

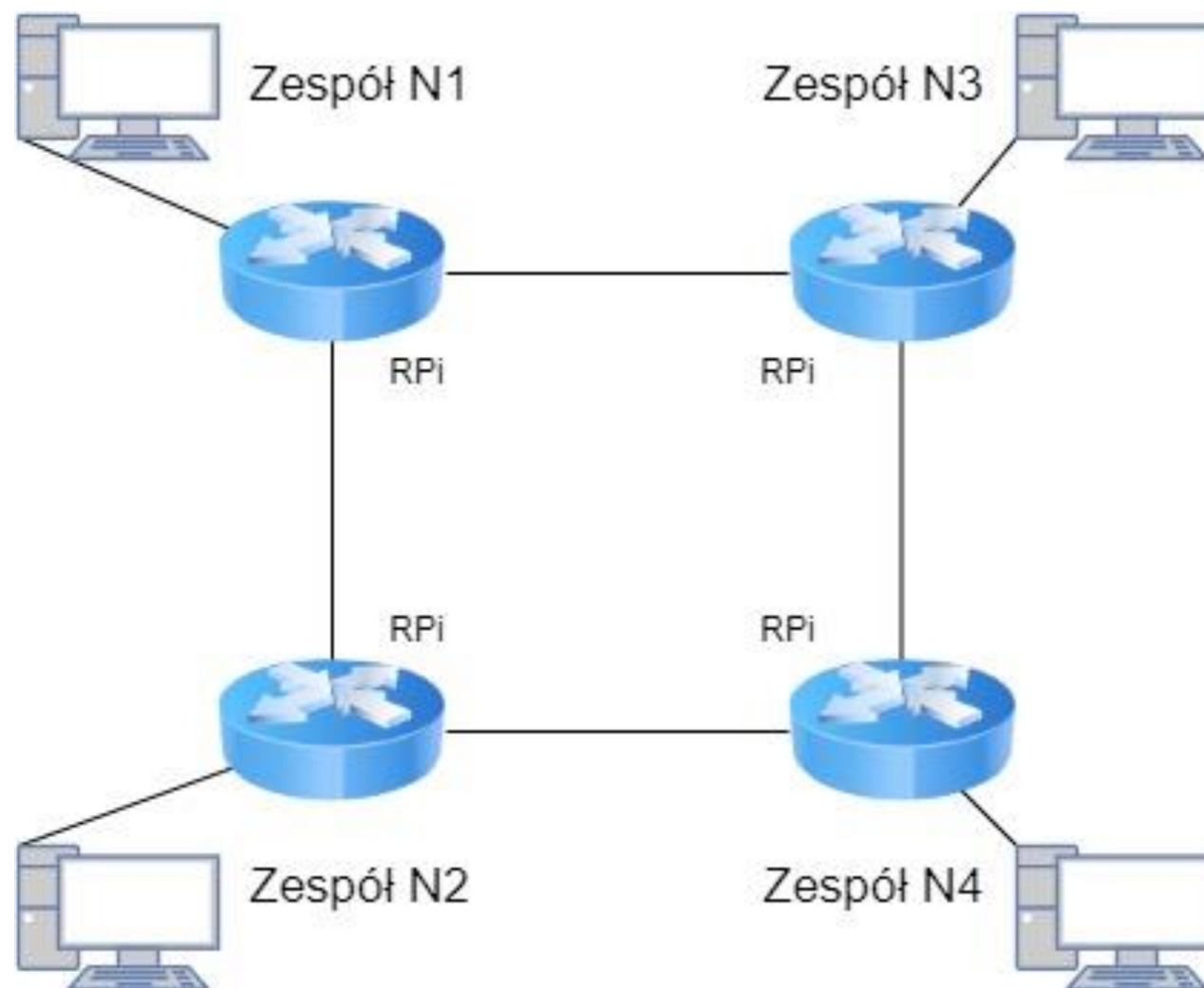
Warsztat 6

mgr inż. Jan Palimąka

Cel projektu

- Stworzenie routera sieciowego z obsługą uproszczonego OSPF
 - Raspberry PI
 - Płaszczyzna przekazu danych zdefiniowana w P4 dla BMv2
 - Płaszczyzna sterowania napisana w Pythonie
- Połączenie routerów różnych implementacji (zespołów) w jedną sieć

Cel projektu



Ocenianie projektu

- Punkty przyznawane na koniec projektu (ostanie zajęcia)
 - Na podstawie kodu źródłowego i działania w praktyce
 - I/lub na podstawie testów PTF (muszą zostać przygotowane)
- Sprawozdanie
 - Kod źródłowy
 - Sposób uruchomienia
 - Wydruk z testów PTF (jeśli będą)

