



# OSPF VXLAN käyttöönotto

## Tieto- ja viestintätekniikka

Matti Nordling

Opinnäytetyö, AMK

Joulukuu 2023

Tietojenkäsittely ja tietoliikenne

Insinööri (AMK), tieto- ja viestintätekniikka

**Nordling, Matti**

## **OSPF VXLAN käyttöönotto**

Jyväskylä: Jyväskylän ammattikorkeakoulu. Marraskuu 2023, 53 sivua

Tietojenkäsittely ja tietoliikenne Tieto- ja viestintätekniikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: kyllä

### **Tiivistelmä**

Taustana OSPF VXLAN käyttöönotossa oli palvelinvirtualisoinnin huomattava kasvava määrä konesaliympäristöissä ja VXLAN-tekniikan yleistyminen konesaliympäristöissä. Motivaationa oli kerätä tietoa verkkotekniikan perusteista ja mahdollisista ongelmatilanteista erilaisten verkkotekniikoiden parissa liittyen VXLAN tekniikkaan.

Opinnäytetyön tarkoituksena oli käyttöönottaa VXLAN-tekniikka OSPF-reititysprotokollan avulla ja tarkastella tekniikoiden toiminnallisuutta. Erillistä toimeksiantajaa työllä ei ollut, vaan aihe oli itse keksitty. Tavoitteena oli esittää VXLAN ja OSPF käyttöönotto konfiguroimalla tekniikoita Cisco CSR1000V virtuaalisten reitittimien avulla ja tutkia pakettitasolla, kuinka reitittimet kuljettavat viestejä verkossa toisillensa ja virtuaalisille tietokoneille.

Tavoitteena oli saada luotua toimiva verkkoympäristö ja konfiguraatiot, joiden avulla oli mahdollista tutkia ja todeta tekniikoiden toimivuutta. Toisena tavoitteena oli kerätä opinnäytetyöhön virallisista standardeista dataa, joiden avulla johdatella lukijaa ymmärtämään miksi VXLAN-tekniikka on kehitetty ja sitä on otettu käyttöön.

Toteutus tapahtui käytännössä virtuaalisessa ympäristössä. Isäntä koneena käytettiin omaa tietokonetta ja verkkolaitteita virtualisoitiin VMWARE ympäristössä, josta verkkolaitteiden toiminnallisuus saatiin vietyä GNS3 verkkosimulaattori sovellukseen. Verkkosimulaattorissa virtualisoiduista laitteista saatiin rakennettua oma verkko. Käyttöönotto vaiheessa tarkasteltiin VMWARE ja GNS3 sovelluksien asennus. Sovellusten käyttämistä päästiin tarkastelemaan Cisco CSR1000V-virtuaalireitittimien asennus ja esikonfigurointi vaiheessa.

Reitittimien ja pakettianalysointorin dataa päästiin tarkastelemaan käyttöönotossa. Pakettianalysointorin avulla päästiin tarkastelemaan paketti kerrallaan, kuinka OSPF naapuruussuhde muodostuu reitittimien välille. Todettiin myös kuinka virtuaalinen rajapinta löytää toisen rajapinnan VXLAN verkossa.

Opinnäytetyössä onnistuttiin käyttöönottamaan OSPF- ja VXLAN-tekniikat onnistuneesti omassa virtuaalisessa verkossa ja todistamaan niiden toimivuus. Tuloksista voi olla hyötyä tekniikoista kiinnostuneille henkilöille, koska aineisto VXLAN-tekniikasta on etenkin suomen kielellä hyvin rajallista tällä hetkellä.

### **Avainsanat (asiasanat)**

OSPF, VXLAN, Virtualisointi

### **Muut tiedot (salassa pidettävät liitteet)**

**Nordling, Matti**

**OSPF VXLAN introduction**

Jyväskylä: JAMK University of Applied Sciences, November 2023, 53 Pages

Information and Communications. Degree Programme in Information and Communication Technology. Bachelor's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

**Abstract**

The background for the introduction of OSPF VXLAN was the huge increase in server virtualization in data center environments and the widespread adoption of VXLAN technology in data center settings. The motivation was to gather information about the fundamentals of network technology and potential issues related to various networking technologies focusing on VXLAN technology.

