

## 1. Temat projektu

Tematem projektu było stworzenie load balancera działającego w oparciu o zadaną topologię z zastosowaniem algorytmu Least Connections

Projekt bazuje na kontrolerze Floodlight v1.0, v1.1 działającym zgodnie z protokołem OpenFlow.

Głównym komponentem jest pakiet `src/main/java/pl/agh/edu/pl`. Kilka użytych klas jest podobnych lub pokrywających się z kodem wykorzystywanym podczas laboratorium.

Dodatkowo zaimplementowano następujące klasy:

- Hosts
- Route
- SdnLinkListener

## 2. Opis algorytmu

Gdy w sieci pojawia się pakiet nowego przepływu, kontroler wybiera port w danym switchu, z którego wychodzi najmniej aktywnych połączeń w kierunku pozostałych urządzeń. Operacja ta jest powtarzana na każdym switchu, przez który przejdą pakiety. Następnie trasa dla danego przepływu jest instalowana przez kontroler w przełącznikach poprzez dodawanie odpowiednich wpisów w tablicach przepływu. Statystyki dotyczące aktywnych połączeń są na bieżąco zbierane przez kontroler. Dodatkowo zastosowano mechanizm, który zapobiega powstawaniu pętli w sieci.

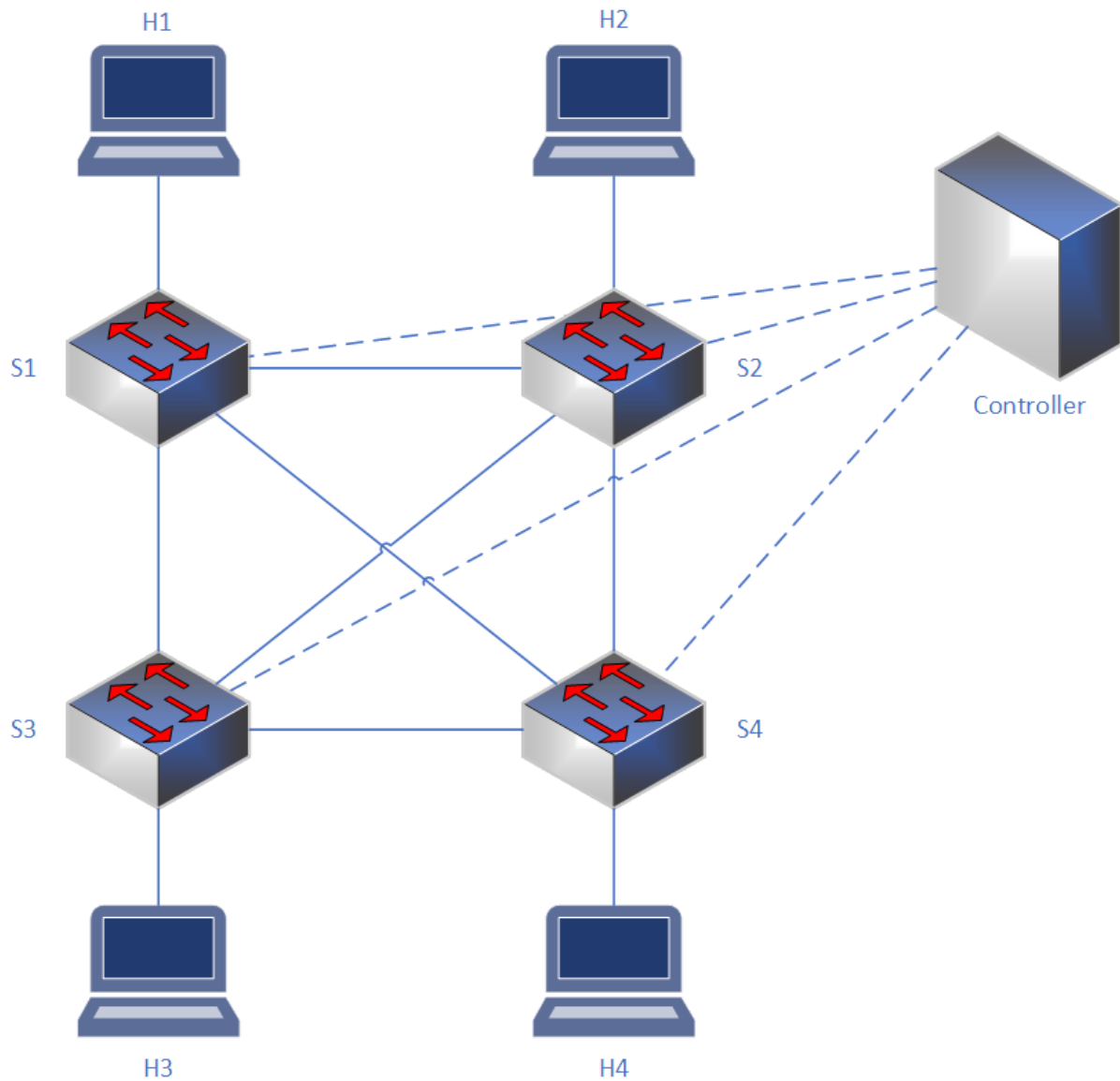
## 3. Psuedokod oraz diagram blokowy

Odpowiednie pliki są dostępne w repozytorium pod nazwami: Pseudokod.pdf oraz DiagramBlokowy.pdf.

## 4. Założenia

- Kolejny przepływ pojawiający się w sieci zostanie przełączony do hosta docelowego poprzez łącza o najmniejszej liczbie aktywnych przepływów
- Algorytm Least Connections został zaimplementowany dla konkretnej topologii i nie jest generyczny
- Działanie algorytmu testowane jest poprzez symulację połączenia wielu instancji klienta programu iperf z serwerem

## 5. Topologia sieci w projekcie



## 6. Wykorzystane narzędzia i generatory ruchu

- Floodlight
