

Kim Kurillo

Ohjelmisto-ohjatut verkot (SDN)

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

9.11.2014

Tekijä Otsikko	Kim Kurillo Ohjelmisto-ohjatut verkot (SDN)
Sivumäärä Aika	37 sivua + 5 liitettä 9.11.2014
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Tietoverkot
Ohjaaja(t)	Tietoliikennekonsultti Jaakko Rautanen Yliopettaja Janne Salonen
<p>Tämän insinööriyön tavoitteena on selvittää ohjelmisto-ohjattujen verkkojen (Software-Defined Networking) toimintaa sekä sen tarjoamia käyttömahdollisuuksia tietoverkoissa. Työ on tehty Cygate Oy:lle, jonka palveluihin kuuluu tietoverkkoratkaisujen suunnittelu ja ylläpito, joten SDN:n kaltaisen uuden tekniikan osaaminen on tärkeää tietoa.</p> <p>Työn ensimmäisessä osuudessa käydään läpi ohjelmisto-ohjattujen verkkojen teoriaa, ominaisuuksia ja toimintaperiaatetta. Teoriaosuudessa keskitytään tarkemmin OpenFlow-viestintäprotokollaan, joka mahdollistaa ohjelmisto-ohjattujen verkkojen toiminnan sekä eri valmistajien näkemykset tekniikasta.</p> <p>Työn käytännön osuudessa rakennettiin testiverkko Cygaten olemassa olevaan laboratorioverkkoon. Testiverkko koostuu HP:n kytkimistä ja eri valmistajien ohjelmistoratkaisuista. Käytännön osuuden tarkoituksena on selvittää, miten eri ohjelmistoratkaisut eroavat toisistaan sekä saada yleiskuva SDN-verkkojen toiminnasta.</p> <p>Työn tulokset vastasivat alussa määriteltyjä tavoitteita. SDN tarjoaa paljon uusia mahdollisuuksia tietoverkkoihin. SDN on viime aikoina ottanut suuria harppauksia kehityksessä, ja sitä mukaa kerännyt myös paljon kannatusta. On siis nähtävissä, että SDN tulee lähitulevaisuudessa yleistymään yritysverkoissa. Cygate oli tyytyväinen työn tuloksiin ja sai tärkeää tietoa, jota se voi jatkossa käyttää itse tai silloin, kun konsultoi asiakkaitaan.</p>	
Avainsanat	OpenFlow, SDN, HP VAN, OpenDaylight

Author Title	Kim Kurillo Software Defined networking (SDN)
Number of Pages Date	37 pages + 5 appendices 9 November 2014
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Data Networks
Instructor(s)	Jaakko Rautanen, Network Consultant Janne Salonen, Principal Lecturer
<p>The purpose of this bachelor's thesis was to research the features and uses of Software Defined Networking. The thesis was carried out for Cygate Oy, a leading provider of network solutions in the Nordic countries. Cygate Oy design, install and maintain networks, so it is important for the company to be up to date on new network technologies.</p> <p>The first part of the thesis focuses on the theory of Software Defined Networking and the features it offers. Also the thesis explores the OpenFlow communication protocol that is one of the main features that enable SDN. In addition, it reviews different software solutions.</p> <p>The final part of the thesis consists of laboratory tests that were conducted in Cygate's laboratory environment. The SDN test network includes HP switches and a couple of different software solutions. The main purpose of the practical part of the thesis was to find out how SDN works in a real network environment.</p> <p>The results of the thesis met the objectives that were defined at the beginning. SDN presents countless new opportunities in data networks. The recent development and interest in Software Defined Networking indicates that companies will start to adopt it in the near future. Cygate Oy was satisfied with the results and obtained valuable information about Software Defined Networking.</p>	
Keywords	OpenFlow, SDN, HP VAN, OpenDaylight

Sisällys

Lyhenteet

1	Johdanto	1
2	Ohjelmisto-ohjatut verkot	2
2.1	Historia ja nykytilanne	4
2.2	OpenFlow - viestintäprotokolla	5
2.2.1	OpenFlow-kanava ja yhteyden muodostaminen	7
2.2.2	OpenFlow-taulut ja kytkimen toimintaperiaate	9
2.3	SDN-kontrollerit	10
2.4	Northbound-rajapinta	11
3	Käytännön SDN-ratkaisuja	12
3.1	HP VAN -kontrolleri	12
3.2	OpenDaylight-kontrolleri	13
3.3	OpenFlow-kytkimet	15
4	Laboratorioverkon toteutus	16
4.1	Suunnittelu ja tavoitteet	16
4.2	Kytkinten konfigurointi ja päivitys	18
4.3	HP VAN -kontrollerin testit	21
4.4	OpenDaylight-kontrollerin testit	29
5	Tulevaisuus ja jatkokehitys	33
6	Yhteenveto	34
	Lähteet	35

Liitteet

- Liite 1. Kytkimen HP3 esimerkkikonfiguraatio
- Liite 2. Fyysinen verkkotopologia
- Liite 3. Looginen verkkotopologia
- Liite 4. SDN-verkon looginen verkkotopologia
- Liite 5. Mininet verkkotopologian Python-skripti

Lyhenteet

ACL	Access Control List. Pääsyylista, jolla voidaan rajata pääsyoikeuksia liitettyihin kohteisiin.
API	Application Programming Interface. Ohjelmointirajapinta, jonka mukaan ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja keskenään.
CLI	Command-Line Interface. Komentoriviin perustuva käyttöliittymä.
Controller	Tietokoneelle asennettava ohjain, joka hallitsee SDN-verkkoa.
Control Plane	Suomenkos hallintakerros.
Data Plane	Suomenkos tiedonvälityskerros.
HP	Hewlett-Packard, laitevalmistaja.
HP VAN	HP Virtual Application Networks, Hewlett-Packardin kehittämä SDN-ohjain.
Northbound API	Ohjelmointirajapinta, jonka kautta sovellukset keskustelevat SDN-kontrollerin kanssa
OpenDaylight	Linux Foundationin ylläpitämä projekti, jonka tarkoituksena on edistää SDN:ää. OpenDaylight on myös kehittänyt Hydrogen-kontrollerin
OpenFlow	Viestintäprotokolla, joka toimii Southbound-ohjelmointirajapintana kontrollerin ja verkkolaitteiden välillä.
SDN	Software-Defined Networking. Ohjelmisto-ohjatut verkot.

Southbound API	Ohjelmointirajapinta, jonka kautta SDN-kontrolleri välittää käskyjä verkkolaitteille
STP	Spanning Tree Protocol, protokolla, joka estää silmukoiden muodostumisen kytkinverkossa
TCP/IP	Tietoverkkoprotokollien kokoelma, joka vastaa päätelaitteiden osoitteistamisesta ja pakettien reitittämisestä
TLS	Salausprotokolla, jonka avulla voidaan suojata tietoliikennettä verkossa
WAN	Wide Area Network. Laajaverkko.
Rack unit	Standardoitu mittayksikkö, joka kuvaa laitepaikan leveyttä ja korkeutta laitekaapissa. 1U on korkeudeltaan 1,75 tuumaa.

1 Johdanto

Tämän insinööriyön tavoitteena on selvittää ohjelmisto-ohjattujen verkkojen (Software-Defined Networking) toimintaa sekä sen tarjoamia uusia käyttömahdollisuuksia tietoverkoissa. Työn on tilannut Cygate Oy, jossa itse toimin palvelukeskuksessa harjoittelijana tietoliikenneasiantuntijan tehtävissä.

Cygate Oy on osa Cygate-konsernia, jolla on toimintaa Ruotsissa ja Suomessa. Cygaten omistaa TeliaSonera. Konsernilla on noin 550 työntekijää, joista yli 165 työskentelee Suomessa. Cygate suunnittelee, asentaa ja ylläpitää tietoverkkoratkaisuja asiakkailleen, joihin kuuluu eri toimialojen yrityksiä, julkishallintoa sekä internetpalveluiden tarjoajia.

SDN:n tavoite on tehdä verkosta mukautuva, helposti hallittava ja kustannustehokas. Käsitteenä SDN on ollut olemassa jo 90-luvulta asti, mutta etenkin viime vuosina kehitys on ottanut suuren harppauksen, kun vuonna 2011 perustettu Open Networking Foundation on alkanut edistämään SDN-teknologiaa ja OpenFlow-protokollaa.

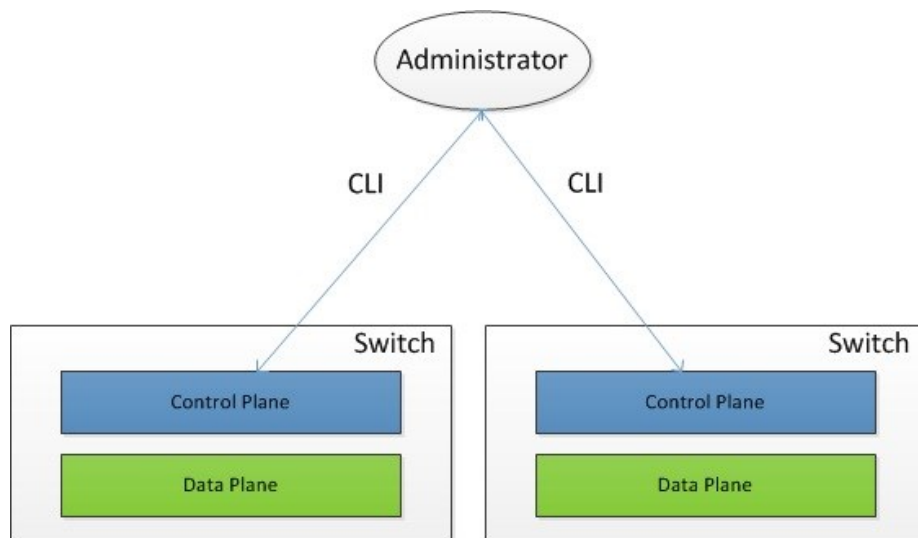
Tämä työ on jakaantunut kahteen pääosaan. Teoriaosuudessa käydään läpi ohjelmisto-ohjattujen verkkojen toimintaperiaatetta, sekä historiaa ja nykytilannetta. Osuudessa käydään myös läpi, millaisia mahdollisuuksia SDN avaa tietoverkoille sekä eri valmistajien omia näkemyksiä kyseisestä tekniikasta.

Insinööriyön toinen pääosa, käytännön osuus, toteutetaan Cygaten laboratoriossa, johon suunnitellaan muutama eri laboratorioverkko käyttäen HP:n verkkolaitteita ja eri valmistajien ohjelmistoratkaisuja. Käytännön osuuden tarkoituksena on selvittää, miten valmistajien näkemykset eroavat toisistaan sekä saada yleistä kuvaa SDN-verkkojen toiminnasta.

Insinööriyön tilaajan päätavoite on selvittää yleisesti, miten SDN-ratkaisut toimivat, sekä voisiko niitä mahdollisesti soveltaa tulevaisuudessa omaan tai asiakkaan verkkoon.

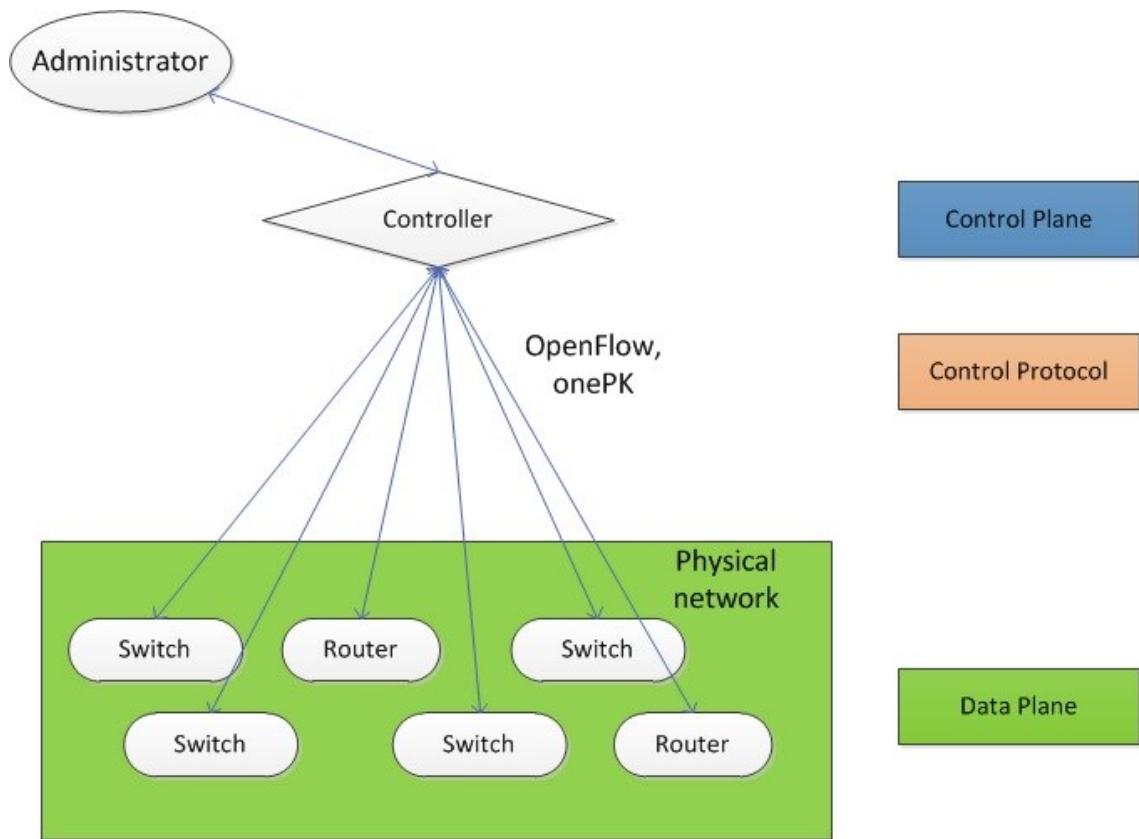
2 Ohjelmisto-ohjatut verkot

Ohjelmisto-ohjatut verkot (Software-Defined Networking, SDN), on lähestymistapa tietoverkkoihin, jotka mahdollistavat ohjelmoinnin näkökulmasta pääsyn verkkolaitteiden sisäiseen toimintaan ja siten mahdollistaa avoimien ohjelmistojen ajamisen aiemmin suljetuilla verkkolaitteilla. Verkkolaitteen sisäinen toiminta perustuu yleisesti hallintakerrokseen (Control Plane) ja tiedonvälityskerrokseen (Data Plane, tunnetaan myös Forwarding plane). Hallintakerroksen tehtävä on hoitaa laskennallinen toiminta eli päättää mm. pakettien ohjaamisesta ja reititysprotokollilla opituista reiteistä sekä hallintayhteyksien ylläpidosta, esimerkiksi SSH:sta. Tiedonvälityskerros sen sijaan välittää verkkolaitteelle tulleita paketteja hallintakerrokselta saamallaan käskyillä. Tämä on "perinteinen" malli, jossa verkkolaite itsenäisesti hoitaa kaiken toiminnan, ns. älyn ja työskenteilyn. Verkkolaitteen hallinta tapahtuu joko CLI:n (Command Line Interface) tai graafisen käyttöliittymän (GUI) kautta, joka on osittain hidasta ja sidottu laitevalmistajan omaan ohjelmistoon (kuva 1). Perinteisesti on ollut olemassa verkonhallintasovelluksia, mutta ne eivät ole yleistyneet kovinkaan laajasti. Verkon konfigurointia hidastaa se, että jokaisen laitteen konfiguraatio käydään laite laitteelta läpi. Nykypäivän monet verkko- ja konesaliympäristöt ovat todella suuria ja monimutkaisia, ja yhden uuden palvelun perustaminen olemassa olevaan ympäristöön voi vaatia kymmenien laitteiden konfigurointia.



