SDN Opdracht 2

BASISNETWERK MET VLAN-SEGMENTATIE

H.DONKERBROEK - 374181 | B.BOELENS - 453713

SDN Opdracht 2 portfolio

H.Donkerbroek, 374181

B.Boelens, 453713

Vak: Software Defined Networking Practicum

Docent: Eleanor van Wijck

Oplevering: 16/10/25

Inhoud

1. Inleiding 2
2. Theoretisch kader 2
   2.1. Wat is SDN? 2
   2.2. Wat is OpenFlow? 3
   2.3. Wat is een flow entry en hoe wordt deze gelezen? 3
   2.4. Wat doen de verschillende OpenFlow-tabellen? 4
   2.5. Hoe werkt Mininet? 4
   2.6. Wat is Faucet? 4
3. Eindproduct 6
   3.1. Schematische netwerktekening 6
   3.2. Flow entries 6
   3.3. Onderbouwing ontwerpkeuzes 7
   3.4. Functioneel bewijs 8
4. Reflectie en evaluatie 8
5. Literatuurlijst 9
6. Bijlagen 10
   6.1. Bijlage 1 [Flow dump] 10
   6.2. Bijlage 2 [Net command] 11

6.3. Bijlage 3 [Links command] 11
6.4. Bijlage 4 [Port security] 12
6.5. Bijlage 5 [Rate limiting] 13
6.6. Bijlage 6 [VLAN segmentatie] 14
6.7. Bijlage 7 [Hosts naar ISP] 15
6.8. Bijlage 8 [Logboek] 16
7. Github & Video 17

1. Inleiding

Het College van Bestuur (CvB) heeft besloten om de huidige, verouderde Cisco-netwerkinfrastructuur volledig te vervangen door een moderne Software Defined Networking (SDN) oplossing. Om kosten te besparen is ervoor gekozen om deze migratie in-house uit te voeren door het bestaande IT-team in plaats van externe consultants in te huren.

Het doel van dit project is het ontwikkelen van een schaalbaar, veilig en toekomstbestendig SDN-netwerk dat voldoet aan de eisen van het CvB en volledig beheerd kan worden door het interne IT-team. Het is een SDN netwerk met een centrale faucet controller die alle switches beheert.

2. Theoretisch kader

2.1. Wat is SDN?

Software-Defined Networking (SDN) is een netwerkarchitectuur waarbij de controle laag wordt losgekoppeld van de datalaag. In normale netwerken beslissen switches of routers zelf hoe pakketten worden verwerkt. In SDN worden de switches en routers vanuit een centrale controller aangestuurd. Een SDN netwerk is flexibeler dan een klassiek netwerk omdat het netwerkgedrag softwarematig kan worden aangepast zonder fysieke aanpassingen. Door de centrale controller is het netwerk ook snel en gemakkelijk te optimaliseren.

2.2. Wat is OpenFlow?

OpenFlow is het meest gebruikte communicatieprotocol voor SDN. Het zorgt ervoor dat de controller in staat is regels te installeren in switches, zodat deze pakketten kunnen matchen en verwerken.

Een switch bevat flow tables. In deze tables staan flow entries die beschrijven hoe pakketten met bepaalde header-kenmerken moeten worden behandeld. Als een pakket geen match vind wordt het pakket naar de controller gestuurd. De controller beslist wat met het pakket moet gebeuren en geeft een nieuwe regel terug. De controller beheert de regels en de switch voert alleen de instructies van de controller uit. Het is een "domme" switch omdat het alleen pakketten door geeft en niet zelf hoeft te beslissen.

2.3. Wat is een flow entry en hoe wordt deze gelezen?

Een flow entry is een regel in een flow table die bepaalt hoe pakketten met bepaalde kenmerken verwerkt moeten worden.

Een flow entry bevat:

- Match fields: de criteria waarop pakketten gematcht worden zoals; poort, MAC-adres, IP.
- Priority: welke regel wint als meerdere entries matchen
- Counters: statistieken (aantal pakketten, bytes).
- Instructions/Actions: wat er met een pakket met gebeuren(doorsturen, droppen, headers wijzigen, naar controller sturen).
- Timeouts (optioneel): hoelang de regel geldig is.

  Flow entry voorbeeld:

  Match: in_port=1, eth_type=0x0800, ipv4_dst=10.0.0.2
  Action: output=2

Alle IPV4-pakketten die binnenkomen op poort 1 en bestemming 10.0.0.2 hebben, worden doorgestuurd naar poort 2.

2.4. Wat doen de verschillende OpenFlow-tabellen?

Binnen OpenFlow is er sprake van een aantal verschillende tabellen. Elke tabel heeft een eigen structuur en functie.

1. Flow table

Deze tabel bevat de flow entries die bepalen hoe pakketten worden verwerkt.
Elke flow entry bevat een match criteria, een action en een counter.

2. Group table

De group table ondersteunt bepaalde geavanceerde actiesets zoals bijvoorbeeld load balancing, multicast, broadcast en failover.

3. Meter table

Deze tabel beheert de bandbreedte en QoS en definieert acties zoals de rate limiting pet VLAN en de prioritering van het verkeer.

4. Table pipeline

De table pipeline houdt eigenlijk in dat flow tables in series kunnen worden uitgevoerd. Een pakket wordt na verwerking van de ene tabel doorgestuurd naar de andere tabel. Dit maakt het mogelijk om complexe policies zoals VLAN-segmentatie, firewalls en forwarding te implementeren.

2.5. Hoe werkt Mininet?

Mininet is een tool om netwerken virtueel te simuleren op één computer. Je bouwt een netwerk met OpenFlow switches en hosts die pakketten kunnen versturen en ontvangen. Binnen Mininet kan je bepaalde commando's zoals pingall en iperf gebruiken om snel de connectiviteit en latency binnen het netwerk te testen.

Het voordeel van Mininet is dat het de mogelijkheid biedt om custom python scripts te gebruiken om een netwerk te ontwikkelen. Dit zorgt ervoor dat een netwerk snel en logisch, virtueel opgebouwd kan worden.

