4m. MONITORING OF ETHERNET/IP NETWORK TRAFFIC.

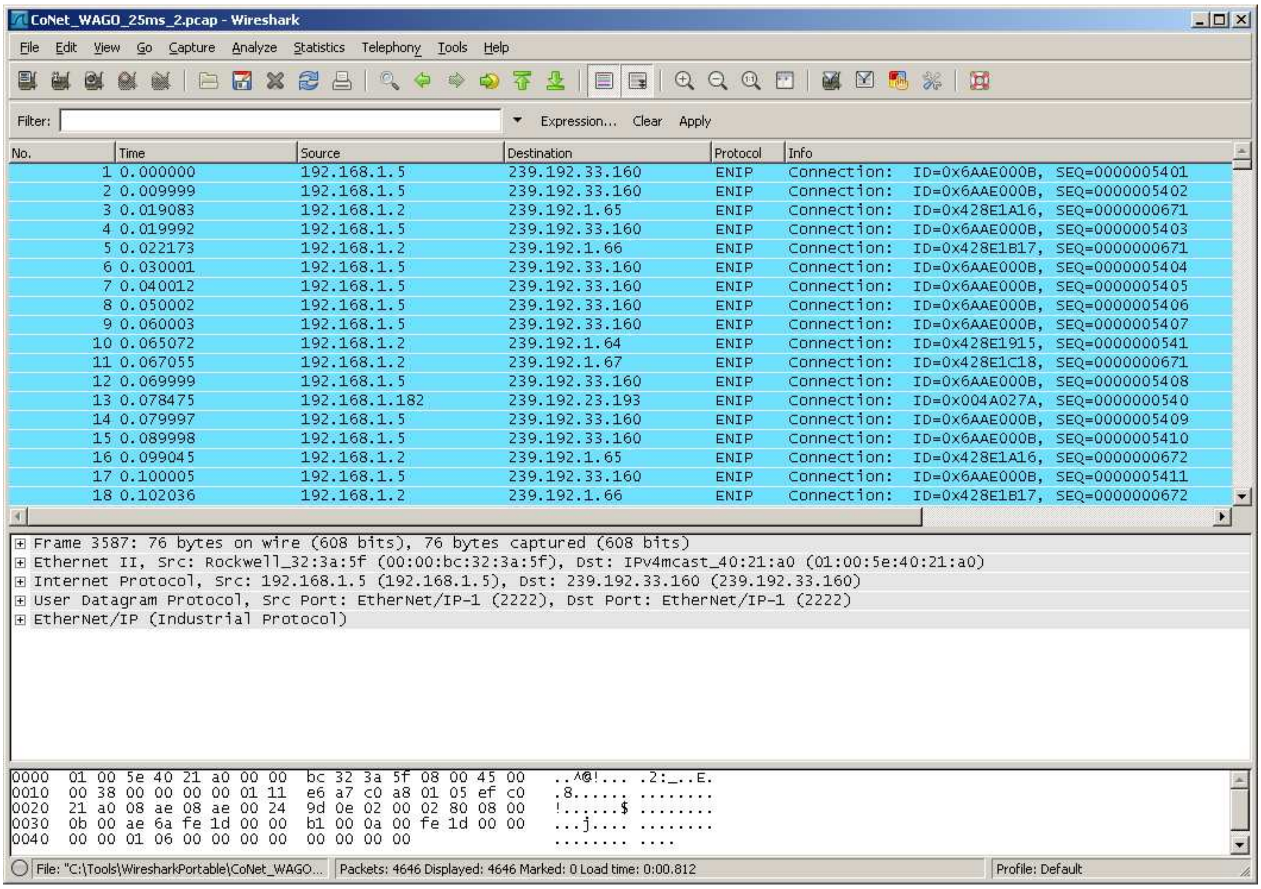
Wireshark (see Section 6) is a network packet analyser . It is used to:

- troubleshoot network problems,
- examine security problems,
- debug protocol implementations,
- learn network protocol internals.

Wireshark can capture traffic from many different network media types, including dial-up connection, cable ethernet LAN and Wi-Fi LAN. It supports more than 900 types of communication protocols. Wireshark is an open source software project, and is released under the GNU General Public Licence (GPL).

The main goal of this exercise is to analyse Ethernet/IP network traffic on the Aero Lift laboratory setup (see Chapter 5) by utilizing the Wireshark application described in Chapter 6. A further aim is to assign the multicast addresses generated on the Ethernet/IP network to the data packets sent by each network node.