Kuva 1: Verkkolaitteen perinteinen hallinta ja toiminta

SDN mahdollistaa sen, että hallintakerros voidaan erottaa fyysisesti erilleen liikennettä välittävästä laitteesta. Täten hallinta voidaan keskittää yhteen paikkaan, kontrolleriin, joka voi olla esimerkiksi virtuaalipalvelimella pyörivä ohjelmisto ja jonka tiedonvälitys jää fyysiselle verkkolaitteelle (kuva 2). Kyseinen toimintamalli helpottaa verkonhallintaa merkittävästi, sillä kontrollerilta voidaan keskitetysti jakaa käskyt koko fyysiselle verkolle.



Kuva 2: SDN-verkon toimintamalli

Jotta kyseinen toiminnallisuus saadaan aikaan, täytyy fyysisen verkkolaitteen ja kontrollerin välillä olla yhteys. Tähän tarkoitukseen tarvitaan jokin viestintäprotokolla, joka mahdollistaa hallintakerroksen ja tiedonvälityskerroksen välisen keskustelun. Yksi tällainen viestintäprotokolla on OpenFlow, joka on avoin standardi ja muotoutumassa yleisimmäksi SDN:n mahdollistavaksi viestintäprotokollaksi.

2.1 Historia ja nykytilanne

1990-luvun puolivälin aikoihin, kun Internetin ja tietoverkkojen nousu oli suuressa kasvussa ja silloisten tiedonsiirtotekniikoiden odotettiin jäävän kehityksen jalkoihin, tuli tutkijoille tarve kehittää ja testata uusia tekniikoita. Tutkijoiden ainoa tapa testaamiseen oli pienissä testiympäristöissä simuloida verkkoja. Ainoastaan suuren kiinnostuksen ja

rahoituksen turvin idea voitiin viedä IETF:lle (Internet Engineering Task Force), jossa se standardoitiin protokollaksi. Monille yliopistoille ja tutkimusryhmille rahoituksen saaminen ei ollut mahdollista, joten uusien ideoiden kehittäminen ei onnistunut. Tämän takia tutkijat alkoivat kehittää uutta tekniikkaa, joka avaisi tietoverkkolaitteiden hallintakerroksen (Control Plane), ja tekisi siitä avoimen ja ohjelmoitavan rajapinnan. Vuonna 1997 yhdysvaltalainen teleoperaattori AT&T alkoi kehittämään ohjelmistopohjaista kytkintä, joka voisi konfiguroida fyysisille verkkolaitteille uusia palveluita. Silloiset testit eivät onnistuneet, koska ”ohjelmistokytkin” ei pystynyt vaikuttamaan kaikkiin fyysisen verkkolaitteen hallintakerroksen toimintoihin, eli ohjelmointirajapinta (API) ei ollut tarpeeksi kehittynyt. [1.]

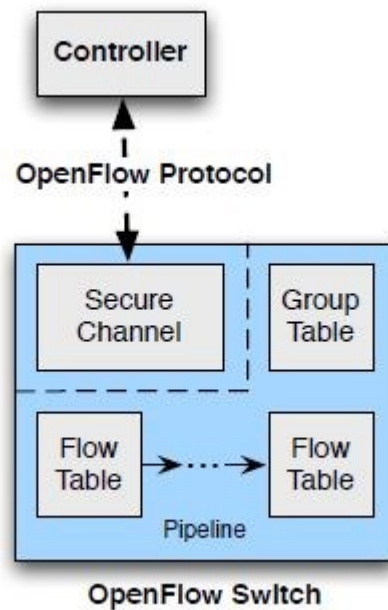
Vuonna 2006 Stanfordin yliopistossa alettiin kehittää Ethane-nimistä projektia, jonka tarkoituksena oli erottaa hallintakerros fyysisestä verkkolaitteesta ja keskittää se kontrollerille sekä luoda niiden välille ohjelmointirajapinta. Vuonna 2008 projektiin liittyi myös UC Berkeleyyn yliopiston tutkijoita ja protokollan nimeksi muotoutui loppujen lopuksi OpenFlow. OpenFlow-protokolla mahdollisti viimein tutkijoille ja kehittäjille pääsyn ohjelmoimaan verkkolaitteiden toimintaa ja keräsi paljon kannatusta. Vuonna 2011 perustettu voittoa tavoittelematon Open Networking Foundation (ONF) alkoi edistämään SDN-teknologiaa ja OpenFlow-protokollaa. Open Networking Foundation on SDN-teknologian ja protokollien standardoinnista vastaava organisaatio, kuten IETF. Vuonna 2013 organisaatiolla oli 123 jäsenyritystä, kuten Cisco, Juniper, HP, Citrix, VMware, Google. SDN:n avaamista mahdollisuuksista ja suosioista kertoo myös se, että Google käyttää omassa runkoverkossaan SDN:ää ja OpenFlow:ta. [1; 2; 3; 4.]

2.2 OpenFlow - viestintäprotokolla

SDN-arkkitehtuurissa tarvitaan jokin ohjelmointirajapinta (API), jonka mukaan kontrollerin ohjelmat ja verkkolaitteet voivat tehdä pyyntöjä ja vaihtaa tietoja keskenään. Tätä kontrollerin ja verkkolaitteen välistä rajapintaa kutsutaan englanniksi Southbound interfaceiksi. Southbound-ohjelmointirajapintoja on useita, mutta vanhin ja vakiintunein on Open Networking Foundationin (ONF) kehittämä OpenFlow. Suurimmat tietoliikenne- ja verkkolaitteiden valmistajat, kuten Cisco, Juniper, HP ja Extreme Networks tuke-

vat OpenFlow-protokollaa omilla laitealustoillaan ja ovat myös mukana protokollan kehityksessä.

OpenFlow-kytkimiä on kahdenlaisia: puhtaasti OpenFlow-protokollalla toimivia sekä hybridikytkimiä. Puhtaalla OpenFlow-kytkimellä tarkoitetaan ohjelmistokytkimiä, jotka eivät ole laitteistoriippuvaisia ja kykenevät ainoastaan OpenFlow-kontrollerin määritteliin operaatioihin, eivätkä sisällä omaa sisäistä hallintarakennetta (control plane). Hybridi-kytkimet sen sijaan tukevat OpenFlow'n lisäksi myös tavallisia Ethernet-kytkimen toimintoja, kuten L2-kytkentää ja L3-reititystä. Käytännössä hybridikytkin toimii siten, että esimerkiksi osa porteista sidotaan OpenFlow-virtuaaliverkkoon, jonka esimerkiksi QoS-ominaisuuksia säädellään SDN-kontrollerilta ja muu toiminta perustuu perinteisiin menetelmiin. OpenFlow-kytkimen sisäinen toiminta koostuu kolmesta osasta: OpenFlow-kanavasta (Secure Channel), OpenFlow-protokollasta sekä OpenFlow-tauluista (eng. Flow-table) (kuva 3).

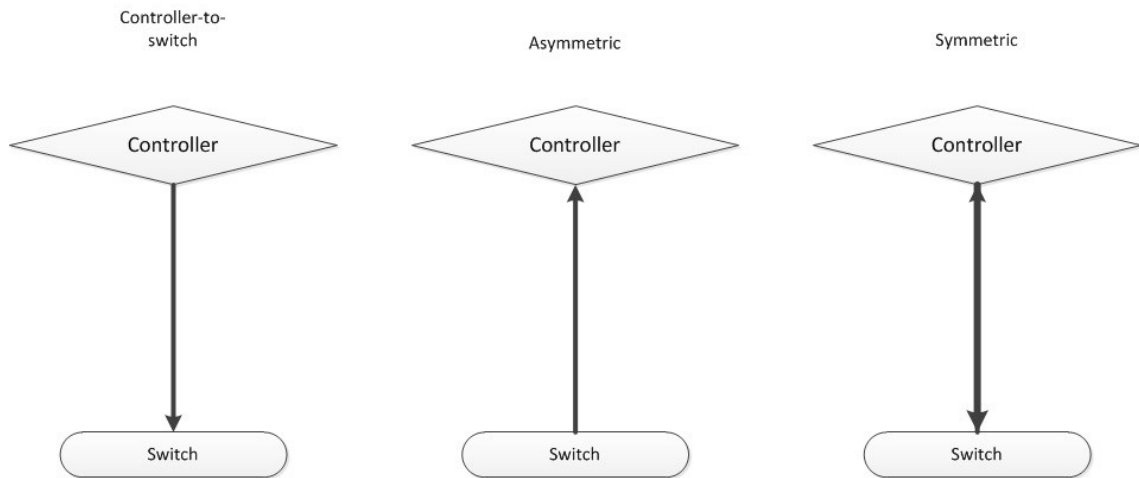


Kuva 3: OpenFlow-kytkimen sisäiset toiminnot

2.2.1 OpenFlow-kanava ja yhteyden muodostaminen

OpenFlow-kanavaksi kutsutaan rajapintaa jokaisen kontrollerin ja kytkimen välillä, jonka kautta kontrolleri hallitsee kytkintä sekä mahdollistaa molemminpuolisen viestinnän. Tyypillisesti OpenFlow-kontrollerilla on useampi yhtäaikainen OpenFlow-kanava useisiin kytkimiin, mutta myös kytkimillä voi olla useampi kanava eri kontrollereihin, mikä mahdollistaa vikasietoisuuden verkossa. OpenFlow-kanava voidaan muodostaa salatuna käyttäen TLS-salausprotokollaa tai salaamattomana TCP-yhteyden yli. Ainoa vaatimus yhteyden muodostamiseen on olemassa oleva TCP/IP-yhteys kytkimen ja kontrollerin välillä. Eli todellisuudessa SDN-verkko tarvitsee rinnalleen erillisen hallintaverkon, jonka kautta yhteys kontrolleriin voidaan muodostaa. Eri laitevalmistajilla on myös tähän ratkaisuja, kuten HP:n Comware-sarjan kytkimissä tuettu ”OpenFlow out of band management vlan”.

OpenFlow-protokolla tukee kolmea erilaista viestityyppiä, joilla kontrollerin ja kytkimen välisiä tietoja jaetaan (kuva 4). Kontrollerilta kytkimille menevissä viesteissä kontrolleri voi asettaa kytkimelle uusia konfigurointiparametrejä, muuttaa kytkimen OpenFlow-taulujen sisältöä tai lukea tietoja kytkimen ominaisuuksista. Asymmetrisissä viesteissä kytkin aloittaa liikenteen kontrollerista välittämättä, esimerkiksi jos verkossa tapahtuu muutos, joka vaikuttaa kytkimen portin tilaan, välittyy tieto tapahtumasta kontrollerille. Asymmetrisiä viestejä käytetään myös virheilmoitusten välittämiseen. Symmetristä viestintää käytetään OpenFlow-kanavan muodostamisen yhteydessä, jolloin sekä kontrolleri että kytkin lähettävät Hello-paketteja. Myös yhteyden ylläpitoon käytetään symmetristä viestintää.

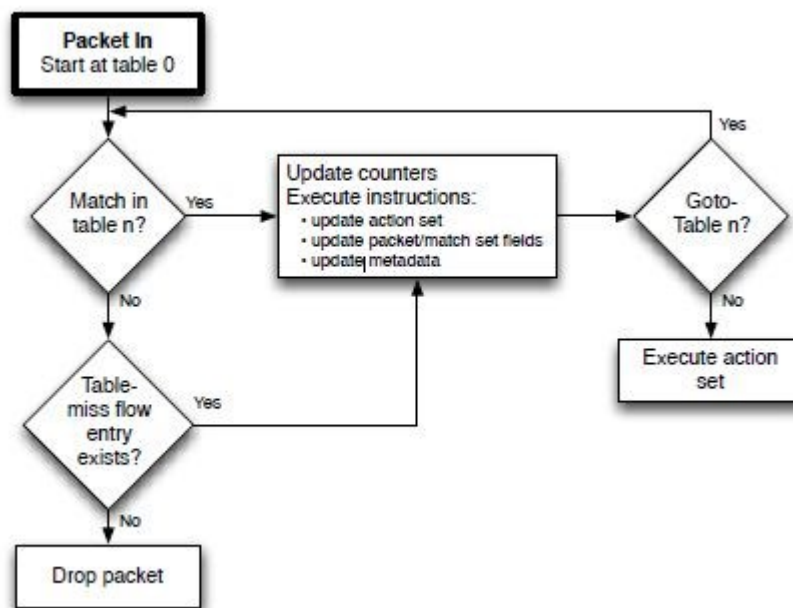


Kuva 4: OpenFlow-kanavan eri viestinnän muodot

OpenFlow-kanavan muodostamisen aloittaa aina kytkin, ja käyttää vakiona TLS-yhteyttä sekä vakioporttia TCP 6633 (OpenFlow versio 1.3). Tulevaisuudessa OpenFlow-kytkimet tulevat käyttämään vakiona porttia TCP 6653, joka on IANA-rekisteröity. Kytkimen on tiedettävä kontrollerin IP-osoite, jotta yhteyden muodostaminen voidaan aloittaa. Kun yhteys on muodostettu, molemmat osapuolet lähettävät Hello-paketin toisilleen, jossa on määritelty viimeisin tuettu OpenFlow-protokollan versio, joka käyttää symmetristä viestintää. Kun osapuolet ovat sopineet käytettävästä OpenFlow-protokollan versiosta, OpenFlow-kanava on muodostettu. Mikäli kytkin ja kontrolleri eivät tue samoja OpenFlow-protokollan versioita tai olemassa oleva OpenFlow-kanava menettää yhteyden, niin kytkin joutuu vikatilaan. OpenFlow-kytkimellä on kaksi määriteltyä vikatilaa: "fail secure mode" ja "fail standalone mode". Kun kytkin on "fail secure mode:ssa", niin se tiputtaa kaiken liikenteen, joka on määritetty kontrollerille meneväksi. OpenFlow-liikenne eli kytkimille määritetyt vuot jatkavat toimintaa, vaikka yhteys kontrolleriin on menetetty. Vuolle määritetään sitä luotaessa timeout-arvo, jonka mukaan se vanhenee ja siten poistuu kytkimen vuotaulusta. Hybridi-kytkimet voivat edelleen jatkaa toimintaa, vaikka ne ovat "fail standalone mode":ssa, koska ne tukevat tavallisia Ethernet-kytkimen toimintoja eivätkä ole täysin riippuvaisia kontrollerista. [5.]

