

Printer - Simple IP router

PRINTler



PRINTler a simple IP router for PRIN course on WUT

1 Business reqs

PROF

1.1 IP Router

1.1.1 Opis

Routing

User wpisuje do tabeli wpis:

- docelowy adres IP (adres hosta) (to jest kluczem przeszukiwania tabeli)
- na jaki port kierowac ten pakiet (czyli za pomoca, ktorego portu mamy osiagnac dany host docelowy) (to jest parametr akcji)
- adres MAC next-hopa, ktory ma zostac wpisany w pakiet w warstwie ETH (zeby next-hop go nie odrzucil na L2)

TTL

Dodatkowo pakiet IP ma miec odczytywane pole TTL i odrzucane jesli jest ono mniejsze niz 2.

Checksum

Dodatkowo aktualizować ma się sumę kontrolną.

1.1.2

- ☐ Routing
- ☐ TTL
- ☐ checksum

1.1.3 Co to implikuje

Sieć złożona z dwóch hostów oraz switcha. Wszystkie urządzenia mają mieć przypisane statyczne adresy IP oraz MAC. Trzeba to zdefiniować w [1sw_demo.py](#).

checksum

Na poprzednich zajęciach nie musieliśmy tego robić, bo nic nie zmienialiśmy w protokole IP. Hosty będą odrzucać pakiety ze złą sumą kontrolną, więc od tego należy zacząć implementację.

1.1.4 Testy

Routing

Mozna wymyślić topologie oraz jakie wpisy dodać do tabeli, żeby przetestować routing.

TTL

Po odpaleniu Mininet możesz wejść w jego iptables i dodać, żeby wszystkie pakiety IP wysyłały z TTL 1.

```
Mininet CLI> h1 iptables -t mangle -A POSTROUTING -j TTL --ttl-set 2
```

checksum

Jeśli będzie błędna, to host odrzuci.

PROF

1.2 IP Filter

1.2.1 Opis

Użytkownik wpisuje do tabeli wpis, który reprezentuje trójkę, która identyfikuje pakiet, jaki należy odrzucać. Ta trójka to:

- docelowy adres IP
- protokół warstwy transportowej
- port

np. {10.0.0.1, TCP, 80}

ale może też być wildcard, czyli np. {10.0.0.1, TCP, *} to tego będzie Ci potrzebny ten ternary operator.

1.2.2 Reqs

- ☐ Basic filter
- ☐ Wildcards

1.2.3 Co to implikuje

Nic.

1.2.4 Testy

Przez mininet można wejść na hosty przez xterm do ich shell'a i użyć scapy. Na jednym uruchamia się klient a na drugim serwer i można testować connections TCP lub UDP.

Note: scapy będzie trzeba za każdym razem instalować podczas runtime. Albo możesz zmodyfikować skrypt [1sw_demo.py](#)

Alternatywą dla scapy jest [nc](#).

Basic filter

Można wymyśleć wpisy do tabeli i potem scapy'm generować ruch.

Wildcards

Po stronie Trift będzie to samo. Inne tylko testy na scapy.

1.3 Stats

1.3.1 Opis

Router ma zbierać statystyki dotyczące ...(lista poniżej). User za pomocą Trift może sobie je query'ować. Nie wiem jeszcze jakie komendy trift to robią.

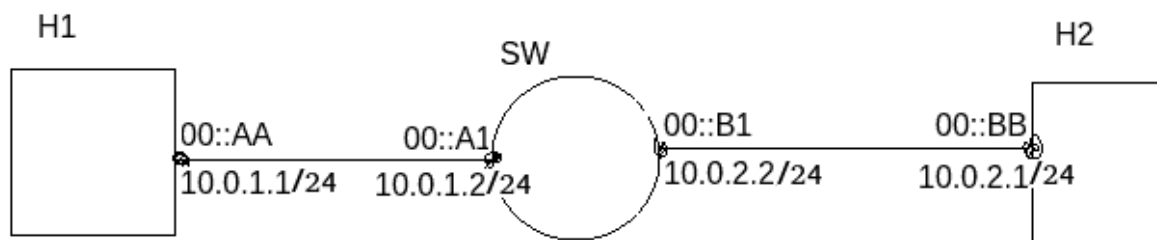
Jakie staty:

- Dla każdego portu liczba pakietów
 - odebranych
 - przesłanych dalej
 - wysłanych
- Dla całego switcha
 - liczba pakietów odrzuconych

Każdy port na warstwie fizycznej najniższej ma Receiver oraz Transceiver. Więc jak mówimy o porcie, to on w danej sytuacji (procesowania single pakietu) pełni rolę albo Tx albo Rx.

Pojęcia odebrać/wysłać pakiet są w odniesieniu do switcha - external world. Więc odebrać może tylko Rx. Wysłać tylko Tx. Pojęcie przesłać dalej jest w obrębie switcha, więc może to zrobić jedynie Rx.

1.3.2 Topologia



Tak należy zmodyfikować plik [1sw_demo.py](#).

Na tym rysunku dodaj maske podsieci /24 do adresów IP bo to sugeruje maske 32 jak nie ma jej.

2 Przebieg cwiczenia

Najpierw należy przygotować topologię według rysunku z sekcji 1.3.2. Polega to na modyfikacji pliku 1sw_demo.py.

```
sudo python3 1sw_demo.py --behavioral-exe=/usr/bin/simple_switch --json template.json
```

Następnie za pomocą Mininet CLI przetestowano czy zaszły w topologii odpowiednie zmiany.