

# Softvérový viacvrstvový prepínač

---

## Zadanie

Navrhните a implementujte softvérový viacvrstvový prepínač na základe znalostí získaných z predmetu Počítačové a komunikačné siete (PKS). Pri spracovaní koncepcie návrhu prepínača uvažujte viacportový prepínač – implementovaná aplikačná logika by mala umožniť použitie N portov. Ako výsledná implementácia postačuje riešenie s **dvojportovým prepínačom** (dve sieťové karty, port 1 a port 2), pričom ovládanie sieťových rozhraní realizujete príslušnými paketovými ovládačmi. Prepínač navrhните a implementujte v jazyku **C++** alebo **C#** (ďalšími povolenými jazykmi sú **Java** alebo **Python**). Navrhните prepínač tak, aby spĺňal požiadavky z úloh 1-5.

## Úloha 1: Prepínacia tabuľka

Zobrazoval **prepínicu tabuľku** vo formáte *MAC adresa – číslo portu – aktuálny časovač záznamu*. Prepínač sa obsah svojej prepínacej tabuľky učí priebežne a **aktuálny stav zobrazuje** cez grafické používateľské rozhranie (obsah sa **automaticky** aktualizuje, nie pomocou tlačidla). Umožnite **vyčistiť** prepínicu **tabuľku** pomocou **tlačidla**. Časovač pre vypršanie záznamov nech je konfigurovateľný (pozn.: nezabudnite **ošetriť vytiahnutie** kábla, ako aj výmenu **káblov medzi portami – napr. pomocou monitorovania premávky na základe staticky definovaných MAC / IP adries, spravidla nekonečný ping medzi 2 zariadeniami**).

## Úloha 2: Štatistiky

Poskytoval **štatistické informácie** vrstvy 2-4 RM OSI o počte (prijatých/odoslaných) PDU na každom porte v **smere IN** aj **OUT**, ktoré budú zreteľne zobrazovať správne fungovanie prepínača. Umožnite **resetovať** štatistické informácie prostredníctvom GUI. Štatistické informácie nech zobrazujú minimálne informácie o PDU typu **Ethernet II, ARP, IP, TCP, UDP, ICMP, HTTP**.

## Úloha 3: Filtrácia komunikácie

Filtroval komunikáciu na 2.-4. vrstve RM OSI vrátane **portov transportnej vrstvy** a **typov ICMP** (bez použitia vstavaných PCAP funkcií filtrovania). Riešenie navrhните ako **zoznam pravidiel** vyhodnocovaných sekvenčne tak, aby bolo možné naraz realizovať ľubovoľnú kombináciu filtrov. Napr. pre danú IP povoliť iba HTTP komunikáciu a zároveň pre danú MAC zakázať "ping". Umožnite aj **kombináciu** zdrojových a cieľových MAC a IP adries, príp. portov. **Zobrazujte tabuľku** zadaných **pravidiel** a umožnite ich aj jednotlivo odstraňovať. Filtre rozlišujte **v smere "in/out"** na každom porte prepínača (takisto zohľadniť v návrhu). Napr. Host A sa nedostane von na web (HTTP), ale u neho bežiaci server nginx (HTTP) bude dostupný.

## Úloha 4: CDP alebo Syslog

Realizoval jednu z nasledujúcich funkcionalít (príp. inú po dohode s cvičiacim – zmena musí byť schválená cvičiacim do začiatku 3. cvičenia):

### Variant A: Cisco Discovery Protocol (CDP)

Implementácia protokolu CDP, pričom stačí:

1. Prehľadne ukázať pri každom zázname o susedovi: *remote hostname - local port - remote port*.
2. Lokálne **označenie zariadenia** nech je **konfigurovateľné** prostredníctvom GUI.
3. Zabezpečiť vypršanie **časového limitu pre susedov** (timeout), podporovať **viacerých susedov na 1 porte** (segmente).

4. Zabezpečiť **kompatibilitu s Cisco zariadeniami** (rozpoznať ho ako suseda). Umožnite spustenie/zastavenie CDP funkcionality na prepínači.

### **Variant B: System Logging (Syslog)**

Implementácia Syslog klienta, pričom je potrebné:

1. Zabezpečiť **aspoň 3 úrovne** dôležitosti správ (severity level).
2. Umožniť nakonfigurovať prepínaču **zdrojovú IP adresu**, z ktorej sa budú správy odosielať.
3. **Nakonfigurovať IP adresu** vzdialeného Syslog servera.
4. Zasielané správy musia obsahovať **časovú pečiatku** (angl. timestamp).
5. Zvoľte **aspoň 5 činností** (descriptions), ktoré budete pomocou Syslog zaznamenávať (napr. „Zariadenie s MAC X sa premiestnilo z portu 1 na port 2“).

