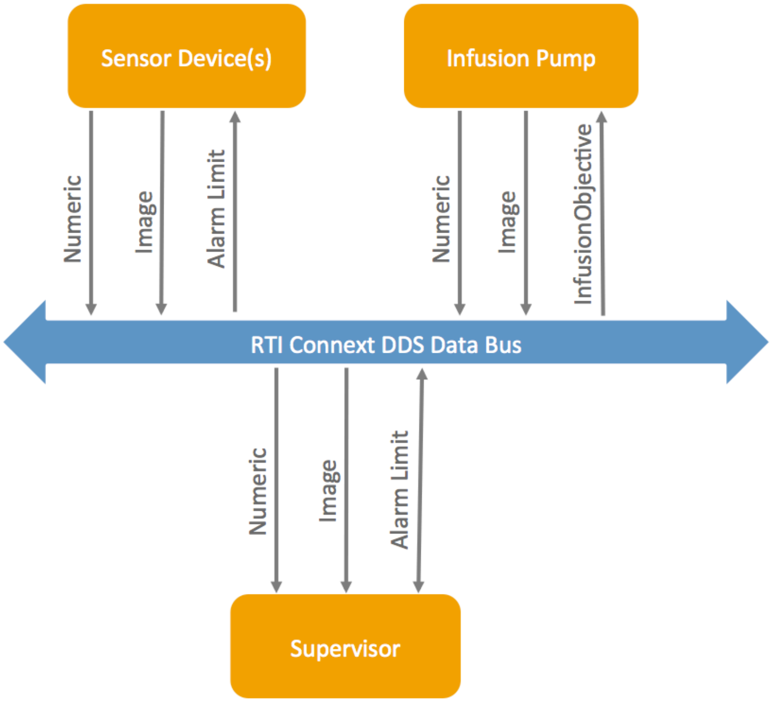
Integrated Clinical Environment (ICE) Overview:

The ICE system contains simulated devices that monitor patient data, a simulated infusion pump that monitors a drug to a patient, and a supervisor app that displays patient vitals, alarms, and is used to control the valid ranges for the system. For example, the supervisor app can decide what a valid pulse rate is for a patient. The supervisor app and the device apps can both show alarms in certain circumstances.



What's in the system?

Simulated devices:

- Simulated devices send their device ID, an image representing the device, and numeric data such as patient vital signs. Simulated devices receive alarm limits that tell them the ranges of vital sign values that should produce an alarm (in the case where the device can display an alarm).

Devices send and receive much more than this, but for simplicity we will describe only some of the important values.

Infusion pump:

- The infusion pump produces and consumes a lot of the same data as the other devices, but it also needs to be able to receive a command telling it to stop infusing immediately.

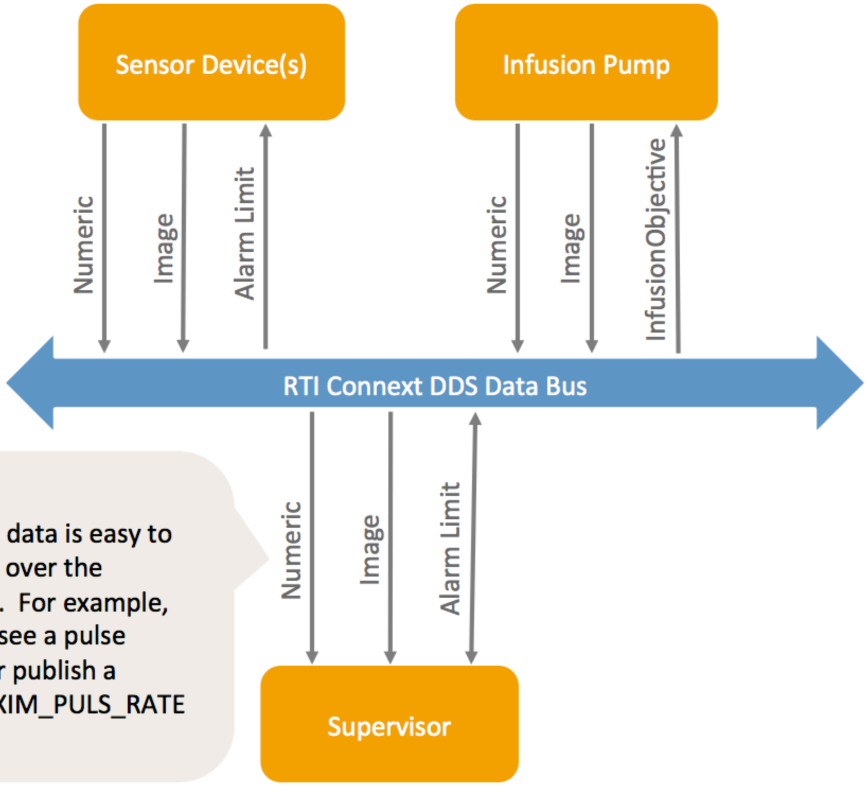
Supervisor:

