Wireshark ZigBee Sniffer

Configuration and Use

Table of Contents

Overview

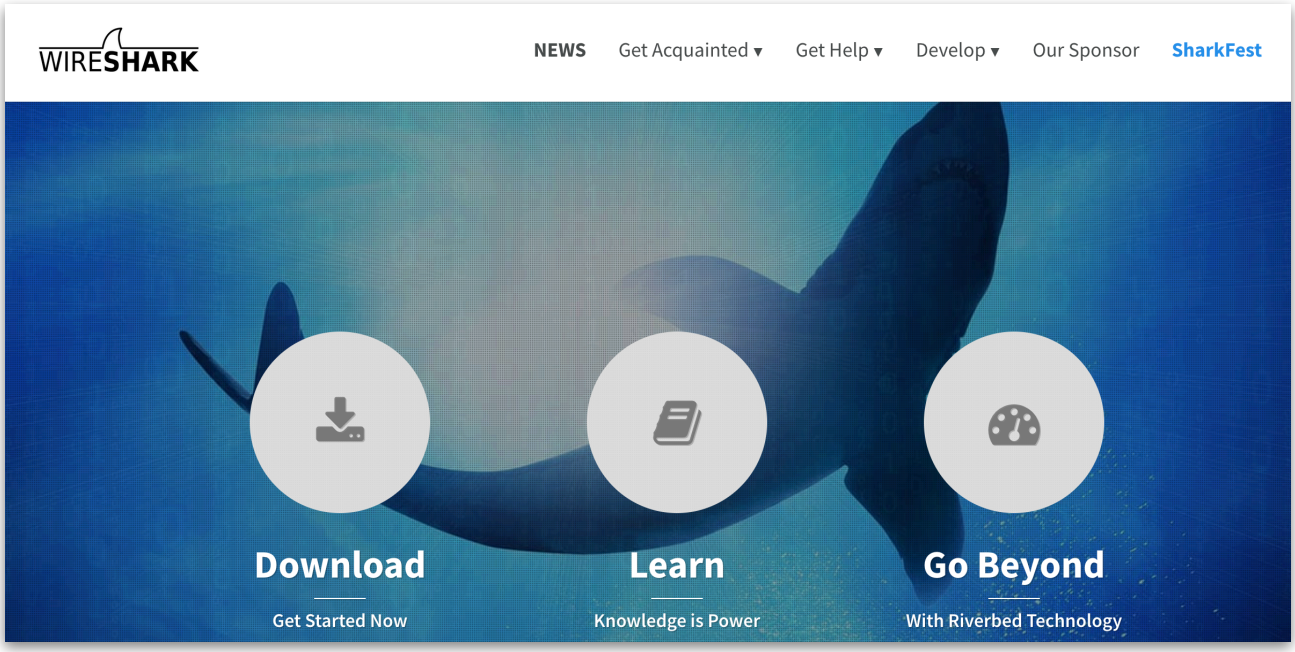
A ZigBee sniffer allows capture and display of data frames transmitted between ZigBee devices. It captures the data received with a separate radio from that used by the main system, and can display low level frames that can be useful for debugging problems on the network. Often these low level issues can not be otherwise debugged using higher level APIs that connect to the coordinator as this information is simply not provided.

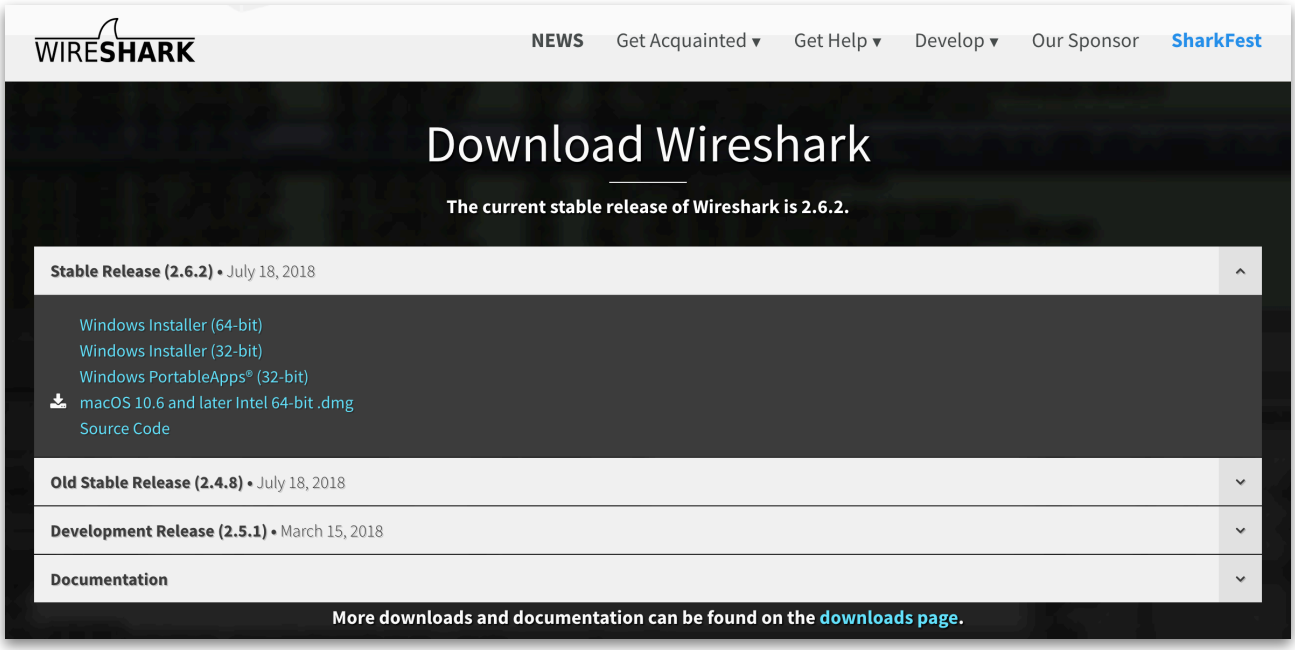
This document provides information on setting up and using the Wireshark software. Different hardware is available for use with this including the Texas Instruments CC2531 dongle that is commonly available at low cost in online marketplaces such as eBay and AliExpress, and the CEL MeshConnect dongle, or other dongles using the Silabs Ember chipset can also be used.

There are many resources available for using Wireshark, and this document does not intend to replace a good understanding of the software or protocol analyses principals. It does however provide the user with a basic understanding of ZigBee and Wireshark a quick start guide to packet sniffing for the purposes of providing feedback to Z-Smart Systems when reporting ZigBee issues.

