**WIFI基本知识整理**

这里对wifi的802.11协议中比较常见的知识做一个基本的总结和整理，便于后续的学习。因为无线网络中涉及术语很多，并且许多协议都是用英文描述，所以有些地方翻译出来会有歧义，这种情况就直接英文来描述了。

**主要内容：**

**一、基本概述**

**二、实践基础**

**三、一些原理**

**四、补充**

**五、其它**

**一、基本概述**

**============================**

1、有线和无线网络

目前有线网络中最著名的是以太网(Ethenet)，但是无线网络WLAN是一个很有前景的发展领域，虽然可能不会完全取代以太网，但是它正拥有越来越多的用户，无线网络中最有前景的是Wifi。本文介绍无线网络相关内容。

无线网络相比有线网络，还是有许多的缺点的：

（\*）通信双方因为是通过无线进行通信，所以通信之前需要建立连接；而有线网络就直接用线缆连接，不用这个过程了。

（\*）通信双方通信方式是半双工的通信方式；而有线网络可以是全双工。

（\*）通信时在网络层以下出错的概率非常高，所以帧的重传概率很大，需要在网络层之下的协议添加重传的机制（不能只依赖上面TCP/IP的延时等待重传等开销来保证）；而有线网络出错概率非常小，无需在网络层有如此复杂的机制。

（\*）数据是在无线环境下进行的，所以抓包非常容易，存在安全隐患。

（\*）因为收发无线信号，所以功耗较大，对电池来说是一个考验。

（\*）相对有线网络吞吐量低，这一点正在逐步改善，802.11n协议可以达到600Mbps的吞吐量。

2、协议

Ethenet和Wifi采用的协议都属于IEEE 802协议集。其中，Ethenet以802.3协议做为其网络层以下的协议；而Wifi以802.11做为其网络层以下的协议。无论是有线网络，还是无线网络，其网络层以上的部分，基本一样。

这里主要关注的是Wifi网络中相关的内容。Wifi的802.11协议包含许多子部分。其中按照时间顺序发展，主要有：

（1）802.11a，1999年9月制定，工作在5gHZ的频率范围（频段宽度325MHZ），最大传输速率54mbps，但当时不是很流行，所以使用的不多。

（2）802.11b，1999年9月制定，时间比802.11a稍晚，工作在2.4g的频率范围（频段宽度83.5MHZ），最大传输速率11mbps。

（3）802.11g，2003年6月制定，工作在2.4gHZ频率范围（频段宽度83.5MHZ），最大传输速率54mbps。

（4）802.11n，2009年才被IEEE批准，在2.4gHZ和5gHZ均可工作，最大的传输速率为600mbps。

这些协议均为无线网络的通信所需的基本协议，最新发展的，一般要比最初的有所改善。

另外值得注意的是，802.11n在MAC层上进行了一些重要的改进，所以导致网络性能有了很大的提升例如：

（\*）因为传输速率在很大的程度上取决于Channel（信道）的ChannelWidth有多宽，而802.11n中采用了一种技术，可以在传输数据的时候将两个信道合并为一个，再进行传输，极大地提高了传输速率（这又称HT-40，high through）。

（\*）802.11n的MIMO（多输入输出）特性，使得两对天线可以在同时同Channel上传输数据，而两者却能够不相互干扰（采用了OFDM特殊的调制技术）

3、术语

讲述之前，我们需要对无线网络中一些常用的术语有所了解。这里先列出一些，后面描述中出现的新的术语，将会在描述中解释。

（\*）LAN：即局域网，是路由和主机组成的内部局域网，一般为有线网络。

（\*）WAN：即广域网，是外部一个更大的局域网。

（\*）WLAN（Wireless LAN，即无线局域网）：前面我们说过LAN是局域网，其实大多数指的是有线网络中的局域网，无线网络中的局域网，一般用WLAN。

（\*）AP（Access point的简称，即访问点，接入点）：是一个无线网络中的特殊节点，通过这个节点，无线网络中的其它类型节点可以和无线网络外部以及内部进行通信。这里，AP和无线路由都在一台设备上（即Cisco E3000）。

（\*）Station（工作站）：表示连接到无线网络中的设备，这些设备通过AP，可以和内部其它设备或者无线网络外部通信。

（\*）Assosiate：连接。如果一个Station想要加入到无线网络中，需要和这个无线网络中的AP关联（即Assosiate）。

（\*）SSID：用来标识一个无线网络，后面会详细介绍，我们这里只需了解，每个无线网络都有它自己的SSID。

（\*）BSSID：用来标识一个BSS，其格式和MAC地址一样，是48位的地址格式。一般来说，它就是所处的无线接入点的MAC地址。某种程度来说，它的作用和SSID类似，但是SSID是网络的名字，是给人看的，BSSID是给机器看的，BSSID类似MAC地址。

（\*）BSS（Basic Service Set）：由一组相互通信的工作站组成，是802.11无线网络的基本组件。主要有两种类型的IBSS和基础结构型网络。IBSS又叫ADHOC，组网是临时的，通信方式为Station<->Station，这里不关注这种组网方式；我们关注的基础结构形网络，其通信方式是Station<->AP<->Station，也就是所有无线网络中的设备要想通信，都得经过AP。在无线网络的基础形网络中，最重要的两类设备：AP和Station。

