 5G 的大频段下，还分有不同的信道。2.4G 总共 14 个信道，一般用的比较多的是 1、6、11 信道，因为这三个信道是互相不重叠的；5G 的信道就很多了，我们国家准许使用的有 36, 38, 40, 42, 44, 46, 48, 149, 153, 157, 161 信道。

至于你自己家里的 WiFi 使用的是哪个信道，这个是可以自己调的。比如我们寝室的信道是 149，对应的通信频率就是 5.745GHz。所以如果你发现 WiFi 比较慢，有可能是你的信道和隔壁的信道重合了，你可以尝试把信道调走，以恢复上网速度。

### 2.2 SSID 与 AP

之前说到，5G 的覆盖范围，相比起 2.4G 来说是要小一些的；但是我们学校的网络，采用的频段也是 5G。那我们就不难想到，学校部署 WiFi 的思路，或者说框架，肯定是我们一般宿舍里不一样的。

首先明确几个关于 WiFi 的概念。

SSID (Service Set Identifier)，通俗来说就是用户能看到的 WiFi 的名字。输入了正确的密码，通过了身份验证，那你就算是连上了这个 WiFi。

但在现实意义上，你的手机连接的不是 SSID，而是 AP（Access Point），俗称热点，也就是内网的一个连接点；你的设备实际上是先连上这个 AP，然后 AP 再去接入有线网络。

有了这些概念，我们就不难猜出宿舍和学校分别采用哪种部署方式——宿舍 WiFi 的部署方式是 BSS（Basic Service Set）= 1 SSID + 1 AP + n Devices；学校的部署方式是 ESS（Extended Service Set）= 1 SSID + n AP + n Devices。

也就是说，你在品学楼连的是一个 AP，在立人楼可能连的就是另一个 AP，但它们有同一个名字 SSID，叫 UESTC-WiFi，所以你不会明显地感觉到自己在几个热点之间切换。

## 2.3 通信协议

最后来说一说 WiFi 采用的通信协议。

一开始采用的是 IEEE 802.3 标准，在这个标准下，设备容易受干扰，而且各个厂商做出来的产品有可能不兼容。

后来的十几年里，WiFi 协议经历了快速的迭代，从 WiFi 1 一直到现在的 WiFi 6，也就是 802.11ax 标准，现在的协议已经采用了非常先进的技术，里面有我们熟悉的 OFDMA（4G 就基于这个技术），也就是 OFDM+FDMA（各个用户简单的瓜分带宽，1G 就基于这个技术），MIMO，1024QAM 等等。

---

## 三、Keepalive

想把这个事情拎出来讲，是因为之前写了一个自动化程序，然后因为这个 keepalive 出了 bug，当时找这个 bug 找了我好几天。研究之后感觉这个事情挺有意思，所以想拿出来讲一讲。TCP 的 keepalive 和 HTTP 的 keep-alive 是两码事，我们会分别讲述。

### 2.1 TCP 的 keepalive

这个是我看 bilibili 直播的时候，拿 Fiddler 抓的一个包。可以看到，直播间会不时给我发个包，叫做 heartbeat（心跳），它返回了一个 JSON，大致意思是表示我确实还

在看这场直播。如果直播间连续几次给我发这个包，我这边都没有回答（ACK），那直播间就知道我已经不在看这个直播了，然后它就会释放这个连接。

——这个就是 TCP 的 keepalive。它用于保持客户端和服务端的连接。也因此这个包叫做 heartbeat，心跳不跳了就是客户端挂了。设想一下，如果 TCP 层没有 keepalive 的机制，那么一旦一方意外断开连接，却没有发送 FIN 给另一方的话，那么另一方就会一直以为这个连接还是活着的，这个连接就会一直存在下去；长期累积下来，会无谓地、而且大幅地占用服务器资源。

## 2.2 HTTP 的 keep-alive

普通的 http 连接是客户端连接上服务端，然后结束请求后，由客户端或者服务端进行 http 连接的关闭。下次再发送请求的时候，还是重复这个流程，建立连接-传送数据-关闭连接。但其实像我们 QQ 聊天，可能几秒钟就交换一条信息，那这种重复的建立和关闭连接是完全没有必要的，完全可以让这个连接就这么存在下去，然后我们每次就只要传送数据就可以了。

——这个就是 http 的 keep-alive。客户端请求的时候加一句 { connection: keep-alive }，服务端要是也接受这个 keep-alive 的话，这个连接就可以复用了，减少新建和断开 TCP 连接的消耗。另外，一开始的 HTTP 1.0 协议默认是不开 keep-alive 的，是从 HTTP 1.1 开始才默认支持 keep-alive。

## 2.3 二者的区别

打个比方，我来电子科技大学上学，周一到周五都一直在上课，那这个过程就可以看作我和学校在交换数据；周六周日不上课了，我要是回家那就是断开连接，要是住宿舍那就是 keep-alive（HTTP）。显然住校肯定划算多了，像我回家要买机票，费时费钱，还不如保持这个连接。

但 HTTP 的 keep-alive 只能确保我一直在学校，但不能确保我会来上课。所以有些课的老师就会不定期地点下名，确认你确实来上课了，你确实在交换数据。这个就是 TCP 的 keepalive。

HTTP 做的事情是 keep connection alive，TCP 做的事情是 keep you alive。