



# 网络编程基础

如何使用 WireShark 抓包分析

# 目录

- 背景介绍
- 程序安装
- 程序使用



# 背景介绍

# 背景介绍

- 网络世界的数据实际上都是封装成一个个的数据包，你发给我一个数据包，我传给你一个数据包。
- 因此，在网络编程的过程中，经常需要利用抓包工具对开发板发出或接收到的数据包进行抓包分析。
- **wireshark** 是一个非常好用的抓包工具，使用 **wireshark** 工具抓包分析，是学习网络编程必不可少的一项技能。



# 程序安装

# 安装 wireshark

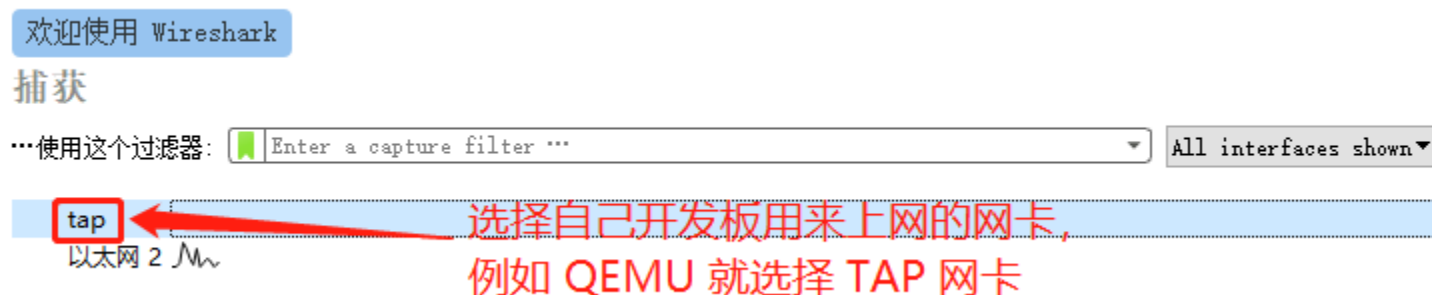
- 下载路径：<https://www.wireshark.org/>
- 下载完 wireshark，一路默认安装就行。
- 安装完软件之后，双击打开 wireshark 软件。



# 程序使用

# 选择与开发板相对应的网卡

- 打开 wireshark 之后，会给出你的所有网卡信息，让你选择一个要抓包的网卡
- 如下图所示，选择自己开发板用来上网的网卡，双击就开始抓包了。





# wireshark 主界面介绍

- wireshark 的主界面和一般的 Windows 软件很像，也有菜单栏，工具栏，地址栏等。
- 还有一些它自己特有的窗口，比如：显示过滤器、封包列表、封包详细信息、十六进制数据显示区等。
- 如图所示，选择完网卡之后其实就已经开始抓包了。

# wires

- wiresha  
栏等。
- 还有一些  
十六进制
- 如图所

正在捕获 tap

开始抓包 重新开始抓包

0. 工具栏

1. 显示过滤器

Apply a display filter ... <Ctrl-/> 表达式...

2. 封包列表

No.	Time	Source	Destination	Protocol	Len	Info
1	0.000000	00:ff:1a:62:95:7a	LLDP Multicast	LLDP	58	TTL = 3601
2	0.010115	00:ff:1a:62:95:7a	Broadcast	ARP	42	Who has 192.168.137.1? Tell
3	0.010177	::	ff02::1:ff93:1998	ICMPv6	78	Neighbor Solicitation for f
4	0.010204	fe80::d462:3cb4:193:1998	ff02::2	ICMPv6	62	Router Solicitation
5	0.010239	fe80::d462:3cb4:193:1998	ff02::16	ICMPv6	1...	Multicast Listener Report M
6	0.025950	fe80::d462:3cb4:193:1998	ff02::1:2	DHCPv6	1...	Solicit XID: 0x198319 CID:
7	0.510162	fe80::d462:3cb4:193:1998	ff02::16	ICMPv6	1...	Multicast Listener Report M
8	0.523856	0.0.0.0	255.255.255.255	DHCP	3...	DHCP Discover - Transaction
9	0.526268	192.168.137.1	192.168.137.135	DHCP	3...	DHCP Offer - Transaction
10	0.526572	0.0.0.0	255.255.255.255	DHCP	3...	DHCP Discover - Transaction

3. 封包详细信息

> Frame 3: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0

> Ethernet II, Src: 00:ff:1a:62:95:7a (00:ff:1a:62:95:7a), Dst: IPv6mcast\_ff:93:19:98 (33:33:ff:93:19

> Internet Protocol Version 6, Src: ::, Dst: ff02::1:ff93:1998

> Internet Control Message Protocol v6

4. 十六进制数据

0000	33 33 ff 93 19 98 00 ff	1a 62 95 7a 86 dd 60 00	33..... -b-z...`
0010	00 00 00 18 3a ff 00 00	00 00 00 00 00 00 00 00	.....:.....
0020	00 00 00 00 00 00 ff 02	00 00 00 00 00 00 00 00	.....
0030	00 01 ff 93 19 98 87 00	35 ba 00 00 00 00 fe 80	..... 5.....
0040	00 00 00 00 00 00 d4 62	3c b4 01 93 19 98	.....b <.....