（\*）DS（Distributed System）：即分布式系统。分布式系统属于802.11逻辑组件，负责将帧转发至目的地址，802.11并未规定其技术细节，大多数商业产品以桥接引擎合分步式系统媒介共同构成分布式系统。分步式系统是接入点之间转发帧的骨干网络，一般是以太网。其实，骨干网络并不是分步系统的全部，而是其媒介。主要有三点：骨干网（例如以太网）、桥接器（具有有线无线两个网络接口的接入点包含它）、属于骨干网上的接入点所管辖的基础性网络的station通信（和外界或者BSS内部的station）必须经过DS、而外部路由只知道station的mac地址，所以也需要通过分布式系统才能知道station的具体位置并且正确送到。分步式系统中的接入点之间必须相互传递与之关联的工作站的信息，这样整个分步式系统才能知道哪个station和哪个ap关联，保证分步式系统正常工作（即转达给正确的station）。分步式系统也可以是使用无线媒介(WDS)，不一定一定是以太网。总之，分步式系统骨干网络（例如以太网）做为媒介，连接各个接入点，每个接入点与其内的station可构成BSS，各个接入点中的桥接控制器有到达骨干网络和其内部BSS无线网的接口（类似两个MAC地址），station通信需要通过分布式系统。

**二、实践基础**

**============================**

1、一些参数

（\*）MAC

MAC（即Medium/MediaAccess Control, 介质访问控制），是数据链路层的一部分。MAC地址是烧录在NetworkInterfaceCard(即网卡，简称NIC)里的，它也叫硬件地址,是由48位（即bit，一字节为8位，即1byte=8bits)16进制的数字组成。其中0-23位叫做组织唯一标志符（organizationally unique，简称OUI），是识别LAN（局域网）节点的标识（在有些抓包工具抓包的时候会将前三个字节映射成某种组织名称的字符，也可以选择不显示这种映射）。24-47位是由厂家自己分配。

（\*）SSID

表示一个子网的名字，无线路由通过这个名字可以为其它设备标识这个无线路由的子网。设备进行扫描的时候，就会将相应SSID扫描到，然后就能够选择相应的SSID连接到相应的无线网络（当然不扫描，理论上也可以直接指定自己事先已经知道的ssid进行连接）。SSID可以和其它的重复，这样扫描的时候会看到两个同样SSID的无线网络，其实这一般用于将一个无线网络扩大的情况（毕竟无线路由器无线信号的覆盖范围是有线的）：当想要扩大一个无线网络（即SSID固定）的范围的时候，可以给多个路由设置相同的SSID来达到这个目的。（这也是漫游的原理，漫游的时候，我们可以在远方或者本地都能够打电话，也就是访问移动通信网络）。

SSID和BSSID不一定一一对应，一个BSSID在不同的Channel上面可能会对应到多个SSID，但是它们在一个Channel是一一对应的；另外，漫游的时候，虽然SSID不变，但是BSSID一定是会变化的。我们经常可以看到实际数据包中的AP的MAC地址和BSSID只差几位，其实实际设备的MAC地址可能只有一个，和BSSID没什么对应关系。在一个包含了路由功能和AP功能的无线路由器（Fat AP）上面，很可能是：路由器有两个MAC地址，一个用于外网（WAN），一个用于内网(WLAN和LAN)，一般路由器上面或者配置路由器的网页上面只标注外网的MAC地址；内网的MAC地址和外网MAC地址一般只有几位不同（甚至连续，也有些相差很多的例外）。

（\*）Band（频率范围）

一般ap可以支持5g或2.4g两个频率范围段的无线信号。如果两者同时可以设置，而不是互斥那么，这个路由器还能够同时支持两种频段（频段即Band），这相当于这个ap可建立两个无线网络，它们采用不同的频段（这类似收音机在长波范围内收音和短波范围内收音）。

（\*）Channel（信道）

Channel是对频段的进一步划分（将5G或者2.4G的频段范围再划分为几个小的频段，每个频段称作一个Channel），有”5.18GHZ“，“Auto(DFS)”等等，处于不同传输信道上面的数据，如果信道覆盖范围没有重叠，那么不会相互干扰。对于信道的使用，在国际上有所规定。其中有些信道是无需授权即可直接使用的（究竟是那个频段的那个信道，依照各个国家而不同），无需授权使用的意思是，传输数据的时候（无论以哪种无线方式），可以让设备收发的功率导致传输时的数据进入该信道的频率并在该信道所在频段宽度内进行传输；授权的使用的意思是，不允许传输时使用授权信道进行，否则会违反规定，并且干扰该信道上其他数据的传输。另外，除了wifi，微波、红外线、蓝牙（使用802.15协议）的工作频段也都有在2.4gHZ范围内的，所以，它们传输的时候会对wifi传输造成干扰，因为两者在不同的协议下进行通信，所以互相将对方传输的信号识别为噪声。 有时候配置AP的时候，Channel中有一个类似“Auto”的选项值，这表示打开AP的时候，AP自己Scan周围的环境，选择一个干扰最小的Channel来进行通信，当选择好了一个Channel的时候，一般就不会改变了。