2.6. Wat is Faucet?

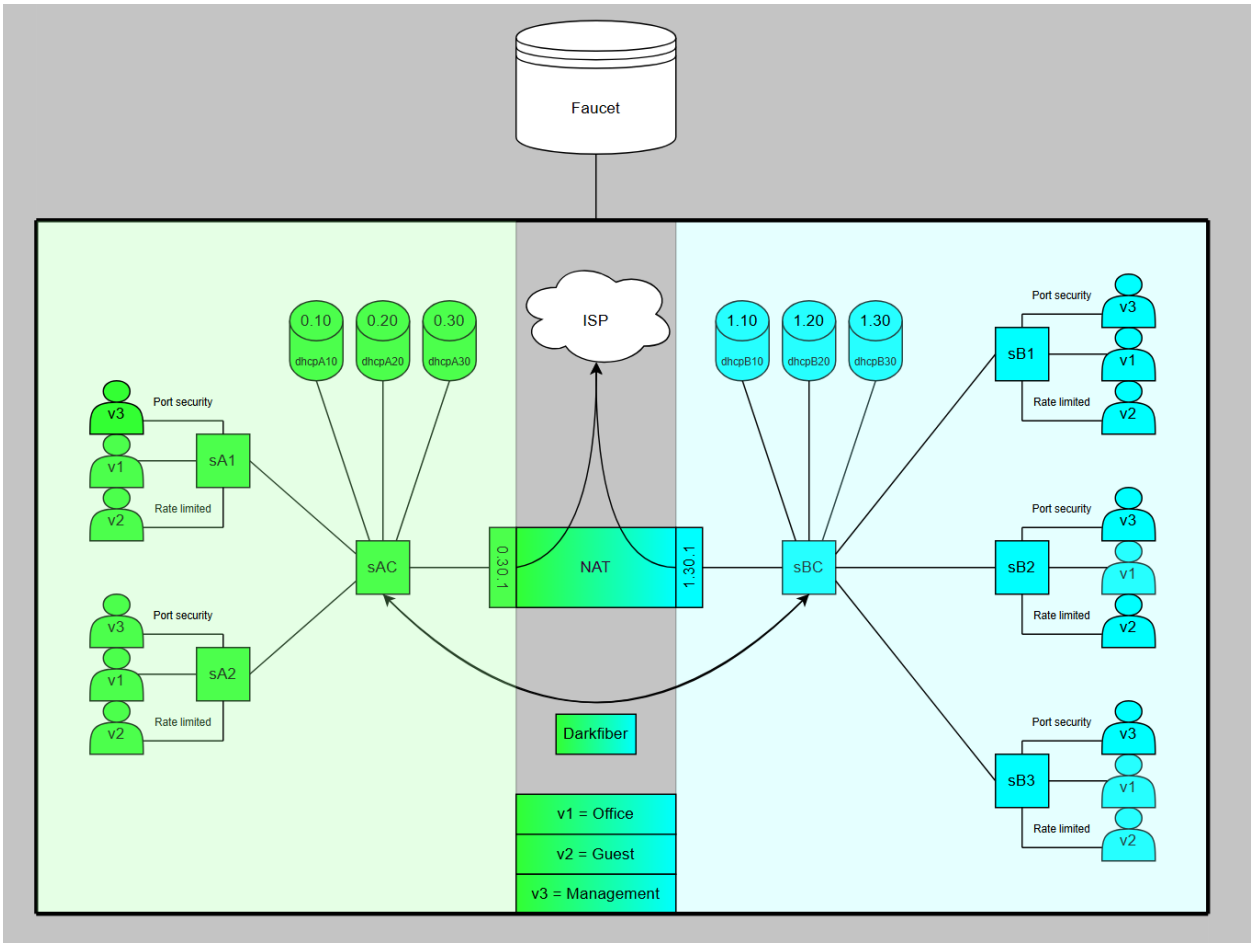
Faucet is een OpenFlow controller die controleert hoe het dataplane verkeer verwerkt wordt mede door het centraal beheren en configureren van de switches in een netwerk. Deze switches worden configureerd d.m.v. YAML files. (Yet Another Markup Language)

Elke switch krijgt een logische naam en interfaces worden toegewezen aan specifieke VLAN's of policies. Vervolgens leest de Faucet controller elke YAML en zet het om naar flow entries. Voor elke switch wordt er bepaald welke flows er nodig zijn om packetmatching en bepaalde acties uit te voeren en hoe deze switch bepaalde NFV-functies moet doen.

Faucet verbind via OpenFlow met de switches en de switches voegen de regels toe aan hun flow table. Switches volgen de flow entries voor elk pakket wat er binnenkomt, als er een pakket binnenkomt waarvoor geen flow entry bestaat dan stuurt de switch dit naar de controller. De controller bepaald dan wat er moet gebeuren.

3. Eindproduct

3.1. Schematische netwerktekening



3.2. Flow entries

Na verder onderzoek is geconstateerd dat faucet hybride, pro-actief te werk gaat. Dat houdt in dat faucet zelf bepaalde flows installeert voordat er überhaupt verkeer op het netwerk is. Dit gedrag komt voort uit het idee dat het mogelijk is dat een bepaalde flow nodig kan zijn, dus dan wordt die flow aangemaakt. Denk bijvoorbeeld aan routing en vlans. Daarnaast installeert faucet L2 flows als er verkeer in het netwerk is. Denk hierbij aan het leren van MAC-adressen.

Het feit dat faucet pro-actief flows installeert zorgt ervoor dat er veel flows aanwezig zijn die nooit gehit worden. Ze bestaan omdat er een mogelijkheid is dat ze nodig zijn, maar in het mininet script worden ze nooit gehit. Daarom worden alleen de flows die hits hebben in het netwerk behandeld.

Bijlage 1 [Flow dumps] laat de dump van alle switches zien, hier wordt bovenstaande uitleg visueel onderbouwd. Omdat er ontzettend veel flows zijn, worden deze niet allemaal benoemd.

cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=0, n_bytes=0, priority=20477,ip,in_port="sA1-eth2",nw_dst=10.0.20.0/24 actions=meter:1,goto_table:1

Dit is een rate limiting regel, die zegt dat als verkeer op sA1 binnenkomt van vlan 20, dat dit doorgezet moet worden naar de meter.

cookie=0x5adc15c0, duration=8683s, table=2, n_packets=0, n_bytes=0, priority=8191,in_port="sA1-eth3",dl_vlan=300,dl_src=00:00:0a:01:01:01 actions=goto_table:5

Dit is een port security regel op sA1. Op sA1 poort 3 in vlan 300, als je dit MAC adres hebt, goto table 5. Als deze regel niet matched (iemand sluit zijn eigen laptop aan en krijgt dus een ander MAC) dan wordt het gedropt.

