Computer Networks - Lab Session2:Layer 2: CSMA-CD and VLANs

Ganapathy Ram Krishnakumar, Alisa Patotskaya,
Riyane Sid-Lakhdar, Matthias Kohl

May 10, 2016

Contents

1 CSMA/CD

Thanks to the tool "udpmt", we have generate a UDP traffic from the machine "182.168.0.3" totitletitle the machine "192.168.0.4". We could notice that there is no collisions were detected. This caused by the fact that before sending a frame according to CSMA/CD protocol, a station always listens-in to the cable to check that no other station is already in the process of sending data.The target machine has received the statistics on figure 1

packets	errs	idrops	bytes	packets	errs	bytes	colls
3	0	0	656	0	0	0	0
7	0	0	975	0	0	0	0
3	0	0	421	0	0	0	0
4	0	0	791	0	0	0	0
1	0	0	237	0	0	0	0
1	0	0	237	0	0	0	0
1	0	0	60	13650	0	22587366	0
63	0	0	11825	16252	0	24660366	0
13	0	0	3380	16260	0	24660032	0
6	0	0	1562	16250	0	24660032	0
1	0	0	252	16260	0	24660032	0
3	0	0	394	16251	0	24660078	0
7	0	0	661	16250	0	24660032	0
4	0	0	619	16260	0	24198262	0
4	0	0	629	16250	0	24351176	0
3	0	0	588	16260	0	24660032	0
3	0	0	462	16250	0	24660032	0
5	0	0	466	16260	0	24660032	0
7	0	0	896	16250	0	24660032	0
6	0	0	526	16250	0	24660032	0
3	0	0	276	16260	0	24660032	0
input			(Total)	output			
packets	errs	idrops	bytes	packets	errs	bytes	colls
2	0	0	184	16250	0	24660032	0
8	0	0	716	16260	0	24660032	0
6	0	0	665	16250	0	24660032	0
5	0	0	562	16260	0	24660032	0
6	0	0	526	16250	0	24660032	0
3	0	0	388	16250	0	24660032	0
4	0	0	507	16260	0	24660032	0
5	0	0	535	16250	0	24551024	0
7	0	0	895	16260	0	23998414	0
6	0	0	845	16250	0	24660032	0
5	0	0	772	16260	0	24660032	0
5	0	0	896	16250	0	24660032	0
3	0	0	244	16250	0	24660032	0
6	0	0	798	16260	0	24660032	0
3	0	0	276	16250	0	24660032	0
2	0	0	184	16260	0	24660032	0
3	0	0	276	16250	0	24660032	0
2	0	0	184	16260	0	24660032	0
4	0	0	336	16250	0	24660032	0
4	0	0	502	16251	0	24660078	0
2	0	0	184	16260	0	24660032	0

Figure 1: UDP traffic of target machine (two machines in network)

Using "netstat", we have also observed (figure 2)the number of collisions (observed on the source machine).

packets	errs	idrops	bytes	packets	errs	bytes	colls
12	0	0	3268	16250	0	24660032	0
5	0	0	1522	16260	0	24660032	0
0	0	0	0	16250	0	24660032	0
2	0	0	264	16260	0	24660032	0
1	0	0	60	16250	0	24660032	0
3	0	0	660	16260	0	24660032	0
5	0	0	678	16250	0	24660032	0
6	56	0	887	12676	0	19265650	471
4	201	0	684	7263	0	10788764	1454
1	154	0	60	10135	0	15412520	1479
0	155	0	0	8469	0	13100642	1509
1	174	0	239	8429	0	12330016	1489
0	179	0	0	7673	0	11559390	1490
2	196	0	684	7006	0	10788764	1468
197	202	0	13770	8268	0	12330016	1497
200	64	0	13970	3197	0	3082562	715
1	0	0	60	0	0	0	0
1	0	0	226	0	0	0	0

Figure 2: UDP traffic of target machine (three machines in network)

If we add a third machine which sends UDP packets on the network, we can notice that the number of collisions increases up to 1509.