- The supervisor receives the device IDs, images, and vital signs from the devices. It receives the status of the infusion pump.
- The supervisor sends and receives alarm limits, which are used to define what the valid ranges are for each vital sign. Both the supervisor and some devices use these values to decide whether to display an alarm. Or not. This is what we will compromise.

Exercise 1:

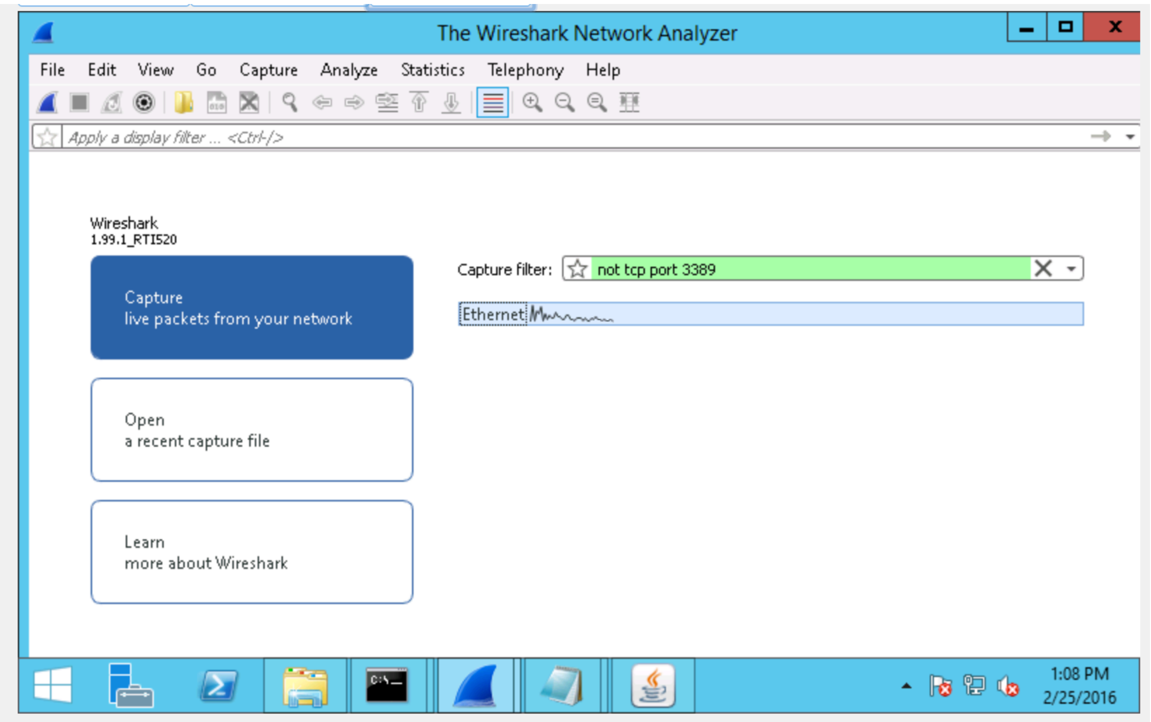
Overview:

The ICE demo allows you to create simulated devices, and to view them and control their behavior. In this exercise, you will run the ICE demo with no security enabled at all. The goal of this exercise is to understand the data in the ICE system, to visualize it over the network, and to understand what a “sniffer” app (legitimate or otherwise) can see.

