cisco ARP

ARP协议深入解析

此文预备知识：OSI七层模型及数据封装和解封装的过程。

相信用过以太网的人都听说过ARP这个协议吧？（没有听过？&*%%$*^#）名词解释我就略过了，不知道ARP的中文名字的人自己查字典（好像字典里也没有？）。好了，废话讲完了，进入正题吧。

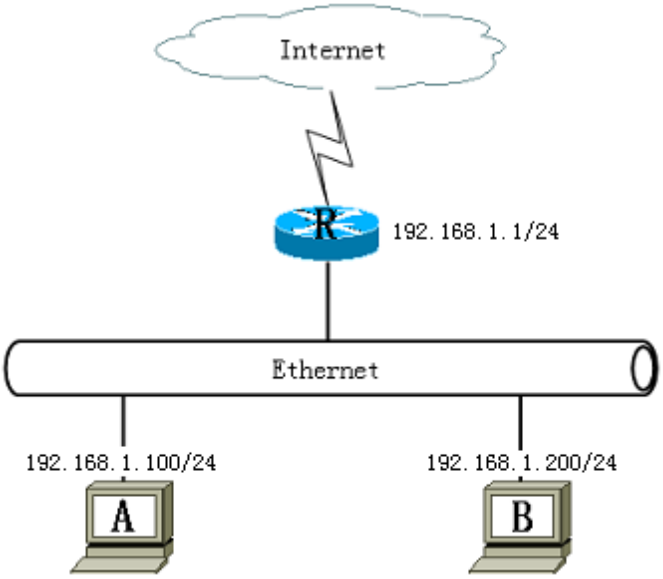
热身：

ARP协议是用在已知三层地址但不知二层地址（知道IP不知道MAC地址）的情况下用来查询目的IP地址所对应的MAC地址的一种辅助类协议。

什么时候需要用到ARP协议？

每台三层或三层以上的设备都会维护一张ARP表（WINDOWS用户在命令行下输入arp -a命令就能看到ARP表了），这个表记录了三层地址和二层地址的对应情况，以便实现以太网的点到点通信。如果要发起一个三层或以上的通信，那么源设备首先会查询自己的ARP表有没有符合的条目，如果没有，这时就要用到ARP协议来创建这个条目了。