- Mininet
- miniNAM
- iperf (serwer + klient)

## 7. Napotkane problemy

- dwa możliwe podejścia (zliczanie połączeń na łączy lub całej ścieżce) - brak możliwości zebrania informacji o całych ścieżkach
- ErrorStatistics Pooling - nie usuwamy switchy po ich odłączeniu

## 8. Wprowadzone uproszczenia w związku z problemami

- Nie musimy sprawdzać który switch jest po drugiej stronie łącza, ponieważ porty są odpowiednio ponumerowane (np. na końcu łącza wychodzącego z portu eth2 w switchu nr 1 jest switch nr 2)
- Jeśli przepływy pojawiają się jednocześnie (w minimalnym odstępie czasowym) to algorytm nie reaguje poprawnie - zakładamy więc że kolejne przepływy przychodzą w odstępie 3-sekundowym

## 9. Sposób uruchamiania:

- Uruchomienie kontrolera Floodlight (w sposób przedstawiony na laboratorium nr 2)
- Uruchomienie programu mininet z podaniem adresu IP uruchomionego wcześniej kontrolera (w sposób przedstawiony na laboratorium nr 2)
- Uruchomienie programu MiniNAM z plikiem topologii:  
`sudo python2 MiniNAM/MiniNAM.py --custom=topology.py --topo=mytopo  
--controller=remote`
- Uruchomienie konsoli xterm na hoście, do którego będzie wysyłany ruch, np. h2  
`xterm h2`
- uruchomienie skryptu tworzącego instancje serwerów iperf (w tym przypadku jedna)  
`./iperf-server.sh 1`
- Uruchomienie konsoli xterm na hoście, który będzie generował ruch, np. h1  
`xterm h1`
- Uruchomienie skryptu tworzącego instancje klientów programu iperf (przypadku 10 klientów uruchamiających się w odstępie 3 sekund, każdy generuje ruch przez 50 sekund, łączy się z portem 5000 na serwerze; jako argument przyjmowany adres IP hosta, na którym uruchomiono serwer)  
`./iperf-client-parallel.sh 10.0.0.2`