Wireshark

Wireshark is a packet capture and analyses tool that can be downloaded freely from the web. It is capable of displaying the contents of ZigBee packets, and allows the debugging of low level protocol issues. Wireshark can be downloaded from https://www.wireshark.org/.

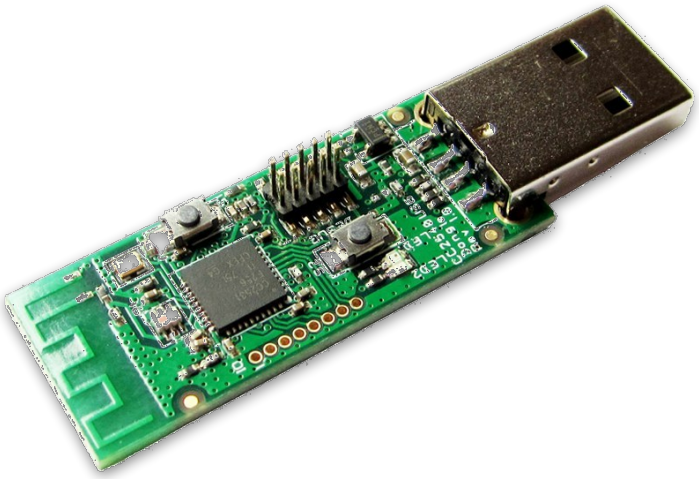




Dongle Specific Sniffers

Wireshark does not directly interact with ZigBee hardware - it requires a dongle and associated sniffer software to provide it with the data to analyse. The sniffer is very simple - it is only grabbing the low level frames from the dongle and passing them to Wireshark. Note that when used as a sniffer, the dongle is not running in its standard mode and is not used as a node on a ZigBee network. It will normally either run different firmware, or be configured into a specific mode to provide the low level data required.

Texas CC2531



Sniffer software in Python is available here -:

```
https://github.com/andrewdodd/ccsniffpiper
```

To install on a Mac OS X -:

brew install libusb

easy_install pip

pip install pyusb

Note that I haven't tried this software and it may create named pipes that mean Wireshark is started slightly differently than described below.

TBD…

Ember NCP Sniffer

The Z-Smart Systems Ember sniffer software is written in Java so can be run on any computer with a Java VM. It can be used with most standard Ember NCP firmware (any containing the mfglib library), and when running will place the dongle into a special mode where low level frames are provided as required for the sniffer application.

The sniffer application can directly write the Wireshark *pcap* files and Silabs ISD files.

The sniffer is available on Github

```
https://github.com/zsmartsystems/com.zsmartsystems.zigbee.sniffer
```

The sniffer is a console application and requires configuring through the command line -:

```
usage: ZigBeeSniffer
 -?,--help                          Print usage information
 -a,--ipaddr <remote IP address>    Set the remote IP address
 -b,--baud <baud>                   Set the port baud rate
 -c,--channel <channel id>          Set the ZigBee channel ID
 -f,--flow <type>                   Set the flow control (none | hardware |
                                    software)
 -l,--local                         Log times in local time
 -m,--maxpcap <length>              Maximum filesize for Wireshark files
 -p,--port <port name>              Set the port
 -r,--ipport <remote IP port>       Set the remote IP port
 -s,--silabs <filename>             Log data to a Silabs ISD compatible
                                    event log
 -t,--timeout <seconds>             NCP restart timeout in seconds
 -w,--pcap <filename>               Log data to a Wireshark pcap compatible
                                    log
```

Note that the IP address will default to the local host on the assumption that you are running Wireshark on the same computer as the sniffer. The `ipport` will default to 17754 which is the port used for the ZigBee Encapsulation Protocol - changing this may stop Wireshark displaying ZigBee data.

Example command line -:

```
java -jar ZigBeeSniffer.jar -port /dev/tty.SLAB_USBtoUART -baud 115200 -flow hardware
```

The software will print an output to the console for each packet that is received to allow confirmation it is working. When running Wireshark, these should also be seen in the Wireshark window.

```
WiresharkZepFrame [sequence=00000000, lqi=255, data={41 88 41 EF CD FF FF 00 00 09 12 FC FF 00 00 01 81 01 00 00 00 00 08 22 00 28 00 10 01 00 01 00 00 00 00 08 22 00 00 29 B8 AC EB 6B B7 FF 80}]
WiresharkZepFrame [sequence=00000001, lqi=255, data={41 88 42 EF CD FF FF 00 00 09 12 FC FF 00 00 01 82 01 00 00 00 00 08 22 00 28 01 10 01 00 01 00 00 00 00 08 22 00 00 B2 19 57 C0 9B DF FF 80}]
WiresharkZepFrame [sequence=00000002, lqi=255, data={41 88 43 EF CD FF FF 00 00 09 12 FC FF 00 00 01 83 01 00 00 00 00 08 22 00 28 02 10 01 00 01 00 00 00 00 08 22 00 00 11 35 D3 A1 50 E0 FF 80}]
WiresharkZepFrame [sequence=00000003, lqi=255, data={41 88 44 EF CD FF FF 00 00 09 12 FC FF 00 00 01 84 01 00 00 00 00 08 22 00 28 03 10 01 00 01 00 00 00 00 08 22 00 00 26 82 40 4F 7C 8E FF 80}]
```

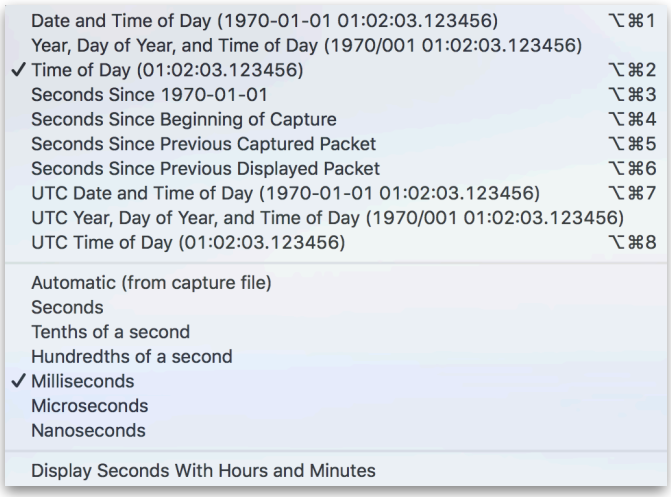
If the NCP fails to receive a valid frame with the timeout period set with the -t command line parameter, then the NCP will be restarted. This will allow the sniffer to recover from serial port or NCP communications problems. The timer defaults to 30 seconds.