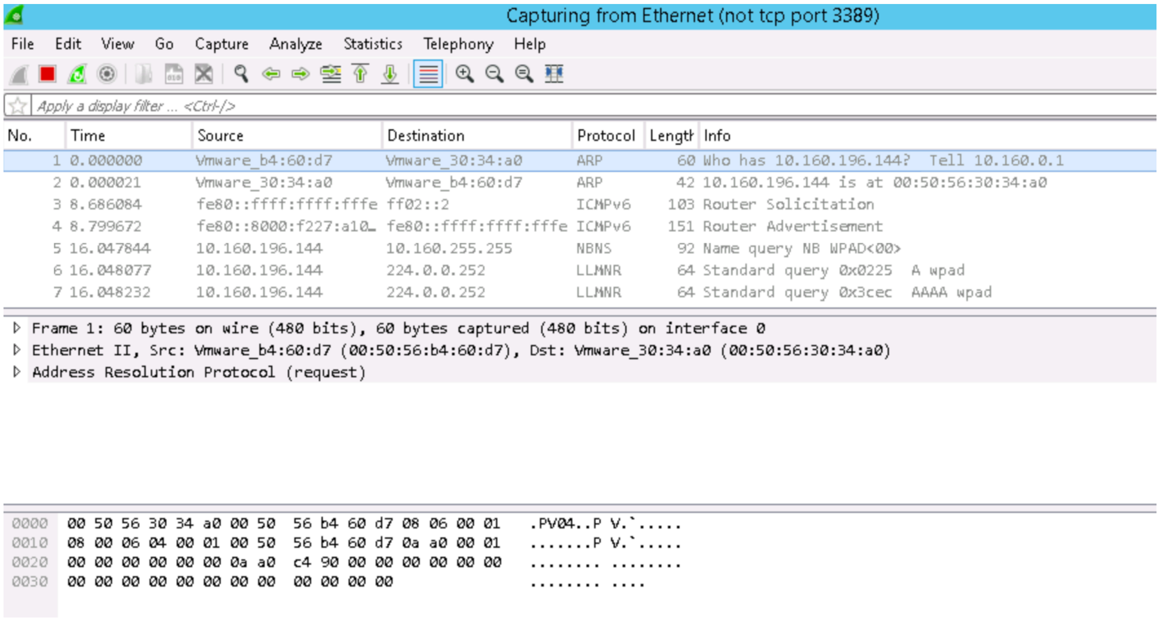


Exercise:

1. Open Wireshark, and start capturing traffic. Double-click on the “Ethernet” button and you will start to see packets. (If you don’t see any traffic right away, you will see some when you start the ICE application.