2.2.2 OpenFlow-taulut ja kytkimen toimintaperiaate

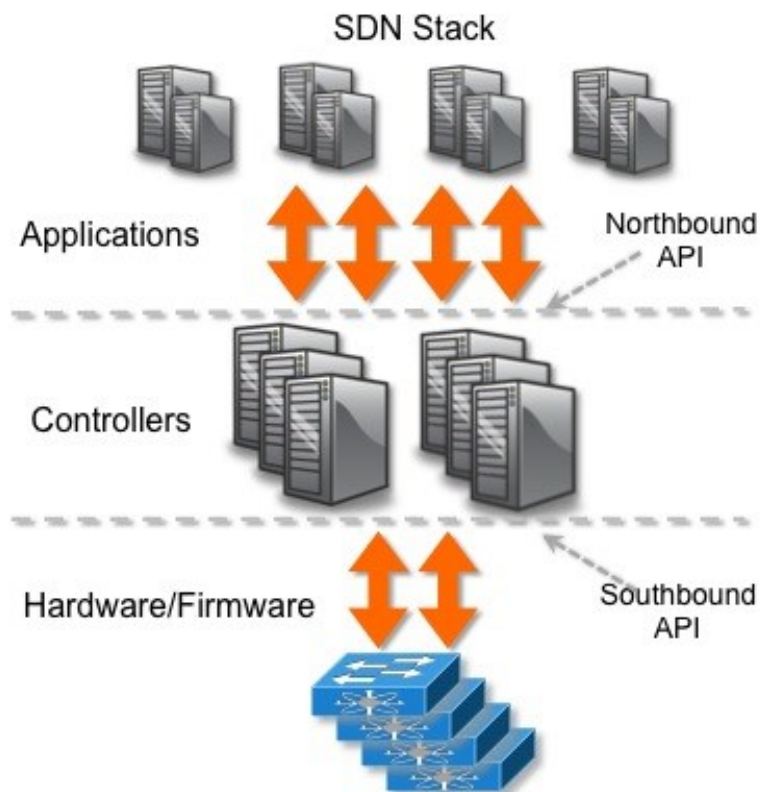
OpenFlow-kytkimen sisäinen pakettien välittäminen tapahtuu tiedonkäsittelykanavassa (eng. OpenFlow pipeline). Kytkimen tiedonkäsittelykanava sisältää yhden tai useamman vuotaulun (eng. Flow table), joka sisältää useita merkintöjä ja toimenpiteitä, joita saapuville paketeille suoritetaan (kuva 5). Ensimmäinen vuotaulu on aina numeroltaan 0 ja tiedonkäsittelykanavaan tulevien pakettien käsittely aloitetaan aina siinä. Pakettia verrataan vuotaulun sisältämiin merkintöihin ja sille voidaan suorittaa taulussa määriteltäviä toimintoja, kuten siirtää paketti käsiteltäväksi vuotauluun n tai vaikka ohjata ulos kytkimen fyysisestä portista. Kun paketti on käynyt läpi kaikki vuotaulut tiedonkäsittelykanavassa, eikä se ole sopinut mihinkään toimintoon, niin paketti voidaan ohjata kontrollerille käsiteltäväksi tai se voidaan suoraan tiputtaa pois.



Kuva 5: OpenFlow-tiedonkäsittelykanavan toiminta

2.3 SDN-kontrollerit

SDN:n kokonaisuutta voidaan ajatella pinona (kuva 6), jossa kontrolleri toimii SDN-verkon ytimenä, ja sen alapuolella on verkkolaitteet ja yläpuolella sovellukset. Edellisessä luvussa käsitelty OpenFlow-protokolla toimii Southbound-ohjelmointirajapintana, jonka kautta kontrolleri välittää käskyjä kytkimille ja reitittimille. SDN-pinon päällä sijaitsevat sovellukset keskustelevat kontrollerin kanssa käyttäen Northbound-ohjelmointirajapintaa ja tarjoavat verkolle palveluita, kuten kuormantasausta. Käytännössä Southbound- ja Northbound-ohjelmointirajapinnat mahdollistavat verkon eri komponenttien välisen keskustelun. [6.]



Kuva 6: SDN-pino [24]

SDN-kontrollerit ovat yleensä modulaarisia ja sisältävät erilaisia toimintoja, kuten verkkotopologian ja statistiikan hahmottamista. Kehittyneempiä sovelluksia voidaan ohjelmoida kontrollerille käyttäen Northbound-ohjelmointirajapintaa. Vuonna 2012

kontrollereita oli saatavilla tai kehityksessä yli 20, ja nykyään määrä on vain lisääntynyt. Kontrollereiden välillä on paljon eroavaisuuksia erityisesti Northbound-rajapinnan kanssa, joka on osaltaan vaikuttanut sovelluskehityksen hitauteen. Suurin osa kontrollereista tukee OpenFlow-protokollaa, mutta on olemassa myös muita Southbound-protokollia, esimerkiksi XMPP ja NETCONF, joita Juniperin Contrail-kontrolleri käyttää. [7.]

2.4 Northbound-rajapinta

Ohjelmisto-ohjattujen verkkojen yksi päätavoitteista on mahdollistaa avointen sovellusten ajaminen verkkolaitteilla. Tähän tarkoitukseen SDN käyttää Northbound-ohjelmointirajapintaa, jonka ansiosta verkkosovellusten ei tarvitse enää olla laitevalmistajariippuvaisia. Avoin ohjelmointirajapinta takaa sen, että kuka tahansa voi luoda ja muokata sovelluksia omiin tarpeisiinsa; käyttäjiä voivat olla esimerkiksi suuret verkko-operaattorit tai yliopistot. [8.]

Kuten edellisessä luvussa mainittiin, kontrollereita on saatavilla yli 20, joista monet käyttävät eri Northbound-rajapintaa. HP VAN- ja OpenDaylight-kontrollerit käyttävät molemmat REST-ohjelmointirajapintaa, joka perustuu HTTP-protokollaan ja on yleisesti käytössä. Koska SDN:llä ei ole standardoituja Northbound-protokollia ainakaan toistaiseksi, erityisesti sovelluskehitys on hidasta. Ohjelmistokehittäjille on riski aloittaa kehitys alustalle, joka ei välttämättä kerää tarpeeksi kannatusta ja käyttäjiä. Open Networking Foundation perusti vuonna 2012 työryhmän, jonka tarkoituksena on kerätä tietoa eri ohjelmointirajapinnoista, niiden käyttötarkoituksista sekä antaa käytännön esimerkkejä sovelluksista. On mahdollista, että SDN ei saa vielä standardeja lähiaikoina tai ei välttämättä milloinkaan. OpenFlow-protokollan kehittäjän, Martin Casadon mukaan, SDN on vielä niin alkutekijöissä, että olisi liian aikaista lähteä standardoimaan asioita. Kunhan SDN yleistyy ja saadaan enemmän käytännön esimerkkejä tuotantoverkoista, voidaan alkaa miettiä standardoimista. [9; 10.]

3 Käytännön SDN-ratkaisuja

3.1 HP VAN -kontrolleri

Ensimmäiseksi testikontrolleriksi valittiin HP VAN (Virtual Application Networking), joka on HP:n SDN-ekosysteemin ydin. SDN-ekosysteemin tarkoituksena on tarjota avoin kehitysympäristö ja työkalut, joiden avulla käyttäjät ja yritykset voivat kehittää kontrollerille uusia sovelluksia. Kontrolleri pohjautuu Linux/Java/OSGI-komponentteihin ja Northbound-ohjelmointirajapintana käytetään REST:iä, jonka avulla sovellukset kommunikoi- vat kontrollerin kanssa. Southbound-rajapintana HP VAN tukee ainoastaan Open-Flow'ta, ja uusimmalla ohjelmistoversiolla OpenFlow 1.0:a ja 1.3:a. Kontrolleri voi toi- mia itsenäisesti, tai se voidaan asettaa enintään kolmen kontrollerin ryhmään, joka lisää SDN-verkon vikasietoisuutta.

HP VAN -kontrollerin suositellut järjestelmävaatimukset noin 200 laitteen testiympäris- töön:

- 3,0 GHz x86-64 quad core processor or equivalent
- 8 GB RAM
- 40 GB disk space
- One 1 Gbps Ethernet NIC
- Software requirements: Ubuntu 12.04 LTS 64-bit Server with Java 7

Kontrolleri ei vakiona eli ns. "out of the boxina", sisällä kovin montaa applikaatiota. Va- kiokomponentteihin ja toimintoihin sisältyy mm. L2-kytkentä ja verkkotopologian hah- mottaminen, mutta esimerkiksi L3-reititykseen, HP VAN ei kykene vakiona (kuva 7).

Resource	HP VAN SDN Controller versions 2.2 and 2.3
Controller team size	3
Maximum connected OpenFlow devices (single controller)	4000

Maximum connected OpenFlow devices (controller team)	4000
Maximum number of connected hosts	14 000
Maximum connected OpenFlow ports (single controller)	50 000
Maximum connected OpenFlow ports (controller team)	50 000
Flows (packets) per second (single controller)	2,3M
Flows (packets) per second (controller team)	6,5M

Taulukko 1: HP VAN -kontrollerin maksimikonfiguraatiot [11.]

General / Applications				
New	Upgrade	Uninstall	Enable	Disable
Name	Version	State		
▶ Path Diagnostics	2.2.5	ACTIVE		
▶ Link Manager	2.2.5	ACTIVE		
▶ Node Manager	2.2.5	ACTIVE		
▶ Path Daemon	2.2.5	ACTIVE		
▶ Topology Manager	2.2.5	ACTIVE		
▶ Topology Viewer	2.2.5	ACTIVE		

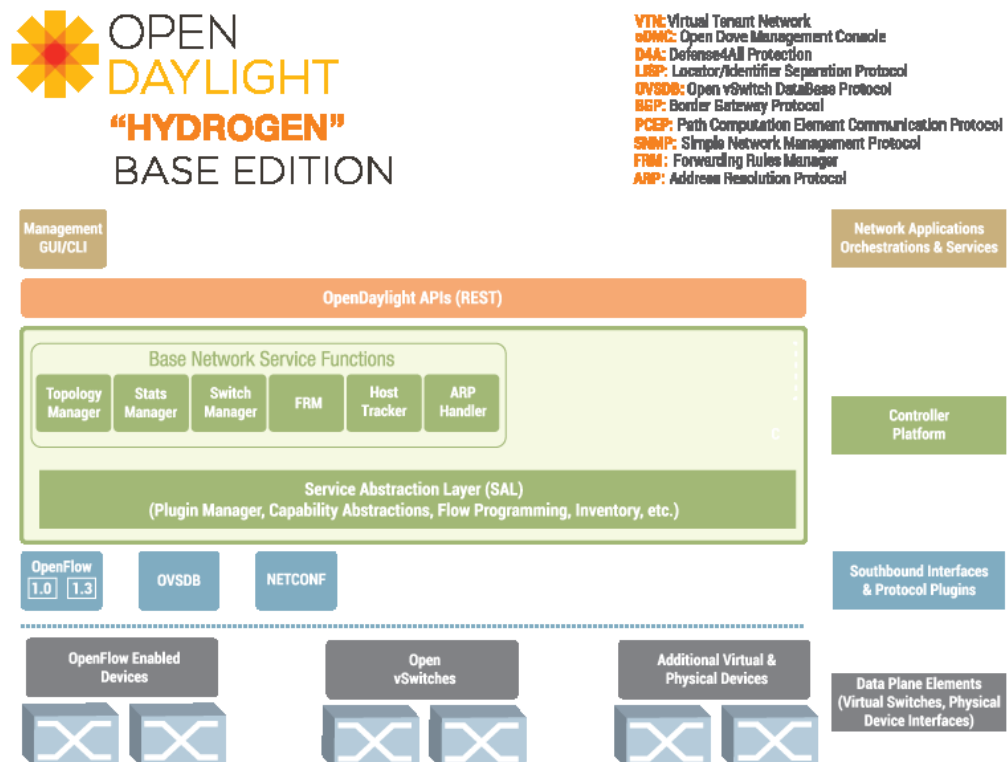
Kuva 7: HP VAN -kontrollerin vakiosovellukset

3.2 OpenDaylight-kontrolleri

Toiseksi kontrolleriksi valittiin avoimeen lähdekoodiin perustuva OpenDaylight-projektin kehittämä Hydrogen-kontrolleri. OpenDaylight on Linux Foundationin ylläpitämä yhteistyöprojekti, jonka tarkoituksena on edistää SDN:n yleistymistä. Projekti toimii avoimena ja yhteisövetoisena, jonka jäsenenä on alan suurimpia yrityksiä kuten Cisco, Juniper, IBM, HP ja Microsoft. Jäsenyritykset tukevat projektia kiinnittämällä omia verkko- ja ohjelmistoinsinöörejään OpenDaylightin kehitykseen ja ansaitsevat siten korkeampia jäsenyystasoja, joita ovat platina, kulta, hopea. Projekti on saanut hyvän vastaanoton yhteisöltä etenkin Linux Foundationin tuen ansiosta, mutta myös huolenaiheena on,

että suurimmat vaikuttajat, kuten Cisco ja Juniper hidastaisivat innovaatiota säilyttääkseen oman markkina-asemansa. [12.]