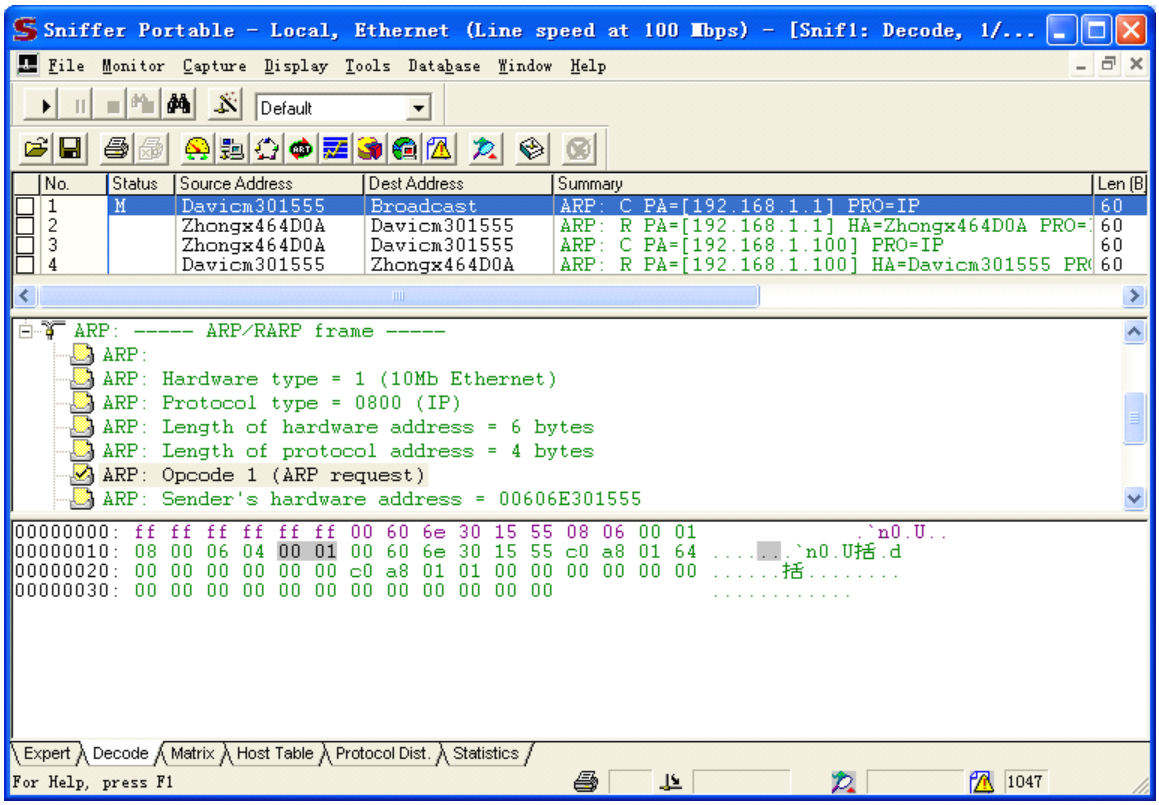
在介绍实例前先介绍一下我们实验用的拓扑：

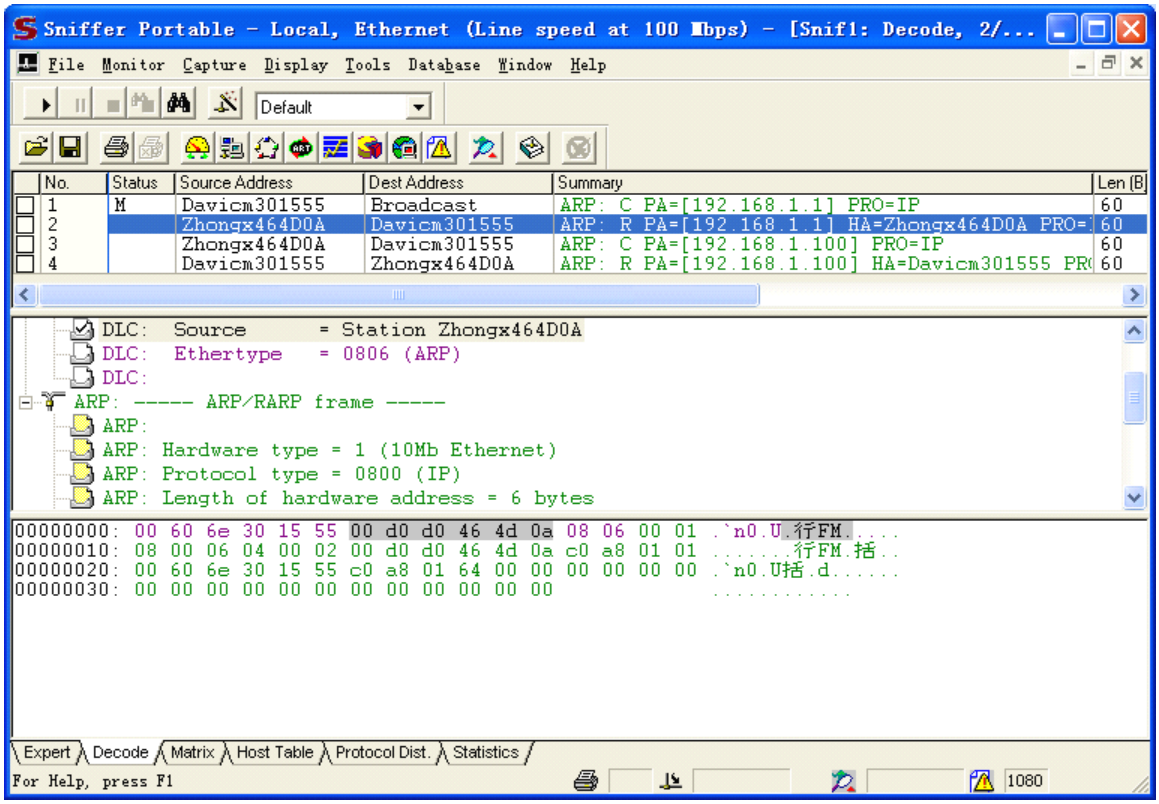


以后的实验都是基于这个拓扑。

我们用的是192.168.1.100这台，MAC地址是00-60-6e-30-15-55。
下面是我用SNIFFER抓的ARP包，一共有四个，我们只看前两个，第一个是从本机向路由器发起的一个ARP请求，第二个是路由器收到请求后做出的响应：

.





在分析 ARP 报文之前，有必要先对 ARP 报文的结构给大家介绍一下，参照下表：

1	2	3	4	5	6	7	8	9	10	11	12
6字节	6字节	2字节	2字节	2字节	1字节	1字节	2字节	6字节	4字节	6字节	4字节
目的MAC	源MAC	协议类型	硬件类型	协议类型	硬件地址长度	协议地址长度	操作类型	发送者MAC	发送者IP	接收者MAC	接收者IP

前面红色部分属于二层帧头的内容，黑色部分属于三层 IP 头内容（不知道二层和三层是什么？赶快去翻 OSI 分层模型！现在知道 OSI 的重要了吧！）

各字段作用：

1. 目的 MAC：该 ARP 帧将要到达的目的 MAC 地址
2. 源 MAC：发起该 ARP 帧的源主机接口的 MAC 地址
3. 协议类型：用来描述该帧将用什么协议来处理。如果值为 806,则表示是将该帧交给 ARP 协议处理，也就是说这个帧是一个 ARP 帧
4. 硬件类型：用来描述二层协议，当值为 1 时表示是 10M 以太网（100M 的帧结构和 10M 一样，所以也是 1）
5. 协议类型：用来描述多种三层协议，比如 IP、IPX、Apple Talk 等等（IP 为 800）
6. 硬件地址长度：指定二层地址的长度，MAC 地址为 6 字节，所以值为 6
7. 协议地址长度：指定三层地址的长度，IP 地址为 4 字节，所以值为 4

