Computer Networks - Lab Session #4
Layer 2: CSMA-CD & VLANs

1 Observing CSMA/CD protocol on a simple example

The protocol of the MAC (Multiple Access Carrier) sub-layer of layer 2 is called CSMA/CD (Carrier Sense Multiple Access/Collision Detection), and has been standardized by the IEEE under the name 802.3; it enables one to control access to the Ethernet medium. It is implemented on the Ethernet card added to workstations (refer to Introduction to the platform).

It is based on the following mechanism: before sending a frame, a station always listens-in to the cable to check that no other station is already in the process of sending data. When the medium becomes available, the station sends its **frame**.

It sometimes happens that two stations connected to the same Ethernet cable decide to send a frame simultaneously (or nearly simultaneously). In this case, the electric signals combine with each other and neither of the two frames can be read: when this happens, we say that a collision has occurred. Each station is capable of detecting these **collisions** while it sends a frame. When a collision is detected, each station stops transmitting and then waits for a random time interval before attempting to re-send the entire frame.

Note : the information interchanged on the physical medium constitute packets that have a particular format called a frame. A frame always contains the address of the system that sent it, and the address of the system for which it is destined. In the case of Ethernet networks, the frames interchanged have a maximum length of 1518 bytes, and a minimum length of 64 bytes.

The aim here is to study one aspect of CSMA/CD protocol.

Operation 0

In a window of one workstation, run netstat so that it displays at regular intervals (let's say 2 seconds) the number of collisions observed on this interface :

```
# netstat -w 2 xl0
```

In another window, use udpmt (see below) to generate traffic to a second system. You should previously have started the `udptarget` server on the target system. The traffic used here is of UDP stream type. ‖ *Note the number of collisions.*

Notes :

- Using UDP, the number of packets sent may differ from the number of packets received;
- netstat states the number of collisions per interface. It must be run on the same system as udpmt.

On a third system, run udpmt targeting the fourth workstation of the platform.
‖ *Note the variation in the number of collisions. Explain this variation, discussing the CSMA/CD protocol.*

2 Analyzing network performances

In this section, you are going to use the udpmt utility to calculate the effective bandwidth of the network. Remember that the effective bandwidth is the bandwidth that the user actually experiences, compared with the theoretical bandwidth, which is a characteristic of the line.

To calculate this bandwidth, `udpmt` sends packets of variable size to another system on the network. The packets are necessarily intended for a system of your choice.

The syntax for udpmt is as follows:

```
# udpmt [-p port] target_host
# udptarget [-p port]
```

Other sender *options* include

- `-s <size>`: packet size in bytes
- `-t <int>`: timer interval in microseconds (default 500)
- `-d <secs>`: test duration in seconds (default transmit until stopped)

On the target, one can specify the display interval in ms : `-i <ms>`.

2.1 Bandwidth measurement on an unloaded network

The network is said to be unloaded (or little loaded) when it is not carrying much traffic? in other words, when it is nearly not in use. This is effectively the case for your network, on which the four stations are not communicating.

Operation 1

On one of the four stations, use udpmt to continuously send UPD packets to another station (refer to figure 1).
‖ *Note the report output from udpmt, varying the size of the packets (from 60 to 2880 bytes). Measure finely the passage between 1460 bytes and 1480 bytes. What do you deduce, remembering that the theoretical bandwidth of Ethernet is 100 Mb/sec?*

To understand why the Ethernet's effective bandwidth does not attain the theoretical bandwidth, start the ethereal utility on a third system, and sniff some frames interchanged during the execution of udpmt. Analyze the packets interchanged, browsing through the ethereal window.

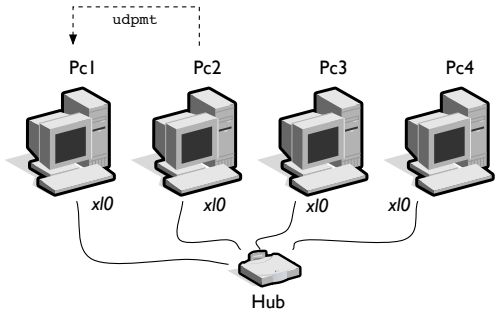

Figure 1: Network performance evaluation

Explain what happens during a udpmt. Try to precisely locate the figures you note. Explain, in particular, the change between 1460 bytes and 1480 bytes.