OpenDaylight-projekti julkaisi ensimmäisen kontrollerinsa, Hydrogenin, helmikuussa 2014, vain vuoden projektin perustamisen jälkeen. Kontrolleri on modulaarinen ja perustuu Java/OSGI-komponentteihin sekä käyttää Northbound-ohjelmointirajapintana RESTiä. Hydrogen-kontrollerista on saatavilla 3 eri julkaisupakettia, jotka on suunnattu erilaisille käyttäjäryhmille ja sisältävät monenlaisia sovelluksia: Base Editionin, Virtualization Editionin ja Service Provider Editionin. Base Edition on tarkoitettu käyttäjille, jotka haluavat testata SDN:ää ja OpenFlow:ta yleisesti. Virtualization- ja Service Provider -julkaisupaketit sisältävät muun muassa kuormantasaukseen ja tietoturvaan liittyviä sovelluksia. Kaikki Hydrogen-kontrollerin julkaisupaketit tukevat Southbound-ohjelmointirajapintana OpenFlow:n versioita 1.0 ja 1.3, sekä OVSDB- ja NETCONF-protokollia (kuva 8). [13.]



Kuva 8: Hydrogen-kontrollerin Base Editionin arkkitehtuuri [13.]

3.3 OpenFlow-kytkimet

Laboratorioverkon toteutukseen päätettiin käyttää HP:n 5900AF-48G-4XG-2QSFP+ Switch (JG510A) -kytkimiä. HP on vahvasti mukana SDN:n kehitystyössä, ja se näkyy myös laitevalmistajan markkinoilla olevissa kytkimissä. Uusimpiin ProVision-, ProCurve- ja Comware-sarjojen kytkimiin on saatavilla OpenFlow-tuki ja myös vanhempia kytkimiä voidaan päivittää tukemaan OpenFlow'ta. Comwaren HP5900-sarjan kytkimet valittiin testilaitteiksi, koska ne tukevat OpenFlow:n versiota 1.3, joka on tällä hetkellä viimeisin laitevalmistajien tukema versio. Myös HP VAN- ja OpenDaylight-kontrollerien uusimmat versiota tukevat OpenFlow 1.3ta.

HP 5900 (JG510A) -kytkimet ovat ns. Top-of-Rack-kytkimiä (ToR), mikä tarkoittaa, että ne ovat porttimäärältään pieniä ja yleensä kooltaan 1-2U:ta (rack unit). Nimensä mukaisesti ToR-kytkimet on tarkoitettu sijoitettavaksi laiteräkin yläosaan, josta ne tarjoavat yhteydet laiteräkkiin kiinnitetyille palvelimille, ja ovat ideaaleja erityisesti datakeskuksille.

Kytkimen tarkemmat tiedot:

- Ports: 48 autosensing 10/100/1000 ports. 4 fixed 1000/10000 SFP+ ports. 2 QSFP+ 40-GbE ports
- Additional Ports and slots: 1 RJ-45 console port. 1 RJ-45 out-of-band management port. 1 USB 2.0 port.
- Power: 2 power supply slots.
- Dimensions: 1 Rack unit (U) height.
- Weight: 13kg
- Routing / Switching capacity: 336 Gb/s.

4 Laboratorion toteutus

SDN-verkon toteutus tehtiin Cygaten tietoverkkolaboratorion tiloissa, yhteistyössä tietoliikennekonsultti, Jaakko Rautasen kanssa. Tässä luvussa kerrotaan, miten laboratorion verkko suunniteltiin ja konfiguroitiin, sekä käsitellään työn tavoitteet. Osa tässä luvussa mainituista laitenimistä ja IP-osoitteista on muutettu tietoturvasyistä.

4.1 Suunnittelu ja tavoitteet

Työn tavoitteena oli selvittää, miten ohjelmisto-ohjatut verkot toimivat käytännössä, sekä testata muutamia tällä hetkellä tarjolla olevia kontrollereita ja sovelluksia. Verkon testeihin lukeutuivat aliverkon sisäinen kytkentä, aliverkkojen välinen reititys sekä SDN-verkon liittyminen ulkomaailmaan eli Internetiin. Verkolle tehtiin myös vikatilannetestejä, joiden avulla pyrittiin selvittämään, miten controllerit ja sovellukset toimivat vikatilanteen sattuessa. Laboratoriotestit suoritettiin HP VAN- ja OpenDaylight Hydrogen-kontrollereilla. Hydrogen-controlleri sisältää vakiona kaikki komponentit ja toiminnot, joita työssä haluttiin testata, mutta HP VAN -controllerin testeihin otettiin mukaan myös erillinen sovellus. SDN-testiverkolle määritellyt tavoitteet ja testit:

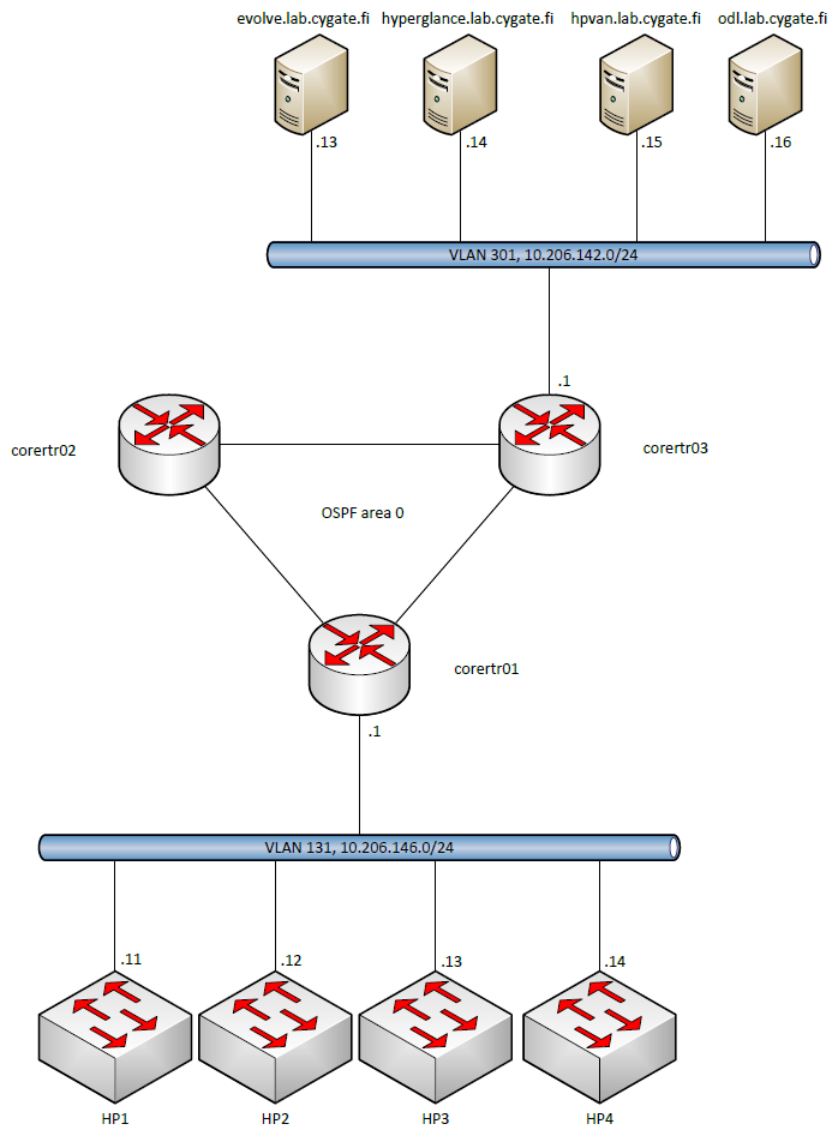
- SDN-verkon sisäinen hallintaverkko (eng. in-band management)
- SDN-verkon toimivuus, kun kytkimet käyttävät STP:tä eli Spanning Tree Protocol
- SDN-verkon toimivuus, kun controlleri hoitaa silmukan poiston verkossa
- vuon (eng. flow) luominen manuaalisesti, jossa ei ole aikakatkaisua
- aliverkon sisäiset yhteydet
- vuon luominen manuaalisesti aliverkkojen välille
- SDN-verkon reititys
- vikatilannetesti: SDN-verkon sisäisen linkin katkaiseminen
- vikatilannetesti: Controllerin yhteyden katkaiseminen.

SDN-verkko liitettiin Cygaten laboratorioverkkoon. Cygaten laboratorioverkko koostuu reitittävästä runkoverkosta ja sen takana olevista useista VLANeista, jotka on jaettu muun muassa asiantuntijoiden käyttöön. Laboratorioverkkoon on myös liitetty VMware ESX -virtuaalipalvelinalusta, johon SDN-kontrollerit ja sovellukset asennettiin. Työssä käytetyt verkot eli VLAN:it:

- VLAN 10, sisäinen SDN-verkko työasemille
- VLAN 131, kytkimien hallintaverkko
- VLAN 139, reititetty SDN-verkko
- VLAN 301, verkko virtuaalipalvelimille.

Näistä verkoista VLAN:it 131, 139 ja 301 on reititetty laboratorioverkossa käyttäen OSPFää eli Open Shortest Path Firstiä, reititysprotokollaa. OSPF on dynaaminen reititysprotokolla, joka käyttää Dijkstran algoritmia määrittääkseen optimaalisimman reitin. OSPF:än käyttäminen oli työn kannalta helpoin toteutus, sillä olemassa olevaan runkoverkkoon liittyminen ei vaatinut konfigurointia. [14.]

Työasemia varten luotu VLAN 10 on konfiguroitu ainoastaan SDN-verkon kytkimille, eikä sitä mainosteta laboratorion runkoverkolle. Kuvassa 9 näkyy verkon looginen toteutus hallintaverkon ja virtuaalipalvelinten osalta. Seuraavassa alaluvussa käsitellään tarkemmin kytkinten konfigurointia sekä kytkinten ja kontrollerin välistä yhteyttä. [14.]

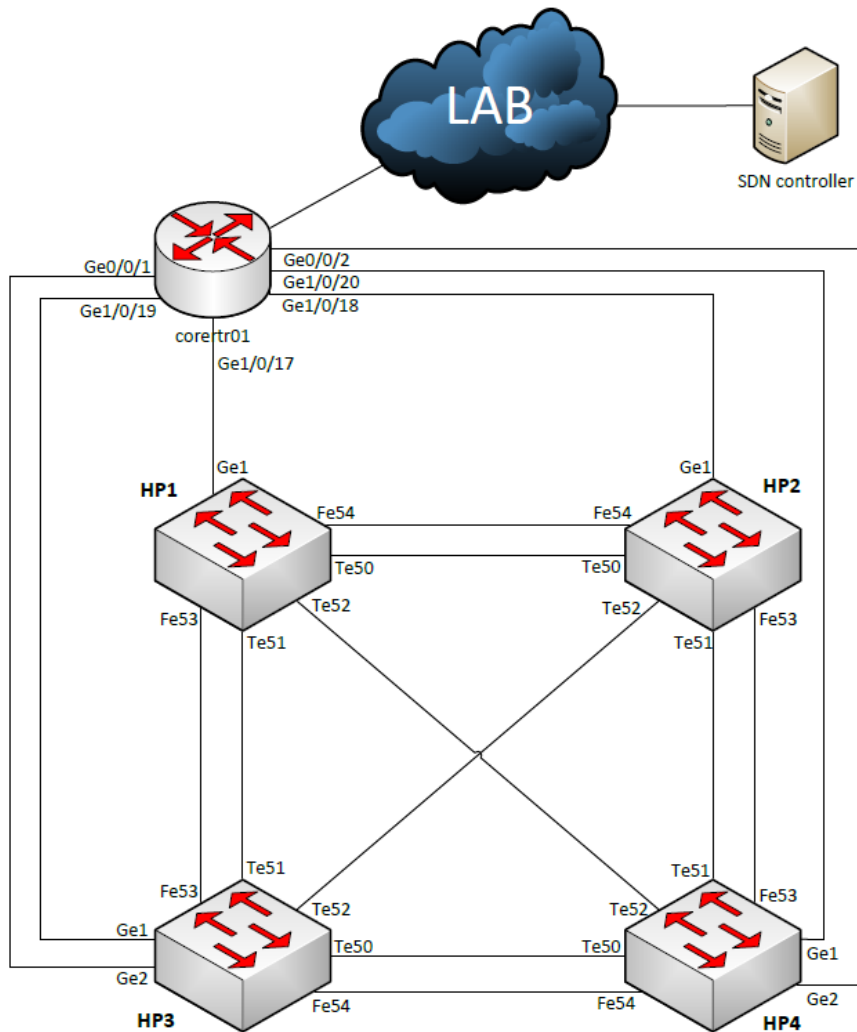


Kuva 9: Looginen verkkotopologia

4.2 Kytöinten konfigurointi ja päivitys

Tässä aluvussa käsitellään, miten kytkimet on liitetty toisiinsa ja yhdistetty laboratorioverkkoon, sekä käydään läpi kytkimen ja SDN-verkon perusasetukset. Kytkiminä käytettiin neljää 54-porttista HP:n 5900-kytkintä, jotka sisältävät kytkinominaisuuksien lisäksi myös L3-tason reititysominaisuudet. Kytkimet liitettiin toisiinsa useammalla toisiinsa vahvistavilla linkeillä, jotta verkkoon saatiin vikasietoisuutta, mutta myös sen takia,

että kytkimille saatiin luotua SDN-verkon ulkopuolinen hallintaverkko. Kytkinten väliset linkit ovat valokuitukaapelia, ja työasemien sekä runkoverkkoon liittyminen on toteutettu kuparikaapelilla (kuva 10).



Kuva 10: Fyysinen verkkotopologia [14.]

Kytkimien asennuksen yhteydessä kytkimille tehtiin peruskonfiguraatio, johon kuului järjestelmän asetukset, etähallinnan asetukset ja salliminen sekä käyttäjätilin luonti. Kytkimille määritettiin hallintaosoitteet (kuva 9), joiden kautta saatiin SSH-yhteydet kytkimille. Hallintayhteyden lisäksi kytkimet liitettiin myös laboratorioverkossa sijaitsevaan

konsolipalvelimeen, jonka kautta kytkimille saatiin myös yhteys vikatilanteiden sattuesssa.

SDN-verkko liitettiin hallintaverkkoon, VLAN 131, yhden kytkinportin kautta runkoreitittimelle, sillä OpenFlow-kytkimiä ei saa kytkeä silmukkaan SDN-verkon ulkopuolisten kytkinten kanssa. Liittyminen runkoverkkoon toteutettiin kytkimeltä HP1. [15.]