The purpose of the thesis was to implement VXLAN technology using the OSPF routing protocol and to examine the functionality of the technologies. The work did not have a separate client. Instead, the topic was self-conceived. The goal was to present VXLAN and OSPF implementation by configuring these technologies using Cisco CSR1000V virtual routers and to investigate at the packet level how routers transmit messages in the network between each other and to virtual machines.

The aim was to create a functional network environment and configurations that allowed for the investigation and confirmation of the technologies' functionality. Another objective was to gather information from official standards for the thesis to guide the reader in understanding why VXLAN technology has been developed and implemented.

The implementation took place in a virtual environment. The host machine was a personal computer and network devices were virtualized in the VMWARE environment. The functionality of network devices was then imported into the GNS3 network simulator application. The network was constructed from the virtualized devices in the simulator. During the implementation phase, the installation of VMWARE and GNS3 applications was examined. The use of the applications was observed during the installation and pre-configuration phase of Cisco CSR1000V virtual routers.

Router and packet analyzer data were examined during the implementation. The packet analyzer allowed for a detailed examination of how the OSPF neighbor relationship is formed between routers and it was also observed how a virtual interface finds another interface in the VXLAN network.

The thesis successfully implemented OSPF and VXLAN technologies in a virtual network, demonstrating their functionality. The results may be beneficial for individuals interested in these technologies, as information about VXLAN technology in the Finnish language is currently limited.

**Keywords/tags (subjects)**

OSPF, VXLAN, Introduction, Virtualization

**Miscellaneous (Confidential information)**

## Sisältö

<b>1</b>	<b>Johdanto .....</b>	<b>7</b>
<b>2</b>	<b>Tietoverkkotekniikan perusteita .....</b>	<b>8</b>
2.1	VLAN .....	8
2.1.1	VLAN ongelmat .....	8
2.2	Spanning tree .....	9
2.2.1	Spanning tree toiminta .....	10
2.2.2	MAC-osoitetaulu .....	12
2.2.3	Silmukka Ethernet verkossa .....	13
<b>3</b>	<b>OSI-malli.....</b>	<b>14</b>
3.1	Kerrokset .....	14
3.1.1	Physical .....	14
3.1.2	Data Link .....	14
3.1.3	Network .....	15
3.1.4	Transport .....	15
3.1.5	Session .....	15
3.1.6	Presentation.....	15
3.1.7	Application .....	16
<b>4</b>	<b>IP-osoite .....</b>	<b>16</b>
<b>5</b>	<b>VXLAN .....</b>	<b>17</b>
5.1	Ominaisuudet .....	17
5.2	VXLAN paketti ja toiminta .....	18
<b>6</b>	<b>OSPF.....</b>	<b>19</b>
6.1	Perusteet .....	19
6.2	Reititys.....	20
6.2.1	Point-to-point verkot .....	23
6.2.2	Broadcast networks .....	23
6.2.3	Non-Broadcast networks .....	23
6.3	Autonominen järjestelmä .....	23
<b>7</b>	<b>Käyttöönotto .....</b>	<b>24</b>
7.1	Virtualisointialusta GNS3 .....	25
7.1.1	GNS3 asennus ja konfigurointi.....	26
7.2	VMware Workstation .....	27
7.2.1	VMware Workstation asennus ja konfigurointi .....	28

7.3 Cisco CSR1000V .....	31
7.3.1 CSR1000V käyttöönotto ja esikonfigurointi .....	31
7.4 Topologia ja konfigurointi .....	37
<b>8 Tulokset ja johtopäätökset.....</b>	<b>47</b>
<b>9 Pohdinta.....</b>	<b>48</b>
<b>10 Lyhtenteet .....</b>	<b>50</b>
<b>Lähteet .....</b>	<b>51</b>
<b>Liitteet .....</b>	<b>53</b>
Liite 1. R1 reitittimen konfiguraatio .....	53
Liite 2. R2 reitittimen konfiguraatio .....	55
Liite 3. R3 reitittimen konfiguraatio .....	57