Software Operation

Configuration

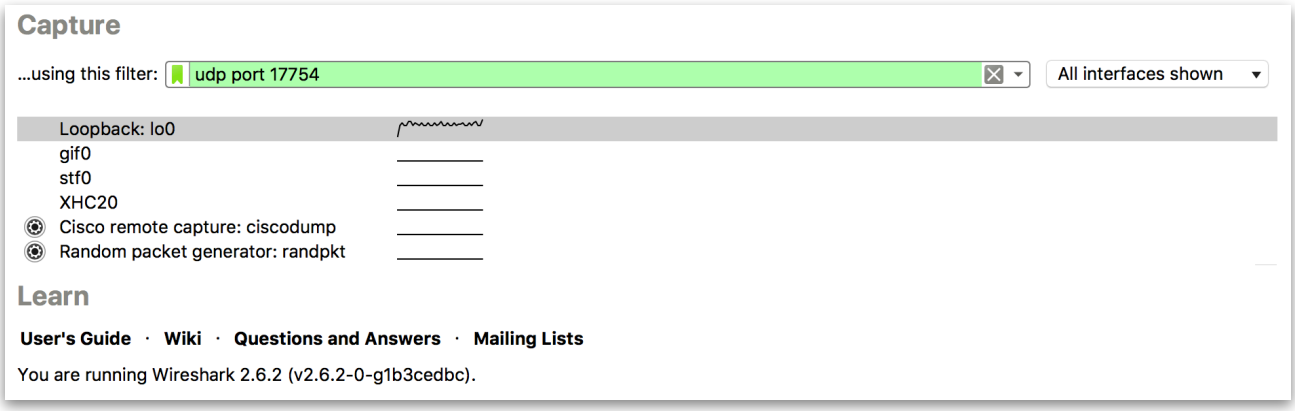
A few options that may be worth considering to make the software more usable -:

Time format - setting to time of day may help to coordinate different log files, such as the log generated by the Z-Smart Systems ZigBee framework. By default, the time will be in microseconds - milliseconds is probably sufficient.

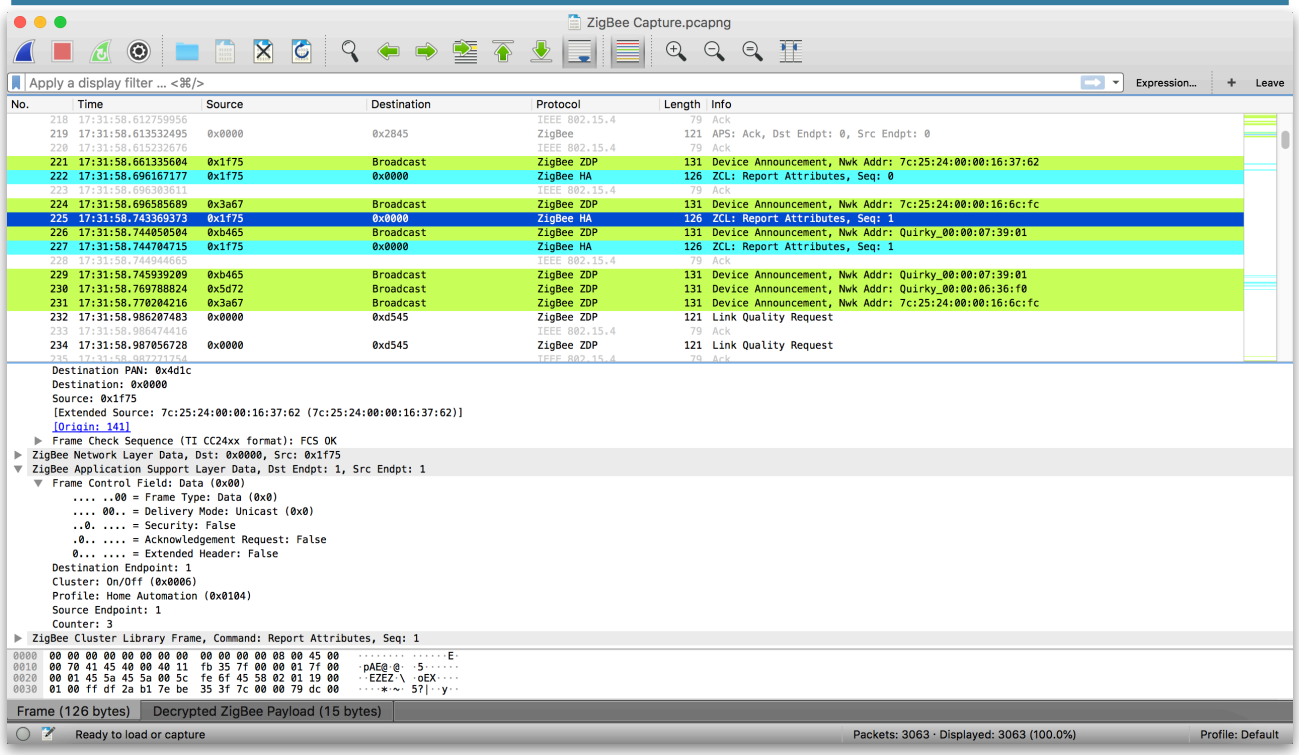


Getting Started

From the Wireshark start screen, you need to start receiving the ZEP (ZigBee Encapsulation Protocol) frames that are sent to port 17754. The filter `udp port 17754` will ensure that only ZigBee frames are captured. If you are running the sniffer on the same computer as Wireshark, then you will probably want to use the Loopback interface - otherwise select another suitable interface.