Hallintayhteysportit konfiguroitiin kytkimille komennoilla:

```
interface GigabitEthernet1/0/1
  port link-mode bridge
  port access vlan 131
```

Hallintayhteyden konfiguroinnin jälkeen kytkimet saivat yhteyden koko laboratorioverkkoon ja samalla myös virtuaalipalvelinalustaan, jossa SDN-kontrollerit sijaitsivat. Kytkimille konfiguroitiin OpenFlow-tila (eng. instance), jossa määriteltiin, mihin VLAN:iin OpenFlow sallitaan sekä IP-osoite, josta kontrolleri löytyy. OpenFlow'n peruskonfiguraatio tehtiin seuraavilla komennoilla:

```
openflow instance 10
  classification vlan 10
  controller 1 address ip x.x.x.x
```

OpenFlow-tilan konfiguroinnin jälkeen ja yhteyden muodostuttua kontrolleri säätelee kaikkea VLAN 10:n liikennettä. Jotta OpenFlow-liikenne SDN-verkossa ja työasemilla toimisi, tarvitsi kytkinten väliset linkit ja työasemaportit liittää VLAN 10:een. Kytkinten väliset runkoportit (eng. trunk) konfigurointiin komennoilla:

```
interface Ten-GigabitEthernet1/0/50
  port link-mode bridge
  port link-type trunk
  port trunk permit vlan 1 10
  port trunk pvid vlan 10
```

Työasemaportit komennoilla:

```
interface GigabitEthernet1/0/2
  port link-mode bridge
  port access vlan 10
```

Yksi työn tavoitteista oli testata STP-protokollan toimintaa SDN-verkossa. STP:n tehtävä on estää silmukoiden syntyminen L2-tason verkossa. STP asettaa linkkejä estävään tilaan, mikäli ne aiheuttaisivat silmukan verkossa. Jos verkossa tapahtuu muutos, jonka seurauksena topologia muuttuu, STP laskee automaattisesti uuden topologian ja tarvittaessa sallii tai estää linkkejä. Vakiona SDN-verkon kytkimet konfiguroitiin käyttämään STP:tä, mutta molemmat kontrollerit osaavat myös muodostaa silmukattoman verkko-topologian käyttäen lyhimmän reitin algoritmia (eng. Shortest Path First), joten molemmat tavat testattiin laboratorioissa. Kytkimille konfiguroitiin STP-komennoilla:

```
stp vlan 10 enable
  stp mode pvst
  stp global enable
```

STP otettiin pois käytöstä komennolla:

```
undo stp vlan 10 enable
```

4.3 HP VAN -kontrollerin testit

HP VAN -kontrollerin osalta laboratoriotesteihin otettiin mukaan erillinen SDN-sovellus, Hyperglance, jonka avulla vuon (eng. flow) luominen manuaalisesti onnistuu. Hyperglancen ominaisuuksiin kuuluu myös 3D-topologian hahmottaminen ja kehittyneempi vuonhallinta, joiden avulla SDN-verkonhallinta helpottuu.

SDN-verkon sisäinen hallintaverkko

HP:n Comware-sarjan kytkimillä on ominaisuutena sallia OpenFlow-tilan sisäinen hallintayhteys. Sisäinen hallintayhteys helpottaa verkkotopologian suunnittelua siten, että erillistä hallintayhteyttä ei tarvitse viedä jokaiselle SDN-kytkimelle erikseen, vaan hallintaverkko voidaan liittää SDN-verkon reunalla olevaan kytkimeen. OpenFlow-tilaan määritellään SDN-aliverkkojen lisäksi parametri "in-band management vlan x", jolloin erillinen hallintaverkko sallitaan SDN-verkon rinnalle.

```
openflow instance 10
classification vlan 10
in-band management vlan 131
controller 1 address ip x.x.x.x
active instance
```

SDN-verkon sisäistä hallintaverkkoa ei saatu testeissä toimimaan, sillä OpenFlow-tila ei sallinut konfiguraatiossa erillistä aliverkkoa, joka ei ole yhteydessä kontrolleriin. Testien tulosten takia päädyttiin ratkaisuun, jossa muodostettiin SDN-verkon ulkopuolinen hallintayhteys eli lisättiin kytkimien välille linkit, joissa sallittiin hallintaverkko vlan 131.

```
interface FortyGigE1/0/53
port link-mode bridge
description HP1_port_1/0/53
port access vlan 131
```

Spanning tree -testit

Molemmat laboratorioverkkoon valituista kontrollereista sisältävät ominaisuutena silmukan poiston, joten testiin otettiin SDN-verkon toimivuus, kun kontrolleri laskee silmukattoman verkon sekä kun kytkimet käyttävät STP-protokollaa.

Spanning tree poistettiin kytkimien konfiguraatiosta, jolloin kontrolleri laskee uuden silmukattoman topologian:

```
<HP3>display current | include stp
undo stp vlan 10 enable
stp mode pvst
stp global enable
<HP3>
```

Tarkistettiin kytkimen spanning tree vlan 10:n osalta:

```
<HP3>display stp brief
VLAN ID   Port                               Role STP State   Protection
131       FortyGigE1/0/53                   Desi Forwarding None
131       FortyGigE1/0/54                   Desi Forwarding None
```

Varmistetaan vielä kytkimen OpenFlow-tilan määrittymiset:

```
<HP3>display openflow instance 10
Instance 10 information:

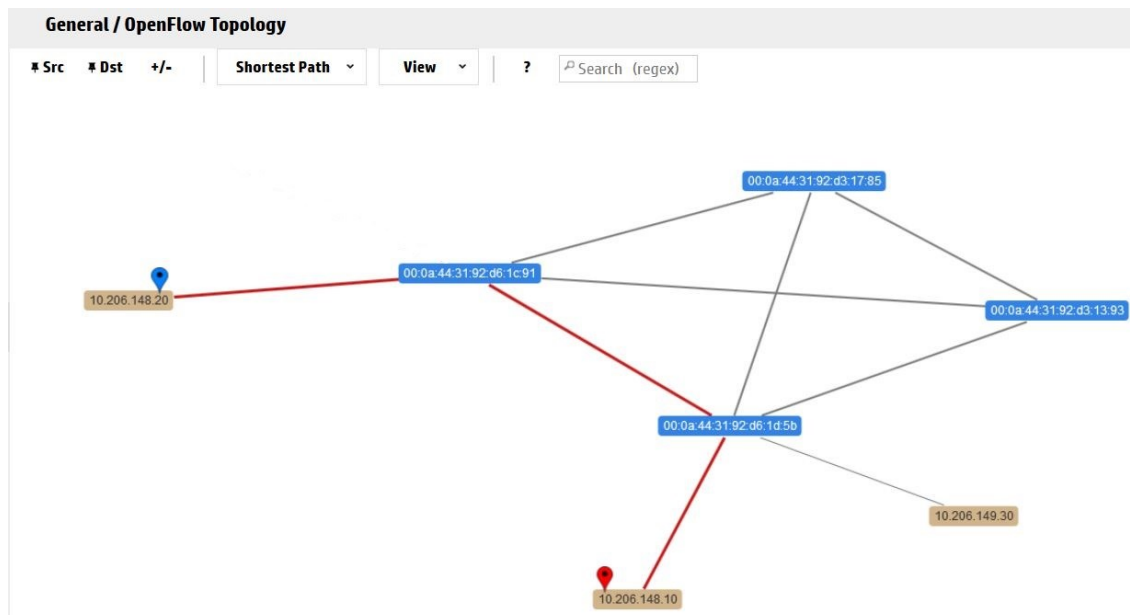
Configuration information:
  Description      : vlan10
  Active status   : active
  Inactive configuration:
    none
  Active configuration:
    Classification VLAN, total VLANs(1)
      10
    In-band management VLAN, total VLANs(0)
      empty VLAN
  Connect mode: multiple
  MAC address learning: Enabled
  Flow table:
    Table ID(type): 0(Extensibility), count: 5
  Flow-entry max-limit: 65535
```

```

Datapath ID: 0x000a443192d61d5b
Port information:
  GigabitEthernet1/0/1
  GigabitEthernet1/0/2
  Ten-GigabitEthernet1/0/50
  Ten-GigabitEthernet1/0/51
  Ten-GigabitEthernet1/0/52
Active channel information:
  Controller 1 IP address: 10.206.142.15 port: 6633
<HP3>

```

Kontrollerin graafisesta käyttöliittymästä nähdään kaikki kontrolleriin liitetyt laitteet sekä IP- ja MAC-osoitteilla, ja niitä voidaan liikutella työpöydällä. Kontrollerilta voidaan nähdä lyhin reitti (eng, Shortest Path) kahden pisteen välille, tässä tapauksessa kahden SDN-verkon työaseman välillä (kuva 11). Kontrolleri osaa siis muodostaa SDN-verkon kahden työaseman välille reitin sekä tehdä verkosta silmukattoman.



Kuva 11: HP VAN graafinen käyttöliittymä ja SPF-algoritmin käyttö

Seuraavaksi kytkimille määritettiin spanning tree:

```
[HP3]stp vlan 10 enable
```

Kytkimet neuvottelevat SDN-aliverkon vlan 10 uuden silmukattoman topologian, ja sen toiminta tarkistettiin:

```
<HP3>display stp brief
```

VLAN ID	Port	Role	STP State	Protection
10	GigabitEthernet1/0/1	Desi	Forwarding	None
10	GigabitEthernet1/0/2	Desi	Forwarding	None
10	Ten-GigabitEthernet1/0/50	Desi	Forwarding	None
10	Ten-GigabitEthernet1/0/51	Desi	Forwarding	None
10	Ten-GigabitEthernet1/0/52	Desi	Forwarding	None
131	FortyGigE1/0/53	Desi	Forwarding	None
131	FortyGigE1/0/54	Desi	Forwarding	None

Kytkimen HP3 osalta topologia ei muuttunut, sillä kytkin valittiin juurikytkimeksi, jonka kaikki portit välittävät liikennettä. Spanning tree -testien suorittamisen jälkeen voitiin todeta, että SDN-verkon silmukan poisto voidaan toteuttaa sekä kontrollerilla että kytkimillä.

Aliverkon sisäiset yhteydet

Yksi kytkinverkon tärkeistä ominaisuuksista on aliverkon sisäiset yhteydet eli L2-kytkentä. Työn kannalta oli tärkeää selvittää, pystyykö SDN:llä toteuttamaan yksinkertaisen kytkinverkon ja tarjoamaan yhteydet työasemille sekä myös samalla selvittää verkohallintaan liittyviä kysymyksiä.

SDN-verkkoon luotiin kaksi aliverkkoa ja molempiin kaksi toisistaan erillään olevaa työasemaa, jotta liikennettä voitiin tarkkailla mahdollisimman tarkasti. Luotaessa liikennettä työasemalta host1 (IP: 10.206.148.10/24) kytkimen SDN-verkkoon liitetty portti nappaa tiedon ja ohjaa sen kontrollerille. Kontrolleri määrittelee saatujen tietojen pohjalta vuon, jonka se jakaa verkkoon niille kytkimille, joiden läpi liikenteen tulisi kulkea. Kont-

rollerin luoma vuo sisältää tiedon lähde- ja kohdeosoitteista sekä sisään- ja ulostuloportit (kuva 12 ja 13).

Table ID	Priority	Packets	Bytes	Matches	Actions/Instructions
▸ 0	28888	2569	n/a	in_port: 1 eth_dst: 00:50:56:99:3f:02 eth_src: 00:50:56:99:30:7c eth_type: ipv4 ipv4_src: 10.206.148.10 ipv4_dst: 10.206.148.20	apply_actions: output: 3
▸ 0	28888	2662	n/a	in_port: 3 eth_dst: 00:50:56:99:30:7c eth_src: 00:50:56:99:3f:02 eth_type: ipv4 ipv4_src: 10.206.148.20 ipv4_dst: 10.206.148.10	apply_actions: output: 1
▸ 0	0	495886	n/a		apply_actions: output: 4294967293

Kuva 12: HP VAN vuomerkintä (eng. flow entry)

```
[HP3]dis openflow instance 10 flow
Instance 10 flow table information:

Table 0 information:
  Table type: Extensibility, flow entry count: 3, total flow entry count: 3

MissRule flow entry information:
  cookie: 0xffff000000000000, priority: 0, hard time: 0, idle time: 0, flags:
  flow_send_rem, byte count: --, packet count: 495592
Match information: any
Instruction information:
  Write actions:
    Output interface: Controller, send length: 65509 bytes

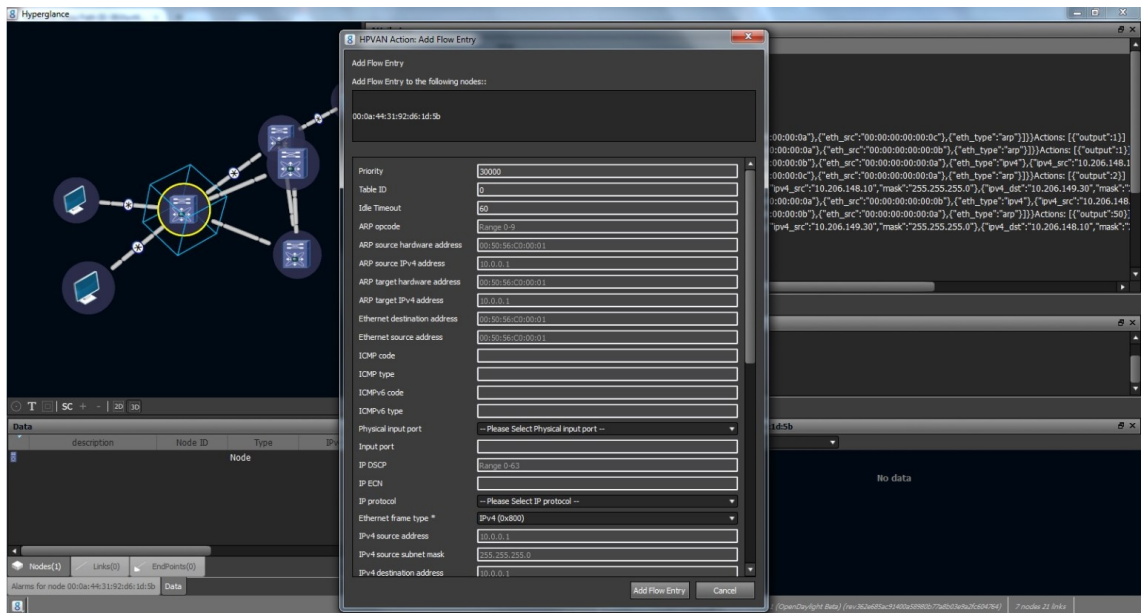
Flow entry 6 information:
  cookie: 0xffff000000002328, priority: 28888, hard time: 0, idle time: 60,
  flags: flow_send_rem, byte count: --, packet count: 2389
Match information:
  Input interface: GE1/0/1
  Ethernet destination MAC address: 0050-5699-3f02
  Ethernet destination MAC address mask: ffff-ffff-ffff
  Ethernet source MAC address: 0050-5699-307c
  Ethernet source MAC address mask: ffff-ffff-ffff
  Ethernet type: 0x0800
  IPv4 source address: 10.206.148.10, mask: 255.255.255.255
  IPv4 destination address: 10.206.148.20, mask: 255.255.255.255
Instruction information:
  Write actions:
    Output interface: GE1/0/3

Flow entry 7 information:
  cookie: 0xffff000000002328, priority: 28888, hard time: 0, idle time: 60,
  flags: flow_send_rem, byte count: --, packet count: 2482
Match information:
  Input interface: GE1/0/3
  Ethernet destination MAC address: 0050-5699-307c
  Ethernet destination MAC address mask: ffff-ffff-ffff
  Ethernet source MAC address: 0050-5699-3f02
  Ethernet source MAC address mask: ffff-ffff-ffff
  Ethernet type: 0x0800
  IPv4 source address: 10.206.148.20, mask: 255.255.255.255
  IPv4 destination address: 10.206.148.10, mask: 255.255.255.255
Instruction information:
  Write actions:
    Output interface: GE1/0/1

[HP3]
```