8. 操作类型：描述该数据包所做的 ARP 动作（请求为 1、响应为 2）
9. 发送者 MAC：告诉对方自己是谁（二层），这样对方可以知道是谁对它做了 ARP 动作，正常情况下是发起该 ARP 包的主机接口的 MAC 地址，但如果是病毒或黑客程序，则会更改此字段以达到隐藏自己的目的
10. 发送者 IP：告诉对方自己是谁（三层），这样对方可以知道是谁对它做了 ARP 动作，正常情况下是发起该 ARP 包的主机接口的 IP 地址，但如果是病毒或黑客程序，则会更改此字段以达到隐藏自己的目的
11. 接收者 MAC：最终接收该 ARP 包的网络接口的 MAC 地址
12. 接收者 IP：最终接收该 ARP 包的网络接口的 IP 地址

是不是很晕？晕就对了，呵呵，不要急，慢慢往下看。

终于可以进入主题了，现在我们先看第一个抓图，看看前 6 字节（表里的第一个字段），值为 FF-FF-FF-FF-FF-FF，对应表里的目的 MAC，这个全 F 的 MAC 是个什么地址呢？我们把这换算成二进制就得到了 48 个 1,在网络的地址规则中，一般遇到全 1 的都是广播地址，这里也不例外，它是一个二层的广播地址。为什么是广播呢？文章开头说了，只有当 ARP 表里没有此次操作的目的主机所对应的 ARP 条目（即目的主机 IP 已知但 MAC 地址未知）的情况下，就会发起 ARP 查询，由于不知道对方的 MAC 地址，那么怎么才能发送到对方去呢？为了保证目的主机能收到，所以只好用广播了。但是第二个图里是一个单播 MAC 地址，为什么是单播呢？原因是这个包是一个 ARP 响应包。由于第一个是 00-60-6e-30-15-55 向 FF-FF-FF-FF-FF-FF 请求 192.168.1.1 这个 IP 对应的 MAC 地址，所以当拥有 192.168.1.1 这个 IP 的主机接收到这个广播 ARP 请求后，看到是 00-60-6e-30-15-55 请求自己的 MAC 地址，所以它可以直接向 00-60-6e-30-15-55 发送回单播 ARP 响应，这样就节省了不必要的浪费（广播是非常浪费资源的）。

再看紧接着的 6 字节（表里第二字段），值为 00-60-6e-30-15-55，这个 MAC 地址为发起该 ARP 的主机接口的 MAC 地址，即源 MAC 地址。这个 MAC 地址也就是我自己这台电脑（192.168.1.100）的 MAC 地址。

第三字段 2 字节，是协议类型，ARP 的类型值为 806H（H 表示十六进制）,所以我们可以看到图中的值为 0806，表示该帧为 ARP 帧。

第四字段 2 字节，是硬件类型，IP 头从该字段开始。由于我们用的是标准以太网（快速以太网也是标准的以太网帧），所以值为 1H（0001），表示是 10M 以太网（实际是 100M）。

第五字段 2 字节，是协议类型，我们用的是 IP 协议，，IP 对应的值是 800H，所以显示值为 0800。

第六字段 1 字节，为硬件地址长度偏移，这个值告诉处理该帧的程序，读取硬件地址时读到哪里结束。MAC 地址为 6 字节，所以该字段的值就为 6，程序从硬件地址字段开始读取，读完 6 字节后认为该硬件地址已经读取完毕了。

第七字段 1 字节，为协议地址长度偏移，这个值告诉处理该帧的程序，读取协议地址时读到哪里结束。IP 地址为 4 字节，所以该字段的值就为 4，程序从协议地址字段开始读取，读完 4 字节后认为该协议地址已经读取完毕了。

第八字段 2 字节，为操作类型，ARP 提供几种操作类型，用来表示这个报文的类型，ARP 请求为 1，ARP 响应为 2，RARP 请求为 3，RARP 响应为 4，我们一般常用的只有前两种。第一个图里是 1H（0001），所以这是一个 ARP 请求，第二个图里是 2H（0002），所以这是一个

ARP 响应。

第九字段 6 字节，为发送者 MAC 地址，该地址用来告诉本次操作的对方，是哪个 MAC 地址对它进行了操作。

第十字段 4 字节，为发送者 IP 地址，该地址用来告诉本次操作的对方，是哪个 IP 地址对它进行了操作。

第十一字段 6 字节，为接收者 MAC 地址，用来告诉本次操作的对方，是哪个 MAC 地址应该接收并处理该帧。

第十二字段 4 字节，为接收者 IP 地址，用来告诉本次操作的对方，是哪个 IP 地址应该接收并处理该 ARP 报文。

是不是发现后面还有很多 0 呢？呵呵，其实不用奇怪，那些 0 是没有用的，它只是做个填充的作用，由于 IP 报文规定最小不能小于 60 字节（为什么？自己查 TCP/IP），而 ARP 只用了 42 个字节，所以它需要用 0（其实不一定是“0”）来填充剩下的 12 字节。