## Kuviot

Kuvio 1 STP topologia.....	9
Kuvio 2 Broadcast storm .....	13
Kuvio 3 VXLAN otsikko .....	19
Kuvio 4 OSPF puumalli .....	20
Kuvio 5 OSPF kättely Wireshark pakettianalysaattorissa .....	22
Kuvio 6 Prosessikaavio .....	24
Kuvio 7 GNS3 testi verkkotopologia.....	25
Kuvio 8 Hypervisor eroavaisuudet .....	27
Kuvio 9 VMware Workstation Pro .....	28
Kuvio 10 Virtual Network Editor .....	29
Kuvio 11 Windows Network Connections .....	30
Kuvio 12 CSR1000V tuonti VMware virtuaalisovellukseen.....	31
Kuvio 13 CSR1000V-virtuaalikoneen määritykset.....	32
Kuvio 14 CSR1000V liitäntä komentoja.....	33
Kuvio 15 Komentokehotteen ulostulon konfigurointi .....	34
Kuvio 16 CSR1000V uudelleenkäynnistäminen .....	35
Kuvio 17 GNS3 VMware virtuaalikoneen lisäys .....	36
Kuvio 18 Asetuksien määrittäminen .....	36
Kuvio 19 Käyttöönoton topologia .....	37
Kuvio 20 Pakettianalysaattorin käynnistys .....	39
Kuvio 21 ICMP kysely paketti analysaattorissa .....	40

Kuvio 22 OSPF pakettianalyssaattorissa .....	42
Kuvio 23 PC1 ping PC3 virtuaalikoneelle.....	45
Kuvio 24 PC1 lähettämä ARP kysely VXLAN-verkossa.....	46
Kuvio 25 PC3 virtuaalikoneen lähettämä vastaus.....	46
Kuvio 26 Pakettianalyssaattorin tuloste PC1 virtuaalikoneen ICMP kyselystä .....	47

## **Taulukot**

Taulukko 1 Porttien kustannusluvut .....	10
Taulukko 2 Porttien tyypit.....	11
Taulukko 3 Porttien tilat .....	12
Taulukko 4 OSPF-paketit .....	21
Taulukko 5 Käyttöönoton IP-osoitteet ja -alueet.....	38

# 1 Johdanto

Noin kahden vuosikymmenen ajan palvelinvirtualisointi on ollut kasvava trendi myös konesaleissa. Tämä trendi on tuonut uusia haasteita ja vaatimuksia konesalitietoverkkojen ratkaisuihin. Fyysisiltä verkkolaitteilta vaaditaan nykyään paljon sen myötä, että fyysisillä palvelimilla suoritetaan monia virtuaalisia laitteita. Yhden fyysisen portin päässä voi olla monia virtuaalisia laitteita yksilöllisillä MAC-osoitteilla, jotka verkkolaitteen on muistettava. Tietoverkkoprotokollien on pysyttävä virtualisoinnin kehityksen mukana.

Opinnäytetyö VXLAN&OSPF toteutus tutkii 2014 julkaistua VXLAN verkkoprotokollaa, jonka avulla on pyritty korjaamaan konesaliympäristöissä virtualisoinnin kasvusta johtuvia VLAN-tekniikan puutteita. VLAN-tekniikassa eri virtuaaliverkkoja omalla tunnisteella on mahdollista olla jopa 4094 kappaletta. Tämä ei kuitenkaan virtualisoinnin myötä isoissa konesaleissa enää riitä. Usein konesaleissa tarjotaan palveluja jopa sadoille eri vuokralaiselle. Vuokralaiset voivat vaatia oman verkkonsa rakenteen takia monta eri virtuaalista lähiverkkoa omalla tunnisteellansa. Tämän takia määrittäen VLAN-tekniikan 4094 tunnistetta on liian vähän OSI-mallin tasolla kaksi verkkorakenteeltaan oleviin konesaleihin.

