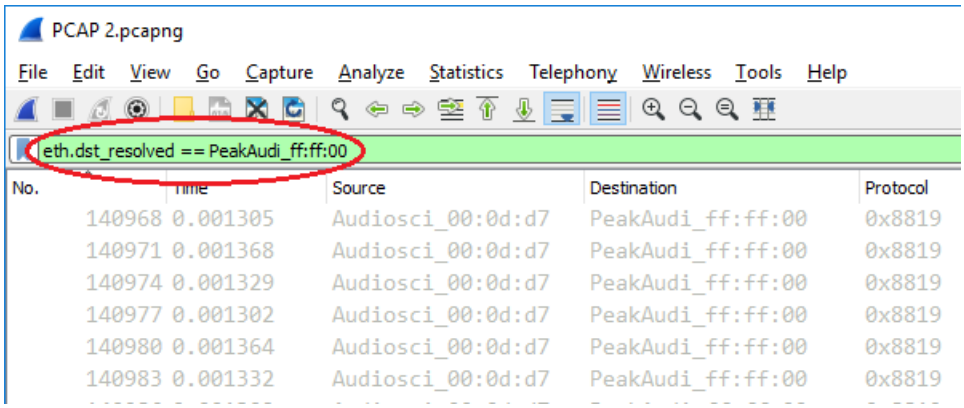
Sign of a Healthy Network

How to Analyze/Determine in Wireshark

The sign of a healthy network (i.e., "healthy" for CobraNet digital audio) is a regular and reliable CobraNet Conductor Beat in Wireshark captures. This check can be easily done in Wireshark as described in this document. First, capture some packets, from one to five minutes worth. (More can be done, but will be slower to process in Wireshark.)

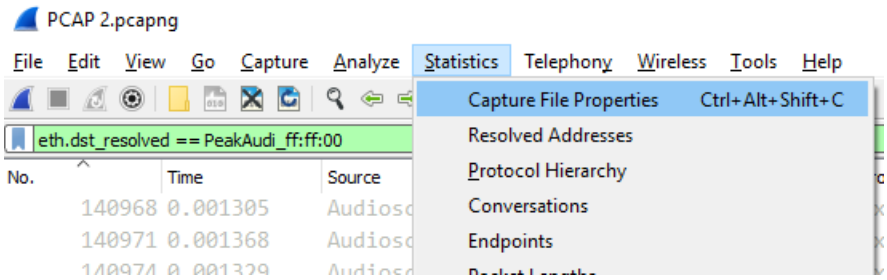
Step 1: Filter on Beat Packets

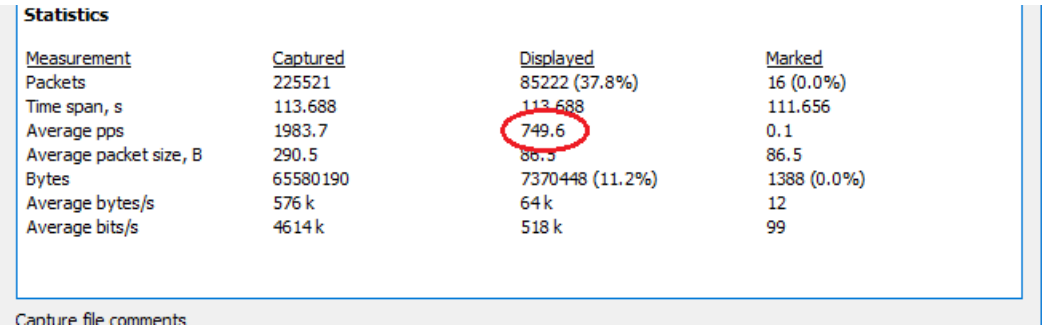
Set the Wireshark display filter to "eth.dst_resolved == PeakAudi_ff:ff:00" (without the quotes). This should yield a display with only Beat packets, such as shown below.