Kuva 13: HP VAN vuomerkintä kytkimellä

HP VAN -kontrolleri osaa siis automaattisesti luoda vuot ja jakaa ne kytkimille, kun se saa liikenteen aloituksessa lähde- ja kohdeosoitteet. HP VAN:n käyttöliittymä ei tosin sisällä ominaisuutta luoda manuaalisesti tai muokata olemassa olevia vuomerkentöjä. Kyseisiin ominaisuuksiin käytimme testeissä erillistä sovellusta, Hyperglancea. Hyperglancen monipuoliset hallintatyökalut antavat täyden muokkausmahdollisuuden Open-Flow'n kenttiin, joten monimutkaisempien, kuten vlan tagging, vuomerkentöjen luominen on mahdollista. Hyperglancella luotiin samanlaiset vuot, kuin kontrolleri loi automaattisesti, mutta Hyperglancen vuohon ei asetettu aikakatkaisua, joten vuomerkentä ei poistu kytkimeltä ilman erillistä poistoa (kuva 14). Kyseistä toiminnallisuutta testattiin tarkemmin vikatilannetesteissä.



Kuva 14: Hyperglance-sovelluksen graafinen käyttöliittymä

SDN-verkon reititys

HP VAN -kontrollerilla ei voitu testata reititystä, sillä kontrollerin vakio-ominaisuuksiin ei kuulu layer 3 -reititys eikä laboratoriotestien suorittamisen aikaan ollut yleisesti saatavilla erillistä sovellusta HP VAN:lle, joka osaisi reitittää eri aliverkkojen välillä.

Vikatilannetestit

Vikatilannetesteihin valittiin kaksi todennäköisyydeltään yleisintä vikatilannetta, joita SDN-verkossa voidaan kohdata: sisäisen linkin katkeaminen ja kontrollerin yhteyden katkeaminen. Testit suoritettiin fyysisesti katkaisemalla testiverkon linkkejä ja kontrollerilta seurattiin verkon vikasetoisuutta sekä sopeutumista muuttuneeseen topologiaan.

SDN-verkon vikasetoisuutta testattiin ensimmäiseksi ottamalla irti yksi 10 gigabitin sisäinen linkki, jonka seurauksena liikenteen pitäisi siirtyä käyttämään toista linkkiä. Kontrolleri havaitsi muutoksen heti ja automaattisesti laski muuttuneelle topologialle uuden reitin työasemien välille. Muutoksen aikana yksi ping-paketti putosi, joten aikaa muutokselle kului alle 1 sekunti.

Paljon vakavampi SDN-verkkoon kohdistuva vikatilanne, kontrollerin yhteyden menettäminen, oli seuraava vikatilannetesti. Yhteyden menetystä simuloitiin katkaisemalla hallintaverkon linkki, jota kautta kytkimet ottivat yhteyttä kontrolleriin. Kytkimet asettavat itsensä fail secure -tilaan, joka tarkoittaa käytännössä sitä, että kaikki kontrollerille tarkoitettu liikenne tiputetaan ja kytkimet siirtävät liikennettä niiden vuomerkintöjen mukaan, joita kytkimen vuotaulussa on aktiivisena (kuva 15). SDN-verkon liikenne toimii niin kauan, kunhan vuomerkinnät pysyvät kytkinten vuotaulussa. Vuomerkinnöillä on kaksi aikakatkaisuarvoa, jotka asetetaan sitä luotaessa; "idle timeout" ja "hard timeout". Jos idle timeoutin aikana vuota ei käytetä, niin se poistuu kytkimen vuotaulusta. Hard timeout on vuomerkinnän kokonaiselinaika. Vakiona HP VAN luo 300 sekunnin idle timeout -arvot vuomerkintöihin, joten jos kytkin menettää yhteyden kontrolleriin, niin se jatkaa liikenteen välittämistä korkeintaan 5 minuuttia. Vikatilannetestissä Hyperglan-cellä luotiin manuaalisesti samanlainen vuomerkintä, jonka kontrolleri oli luonut, mutta

ilman aikakatkaisuarvoa. Todettiin, että SDN-verkon liikenteenvälitys ei keskeytynyt missään vaiheessa, vaikka yhteys kontrolleriin oli menetetty.

```

ge agent new neighbor created on Port GigabitEthernet1/0/1 (IfIndex 1), Chassis
ID is 0023-9c00-9a80, Port ID is 557.

dis openflow sum
Fail-open mode: Se - Secure mode, Sa - Standalone mode

ID      Status   Datapath-ID      Channel    Table-num  Port-num  Reactivate
10      Active   0x000a443192d31393  Failed(Se)  1          3         N

[HP1-GigabitEthernet1/0/1]%Mar 18 06:01:41:436 2011 HP1 OFP/5/OFP_CONNECT: Openf
low instance 10, controller 1 is connected.
%Mar 18 06:01:41:446 2011 HP1 OFP/5/OFP_FLOW_ADD_TABLE_MISS: Openflow instance 1
0 controller 1: add table miss flow entry, xid 0x25ee13, cookie 0xffff0000000000
00, table id 0.
shutdown
[HP1-GigabitEthernet1/0/1]%Mar 18 06:01:50:360 2011 HP1 IFNET/3/PHY_UPDOWN: Giga
bitEthernet1/0/1 link status is down.
%Mar 18 06:01:50:361 2011 HP1 IFNET/5/LINK_UPDOWN: Line protocol on the interfac
e GigabitEthernet1/0/1 is down.
%Mar 18 06:02:03:428 2011 HP1 OFP/5/OFP_CONNECT: Openflow instance 10, controlle
r 1 is disconnected.
%Mar 18 06:02:03:430 2011 HP1 OFP/5/OFP_FAIL_OPEN: Openflow instance 10 is in fa
il secure mode.

```

Kuva 15: Kytkin fail secure -tilassa, kun yhteys kontrolleriin on katkennut

4.4 OpenDaylight-kontrollerin testit

OpenDaylight Hydrogen -kontrollerin testiympäristö pysyi kontrollerin muutosta lukuun ottamatta täysin identtisenä. Kytkimille määritettiin ODL-kontrollerin IP-osoite ja Open-Flow-tila käynnistettiin uudestaan, jotta HP VAN -kontrollerin vuomerkinnot poistuvat kytkinten vuotauluista.

```

openflow instance 10
  classification vlan 10
  controller 1 address ip 10.206.142.16
  active instance

```

Kytöinten ja kontrollerin välille muodostui onnistunut yhteys ja protokollaksi keskusteltiin OpenFlow 1.3. Kontrolleri muodosti graafiseen käyttöliittymään verkkotopologian, jossa näkyi kytkimet ja niiden väliset linkit, mutta työasemat eivät ilmestyneet kontrollerille eikä liikenne toiminut. Hydrogen ei siis osannut automaattisesti luoda vuomerkintöjä. Kontrollerin käyttöliittymässä on työkalu, jonka avulla voidaan luoda manuaalisesti vuomerkintöjä, kuten Hyperglancella. Kontrollerilta luotiin vuot, jotka sallisivat liikenteen kahden työaseman välillä. Kontrolleri asensi vuot kytkimille, mutta liikenne ei edelleenkään toiminut. Tarkastellessa kytkimen vuotaulua CLI:n kautta huomattiin, että vuon toiminto oli asetettu tiputtamaan liikenne (eng. Action = drop). Kontrollerin käyttöliittymästä tarkastettiin, että vuon toiminto oli asetettu välittämään liikenne ulos fyysisestä portista. Epäiltiin, että kontrolleri ei tue kytkimiä tai OpenFlow versiota 1.3, vaikka se on ilmoitettu tuetuksi. Vianselvitykseen kulutettiin liian paljon aikaa, ja valitettavasti aika loppui laboratoriotesteille. Myöhemmin selvisi, että Hydrogen-kontrollerin graafinen käyttöliittymä tukee ainoastaan OpenFlow-versiota 1.0, jonka toimintokentät poikkeavat version 1.3 kentistä. Ainoa tapa luoda Hydrogenilla OpenFlow 1.3 -pohjainen vuo, on tehdä se suoraan Northbound-ohjelmointirajapinnan kautta. Päädyttiin ratkaisuun testata Hydrogen-kontrolleria virtuaaliympäristössä. [16.]

Mininet-verkkosimulaattori

OpenDaylight-testit suoritettiin Mininet-emulaattorilla, joka luo realistisen virtuaaliverkon ja soveltuu erityisesti kehittäjien ja SDN-testaajien käyttöön, sillä sen avulla voidaan ajaa muuntelematonta ohjelmistokoodia. OpenDaylight-projektin kautta on suoraan ladattavissa virtuaalikone, johon on asennettu Hydrogen-kontrolleri sekä Mininet-sovellus.

Topologian luonti Mininettiin

Mininettiin on mahdollista luoda omia verkkotopologioita käyttäen Python-skriptejä. Testejä varten luotiin samanlainen verkkotopologia, joka oli jo aiemmin rakennettu testilaboratorioon, eli neljä toisiinsa kytkettyä kytkintä ja kaksi työasemaa. Mininettiin luotiin virtuaaliverkko käyttäen Python-skriptiä (liite 5) sekä Open Virtual Switch (OVSK) -virtuaalikytkimiä. Kytkimet määriteltiin käyttämään OpenFlow-versiota 1.3 (kuva 16).

```
sudo mn --custom test-topo.py --topo mytopo --controller 'remote' --switch ovsk,protocols=OpenFlow13
```

```
mininet@mininet-vm:~/mininet/custom> sudo mn --custom test-topo.py --topo mytopo --controller 'remote' --switch ovsk,protocols=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
have protocol params!
s1 have protocol params!
s2 have protocol params!
s3 have protocol params!
s4
*** Adding links:
(h1, s3) (h2, s4) (s1, s2) (s1, s3) (s1, s4) (s2, s3) (s2, s4) (s3, s4)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 4 switches
s1 OVSSwitch opts: protocols=OpenFlow13
s2 OVSSwitch opts: protocols=OpenFlow13
s3 OVSSwitch opts: protocols=OpenFlow13
s4 OVSSwitch opts: protocols=OpenFlow13

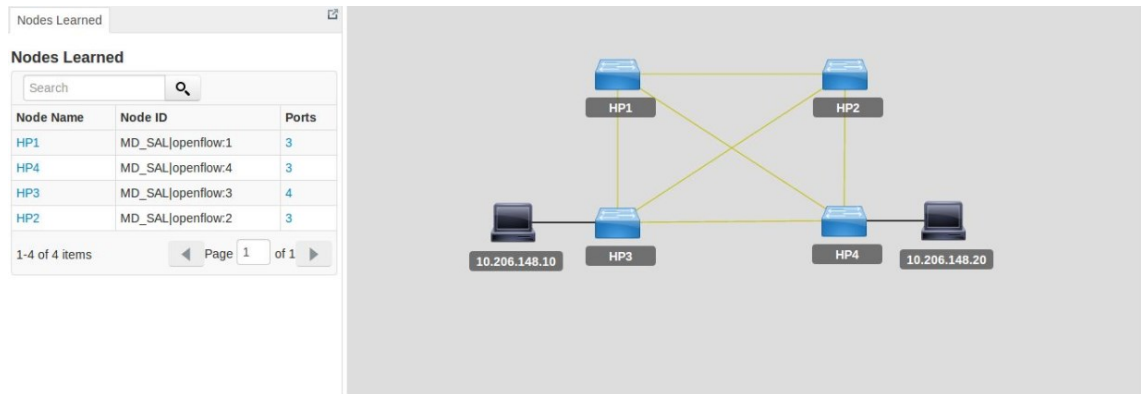
*** Starting CLI:
mininet> net
h1 h1-eth0:s3-eth4
h2 h2-eth0:s4-eth4
s1 lo: s1-eth1:s2-eth1 s1-eth2:s3-eth1 s1-eth3:s4-eth1
s2 lo: s2-eth1:s1-eth1 s2-eth2:s3-eth2 s2-eth3:s4-eth2
s3 lo: s3-eth1:s1-eth2 s3-eth2:s2-eth2 s3-eth3:s4-eth3 s3-eth4:h1-eth0
s4 lo: s4-eth1:s1-eth3 s4-eth2:s2-eth3 s4-eth3:s3-eth3 s4-eth4:h2-eth0
c0
mininet> pingall
*** Ping: testing ping reachability
h1 -> X
h2 -> h1
*** Results: 50% dropped (1/2 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```

Kuva 16: Mininet-virtuaaliverkon luonti

OpenDaylight-kontrollerin näkymä ja vuon määrittäminen

Kontrolleri näkee Mininetillä luodun verkon, aivan kuten fyysisenkin verkon, ja se tukee virtuaalisena myös samoja toimintoja (kuva 17). Koska tiedossa oli, ettei kontrollerin työkaluilla voinut luoda OpenFlow 1.3 -pohjaisia vuomerkeitä, tarvitsi vuo asettaa kontrollerille suoraan Northbound-ohjelmointirajapinnan kautta. Hydrogen käyttää REST-ohjelmointirajapintaa kommunikointiin sovellusten kanssa. Kontrollerille lähetettiin REST-kutsu restconf-protokollan kautta, johon määriteltiin OpenFlow 1.3:n mukaiset tyyppikoodit, säännöt ja toiminnot (kuva 18). OpenFlow-protokollan jokaisesta versiosta on saatavilla versiokohtaiset määrittäykset. Kontrolleri vastaanotti vuon ja lähetti

sen edelleen vuossa määritellylle kytkimelle, joka alkoi välittää liikennettä (kuva 19).
[17.]



Kuva 17: OpenDaylight Hydrogen -kontrollerin verkkonäkymä

Add Flow

http://127.0.0.1:8080/restconf/config/opendaylight-inventory:nodes/node/openflow:3/table/0/flow/1 PUT URL params Headers (3)

form-data x-www-form-urlencoded raw XML

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <flow xmlns="urn:opendaylight:flow:inventory">
3   <priority>2</priority>
4   <flow-name>test</flow-name>
5   <match>
6     <ethernet-match>
7       <ethernet-type>
8         <type>2048</type>
9       </ethernet-type>
10      </ethernet-match>
11      <ipv4-destination>10.206.148.20/32</ipv4-destination>
12    </match>
13    <id>1</id>
14    <table-id>0</table-id>
15    <instructions>
16      <instruction>
17        <order>0</order>
18        <apply-actions>
19          <action>
20            <order>0</order>
21            <output-action>
22              <output-node-connector>3</output-node-connector>
23              <max-length>60</max-length>
24            </output-action>
25          </action>
26        </apply-actions>
27      </instruction>
28    </instructions>
29  </flow>
30