3.3. Onderbouwing ontwerpkeuzes

De topologie is voor het meerendeel opgezet volgens de opdracht.

- Twee gebouwen
- A heeft 2 switches en B heeft 3 switches (1 per verdieping)
- De switches in een gebouw gaan naar de centrale patchkast (sAC, sBC)
- De centrale patchkast is verbonden via darkfiber met het andere gebouw
- Beide gebouwen hebben een link naar een isp
- Beide gebouwen maken onderscheid tussen 3 vlans

Om dit te realiseren hebben wij twee gebouwen gedefinieerd met 3 vlans met 1 DHCP server per vlan. Vervolgens is er gekozen om een NAT host te simuleren om de uplink van de gebouwen naar de ISP op te zetten. In een eerder stadium van de ontwikkeling had gebouw A zijn eigen nat en gebouw B ook. Daarbij heeft gebouw B 3 eigen vlans gekregen met 3 eigen DHCP servers. De reden hiervoor is dat er aangegeven stond dat “Gebouw B heeft net als Gebouw A een eigen verbinding met een ISP. Deze lijnen kunnen redundantie bieden of gebruikt worden voor load balancing of segmentatie op netwerkniveau.”

Om ervoor te zorgen dat beide gebouwen een eigen uplink hadden was totale segmentatie van de gebouwen logisch, zodat gebouw A als gateway 10.0.30.1 kon gebruiken voor de nat en gebouw B 10.1.30.1. (Bij de DHCP lease wordt een gateway meegegeven, dus daarom moesten de twee gebouwen aparte vlans en DHCP servers hebben.)

Na een veel testen is het ons niet gelukt om 2 NAT nodes te implementeren. Daarom is ervoor gekozen om 1 shared NAT te gebruiken voor beide gebouwen. Hierbij zijn wel de 3 vlans in A en 3 vlans in B gebleven, tevens als de 3 DHCP servers per gebouw. Dit zorgt voor een betere segmentatie, een logischere indeling per gebouw en meer plek in het netwerk. Ook houdt dit de deur open voor toekomstige uitbreiding waarbij een tweede NAT toegevoegd moet worden.

Het gebruik van 1 NAT heeft ook een onderhoudsvoordeel, aangezien je niet twee instanties hoeft te overzien. Het verkeer tussen de gebouwen gaat niet door de NAT, hiervoor wordt expliciet de darkfiber gebruikt. Zoals ook te zien in de netwerktekening, heeft de NAT één interface in beide gebouwen en gebruiken zij deze alleen om de ISP te kunnen bereiken.

Er is gekozen voor het gebruik van DHCP-servers i.p.v. hardcoded IP adressen om voor schaalbaarheid en praktische implementatie te zorgen. Binnen de huidige implementatie is het mogelijk om meer dan 1200 hosts toe te voegen die allemaal direct een IP krijgen via DHCP.

3.4. Functioneel bewijs

Zie hoofdstuk 6 Bijlage.

4. Reflectie en evaluatie

- Zijn de doelstellingen bereikt?

Los van de twee independent NAT links zijn alle doelstellingen bereikt. Er is wel NAT aanwezig met de link naar de ISP waardoor er wel internet access is. Dus alle doelstellingen zijn behaald maar niet helemaal op de gewenste manier.

- Wat ging goed en wat ging lastig?

In principe ging de opdracht in zijn geheel vrij voorspoedig. De grootste problemen waar wij tegen aan zijn gelopen is dat er maar beperkte documentatie te vinden is omtrent het gebruik van mininet samen met faucet. Daarnaast vonden wij de lesindeling ook wel lastig. Iedereen was net begonnen met de opdracht en toen werd er eigenlijk al aangegeven dat dit de laatste les was voor de oplevering. Wij zijn gaandeweg de opdracht dus ook wel tegen dingen aangelopen waar wij eigenlijk wel ondersteuning bij hadden willen hebben.

- Was de aanpak effectief?

De aanpak hebben wij als effectief ervaren. Eigenlijk zijn wij altijd met z'n tweeën bezig geweest met de opdracht, deze manier heeft ervoor gezorgd dat wij allebei wisten waar we mee bezig waren en wat de logische volgende stap is.

- Wat zou je de volgende keer anders doen?

Puur iets anders doen zit er volgens ons niet echt in. Wij zijn van mening dat wij de opdracht in volledigheid hebben volbracht.

- Wat heb je geleerd en toegepast?

Wij hebben geleerd wat mininet en faucet is, hoe deze werken en hoe je een SDN netwerk opzet en onderhoud. Daarnaast heeft deze opdracht ons een beter inzicht gegeven in gecentraliseerde controle.

- Welke uitbreiding zou je met meer tijd/middelen doen?

Met meer tijd en middelen zouden wij toch een mogelijkheid willen vinden om 2 NATs werkend te krijgen, tevens als een tweede faucet controller zodat daar ook redundantie in zit.