（\*）Channel Width（信道宽度）

这里的Channel Width是信道的带宽，有”20M HZ“、”40M HZ“等，它表示一个Channel片段的宽度（假设5g的频段宽度总共为100M，平均划分为互不干扰的10个Channel，那么每个Channel的Channel Width就为100M/10=10M，实际Channel并不一定是完全不重叠的）。这个参数可能依赖于一些其它的选项，例如不是802.11N的协议，就可能不会有40M HZ的Channel Width（N模式有一个特点就是可以把两个Channel合并，通过提高ChannelWidth来提高吞吐量）。例如选择了"20M HZ"这个Channel Width之后，后面再选择一个“5.18GHZ”的Channel，则表示以5.18GHZ为中心的前"10M HZ"以及其后面的"10M HZ"频带范围被占用。

至此可知，配置无线AP的时候，如果屋子里面有很多的AP(也就是无线路由接入点）的话，仔细设置它们的Channel Width和Channel可以保证它们相互之间的干扰（类似收音机里面的串台）尽可能小。当然，如果相互干扰了，那么Net Mode所指定的协议也会有相应的处理方式让他们之间进行协调（例如让谁先通信谁等一会再通信之类的），但是这样网络的性能就不如没有干扰的时候好了。

（\*）Wireless Security（无线网络的安全性）

这里主要涉及WEP、WPA、WPA2和RC4、TKIP、AES。

IEEE 802.11 所制定的是技术性标准 ,Wi-Fi 联盟所制定的是商业化标准 , 而 Wi-Fi 所制定的商业化标准基本上也都符合 IEEE 所制定的技术性标准。WEP 是1999年9月通过的 IEEE 802.11 标准的一部分；WPA(Wi-Fi Protected Access) 事实上就是由 Wi-Fi 联盟所制定的安全性标准 , 这个商业化标准存在的目的就是为了要支持 IEEE 802.11i 这个以技术为导向的安全性标准；而 WPA2 其实就是 WPA 的第二个版本。直观点说，WEP是较老的认证方法它有好几个弱点，因此在2003年被WPA淘汰，WPA又在2004年由完整的 IEEE 802.11i 标准（又称为 WPA2）所取代。

WEP（Wired Equivalent Privacy），采用名为RC4的RSA加密技术；WPA(Wi-Fi Protected Access) ，采用新的TKIP算法，TKIP算法保留了RC4所以也有其弱点，但是这个时候更好的CCMP还没完成，所以先在WPA上用TKIP技术；WPA2是WPA的第2个版本，采用CCMP加密协定（在有些路由器等设备上设定加密协定或者加密算法的时候，可能会用类似AES之类的字眼替代CCMP）。所以WPA2+AES是安全性最强的。

另外，在有些无线网路设备的参数中会看到像 WPA-Enterprise / WPA2-Enterprise 以及 WPA-Personal / WPA2-Personal 的字眼 , 其实 WPA-Enterprise / WPA2-Enterprise 就是 WPA / WPA2 ； WPA-Personal / WPA2-Personal 其实就是 WPA-PSK / WPA2-PSK, 也就是以 ”pre-share key” 或 ” passphrase” 的验证 (authentication) 模式来代替 IEEE 802.1X/EAP 的验证模式 ,PSK 模式下不须使用验证服务器 ( 例如 RADIUS Server), 所以特别适合家用或 SOHO 的使用者。

还有，wep是旧的加密方式，工作于802.11B/G模式下而802.11N草案并不支持此加密方式，所以如果802.11N的设备采用wep加密方式后，它也只会工作在802.11b/g模式下，N的性能发挥不出来。

实际中，在有些路由器上面，设置的时候，可能不是严格按照这个规定来设置的（例如设定了采用WPA方式，还可以选择AES），但是大体一样。

（\*）Region（区域）

一般在无线网络中的AP上都有一个参数，表明它是处于哪个Region（地区）。Station根据AP中设置的Region调整其相应的发射功率以遵守该地区的规定。AP的调整过程一般都是手动设定，设置好AP所处的Region之后，这些信息就会在AP发送的Beacon帧（后面会说到）中包含了；通过这个AP连接到无线网络上的Station，从Beacon帧中了解到这些Region信息，并且根据这些信息中的规定和AP进行通信。如果AP开始设置错了，那么Station和AP通信的时候，采用的将会是不符合Region规定的频段，可能会对该Region中的其它传输网络造成干扰，这应当是“非法”的。

（\*）Transmission Rate

设置传输速率。这里采用不同的无线网络传输协议（802.11a，802.11b，802.11g等），那么可以设置的速率范围有所不同，这里的速度是指理论的速度，实际中，由于各种干扰因素，传输的速率可能会比设置的小。

一般而言，在无线网络中，对于某种协议的性能进行描述时，我们需要注意的是，描述时提到的传输速率（Datarate）和吞吐量（ Throughput）是不同的。Datarate是理论上面最大数据传输速率，而Throughput是数据的实际最大吞吐量。因为厂家以及传输时所使用的协议等各种因素造成的开销，会导致实际吞吐量比理论吞吐量要小，一般实际最大吞吐为理论最大的50%左右（一个不太准确但是相对直观的估计：在网络中，高清视频所需的Throughput也就30mbps左右，网络上一般的视频也就4mbps左右）。

（\*）Qos（质量保证）

无线网络中的QOS是质量保证，大致的意思是，传输数据的时候，考虑各种因素（例如收费策略，所处地区等），以一定的优先级来保证传输的特定要求（一般就是速度），如果带宽足够的话，QOS反而不需要了。

