6.3. Filtering packets while viewing

Wireshark has two filtering languages: One used when capturing packets, and one used when displaying packets. In this section we explore that second type of filter: Display filters. The first one has already been dealt with in Section 4.13, “Filtering while capturing”.

Display filters allow you to concentrate on the packets you are interested in while hiding the currently uninteresting ones. They allow you to select packets by:

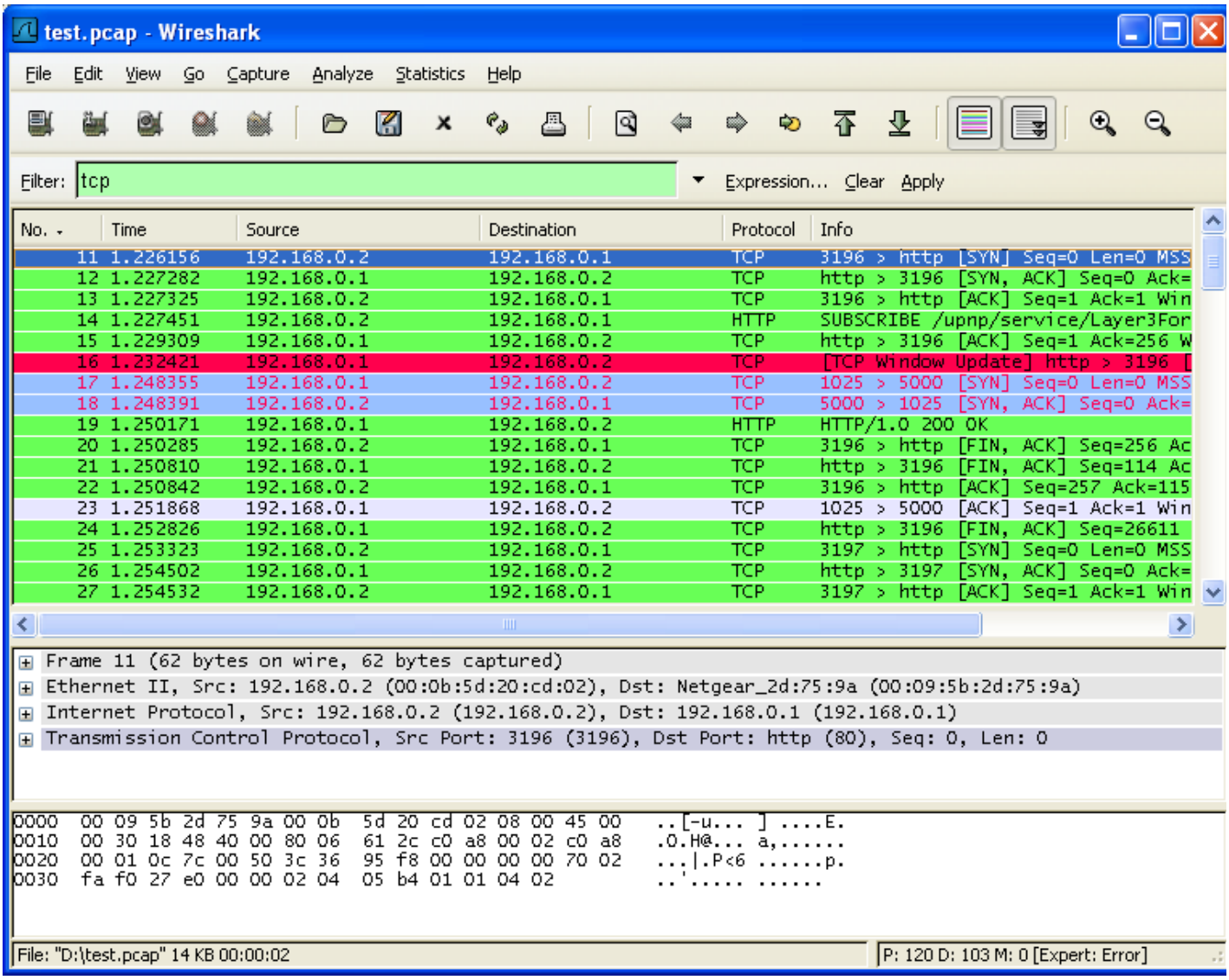
- Protocol
- The presence of a field
- The values of fields
- A comparison between fields
- … and a lot more!