Here are a few points that will help you to understand:

- `udpmt` udpmt uses the UDP protocol to send its messages and measure bandwidth;
- UDP (like TCP) adds control bytes (called the header) to the data provided to it. These UDP packets (header + data) are passed to another protocol (IP), which also adds control bytes. This IP protocol then passes everything to the Ethernet protocol, which deals with physically sending the bytes over the cable. Ethernet also adds a header. For the moment, don't worry about understanding the signification of the fields of the various headers. Instead, just look at the packet length and header length fields.

Reminder : Figure 2 shows the efficiency curves of the level 2 protocol used in Ethernet cards (CSMA/CD):

State the curve bandwidth = function(packet size) that you obtain with one single piece of traffic (case of one single station attempting to send), and comment on it. Check the size of the packets that circulate in each case.

2.2 Measuring bandwidth on a loaded network

Operation 2

Choose two stations, S1 and S2, to be used for loading the network (continuous transmission of UDP packets).

As before, start udpmt between these two stations.

Then run udpmt between the remaining two stations, as shown in Figure 3.

Build the average bandwidth curves, varying the size of packets as previously. What do you notice?

Compare the results with those obtained previously. What do you deduce about network capacity?

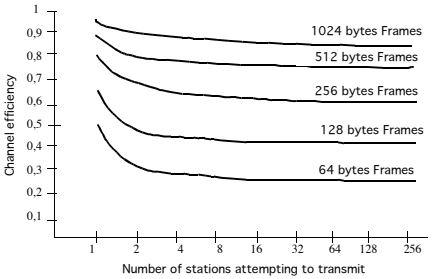

Figure 2: Efficiency of CSMA/CD

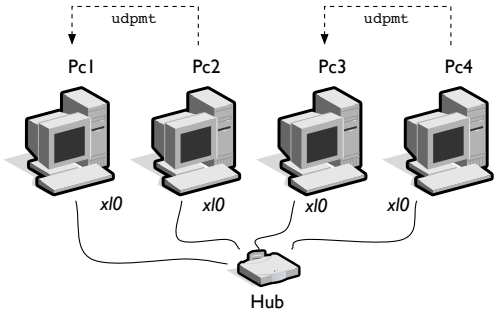

Figure 3: 2 streams

Operation 3 Restart the same operation, but first replace the hub with a switch (see Figure4). When the LEDs are active, start the measurements (if the LEDs do not illuminate, execute ping commands from each of the stations).

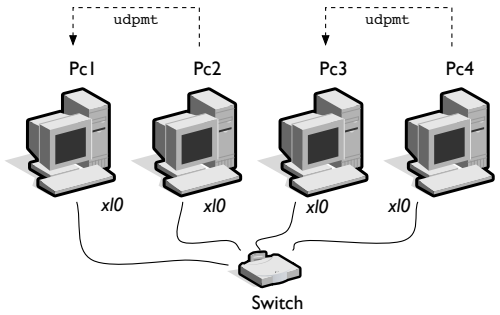

Figure 4: 2 streams and a switch

Build the average bandwidth curves for the two udpmt-generating stations. What do you notice?
Compare these results with those obtained during the preceding operation.
What do you deduce about the operation of a switch? Try to argue the benefits of a switch within a local area network.

Operation 4

Replace the switch with a hub.

Now run udpmt from three stations to a fourth, as shown in Figure 5.

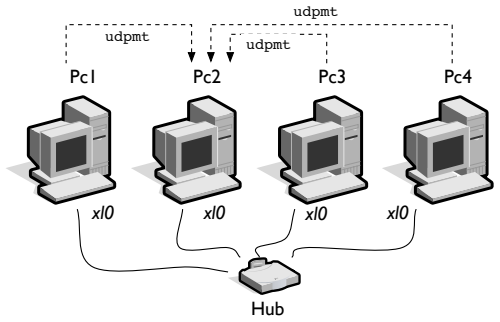

Figure 5: 3 convergent measurement streams

Build the curves and compare them with the previous results.