The main window of Wireshark, filled with sample data gathered on the Ethernet/IP network, is shown in Fig. 4m.1. Each line in the top panel of the Wireshark window corresponds to a single packet seen on the Ethernet/IP network. The first column shows the time of the packet (relative to the initiation of the capture), the next columns: source and destination IP addresses, the Ethernet/IP protocol (ENIP) used and some additional information about the packet. The user can drill down and obtain more information by clicking on a row. This causes the bottom two window panels to fill with information.



```
No.     Time       Source          Destination     Protocol Info
      1 0.000000   192.168.1.5     239.192.33.160  ENIP     Connection:  ID=0x6AAE000B, SEQ=0000005401
      2 0.009999   192.168.1.5     239.192.33.160  ENIP     Connection:  ID=0x6AAE000B, SEQ=0000005402
      3 0.019083   192.168.1.2     239.192.1.65    ENIP     Connection:  ID=0x428E1A16, SEQ=0000000671
      4 0.019992   192.168.1.5     239.192.33.160  ENIP     Connection:  ID=0x6AAE000B, SEQ=0000005403
      5 0.022173   192.168.1.2     239.192.1.66    ENIP     Connection:  ID=0x428E1B17, SEQ=0000000671
      6 0.030001   192.168.1.5     239.192.33.160  ENIP     Connection:  ID=0x6AAE000B, SEQ=0000005404
      7 0.040012   192.168.1.5     239.192.33.160  ENIP     Connection:  ID=0x6AAE000B, SEQ=0000005405
      8 0.050002   192.168.1.5     239.192.33.160  ENIP     Connection:  ID=0x6AAE000B, SEQ=0000005406
      9 0.060003   192.168.1.5     239.192.33.160  ENIP     Connection:  ID=0x6AAE000B, SEQ=0000005407
     10 0.065072   192.168.1.2     239.192.1.64    ENIP     Connection:  ID=0x428E1915, SEQ=0000000541
     11 0.067055   192.168.1.2     239.192.1.67    ENIP     Connection:  ID=0x428E1C18, SEQ=0000000671
     12 0.069999   192.168.1.5     239.192.33.160  ENIP     Connection:  ID=0x6AAE000B, SEQ=0000005408
     13 0.078475   192.168.1.182   239.192.23.193  ENIP     Connection:  ID=0x004A027A, SEQ=0000000540
     14 0.079997   192.168.1.5     239.192.33.160  ENIP     Connection:  ID=0x6AAE000B, SEQ=0000005409
     15 0.089998   192.168.1.5     239.192.33.160  ENIP     Connection:  ID=0x6AAE000B, SEQ=0000005410
     16 0.099045   192.168.1.2     239.192.1.65    ENIP     Connection:  ID=0x428E1A16, SEQ=0000000672
     17 0.100005   192.168.1.5     239.192.33.160  ENIP     Connection:  ID=0x6AAE000B, SEQ=0000005411
     18 0.102036   192.168.1.2     239.192.1.66    ENIP     Connection:  ID=0x428E1B17, SEQ=0000000672
```

```
⊞ Frame 3587: 76 bytes on wire (608 bits), 76 bytes captured (608 bits)
⊞ Ethernet II, Src: Rockwell_32:3a:5f (00:00:bc:32:3a:5f), Dst: IPv4mcast_40:21:a0 (01:00:5e:40:21:a0)
⊞ Internet Protocol, Src: 192.168.1.5 (192.168.1.5), Dst: 239.192.33.160 (239.192.33.160)
⊞ User Datagram Protocol, Src Port: EtherNet/IP-1 (2222), Dst Port: EtherNet/IP-1 (2222)
⊞ EtherNet/IP (Industrial Protocol)
```

```
0000  01 00 5e 40 21 a0 00 00  bc 32 3a 5f 08 00 45 00   ..^@!... .2:_..E.
0010  00 38 00 00 00 00 01 11  e6 a7 c0 a8 01 05 ef c0   .8...... ........
0020  21 a0 08 ae 08 ae 00 24  9d 0e 02 00 02 80 08 00   !......$ ........
0030  0b 00 ae 6a fe 1d 00 00  b1 00 0a 00 fe 1d 00 00   ...j.... ........
0040  00 00 01 06 00 00 00 00  00 00 00 00               ........ ....
```