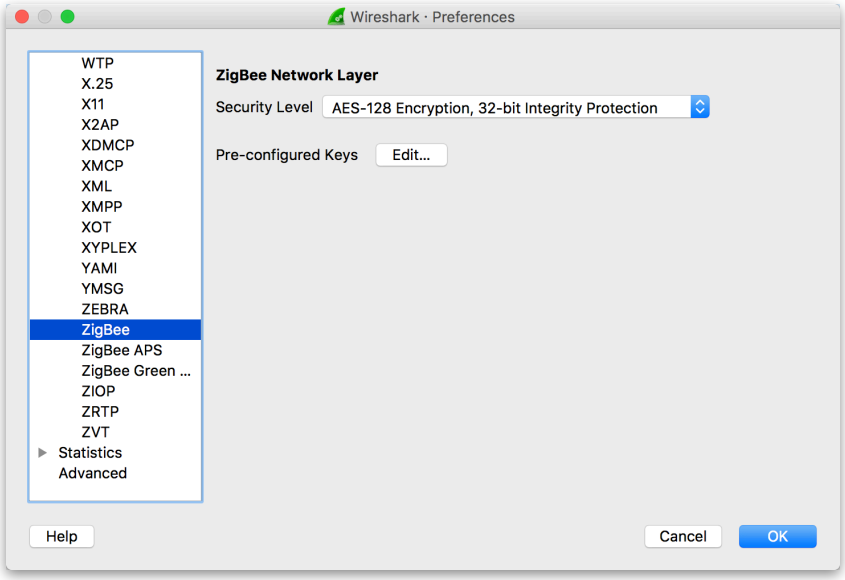
Once Wireshark starts to capture packets, they will be displayed in the main screen as seen below. This is broken into 3 main areas - the list of received packets, the detailed packet information, and the raw packet data.



Setting Security Keys

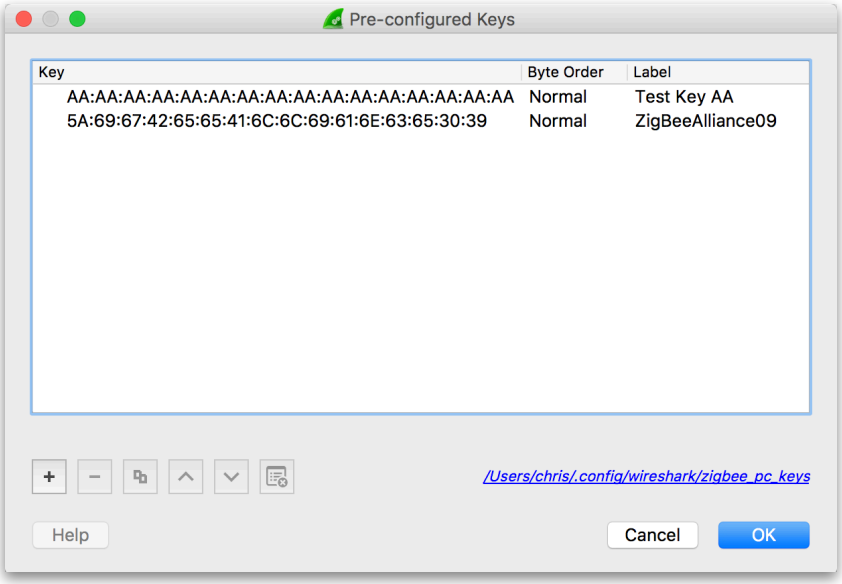
To properly decode ZigBee frames, the keys must be added to Wireshark preferences. If Wireshark doesn't know your keys, it will not display the full packet contents and will effectively be useless for analysing data.

At least the network key is required. Select the preferences, and go to the ZigBee Protocol preferences page. Select the Edit button to change the Pre-configured Keys.



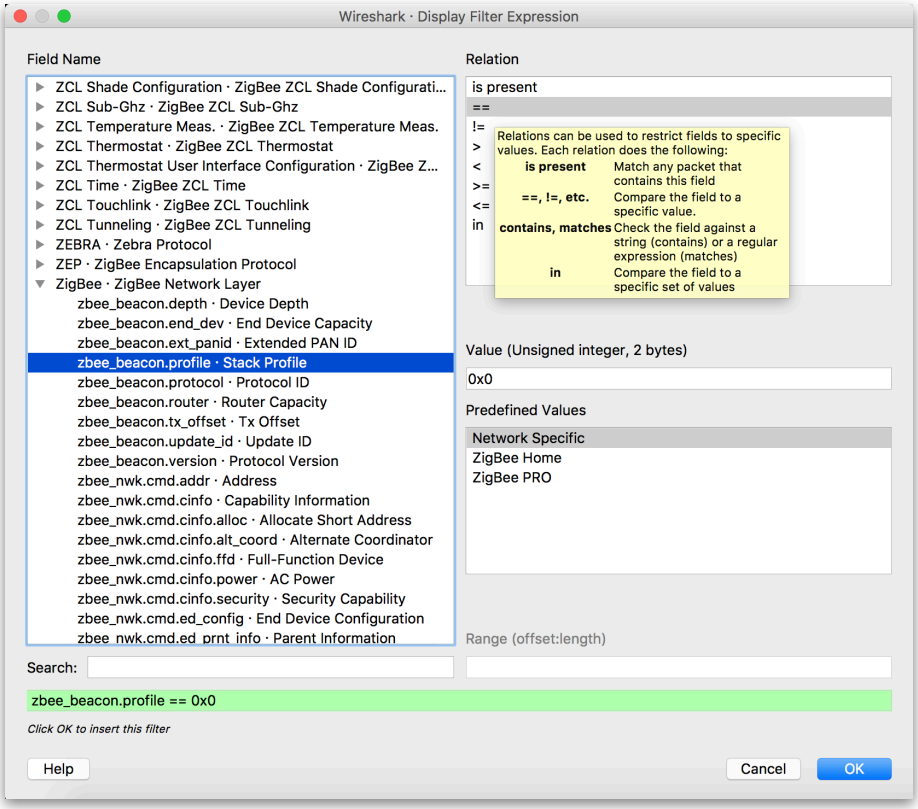
Enter your network key, and it is also advisable to enter the ZigBeeAlliance09 key that is normally used to allow ZHA devices to join the network.

```
5A:69:67:42:65:65:41:6C:6C:69:61:6E:63:65:30:39
```

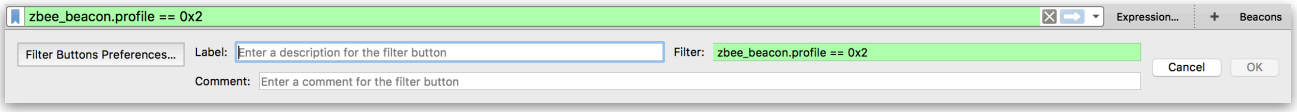


Filters

Wireshark has a powerful filter system to allow frames to be selectively displayed. The Expression... button in the filter bar will display the Filter Expression box.