Step 2: Look at Statistics

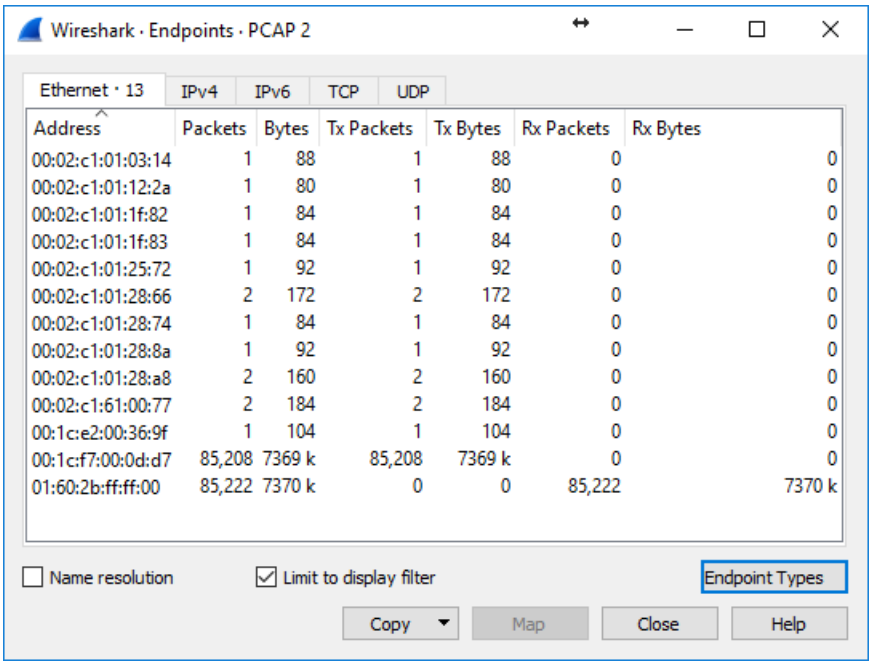
Go to the **Statistics** Menu, **Capture File Properties** option. This brings up a window that shows many things. You are interested in the **Statistics** section near the bottom of the Details pane, such as the example shown below. The important property is circled. This should be 750.0, exactly.





Step 3: Look for Other Devices Trying to Be Conductor

The primary conductor is the source device one sees on the main Wireshark page (e.g., AudioSci_00:0d:d7 in the example above). One way to find whether there are other devices trying to be CobraNet Conductor is to go to the **Statistics** menu, **Endpoints** option. After this page fully loaded, check the **Limit to Display Filter** checkbox at the bottom. What you end up with **should** be only two lines: the primary conductor and the destination address of PeakAudi_ff:ff:00. If you see multiple additional lines such as shown below, at least some portions of the network is unhealthy for CobraNet.



Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
00:02:c1:01:03:14	1	88	1	88	0	0
00:02:c1:01:12:2a	1	80	1	80	0	0
00:02:c1:01:1f:82	1	84	1	84	0	0
00:02:c1:01:1f:83	1	84	1	84	0	0
00:02:c1:01:25:72	1	92	1	92	0	0
00:02:c1:01:28:66	2	172	2	172	0	0
00:02:c1:01:28:74	1	84	1	84	0	0
00:02:c1:01:28:8a	1	92	1	92	0	0
00:02:c1:01:28:a8	2	160	2	160	0	0
00:02:c1:61:00:77	2	184	2	184	0	0
00:1c:e2:00:36:9f	1	104	1	104	0	0
00:1c:f7:00:0d:d7	85,208	7369 k	85,208	7369 k	0	0
01:60:2b:ff:ff:00	85,222	7370 k	0	0	85,222	7370 k

Note: It is possible to see problems in this last step even though things appear OK in the previous step. That scenario would mean that possibly only some portion of the network is not healthy. There are several ways this can happen:

- There might be a situation where some endpoints, such as mic stations, are on a cheap, unmanaged switch somewhere. This switch is not reliably passing the primary conductor beat packets to the endpoints, so they periodically try to take over.
- Some portion is linked via a switch-to-switch link that has bottlenecks. A couple of ways this can happen includes:
  - Contention with other network traffic on the link (even when on separate VLANs) such as closed-circuit TV cameras or VoIP phones.
  - It could also be induced by network policy that puts a broadcast bandwidth limit on ports. Much of the CobraNet traffic can appear to be broadcasts to the switches and so they throttle this traffic down, causing dropouts in the Beat packets reaching endpoints on that portion of the network.