好了，ARP 报文介绍完毕，现在我们来讲一下 ARP 的过程。

一. 当 A 要访问 R，这时，A 的 ARP 表是空的，即没有 R 的相关条目，所以为了产生 ARP 条目，A 要向网络中发送一个 ARP 请求广播。

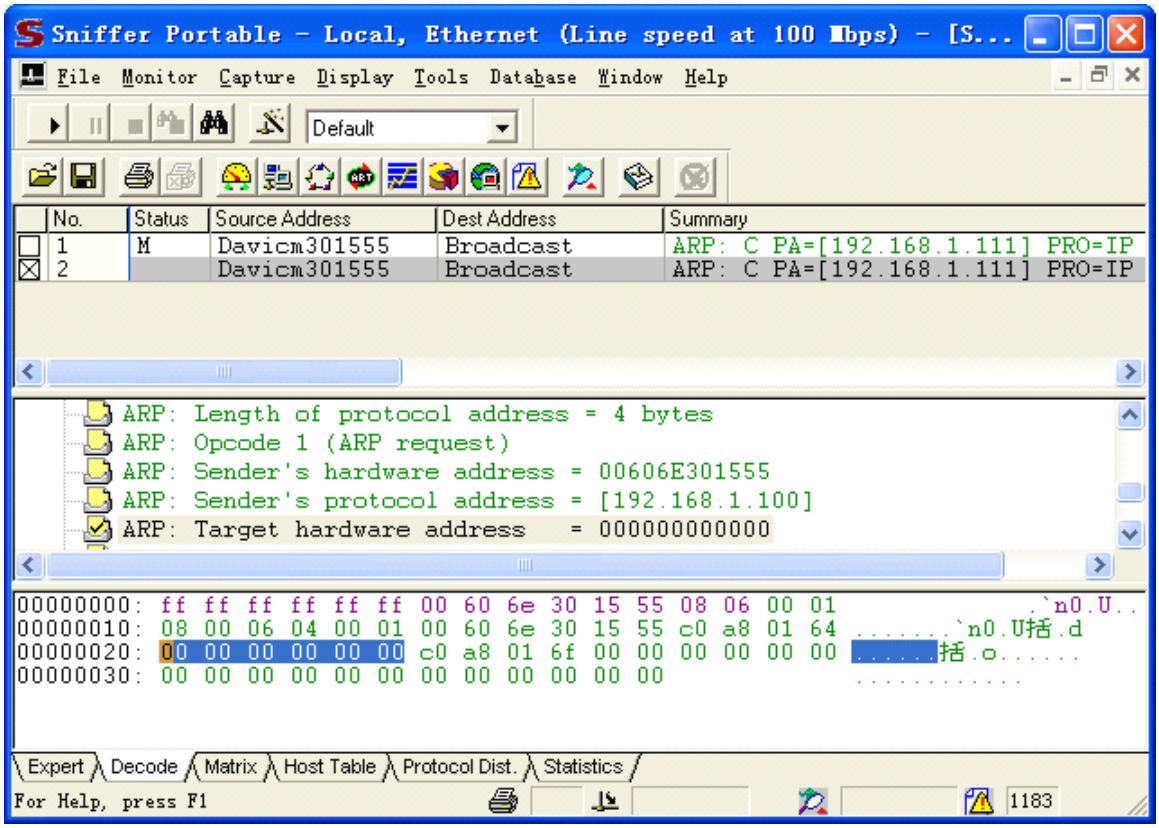
二. B 和 R 都会收到这个广播。因为是广播，所以网络上的任何一个节点都要对该帧进行进一步处理。于是 B 检查该帧的协议类型，发现是 806H（ARP），便把该帧的帧头去掉，检查其 IP 头，发现是发向 192.168.1.1 的 ARP 报文而不是自己的，于是将该报文丢弃。

三. 而 R 收到这个帧后发现是个广播，所以也会检查该帧的协议类型，发现是 806H（ARP），便把该帧的帧头去掉，露出 IP 头，发现该报文是发给自己的，就再检查操作类型，发现是 1（请求），于是便读取该帧的发送者 MAC 和发送者 IP，把它们作为 ARP 响应的目的 MAC 和目的 IP 发送回去，这就是为什么 ARP 请求是广播，而响应是单播的原因。有没有发现这个帧的接收者 MAC 字段为全 0 呢？这是由于发送者不知道接收者的 MAC 地址，所以把这个字段留空，让接收者自己填写。

四. 当 A 收到来自 R 的 ARP 单播响应时，首先检查该帧的目的 MAC 是不是自己，发现匹配，于是再检查协议类型，发现是 ARP 协议，于是去掉二层帧头，送到 IP 层，IP 协议发现目的 IP 地址是自己，于是进一步检查操作类型，发现是 2（响应），于是读取发送者 MAC 和发送者 IP，把它们记录进 ARP 表。