Tavoitteena opinnäytetyössä on tarkastella tietoverkkojen perusteita ja pyrkiä viittaamaan perusteista olennaisia asioita liittyen VXLAN- ja OSPF-tekniikoihin. Verkkotekniikan perusteista on valittu aiheita, jotka liittyvät työssä käytettäviin tekniikoihin. Valinta perustuu siihen, että opinnäytetyön tavoitteena on kertoa VXLAN- ja OSPF-tekniikoista, niin että lukija saa tietoperustan ymmärtää tekniikoita.

Opinnäytetyön toteutusosiossa konfiguroidaan Ciscon C1000V virtuaalireitittimille OSPF- ja VXLAN-tekniikat. Toteutuksessa rakennetaan virtuaalisista reitittimistä ympäristö ja tarkastellaan protokollien toimivuutta pakettianalysaattorin avulla. Pakettianalysaattorin avulla pyritään myös toteamaan ja todistamaan teoriaosiossa tarkasteltujen protokollien toiminnallisuutta.

## 2 Tietoverkkotekniikan perusteita

### 2.1 VLAN

VLAN eli virtuaalilähiverkko voidaan käsittää joukkona verkkoon liitettyjä laitteita, jotka ovat loogisesti liitetty samaan verkkoon riippumatta niiden fyysisestä sijainnista. VLAN on tekniikka, joka toimii OSI-mallin toisessa kerroksessa. VLAN:n perustoimintaperiaate voidaan tiivistää siihen, että tekniikassa Data-kehikseen liitetään tagi, joka sisältää VLAN ID:n. VLAN ID on 12-bittiä pitkä luku, joten virtuaalilähiverkkoja voi kaikkiaan olla jopa 4094. Tämän ID:n avulla saadaan luotua virtuaalisia verkkoja, jotka ovat eristyksissään toisistaan. Verkkoja eristämisestä on hyötyä tietoturvan kannalta. Luotetut laitteet voidaan määrittää eri virtuaaliseen verkkoon kuin vieraat laitteet. (C, Panek 2020.)

Esimerkiksi palveluntarjoaja ei voi laittaa kaikkia asiakkaittensa verkkosivujen palvelimien sisäverkkoja saman VLAN ID:n alle, koska tällöin ne oppisivat toistensa fyysiset MAC- ja IP-osoitteet. Tämä on tietenkin tietoturva riski, koska palvelin voi muodostaa yhteyden toiseen palvelimeen kerroksella kaksi, joka ohittaa mahdollisen palomuurin.

#### 2.1.1 VLAN ongelmat

Kasvavissa konesaleissa ja pilviympäristöjen lisääntyessä on todettu ongelmia VLAN:n joustavuuden ja skaalautuvuuden kanssa. 4094 VLAN ID:tä ei enää riitä kasvavan asiakasmäärän ja asiakkaiden tarpeisiin. Etenkin pilvipalveluiden ja virtualisoinnin kasvava määrä on tuonut VLAN:n ongelmat esiin. Skaalautuvuusongelma VLAN-tekniikassa voidaan käsittää paremmin esimerkin avulla. (Howard 2023.)