## 10. Potwierdzenie działania

Zrzut ekranu po wykonaniu komendy *sudo ovs-ofctl dump-flows s1*

```
mzurowsk@mzurowsk-mint:~/Documents$ sudo ovs-ofctl dump-flows s1
cookie=0x0, duration=70.200s, table=0, n_packets=6257, n_bytes=9460584, idle_timeout=5, priority=100,udp,in_port="s1-eth1",dl_src=36:07:c9:f5:b6:80,dl_dst=ba:89:b7:36:21:78,nw_src=10.0.0.1,nw_dst=10.0.0.2,tp_src=51533,tp_dst=5000 actions=output:"s1-eth2"
cookie=0x0, duration=67.198s, table=0, n_packets=5989, n_bytes=9055368, idle_timeout=5, priority=100,udp,in_port="s1-eth1",dl_src=36:07:c9:f5:b6:80,dl_dst=ba:89:b7:36:21:78,nw_src=10.0.0.1,nw_dst=10.0.0.2,tp_src=37733,tp_dst=5000 actions=output:"s1-eth3"
cookie=0x0, duration=64.194s, table=0, n_packets=5721, n_bytes=8650152, idle_timeout=5, priority=100,udp,in_port="s1-eth1",dl_src=36:07:c9:f5:b6:80,dl_dst=ba:89:b7:36:21:78,nw_src=10.0.0.1,nw_dst=10.0.0.2,tp_src=57324,tp_dst=5000 actions=output:"s1-eth4"
cookie=0x0, duration=61.192s, table=0, n_packets=5454, n_bytes=8246448, idle_timeout=5, priority=100,udp,in_port="s1-eth1",dl_src=36:07:c9:f5:b6:80,dl_dst=ba:89:b7:36:21:78,nw_src=10.0.0.1,nw_dst=10.0.0.2,tp_src=34442,tp_dst=5000 actions=output:"s1-eth2"
cookie=0x0, duration=58.188s, table=0, n_packets=5186, n_bytes=7841232, idle_timeout=5, priority=100,udp,in_port="s1-eth1",dl_src=36:07:c9:f5:b6:80,dl_dst=ba:89:b7:36:21:78,nw_src=10.0.0.1,nw_dst=10.0.0.2,tp_src=55173,tp_dst=5000 actions=output:"s1-eth4"
cookie=0x0, duration=55.186s, table=0, n_packets=4918, n_bytes=7436016, idle_timeout=5, priority=100,udp,in_port="s1-eth1",dl_src=36:07:c9:f5:b6:80,dl_dst=ba:89:b7:36:21:78,nw_src=10.0.0.1,nw_dst=10.0.0.2,tp_src=55368,tp_dst=5000 actions=output:"s1-eth3"
cookie=0x0, duration=52.183s, table=0, n_packets=4650, n_bytes=7030800, idle_timeout=5, priority=100,udp,in_port="s1-eth1",dl_src=36:07:c9:f5:b6:80,dl_dst=ba:89:b7:36:21:78,nw_src=10.0.0.1,nw_dst=10.0.0.2,tp_src=35423,tp_dst=5000 actions=output:"s1-eth2"
cookie=0x0, duration=49.180s, table=0, n_packets=4383, n_bytes=6627096, idle_timeout=5, priority=100,udp,in_port="s1-eth1",dl_src=36:07:c9:f5:b6:80,dl_dst=ba:89:b7:36:21:78,nw_src=10.0.0.1,nw_dst=10.0.0.2,tp_src=45679,tp_dst=5000 actions=output:"s1-eth3"
cookie=0x0, duration=46.183s, table=0, n_packets=4115, n_bytes=6221880, idle_timeout=5, priority=100,udp,in_port="s1-eth1",dl_src=36:07:c9:f5:b6:80,dl_dst=ba:89:b7:36:21:78,nw_src=10.0.0.1,nw_dst=10.0.0.2,tp_src=45802,tp_dst=5000 actions=output:"s1-eth4"
cookie=0x0, duration=43.178s, table=0, n_packets=3847, n_bytes=5816664, idle_timeout=5, priority=100,udp,in_port="s1-eth1",dl_src=36:07:c9:f5:b6:80,dl_dst=ba:89:b7:36:21:78,nw_src=10.0.0.1,nw_dst=10.0.0.2,tp_src=43637,tp_dst=5000 actions=output:"s1-eth2"
```