Capture filter: not tcp port 3389

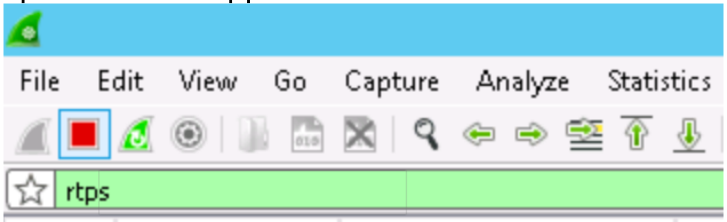

Capturing from Ethernet (not tcp port 3389)
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Help
Apply a display filter ... <Ctrl-/>

```
No.  Time       Source                 Destination            Protocol Length Info
   1 0.000000   Vmware_b4:60:d7        Vmware_30:34:a0        ARP          60 Who has 10.160.196.144?  Tell 10.160.0.1
   2 0.000021   Vmware_30:34:a0        Vmware_b4:60:d7        ARP          42 10.160.196.144 is at 00:50:56:30:34:a0
   3 8.686084   fe80::ffff:ffff:fffe   ff02::2                ICMPv6      103 Router Solicitation
   4 8.799672   fe80::8000:f227:a10…   fe80::ffff:ffff:fffe   ICMPv6      151 Router Advertisement
   5 16.047844  10.160.196.144         10.160.255.255         NBNS         92 Name query NB WPAD<00>
   6 16.048077  10.160.196.144         224.0.0.252            LLMNR        64 Standard query 0x0225  A wpad
   7 16.048232  10.160.196.144         224.0.0.252            LLMNR        64 Standard query 0x3cec  AAAA wpad
```

```
▷ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▷ Ethernet II, Src: Vmware_b4:60:d7 (00:50:56:b4:60:d7), Dst: Vmware_30:34:a0 (00:50:56:30:34:a0)
▷ Address Resolution Protocol (request)
```