Operation 5

Now perform the experiment illustrated in Figure 6.

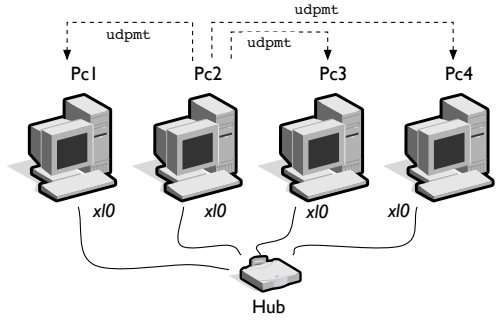

Figure 6: 3 divergent traffic

Build the curves and compare them with the others.
What conclusions do you draw from all these measurements?

3 TCP / UDP interaction

The purpose of this operation is to observe the differences of behavior between TCP and UDP.

Operation 6

How do you explain the number of collisions that occur when a single TCP stream is present on the network?
Start two `tcpmt` *(Figure 3) sending TCP traffic. What can you say about the changes in the number of collisions on each interface, and about the sharing of the bandwidth between the various traffic streams?*
Replace one of the preceding two TCP traffic with UDP traffic. What do you observe?

remark: `tcpmt` (respectively `tcptarget`) can be executed similarly as `udpmt` (respvt `udptarget`).

4 Virtual Networks - VLANs

A Local Area Network (LAN) is defined as a single diffusion domain. If a hub interconnects all the LAN stations, the collision and flooding domains are identical. A switch has the strong asset to create one collision domain per port, or to eliminate it entirely if it supports full duplex communications and if one single station is plugged to each port (figure 7).

Some switches (in particular the *HP Procurve 6108*) allows to create several virtual LAN (VLANs). One VLAN is a LAN that groups virtually together different stations: the station are grouped according to logical (and non physical) criteria (MAC address, port number, protocol, etc.) . All these virtual LAN behave like a physical LAN.

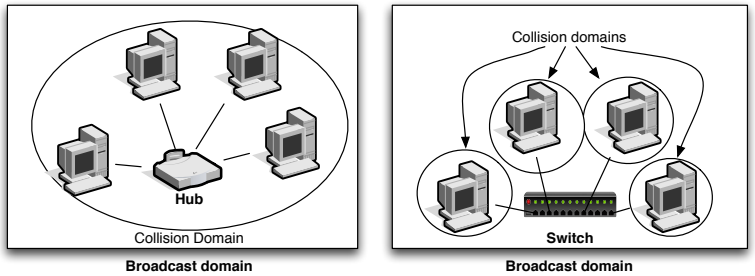

Figure 7: Collision domain : Hub/Switch

Administering a switch consists in defining VLAN. By default, all the ports are attributed to one single VLAN. You can administer your switch via telnet, a serial line, or via a web interface. In the last case, you have naturally to define before its IP parameters.

4.1 Switch configuration

You must verify that all the ports are assigned to the default VLAN (VLAN id=1). You can connect to the switch via `telnet 192.168.0.254`, the IP address of the switch. If it does not work, let do a reset as described below. Of course, a switch a layer 2 device (but it implements the layer 3, and the whole IP stack for its administration).

`telnet` will allow a privileged connection: you can take a look on the configuration and pass in configuration mode. The prompt is then :

```
HP ProCurve Switch 6108#
```

Small howto (CLI) :

- There exists a hierarchy in the commands: each level becomes more and more specific. The ? prints the available commands.
- The tabulation allows an automatic completion ;
- Shortcuts are allowed (if there does not exist any ambiguity):
  `sh ru` is equivalent to `show running-configuration`
- `exit` (ou `^z`) exist from the current mode
- To cancel an action, let use `no` followed by the command
- When you use a serial line, you must configure the program with 9600 bauds, 8N1 parity, and deactivate the hardware flow control
- You can reinitialize the switch by maintaining the Reset button pressed and while rebooting the switch

Verify the current configuration with `show running-config`. If it is not, let reset its configuration.

```
vlan 1
   name "DEFAULT_VLAN"
   untagged 1-8
   ip address 192.168.0.254 255.255.255.0
   exit
```

To enter in configuration mode, let enter `configure terminal` and the prompt will become :

```
   HP ProCurve Switch 6108(config)#
```

4.2 VLAN management

We will focus on level 1 VLAN (defined by port) and *VLANs 802.1Q* to tag frames according to their VLAN number.