5. 地址栏

tap: <live capture in progress> 分组: 131 · 已显示: 131 (100.0%) Profile: Default

具栏，地址

详细信息、

# 显示过滤器的使用

- 合理的使用显示过滤器，有助于我们从下面的封包列表里快速找到我们要关注网络包。
- 假如我们的开发板获取到的 IP 地址为 192.168.137.135 ， 这样在显示过滤器里输入 `ip.src == 192.168.137.135 or ip.dst == 192.168.137.135` 然后点击右边的那个小箭头就会执行这个过滤条件。
- 过滤之后，剩下的都是和我们限定的 IP 相关的包。
- 当然也有一些其他的过滤条件，比如：
  - 按 mac 地址过滤 `eth.addr == 52:54:00:11:22:33`
  - 只显示 TCP 协议的数据包 `tcp`
  - 只显示 UDP 协议的数据包 `udp`

# 封包列表介绍

- 从封包列表里我们也可以看到一些跟封包相关的信息，例如：来源的 ip，还有用到的通信协议等。

# 封包列

- 从封包列用到的通

The screenshot shows a Wireshark interface with a live capture of traffic on the 'tap' interface. The filter is set to 'ip.src == 192.168.137.135 or ip.dst == 192.168.137.135'. The packet list pane shows 19 packets, with columns for No., Time, Source, Destination, Protocol, Length, and Info. Red boxes highlight the 'No.' column (labeled '包序号'), the 'Time' column (labeled '接收到的时间'), the 'Source' column (labeled '源ip地址'), and the 'Destination' column (labeled '目标ip地址'). The 'Info' column is labeled '包的详细信息'. The details pane for packet 4 shows the following structure:

- > Frame 4: 344 bytes on wire (2752 bits), 344 bytes captured (2752 bits) on interface 0
- > Ethernet II, Src: 00:ff:1a:62:95:7a (00:ff:1a:62:95:7a), Dst: RealtekU\_11:22:33 (52:54:00:11:22:33)
- > Internet Protocol Version 4, Src: 192.168.137.1, Dst: 192.168.137.135
- > User Datagram Protocol, Src Port: 67, Dst Port: 68
- > Bootstrap Protocol (ACK)

The hex dump at the bottom shows the raw bytes of the packet, with ASCII characters on the right side.

J ip, 还有

# 封包详细信息介绍

- 下面介绍一下封包详细信息那一栏，它和底下的十六进制数据是相通的。封包的详细信息就是从底下的实际数据解包显示出来的。下面这张图展示了封包的一些详细信息。

# 封包详

- 下面介绍的详细作一些详细

The image shows a Wireshark capture of network traffic. The main pane displays a list of packets, with packet 15 selected. The packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
8	54.396312	192.168.137.1	192.168.137.135	DHCP	344	DHCP ACK - Transaction ID 0x4
11	109.026638	192.168.137.135	192.168.137.1	DHCP	350	DHCP Request - Transaction ID 0x4
12	109.034830	192.168.137.1	192.168.137.135	DHCP	344	DHCP ACK - Transaction ID 0x4
13	154.652599	192.168.137.135	192.168.137.1	DNS	73	Standard query 0x5db4 A www.baidu.
14	154.653418	192.168.137.1	192.168.137.135	DNS	132	Standard query response 0x5db4 A w
15	154.656395	192.168.137.135	111.13.100.91	ICMP	74	Echo (ping) request id=0xafaf, se
16	154.682845	111.13.100.91	192.168.137.135	ICMP	74	Echo (ping) reply id=0xafaf, se
17	154.878592	192.168.137.135	111.13.100.91	ICMP	74	Echo (ping) request id=0xafaf, se
18	154.904847	111.13.100.91	192.168.137.135	ICMP	74	Echo (ping) reply id=0xafaf, se
19	155.101376	192.168.137.135	111.13.100.91	ICMP	74	Echo (ping) request id=0xafaf, se

The packet details pane for packet 15 shows the following structure:

- Frame 15: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0 **物理层数据**
- Ethernet II, Src: RealtekU\_11:22:33 (52:54:00:11:22:33), Dst: 00:ff:1a:62:95:7a (00:ff:1a:62:95:7a) **数据链路层以太网帧头部信息**
- Internet Protocol Version 4, Src: 192.168.137.135, Dst: 111.13.100.91 **互联网层IP包头部信息**
- Internet Control Message Protocol **ICMP协议**

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000  00 ff 1a 62 95 7a 52 54 00 11 22 33 08 00 45 00  ...b.zRT .."3..E.
0010  00 3c 00 24 00 00 ff 01 9e 04 c0 a8 89 87 6f 0d  <.$.....o.
0020  64 5b 08 00 57 4e af af 00 01 00 01 02 03 04 05  d[.WN.....
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15  .....
0040  16 17 18 19 1a 1b 1c 1d 1e 1f  .....

```

的。封包  
了封包的