（\*）RTS Threshold / CTS Protection Mode：

这里的RTS是Request-To-Send的简写，CTS是Clear-To-Send的简写。设置好RTS的阈值之后，如果超过这个阈值就会在发送信息之前先发送RTS，以减少干扰，相应的CTS会回应之前的RTS。一般都是AP发送CTS数据，而Station发送RTS数据。

这里对RTS和CTS做一个简单解释：假设在同一个AP所覆盖的无线网络范围内的两个Station A和B，它们之间可能会因为距离的原因互相不可见（例如它们在AP网络范围的两端，而这两端的距离大于两者的信号覆盖范围），但是AP却知道它们是在自己的范围内。当一个A想要在AP的网络中进行通信的时候，必定要经过AP转发它的信息，由于A不知道B的存在，所以如果同时B也通过AP进行网络通信，那么会出现AP同时收到A、B两个Station的通信请求，而这在无线网络中是不允许的（无线网络中，同一时刻不能有多个人传输数据）。在这种情况下，B和A互相干扰了对方的通信，但是却互相不可见（不可见的节点互相被称作隐藏节点）。如果在一个网络中，这样的隐藏节点很多，那么势必会影响网络的性能（因为数据一旦发送失败，就要重传，隐藏节点会导致重传的机率增大）。这个时候，可采用RTS和CTS机制。即：在A想要通信的时候，先广播发送RTS给AP，告诉AP“它想要通信”，同时接受到RTS的别的Station（它们对发送RTS的Station而言可见）会知道A将要发送数据，于是它们不会发送数据以免干扰A；AP收到RTS之后，会广播发送CTS，告诉所有在AP范围内的Station（包括对A而言的隐藏节点B）”A将要通信（同时也相当于告诉A，A可以无干扰的发送信息了）”，这样对A而言的隐藏节点B也知道有一个A的存在并且要发送信息了，于是B就不会干扰A了。 这里，A和B两者可以在不同的网络上，也就是说，不同网络的工作站之间也可以通过RTS/CTS来清除相互的干扰。

（\*）Beacon Interval：

表示无线路由定期广播其SSID的时间间隔。这个一般不会特别设置，就采用默认值即可。如果不广播了，那么Station端扫描的时候可能会发现不定期广播的AP对应的SSID的网络不见了，所以可能会断开连接。这里定期广播，表示AP会定时向其范围内广播SSID的信息，以表示AP的存在，这样Station进入一个区域之后，就能够通过扫描知道这个区域是否有AP的存在。当然，除了AP广播SSID以告知其无线网络存在之外，Station也可主动广播探寻包，在其能够覆盖的范围内询问是否有AP存在（即我们通常所说的扫描寻找接入点）。

（\*）DTIM Interval：

DTIM/TIM表示告诉Station，AP在为Station做package buffer（例如Station睡眠的时候）的缓存时间。为了节省电池使用时间，处于无线网络中的Station可能会在一定时间之后自动进入休眠状态。这个时候，AP会为这个Station缓存发送给它的数据，而处于休眠状态的Station只会在一定时间间隔内给AP发送一个数据帧，以确认是否有发送给自己的数据存在。例如，当我们在主机上ping另外一台睡眠的机器的时候，收到另外一台机器响应的时间，要比它不睡眠的时候响应的时间长很多。

（\*）Fragmentation Threshold：

表示一个package的分片阈值。我们可以设置分片大小，当发送的数据包超过这个阈值之后，802.11协议会自动对这个数据包进行分割。如果设置的这个分片值越小，那么整个数据包越容易传输成功（因为如果出错，那么只需要传送一个片段而不是整个包，无线wifi网络中数据传输时出错的概率比有线的以太网要大的多的多），当然开销也越大（因为需要额外的信息标记每个分片，以及各个分片传输成功之后涉及到的重组问题）。

2、抓包

一般来说，我们的机器上面的软件抓取无线网卡上面的包的时候，其实这些包的目标地址都是这个机器的无线网卡，因为不是发给这个机器无线网卡的包都被网卡过滤了。所以如果我们想要抓取所处无线网络环境下所有的包的时候，需要给机器配备一种特殊的设备（sniffer就是嗅探器），然后再通过抓包工具抓取并分析。有一个硬件设备叫做AirPcap，就是做这个用的，大有几百到上千美金，它可以同时做为嗅探器或者无线网卡使用，不过做为嗅探器的时候，会抓取所有经过它的包。这个工具目前只有Windows上面的驱动，所以使用这个工具，只能在Windows上面，配合Wireshark抓包软件进行抓包。

这里假设采用AirPcap嗅探，Wireshark软件抓包（其它抓包软件，例如linux下面的tcpdump等分析类似）。不用图形方式详细展示具体的抓包过程以及分析方法了，主要说一下抓包（这里的包实际主要指的是网络层以下的包，更常见的称呼应该是数据帧）时候需要注意的问题。

（\*）Wireshark展示包的时候，大致都是按照协议规定的字段展示，也些地方按照它自己特定的方式展示。因为这里着重讲述一些抓包时注意的基本原理上面的东西，所以不会对此进行过多阐述。大致就是：Wireshark软件中，对包展示的时候，按照协议规定的字段分别用Header和Body两个部分展示；另外，在Header之前还有两个部分是Wireshark为方便用户而展示的包的大小、时间等全局信息（例如见过表示这个包在B和G mode中的Channel 1时，用"BG1"表示）。所以，其实我们分析的时候，实际应该按照后面的Header和Body两个部分进行。 后面将基于以上所述，进行进一步的讲解。