To select packets based on protocol type, simply type the protocol in which you are interested in the *Filter:* field in the filter toolbar of the Wireshark window and press enter to initiate the filter. Figure 6.6, “Filtering on the TCP protocol” shows an example of what happens when you type *tcp* in the filter field.

Note

All protocol and field names are entered in lowercase. Also, don’t forget to press enter after entering the filter expression.

Figure 6.6. Filtering on the TCP protocol



As you might have noticed, only packets of the TCP protocol are displayed now (e.g. packets 1-10 are hidden). The packet numbering will remain as before, so the first packet shown is now packet number 11.

Note

When using a display filter, all packets remain in the capture file. The display filter only changes the display of the capture file but not its content!

You can filter on any protocol that Wireshark understands. You can also filter on any field that a dissector adds to the tree view, but only if the dissector has added an abbreviation for the field. A list of such fields is available in Wireshark in the *Add Expression…* dialog box. You can find more information on the *Add Expression…* dialog box in Section 6.5, “The “Filter Expression” dialog box”.

For example, to narrow the packet list pane down to only those packets to or from the IP address 192.168.0.1, use `ip.addr==192.168.0.1`.

Note

To remove the filter, click on the **Clear** button to the right of the filter field.

Some examples:

FTP: File Transfer Protocol.
https://en.wikipedia.org/wiki/File_Transfer_Protocol

HTTP: The Hypertext Transfer Protocol.
https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

For FTP data packets: (To find the the packets uploaded/downloaded)

```
tcp.port eq 20 or ftp-data
```

For FTP command packets: (To find the username and password)

```
tcp.port eq 21 or ftp
```

For HTTP packets:

```
tcp.port eq 80 or http
```

To filter packets from specific IP:

```
ip.addr == 10.43.54.65
```

Wireshark will only display packets matching with the applied filter. For example, using **ftp-data** will display all TCP packets that were exchanged for FTP communication.

Since it takes multiple packets to transfer a typical file (> ~1460 Bytes in size), you would need to recreate the transferred file by extracting the data payload from all the packets involved in the given FTP or HTTP communication. It can be achieved by using “Follow TCP Stream” feature of Wireshark.

7.2. Following TCP streams

If you are working with TCP based protocols it can be very helpful to see the data from a TCP stream in the way that the application layer sees it. Perhaps you are looking for passwords in a Telnet stream, or you are trying to make sense of a data stream. Maybe you just need a display filter to show only the packets of that TCP stream. If so, Wireshark's ability to follow a TCP stream will be useful to you.