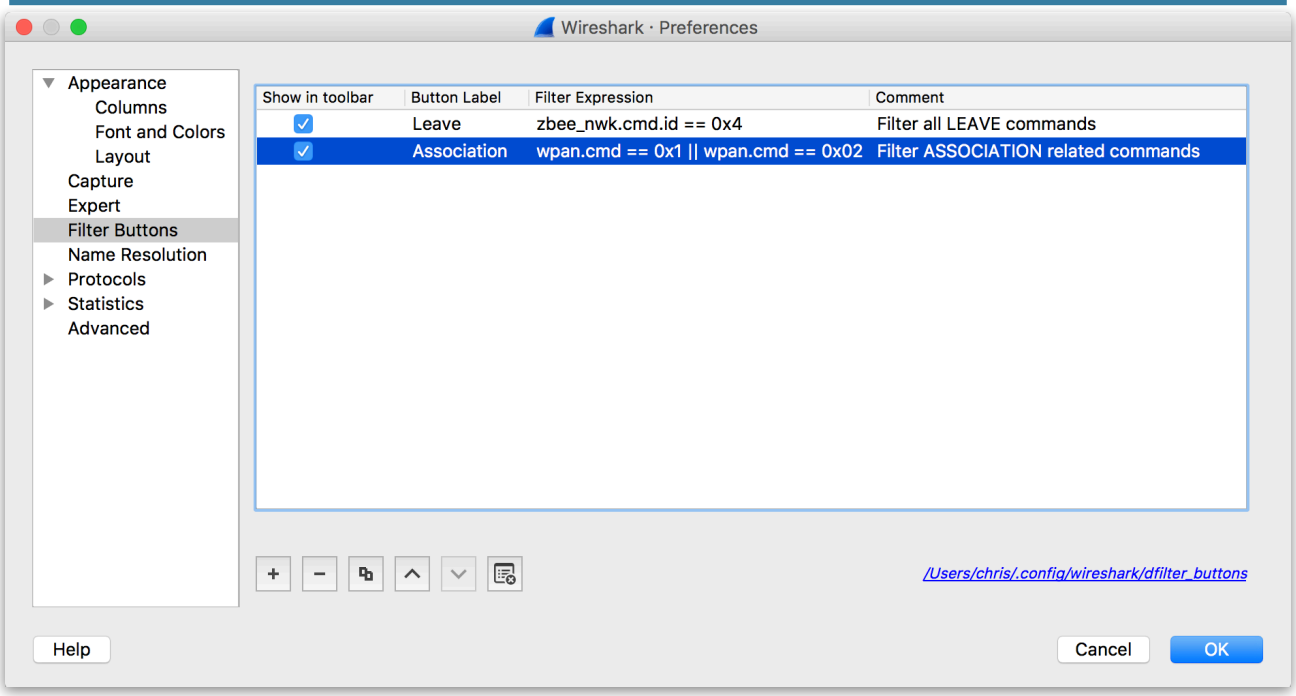
Buttons can be configured to allow quick access to often used filters



The following provides a useful set of filter buttons -:

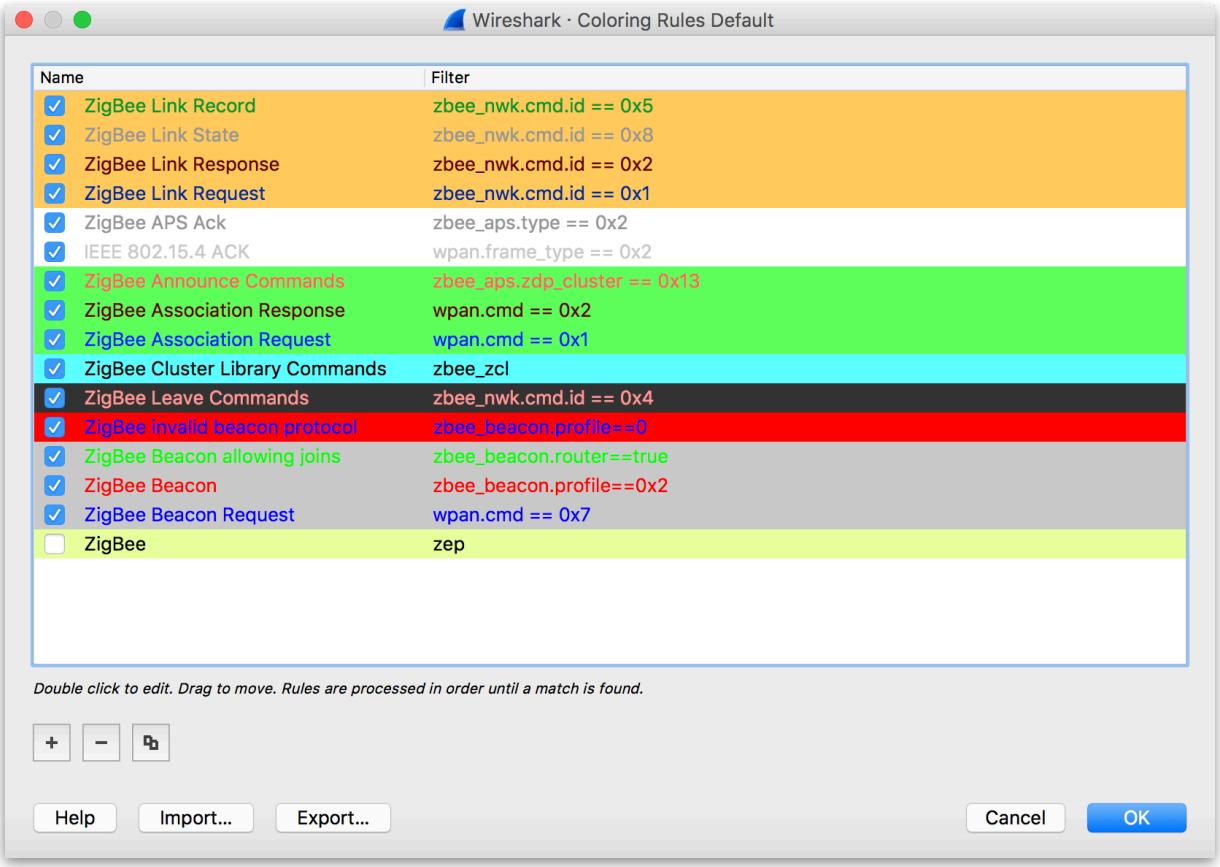
```
# This file is automatically generated, DO NOT MODIFY.
"TRUE","Leave","zbee_nwk.cmd.id == 0x4","Filter all LEAVE commands"
"TRUE","Association","wpan.cmd == 0x1 || wpan.cmd == 0x02","Filter ASSOCIATION related commands"
```

To add these to Wireshark, click the Filter Button Preferences… button above, then click on the link in the bottom right corner of the following window to edit the file. Wireshark must be restarted before the changes will be visible.