（\*）抓包的时候，需要首先确认这个包是否是完整、正确的包。只要是校验位（checksum）不对的，就是错误的包，也无法确定接收的时候那里出了差错，所以这个包是应该忽略的，几乎没有分析的价值。另外，抓包的时候，由于干扰等原因，抓取的内容可能不是在实际传输所处的Channel上的包（例如在Channel 1上面嗅探，却嗅探到了Channel 2上的包）。

（\*）抓取授权阶段的包，需要注意实际的授权是在后面进行的。Authentication的时候，开始阶段实际是Open的（即无授权），也就是说，开始实际已经建立好了连接，所以我们在抓包的时候，开始看到的一般都是通过验证，但是在后面紧接着采用了类似802.11x等安全加强的协议，来进行再次鉴权认证，如果这里无法通过则立即将已经建立的Association断开。这样的机制，是因为原来的802.11没有充分考虑安全才会这样的，这样也兼容了以前的802.11。

（\*）抓取的包的数据，要注意这个包是否是被加过密的。根据协议标准的描述，包中如果有dataprotected字段，则表示这个数据本身是被加了密的，不知道这个数据具体是什么，当然，如果有密码，wireshark也有一个可以按照这个密码解密的工具，有时候不好用。这里所说的数据加密和网络的加密不一样，可能访问网络本身是需要密码（网络是security的），而数据本身没有crpted（加密）。对于一个加了密的数据包，我们一般看不出来这个包到底是做什么用的或者什么类型的等等。

（\*）抓包的时候，要注意包中指示的源和目的地址以及包的序号。在无线网络中通信的时候，我们抓包的时候可能会看到被抓取的包对应AP的MAC地址是不存在的，其实抓包时AP的MAC是BSSID，它和实际标注的MAC地址不一定一样（但是一般都差不多，也就是之后最后面的几位不一样）。有时候，我们看到抓取的包中的MAC地址有许多只相差几位，那么可能它们都属于一个设备（因为虽然设备可能只标注了一个网卡的MAC地址，但是它却“虚拟”出或者实际有多个MAC地址），所以当我们看到包中对应两个AP的MAC地址几乎一样的时候，一般来说，这两个MAC地址很可能就是一个设备的。还有在抓包的时候，一个地址上面的包的sequence（序号）是连续的，除非丢包了导致重复或者缺失。如果一个设备虚拟出来两个地址，那么也可能由于没有经过什么处理，导致这两个地址上面的包共同起来是连续的（如前所述，这两个地址和MAC很接近，应该是BSSID）。

（\*）抓取的数据帧如果是广播帧则不需要确认（ACK），如果是单播帧，则一般需要确认（ACK）。例如，Probe帧是广播帧，所以它无对应的ACK确认帧，对Probe的回复则叫做Probe Response；注意ACK帧本身用于确认，是单播的，但是它本身却不需要再被确认了。从包中的目的MAC地址中，可以看出这个包是广播／多播帧还是单播帧。MAC第一个字节的第一个位是1，表示组播，前两位是1表示广播，第一个字节第一个位是0表示单播。这里注意，MAC不是值，而是一个Pattern，所以没有Endian之说，也没有那个位高，那个MAC大之说。例如：“a8:27:26:....:b7”，这里第一个字节就是a8（10101000），其第一个字节的第一位就是8的最“右”位，即“0”，所以它的第一个字节的第一个位是0，是一个单播地址。其实，这里涉及到大端小端问题，后面也会讲到，总之，以太网线路上按“Big Endian”字节序传送报文（也就是最高字节先传送），而比特序是”Little Endian”（也就是字节内最低位先传送）所以，一个十六进制表示法表示的MAC地址01-80-C2-00-00-00，传送时的bit顺序就是：1000 0000 0000 0001 0100 0011 0000 0000 0000 0000 0000 0000。

（\*）使用Wire Shark在抓包或者显示包的时候，都可以设置过滤器（filter）。抓包时候设置的过滤器叫做capture filter，它是用BPF（berkerley package filter）这个比较通用的语言来描述（注意这不是Wireshark专用的filter语言，而是一个通用的语言）。但是抓包期间的过滤，有时候不准，所以我们一般先将所有的包抓取下来，然后用WireShark中显示的过滤器（即view filter）来显示我们关注的包，这里我们可以用macro来定义比较复杂的显示过滤条件。保存的时候，可以用按照显示过滤还是抓取过滤的方式保存内容。

（\*）尽量不要抓取Channel Width为40MHZ的Channel上的帧。我们还需要注意的是，使用Sniffer抓取无线网络包的时候，AirPcap无法正常抓取40MHZ Channel Width的包，或者说对抓取这个Channel Width上面的包支持不好。如果非要抓取40MHZ Channel Width的包，那么就在40或者36号Channel上面进行抓取，并在Wireshark上面设置“channel=36，offset+1”（平时offset都是0），这样能够抓取 Channel Width为40MHZ的包（但是，其他Channel上面的40mHZ的包还是无法抓取），这是由AirPcap内部的芯片固件的问题决定的（估计broad com芯片公司也不愿花过多的精力来支持这个很少有人用的抓包工具的这个功能）。