Packet Size / Try (secs)	1	2	3	4	5	6
60	10510.1	47617.0	47619.8	47616.0	47620.8	47616.0
500	23168.0	88328.0	88340.0	88344.0	88340.0	88328.0
1000	87672.0	93808.0	93808.0	93800.0	93808.0	93808.0
1460	40599.7	95682.6	95702.9	95702.9	95682.6	95670.9
1472	78322.2	95715.3	95703.6	95703.6	95703.6	95715.3
1480	37319.7	91239.0	91250.9	91250.9	91227.2	91250.9
1500	43560.0	92372.4	92355.6	92360.4	92364.0	92460.0
2880	37716.5	95869.4	95869.4	95869.4	95869.4	95892.5

Table 1: Measured bandwidth

We could recreate this collisions using only two machines by making each machine send and receive UDP packets. For this purpose we could use a half-duplex crossover Ethernet cable. In case if both machines will communicate at the same time, the collision will occur. It could be fixed by using full-duplex to make machines send and receive packages at the same time.

2 Analyzing network performances

2.1 Bandwidth measurement on an unloaded network

In the Table 1 represented measured bandwidth values for different packet sizes, observed thanks to "udpmt".
First of all, this bandwidth is measured as the number of UDP data sent per second. However, each UDP packet is loaded with headers from lower layer protocols (such as Ethernet and IP). Thus, to each UDP packet of size s bits, s_+ bits are sent (with $s_+ > s$). Because of this we never obtain theoretical bandwidth.
Second of all, when the packet size is 1460, each packet is sent using only one transfer. If the packet size is 1480 bytes, each packet is split into 2 packets before it is sent. However, the size of second packet is very small. Thus the bandwidth is decreased from 78322 to 337319.
Third of all, we could notice that the worst bandwidth we obtain with small packet size and the best is with packet size almost equal to maximum (see the Figure 3). This caused by the fact that in case of small packets the size of headers almost equals to size of packet itself.

2.2 Bandwidth measurement on an loaded network

2.2.1 Hub based network

In the figure 4 are represented the UDP statistics of the two loaded parts of the traffic.

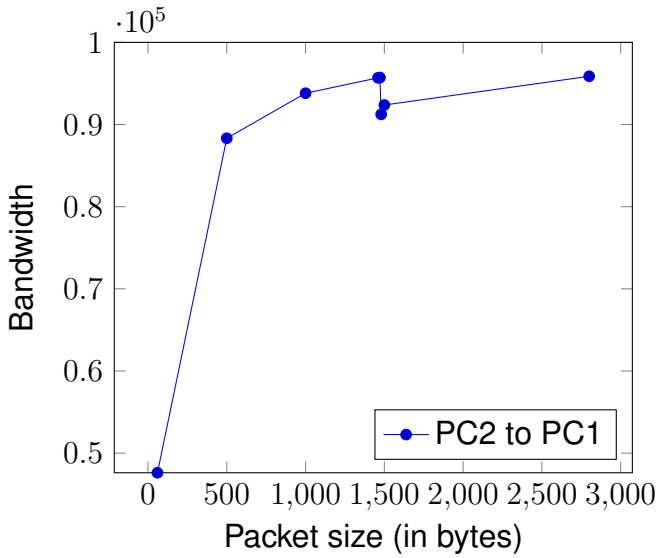


Figure 3: Network bandwidth

input			(Total)	output			
packets	errs	idrops	bytes	packets	errs	bytes	colls
0	0	0	0	0	0	0	0
11745	0	0	17779411	0	0	0	0
16257	2	0	24609180	0	0	0	0
16255	4	0	24608779	0	0	0	0
16232	0	0	24573976	0	0	0	0
16248	52	0	24596914	1	0	46	0
16252	10	0	24605528	0	0	0	0
16255	3	0	24610070	0	0	0	0
16246	22	0	24592285	0	0	0	0
16252	22	0	24602684	0	0	0	0
16256	2	0	24607326	0	0	0	0
16259	0	0	24612002	0	0	0	0
15948	322	0	24142436	398	0	27860	98
15993	257	0	24212092	398	0	27860	112
5340	649	0	8078043	199	0	13930	178
9	319	0	1590	0	0	0	0

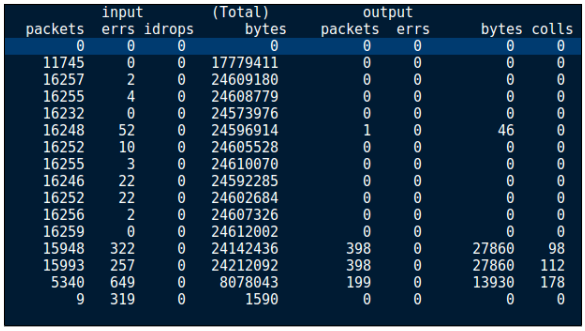
Figure 4: Bandwidth measures on a hub based network

We have noticed that the bandwidths are much lower than those in the unloaded network. This loss of efficiency may be explained by the higher number of collision: As the network is loaded, the probability that 2 packets be sent at the same time (hence get lost because of a collision) is higher. Which makes the transmission of this packet slower.

