ISOTDAQ Networking Lab (Lab 9)

Intro and Guidelines

The lab purpose is to introduce you into the field of networking and data networks for High Energy Physics experiments.

I. Introduction

A computer network or data network is a telecommunications network that allows computers to exchange data. In computer networks, networked computing devices pass data to each other along data connections. The connections (network links) between nodes are established using either cable media or wireless media. The largest and best-known computer network is the Internet.

A switch is a networking device used to connect many devices (e.g. computers) together on a Local Area Network (LAN). A router or a switch with routing features is a networking device used to connect Local Area Networks (LANs) together. A network administrator has limited control over the traffic between devices on the same LAN and a much higher degree of control over the traffic between devices found in different LANs. To avoid this limitation one will usually group devices in LANs by their function, user group or security restrictions then will use routers or advanced switches with routing capability to connect these LANs together.

To ease the network design and improve network efficiency, management and security, most switches today support a logical grouping of physical ports known as a Virtual LAN (VLAN). If 2 devices are connected to the same VLAN they behave as being in the same LAN, and if 2 devices are connected to different VLANs, even if they are connected to the same physical switch, they behave as being connected to different LANs.

Most data acquisition systems have a computer network used to build and filter the collision data recorded by the detector. All the computers in the DAQ network are interconnected using a set of pizza-box switches and one (for small DAQ systems) or several (for more complex DAQ systems) core networking equipment such as advanced switches or routers. The DAQ networks today make extensive use of the VLAN and routing technology to logically group and control the packet flows generated by the readout, collector, filtering or storage DAQ equipment.

To monitor a computer network, protocols such as SNMP (Simple Network Management Protocol), sFlow or NetFlow can be used to gather statistics from networking devices. SNMP is one of the most simple and widely used protocols for network monitoring and configuration. SNMP exposes management data in the form of variables on the managed systems, which in turn describe the system parameters. These variables can then be queried (and sometimes set) by management or monitoring applications. In a typical SNMP use-case an administrative computer has the task of monitoring the traffic load of a group of hosts or devices on a computer network. Each managed system (also called a slave) executes a software component called an *agent*, which reports information via SNMP to the managing systems (also called masters).

The information gathered via the SNMP protocol (ex: number of MB/s going in or out from one port, errors, discards, interface speed etc.) can be stored in a standard database (like Oracle, MSSQL, MySQL etc.) or in RRD (Round Robin Database) files. To create, store and retrieve data from an RRD file the rrdtool application and library can be used.

Traditional networks are sometimes difficult or cumbersome to optimize for DAQ applications, which have

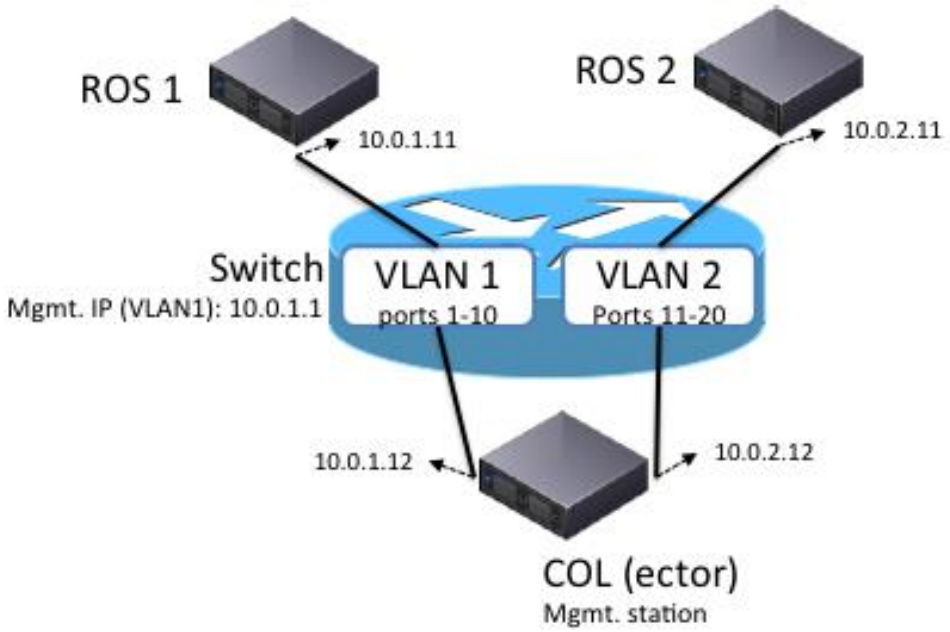
a clearly pre-defined traffic pattern and generally require high throughput. Software Defined Networking is a new paradigm for networking, which gives a user complete flexibility on how to forward flows through a network, using the OpenFlow protocol. Though not fully mature, the technology has great potential for DAQ, as it offers the support for implementing fully customizable multi-path high throughput networks.