另外，假设一个无线工作站是基于Android系统的（例如智能手机或者平板电子书）那么我们可以利用“wpa\_cli status”命令来可以查看当前设备的连接的SSID，BSSID，MAC，IP等信息，（这里“cli”=“command line interface”)。 还有更“复杂”的命令“wc”和“wl”，其中wc是比较上层的命令，wl是下层的命令（是基于芯片是否支持的，例如wl在broadcom芯片上支持，但是在ti上面就没有了）。

**三、一些原理**

**============================**

1、常见的帧

在802.11中的帧有三种类型：管理帧（Management Frame，例如Beacon帧、Association帧）、控制帧（Control Frame，例如RTS帧、CTS帧、ACK帧）、数据帧（Data Frame，承载数据的载体，其中的DS字段用来标识方向很重要）。帧头部中的类型字段中会标识出该帧属于哪个字段。

（\*）ACK帧

单播(unicast)帧都需要用ACK来确认，ACK本身不是广播帧，ACK在MAC上是unicast的，帧中有receive地址字段（用来标识是对谁的确认），但是它却不需要再确认了。ACK只有接收地址（receive）而无源地址（src）和序号（sequence），因为发送和接受是一个整体，发送之后，其他人（除了这个发送的接受者）都不会再发送数据了（无线协议中的冲突避免机制），所以接受者会发送一个没有src的ack帧给receiver，而接收ACK的一端会根据这个知道它收到了一个ACK帧（其实根据协议，应当把发送单播帧和收到它相应的ACK看作一个原子的不可分割的整体，表示一次成功的通信）。

（\*）Beacon帧

Beacon帧定时广播发送，主要用来通知网络AP的存在性。Station和AP建立Association的时候，也需要用到Beacon。Station可以通过Scan来扫描到Beacon，从而得知AP的存在，也可以在扫描的时候通过主动发送Probe来探寻AP是否存在。也就是说，建立Association的时候有主动的扫描或者被动的扫描两种方式。另外，Beacon还包含了关于Power Save、以及地区等信息。

（\*）Association帧

通常Association帧都有Probe Request和相应的Probe Response。Association的Request中有其所需要的Channel以及Data Rate等状态，以便让AP决定是否让它与自己建立Association。而关联是否成功，主要是看Response中的Status code是否为Success。

（\*）Data帧

Data Frame具有方向，这个方向用DS（分布式系统）字段来标识，以区分不同类型帧中关于地址的解析方式；其它的类型Frame例如Control Frame或者管理帧中，这个字段是全零。这个字段用两位表示，这两个位的含义分别表示“To Ds”和“From Ds”，大致含义如下：

（a）To DS：表示Station->AP，一般也叫Upload。

（b）From DS表示AP->Station，一般也叫Download。

这里，我们可以大致将DS看做AP，To/From是从AP的角度来考虑的。To DS就是让AP干活。另外Data Frame中还有一个比较重要的字段就是Sequence，表示帧的序号。重传帧序号一样，但是多了一个Retry的字段表示该帧是重传的。

为了便于理解，这里再次详细解释一下DS字段的含义：

To DS=0,From DS=0:表示Station之间的AD Hoc类似的通信，或者控制侦、管理侦。

To DS=0,From DS=1:Station接收的侦。

To DS=1,From DS = 0:Station发送的侦。

To DS=1,From DS = 1:无线桥接器上的数据侦。

这里，我们主要关注To DS和From DS分别是01和10的情况，DS虽然大致等于AP但是它不是AP，它其实是一个系统，从Station的角度来看，比较容易理解。并且To DS和From DS一定是无线网络上面数据侦才有的字段。

2、帧和大端小端

Ethernet和802.11都是按照Little Endian的方式来传输数据，也就是说，而MAC层传输的时候，是采用Little Endian的方式，一个字节一个字节的传输的，前面的低位字节先传输，后面的高位字节后传输（传输单位不是按位而是字节）；在协议标准上描述一个帧的时候，一般是先按照Little Endian的方式对其进行总体描述，然后具体细节说每个字段的值，这时候这个字段值是Big Endian方式表示的，这一点应当注意。

例如，协议标准中可能能对某个帧格式做如下的描述：

|b0|b1|b2|b3|b4|b5|b6|b7|b8|b9|...|...|

这里，最低位b0在最前面，所以这里采用的就是小端的方式来描述帧的总体格式信息。传输的时候，就按照这里的方式，以字节为单位向物理层进行传输（先传b0~b7然后b8~b16等等）。 但是，在解释这个帧的各个域的时候却采用大端的方式进行描述。假设b3=0,b2=1,b1=0,b0=0四者共同组成一个名字为“FLAG”的域，那么会有类似如下的描述：

FLAG=4(即FLAG为0100)：表示XXX。

所以，协议标准中具体描述某个域的时候，一般直接用大端方式表示的数值(b3b2b1b0=0100)来描述；而传输数据帧或者在协议标准中描述整体帧的时候，中给出的却是小端的方式(b0b1b2b3=0010)。 这里的每个字段都是帧的一个部分，在管理帧（后面会说）中长度不固定的部分又叫IE(information Element) 。

另外注意，内存地址是用来标记每个字节的而不是位，所以内存里面大端小端也是以字节而不是位为单位的（前面描述“大端“、”小端”的时候却以位序而非字节序，这一点需要明辨，不要混淆）。假设奔腾的机器，CPU为32位，采用Little Endian方式，那么表示1这个int类型整数的时候，假设它在数值上是十六进制的"00000001"，那么存放在内存中却是由低位到高位依次存放的，由低到高地址依次为："01"、"00"、"00"、"00"（也就是说小端方式存放在内存中的时候，是按照含有最低位的字节存放在低地址，注意是字节，在内存中“位”没有地址，所以没有大端小端一说）。在传递帧的时候，也是按照一个字节一个字节的传输，而一个字节内部在实际上其实没有什么端的分别，但是wireshark一律使用“b7b6b5b4b3b2b1b0”这样的方式来用大端的方式显示。