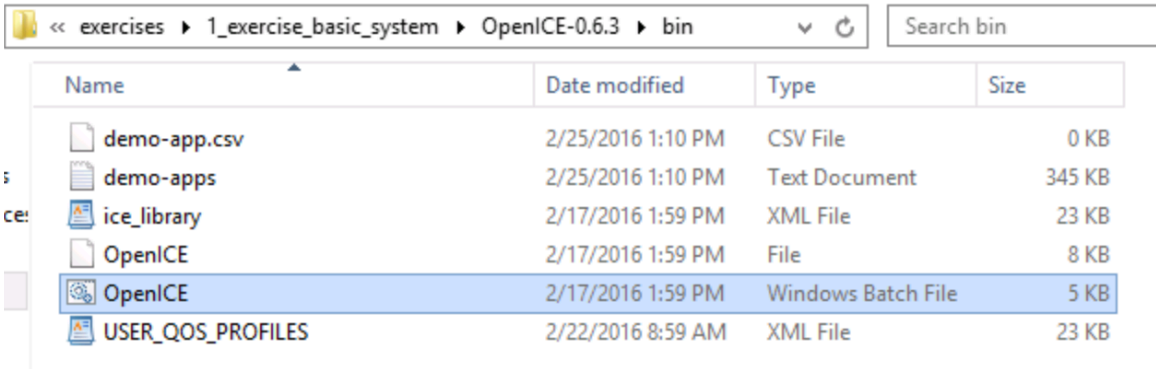
```
0000  00 50 56 30 34 a0 00 50  56 b4 60 d7 08 06 00 01   .PV04..P V.`.....
0010  08 00 06 04 00 01 00 50  56 b4 60 d7 0a a0 00 01   .......P V.`.....
0020  00 00 00 00 00 00 0a a0  c4 90 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

You may want to filter for RTPS traffic if you would like to see only the data specific to our application:



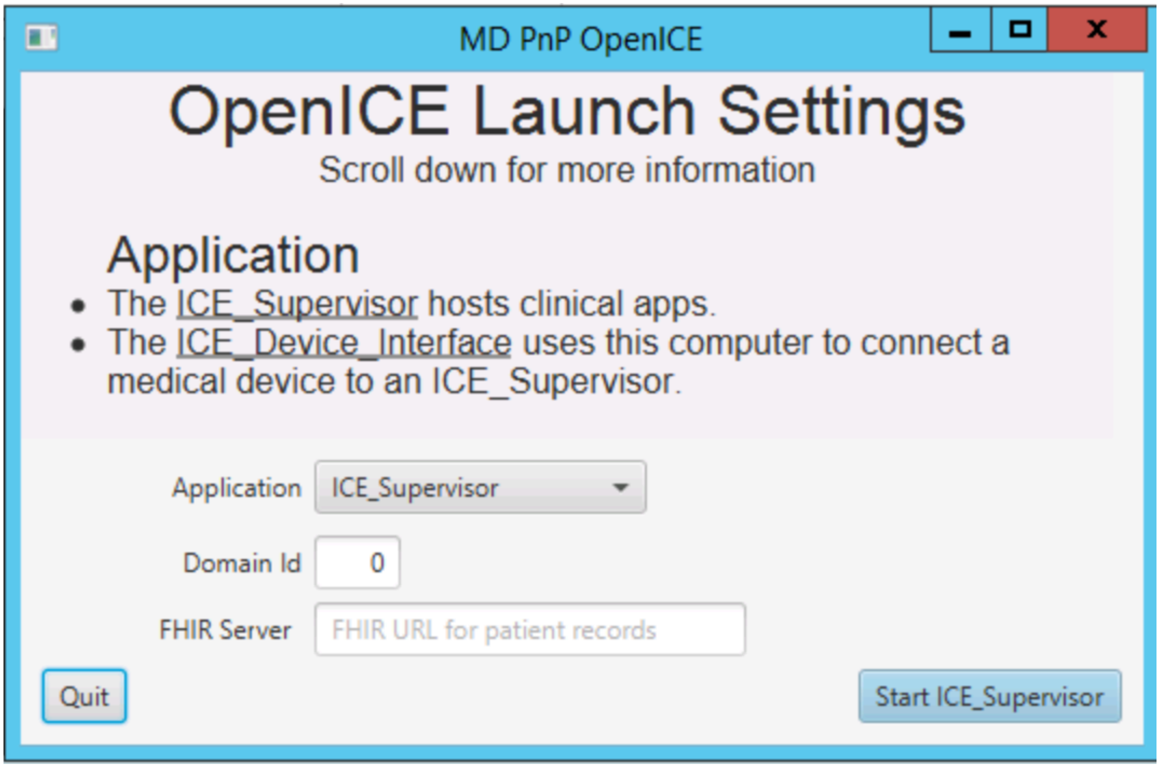
rtps

2. Go to the OpenICE-0.6.3\bin. Run the OpenICE.bat file.
(**Note:** for the purposes of these exercises, you must always run the open ICE project from within the <exercise>\OpenICE-0.6.3\bin directory, either by double-clicking it from the windows GUI or changing into that directory at the command-prompt.)



Name	Date modified	Type	Size
demo-app.csv	2/25/2016 1:10 PM	CSV File	0 KB
demo-apps	2/25/2016 1:10 PM	Text Document	345 KB
ice_library	2/17/2016 1:59 PM	XML File	23 KB
OpenICE	2/17/2016 1:59 PM	File	8 KB
OpenICE	2/17/2016 1:59 PM	Windows Batch File	5 KB
USER_QOS_PROFILES	2/22/2016 8:59 AM	XML File	23 KB

3. Keep the application in domain 0 (the default)
4. Click on "Start ICE_Supervisor"



OpenICE Launch Settings