II. Objectives

To simulate a very basic DAQ network we will use 2 computers (ROS1 and ROS2) as readout systems, a switch with management capabilities as the DAQ network and another computer (COL) to act as a data builder/collector.

In the first part you will have to configure the DAQ network (i.e. the switch) to be able to manage and monitor it remotely. Then you will configure 2 VLANs and map the existing ports to one or the other accordingly to the setup described in the following section. Once the DAQ switch is up and running you'll configure the monitoring scripts to match the existing configuration and will use them to monitor the traffic load on the connected ports. In the end you'll configure VLAN 2 to act as an SDN switch and preconfigure a set of rules to allow traffic to flow between the ROS2 and COL computers. If there is still time left you can also configure inter-vlan routing between VLAN 1 and VLAN 2.

III. Setup



IV. Step-by-step Guideline

1. Basic Configuration

a) Power up the switch. Connect the cables between the computers and the switch. Check the connectivity light. (It doesn't matter what port numbers you chose to use for connecting the PCs);

b) Login to the COL PC (user: student, password: student);

c) Connect the serial cable (DB9-DB9) from the serial port on the switch to the serial port on the NETMON computer;

d) Start the "screen" application to connect to the switch:

   i. student> **sudo screen /dev/ttyS0**

   ii. sw-daq> **enable**

   iii. sw-daq# **show running-config**

e) Check that the switch has the ip address set to 10.0.1.1 (netmask 255.255.255.0). If the ip is not set please set it:

   i. sw-daq# **configure terminal**

   ii. sw-daq(config)# **vlan 1**

   iii. sw-daq(vlan-1)# **ip address 10.0.1.1 255.255.255.0**

f) Have a look at the VLANs already configured on the switch (show running config). Are the connected ports all in the same vlan? Is it important to have all the ports in the same vlan? If necessary please make adjustments to where the cables are connected;

g) Try to ping the switch;

   i. student> **ping 10.0.1.1**

h) Try to request some simple information from the switch via SNMP:

   i. student> **snmpget –v 2c –c public 10.0.1.1 sysDescr.0**

i) Use wireshark to have a detailed look at the network traffic;

   i. student> **sudo su –**

   ii. student$ **wireshark**

Decoding for the SNMP OIDs related to traffic information

.1.3.6.1.2.1.2.2.1.10 = .iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).interfaces(2).ifTable(2).ifEntry(1).ifInOctets(10)

i. student> **snmptranslate –Of .1.3.6.1.2.1.2.2.1.10**

ii. student> **snmptranslate .1.3.6.1.2.1.2.2.1.10**

.1.3.6.1.2.1.2.2.1.16 = .iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).interfaces(2).ifTable(2).ifEntry(1).ifOutOctets(16)

i. student> **snmptranslate –Of .1.3.6.1.2.1.2.2.1.16**

ii. student> **snmptranslate .1.3.6.1.2.1.2.2.1.16**

2. VLAN Configuration

i. sw-daq (config)# **vlan 2**

ii. sw-daq (vlan-2)# **name Vlan2**

iii. sw-daq (vlan-2)# **untagged 11-20**

3. Network Monitoring

a) Have a look at the following files located in the swmon folder:

iii. student> **cd ~/swmon**

iv. **switch_stats_create.sh, switch_stats.sh, switch_graph.sh**.