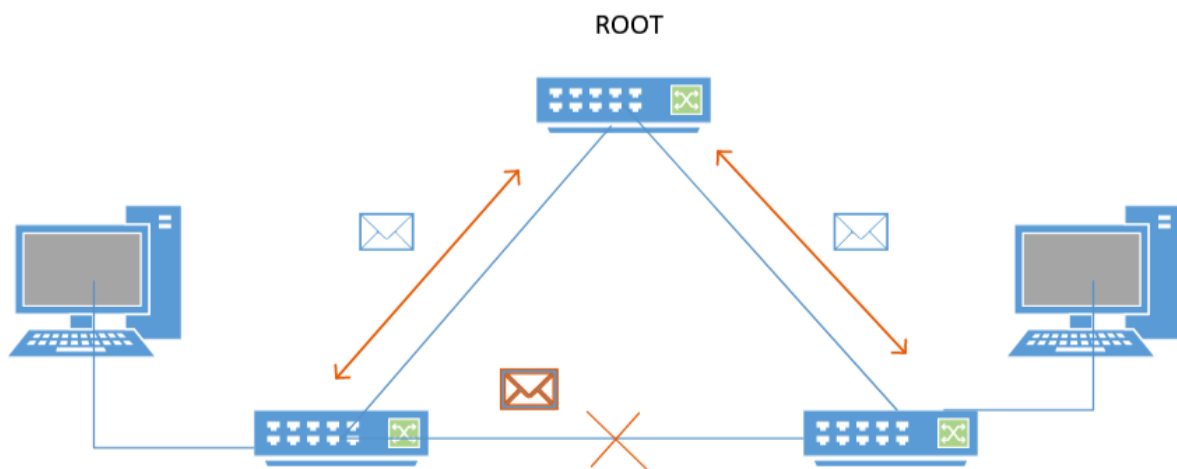
Esimerkki:

Konesalissa on kymmeniä asiakkaita ja asiakkailta kymmeniä virtuaalipalvelimia. Jokaisella palvelimella on monta verkkokorttia, jotka keskustelevalt eri VLAN ID:n alla. Asiakkailta on myös tarpeita eristää ja hallita monia osastoja ja laitteita verkoissaan. 4094 VLAN ID:tä tulisi täyttymään esimerkiksi nopeasti asiakasmäärän kasvaessa.



## 2.2 Spanning tree

Spanning tree on verkkoprotokolla, jota käytetään kerroksen kaksi verkossa silmukoiden tunnistamiseen. Protokollaa tarvitaan estämään silmukoiden syntyminen ja sen avulla voidaan luoda toimintavarmoja reittejä verkon sisällä. Spanning tree protokollasta 802.1D on tehty monia eri versioita vuosien varrella, kuitenkin näissä eri versioissa ilmenee hyvin samantapaisia toimintatapoja. (Spanning Tree Protocol Overview 2023.)



Kuvio 1 STP topologia

Spanning tree protokollan on kehittänyt Radia Perlman vuonna 1990. Protokolla kehitettiin, koska tavoitteena oli saada aikaiseksi toimintavarma verkko, joka voisi sisältää ylimääräisiä linkkejä. Tavoitteena oli kehittää looginen silmukka vapaa verkko, josta voitaisiin dynaamisesti estää ylimääräiset linkit. Ylimääräisten tai pikemminkin varalinkkien avulla oli mahdollista saada verkosta toimintavarma aktiivisen linkin alas mentäessä. (C. Paciello 2016.)

IEEE standardisoi Spanning Tree protokollan nimellä 802.1D vuonna 1990. Tähän aikaan loogisia verkkoja eli virtuaalisia lähiverkkoja ei vielä ollut, joten protokolla käyttö keskittyi yhteen lokaaliin verkkoon. STP:stä kehitettiinkin Ciscon toimesta kehittyneempi Per-VLAN ST PLUS PVST+ protokolla. (C. Paciello 2016.)

### 2.2.1 Spanning tree toiminta

Sisäverkossa kytkimien kesken STP-protokolla laskee Spanning tree algoritmilla (STA) juurikytkimen, joka on verkossa niin sanottu pääkytkin. STA-protokolla laskee parhaat reitit muilta kytkimiltä pääkytkimelle. Äänestysprosessissa siis äänestetään pääkytkin ja parhaat reitit muilta kytkimiltä pääkytkimelle. Tämän prosessin avulla kytkimet voivat asettaa portteja Blocking-tilaan, jotta saadaan aikaiseksi verkko, jossa ei ole silmukoita. (C. Paciello 2016.)

Kytkimet omaavat BID-luvun eli Bridge ID:n. BID-luku on tyypillisesti kahdeksan tavun kokoinen, joka muodostuu kahden tavun kokoisesta Bridge Prioritystä ja kuuden tavun kokoisesta MAC-Osoitteesta. STA käyttää pääkytkimen valitsemisessa BID-lukuja vertaillen niitä. Pienin BID-luvun omaava kytkin valitaan pääkytkimeksi. Pääkytkin on mahdollista asettaa myös manuaalisesti, jolloin kytkimelle asetetaan pienin BID-luku. (C. Paciello 2016.)