○ File: "C:\Tools\WiresharkPortable\CoNet_WAGO... | Packets: 4646 Displayed: 4646 Marked: 0 Load time: 0:00.812 | Profile: Default

Fig. 4m.1. The main window of Wireshark

For real-time messaging, Ethernet/IP employs the UDP over IP, which allows a datagram to be multicast to a group of destination addresses. This producer-consumer multicast model is called “Implicit I/O Data connection” and provides I/O data sending at regular time intervals. Hence, to properly analyse network traffic it is necessary to understand the concept of multicast addressing which is used in the Ethernet/IP protocol. IP multicast addresses (Layer 3 of the OSI) have been assigned to the old Class "D" address space by the Internet Assigned Number Authority (IANA). Addresses in this space are denoted with a binary "1110" prefix in the first four bits of the first octet, as shown in Fig. 4m.2. This results in IP multicast addresses spanning a range from 224.0.0.0 through 239.255.255.255. The remaining 28 bits identify the multicast "Group" the datagram is sent to.

Octet 1	Octet 2	Octet 3	Octet 4
1110xxxx	**xxxxxxxx**	**xxxxxxxx**	**xxxxxxxx**

Fig. 4m.2. IP multicast address format

IP multicast frames all use MAC layer addresses beginning with the 24-bit prefix of 0x0100.5Exx.xxxx. With only half of these MAC addresses available for use by IP Multicast, 23 bits of MAC address space is available for mapping Layer 3 IP multicast addresses onto Layer 2 MAC addresses. All Layer 3 IP multicast addresses have the first four of the 32 bits set to 0x1110, leaving 28 bits of meaningful IP multicast address information. These 28 bits must map onto only 23 bits of the available MAC address. This mapping (for IP multicast equal to 239.192.1.65) is shown graphically in Fig. 4m.3.

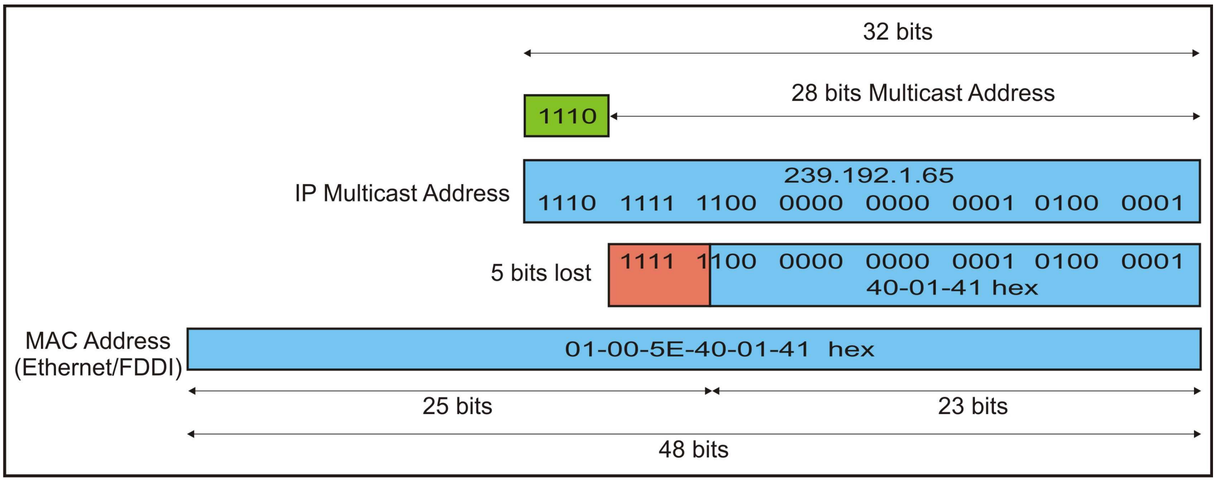


Fig. 4m.3. Multicast MAC address mapping

As can be seen in Fig. 4m.3 all the 28 bits of a Layer 3 multicast address cannot be mapped onto the available 23 bits of MAC address space. This means that each IP multicast MAC address can represent 32 IP multicast addresses (32:1 address ambiguity when a Layer 3 IP multicast address is mapped onto a Layer 2 MAC address). An example of the ambiguity in the mapping process is shown in Fig. 4m.4.