总之，需要注意网络层下面的帧的大端小端问题（不是网络中的字节序，TCP/IP中规定的网络字节序是Big Endian），大致就是：协议规定，传输的时候使用Little Endian；标准描述的时候用Big Endian和Little Endian都用；另外，Wire shark软件抓的包中，好象全都用Big Endian来进行标示（无论是信息窗口还是内存窗口都这样展示）。

3、CSMA/CA的机制

与以太网的CSMA/CD机制（冲突检测）相对，802.11采用的CSMA/CA机制（冲突避免）。采用这个机制，可以保证每次通信的原子性（即每次通信所需要传输的多种不同类型的帧之间没有夹杂其它通信的帧的干扰），大体过程是：

（a）链路空闲下来之后，所有Station在发送帧之前都首先等待一段时间（即DIFS，又称帧间隔时间）；

（b）到达DIFS之后，所有的Station进入竞争时间窗口（就是竞争期间），将这个竞争时间窗口分割成多个Slot（退避时间间隔），然后每个Station随机选择一个Slot；

（c）当某个Station到达它的Slot对应的时间之后，就开始发送数据。这里，选择的Slot越靠前，则表示Station在DIFS之后再等待的时间（退避时间）越短，也就会越早发送实际数据；

（d）退避窗口的Slot有多个，选择的时候，可能某个Slot被多个站点同时选取，这个时候发送会产生真正的数据冲突（如果多个人同时发送，那么它们都要经过AP来转发，AP无法同时听见多个人的“说话声音”）那么Station就会再重新选择并发送；

（e）当一个Station发送数据之后，所有Station会检测到链路忙，于是放弃尝试发送，等那个Station发送完数据之后，链路开始空闲，于是又进入到（a）重新开始这个过程。

对于以上的机制，如果我们让某个Station经过DIFS之后，选择的Slot越小，就意味着它发送帧的机会越大，也就是说这个Station的优先权越高。这就是Qos（质量保证）的基本，前面也说过，Qos就是“以一定的优先级来保证传输的特定要求”，要获得这种优先级，就要有相应的条件（例如“花钱”）（有一种不常用的无竞争发送，其实就是DIFS之后，不退避而直接发送）。

另外，其实对物理层上来说，所有的发送都是广播，单播与否只是在链路层以上分辨的。上面提到的检测链路是否忙，可以从链路上用软件方式进行（例如增加帧的特殊字段），也可以直接在物理层上进行，实际因为在物理层上成本较高，经常用的是前者，具体参见协议。软件检测大致的思路就是，进行一个通信的时候，这个通信包含多个帧，每个帧有不同的作用，发送的第一帧的时候，会通过其中的某个特殊字段（Duration字段，也叫NAV，即网络分配向量，是一个延迟时间值）告诉所有其它Station，在未来的一段时间内，链路被占用，以完成整个通信过程。这样，其它Station在此期间就不会发送数据干扰这次通信了，以后这个通信的每一帧以及其ACK确认帧之间都会有一个很小的时间间隔（小于DIFS，即SIFS），并且每帧会视情况延长那个Duration字段，保证整个通信期间确实不会有其它人干扰，这样整个通信就是原子性的了。

4、帧的来源和目的地址

因为无线网络中没有采用有线电缆而是采用无线电波做为传输介质，所以需要将其网络层以下的帧格式封装的更复杂，才能像在有线网络那样传输数据。其中，仅从标识帧的来源和去向方面，无线网络中的帧就需要有四个地址，而不像以太网那样简单只有有两个地址（源和目的）。这四个地址分别是：

SRC：源地址（SA），和以太网中的一样，就是发帧的最初地址，在以太网和wifi中帧格式转换的时候，互相可以直接复制。

DST：目的地址（DA），和以太网中的一样，就是最终接受数据帧的地址，在以太网和wifi中帧格式转换的时候，互相可以直接复制。

TX：也就是Transmiter（TA），表示无线网络中目前实际发送帧者的地址（可能是最初发帧的人，也可能是转发时候的路由）。

RX：也就是Receiver（RA），表示无线网络中，目前实际接收帧者的地址（可能是最终的接收者，也可能是接收帧以便转发给接收者的ap）。

注意，其实，还有一个BSSID，用来区分不同网络的标识。在802.11帧中，有四个地址字段，一般只用到其中的三个，并且，这四个字段对应哪种地址或者使用哪些地址，根据帧中的另外一个DS字段以及帧的类型而有不同的解释。

举例：

（1）无线网络中的Station和以太网中的Host进行通信：

Station<- - - - ->AP<---------->Host

a）当Station->Host的时候：

首先Station->AP，这时候Src=Station,Dst=Host,Tx=Station,Rx=AP，然后AP->Host，这时候Src=Station,Dst=Host，因为AP转发的时候，是在以太网中，所以没有Tx和Rx。

b）当Host->Station的时候：

首先Host->AP，这时候Src=Host，Dst=Station，然后AP->Station，这时候，Src=Host,Dst=Station,Tx=AP,Rx=Station。