Taulukko 1 Porttien kustannusluvut

Portin nopeus	STP portin kustannusluku
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2

Jokaisen kytkimen portilla on STP:tä käyttävässä verkossa oma kustannusluku. Kytkimet määrittävät parhaan reitin pääkytkimelle vertailemalla kustannuslukuja. Alimman kustannusluku arvon omaava portti voittaa äänestys prosessin ja portista määritetään Root Port eli paras reitti

pääkytkimelle. Porttien kustannusluvut ovat myös käyttäjän määritettävissä eli konfiguroitavissa. (Spanning Tree Protocol Overview 2023.)

Monessa tapauksessa porttien kustannusluvut ovat arvoltaan samat. Tällöin STA ottaa huomioon myös BID-luvun, jonka avulla määritetään alimman arvon omaava portti. Tapauksessa, jossa porttien kustannusluvut ja BID-luvut täsmäävät, otetaan mukaan vielä porttien ID-luku. ID:n avulla saadaan viimeistään määritettyä jostain portista Root Port eli paras reitti pääkytkikemelle. (C. Paciello 2016.)

Taulukko 2 Porttien tyypit

Blocking	Portti ei välitä viestejä, mutta lukee BPDU viestejä.
Listening	Portti ei välitä viestejä, mutta lukee BPDU viestejä.
Learning	Portti ei välitä viestejä, mutta lukee BPDU viestejä ja päivittää MAC-Osoitetaulua.
Forwarding	Portti välittää saamansa liikenteen.
Disabled	Portti on konfiguroitu pois päältä.

Taulukko 3 Porttien tilat

Root Bridge	Juurikytkin
Root port	Paras reitti juurikytkimelle.
Designated Port	Forwarding-tilassa. Välittää liikennettä.
Non-designated Port	Blocking-tilassa

### 2.2.2 MAC-osoitetaulu

MAC-osoite on tyypillisesti 48-bitin pituinen luku, joka esitetään 12 heksadesimaali merkin avulla. MAC-osoite on uniikki tunniste jokaisella laitteella, eikä se tyypillisesti ole vaihdettavissa. MAC-Osoite on yleensä ns. kovakoodattu laitteen verkkokorttiin.

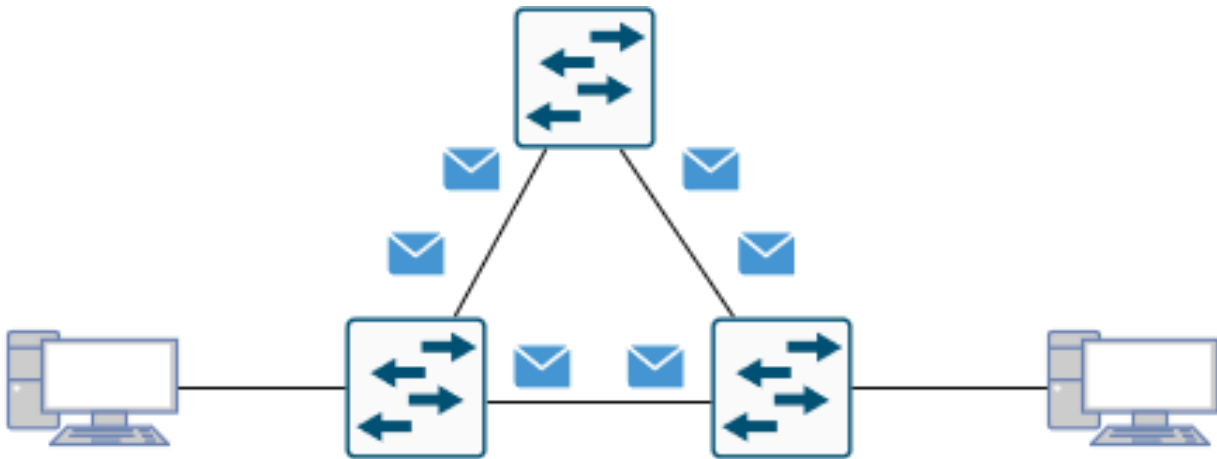
Esimerkki:

0A-D2-3E-E4-98-F0 Langattoman Wifi-adapterin MAC-osoite

Ethernet verkossa kytkimet ylläpitävät MAC-osoitetaulua, jonka avulla ne tietävät mihin porttiin saatu kehys pitää lähettää, jotta haluttu laite saavuttaa viestin. Kytkimen vastaanottaessa viestin kytkin tarkistaa, onko kohdeosoite MAC-osoitetaulussa. MAC-osoitteen puuttuessa taulusta, kytkin lähettää kaikista muista porteistaan paitsi siitä, mistä kehys on vastaanotettu yleislähetyskyselyn, jolla kytkin pyrkii selvittämään missä MAC-osoitteen omaava laite sijaitsee. Laite, joka tietää kysytyn osoitteen vastaa kytkimelle ja kytkin lisää MAC-osoitteen tauluunsa lukemalla data kehysessä olevan lähde MAC-osoitteen. Kytkimen MAC-osoitetaulussa lukee siis missä fyysisessä tai loogisessa portissa jokainen opittu MAC-osoite sijaitsee. (How Switches Work N.d.)

### 2.2.3 Silmukka Ethernet verkossa

Spanning tree protokolla on ratkaisu silmukoihin Ethernet verkossa. Ilman Spanning Tree protokollaa silmukan sisältävässä kytkin verkossa esiintyy ainakin kolmea ongelmaa. Seuraavat ongelmat koskevat vain siis sisäverkkoja, joissa ei ole käytössä STP:tä.



Kuvio 2 Broadcast storm

Yleislähetys myrsky syntyy verkossa, kun kytkin vastaanottaa broadcast viestin tai unicast viestin tuntemattomalla osoitteella. Kytkin lähettää viestin kaikista muista, paitsi vastaanotetusta portista ulos. Viestin saavuttua muille kytkimille, ne tekevät saman asian, eli lähettävät viestin omista portistaan ulos. Viesteistä syntyy loputon kierre verkkoon, joka loppuu, kunnes linkki kytkinten välistä irrotetaan tai joku kytkimistä kaatuu todella suuren kuorman takia.

Epävakaa MAC-osoitetaulu syntyy myös silmukkaisessa verkossa. Kuvio 2 verkossa jokaisessa kytkimessä on kaksi linkkiä toisiin kytkimiin. Epävakaus MAC-osoitetaulussa johtuu siitä, kun kytkimen vastaanottavat viestin samalta MAC-osoitteelta eri porteista, jolloin kytkin tallentaa MAC-Osoitetauluunsa osoitteen vastaamaan eri porttia kaiken aikaa.

Duplikaatti viestit ovat myös yleisiä silmukkaisessa verkossa. Kuvio 2 verkossa, vain oikeanpuoleinen kytkin tietää oikeanpuoleisen tietokoneen MAC-osoitteen ja muut kytkimet eivät. Vasen tietokone lähettää oikealla olevalle viestin. Viesti menee suoraan oikeanpuoleisen kytkimen kautta

tietokoneelle. Virhe tapahtuu, kun vasemmanpuoleinen kytkin ei tiedä oikeanpuoleisen tietokoneen osoitetta, se lähettää viestin myös ylhäällä olevalle kytkimelle, joka puolestaan lähettää viestin edelleen oikealle. Tietokone vastaanottaa viestin kahteen kertaan. Tämä voi olla huono tilanne, jos tietokoneella esimerkiksi ohjataan jotain järjestelmää.

### **3 OSI-malli**

OSI-malli eli Open Systems Interconnection reference model on käsitteellinen rakenne tai malli, joka määrittää verkossa laitteiden välillä tapahtuvan kommunikaation seitsemän kerroksen avulla. Jokaiselle kerrokselle on oma määritelty toimintonsa tai tehtävä. OSI-malli on hyödyllinen tietoverkkojen kehittämiseen, opetteluun ja rakentamiseen. (Panek 2020.)

