TECHNISCHE UNIVERSITÄT BERLIN
Fakultät IV – Elektrotechnik und Informatik
FG INET
Prof. Dr. Thomas Zinner
Theresa Enghardt, Apoorv Shukla, Damien Foucard, Thomas Krenc

Routerlab SoSe 2018
Worksheet 3: Workload Generation and Monitoring using Netflow/SNMP

The World Wide Web (WWW) is one of the most essential applications of today's Internet, with millions of users generating requests and Web servers responding to them. Many testbed environments involve generating Web traffic that has similar characteristics as those observed in the real Internet. Examples of such characteristics are requests inter-arrival times, distribution of data (file) transfer sizes, number of concurrent sessions, etc. The process of artificially generating Web traffic is known as Web workload generation. The purpose of this worksheet is twofold: First, you will learn how to generate web workloads using different tools. Second, you will learn how to monitor traffic using *flow summary statistics*.

Broadly speaking a flow is *"an aggregation of similar packets that are close in time"*. One –very typical– way to aggregate packets to a flow is to use a 5-tuple that is composed by the source and destination IP addresses, the port numbers and by the transport protocol (SRCIP, SRCPORT,DSTIP,DSTPORT, PROTOCOL). Note that, with this definition, flows are uni-directional! A flow is said to end when there has not been a matching packet for a certain inactivity timeout (usually 15 seconds) or if, in the case of TCP, a FIN or a RST is observed. Flows have associated flow-level statistics such as the begin and end timestamps, or the number of bytes and packets between others. Some routers and switches implement protocols to export these flow-level statistics. You will measure such statistics for traffic you will generate using *Netflow Version 5* on a Cisco router. Furthermore, you will learn how to monitor the network using other tools such as SNMP.

Question 1: (10 Points) *Basic Setup Configuration*

In this question, you will build the basic setup for the generation of web workload.

(a) Submit the diagram shown in Figure 1 annotated with your VLAN and IP address assignments to the corresponding interfaces. Note that there should be no overlap in the IP address ranges used in the different VLANs. In the following, the Web client, the monitor/collector or the Web server, denote the corresponding load generators displayed on the topology, namely $lg2$, $lg3$ and $lg4$, respectively.

(b) Show us the output of the traceroute command from the Web client (targeting the Web server) and provide us the most relevant parts of your device configurations.

Question 2: (20 Points) *Workload Generation with iperf*

Frequently, it is desired to test or measure the performance of a link or path over which packets are sent. In this section we will experiment with one of the tools that allows to perform throughput measurements: *iperf*.

(a) Install *iperf* using `apt-get install iperf` on both the Web client and the Web server[1]. Read the `man` page of *iperf* and find out how it can be used in client and server mode. First start one *iperf* process as server on the Web client and then start another *iperf* process as client on the same loadgen. Submit a table containing, for at least five iterations, the following values:

- TCP throughput/bandwidth
- Maximum segment size (MSS)
- TCP window size

(b) Now start one *iperf* process on the Web client as client and one *iperf* process on the Web server as server. Submit a table, using the same format as in the previous section, containing the values for at least five iterations.

[1] if you get some errors while installing iperf/iptraf, ignore them

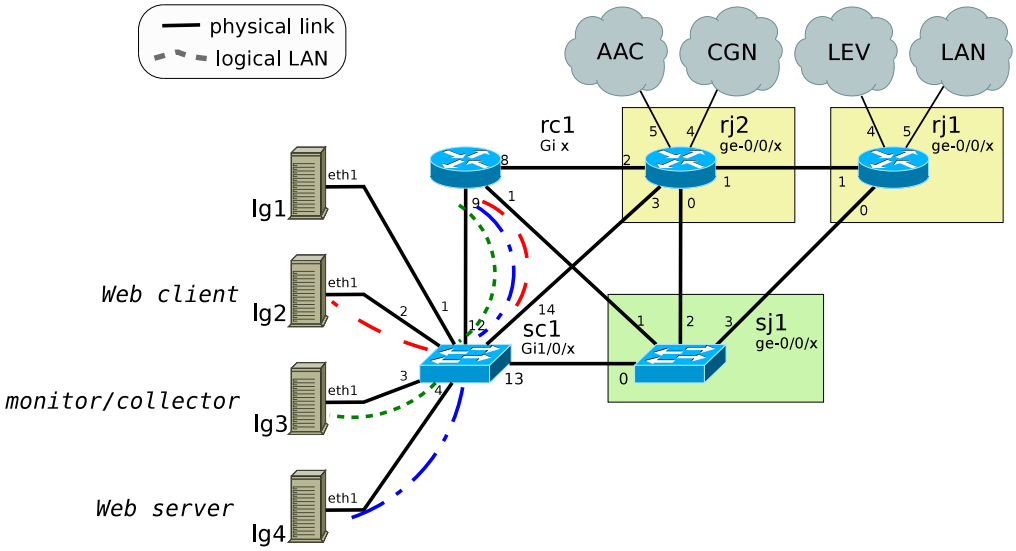

Figure 1: Topology – Basic Configuration for Web Workload Generation

(c) Did you observe differences in the values that you obtain in questions *2a* and *2b*? Enumerate the parameters that have changed. What are the possible factors which are causing these changes?