Scroll down for more information

Application

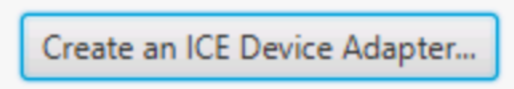
- The ICE_Supervisor hosts clinical apps.
- The ICE_Device_Interface uses this computer to connect a medical device to an ICE_Supervisor.

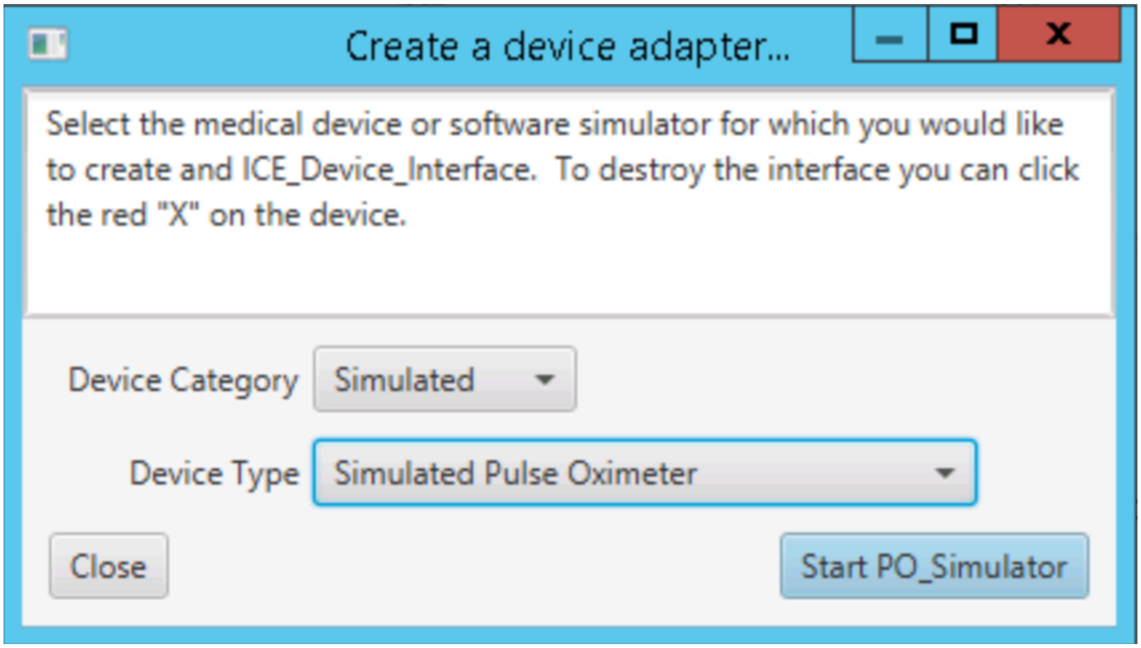
Application: ICE_Supervisor

Domain Id: 0

FHIR Server: FHIR URL for patient records

5. Once the full application comes up, click on "Create an ICE Device Adapter" in the lower right hand corner and create a simulated pulse oximeter. (This has easy data to view).





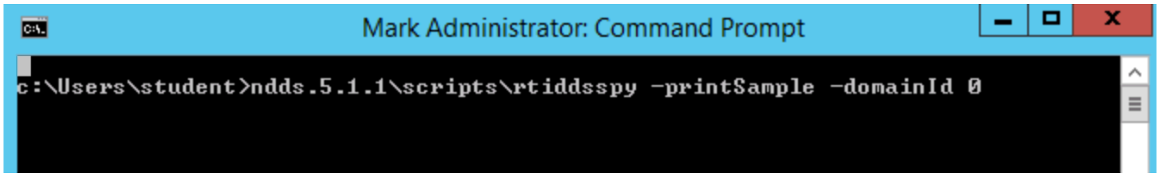
Select the medical device or software simulator for which you would like to create and ICE_Device_Interface. To destroy the interface you can click the red "X" on the device.

Device Category: Simulated

Device Type: Simulated Pulse Oximeter

6. Run the c:\Users\student\ndds.5.1.1\scripts\nddsspy tool on another machine,

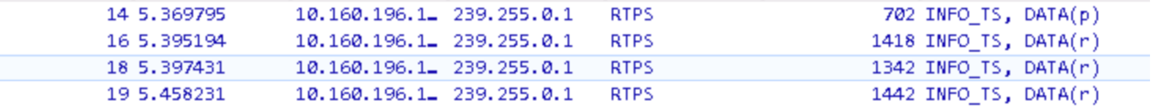
with the option -printSample. This tool is a debugging / simple visualization tool. Without encryption, you can visualize all the data in the system. By running rtiddsspy on a different machine, you will be able to see all the traffic in Wireshark.