The reference configuration is described in figure 8.

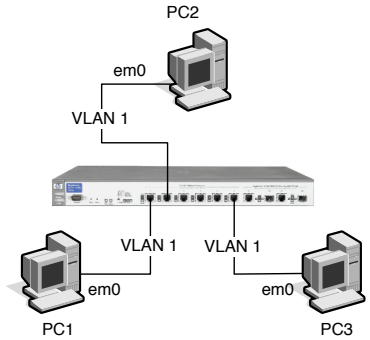

Figure 8: Reference Configuration

Operation 7 Let configure the interfaces of PC1, PC2 and PC3 in the same subnetwork. Let verify communications.

What is the subnetwork so that stations can communicate with the switch? Let prove it.

4.2.1 Level 1 VLANs / ports VLANs

A VLAN is here defined and one port is associated to one single VLAN. All the stations connected to this port will own to this VLAN.

You must first define a new VLAN, and associate a given port to this new VLAN. To create a *VLAN X* and add the *port Y* :

```
HP ProCurve Switch 6108(config)#vlan X
HP ProCurve Switch 6108(vlan-X)#untagged Y
```

The commands `show vlans` and `show vlan X` list respectively the existing VLANs and their associated ports.

Operation 8 Let associate PC3 to a new VLAN (i.e. not the default VLAN).

Let try a ping from PC3 to PC2. What did you remark? Why?

Operation 9 Assign to PC3 a different IP address (not in the original subnetwork. Let configure a new interface to allow PC2 / PC3 communications (cf. figure 9).

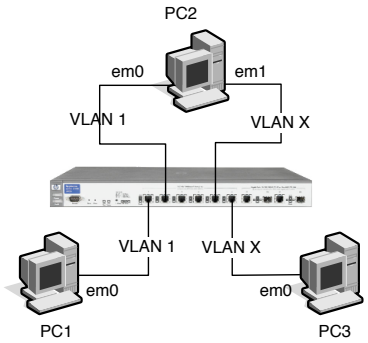

Figure 9: Level 1 VLANs

Let verify that the diffusion domains of the two VLANs are isolated. How do you proceed?

With this approach, a port can only be assigned to a single VLAN. We will now try 802.1Q to tag the frames.

4.2.2 VLANs with tags

We will now introduce the notion of virtual interface. We can associate one or several virtual interfaces to one single physical interface, connected to different VLANs.

We will use two virtual interfaces over em0 on PC2 to receive tagged frames. You must configure a virtual interface by assigning IP parameters, the `vlandev`, the physical interface it will use, the `vlan` (the tag id) (cf. `man ifconfig`). For instance:

```
ifconfig vlan0 create
ifconfig vlan0 vlan 1 vlandev em0
ifconfig vlan0 10.0.0.2/24 up
```

Naturally, the switch configuration must correspond to this installation. For instance, the port of the switch connected to PC2 must be tagged and own to both VLANs. On

the other side, the ports corresponding to PC1 and PC3 will not be tagged but associated to one single VLAN. If you want to configure the *Port Y* to accept frames from *VLAN X*, you must enter the command:

```
HP ProCurve Switch 6108(vlan-X)#tagged Y
```

Operation 10 We want to set up the topology described in figure 10

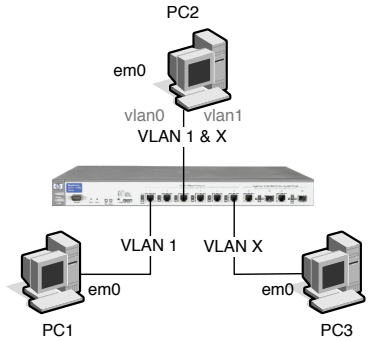

Figure 10: VLANs 802.1Q

Let verify that PC2 can ping PC1 and PC3. Let take a look on the frames with wireshark (on PC1, PC2, and PC3). What do you observe? What is the VLAN field *in the frame?*