5. Literatuurlijst

5.1 Configuration — Faucet documentation. (z.d.).

https://docs.faucet.nz/en/latest/configuration.html

5.2 Contributors, M. P. (z.d.-b). *Mininet Walkthrough - Mininet*. https://mininet.org/walkthrough/

5.3 Serban, C. (2017, 7 maart). *SDN Lesson #2 – Introducing Faucet as an OpenFlow Controller - CostiSer.Ro*. https://costiser.ro/2017/03/07/sdn-lesson-2-introducing-faucet-as-an-openflow-controller/

6. Bijlagen

6.1. Bijlage 1 [Flow dump]

```
mininet> dpctl dump-flows -O OpenFlow13
*** sA1 ------------------------------------------------------------------------
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=1928, n_bytes=133366, priority=20480,in_port="sA1-eth25" actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=281, n_bytes=11802, priority=20480,arp,in_port="sA1-eth2" actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=282, n_bytes=11844, priority=20480,arp,in_port="sA1-eth3" actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=280, n_bytes=11760, priority=20480,arp,in_port="sA1-eth1" actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20471,ip,in_port="sA1-eth2" actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.456s, table=0, n_packets=0, n_bytes=0, priority=20471,ip,in_port="sA1-eth1" actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.456s, table=0, n_packets=0, n_bytes=0, priority=20469,ip,in_port="sA1-eth3" actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=2, n_bytes=684, priority=20479,udp,in_port="sA1-eth2",tp_src=68,tp_dst=67 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=2, n_bytes=684, priority=20479,udp,in_port="sA1-eth3",tp_src=68,tp_dst=67 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=2, n_bytes=684, priority=20479,udp,in_port="sA1-eth1",tp_src=68,tp_dst=67 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=0, n_bytes=0, priority=20478,udp,in_port="sA1-eth2",tp_src=67,tp_dst=68 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=0, n_bytes=0, priority=20478,udp,in_port="sA1-eth3",tp_src=67,tp_dst=68 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=0, n_bytes=0, priority=20478,udp,in_port="sA1-eth1",tp_src=67,tp_dst=68 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=0, n_bytes=0, priority=20477,ip,in_port="sA1-eth2",nw_dst=10.0.20.0/24 actions=meter:1,goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=0, n_bytes=0, priority=20477,ip,in_port="sA1-eth3",nw_dst=10.0.30.0/24 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=0, n_bytes=0, priority=20477,ip,in_port="sA1-eth1",nw_dst=10.0.10.0/24 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.488s, table=0, n_packets=0, n_bytes=0, priority=20476,ip,in_port="sA1-eth2",nw_dst=10.1.20.0/24 actions=meter:1,goto_table:1
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20476,ip,in_port="sA1-eth3",nw_dst=10.1.30.0/24 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20476,ip,in_port="sA1-eth1",nw_dst=10.1.10.0/24 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20475,ip,in_port="sA1-eth2",nw_dst=10.0.30.0/24 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20475,ip,in_port="sA1-eth1",nw_dst=10.0.30.0/24 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20474,ip,in_port="sA1-eth2",nw_dst=10.1.30.0/24 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20474,ip,in_port="sA1-eth1",nw_dst=10.1.30.0/24 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20473,ip,in_port="sA1-eth2",nw_dst=10.0.10.0/24 actions=drop
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20473,ip,in_port="sA1-eth3",nw_dst=10.0.10.0/24 actions=drop
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20473,ip,in_port="sA1-eth1",nw_dst=10.0.20.0/24 actions=drop
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20472,ip,in_port="sA1-eth2",nw_dst=10.1.10.0/24 actions=drop
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20472,ip,in_port="sA1-eth3",nw_dst=10.1.10.0/24 actions=drop
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20472,ip,in_port="sA1-eth1",nw_dst=10.1.20.0/24 actions=drop
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20471,ip,in_port="sA1-eth3",nw_dst=10.0.20.0/24 actions=drop
 cookie=0x5adc15c0, duration=8692.456s, table=0, n_packets=0, n_bytes=0, priority=20470,ip,in_port="sA1-eth3",nw_dst=10.1.20.0/24 actions=drop
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20475,ip,in_port="sA1-eth3",nw_dst=10.0.30.254 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.487s, table=0, n_packets=0, n_bytes=0, priority=20474,ip,in_port="sA1-eth3",nw_dst=10.1.30.254 actions=goto_table:1
 cookie=0x5adc15c0, duration=8692.456s, table=0, n_packets=0, n_bytes=0, priority=0 actions=drop
 cookie=0x5adc15c0, duration=8692.489s, table=1, n_packets=1447, n_bytes=86820, priority=9099,in_port="sA1-eth25",dl_dst=01:80:c2:00:00:00/ff:ff:ff:ff:ff:f0,dl_type=0x88cc actions=CONTROLLER:128
 cookie=0x5adc15c0, duration=0.755s, table=1, n_packets=0, n_bytes=0, hard_timeout=281, priority=8191,dl_vlan=310,dl_src=0e:00:00:00:31:01,dl_dst=9a:95:70:28:85:cc actions=goto_table:4
 cookie=0x5adc15c0, duration=0.739s, table=1, n_packets=0, n_bytes=0, hard_timeout=286, priority=8191,dl_vlan=110,dl_src=0e:00:00:00:11:01,dl_dst=00:00:00:00:00:02 actions=goto_table:4
 cookie=0x5adc15c0, duration=0.739s, table=1, n_packets=0, n_bytes=0, hard_timeout=255, priority=8191,dl_vlan=210,dl_src=0e:00:00:00:21:01,dl_dst=00:00:00:00:00:04 actions=goto_table:4
 cookie=0x5adc15c0, duration=0.739s, table=1, n_packets=0, n_bytes=0, hard_timeout=281, priority=8191,dl_vlan=310,dl_src=0e:00:00:00:31:01,dl_dst=00:00:00:00:00:06 actions=goto_table:4
 cookie=0x5adc15c0, duration=8692.489s, table=1, n_packets=213, n_bytes=12780, priority=4097,in_port="sA1-eth25",vlan_tci=0x0000/0x1fff actions=drop
 cookie=0x5adc15c0, duration=8692.489s, table=1, n_packets=283, n_bytes=12486, priority=4096,in_port="sA1-eth2",vlan_tci=0x0000/0x1fff actions=push_vlan:0x8100,set_field:4296->vlan_vid,goto_table:2
 cookie=0x5adc15c0, duration=8692.489s, table=1, n_packets=284, n_bytes=12528, priority=4096,in_port="sA1-eth3",vlan_tci=0x0000/0x1fff actions=push_vlan:0x8100,set_field:4396->vlan_vid,goto_table:2
 cookie=0x5adc15c0, duration=8692.488s, table=1, n_packets=282, n_bytes=12444, priority=4096,in_port="sA1-eth1",vlan_tci=0x0000/0x1fff actions=push_vlan:0x8100,set_field:4196->vlan_vid,goto_table:2
 cookie=0x5adc15c0, duration=8692.489s, table=1, n_packets=268, n_bytes=33766, priority=4096,in_port="sA1-eth25" actions=goto_table:2
 cookie=0x5adc15c0, duration=8692.488s, table=1, n_packets=0, n_bytes=0, priority=0 actions=drop
 cookie=0x5adc15c0, duration=8692.495s, table=2, n_packets=0, n_bytes=0, priority=20490,dl_type=0x9000 actions=drop
 cookie=0x5adc15c0, duration=8692.495s, table=2, n_packets=0, n_bytes=0, priority=20480,dl_src=ff:ff:ff:ff:ff:ff actions=drop
 cookie=0x5adc15c0, duration=8692.495s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth1",dl_src=0e:00:00:00:10:01 actions=drop
 cookie=0x5adc15c0, duration=8692.495s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth2",dl_src=0e:00:00:00:10:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth3",dl_src=0e:00:00:00:10:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth1",dl_src=0e:00:00:00:11:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth2",dl_src=0e:00:00:00:11:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth3",dl_src=0e:00:00:00:11:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth1",dl_src=0e:00:00:00:20:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth2",dl_src=0e:00:00:00:20:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth3",dl_src=0e:00:00:00:20:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth1",dl_src=0e:00:00:00:21:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth2",dl_src=0e:00:00:00:21:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth3",dl_src=0e:00:00:00:21:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth1",dl_src=0e:00:00:00:30:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth2",dl_src=0e:00:00:00:30:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth3",dl_src=0e:00:00:00:30:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth1",dl_src=0e:00:00:00:31:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth2",dl_src=0e:00:00:00:31:01 actions=drop
 cookie=0x5adc15c0, duration=8692.494s, table=2, n_packets=0, n_bytes=0, priority=20480,in_port="sA1-eth3",dl_src=0e:00:00:00:31:01 actions=drop
 cookie=0x5adc15c0, duration=8692.492s, table=2, n_packets=0, n_bytes=0, priority=16640,in_port="sA1-eth25",dl_vlan=100,dl_dst=01:80:c2:00:00:00/ff:ff:ff:00:00:00 actions=goto_table:6
```