Syslog server bude aplikácia TFTP32 bežiaci na niektorom počítači (prípadne Networkers' Toolkit pre GNS3). Umožnite spustenie/zastavenie Syslog funkcionality na prepínači.

### **Úloha 5 (namiesto úloh 3 a 4): RESTCONF**

Umožňoval **získať a modifikovať** vybrané atribúty aktuálnej konfigurácie softvérového prepínača – stav rozhrania, názov rozhrania, hostname zariadenia, časovač MAC tabuľky – prostredníctvom REST API využitím protokolu HTTP. Využiť je možné buď údajový formát JSON, alebo XML. Nutná je tiež autentifikácia používateľským menom a heslom, tie môžu byť staticky definované. Študent musí predviesť funkčnosť svojho riešenia využitím ľubovoľného klienta, napr. nástroj Postman, ktorý umožňuje odosielať HTTP požiadavky. Všetky zmeny konfigurácie vykonané prostredníctvom RESTCONF musia byť vhodne zobrazené v GUI prepínača.

Študenti, ktorí implementujú bonusovú úlohu, nemusia riešiť úlohy 3 a 4. Potrebná je však analýza RESTCONF, ktorú je potrebné odovzdať v rámci dokumentácie k prepínaču v 4. týždni semestra.

### **Podmienky absolvovania**

Pre účasť na skúške je potrebná funkčná implementácia minimálne logiky prepínača (nestačí hub), t.j. úlohy 1 a 2. Bez splnenia tejto podmienky nebude študent pripustený ku skúške.

### **Obsah dokumentácie**

Dokumentácia musí obsahovať:

1. **Zadanie** úlohy.
2. **Návrh riešenia** obsahujúci podrobné **diagramy** spracovania rámcov (flowchart) s opisom čo sa kde a ako bude vykonávať (**úlohy 1-3**).
3. **Analýzu** protokolov **CDP alebo Syslog** (implementácia bez dostatočnej analýzy nebude hodnotená), ak sa rozhodnete implementovať úlohu 4.
4. **Analýzu** sieťového programovania prostredníctvom **RESTCONF** (implementácia bez dostatočnej analýzy nebude hodnotená), ak sa rozhodnete implementovať úlohu 5. Zároveň dokumentácia musí obsahovať analýzu protokolu RESTCONF a návrh REST API, ktoré budú na účely konfigurácie použité.

Dokumentáciu ako aj výsledný prepínač musí študent odovzdať do príslušného miesta odovzdania v AIS (po vložení súborov nezabudnúť súbory odoslať/odovzdať)Všetky termíny určené miestom odovzdania v AIS sú konečné a za neskoré odovzdanie bude študent hodnotený 0b.

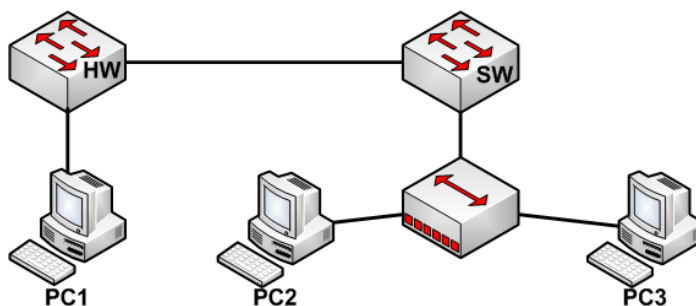
## Hodnotenie zadania

Zadanie sa prezentuje a **hodnotí priebežne po častiach**, podľa pokynov cvičiaceho. Za **oneskorené odovzdanie** (t.j. študent nestihne do daného cvičenia/týždňa vypracovať určenú časť zadania) bude **študent hodnotený 0b** z príslušnej časti zadania. Predbežný plán odovzdávania a bodovania zadania:

- 3. cvičenie (**3b**): **prototyp**, ktorý musí vedieť prijímať a posilať komunikáciu (odchytiť prichádzajúci rámec na porte a poslať rámec von portom) + štatistiky.
- koniec 4. týždňa (**2b + 1b**): **dokumentácia** (max. 2b za úlohy 1-3, 1b za úlohu 4) .
- 7. cvičenie (**10b**): základná funkcionálna **prepínača** (úlohy 1-2).
- koniec 10. týždňa (9b): **filtre (4b) + CDP alebo Syslog (5b)** - len v prípade splnenia všetkých podmienok uvedených v zadaní, inak 0b), alebo **RESTCONF (9b)** – len v prípade splnenia všetkých podmienok uvedených v zadaní, inak 0 b).

## Základná preberacia topológia

Prepínač **SW** predstavuje počítač s vašim **softvérovým prepínačom**, **HW** je **hardvérový** (Cisco) **prepínač**.



*Pozn. prepínač implementujte univerzálne (nie presne na túto topológiu), otestujte sa aj na iných topológiách.*

KONIEC.