Multilayer Software Switch

Assignment

Design and implement a **software multilayer switch** based on the knowledge obtained in the course on Computer and Communication Networks. As a final implementation, a solution with a **two-port switch** (two network cards, port 1 and port 2) is sufficient, while controlling the network interfaces with the appropriate packet controllers. Design and implement the switch in **C++** or **C#** (other allowed languages are Java or Python). Design a switch to meet the requirements of **tasks 1-5**.

Task 1: Media Access Control Table

The solution has to display the MAC table using the following format *MAC address - port number – the lifetime of the record in seconds*. The switch learns the content of its **MAC table** continuously and **displays** the current status via a **graphical user interface** (the content has to be updated automatically, not with a button). **Clear the content** of the MAC table **with a button**. Create the **configurable timer** for the MAC record expiration (Note: don't forget to ensure the correct implementation of the cable pull-out as well as the cable switching between two ports).

Task 2: Statistics

The solution provides RM OSI layer 2-4 PDUs counters (received/sent) on each port in both IN and OUT directions, which clearly shows the correct operation of the switch. Implement a reset button for statistics. The statics has to provide counters for at least Ethernet II, ARP, IP, TCP, UDP, ICMP, and HTTP.

Task 3: Access Control Lists

The solution filters communication on 2.-4. RM OSI layer including **transport layer ports** and **ICMP types** (usage of the built-in PCAP filtering functions is not allowed). Design the solution as **a list of rules** evaluated sequentially so that any combination of filters can be implemented simultaneously, e.g., enable only HTTP communication for the given IP and at the same time disable "ping" for the given MAC. Allow **a combination** of source and destination MAC and IP addresses, or ports. **Display a table** of **specified rules** and allow them to be deleted individually. Differentiate the filters **in the "in/out" direction** on each port of the switch, e.g., Host A will not reach the web (HTTP), but the nginx (HTTP) server running on it will be available.

Task 4: CDP or Syslog

Implement one of the following functionalities (or another one after agreement with the instructor - the change must be approved by the lecturer before the start of the 3rd lab):

Variant A: Cisco Discovery Protocol (CDP)

Implement the Cisco Discovery Protocols by fulfilling the following requirements. The solution:

1. Clearly shows records for each neighbour: **remote hostname - local port - remote port**.
2. Ensure **the expiration of the records** about neighbours (timeout).
3. Support **multiple neighbours** on a single port (segment).
4. Ensure **compatibility with Cisco devices** (it recognizes the implemented solution as a neighbour).
5. **Enable start/stop** CDP functionality on the switch.

Variant B: System Logging (Syslog)

Implement your own Syslog client (without existing external library) fulfilling the following requirements:

1. Implement **at least 3** different severity levels. You can check Cisco online Guides for your cases.
2. Allow the switch to configure **the source IP address** from which messages will be sent.
3. Allow the switch to **configure the IP address** of the remote syslog **server.**
4. Sent messages must contain **timestamps**.
5. Choose **at least 5 activities** (with descriptions) that are recorded by the system using Syslog (e.g., "Mac X device moved from port 1 to port 2").
6. **Start/stop** Syslog functionality on the switch **must be implemented.**

Utilize a TFTPD32 (or Networkers' Toolkit for GNS3) as the syslog server to verify the solution.

Assignment 5 (instead of assignments 3 and 4) RESTCONF

Implement the retrieval and the modification of the selected attributes of the current configuration of a software switch – **interface status, interface name, device hostname, MAC table timer** – via the REST Application Programming Interface using the HTTP protocol. You can implement either JSON or XML data formats. **Authentication** with a username and password **is also required**. However, it can be defined statically. The students are required to demonstrate the functionality of their solution using any REST client, e.g., Postman. All configuration changes made through RESTCONF **must be** appropriately **displayed** in the switch's user interface.

Students who implement the bonus assignment do not need to solve assignments 3 and 4. However, an **analysis of RESTCONF is required**, which must be submitted **as part of the documentation** for the switch in the 4th week of the semester.

Minimal Requirements

To participate in the final exam, the assignment must have at least the functionality of the switch correctly implemented (a hub is not sufficient), i.e., tasks 1 and 2. Without fulfilling this condition, the student will not be admitted to the final exam.

The Content of the Documentation

The documentation must include:

1. The text of the task assignment.
2. Solution proposal containing detailed frames processing diagrams with a description of what will be done where and how (tasks 1-3).
3. Analysis of CDP or Syslog logs (implementation without sufficient analysis will not be evaluated) if you decide to implement task 4. The document has to include the analysis with the sufficient technical depth.
4. Analysis of the network automation using the RESTCONF (implementation without sufficient analysis will not be evaluated) if you decide to implement task 5. The document has to include the analysis of the RESTCONF and the design of the RESP API interface with the sufficient technical depth.

The documentation as well as the resulting switch must be submitted by the student to the appropriate submission point in AIS (after uploading the files, do not forget to send/upload the files) All deadlines determined by the submission point in AIS are final and the student will be graded 0 points for late submission.

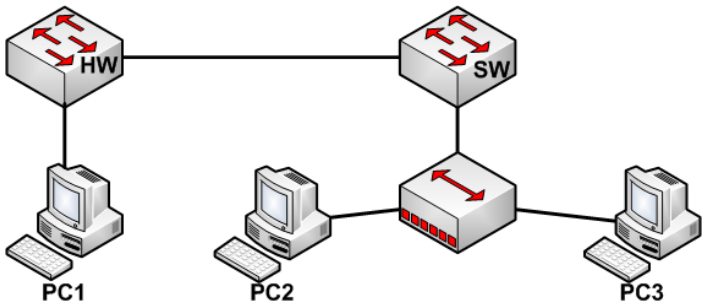
Scoring

The assignment is presented and evaluated continuously in parts, according to the trainer's instructions. For late submission (i.e., the student does not have time to complete the specified part of the assignment by the given exercise/week), the student will be graded 0b for the relevant part of the assignment. Provisional plan for submitting and scoring the assignment:

- 3rd lab (3 points): a prototype that must be able to receive and send communication (capture an incoming frame on a port and send a frame out on a port) + statistics.
- end of the 4th week (2 points + 1 point): documentation (max 2 points for tasks 1-3, 1 point for task 4).
- 7th lab (10 points): minimal functionality of the switch (tasks 1-2).
- end of the 10th week (9 points): filters (4 points) + CDP or Syslog (5 points - only if all conditions specified in the assignment are met, otherwise 0 points).

Assignment Topology

Switch SW represents the computer with your software switch, and HW is the hardware (Cisco) switch.



Note: Implement the switch universally (not exactly on this topology), and test your solution on other topologies as well.

THE END.