至此，一次 ARP 过程完成。

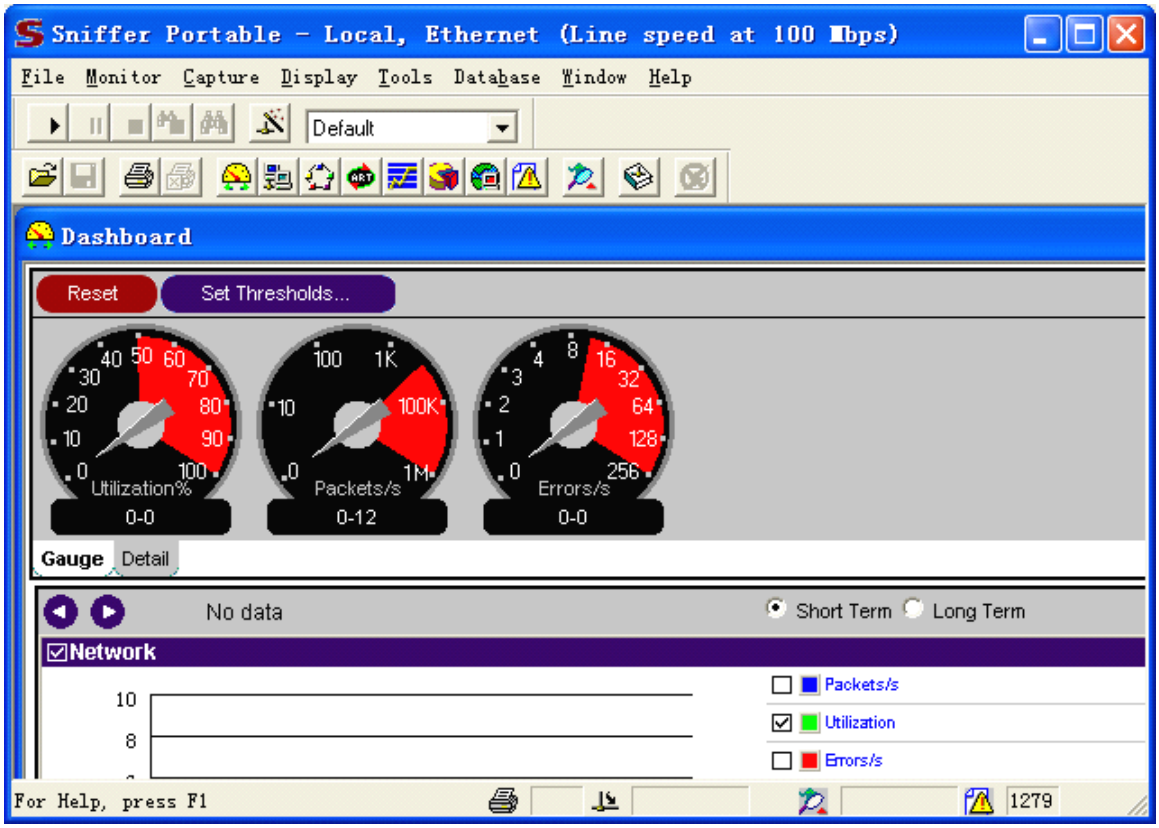
搞了这么多事，ARP 的原理算是搞清楚了，那么我们来分析一个帧：



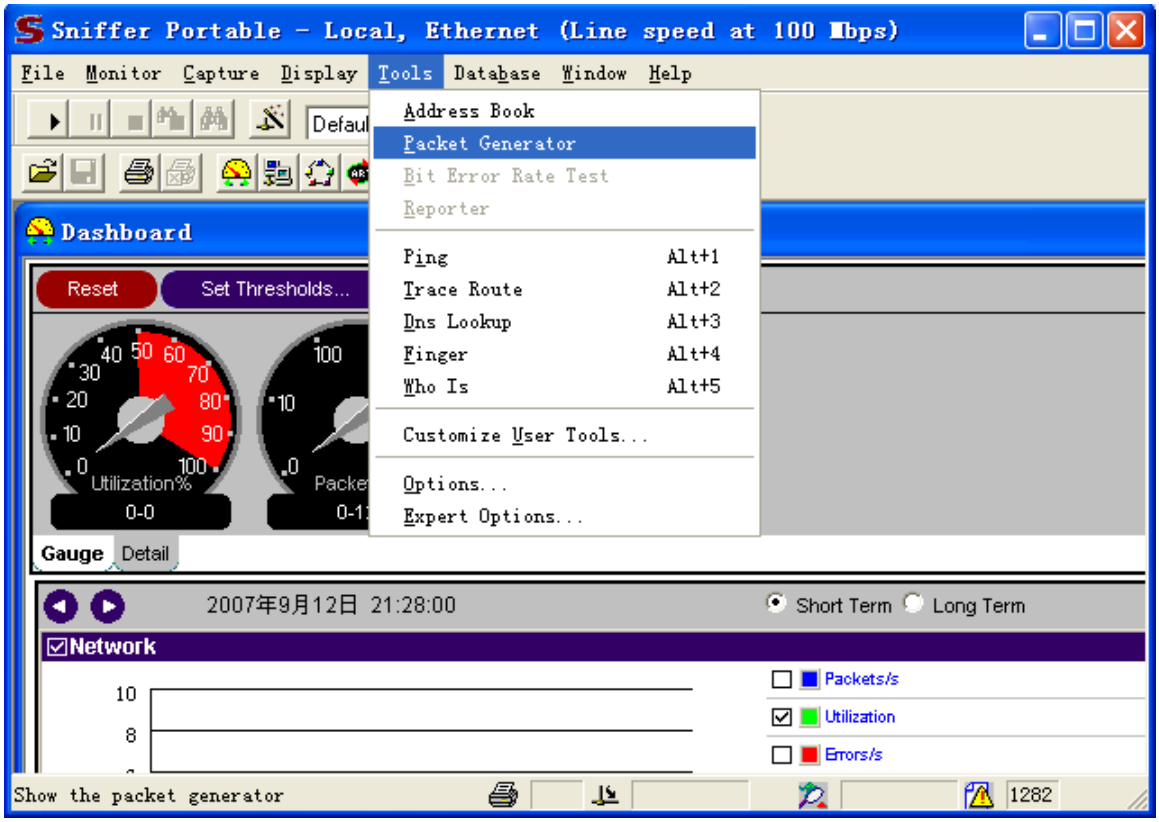
第一个字段是全 1,证明这是一个广播，源 MAC 当然是发送者的 MAC 地址，0806 说明是 ARP 协议，0001 说明是 10M 以太网（实际是 100M），0800 说明是 IP 报文，06 说明物理地址（MAC）的长度是 6 字节，04 