```

Kuva 18: REST-kutsu, jolla määriteltiin vuomerkinä kontrollerille

Node	In Port	DL Src	DL Dst	DL Type	NW Src	NW Dst	NW Proto	TP Src	TP Dst	Actions	Byte Count	Packet Count	Duration Seconds	Idle Timeout	Priority
HP3	*	*	*	IPv4	*	10.206.148.20	*	*	*	OUTPUT = MD_SAL openflow:3:3	0	0	0	300	2

Kuva 19: REST-kutsulla luotu vuo, OpenDaylight-kontrollerin vuonäkymässä

5 Tulevaisuus ja jatkokehitys

On nähtävissä, että kiinnostus SDN:ää kohtaan tulee lisääntymään lähitulevaisuudessa. Sitä mukaa kun sovellukset ja niiden ominaisuudet kehittyvät sille tasolle, että niiden käyttö tuotantoympäristössä on järkevää, monet yritykset alkavat varmasti suunnittelemaan SDN:n käyttöönottoa. Koko olemassa olevien verkkojen päivittämistä SDN-verkoiksi tuskin tulee tapahtumaan, sillä yhteensopivuuden ja toiminnallisuuden kanssa tulee varmasti ongelmia. Testausmielessä joitain tuotantoverkon segmenttejä voidaan toteuttaa SDN-ratkaisuilla, mutta alusta asti SDN-verkoksi suunniteltu kokonaisuus on järkevin tapa. Jo nyt on nähtävillä, että SDN:llä voidaan toteuttaa suuriakin verkkoja, esimerkiksi Googlen palvelinkeskusten välinen WAN-verkko, josta tosin Google ei jaa tarkempaa tietoa. [2.]

Tässä insinööriyössä käytetyiden kontrollereiden kehitys jatkuu myös. OpenDaylight-projekti on kehittänyt Hydrogen-kontrolleria ja julkaissut uuden kontrollerin, Heliumin. Uudessa kontrollerissa on paranneltu vanhoja ominaisuuksia ja myös tuotu paljon uusia. HP jatkaa myös oman SDN-ekosysteemin kehitystä ja on julkaissut SDN-sovelluskaupan, josta yhteensopivien sovellusten hankkiminen on tehty helpoksi. Eriytyisen mielenkiintoinen tuleva asia SDN-maailmassa on, Brocaden OpenDaylight-pohjainen Vyatta-kontrolleri, joka mahdollistaa 3-tason reitityksen SDN-verkoissa. [18.]

Open Networking Foundation on myös julkistanut SDN-sertifikaatit, jotka takaisivat standardoidut pätevyysvaatimukset SDN-asiantuntijoille. Alustavasti on suunniteltu kaksi erilaista sertifikaattia, joista toinen on suunnattu verkkoasiantuntijoille, ja toinen myynnin ja markkinoinnin ammattilaisille. Sertifiointikokeet on määrä aloittaa vuoden 2015 alkupuolella. [19.]

6 Yhteenveto

Työn tavoitteena oli selvittää, miten SDN-ratkaisut toimivat sekä tarjota yleiskuva SDN:stä, jotta Cygate Oy voi paremmin konsultoida asiakkaitaan. Työssä päästiin sen asettamiin tavoitteisiin sekä saatiin Cygaten kannalta tärkeää tietoa.

Työ eteni tavoitteiden asettamisen jälkeen lähes muuttumattomana. Työn alkuvaiheessa aikaa kului paljon tiedon analysointiin, sillä SDN on viime vuosien aikana kehittynyt merkittävästi ja on edelleen kehitysvaiheessa, joten muutoksia tapahtuu paljon. Saatavilla olevan tiedon määrä on myös valtava, sillä useilla eri laite- ja sovelluskehittäjillä on omia näkemyksiä SDN:stä ja esittävät asiat omalla tavallaan. Tärkeimpien asioiden selvittäminen ja kerääminen yhteen useista eri lähteistä vei runsaasti aikaa.

Laboratorioverkon toteutukseen kului lähes suurin osa työhön kulutetusta ajasta. Laitehankintojen tekemiseen ja toimitukseen tuli viivytyksiä työn aikana, joten työ oli lähes pysähdyksissä jonkin aikaa. Myös laboratoriotesteihin kulutettiin paljon aikaa, sillä SDN oli itselleni sekä työnohjaajalle, Jaakko Rautaselle, täysin uutta. Pienten vastoinkäymisten ja testaussuunnitelmien muutosten jälkeen saatiin onnistuneesti testattua SDN-ratkaisuja käytännössä.

Cygate ja työnohjaaja, tietoliikennekonsultti Jaakko Rautanen, olivat tyytyväisiä työssä saavutettuihin tuloksiin. Tekniikkana SDN tarjoaa paljon uusia mahdollisuuksia tietoverkkoihin ja on tällä hetkellä selkeästi siinä tilassa, että sen jatkokehitys ja käyttöönotto lisääntyvät merkittävästi lähitulevaisuudessa. Työn aikana todettiin, että SDN on täysin riippuvainen jatkuvasta sovelluskehitystä, joten sen käyttöönotto tuotantoympäristöön vaatisi verkkoasiantuntijoiden lisäksi myös sovelluskehittäjiä. Tällä hetkellä yleisesti saatavilla olevat SDN-sovellukset ovat vielä siinä vaiheessa, että niiden ominaisuudet ovat puutteellisia, mutta kehittyvät jatkuvasti. SDN:n käyttö on mahdollista tarkasti harkituissa ja rajatuissa ympäristöissä, mutta laajamittainen tuotantokäyttö odottaa vielä tekniikan kypsymistä.

Lähteet

- 1 The Road to SDN: An Intellectual History of Programmable Networks. Tutkielma.
<https://www.cs.princeton.edu/courses/archive/fall13/cos597E/papers/sdnhistory.pdf>. Luettu 8.9.2014.
- 2 Google's SDN WAN backbone. Verkkootikkeli.
<http://www.networkworld.com/article/2189197/lan-wanoogle-s-software-defined-openflow-backbone-/lan-wan/google-s-software-defined-openflow-backbone-drives-wan-links-to-100--utilization.html>. Luettu 8.9.2014.
- 3 The History of SDN. Verkkootikkeli.
<http://www.computerweekly.com/feature/The-history-of-OpenFlow>. Luettu 8.9.2014.
- 4 Introduction to SDN. luento.
https://www.youtube.com/watch?feature=player_detailpage&v=eXsCQdshMr4&t=168. Katsottu 8.9.2014.
- 5 IANA-rekisteri. Verkkodokumentti. <http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?search=6653>. Luettu 1.7.2014.
- 6 SDN-kontrollerit. Verkkodokumentti.
<http://searchsdn.techtarget.com/definition/SDN-controller-software-defined-networking-controller>. Luettu 4.9.2014.
- 7 Juniper Contrailin määrittely. Verkkodokumentti.
<http://www.juniper.net/us/en/local/pdf/whitepapers/2000535-en.pdf>. Luettu 20.9.2014.
- 8 Northbound API. Verkkootikkeli. <http://networkstatic.net/the-northbound-api-2/>. Luettu 9.9.2014.
- 9 Northbound API. Verkkodokumentti. <http://searchsdn.techtarget.com/feature/A-primer-on-northbound-APIs-Their-role-in-a-software-defined-network>. Luettu 9.9.2014.
- 10 ONF-työryhmien esittely. Verkkodokumentti.
<https://www.opennetworking.org/working-groups/working-groups-overview->. Luettu 9.9.2014.

- 11 HP VAN SDN Controller and Applications Support Matrix. Verkkodokumentti. Luettu 25.8.2014.
- 12 OpenDaylight-projektin esittely. Verkkosivusto. <http://www.opendaylight.org/project>. Luettu 13.9.2014.
- 13 OpenDaylight-kontrollerin julkaisupaketit. Verkkosivusto. <http://www.opendaylight.org/software/editions>. Luettu 13.9.2014.
- 14 Rautanen, Jaakko. SDN-palaveri. Cygate Oy, Helsinki. 2.6.2014.
- 15 HP VAN SDN Controller Administrator Guide. Verkkodokumentti. Luettu 14.9.2014.
- 16 Hands-on demo OpenDaylight Summit 2014. <https://www.youtube.com/watch?v=J-7lmVPrxb0#t=3335>. Katsottu 5.11.2014.
- 17 OpenFlow-version 1.3.0 määrittely. Verkkodokumentti. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf>. Luettu 1.3.2014.
- 18 Vyatta-kontrollerin esittely. Verkkosivu. <http://www.brocade.com/products/all/software-defined-networking/brocade-vyatta-controller/index.page>. Luettu 9.11.2014.
- 19 ONF-sertifikaattien esittely. Verkkosivu. <https://www.opennetworking.org/news-and-events/press-releases/1780-open-networking-foundation-introduces-new-sdn-skills-certification-program>. Luettu 9.11.2014.
- 20 Software Defined Networking (SDN) Architectures and Implications (2013 Melbourne) - Stefano Previdi - Distinguished Engineer, CISCO SYSTEMS, INC. Seminaariesitys. https://www.ciscolive.com/online/connect/sessionDetail.wv?SESSION_ID=6669&backBtn=true. Luettu / katsottu 28.1.2014.
- 21 OpenFlow and SDN: Today and Tomorrow (2013 Melbourne) - Ken Hook - Sr. Product Manager, Cisco. Seminaariesitys. https://www.ciscolive.com/online/connect/sessionDetail.wv?SESSION_ID=6712&backBtn=true. Luettu / katsottu 28.1.2014.
- 22 Software Defined Networks and OpenFlow (2013 Orlando) - Frank Brockners - Distinguished Engineer, Cisco. Seminaariesitys.

- https://www.ciscolive.com/online/connect/sessionDetail.wv?SESSION_ID=8231&backBtn=true. Luettu / katsottu 28.1.2014.
- 23 The Hitchhiker's Guide to onePK (2014 Milan) - Shelly Cadora - Technical Marketing Engineer, Cisco. Seminaariesitys.
https://www.ciscolive.com/online/connect/sessionDetail.wv?SESSION_ID=76582&backBtn=true. Luettu / katsottu 28.1.2014.
- 24 OpenDaylight-projektin tekninenmäärittely. Verkkodokumentti.
<http://www.opendaylight.org/project/technical-overview>. Luettu 4.2.2014.
- 25 OpenFlow-version 1.1.0 määrittely. Verkkodokumentti.
<http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf>. Luettu 1.3.2014.
- 26 Rautanen, Jaakko; Toppila, Teemu (HP). SDN-palaveri. Cygate Oy, Helsinki. 13.6.2014.
- 27 Northbound API guide: The new network application. Verkkodokumentti.
<http://searchsdn.techtarget.com/guides/Northbound-API-guide-The-rise-of-the-network-applications>. Luettu 4.9.2014.
- 28 SDN-kontrollerit. Verkkoartikkeli. <https://www.sdncentral.com/sdn-controllers/>. Luettu 4.9.2014.
- 29 Martin Casadon haastattelu ja artikkeli.
<http://www.enterprisenetworkingplanet.com/netsp/openflow-inventor-martin-casado-sdn-vmware-software-defined-networking-video.html>. Luettu 8.9.2014.
- 30 Nadeau, Thomas D.; Gray, Ken. 2013. SDN: Software Defined Networks. O'Reilly Media.
- 31 OpenDaylight-kontrollerin demo. <https://www.youtube.com/watch?v=J-7ImVPPrxb0>. Katsottu 11.9.2014.