Podgląd logów z konsoli kontrolera Floodlight

```
[java] 20:27:15.517 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flows: *** [1=2, 2=4, 4=2] ***** On: *** 00:00:00:00:00:00:00:00
[java] 20:27:15.517 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flow route: ***
[java] 20:27:15.517 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:02] ***
[java] 20:27:15.517 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:03] ***
[java] 20:27:15.517 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:01] ***
[java] 20:27:18.526 INFO [p.e.a.k.SdnLabListener:nioEventLoopGroup-3-5] ***** NEW PACKET IN *****
[java] 20:27:18.526 INFO [p.e.a.k.SdnLabListener:nioEventLoopGroup-3-5] *** Switch: 00:00:00:00:00:00:00:02 ***
[java] 20:27:18.526 INFO [p.e.a.k.PacketExtractor:nioEventLoopGroup-3-5] 10.0.0.1
[java] 20:27:18.526 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flows: *** [1=3, 3=3, 4=2] ***** On: *** 00:00:00:00:00:00:00:02
[java] 20:27:18.526 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flows: *** [1=2, 2=3, 3=2] ***** On: *** 00:00:00:00:00:00:00:04
[java] 20:27:18.526 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flow route: ***
[java] 20:27:18.526 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:02] ***
[java] 20:27:18.526 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:04] ***
[java] 20:27:18.526 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:01] ***
[java] 20:27:21.515 INFO [p.e.a.k.SdnLabListener:nioEventLoopGroup-3-5] ***** NEW PACKET IN *****
[java] 20:27:21.515 INFO [p.e.a.k.SdnLabListener:nioEventLoopGroup-3-5] *** Switch: 00:00:00:00:00:00:00:02 ***
[java] 20:27:21.515 INFO [p.e.a.k.PacketExtractor:nioEventLoopGroup-3-5] 10.0.0.1
[java] 20:27:21.515 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flows: *** [1=2, 3=3, 4=3] ***** On: *** 00:00:00:00:00:00:00:02
[java] 20:27:21.515 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flow route: ***
[java] 20:27:21.515 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:02] ***
[java] 20:27:21.515 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:01] ***
[java] 20:27:24.524 INFO [p.e.a.k.SdnLabListener:nioEventLoopGroup-3-5] ***** NEW PACKET IN *****
[java] 20:27:24.524 INFO [p.e.a.k.SdnLabListener:nioEventLoopGroup-3-5] *** Switch: 00:00:00:00:00:00:00:02 ***
[java] 20:27:24.524 INFO [p.e.a.k.PacketExtractor:nioEventLoopGroup-3-5] 10.0.0.1
[java] 20:27:24.524 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flows: *** [1=3, 3=2, 4=3] ***** On: *** 00:00:00:00:00:00:00:02
[java] 20:27:24.524 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flows: *** [1=2, 2=3, 4=2] ***** On: *** 00:00:00:00:00:00:00:03
[java] 20:27:24.524 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flow route: ***
[java] 20:27:24.524 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:02] ***
[java] 20:27:24.524 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:03] ***
[java] 20:27:24.524 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:01] ***
[java] 20:27:29.124 INFO [n.f.l.l.LinkDiscoveryManager:Scheduled-0] Sending LLDP packets out of all the enabled ports
[java] 20:27:29.128 WARN [n.f.l.l.L.s.notification:nioEventLoopGroup-3-5] Link updated: Link [src=00:00:00:00:00:00:00:03 outPort=2, dst=00:00:00:00:00:00:00:02, inPort=3, latency=2]
[java] 20:27:29.145 WARN [n.f.l.l.L.s.notification:nioEventLoopGroup-3-5] Link updated: Link [src=00:00:00:00:00:00:00:01 outPort=2, dst=00:00:00:00:00:00:00:02, inPort=1, latency=4]
[java] 20:27:29.218 DEBUG [p.e.a.k.SdnLabTopologyListener:Scheduled-4] Received topology status
[java] 20:27:29.666 INFO [p.e.a.k.SdnLabListener:nioEventLoopGroup-3-5] ***** NEW PACKET IN *****
[java] 20:27:29.666 INFO [p.e.a.k.SdnLabListener:nioEventLoopGroup-3-5] *** Switch: 00:00:00:00:00:00:00:02 ***
[java] 20:27:29.666 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flows: *** [1=2, 3=3, 4=2] ***** On: *** 00:00:00:00:00:00:00:02
[java] 20:27:29.666 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** Flow route: ***
[java] 20:27:29.666 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:02] ***
[java] 20:27:29.666 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-5] *** OFSwitch DPID[00:00:00:00:00:00:00:01] ***
[java] 20:27:30.694 INFO [p.e.a.k.SdnLabListener:nioEventLoopGroup-3-2] ***** NEW PACKET IN *****
[java] 20:27:30.694 INFO [p.e.a.k.SdnLabListener:nioEventLoopGroup-3-2] *** Switch: 00:00:00:00:00:00:00:01 ***
[java] 20:27:30.694 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-2] *** Flows: *** [2=2, 3=3, 4=2] ***** On: *** 00:00:00:00:00:00:00:01
[java] 20:27:30.694 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-2] *** Flow route: ***
[java] 20:27:30.694 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-2] *** OFSwitch DPID[00:00:00:00:00:00:00:01] ***
[java] 20:27:30.694 INFO [pl.edu.agh.kt.routes.nioEventLoopGroup-3-2] *** OFSwitch DPID[00:00:00:00:00:00:00:02] ***
```