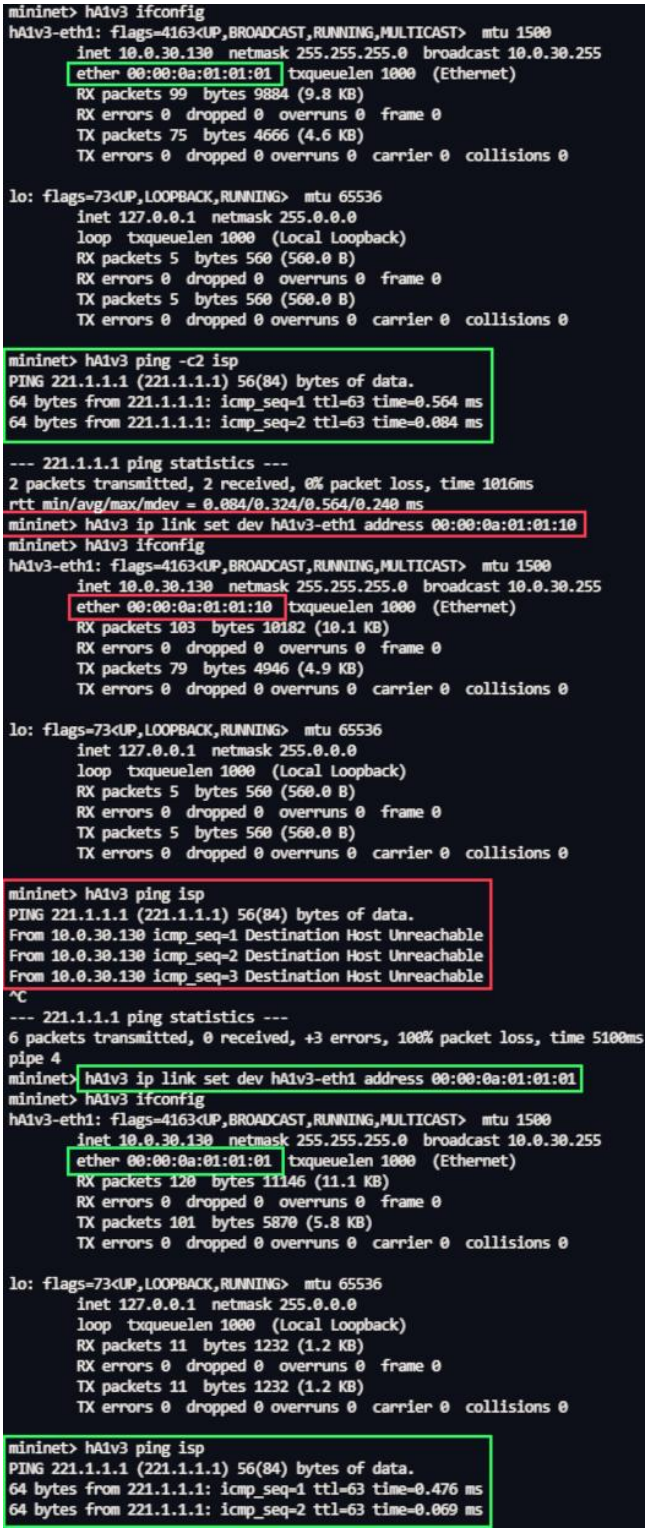
6.2. Bijlage 2 [Net command]

```
mininet> net
dhcp10 dhcp10-eth1:sAC-eth100
dhcp10b dhcp10b-eth1:sBC-eth100
dhcp20 dhcp20-eth1:sAC-eth200
dhcp20b dhcp20b-eth1:sBC-eth200
dhcp30 dhcp30-eth1:sAC-eth300
dhcp30b dhcp30b-eth1:sBC-eth300
hA1v1 hA1v1-eth1:sA1-eth1
hA1v2 hA1v2-eth1:sA1-eth2
hA1v3 hA1v3-eth1:sA1-eth3
hA2v1 hA2v1-eth1:sA2-eth1
hA2v2 hA2v2-eth1:sA2-eth2
hA2v3 hA2v3-eth1:sA2-eth3
hB1v1 hB1v1-eth1:sB1-eth1
hB1v2 hB1v2-eth1:sB1-eth2
hB1v3 hB1v3-eth1:sB1-eth3
hB2v1 hB2v1-eth1:sB2-eth1
hB2v2 hB2v2-eth1:sB2-eth2
hB2v3 hB2v3-eth1:sB2-eth3
hB3v1 hB3v1-eth1:sB3-eth1
hB3v2 hB3v2-eth1:sB3-eth2
hB3v3 hB3v3-eth1:sB3-eth3
isp isp-eth1:nat-eth3
nat nat-eth1:sAC-eth50 nat-eth2:sBC-eth50 nat-eth3:isp-eth1
sA1 lo:  sA1-eth1:hA1v1-eth1 sA1-eth2:hA1v2-eth1 sA1-eth3:hA1v3-eth1 sA1-eth25:sAC-eth2
sA2 lo:  sA2-eth1:hA2v1-eth1 sA2-eth2:hA2v2-eth1 sA2-eth3:hA2v3-eth1 sA2-eth25:sAC-eth3
sAC lo:  sAC-eth1:sBC-eth1 sAC-eth2:sA1-eth25 sAC-eth3:sA2-eth25 sAC-eth50:nat-eth1 sAC-eth100:dhcp10-eth1 sAC-eth200:dhcp20-eth1 sAC-eth300:dhcp30-eth1
sB1 lo:  sB1-eth1:hB1v1-eth1 sB1-eth2:hB1v2-eth1 sB1-eth3:hB1v3-eth1 sB1-eth25:sBC-eth2
sB2 lo:  sB2-eth1:hB2v1-eth1 sB2-eth2:hB2v2-eth1 sB2-eth3:hB2v3-eth1 sB2-eth25:sBC-eth3
sB3 lo:  sB3-eth1:hB3v1-eth1 sB3-eth2:hB3v2-eth1 sB3-eth3:hB3v3-eth1 sB3-eth25:sBC-eth4
sBC lo:  sBC-eth1:sAC-eth1 sBC-eth2:sB1-eth25 sBC-eth3:sB2-eth25 sBC-eth4:sB3-eth25 sBC-eth50:nat-eth2 sBC-eth100:dhcp10b-eth1 sBC-eth200:dhcp20b-eth1 sBC-eth300:dhcp30b-eth1
c0
```