b) Adjust to previous files to match your switch IP;

c) Run switch_stats_create.sh and check for newly created switch_stats.rrd file:

i. student> **./switch_stats_create.sh**

d) Run switch_stats.sh to start polling the switch:

i. student> **./switch_stats.sh &**

e) Open another console and run switch_graph.sh in to start generating traffic plots.

i. student> **./switch_graph.sh &**

f) Open index.html file in a browser (e.g. firefox)

g) Modify **switch_stats_create.sh, switch_stats.sh, switch_graph.sh** to start monitoring only on the active ports (the one connected to the event building farm and the event injector)

4. SDN/Openflow Configuration

The goal in this part is to set VLAN2 in OpenFlow mode and be able to communicate through it (between ROS 2 and COL, on the 10.0.2.0/24 interfaces).

a) Check the mac-address forwarding table on the swithch

i. sw-daq# **show mac-address**

b) Enable OpenFlow for VLAN2

i. sw-daq# **conf t**

ii. sw-daq(config)# **openflow**

iii. sw-daq(openflow)# **enable**

iv. sw-daq(openflow)# **instance vlan2**

v. sw-daq(of-inst-vlan2)# **connection-interruption-mode fail-secure**

vi. sw-daq(of-inst-vlan2)# **listen-port 6633**

vii. sw-daq(of-inst-vlan2)# **member vlan 2**

viii. sw-daq(of-inst-vlan2)# **enable**

c) Verify that the OF instance is active (you can talk to it)

i. student> **dpctl show tcp:10.0.1.1**

d) Check its flow table (there should be no flows), and make shure you cannot ping ROS2 from COL:
   i. student> **dpctl dump-flows tcp:10.0.1.1**
   ii. student@ROS2> **ping 10.0.2.12**

e) Add the simplest flows that enable the communication between the two hosts, and check that it works:
   i. student> **COL_P="11"** # match to your cabling
   ii. student> **ROS2_P="12"** # match to your cabling
   iii. student> **dpctl add-flow tcp:10.0.1.1 in_port=$COL_P,actions=output:$ROS2_P**
   iv. student> **dpctl add-flow tcp:10.0.1.1 in_port=$ROS2_P,actions=output:$COL_P**

   This will forward packets coming at the COL port to ROS 2 and vice-verca. Verify by checking the flow-table
   v. student> **dpctl dump-flows tcp:10.0.1.1**

   Verify that ROS2 can now ping COL:
   vi. student@ROS2> **ping 10.0.2.12**

f) Clear the flows:
   i. student> **dpctl del-flows tcp:10.0.1.1**

g) Add rules that don't rely on the input port, but on the destination mac address (*dl_dst*, see dpctl appendix, or simply run “man dpctl”). You will need to add 3 rules to enable the communication:
   - one rule for broadcast traffic, with action "flood"; will be used for ARP resolution
   - one rule to match the dl_dst of ROS2 and forward to its corresponding port
   - one rule to match the dl_dst of COL and forward to its corresponding port

   You can have a look at **precise_flows.sh** if you don't find the solution fast enough.

5. Inter-VLAN Routing (extra)

The aim of this part is to understand how to talk between two LANs. Both VLANs will be configured in standard mode (non-OpenFlow), and you will have to make ROS 1 talk to ROS 2 through the switch. For this you will need to configure a gateway on each host.

a) Rest vlan2 to normal mode (non-openflow) and check that you can ping COL from ROS2
   i. sw-daq# **conf t**
   ii. sw-daq(config)# **openflow**
   iii. sw-daq(openflow)# **disable**
   iv. sw-daq(openflow)# **no instance vlan2**

b) Configure an ip address on vlan 2, and make sure you can ping it
   i. sw-daq(config)# **vlan 2**
   ii. sw-daq(vlan-2)# **ip address 10.0.2.1 255.255.255.0**
   iii. sw-daq(vlan-2)# **show ip**
   iv. student@COL> **ping 10.0.2.1**

c) Try to ping ROS 2 from ROS 1. It will not work, because ROS-1 will try send the ping through its default gateway, since it has no interface directly connected to 10.0.1.0/24 (check the routing table with **ip route** or **route**)

   i. student@ROS-1> **ping 10.0.2.11**

d) Configure gateways on each host such that the hosts know they can reach each other through the switch (sw-daq), and check that ping works

   i. student@ROS-1> **route add -net 10.0.2.0 netmask 255.255.255.0 gw 10.0.1.1**

   ii. student@ROS-1> **route** # or **ip route**, to check the routing table on the host

   iii. student@ROS-2> **route add -net 10.0.1.0 netmask 255.255.255.0 gw 10.0.2.1**

   iv. student@ROS-2> **route** # or **ip route**, to check the routing table on the host