Projekt - płaszczyzna przekazu danych

- Ethernet
- ARP
- IPv4
- OSPF (uproszczony)
- Grupy multicast

Płaszczyzna sterowania na następnych zajęciach

Ethernet

- Prawidłowe adresy MAC nadawcy i odbiorcy
 - W szczególności między routerami
 - Między routerami może znajdować się switch L2
 - Docelowo nie można użyć arbitralnych (własnych) adresów MAC, tylko te zdefiniowane przez kartę sieciową
- MAC Learning
 - Uczenie się adresów MAC

ARP

- Router powinien odpowiadać na zapytania ARP o adresy IP na swoich interfejsach
 - Tabela z kluczem: adres docelowy IP oraz nr portu
- Sposób I
 - Switch wykrywa zapytanie ARP, po czym odsyła **cały** pakiet do sterownika
 - Sterownik uczy się adresu MAC, IP oraz portu na podstawie pakietu
 - Sterownik wysyła do switcha pakiet, który ma zostać odesłany w odpowiedzi
- Sposób II
 - Switch wykrywa zapytanie ARP i parsuje je
 - Dalej switch wysyła informację o nowym adresie MAC do sterownika (MAC learning)
 - Switch konstruuje pakiet, który zostanie odesłany w odpowiedzi
- Wskazówka: operacje w bloku ingress lub egress można przerwać za pomocą `exit` lub `return`

ARP - nagłówek

Ethernet transmission layer (not necessarily accessible to the user):

48.bit: Ethernet address of destination

48.bit: Ethernet address of sender

16.bit: Protocol type = ether_type\$ADDRESS_RESOLUTION

Ethernet packet data:

16.bit: (ar\$hrd) Hardware address space (e.g., Ethernet, Packet Radio Net.)

16.bit: (ar\$pro) Protocol address space. For Ethernet hardware, this is from the set of type fields ether_typ\$<protocol>.

8.bit: (ar\$hln) byte length of each hardware address

8.bit: (ar\$pln) byte length of each protocol address

16.bit: (ar\$op) opcode (ares_op\$REQUEST | ares_op\$REPLY)

nbytes: (ar\$sha) Hardware address of sender of this packet, n from the ar\$hln field.

mbytes: (ar\$spa) Protocol address of sender of this packet, m from the ar\$pln field.

nbytes: (ar\$tha) Hardware address of target of this packet (if known).

mbytes: (ar\$tpa) Protocol address of target.

- Ethernet ARP protocol: 0x0806
- ar\$hrd = 1 (Ethernet)
- ar\$pro = 0x0800 (dla IPv4)
- ar\$hln = 6 (długość adresu MAC)
- ar\$pln = 4 (długość adresu IPv4)
- ar\$op – rodzaj pakietu ARP, możliwe:
 - ar\$op = 1 (ares_op\$REQUEST)
 - ar\$op = 2 (ares_op\$REPLY)
- ar\$sha – adres MAC nadawcy pakietu
- ar\$spa – adres IP nadawcy pakietu
- ar\$tha – adres MAC odbiorcy pakietu
- ar\$tpa – adres IP odbiorcy pakietu

Źródło: RFC826

Żądanie ARP i odpowiedź ARP to dwa różne pakiety

IPv4

- Tablica routingu
 - Obsługa następnego węzła (next hop)
- Zmniejszanie wartości TTL i odrzucanie pakietów
 - Aktualizacja sumy kontrolnej
- Pakiety, których adres docelowy IP jest na jednym z adresów na danym interfejsie switcha trzeba przekazać do płaszczyzny sterowania
 - W szczególności pakietu protokołu 89 (OSPF)
 - Pakiety ICMP można odrzucać lub przekazać do sterownika

OSPF

- Pakiety OSPF przekazywane do sterownika
 - W programie P4 nie trzeba wprost obsługiwać protokołu OSPF i jego nagłówków
- Wartość pola `protocol` w nagłówku IPv4 dla protokołu OSPF: 89

Port CPU

- Jak korzystać z poziomu P4runtime-shell:
https://github.com/p4lang/p4runtime-shell/blob/main/usage/packet_io.md
- Nagłówki z adnotacjami:
`@controller_header("packet_out")` oraz
`@controller_header("packet_in")` służą do przekazywania metadanych między switchem i sterownikiem
- W parserze nagłówek `packet_out` musi zostać sparsowany (np. na bazie nr interfejsu) - będzie na samym początku
- W deparserze nagłówek `packet_in` musi zostać dodany do pakietu na samym początku

Grupy multicast

- W programie P4 wystarczy wpisać nr grupy do pola `mcast_grp` struktury `standard_metadata`
- Dla P4Runtime-shell jest instrukcja:
<https://github.com/p4lang/p4runtime-shell/blob/main/usage/pre.md>
 - Zamiast funkcji `multicast_group_entry` można klasy `MulticastGroupEntry`
- Przy dodawaniu portu są potrzebne 2 parametry:
 - Nr interfejsu wyjściowego
 - Instancja
 - Krotka (nr portu, instancja) nie może się powtarzać w jednej grupie