2.2.2 Switch based network

The figure 5 represents the UDP statistics of the two loaded parts of the traffic.

We have noticed that the average bandwidth is now the same as on the unloaded network. This observation may be explained by the fact that the switch is acting as a "link" for the application protocol: there is no rooting for the higher layer protocols, hence there are no collisions.
Thus, on this local area network, a switch may be more efficient than a hub when the probability of a collision become significant (loaded network).

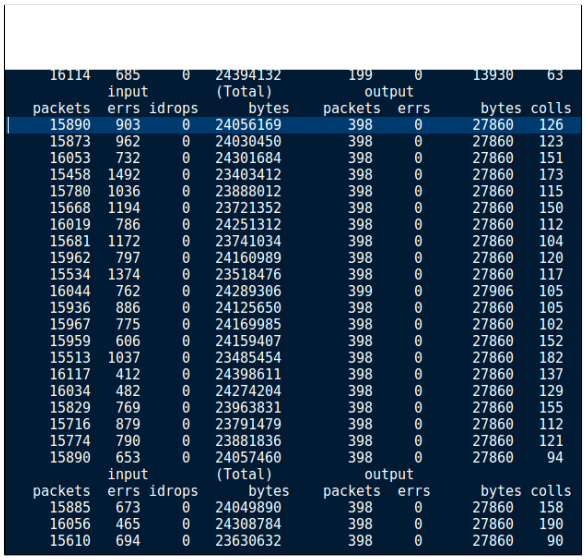


input			(Total)	output			
packets	errs	idrops	bytes	packets	errs	bytes	colls
0	0	0	0	0	0	0	0
11745	0	0	17779411	0	0	0	0
16257	2	0	24609180	0	0	0	0
16255	4	0	24608779	0	0	0	0
16232	0	0	24573976	0	0	0	0
16248	52	0	24596914	1	0	46	0
16252	10	0	24605528	0	0	0	0
16255	3	0	24610070	0	0	0	0
16246	22	0	24592285	0	0	0	0
16252	22	0	24602684	0	0	0	0
16256	2	0	24607326	0	0	0	0
16259	0	0	24612002	0	0	0	0
15948	322	0	24142436	398	0	27860	98
15993	257	0	24212092	398	0	27860	112
5340	649	0	8078043	199	0	13930	178
9	319	0	1590	0	0	0	0

Figure 5: Bandwidth measures on a switch based network

2.2.3 Hub based network, single machine stressed

The figure 6 represents the UDP statistics of the machine that receives all the packets. We could notice that the number of collisions is very high. This caused by the fact that 1, 3 and 4 are sending packages in asynchronous way. Thus a lot of packages are coming through hub at the same time. Because of this bandwidth is decreased compared to previous results.



