

Obsługa 3 protokołów:

- Ethernet
- ARP
- IPv4 (v6 nie trzeba obsługiwać)
- OSPF (uproszczony, w data plane praktycznie nie trzeba tworzyć obsługi tego protokołu)
- obsługa grup multicastowych (jak się wysyła pakiet na wiele portów (tak robi OSPF) - sterownik może wysłać jeden pakiet do switcha, a on wyśle na wszystkie porty)

## Ethernet

MAC Learning, żeby nie trzeba było tego konfigurować ręcznie.

## ARP

Na bazie ARP można też stworzyć MAC Learning, czyli nie tylko Eth frames z protokołem IP, ale też ARP.

Router - czyli Raspberry Pi.

Jeśli coś przyjdzie na port, który ma przypisany inny adres IP, to odrzuć go.

ARP można obsłużyć na dwa sposoby:

Wskazówka: chodzi o sterowanie przepływem return wychodzi z bloku a exit, to exit. Można tworzyć własne bloki i je instancjonować - dopiero w tym momencie objawia się różnica.

## ARP - nagłówek

w pakiecie ETH musi być EtherType -x0806

Pamiętaj o zamianie miejscami.

Jeżeli jakiegos adresu nie znamy w REQUEST to można wpisać tam np. same 1.

---

PROF

## IPv4

Powinna być tablica routingu:

- kierowanie na podstawie adresu IP
- z racji wielu podsieci musi być next-hop

Zmniejszenie wartości TTL i odrzucanie pakietów

Jak dst\_ip jest jednym z naszych, to trzeba dać pakiet do sterownika.

Nie ma wymogu pisania obsługi ICMP - odrzucaj.

## OSPF

Pakiety OSPF przekazuj do sterownika.

OSFP poznasz po 89 w `protocol` w pakiecie IP.

## Port CPU

Przekazywanie z data plane do control plane.

W takim nagłówku są metadane typu na który port pakiet przyszedł, można wysłać od razu jakieś info sterownikowi, żeby miał mniej do parsowania.

Jeżeli przychodzi pakiet z portu CPU, to trzeba sparsować dodatkowo header `packet_out` - bo to pakiet od sterownika.

W deparserze jak coś wysyłasz na port CPU to daj na początek `packet_in`.

## Grupy multicastowe

Żeby zrobić multicast to do pakietu trzeba metadata użyć.