Kytkimen HP3 esimerkkikonfiguraatio

```
[HP3]display current
#
version 7.1.045, Release 2307
#
sysname HP3
#
irf mac-address persistent timer
irf auto-update enable
undo irf link-delay
irf member 1 priority 1
#
lldp global enable
#
system-working-mode standard
fan prefer-direction slot 1 port-to-power
password-recovery enable
#
vlan 1
#
vlan 10
name sdn
#
vlan 131
name Jaakko3
#
openflow instance 10
classification vlan 10
controller 1 address ip 10.206.142.15
#
undo stp vlan 10 enable
stp mode pvst
stp global enable
```

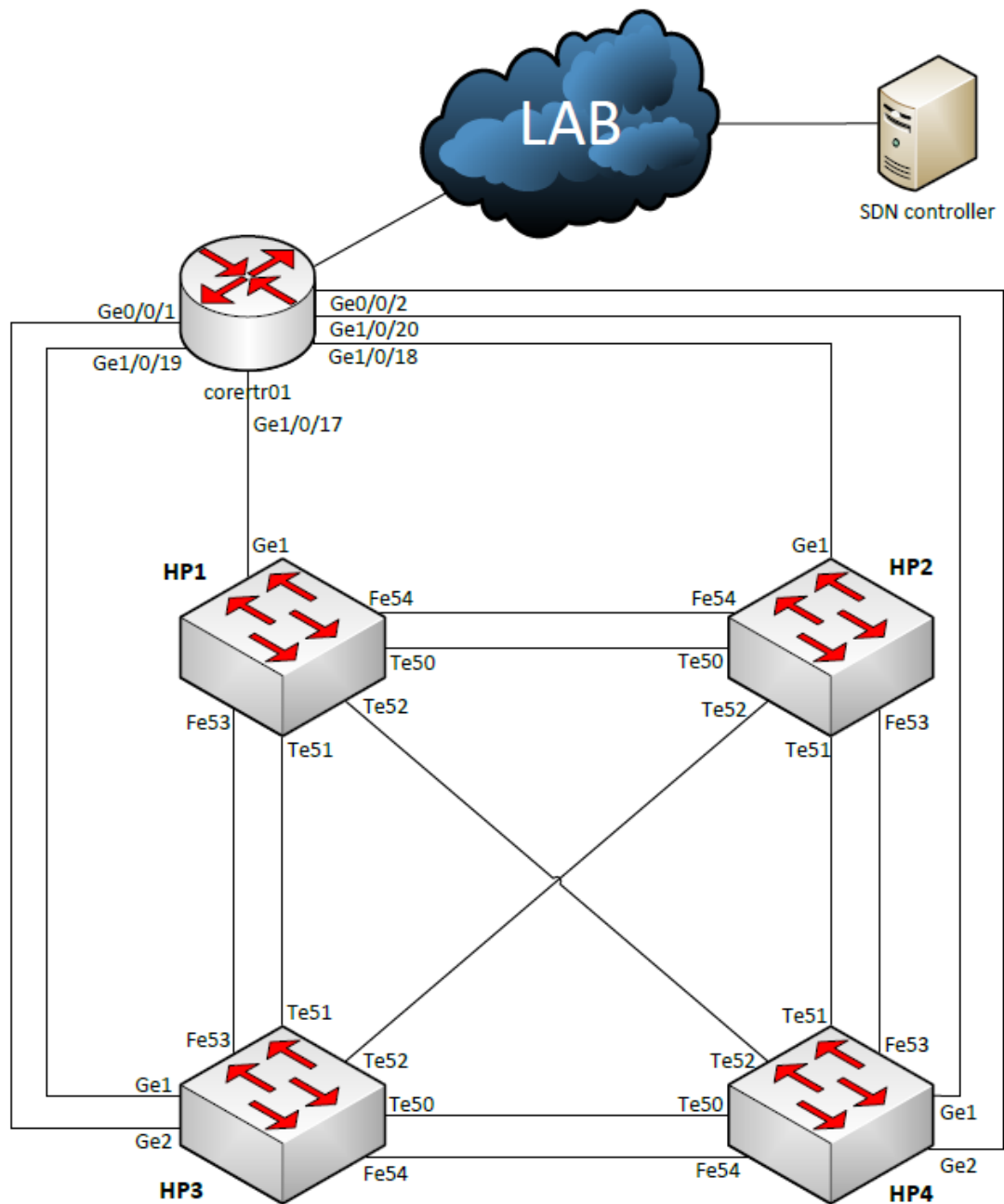
```
#
interface NULL0
#
interface Vlan-interface10
 ip address 10.10.10.3 255.255.255.0
#
interface Vlan-interface131
 ip address 10.206.146.13 255.255.255.0
#
interface FortyGigE1/0/53
 port link-mode bridge
 description SW1-HP5900_port_1/0/53
 port access vlan 131
#
interface FortyGigE1/0/54
 port link-mode bridge
 port access vlan 131
#
interface GigabitEthernet1/0/1
 port link-mode bridge
 description "SDN host 1, coretr01"
 port access vlan 10
#
interface GigabitEthernet1/0/2
 port link-mode bridge
 description "SDN host 3, coretr01"
 port access vlan 10
#
interface GigabitEthernet1/0/3
 port link-mode bridge
 port access vlan 10
 undo stp enable
#
interface GigabitEthernet1/0/4
```

```
port link-mode bridge
#
interface GigabitEthernet1/0/5
port link-mode bridge
#
interface Ten-GigabitEthernet1/0/49
port link-mode bridge
#
interface Ten-GigabitEthernet1/0/50
port link-mode bridge
port link-type trunk
port trunk permit vlan 1 10
port trunk pvid vlan 10
#
interface Ten-GigabitEthernet1/0/51
port link-mode bridge
port link-type trunk
port trunk permit vlan 1 10
port trunk pvid vlan 10
#
interface Ten-GigabitEthernet1/0/52
port link-mode bridge
port link-type trunk
port trunk permit vlan 1 10
port trunk pvid vlan 10
#
scheduler logfile size 16
#
line class aux
user-role network-admin
#
line class vty
user-role network-operator
#
```

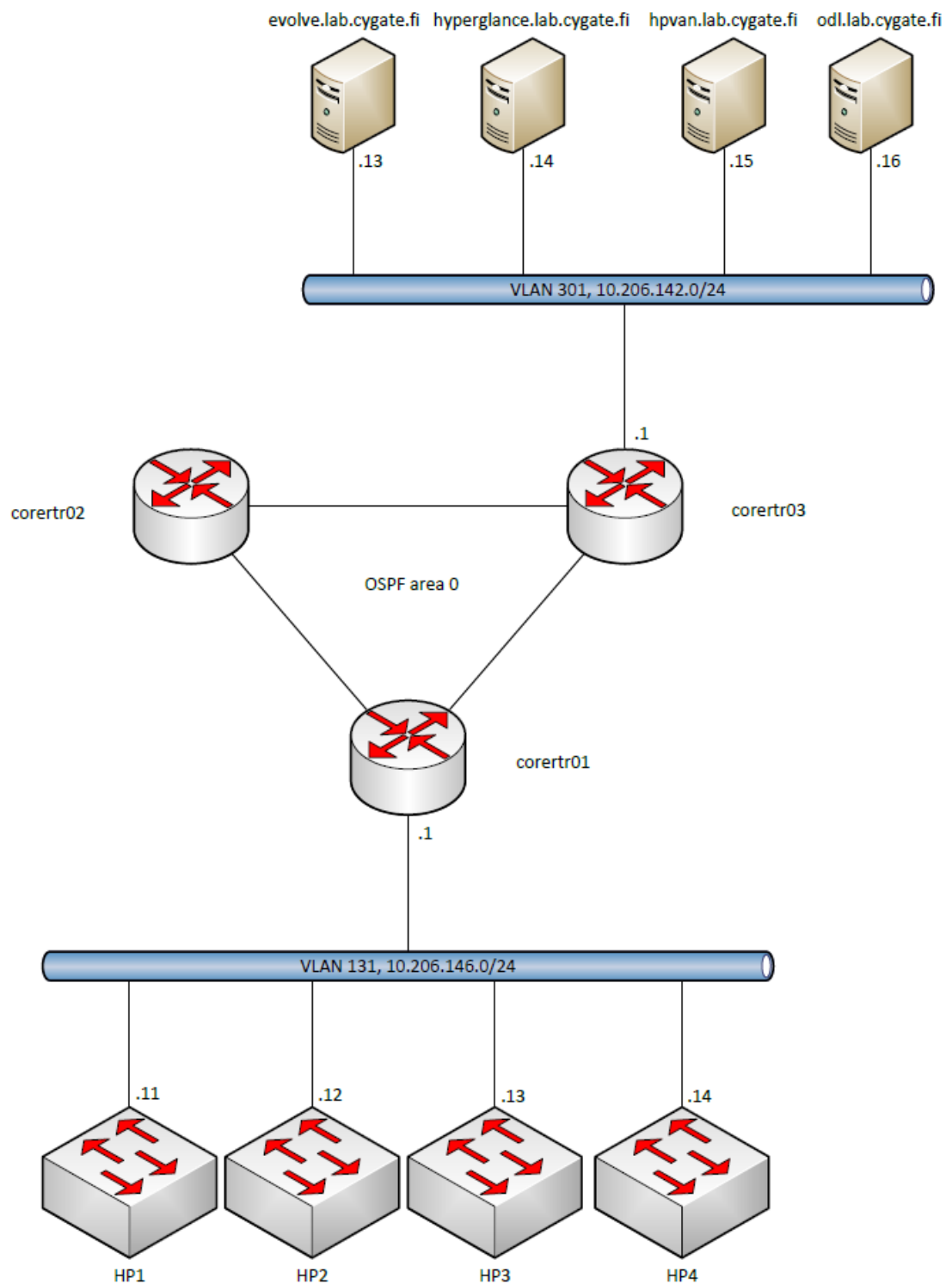
```
line aux 0
  user-role network-admin
#
line vty 0 15
  authentication-mode scheme
  user-role network-operator
  protocol inbound ssh
#
line vty 16 63
  user-role network-operator
#
ip route-static 0.0.0.0 0 10.206.146.1
#
snmp-agent
snmp-agent local-engineid 800063A280443192D3139700000001
snmp-agent community read sdnlabra
snmp-agent sys-info version v3
snmp-agent target-host trap address udp-domain 10.206.168.100 params securi-
tyname sdnlabra v3
snmp-agent trap enable arp
snmp-agent trap enable radius
#
ssh server enable
#
radius scheme system
  user-name-format without-domain
#
domain system
#
domain default enable system
#
role name level-0
  description Predefined level-0 role
#
```

```
role name level-1
  description Predefined level-1 role
#
role name level-2
  description Predefined level-2 role
#
user-group system
#
local-user sdn class manage
  password hash
  $h$6$zAGgcrGmavwlkKxU$HujNP2BVA4hgOjIUD1th6W3orjemp6gNhP+WJJUzoKgm
  38d6n4aGat7SDDtY0miegnyu9jetOPWmoRstWb/61w==
  service-type ssh
  authorization-attribute user-role network-admin
  authorization-attribute user-role network-operator
#
local-user test class manage
  password hash
  $h$6$R1zk/B/tAFT4uYlo$0zP4YAIXSJiYYXCtuZfnMryqhmTcGcLgZWInu9NYwAfaMp
  hFePe2fngPIDdQm7MC3S+CHI9XAU8vikY7xs/vNA==
  service-type ftp
  authorization-attribute work-directory flash:/
  authorization-attribute user-role network-admin
  authorization-attribute user-role network-operator
#
  ftp server enable
#
return
[HP3]
```

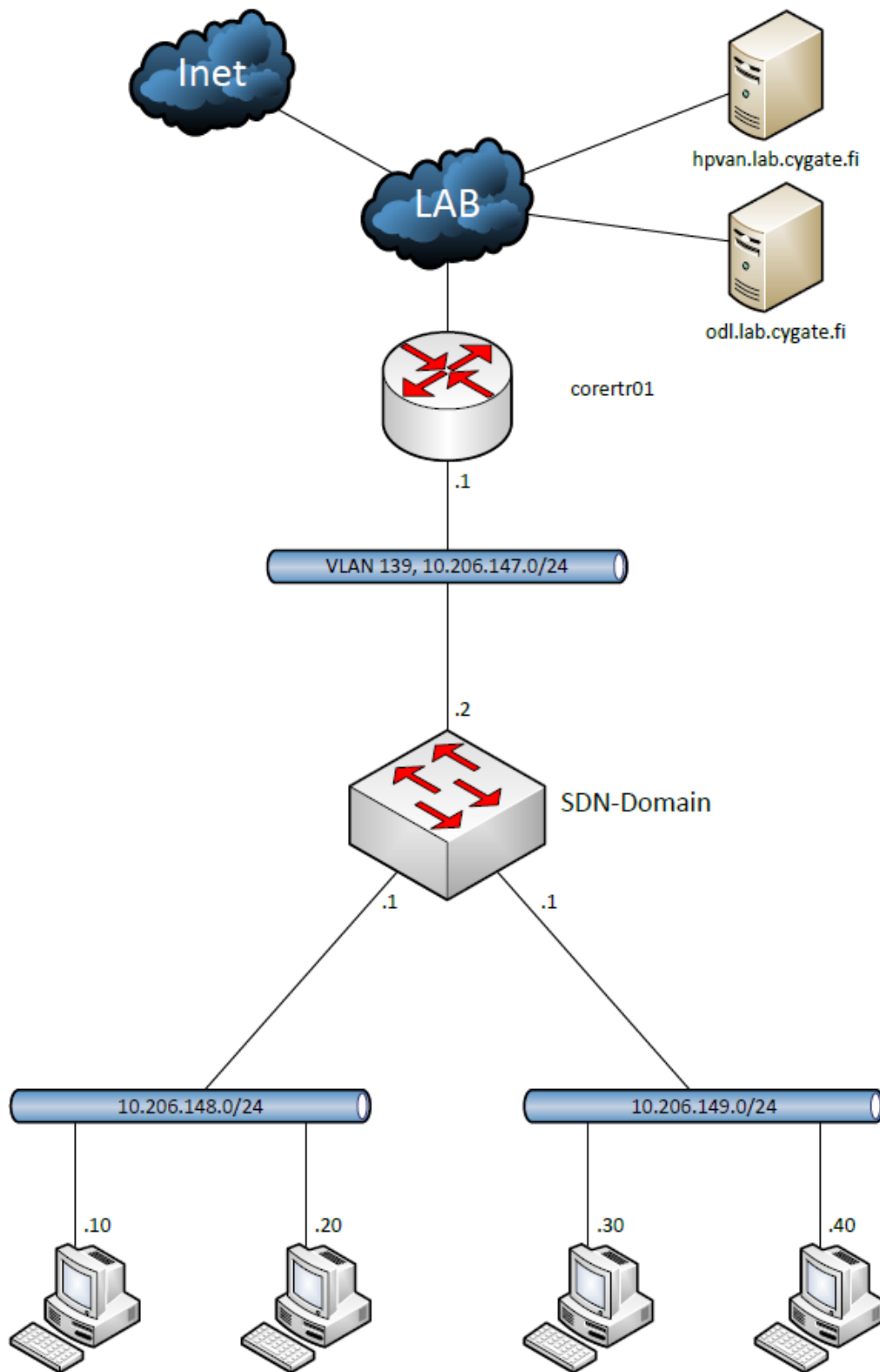

Fyysinen verkkotopologia



Looginen verkkotopologia



SDN-verkon looginen verkkotopologia



Mininet verkkotopologian Python-skripti

```
"""Custom topology

Four interconnected switches plus a host for switches 3 and 4:

    switch --- switch
      |   \ /   |
      |   X   |
      |   / \   |
    host --- switch --- switch --- host

"""

from mininet.topo import Topo
class MyTopo( Topo ):
    "Simple topology example."
    def __init__( self ):
        "Create custom topo."

        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches
        S1 = self.addSwitch( 's1' )
        S2 = self.addSwitch( 's2' )
        S3 = self.addSwitch( 's3' )
        S4 = self.addSwitch( 's4' )
        H1 = self.addHost( 'h1', ip="10.206.148.10" )
        H2 = self.addHost( 'h2', ip="10.206.148.20" )

        # Add links
        self.addLink( S1, S2 )
        self.addLink( S1, S3 )
        self.addLink( S1, S4 )
        self.addLink( S2, S3 )
        self.addLink( S2, S4 )
        self.addLink( S3, S4 )
        self.addLink( H1, S3 )
        self.addLink( H2, S4 )

topos = { 'mytopo': ( lambda: MyTopo() ) }
```