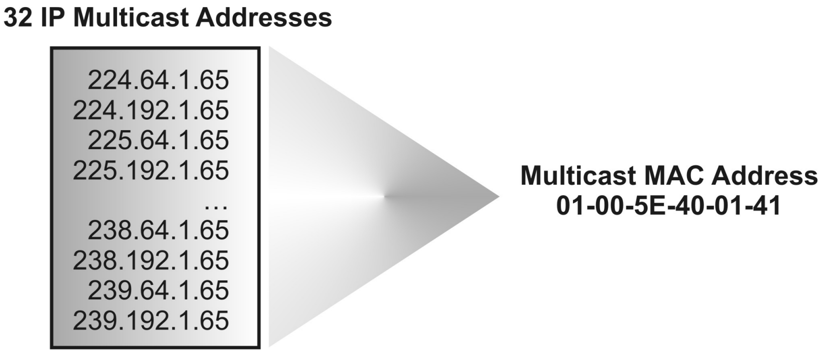


Fig. 4m.4. Ambiguities in the address mapping process

All the IP Multicast addresses are grouped and allocated by the IANA. Currently, these addresses are divided into 10 blocks. The IANA has reserved the range of 239.0.0.0-239.255.255.255 as “Administratively Scoped” addresses for use in private multicast domains. This means that network administrators are free to use multicast addresses in this range inside of their domain without fear of conflicting with others elsewhere on the Internet. These addresses should be used by enterprise networks and should never be forwarded outside the organisation.

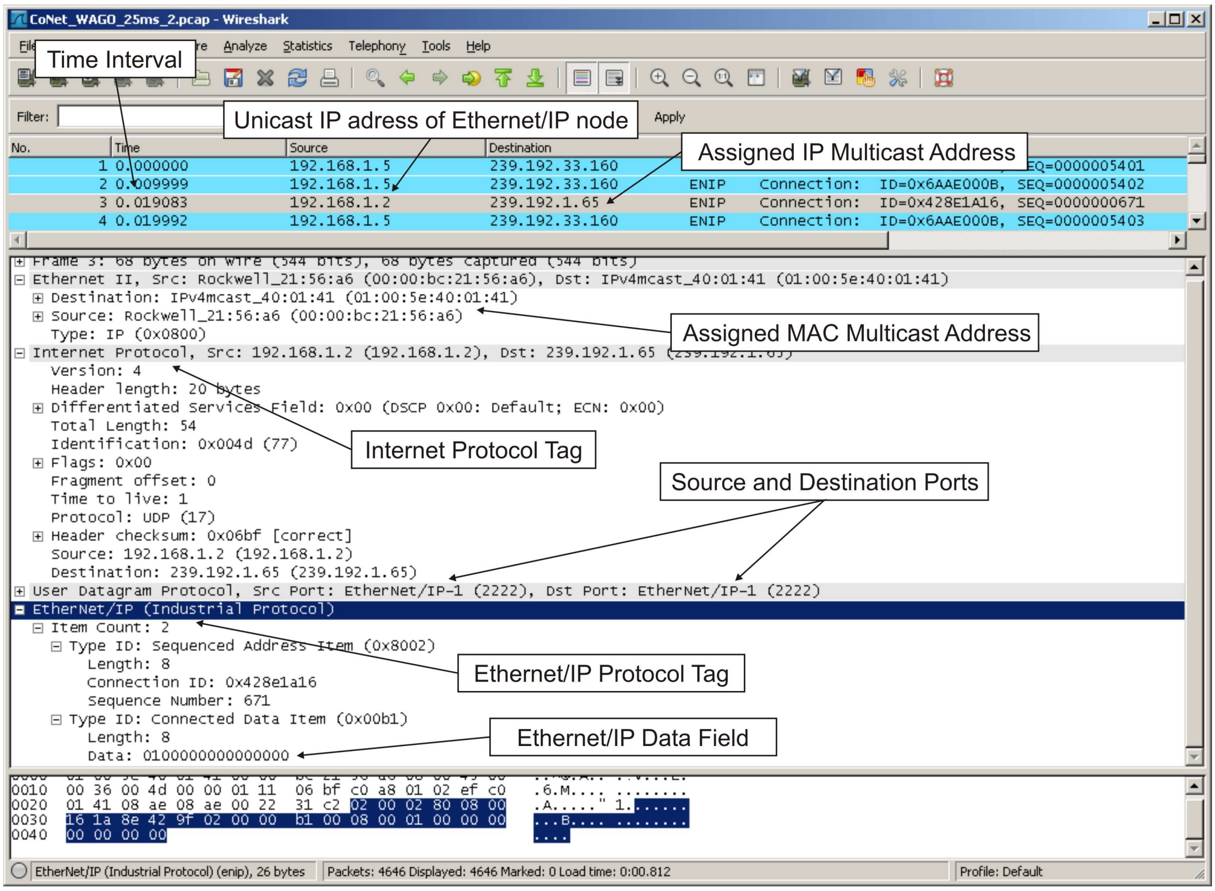
For the users of Allen-Bradley systems this is the most important range of addresses. As can be seen in Fig. 4m.1 the Ethernet/IP multicast addresses start from the same two octets 239.192.xxx.xxx. In fact the all Ethernet/IP multicast addresses may be in the range of 239.192.0.0 to 239.192.255.255.