```
c:\Users\student>ndds.5.1.1\scripts\rtiddsspy -printSample -domainId 0
```

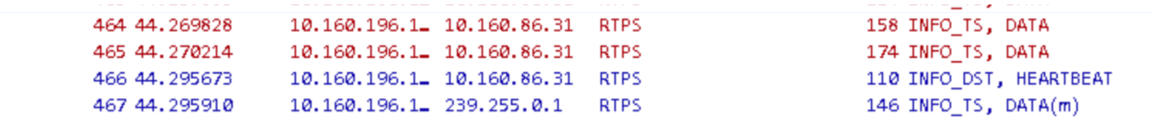
7. View the RTPS traffic using Wireshark. This traffic includes discovery data that describes the data being written and read by each application.

What you should see:
Packets that are marked in blue in Wireshark are discovery data, and they show the metadata about what is being done in the system.



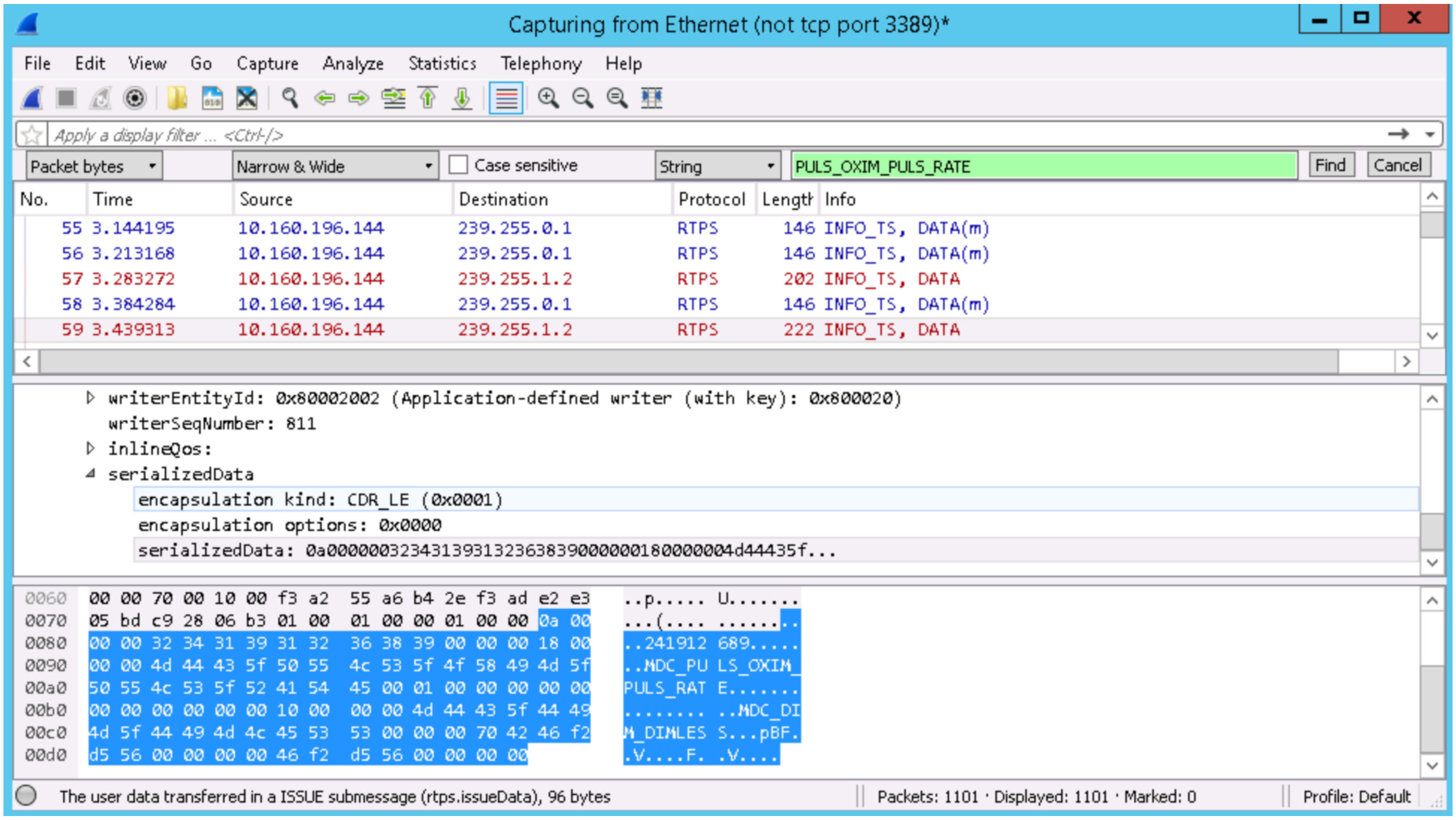
```
     14 5.369795       10.160.196.1…  239.255.0.1    RTPS                          702 INFO_TS, DATA(p)
     16 5.395194       10.160.196.1…  239.255.0.1    RTPS                         1418 INFO_TS, DATA(r)
     18 5.397431       10.160.196.1…  239.255.0.1    RTPS                         1342 INFO_TS, DATA(r)
     19 5.458231       10.160.196.1…  239.255.0.1    RTPS                         1442 INFO_TS, DATA(r)
```

Packets that are marked in red are the runtime data of the system.