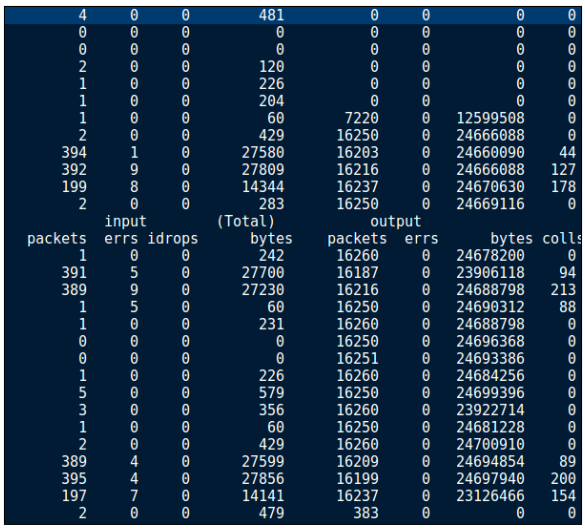
16114	685	0	24394132	199	0	13930	63
input			(Total)	output			
packets	errs	idrops	bytes	packets	errs	bytes	colls
15890	903	0	24056169	398	0	27860	126
15873	962	0	24030450	398	0	27860	123
16053	732	0	24301684	398	0	27860	151
15458	1492	0	23403412	398	0	27860	173
15780	1036	0	23888012	398	0	27860	115
15668	1194	0	23721352	398	0	27860	150
16019	786	0	24251312	398	0	27860	112
15681	1172	0	23741034	398	0	27860	104
15962	797	0	24160989	398	0	27860	120
15534	1374	0	23518476	398	0	27860	117
16044	762	0	24289306	399	0	27906	105
15936	886	0	24125650	398	0	27860	105
15967	775	0	24169985	398	0	27860	102
15959	606	0	24159407	398	0	27860	152
15513	1037	0	23485454	398	0	27860	182
16117	412	0	24398611	398	0	27860	137
16034	482	0	24274204	398	0	27860	129
15829	769	0	23963831	398	0	27860	155
15716	879	0	23791479	398	0	27860	112
15774	790	0	23881836	398	0	27860	121
15890	653	0	24057460	398	0	27860	94
input			(Total)	output			
packets	errs	idrops	bytes	packets	errs	bytes	colls
15885	673	0	24049890	398	0	27860	158
16056	465	0	24308784	398	0	27860	190
15610	694	0	23630632	398	0	27860	90

Figure 6: Collision measured on a hub based network, where one machine receives all the UDP packets

2.2.4 Hub based network, single machine sending packets

The figure 7 represents the UDP statistics of the machine that receives all the packets.

We can notice that once we are sending packages from the second machine to all others, we obtain smaller amount of collisions. This caused by the fact that machines 1, 3 and 4 are sending packages in asynchronous way. Thus a lot of



4	0	0	481	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
2	0	0	120	0	0	0	0
1	0	0	226	0	0	0	0
1	0	0	204	0	0	0	0
1	0	0	60	7220	0	12599508	0
2	0	0	429	16250	0	24666088	0
394	1	0	27580	16203	0	24660090	44
392	9	0	27809	16216	0	24666088	127
199	8	0	14344	16237	0	24670630	178
2	0	0	283	16250	0	24669116	0
input			(Total)	output			
packets	errs	idrops	bytes	packets	errs	bytes	colls
1	0	0	242	16260	0	24678200	0
391	5	0	27700	16187	0	23906118	94
389	9	0	27230	16216	0	24688798	213
1	5	0	60	16250	0	24690312	88
1	0	0	231	16260	0	24688798	0
0	0	0	0	16250	0	24696368	0
0	0	0	0	16251	0	24693386	0
1	0	0	226	16260	0	24684256	0
5	0	0	579	16250	0	24699396	0
3	0	0	356	16260	0	23922714	0
1	0	0	60	16250	0	24681228	0
2	0	0	429	16260	0	24700910	0
389	4	0	27599	16209	0	24694854	89
395	4	0	27856	16199	0	24697940	200
197	7	0	14141	16237	0	23126466	154
2	0	0	479	383	0	0	0

Figure 7: Bandwidth measures on a hub based network, where one machine sends all the UDP packets

packages are coming through hub at the same time. From the other hand, second machine is sending packages to machines 1, 3 and 4 sequentially, thus smaller amount of collisions appear. As far as number of collisions is smaller, the bandwidth in this case is higher.

2.3 TCP / UDP interaction

We could observe, that TCP is heavy-weight Internet protocol - it's header size is 20 bytes (compare to UDP Header size, which is 8 bytes). Thus, as we can notice it on the figure 8, the average throughput for same size packet exchange using UDP is higher than using TCP.

Moreover, there is no possibility to set package size

2.4 Virtual Networks - VLANs

A Local Area Network (LAN) was created with PC1, PC2 and PC3 being in the same sub-network and all of the PC's uses the beg0 port. A connection is then established to configure the switch. We used the sub-network $192.168.0.0/16$ to communicate between all PCs and the switch, which is at address $192.168.0.254$. At the start of this operation, all PCs are in VLAN 1.