All multicast addresses in our Ethernet/IP network are associated with standard, unicast IP addresses of specific network nodes. In addition, one unicast IP address may generate multiple multicast addresses. Typically a node (I/O module) multicasts its data every specified RPI interval time. For example, an input module sends data to a controller at the RPI that you assign to the module during the configuration process (see Chapter 3m).

The user can analyse the main parameters of the developed Ethernet/IP network by highlighting the row he or she is interested in, in the top panel of Wireshark. This causes the bottom two window panels to fill with information as shown in Fig. 4m.5. The most important parameters are: multicast IP and MAC addresses assigned to existing network nodes, unicast IP addresses of the nodes, the time interval (measured from the start of data capturing), Internet Protocol and Ethernet/IP protocol parameters.

As an example, let's analyze the data frame numbered three in Fig. 4m.5. The source unicast IP address 192.168.1.2 corresponds to the POINT_IO 1734-AENT (see Table 5.4 in Chapter 5). The assigned destination IP multicast address is equal to 239.192.1.65 and it is mapped onto the MAC multicast address 01:00:5E:40:01:41 (see Fig. 4m.3). The Time To Live (TTL) parameter from the Internet Protocol tag shows, that the packet is restricted to the same subnet and won't be forwarded by a router. The TTL controls the live time of the datagram to avoid it being looped forever due to routing errors. Routers decrement the TTL of every datagram as it traverses from one network to another, and when its value reaches 0 the packet is dropped. The UDP over IP protocol is employed. Source and destination addresses are typical for the Ethernet/IP protocol and in the shown example adopt the same values equal to 2222.

The Sequenced Address Item (0x8002), Connection ID (0x428e1a16), Connected Data Item (0x00b1) and item data length are the same for this type of transaction. The Data field length directly corresponds to the number of bytes transferred from the POINT_IO 1734-AENT.


CoNet_WAGO_25ms_2.pcap - Wireshark
File
Analyze
Statistics
Telephony
Tools
Help
Filter:
Apply
Time Interval
Unicast IP adress of Ethernet/IP node
Assigned IP Multicast Address
Assigned MAC Multicast Address
Internet Protocol Tag
Source and Destination Ports
Ethernet/IP Protocol Tag
Ethernet/IP Data Field

```
No.        Time        Source          Destination
  1 0.000000    192.168.1.5     239.192.33.160                                              EQ=0000005401
  2 0.009999    192.168.1.5     239.192.33.160    ENIP   Connection:  ID=0x6AAE000B, SEQ=0000005402
  3 0.019083    192.168.1.2     239.192.1.65      ENIP   Connection:  ID=0x428E1A16, SEQ=0000000671
  4 0.019992    192.168.1.5     239.192.33.160    ENIP   Connection:  ID=0x6AAE000B, SEQ=0000005403
```

```
⊞ Frame 3: 68 bytes on wire (544 bits), 68 bytes captured (544 bits)
⊟ Ethernet II, Src: Rockwell_21:56:a6 (00:00:bc:21:56:a6), Dst: IPv4mcast_40:01:41 (01:00:5e:40:01:41)
  ⊞ Destination: IPv4mcast_40:01:41 (01:00:5e:40:01:41)
  ⊞ Source: Rockwell_21:56:a6 (00:00:bc:21:56:a6)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 239.192.1.65 (239.192.1.65)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 54
    Identification: 0x004d (77)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 1
    Protocol: UDP (17)
  ⊞ Header checksum: 0x06bf [correct]
    Source: 192.168.1.2 (192.168.1.2)
    Destination: 239.192.1.65 (239.192.1.65)
⊞ User Datagram Protocol, Src Port: EtherNet/IP-1 (2222), Dst Port: EtherNet/IP-1 (2222)
⊟ EtherNet/IP (Industrial Protocol)
  ⊟ Item Count: 2
    ⊟ Type ID: Sequenced Address Item (0x8002)
        Length: 8
        Connection ID: 0x428e1a16
        Sequence Number: 671
    ⊟ Type ID: Connected Data Item (0x00b1)
        Length: 8
        Data: 0100000000000000
```

```
0010  00 36 00 4d 00 00 01 11  06 bf c0 a8 01 02 ef c0   .6.M.... ........
0020  01 41 08 ae 08 ae 00 22  31 c2 02 00 02 80 08 00   .A....." 1.......
0030  16 1a 8e 42 9f 02 00 00  b1 00 08 00 01 00 00 00   ...B.... ........
0040  00 00 00 00                                        ....
```

```
EtherNet/IP (Industrial Protocol) (enip), 26 bytes | Packets: 4646 Displayed: 4646 Marked: 0 Load time: 0:00.812 | Profile: Default
```