说明协议地址（IP）长度是 4 字节，0001 说明这是一个 ARP 请求，发送者 MAC、发送者 IP 这个不用说了吧，这个接收者 MAC 是全 0,原因上面也说了，是由于发起者不知道对方的 MAC，所以只能留空给接收者自己填了，目的 IP 地址当然就是接收者的 IP 地址了，后面 18 字节是用来填补 60 字节的最小 IP 报文长度的。

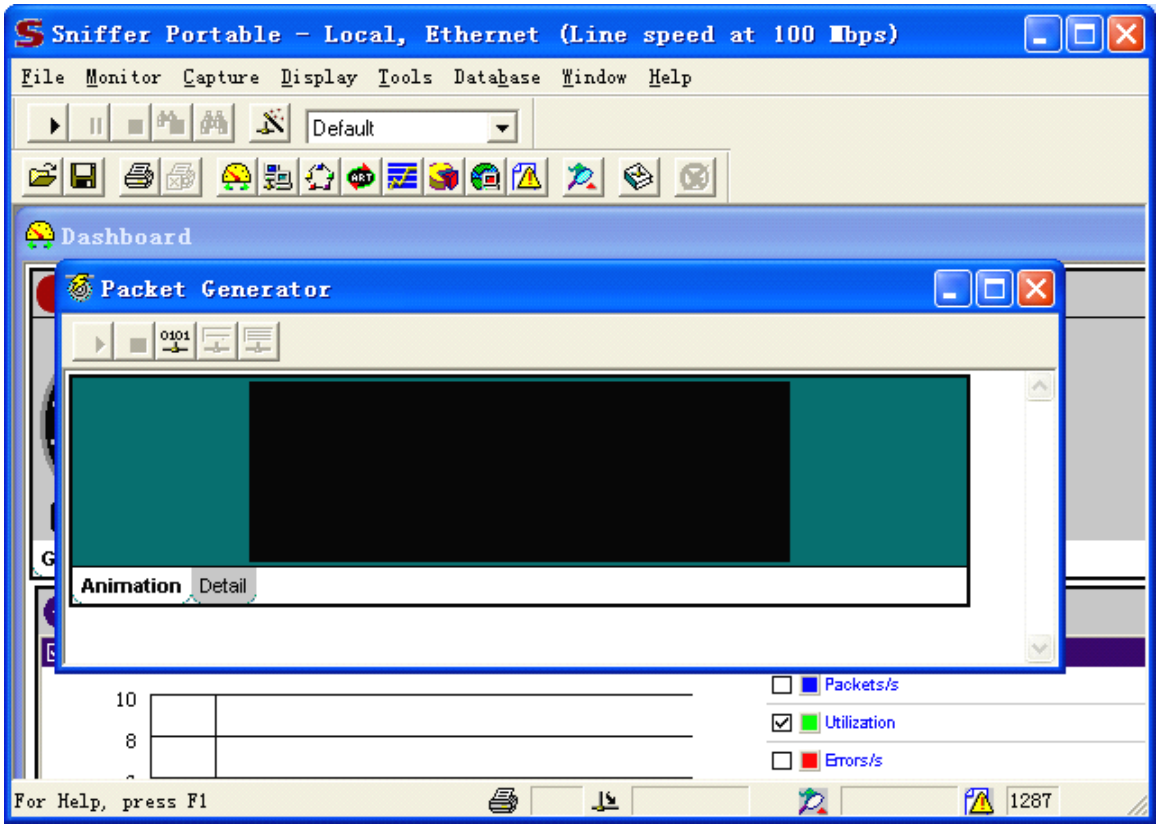
学会分析 IP 报文了，那么我们进一步，来学习怎么伪造它吧！（玩点过瘾的，哈哈！）