Next PC3 is now removed from VLAN 1 and it is assigned to VLAN 2. The following commands are used inside the switch configure terminal.

```
HP ProCurve Switch 6108(config)#vlan 2
HP ProCurve Switch 6108(vlan-2)#untagged 3
```

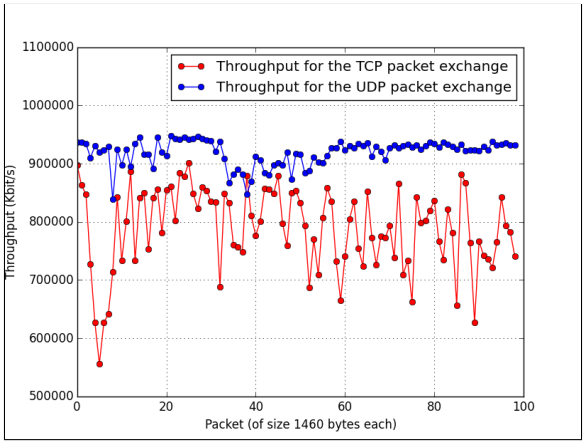


Figure 8: Throughput comparison for packet exchange using TCP and UDP protocols

So, what has happened because of these commands is that the port 3 has been assigned to a different VLAN i.e VLAN 2 and hence they are no more in the same sub-network. When we try to ping PC3 from PC2, it fails because of the fact that the two PC's are on different virtual networks and are isolated.

Since they are on different networks we are not able to communicate and hence we need to assign a different address to PC3 and make sure that we are able to ping PC2 and PC3.

So now we configure the system again and this time we use 2 different ports from PC2 because then we can ensure PC1 and PC2 are in one subnetwork and PC2 and PC3 are in another subnetwork.

Machine name	Interface	IP address
PC1	bge0	192.168.0.1
PC2	bge0	192.168.0.2
PC2	ue0	10.0.0.2
PC3	bge0	10.0.0.3

So now with this we can ping PC1 from PC2 and PC3 from PC2. But that is not the case between PC1 and PC3 because they are still on different sub-networks and they cannot ping each other. To solve this we use the virtual interfaces that are created using the following commands.

```
ifconfig vlan0 create
ifconfig vlan0 vlan 1 vlandev bge0
ifconfig vlan0 192.168.0.2/16 up

ifconfig vlan1 create
ifconfig vlan1 vlan 2 vlandev bge0
ifconfig vlan1 10.0.0.2/24 up
```

Now the IP Configuration of PC2 is as follows:

```
[root@ensipc230 ~]# ifconfig
bge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 150
        options=c019b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,T
        ether 00:0a:f7:03:d0:5f
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
vlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 15
        options=103<RXCSUM,TXCSUM,TSO4>
        ether 00:0a:f7:03:d0:5f
        inet6 fe80::20a:f7ff:fe03:d05f%vlan0 prefixlen 64 scopeid 0x5
        inet 192.168.0.2 netmask 0xffff0000 broadcast 192.168.255.255
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        vlan: 1 parent interface: bge0
vlan1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 15
        options=103<RXCSUM,TXCSUM,TSO4>
        ether 00:0a:f7:03:d0:5f
        inet6 fe80::20a:f7ff:fe03:d05f%vlan1 prefixlen 64 scopeid 0x6
        inet 10.0.0.2 netmask 0xffffff00 broadcast 10.0.0.255
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
        vlan: 2 parent interface: bge0
```

To make sure that the PCs are able to communicate using these virtual VLANs, we can check the frame in wireshark for the interface id's and how they communicate with each other. To check if the system is using the Virtual VLANs, a capture on wireshark was done and Figure-9 shows that the interface that is used here is the VLAN 0 that was created.

The same is done between PC2 and PC3 and this would show that the interface used is the Virtual VLAN 1 and it is shown in Figure-10.

Figure 11 shows a Wireshark trace from PC2 (over the interfaces bge0 and ue0) of a ping command launched from PC1 to PC3. It can be observed that there is never a request and response occurring directly between the IPs $192.168.0.1$(PC1) and $10.0.0.3$(PC3). This is because two VLANs on different subnetworks cannot communicate with each other and hence they connect through these virtual interfaces VLAN 0 and VLAN 1. So only when a request from the same subnetwork come to VLAN 0 which is in PC2 and is connected to PC1, the message gets transferred and thus the message fails. So, what happens here is that from PC1 a request with the destination address as PC3 is sent to PC2 on the VLAN 0 interface. This request is accepted and then this is transferred to VLAN 1 and since the destination address is in the VLAN 1, the reply goes to PC3 and once it responds