(d) *iperf* can perform bidirectional bandwidth measurements. Perform one of such measurements and report the results. Can you reason about a situation where a bidirectional measurement is inadequate? Also, *iperf* features an option to disable Nagle's algorithm. In what scenarios would you want to use this option?

Question 3: (30 Points) *Web workload generation with Harpoon*

Harpoon is a tool to another traffic generator that can be used to generate more flexible traffic workloads. Take a look into the Harpoon manual[2].

(a) Why do you think some people in our group believe that *Harpoon* is more realistic than *iperf*? Motivate your explanation.

(b) Find out what options are available for *Harpoon*. What kinds of distributional models exist for recreating TCP and UDP workloads? Briefly describe each model.

(c) Before learning how to generate traffic in great volume you will have to download the Harpoon sources and build them using the following commands:
wget --no-check-certificate
https://master.routerlab/labcourse/harpoon-master.tar
wget --no-check-certificate
https://master.routerlab/labcourse/harpoon-profiles.tar

tar -xvf harpoon-master.tar
tar -xvf harpoon-profiles.tar
apt-get update
apt-get install g++ gcc make automake libexpat1-dev flow-tools-dev

cd harpoon-master; ./configure
make; make install

[2]https://github.com/jsommers/harpoon/blob/master/README

You should now find the harpoon binary and the run_harpoon.sh at */usr/local/harpoon* on your loadgen image. You will use the traffic profiles you downloaded from our server, which include two files: *web-client.xml* and *web-server.xml*. Have a look at these files carefully and explain in short the meaning of "active sessions", "file sizes" and "interconnection times".

(d) To generate traffic with Harpoon with the default traffic profile you will have to modify the following lines at the end of "web-client.xml" file:

```
<address_pool name="client_source_pool">
    <address ipv4="x.x.x.x/32" port="0" />
</address_pool>

<address_pool name="client_destination_pool">
    <address ipv4="y.y.y.y/32" port="10000" />
</address_pool>
```

Here x.x.x.x/32 is the web client IP address and y.y.y.y/32 is the web server IP address. Modify these values according to your topology. Note that you do not have to modify the "web-server.xml" file. Next, start the Harpoon server process on the web server groupX-lg4 using the following command:

```
./run_harpoon.sh -f ~/web-server.xml -v 10 -w 300
```

You can stop any Harpoon process using ctrl-c. Run the command *netstat -an* to verify that your web server is in 'Listening mode' and provide its output.

Next, you can start the Harpoon client process on groupX-lg2 with the following command:

```
./run_harpoon.sh -f ~/web-client.xml -v 10 -w 300  -c
```

You will have now to find out how much bandwidth is currently being used. There are many tools available for this e.g., *mrtg*, *iptraf*. In this worksheet we will use only *iptraf*. Install *iptraf* on your loadgens using apt-get install iptraf. Read the man page and find out how it can be used. Report how much traffic is being generated in both directions.

Now use the following commands on rc1 and sc1 to monitor traffic.

```
show interfaces <interface> | include bits
```

By default interface statistics are updated every 5 minutes. You should change this interval to 30 seconds on all interfaces that you are monitoring using the following command in interface configuration mode.

```
load-interval 30
```

(e) Provide the Web traffic data rate (per direction) in your network when you run Harpoon with the given profiles. Specifically, provide a **cropped** screenshot of *iptraf* and the **relevant** output for the above mentioned command for one of your routers or switches.

(f) Find out how different Web traffic data rates can be generated by reading the Harpoon documentation. Try different configurations[3] and report for each of them the throughput achieved. Provide the parameters of the "web-client.xml" file[4].

Question 4: (20 Points) *Configuring Netflow on Cisco Router*

At this point of the course you should be already familiar with tcpdump for packet-level capturing. However, there are many situations in which packet-level capturing is not feasible. In this exercise you will learn how to capture aggregated data from routers in the form of flow-level statistics.

(a) Configure *Netflow Version 5* on rc1 with the following commands. You will have first to configure the IP address 100.100.100.100/32 on the the loopback interface "Loopback0". Next, choose port 7777 as the destination port for exporting flows to the flow-collector loadgen.

- At global configuration mode:

  ```
  ip flow-export source Loopback0
  ip flow-export destination <destination_ip>  <destination_port>
  ```

- At the interface configuration mode on the (sub-)interface from where you want to get flows:

  ```
  ip flow ingress
  ```

  To verify that flows are exported correctly, use the command sh ip flow export. Provide us its output.