Colouring Rules

Wireshark can highlight packets in the packet window based on colouring rules. These use the same rules as the filters and can be configured in the View menu, Colouring Rules… option.

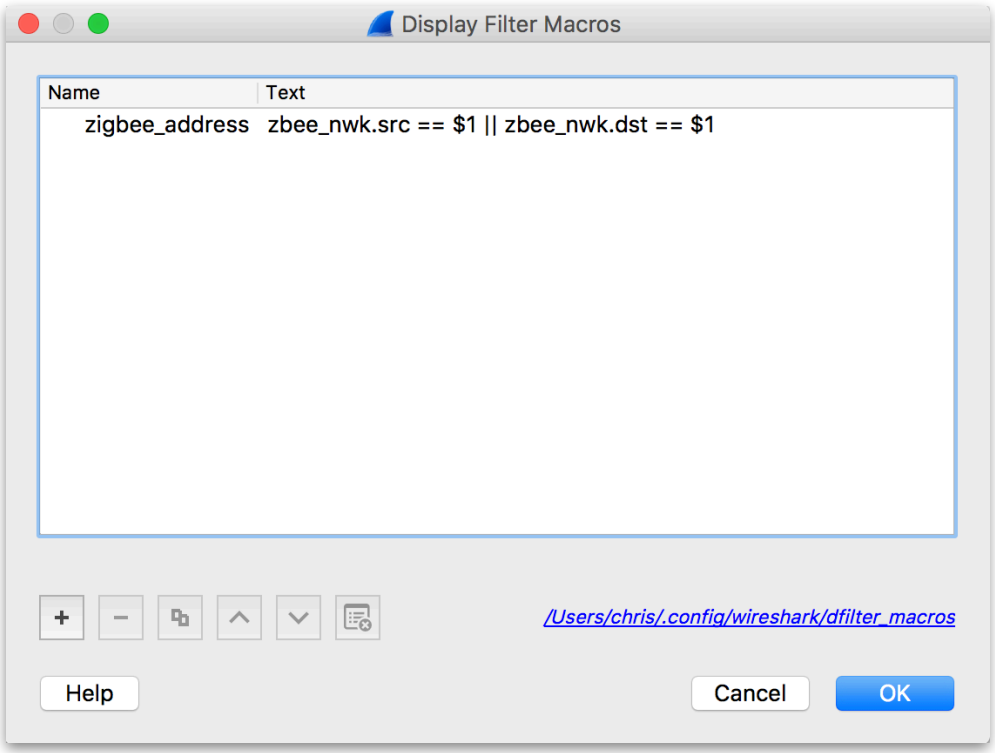


The following provides a useful set of colouring rules that may be imported into Wireshark -:

```
# DO NOT EDIT THIS FILE!  It was created by Wireshark
@ZigBee Link Record@zbee_nwk.cmd.id == 0x5@[65535,52428,26214][0,39321,13107]
@ZigBee Link State@zbee_nwk.cmd.id == 0x8@[65535,52428,26214][39321,39321,39321]
@ZigBee Link Response@zbee_nwk.cmd.id == 0x2@[65535,52428,26214][26214,0,0]
@ZigBee Link Request@zbee_nwk.cmd.id == 0x1@[65535,52428,26214][0,13107,39321]
@ZigBee APS Ack@zbee_aps.type == 0x2@[65535,65535,65535][39321,39321,39321]
@IEEE 802.15.4 ACK@wpan.frame_type == 0x2@[65535,65535,65535][52428,52428,52428]
@ZigBee Announce Commands@zbee_aps.zdp_cluster == 0x13@[26214,65535,26214][65535,26214,26214]
@ZigBee Association Response@wpan.cmd == 0x2@[26214,65535,26214][26214,0,0]
@ZigBee Association Request@wpan.cmd == 0x1@[26214,65535,26214][0,13107,65535]
@ZigBee Cluster Library Commands@zbee_zcl@[26214,65535,65535][0,0,0]
@ZigBee Leave Commands@zbee_nwk.cmd.id == 0x4@[13107,13107,13107][65535,39321,39321]
@ZigBee invalid beacon protocol@zbee_beacon.profile==0@[64764,0,7196][0,0,65535]
@ZigBee Beacon allowing joins@zbee_beacon.router==true@[52428,52428,52428][0,65535,0]
@ZigBee Beacon@zbee_beacon.profile==0x2@[52428,52428,52428][65535,0,0]
@ZigBee Beacon Request@wpan.cmd == 0x7@[52428,52428,52428][0,0,65535]
!@ZigBee@zep@[59624,65535,41377][0,0,0]
```

Display Filter Macros

Display Filter Macros are a mechanism to create shortcuts for complex filters.



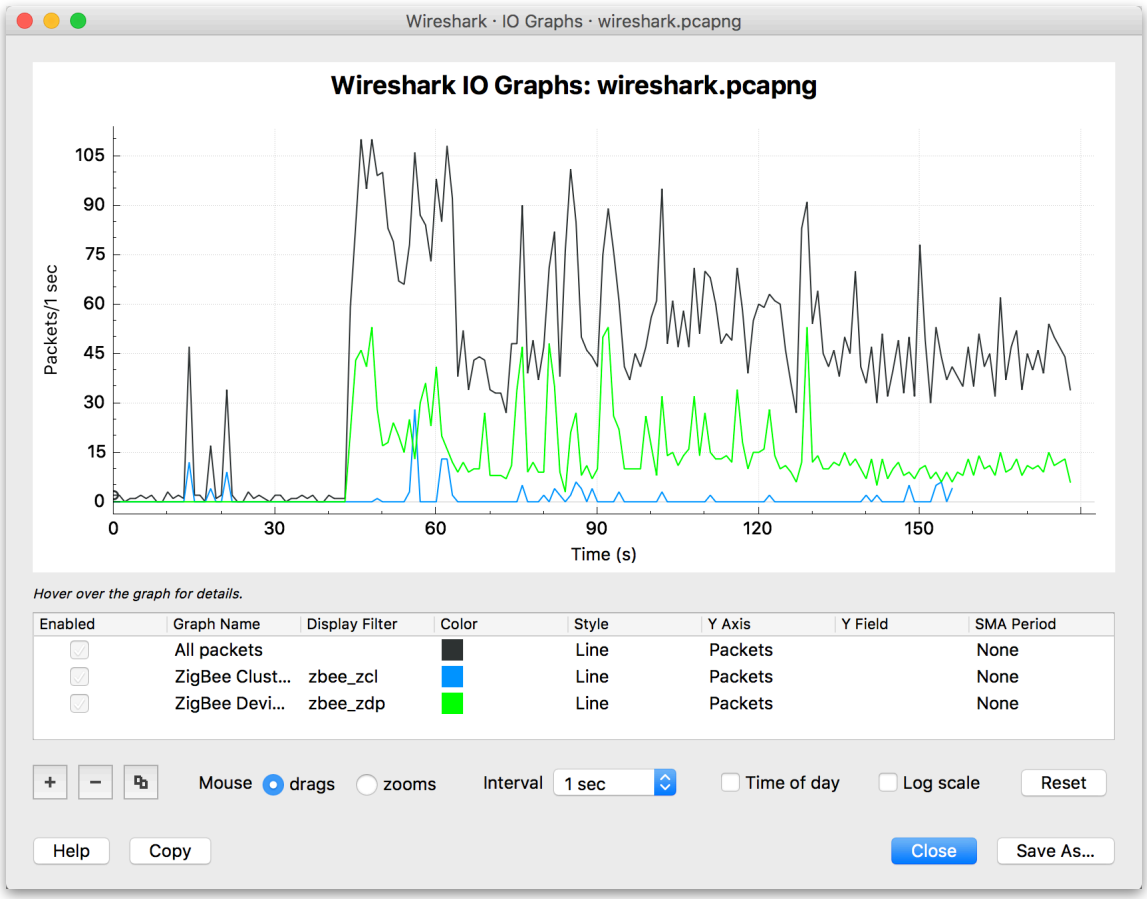
For example defining a display filter macro named *zigbee_address* defined as *zbee_nwk.src == $1 || zbee_nkw.dst == $1* could be used as *${zigbee_address:0x1234}* to display all packets to or from address 0x1234.