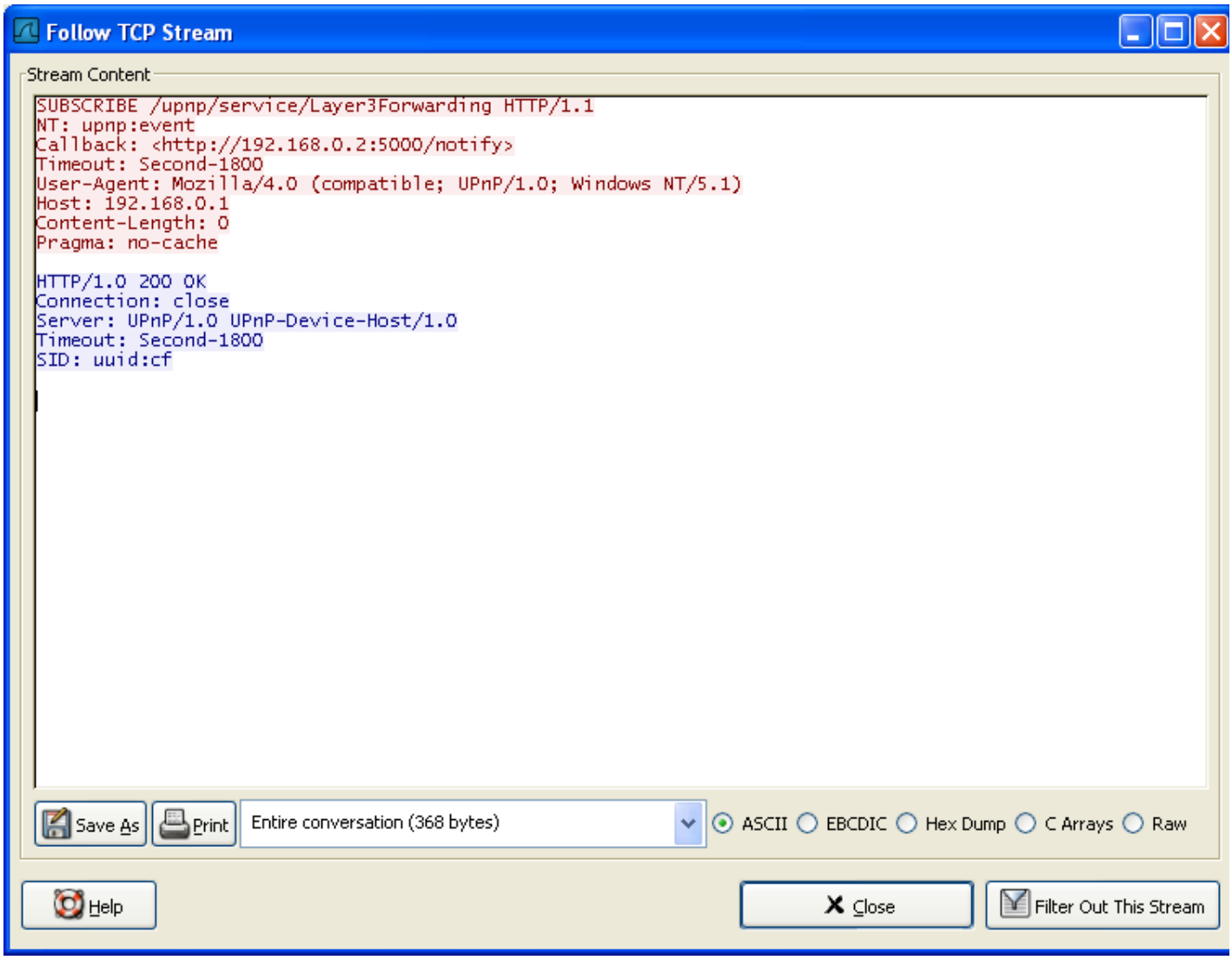
Simply select a TCP packet in the packet list of the stream/connection you are interested in and then select the Follow TCP Stream menu item from the Wireshark Tools menu (or use the context menu in the packet list). Wireshark will set an appropriate display filter and pop up a dialog box with all the data from the TCP stream laid out in order, as shown in Figure 7.1, "The "Follow TCP Stream" dialog box".

Note

Opening the "Follow TCP Stream" installs a display filter to select all the packets in the TCP stream you have selected.

7.2.1. The "Follow TCP Stream" dialog box

Figure 7.1. The "Follow TCP Stream" dialog box



The stream content is displayed in the same sequence as it appeared on the network. Traffic from A to B is marked in red, while traffic from B to A is marked in blue. If you like, you can change these colors in the

"Colors" page if the "Preferences" dialog.

Non-printable characters will be replaced by dots.

The stream content won't be updated while doing a live capture. To get the latest content you'll have to reopen the dialog.

You can choose from the following actions:

1. *Save As*: Save the stream data in the currently selected format.
2. *Print*: Print the stream data in the currently selected format.
3. *Direction*: Choose the stream direction to be displayed ("Entire conversation", "data from A to B only" or "data from B to A only").
4. *Filter out this stream*: Apply a display filter removing the current TCP stream data from the display.
5. *Close*: Close this dialog box, leaving the current display filter in effect.

You can choose to view the data in one of the following formats:

1. *ASCII*: In this view you see the data from each direction in ASCII. Obviously best for ASCII based protocols, e.g. HTTP.
2. *EBCDIC*: For the big-iron freaks out there.
3. *HEX Dump*: This allows you to see all the data. This will require a lot of screen space and is best used with binary protocols.
4. *C Arrays*: This allows you to import the stream data into your own C program.
5. *Raw*: This allows you to load the unaltered stream data into a different program for further examination. The display will look the same as the ASCII setting, but "Save As" will result in a binary file.