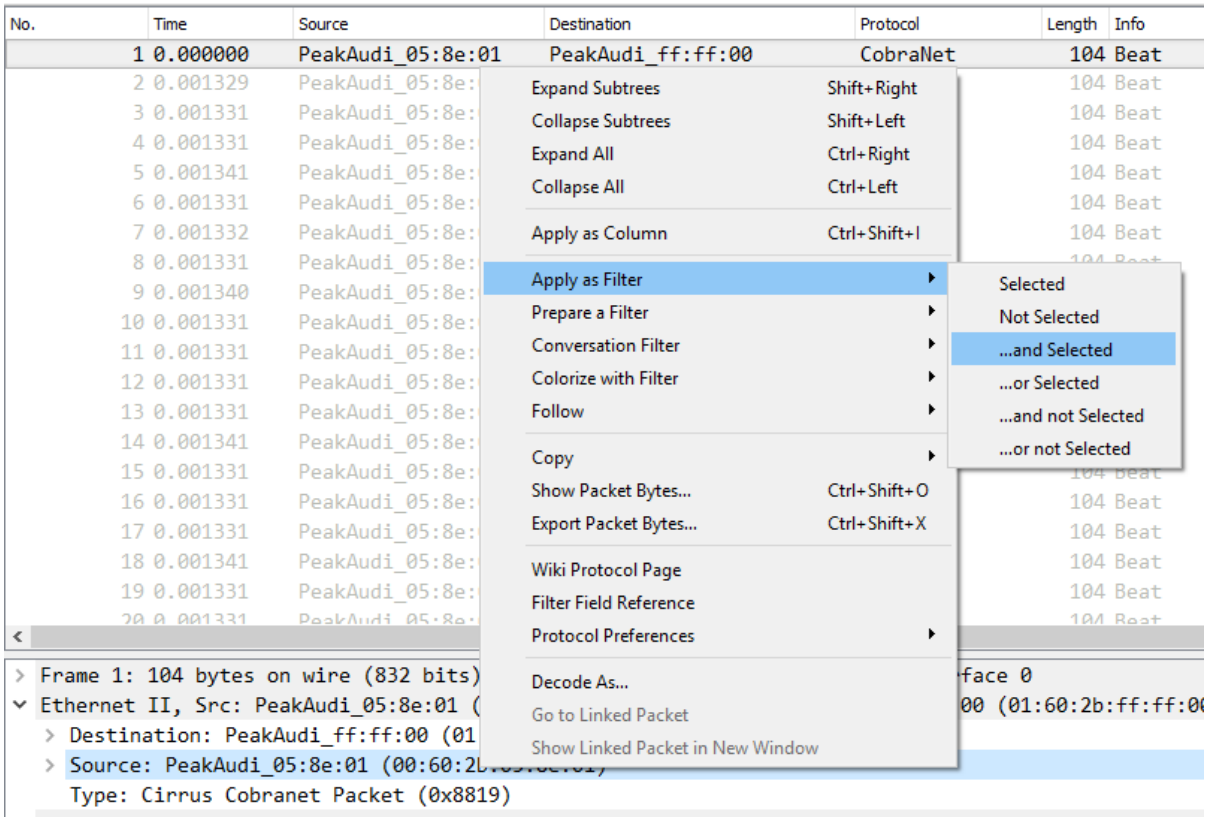
Step 4: Look for Beat Packet Delay Variations

WARNING

When doing this check, one must be sure that packets were captured cleanly. An overburdened computer, perhaps remotely logged into via a screensharing program like remote desktop, that is also doing other functions (like running the public address system software) may compete with Wireshark, such that the packet captures are not done cleanly. When this is the case, apparent delay variation is introduced into the capture file that is NOT REAL, but is only an artifact of the collection process. It is imperative that the packet capturer carefully check for cleanness of the capture, such as checking Windows Task Manager beforehand to insure the collection device is lightly loaded, or using a dedicated packet capture device.