现在我们回到最上面的拓扑图里去，我们现在唯一能操作的设备只有计算机 A，我想让计算机B 上不了网怎么办？现在请我们的主角 SNIFFER 出场（灯光！掌声！鲜花！）！

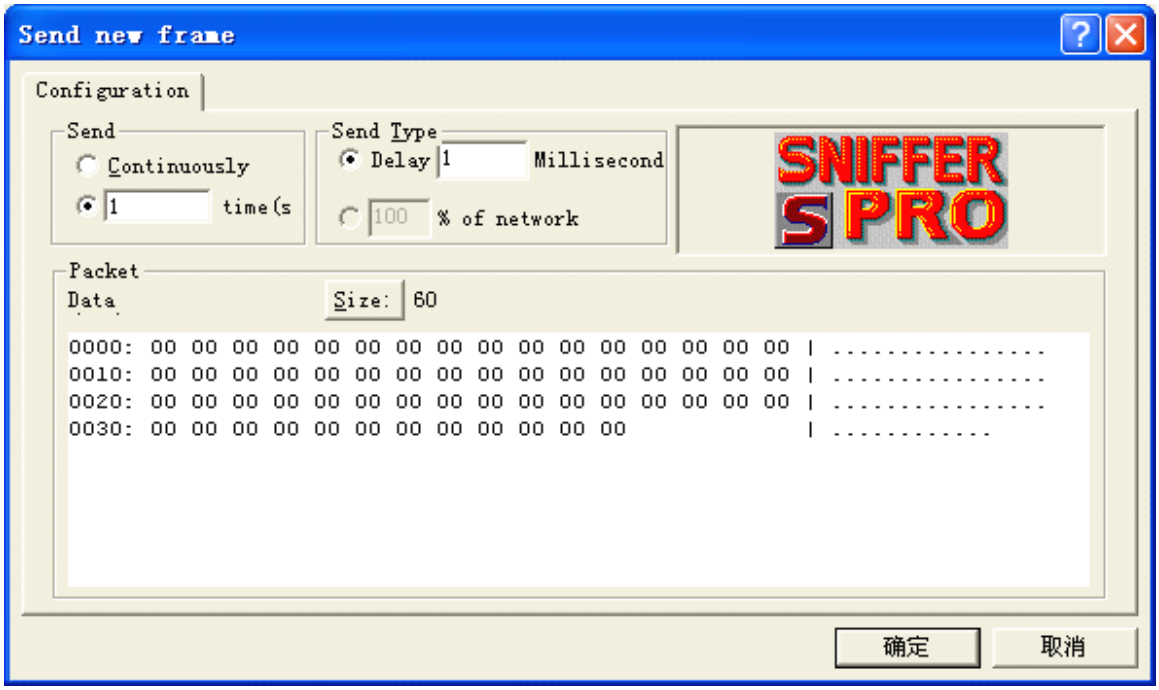


这个就是 SNIFFER 的主界面，我们现在要用到它的包产生器“Packet Generator”：



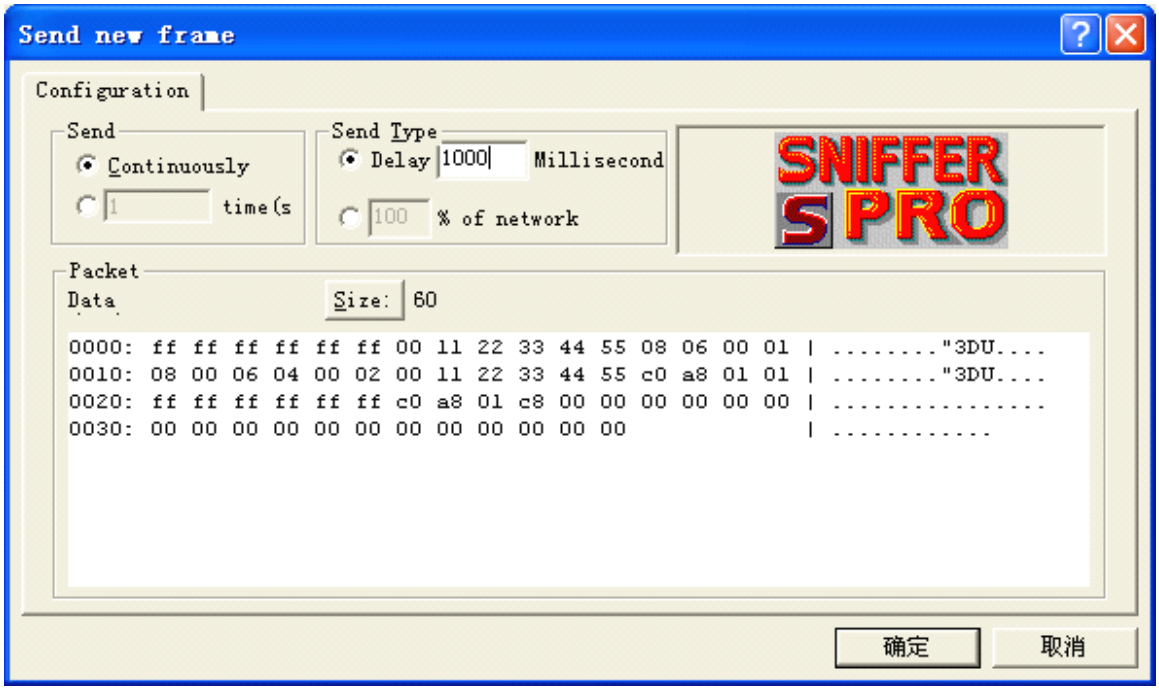


然后点那个 0101 的图标，出现一个包编辑器：



现在我们就可以进行数据的编辑了，为了要达到让主机 B 无法上网，我们要欺骗 B 的 ARP 表，让它产生一条错误的对于网关的 ARP 条目。先看看这几个选项吧，Continuously 是连续发送选项，下面的 time(s 是发送数据的数量，右边的 Delay 是指隔多少毫秒发送一个数据。下面百分

比是分多少网络带宽给 SNIFFER 用。为了保证我们伪造的 ARP 信息不被正确的 ARP 信息覆盖，我们选连续发包，间隔为 1000 毫秒（1 秒发送一个）。



我们来分析一下这个 ARP 包：

a. 字段 1：目的 MAC 为全1，由于我们不知道主机B 的MAC 是多少，只知道 IP 是 192.168.1.200,为了保证对方能收到这个 ARP，所以我们填一个广播地址。

b. 字段 2：源 MAC 为一个随意伪造的 MAC（做坏事当然不能留真实姓名啦^_^），最主要的还不是这个原因，因为源 MAC 和第 9 字段的发送者 MAC 如果不一致，遇到高手就露馅了，嘿嘿！

c. 字段 3－7：略过，自己查上面的表。

d. 字段 8：操作类型为 0002,说明这个包为一个 ARP 响应，因为只有 ARP 响应包携带的信息才会被接收者放入 ARP 表，所以我们才要伪造 ARP 响应而不是 ARP 请求。

e. 字段 9：这个就不用我说了吧！发送者 MAC 当然是伪造的，最好和源 MAC 一致，这样不容易露馅。

f. 字段 10：发送者 IP 当然是网关的，因为这个是要被放到接收者的 ARP 表里去的，这里的接收者当然就是主机 B－192.168.1.200 啦！由于我们的网关是 192.168.1.1,所以这里我们填 16 进制数：C0 A8 01 01，对应十进制的 192 168 1 1