[3] You need to modify web-client.xml only

[4] Submit **only** the parameters that change, not the whole file

(b) You now need a tool that can capture flow data. For this purpose you can use *flow-tools*[5], which is a software package for collecting and processing NetFlow data from Cisco and Juniper routers. For this worksheet you need only *flow-capture* for collecting flows from rc1 router and *flow-report*, which generates reports from flow data. Install *flow-tools* using apt-get install flow-tools on the monitor/collector. Read the man pages of different flow-tools utilities and capture the flows in */tmp/flows* directory of the monitor/collector for at least five minutes of Web traffic using the Harpoon web workload generator as in Question 3. Now generate a flow report[6] from these files. Explain what you see in the flow report. Submit the output of flow-report along with the commands used for generating flow-report.

(c) Submit graph (no text files) for 'Packets per flow distribution', 'Octets/Bytes per flow distribution' and 'Flow Time distribution' from the data you obtain in the flow-report with

1. default Harpoon configuration.
2. Harpoon configuration with maximum throughput obtained in Question 3f.

Question 5: (20 Points) *Network Management with SNMP*

In this section you will learn how to get detailed information from the routers and switches using Simple Network Management Protocol (SNMP).

SNMP is used extensively to exchange management information between components in operational networks. The communication model for SNMP is based on the client-server paradigm. The following parties participate in the protocol: One *Manager* (or *Management Station*) and several *Management Agents*, which are located inside the monitored component (also called *Managed Node*). The Manager is a client and requests information using the *get-request* from the Agent (the server). In addition, it can also transfer information there (*set-request*). The information exchanges are "protected" using a simple shared secret, the *community string*, and they are structured based on an information model: the *Structure of Managed Information, SMI*. According to the SMI, the agents in the managed nodes have a characteristic *Management Information Base (MIB)*, a structure of variables. The Agent-MIB represents a subtree of the standardized Internet-MIB tree.

(a) Enable SNMP on the rc1 router with the following command (read-only access):

```
snmp-server community <your_secret_community_string> RO
```

(b) Install on groupX-lg3 the `snmp` package, which contains the (`snmpget`, `snmpset` and `snmpwalk`) binaries. These are useful tools to perform simple management tasks involving SNMP. You can find the basic invocation syntax in the `man` pages. Use the option `-v1` to avoid conflicts with different versions of SNMP, and use `-Of` to always output full OIDs.

Before you continue, you have to also install the `snmp-mibs-downloader` package, execute the `download-mibs` command and comment the uncommented lines in file `/etc/snmp/snmp.conf`. Now request the value of the Agent MIB object with the OID `iso.org.dod.internet.mgmt.mib-2.system.sysUpTime.0` and show us the output. Note that you will use an existing interface IP address of rc1 in this `snmpget` request. Please make sure you have the `snmpd` (`snmp` daemon) installed. If `snmpd` is not installed, install it with `sudo apt-get install snmpd`. Run `snmpd` in the background with `service snmpd start` and make sure the daemon is indeed running in the background by verifying that the output of `ps -eaf | grep snmpd` is not empty.

(c) SNMP is based on UDP. What consequences does this have for different error situations? E.g., what behaviour did you observe when there was no SNMP agent running on the router? What happens in case of packet loss? What happens if you pass a wrong community string?

(d) Analyze the SNMP packets using `wireshark` or `tshark`. What pieces of information are transmitted in the packets and which protocol fields are present?
How does the OID look? Does `snmpget` show it in the same way? Can you also provide the numeric OID to `snmpget`?

Include your tracefile with your answer.

(e) Set the router MIB variables for *system contact* and *system location* to arbitrary values.

Do an `snmpwalk` on `iso.org.dod.internet.mgmt.mib-2.system`.

Capture the packet exchange, like in exercise (d). What types of requests do you see now? The same ones as before? Explain how *snmpwalk* works.

[5] (https://linux.die.net/man/1/flow-tools)
[6] Use flow-cat along with flow-report

Submission details (more in ISIS):
Please submit an archive (.tar.gz or .zip) containing a *directory*, which contains all files you want to submit. Please have *your group number* in the file name and the directory name.
A report (one single PDF file, named *worksheet(num)-group(num).pdf*) containing the following elements is mandatory:

- Your group number on the first page
- Topology map with relevant routers, switches, *loadgens*, and interfaces, IPs and subnet masks (CIDR).
- For each question, the written answers with the **relevant** portions of output from all commands such as ping, tcpdump, etc in a text format. **No** screenshots of terminal windows are accepted. For ping 3-4 lines of ping requests are usually sufficient.
- For each question all commands needed to configure the *loadgens*.
- For each question all **changed parts** in the configuration of routers and switches (differences to the default config).
- **Never** include the full verbatim switch or router configuration in the pdf report.
- For all questions, state your assumptions, say what you did, describe what you observed, explain your conclusions.

Additionally, please include your config files in the archive.
For each question, please provide the full switch and router configuration in a separate text file named after the device and question, e.g.: *q01-config-sc1.txt*. This makes it easier for us to reproduce your configuration and understand what you did.
We can only grade what we find in your submission and what we understand. Please state your assumptions and observations as clearly as possible.
Due Date: Thursday, May 17th, 2018 (11:55 PM)