   Ping should work now:

   v. student@ROS-1> **ping 10.0.2.11**

VI. Questions[1]

- What is an HOST?
- What is an IP address?
- What is a MAC address?
- What is the difference between a L2 Switch and a Router?
- What is the difference between a L3 switch and a Router?
- What is the purpose of the ARP protocol?
- What happen if you have a static entry in the ARP cache and the NIC for that target computer is changed?
- If IP determines that the packet that it is currently processing is destined for a remote subnet, where does IP send the packet?
- How could you find the physical address of the Ethernet card installed on your computer?
- What is the purpose of the TTL field in the IP frame ?
- You are the network administrator of a Class C network. Your network consists of 100 computers. Your ISP assigns the address 137.138.111.0/24 to your network. Your network requires 10 subnets with at least 10 hosts per subnet. Which subnet mask should you configure to meet this requirement?
- What is the dotted decimal notation of subnet masks for the following ip addresses?
    - 192.168.10.1/23
    - 5.5.5.5/16
    - 203.40.21.58/27
    - 9.2.3.1/9
- What is the prefix notation of the following subnet masks?
    - 255.255.0.0
    - 255.248.0.0
    - 255.255.255.255
- IP Fragmentation. Using wireshark start a new capture. Ping another host using a 2900 as packet size. Stop the capture and view the captured frames. What do you notice?
- Can you think of some disadvantages of OpenFlow / SDN? How many flow entries do you think a typical hardware switch implementation currently supports?

[1] You will not be able to answer the questions based on the material from the ISOTDAQ school only. You will need to dig for information yourself (Google is your friend). This is just a list of things you should look for and understand, if you are interested in networking.

ANNEX 1 - SDN

1. ***dpctl*** *command list (openflow)*

show SWITCH	show basic information
status SWITCH [KEY]	report statistics (about KEY) (Not on HP)
show-protostat SWITCH	report protocol statistics (Not on HP)
dump-desc SWITCH	print switch description
dump-tables SWITCH	print table stats
mod-port SWITCH IFACE ACT	modify port behavior
dump-ports SWITCH [PORT]	print port statistics
desc SWITCH STRING	set switch description
dump-flows SWITCH	print all flow entries
dump-flows SWITCH FLOW	print matching FLOWs
dump-aggregate SWITCH	print aggregate flow statistics
dump-aggregate SWITCH FLOW	print aggregate stats for FLOWs
add-flow SWITCH FLOW	add flow described by FLOW
add-flows SWITCH FILE	add flows from FILE
mod-flows SWITCH FLOW	modify actions of matching FLOWs
del-flows SWITCH [FLOW]	delete matching FLOWs
monitor SWITCH	print packets received from SWITCH
execute SWITCH CMD [ARG...]	execute CMD with ARGS on SWITCH

2. ***dpctl*** *flow fields and syntax*

- in_port=port_no
- dl_vlan=vlanID
- dl_src=mac
- dl_dst=mac
- dl_type=ethertype
- nw_src, nw_dst=ip[/netmask]
- tp_src, tp_dst=port

3. ***dpctl*** *examples*

- dpctl dump-flows tcp:10.0.1.1:6633
- dpctl dump-ports tcp: 10.0.1.1:6633
- dpctl add-flow tcp: 10.0.1.1:6633 in_port=10,actions=output:14
- dpctl add-flow tcp:10.0.1.1:6633 arp,actions=NORMAL

ANNEX 2 – HOST CONFIGURATION

Network configuration at runtime

- /sbin/ifconfig
- /sbin/route

Persistent network configuration

- /etc/sysconfig/network-scripts/ifcfg-eth*
- /etc/sysconfig/network
- /etc/resolv.conf

Reset network configuration to default

- /sbin/service network restart