g. 字段 11：接收者 MAC 地址不能再填 0 了，因为这是个 ARP 响应，既然是响应，就表示不需要再让对方填写该字段了，所以不能是 0,那我们 填个广播吧，为了和前面的目的 MAC 一致。

h. 字段 12：接收者 IP 地址当然要填 192.168.1.200 了，因为我们是要把这个 ARP 发给主机 B,所以填 B 的 IP 地址，其值就是对应的 16 进制 C0 A8 01 C8 了

我们再来分析一下主机 B 收到这个 ARP 响应后都做了些什么：

1) 检查目的 MAC 地址，发现是广播，于是开始处理这个帧
2) 检查协议类型，发现是 ARP 协议，于是送给三层（IP）处理
3) IP 协议检查接收者协议地址（IP 地址），发现是自己，于是进一步处理
4) 检查操作类型，发现是 0002（ARP 响应），于是读取包内的发送者 IP 和发送者 MAC
5) 把发送者 IP 和发送者 MAC 写入 ARP 表内
6) 没了

现在在主机 B 上的命令行里用 arp -a 命令可以看到一条错误的 ARP 条目了：
C:\>arp -a

Interface: 192.168.1.200 --- 0x*****

Internet Address	Physical Address	Type
192.168.1.1	00-11-22-33-44-55	dynamic

由于主机 B 的网关设置为 192.168.1.1 了，而它的 ARP 表的对应 MAC 是一个不存在的 MAC，所以非本网段的流量都会发往 00-11-22-33-44-55 这个假 MAC 地址，结果当然这个 MAC 地址不会给出任何响应（它都不存在，怎么响应？），主机 B 当然也就上不了网了（除了本网段）！

聚生网管等网管软件其实就是利用的这个原理实现网管功能的。它们之所以能够安装在网络任何一台主机上也能控制网络流量，其实就是通过 ARP 欺骗网络中所有其它主机把网关的 MAC 地址指向安装有网管软件的那台主机，这样，其它主机去往外网的流量都被导向这台主机了，再通过这台主机中转出去。其实它就是一个代理软件，只不过多了一个 ARP 欺骗功能而已，具体如何实现的请各位结合上面的实验自己思考了，嘿嘿！