

Warsztat 4

mgr inż. Jan Palimąka

Zadanie – router IP z funkcją filtrowania ruchu

- Kierowanie na podstawie adresu IP
 - Docelowa podsieć (klucz w tabeli) oraz port wyjściowy i następny węzeł (parametry akcji)
 - Obsługa TTL (w tym odrzucanie pakietów)
 - Aktualizacja sumy kontrolnej
- Filtrowanie (blokowanie) ruchu na podstawie:
 - Adres docelowy IP
 - Port docelowy warstwy transportowej
 - Protokół warstwy transportowej
- Hosty w różnych podsieciach
 - Statycznie przypisane adresy MAC lub implementacja obsługi protokołu ARP
 - Podmiana na właściwe adresy MAC
- Brak obsługi protokołu ICMP
- Statystyki
 - Odebrane/wysłane pakiety – oddzielnie dla każdego portu
 - Przesłane pakiety dalej - dla każdego portu
 - Odrzucone pakiety - zbiorczo dla wszystkich portów

Sprawozdanie

- Zamieszczony kod źródłowy
- Wpisywane polecenia (wraz z krótkim opisem), które:
 - Nie były wymienione w instrukcji
 - Należało "odkryć" wg instrukcji
- Termin
 - Grupa poniedziałkowa: 22 kwietnia (do końca dnia)
 - Grupa środowa: 10 kwietnia (do końca dnia)

Zadanie – środowisko

- Skrypty wypracowane na poprzednich zajęciach

Tabela – dopasowanie ternary

- Klucz tabeli składa się z 2 części
 - Wartości klucza (V)
 - Maski (M)
 - Maski i wartość są tej samej długości (mają tyle samo bitów)
- Aby i -ty wpis tabeli mógł być dopasowany do pakietu (P), musi zająć zależność:

$$V_i \& M_i = P \& M_i$$

- Może zostać dopasowanych wiele wpisów w tabeli – wybierany jest ten o najwyższym priorytecie

counter i direct_counter

- Obiekty służące do zliczania bajtów lub pakietów (lub obu tych wartości)
- counter - liczniki trzymane w osobnej tablicy (indeksowanej od 0)
- direct_counter - liczniki trzymane razem ze wpisem w tabeli
- Utworzenie instancji
 - o `counter<index_t>(size, CounterType.???) cnt_instance;`
 - o `direct_counter(CounterType.???) dcnt_instance;`
- `index_t` - typ indeksu (tylko dla `V1MODEL_VERSION >= 20200408`)
- `size` - maksymalna liczba wpisów
- `CounterType` - typ wyliczeniowy, określający rodzaj zliczanych danych
- Zwiększenie wartości licznika:
 - o `cnt_instance.count(index)`
 - o `dcnt_instance.count()`

verify_checksum

- Weryfikuje sumę kontrolną
- W przypadku błędu ustawi pole `checksum_error` na 1 w strukturze `standard_metadata`
- Możliwa do wywołania tylko w bloku `VerifyChecksum`
- `verify_checksum<T, O>(in bool condition, in T data, in O checksum, HashAlgorithm algo);`
 - `condition` - warunek konieczny, aby suma kontrolna była weryfikowana (np. nagłówek został sparsowany)
 - `data` - lista pól
 - `checksum` - aktualna wartość sumy kontrolnej
 - `algo` - algorytm; dla IPv4, TCP, UDP: `csum16`
- Przykłady:
https://github.com/p4lang/p4c/blob/main/testdata/p4_16_samples/checksum-l4-bmv2.p4

update_checksum

- Oblicza sumę kontrolną
- Możliwa do wywołania tylko w bloku ComputeChecksum
- `update_checksum<T, O>(in bool condition,
in T data, inout O checksum,
HashAlgorithm algo);`
 - checksum - pole, w którym zostanie zapisana wartość sumy kontrolnej
- Przykłady - jak dla verify_checksum

Zapisy liczb

$[N_w] L$

- N - liczba bitów, na których liczba ma być zapisana (opcjonalnie)
- L - liczba właściwa
 - Może zawierać znaki podkreślenia (_) - są ignorowane
 - Możliwe prefiksy określające podstawę
 - 0x, 0X - zapis szesnastkowy
 - 0o, 0O - zapis ósemkowy
 - 0d, 0D - zapis dziesiętny (domyślny)
 - 0b, 0B - zapis binarny
- Przykłady: 0, 8w0, 32w0xFFFF_FFFF

Przetwarzanie pakietu po bloku ingress

```
if (a clone primitive action was called) {  
    Create clone(s) of the packet  
}  
if (digest to generate) {  
    Send a digest message to the control plane software  
}  
if (resubmit was called) {  
    Start ingress processing over again for the original packet  
} else if (egress_spec == DROP_PORT) {  
    Drop packet  
} else if (mcast_grp != 0) {  
    Multicast the packet to the output port(s)  
} else {  
    Unicast the packet to the port equal to egress_spec  
}
```

Źródło: https://github.com/p4lang/behavioral-model/blob/main/docs/simple_switch.md

Przetwarzanie pakietu po bloku egress

```
if (a clone primitive action was called) {  
    Create clone(s) of the packet  
}  
if (egress_spec == DROP_PORT) {  
    Drop packet  
} else if (recirculate was called) {  
    Start processing the packet over again  
} else {  
    Send the packet to the port  
}
```

Przydatne materiały

Te same co na poprzednich warsztatach