6.3. Bijlage 3 [Links command]

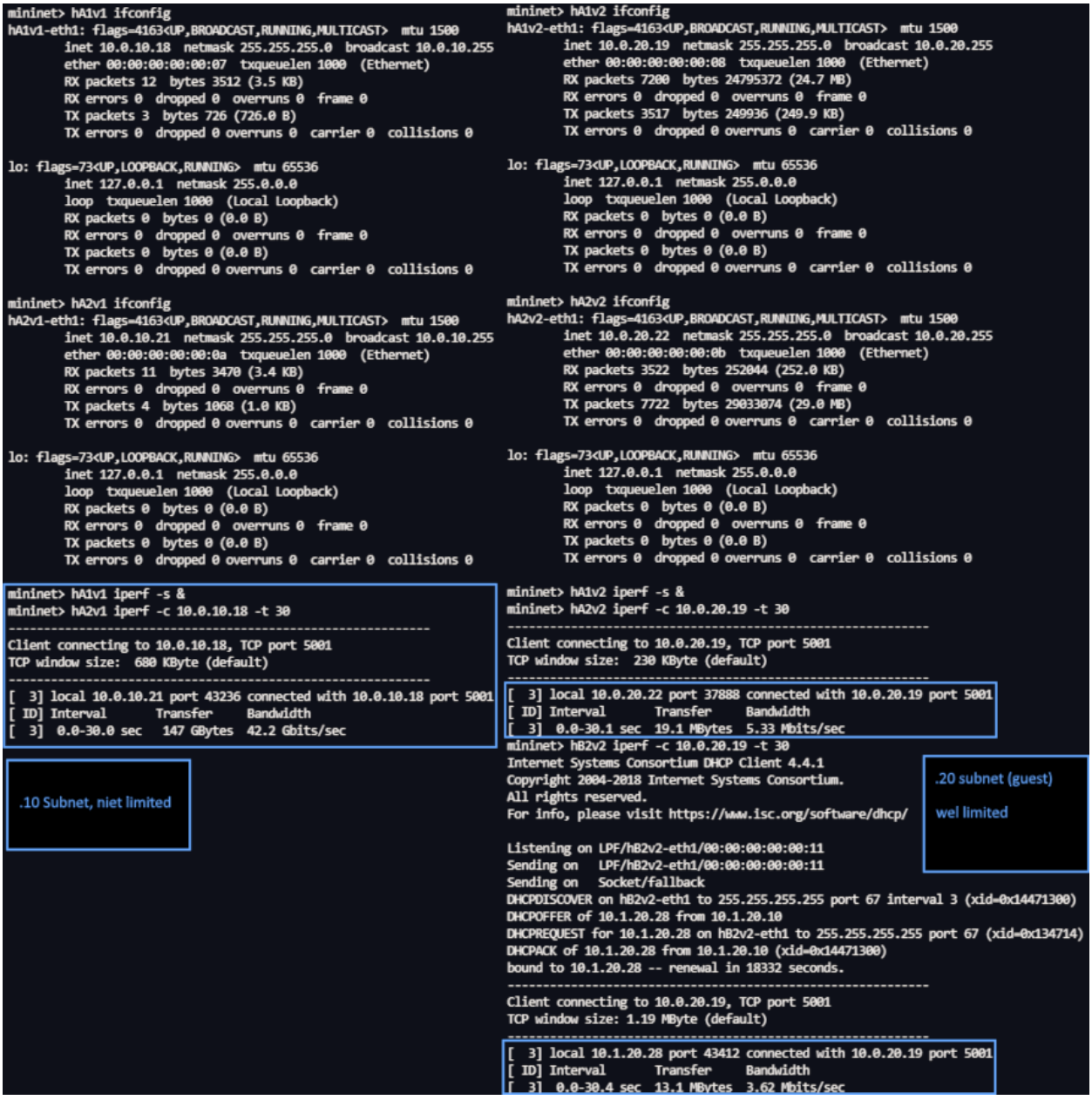
```
mininet> links
nat-eth3<->isp-eth1 (OK OK)
nat-eth1<->sAC-eth50 (OK OK)
nat-eth2<->sBC-eth50 (OK OK)
sA1-eth1<->hA1v1-eth1 (OK OK)
sA1-eth2<->hA1v2-eth1 (OK OK)
sA1-eth3<->hA1v3-eth1 (OK OK)
sA2-eth1<->hA2v1-eth1 (OK OK)
sA2-eth2<->hA2v2-eth1 (OK OK)
sA2-eth3<->hA2v3-eth1 (OK OK)
sAC-eth100<->dhcp10-eth1 (OK OK)
sAC-eth200<->dhcp20-eth1 (OK OK)
sAC-eth300<->dhcp30-eth1 (OK OK)
sAC-eth2<->sA1-eth25 (OK OK)
sAC-eth3<->sA2-eth25 (OK OK)
sAC-eth1<->sBC-eth1 (OK OK)
sB1-eth1<->hB1v1-eth1 (OK OK)
sB1-eth2<->hB1v2-eth1 (OK OK)
sB1-eth3<->hB1v3-eth1 (OK OK)
sB2-eth1<->hB2v1-eth1 (OK OK)
sB2-eth2<->hB2v2-eth1 (OK OK)
sB2-eth3<->hB2v3-eth1 (OK OK)
sB3-eth1<->hB3v1-eth1 (OK OK)
sB3-eth2<->hB3v2-eth1 (OK OK)
sB3-eth3<->hB3v3-eth1 (OK OK)
sBC-eth100<->dhcp10b-eth1 (OK OK)
sBC-eth200<->dhcp20b-eth1 (OK OK)
sBC-eth300<->dhcp30b-eth1 (OK OK)
sBC-eth2<->sB1-eth25 (OK OK)
sBC-eth3<->sB2-eth25 (OK OK)
sBC-eth4<->sB3-eth25 (OK OK)
```