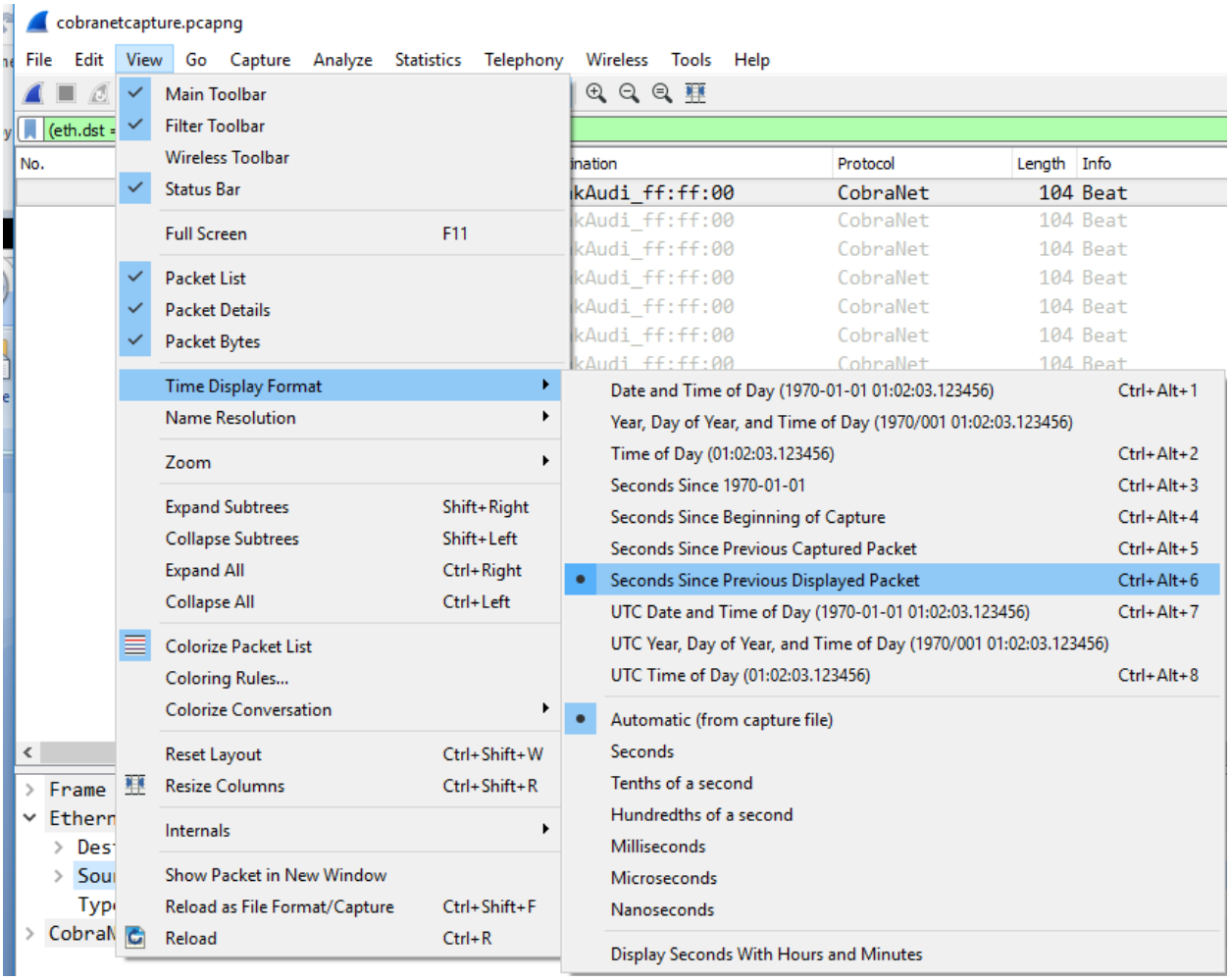
One requirement CobraNet places on the network is that the clocking or Beat packets must arrive everywhere on the network with a minimum variation in arrival time. CobraNet calls it the Beat Packet Delay Variation, and the requirement* is that it be no more than ±0.250 msec. One way to check on this is to first filter on Beat Packets only, as in Step 1.

If there is more than one Conductor in the capture file, also filter on the primary conductor. This may be done by selecting a packet, expanding the Ethernet II section in the decode window, clicking on the Source field, and then right-clicking and selecting **Apply as Filter**, and **…and Selected** in the pop-up menus. An example of this is shown below.

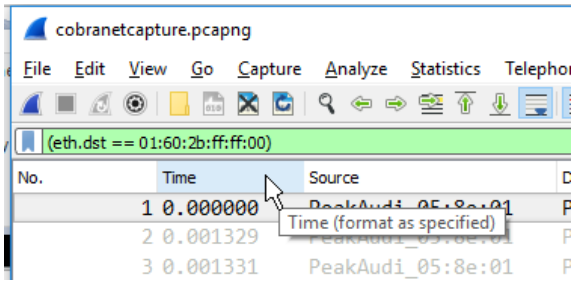


* See: http://www.cobranet.info/support/design/performance

Next, go the **View** menu, select **Time Display Format** and the **Seconds Since Previously Displayed Packet** option. Now, the listing should show packet times as values around 0.001333 seconds.



If one clicks on the **Time** column header, Wireshark will now sort the packets by time, shortest (0.00000) at the top and longer times at the bottom. (Of course, there will always be one packet of time 0.0000, the first in the capture file.)



It is usually most helpful to scroll to the bottom and look at the longest delays in the capture file. These should be no more than 0.001583 seconds. Optionally, you can highlight any too long values, click back on the Packet **No.** column heading to re-sort back into capture order to see what was going on around the time of this long delay.