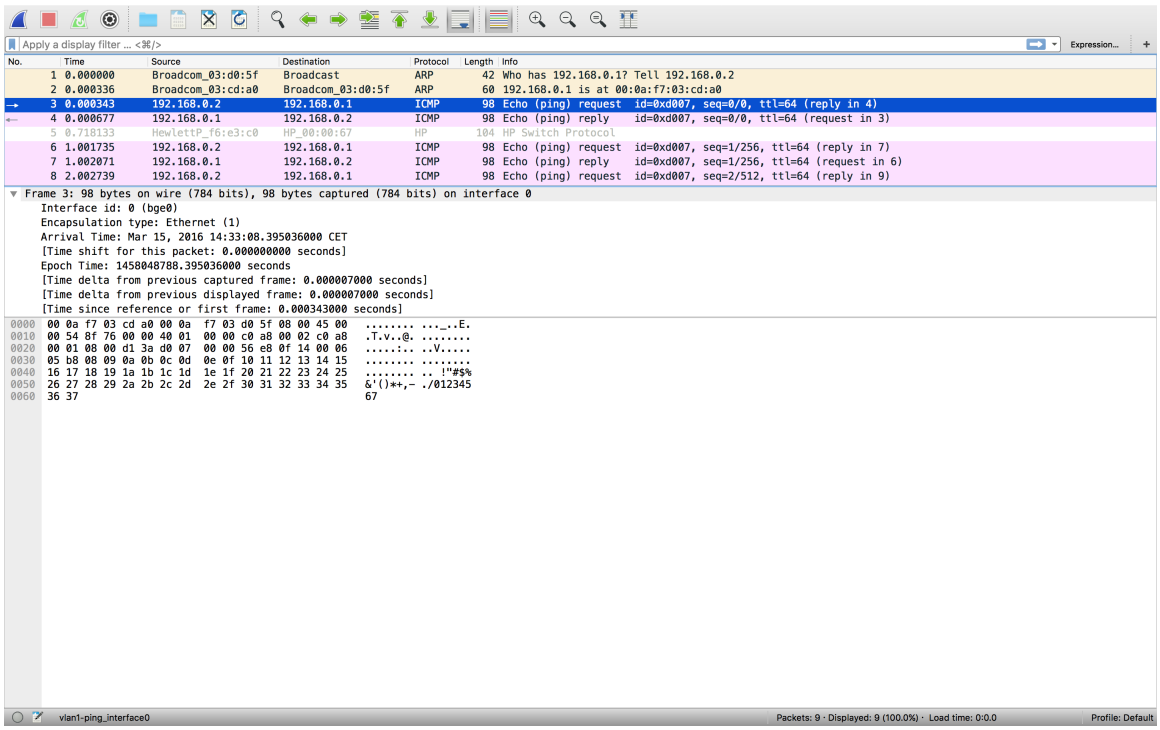


Figure 9: Wireshark trace of a ping command from PC1 to PC2 with the virtual Interface Id 0

the same process happens again but from VLAN 1 to VLAN 0 and this is exactly what happens in Figure 11.