6.4. Bijlage 4 [Port security]



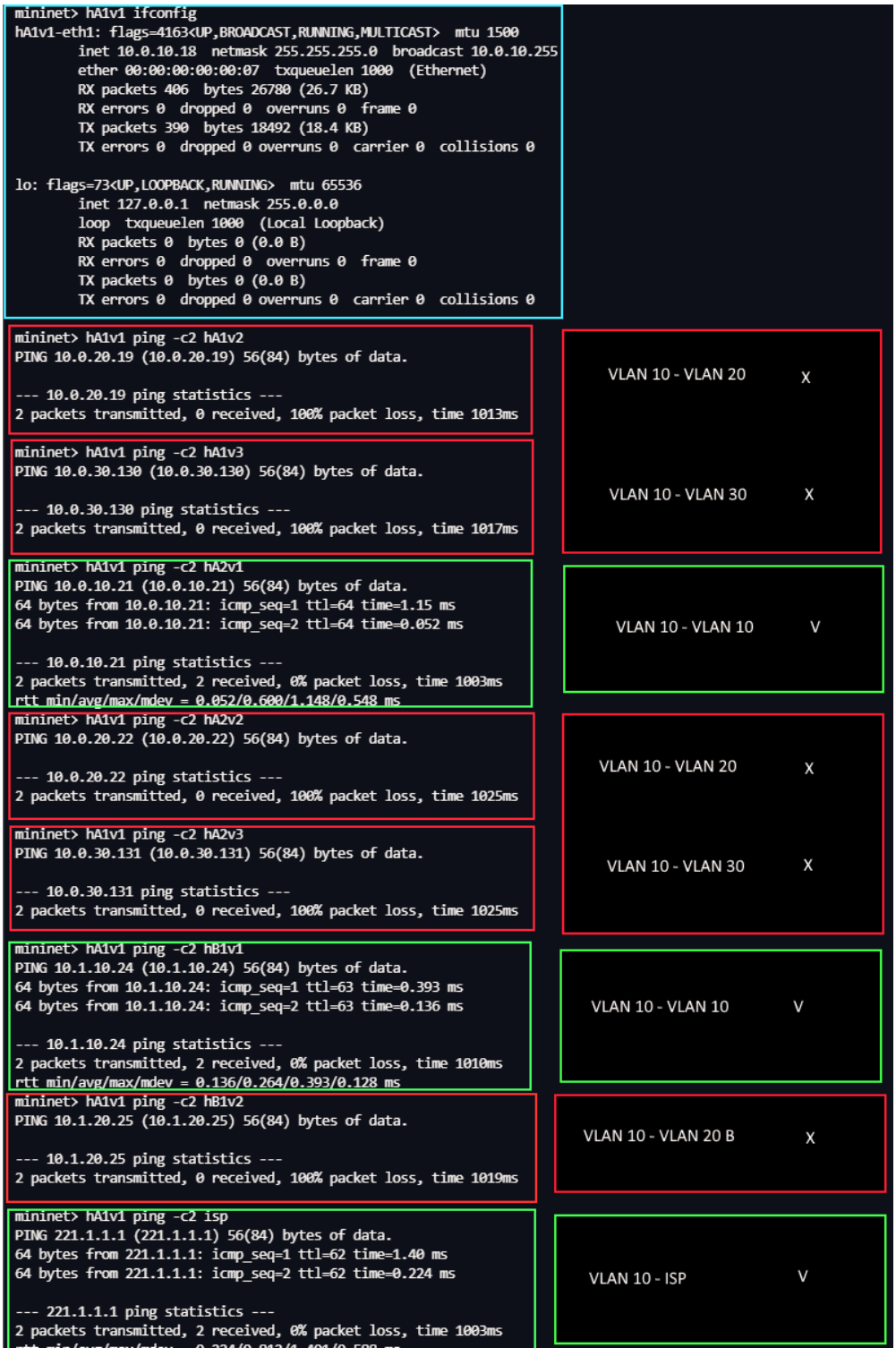
Zodra het MAC adres van hA1v3 (h – Host, A – gebouw A, 1 – Switch 1, v3 – vlan300) verandert, verlies deze host connectie met het netwerk. Oftewel port security.