Macro	Content
zigbee_address	wpan.dst16 == $1 || wpan.src16 == $1

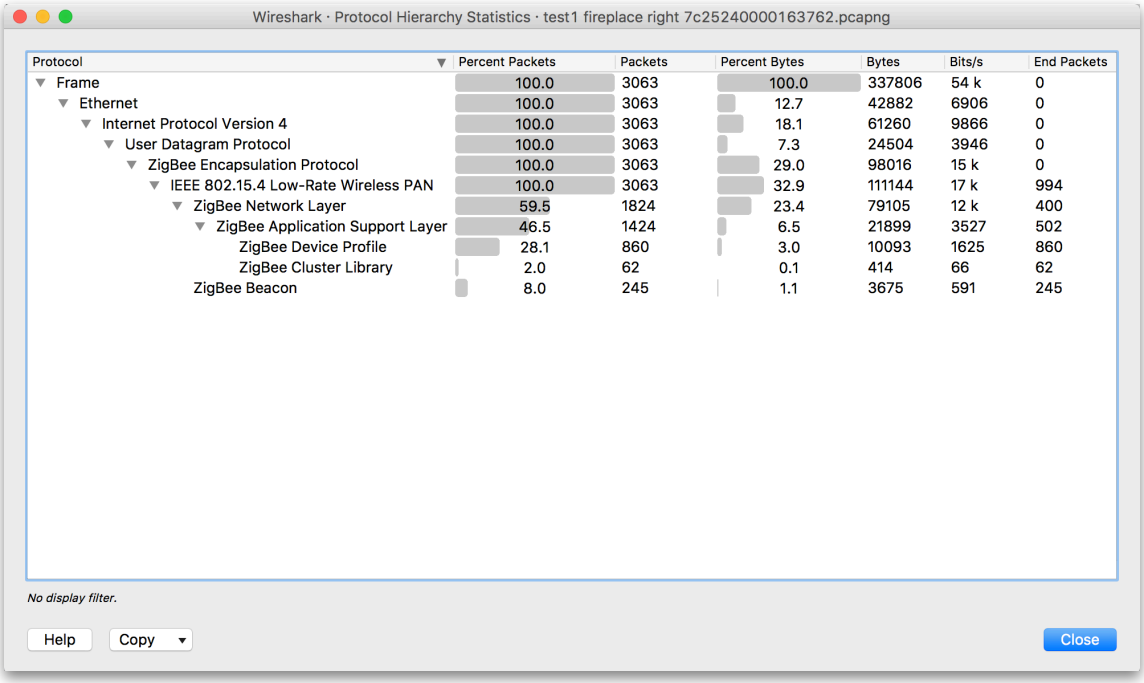
Analysing Data

IO Graph

Wireshark provides a very useful facility to graph events. Any expression can be used to graph the number of events per second.