Fig. 4m.5. Ethernet/IP network monitoring utilizing Wireshark

Below a two examples of exercises that can be carried out at the laboratory setup are presented.

Exercise 1. Assignment of multicast addresses to a specific transaction data exchange.

The main goal of this exercise is to recognize of all multicast addresses assigned to the Ethernet/IP network nodes and to analyse the tags of an Ethernet frames. All necessary data will be collected using the Wireshark program.

1. First, you have to be aware that all components are connected in a proper way (especially all Ethernet cables).
2. Run RSLinx Classic first and next RSLogix 5000.

3. Open an exemplary project.
4. Check if the local and network configuration has been correctly carried out (see Chapter 1m). The most important are the unicast IP addresses of each node and data exchanging between all network nodes.
5. In the next step set the PLC to the **Run** mode and download program to the program memory of the PLC. After downloading, if everything was setup correctly, the "I/O OK" indicator is green.
6. Run the Wireshark program and set capture interface in accordance with Chapter 6. Click the **Start** button next to the name of the interface on which you wish to capture traffic (or press the **Capture -> Start** option form the **Main** menu). The capture process will start immediately.
7. After a several seconds a running capture session should be stopped by pressing the **Stop** icon located on the toolbar or choosing the **Capture -> Stop** option form the **Main** menu.
8. Now, you can view all captured Ethernet frames, assigning a multicast addresses by reading source unicast IP addresses and analysing frame tags. It is important that one unicast IP address can generate a few multicast addresses (each type of data transaction occupies a different multicast address).

Exercise 2. Analysis of the Requested Packed Interval (RPI) time.

The main goal of this exercise is to experimentally analyse variation of the RPI, which decides about refreshing of I/O data over the Ethernet/IP network. All necessary data will be collected using the Wireshark program.

The exercise assumes that students carried out the configuration of the Ethernet/IP network nodes (described in Section 1m) and can check and change the value of the RPI.

1. Repeat steps 1-3 from the Exercise 1.
2. Read RPI parameter set for each I/O module (open the **Module Properties** window and select the **Connection** tag). For each RPI assign the unicast IP address of the corresponding module.
3. Repeat steps 5-7 from the Exercise 1.
4. Locate and write down all type of transactions with the same source and destination addresses, for example: 192.168.1.5 – 239.192.33.160, 192.168.1.182 – 239.192.23.193 etc.

5. Order all of the frames according to the capturing time (the second column of the main window of the Wireshark).
6. Read and write down times for the ten consecutive transactions of the same type.
7. Calculate the RPI by subtracting two consecutive capturing times wrote in step 6.
8. Compare the calculated values of the RPI with values set during the configuration stage.
9. Repeat steps 6-8 for all transactions of the same type.

Bibliography

[1] CISCO Systems Inc., Guidelines for Enterprise IP Multicast Address Allocation, http://www.cisco.com, 2004.

[2] RFC 3171, IANA Guidelines for IPv4 Multicast Address Assignments, Best Current Practice, July 2001, http://tools.ietf.org/html/rfc3171, 2010. [2 dates???]

Author: M. Rosol