```
    464 44.269828      10.160.196.1…  10.160.86.31   RTPS                          158 INFO_TS, DATA
    465 44.270214      10.160.196.1…  10.160.86.31   RTPS                          174 INFO_TS, DATA
    466 44.295673      10.160.196.1…  10.160.86.31   RTPS                          110 INFO_DST, HEARTBEAT
    467 44.295910      10.160.196.1…  239.255.0.1    RTPS                          146 INFO_TS, DATA(m)
```

If you have created a pulse oximeter, some of its data contains the string MDC_PULS_OXIM_PULS_RATE (describing the pulse rate data), and the string MDC_DIM_DIMLESS (saying this data does not have dimensions). You can search for one of these values in Wireshark to see the data as it is being sent.



Capturing from Ethernet (not tcp port 3389)*
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Help
Apply a display filter … <Ctrl-/>
Find
Cancel

Packet bytes · Narrow & Wide · □ Case sensitive · String · PULS_OXIM_PULS_RATE

```
No.  Time      Source          Destination    Protocol  Length Info
  55 3.144195  10.160.196.144  239.255.0.1    RTPS         146 INFO_TS, DATA(m)
  56 3.213168  10.160.196.144  239.255.0.1    RTPS         146 INFO_TS, DATA(m)
  57 3.283272  10.160.196.144  239.255.1.2    RTPS         202 INFO_TS, DATA
  58 3.384284  10.160.196.144  239.255.0.1    RTPS         146 INFO_TS, DATA(m)
  59 3.439313  10.160.196.144  239.255.1.2    RTPS         222 INFO_TS, DATA
```

```
  ▷ writerEntityId: 0x80002002 (Application-defined writer (with key): 0x800020)
    writerSeqNumber: 811
  ▷ inlineQos:
  ◢ serializedData
      encapsulation kind: CDR_LE (0x0001)
      encapsulation options: 0x0000
      serializedData: 0a000000323431393132363839000000180000004d44435f...
```

```
0060  00 00 70 00 10 00 f3 a2  55 a6 b4 2e f3 ad e2 e3   ..p..... U.......
0070  05 bd c9 28 06 b3 01 00  01 00 00 01 00 00 0a 00   ...(.... ........
0080  00 00 32 34 31 39 31 32  36 38 39 00 00 00 18 00   ..241912 689.....
0090  00 00 4d 44 43 5f 50 55  4c 53 5f 4f 58 49 4d 5f   ..MDC_PU LS_OXIM_
00a0  50 55 4c 53 5f 52 41 54  45 00 01 00 00 00 00 00   PULS_RAT E.......
00b0  00 00 00 00 00 00 10 00  00 00 4d 44 43 5f 44 49   ........ ..MDC_DI
00c0  4d 5f 44 49 4d 4c 45 53  53 00 00 00 70 42 46 f2   M_DIMLES S...pBF.
00d0  d5 56 00 00 00 00 46 f2  d5 56 00 00 00 00         .V....F. .V....
```

○ The user data transferred in a ISSUE submessage (rtps.issueData), 96 bytes | Packets: 1101 · Displayed: 1101 · Marked: 0 | Profile: Default