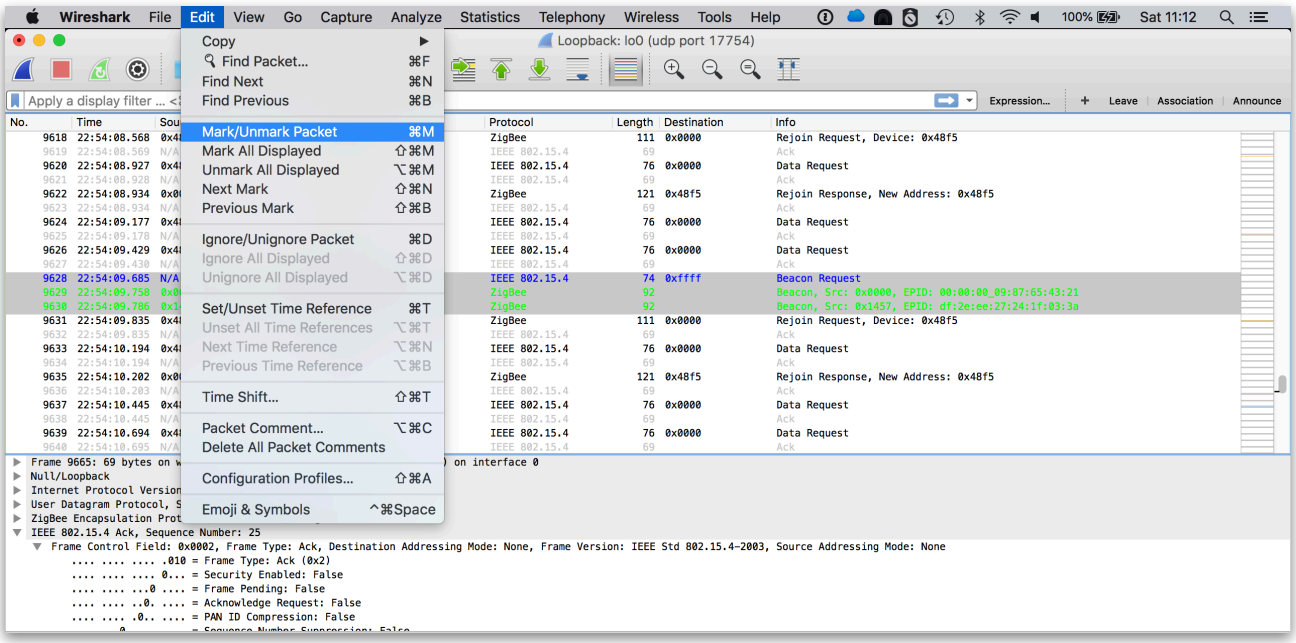
Protocol Statistics



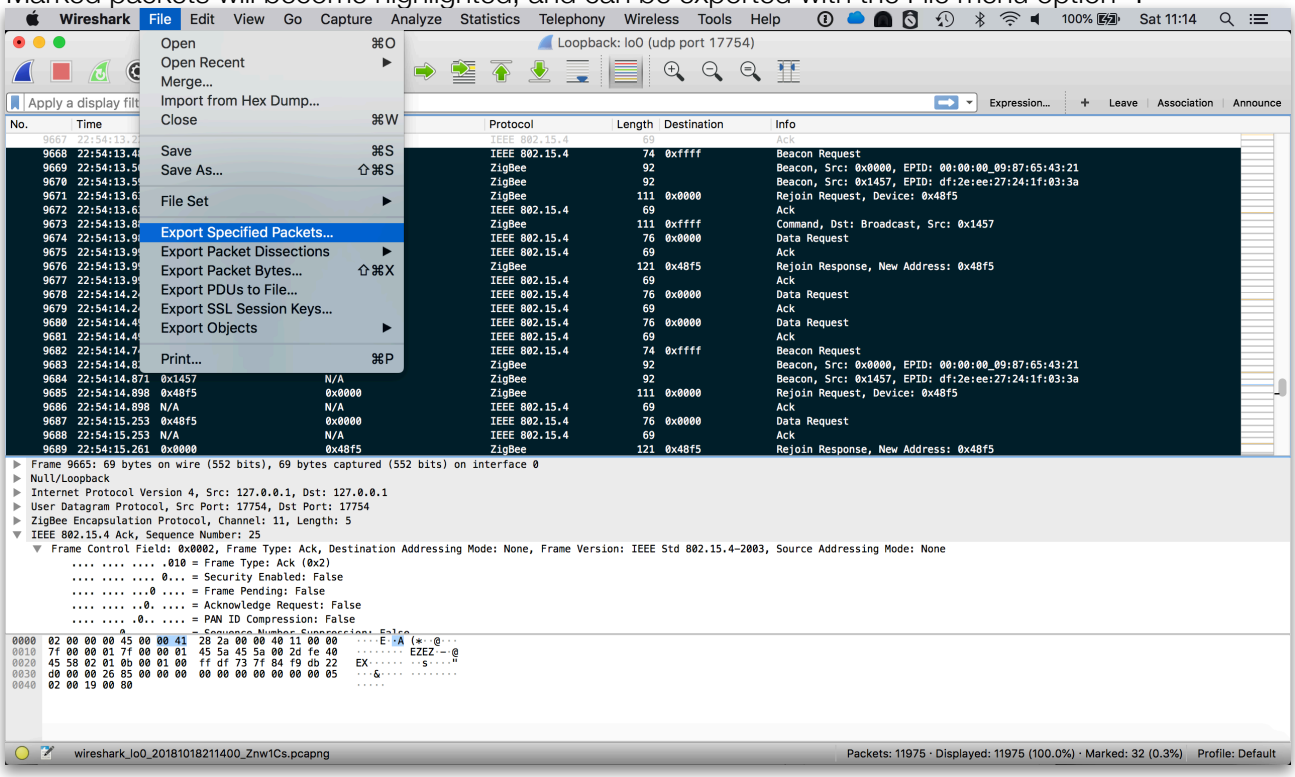
Exporting Selected Packets

It may be necessary to export a select few packets to illustrate a problem you have identified. To do this, you can select the packets, and export them as follows.

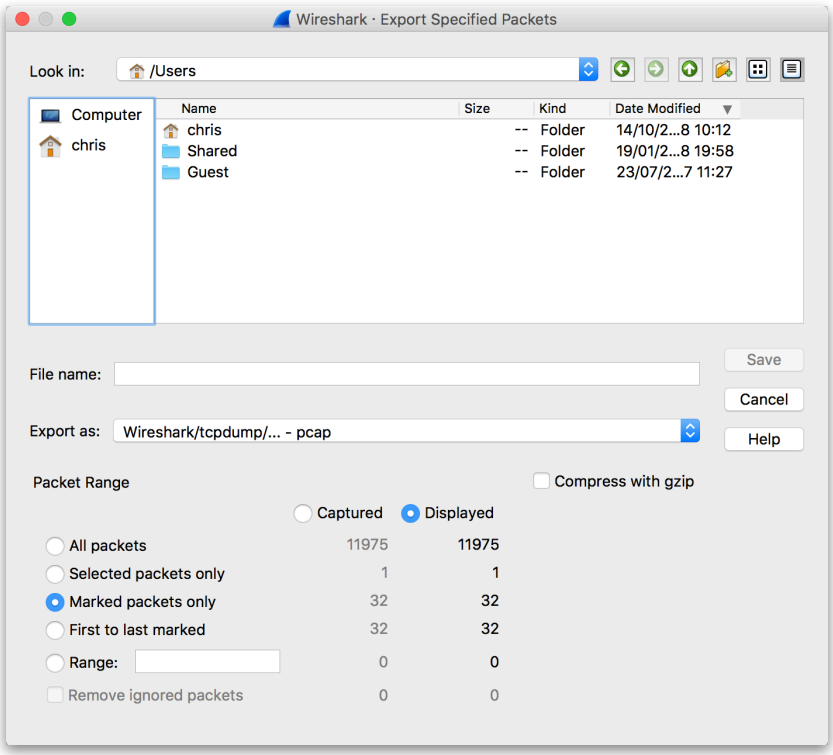
Mark packets using the Edit menu options -:



Marked packets will become highlighted, and can be exported with the File menu option -:



The Export dialog provides an option to select the packet range to be exported - to export just the required packets, select the *Marked packets only* option -:



Note that in order to export the specified packets, live capture must be stopped.