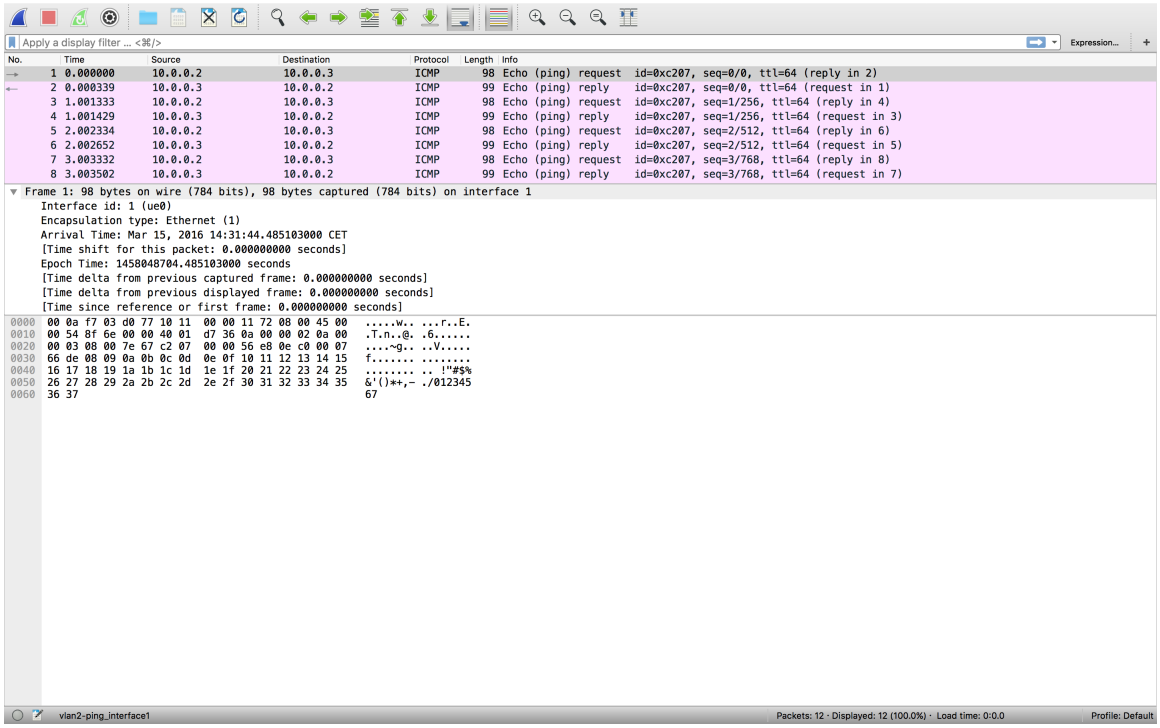


Figure 10: Wireshark trace of a ping command from PC1 to PC2 with the virtual Interface Id 1

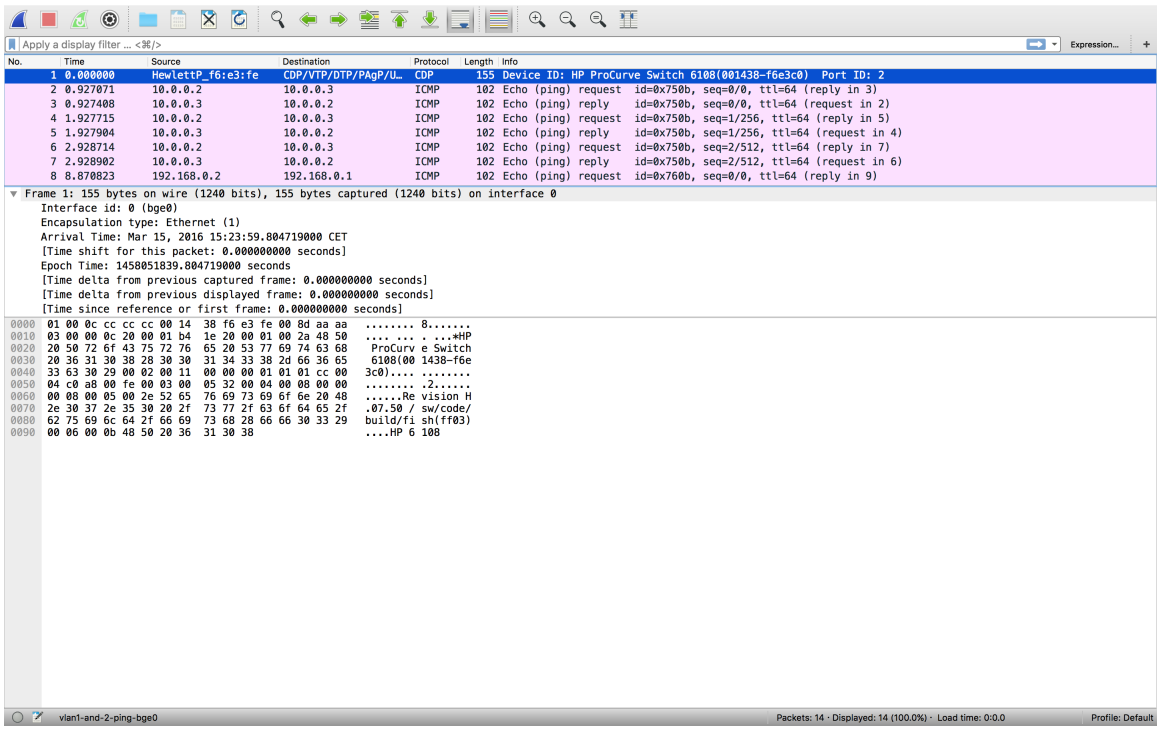


Figure 11: Wireshark trace of a ping command from PC1 to PC3 with the virtual Interface ids.