6.5. Bijlage 5 [Rate limiting]



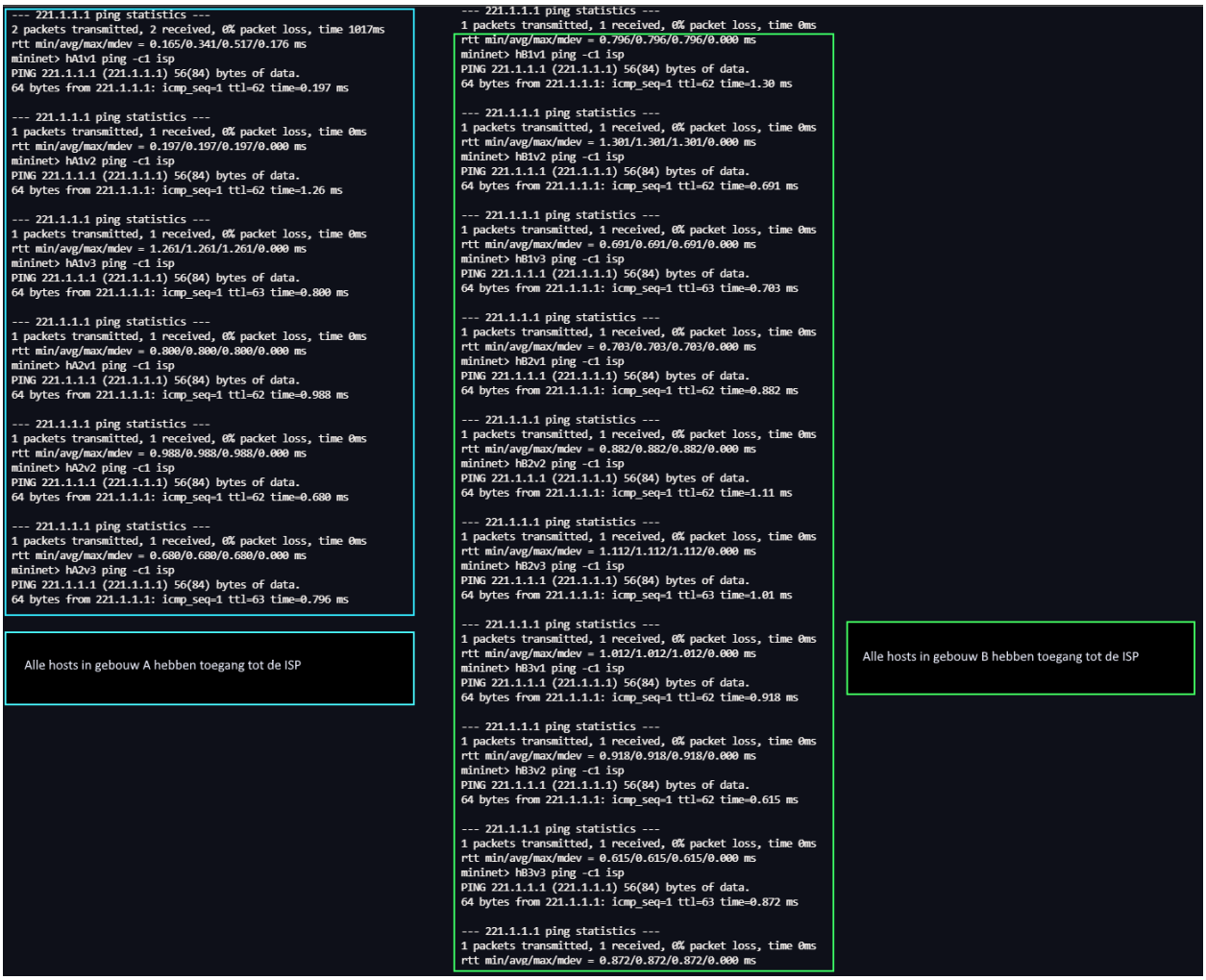
hA1v1, een host op vlan Office, heeft een ongelimiteerde bandbreedte. De afbeelding daarnaast laat zien dat hA1v2, een host op vlan Guest, gelimiteerd is tot 5 Mbits/s. Oftwel rate limiting voor de Guest vlan.

6.6. Bijlage 6 [VLAN segmentatie]



VLANs kunnen geen andere VLANs bereiken, behalve de mirror VLAN in het andere gebouw. Alle VLANs kunnen wel de isp bereiken.

6.7. Bijlage 7 [Hosts naar ISP]



6.8. Bijlage 8 [Logboek]

Datum	Activiteit	Beschrijving
16-09	Mininet & Faucet install	Opzet mininet VM en Faucet controller.
18-09	Topologie	Test topologie gemaakt om mininet te leren.
23-09	Topologie met Faucet	Remote Faucet controller toegevoegd en basis config geimplementeerd.
26-09	DHCP	DHCP host opgezet en geconfigureerd.
29-09	DHCP	DHCP hosts voor elke VLAN, werkend.
02-10	NAT	Basis NAT topologie zonder Faucet test.
03-10	NAT met controller	Remote Faucet controller in NAT topologie.
06-10	2 NATS met DHCP	NAT topologie gecombineerd met DHCP configs, 2 NATs werken niet samen.
07-10	2 NATS met DHCP	Elke keer werkt maar 1 van de 2 NATs, conclusie, wij weten niet hoe je 2 NATs moet gebruiken.
08-10	Faucet met STACKS	Faucet config omgezet naar Stacked i.p.v. trunks. 2 NATs werkt nogsteeds niet.
9-10	1 NAT met DHCP	Terug naar 1 NAT. Werkt wel, DHCP toegevoegd.
11-10	ACLS en VLAN segmentatie	Staande topologie uitgebreid met VLAN segmentatie.
14-10	NFV	Rate limiting voor Guest VLAN en port security voor Management VLAN toegevoegd.

7. Github & Video

Google drive link voor video netwerk:

https://drive.google.com/file/d/10Pib1i7qIZgDRUAnfnP4PtgKmWuYxT_O/view?usp=sharing

Github link voor scripts:

https://github.com/basboelens/SDNhugo