（2）无线网络中的Station之间进行通信：

Station1<- - - - ->AP<- - - - ->Station2

a）当Station1->Station2时

首先Station1->AP，Src=Station1,Dst=Station2,Tx=Station1,Rx=AP，然后AP->Station2，Src=Station1, Dst=Station2, Tx=AP, Rx=Station2。

可见，在无线网络中，始终存在Tx和Rx，但是，这四个地址中还是只有三个地址足矣。

（3）当两个无线网络中的Station进行通信的时候：

Station1<- - - - ->AP1<- - - - ->AP2<- - - - - ->Station2

当Station1->Station2时：

首先Station1->AP1，Src=Station,Dst=Station2,Tx=Station1,Rx=AP1，然后AP1->AP2，Src=Station, Dst=Station2, Tx=AP1, Rx=AP2，然后AP2->Station2，Src=Station1,Dst=Station2,Tx=AP2,Rx=Station2。

注意，这个时候，AP起到桥接的作用，所以四个地址各不相同，同时，AP之间或者Station和AP之间的那部分连接，也可以是以太网。

综上可知，无线网络中的Station想要通信，必须经过AP来进行“转发”，其实，Tx和Rx是无线网络中的发和收，也就是Radio；而Src和Dst是真正的发送源和接收者。

5、Sleep和Power save(节电)

其实，无线网络中的Power save是指Station的Sleep（睡眠），并且这个Sleep并不是整个系统的Sleep，确切来说，应该是其wifi中Receiver（接收天线）的Sleep。Station在睡眠的期间还是可以Transmit（发送）的，只是当AP知道Station的Receiver处于Sleep状态时，就不会给Station发送帧了。Station在Sleep之前，会给AP发送一个特殊的帧，告诉AP说它（Station）要睡眠了，AP通过这个帧来记住是这个Station睡眠了，然后AP就不会给这个Station单独发送数据了。

当有和这个Station通信的包想通过AP转达的给这个Station时候，AP会帮这个Station将它们缓存起来，然后在Beacon广播帧中添加一个特殊的位（实际这个位是一个bitmap中的位，这个bitmap表示所有和该AP建立了关联的Station，而这个睡眠的Station的相应位为被置1则表示有消息要传达给这个Station），来表示这个Station有数据到达了（Beacon是定时广播的帧，前面说过它是用来通知无线网络，这个AP的状态），而不是直接发送给Station。而这个睡眠的Station，会在睡眠期间不时地醒来，以检查Beacon帧中的状态，当发现有给它的数据的时候，就会通过发送一个Power Poll的帧来收取数据，收取之后继续睡眠（所以ping一个睡眠状态的Station，响应的时间要慢好多）。

对于发送给这个Station的广播帧，其处理方式和普通帧有一点不同：当有广播帧要传达给这个Station的时候，AP会为这个Station缓存发送给它的广播帧，但是缓存的时间是DTIM（一般为300ms）。注意：单播帧缓存的时间不一定是多少，广播帧却缓存DTIM的时间。AP每发送一个Beacon的时候，都会将Dtim减少1，而Station睡眠的时候，会不时地醒来，查看一下Beacon帧中的dtim值。当Station发现其DTIM值变成0的时候，就醒来长一些的时间，看看有没有广播给它的数据，如果有的话就用类似Power Save Poll的帧接受，没有则继续睡眠。

这里，接收数据是根据是否有more data类似的字段来确认是否有更多的数据的；重发的帧是用类似retry的字段来标记。另外注意，当Station进行Sleep的时候，还是可以主动Tranmit消息的，当Station主动Transmit消息的时候，它会等待Reply，所以这个时候，Receiver是on的状态。用一个图示来标识Sleep，Receive，Transmit时的电源消耗状况，大致如下：

power

^

trans | ------------------------

| | |

receive | -----------| |

| | |

sleep |--------| |--------------------

|----------------------------------------------------------------------> time

可见不同状态，电源消耗状态不同（传送比接收更耗电），另外，如果电源供电不足，在某个状态中就会出现通信失败的情况。（好像ap上面broadcom芯片中的睡眠之后，醒来立即重新发送的时候经常开始会失败，可能就是这个原因）。

6、建立Association

下面是Station和Ap建立开放Association的过程：

（0）Ap周期性地广播Beacon帧

（1）Station广播Probe Request到达Ap

（2）Ap向Station发送Probe Reponse

（3）Station向Ap发送ACK

（4）Station向Ap发送Authentication Request

（5）Ap向Station发送ACK

（6）Ap向Station发送Authentication Reponse

（7）Station向Ap发送ACK

（8）Station向Ap发送Association Request

（9）Ap向Station发送ACK

（10）Ap向Station发送Association Reponse

（11）Station向Ap发送ACK

（12）Station和Ap开始相互通信。

可见，广播帧不用回复，单播帧需要用ACK确认，ACK本身不用被确认。

**四、补充**

**============================**

有待添加。

**五、其它**

**============================**

本文内容主要来自学习的总结以及网络，主要集中于无线网络中物理层以上相对比较常见的部分，如果想要理解更详细和全面的内容则需参考相关书籍以及网络协议。由于对此方面的知识也是在初步学习之中，若文章中有错误和不完整之处，谢谢读者指正。^\_^

作者：QuietHeart

Email：quiet\_heart000@126.com

日期：2012年08月21日