OSI-malli on International Organization for Standardization kehittämä. Ensimmäinen raakaluonnoksen OSI-mallista esitteli Hubert Zimmermann vuonna 1978. Jalostetun version OSI-Mallista esitteli vuonna 1980 ISO. (Russell 2013.)

#### **3.1 Kerrokset**

##### **3.1.1 Physical**

Ensimmäinen kerros eli toisella nimellä Ethernet-kerros on OSI-mallin alin kerros. Kerros on vastuussa käsittelemättömän datan kuljettamisesta. Kerroksen alaisuudessa määritellään mm. Tiedonsiirtonopeus, liitântätapa, jännite, liitännän suuntaisuus. Esimerkkinä kerroksen vastuulle kuuluu RJ-45 kaapelin kahdeksan pinnin järjestys ja kuinka ne ovat liitettynä verkkolaitteisiin. (Phani 2015.)

##### **3.1.2 Data Link**

Toinen kerros eli Data Link mahdollistaa datakehyksien liikkumisen fyysisen Ethernet kerroksen päällä verkkolaitteiden välillä. Datakehykset mahdollistavat monien erilaisten protokollien toiminnallisuuden, jonka avulla yhteyden muodostaminen, datan välittäminen, virheentunnistaminen- ja korjaus ovat mahdollisia. Tämän tekee mahdolliseksi MAC-osoitteet eli fyysiset osoitteet laitteiden verkkokorteissa. Opinnäytetyössä käytävä VLAN- ja VXLAN-tekniikat tulevat mukaan Data Link kerroksella. Näiden tekniikoiden avulla sisäverkko voidaan jakaa loogisiin alueisiin. (Phani 2015.)

### 3.1.3 Network

Kolmas kerros eli Network on vastuussa reitittämisestä ja tiedon jakamisesta yhden verkon sisällä tai monen eri verkon välillä. Kerroksessa kulkee paketteja, jotka sisältävät otsikon, jossa on tietoa lähde ja kohde IP-osoitteista. Ipv4 tukevat laitteille jaetaan tai asetetaan 32-bittiä pitkä osoite. Kolmas kerros käyttää näitä osoitteita määrittääkseen missä verkkolaite sijaitsee. (Phani 2015.)

### 3.1.4 Transport

Neljäs kerros eli Transport, toisella nimellä kuljetuskerros toimii ylempien kerrosten viestin välittäjänä eli prosessina. Paketin otsikkoon lisätään porttinumero, jolloin verkkolaitteet ymmärtävät viestin päämäärän ja viesti menee oikealle ylemmän kerroksen sovellukselle. Kerros mahdollistaa myös paremman virhekorjauksen kuin Data Link kerros, koska paketteja voidaan kuljetuskerroksen avulla mm. Segmentoida eli jakaa osiin. Tunnetuimpia protokollia kerroksella ovat UDP ja TCP. (Phani 2015.)

### 3.1.5 Session

Viides kerros eli Session on vastuussa istunnon luomisesta, ylläpidosta ja päättämisestä. Istunnossa tapahtuu kommunikaatiota, joka koostuu pyynnöistä ja vastauksista eri vastapuolien kesken. Istunnolla tarkoitetaan kahden pääteen tai kommunikaatiolaitteen välistä aktiivista aikaikkunaa. Kerroksen tärkeä tehtävä on istunnon ylläpito. Ylläpitäminen tarkoittaa esimerkiksi livelähetyksessä äänen ja kuvan synkronointia tai istunnon uudelleenkäynnistämistä vikatilanteissa. (What is Layer 5 of the OSI Model: Session Layer? N.d.)

### 3.1.6 Presentation

Kuudes kerros eli Presentation vastaa siitä, että data on oikeassa muodossa lähettäjältä vastaanottajalle. Tämä tarkoittaa sovelluskohtaisesti esimerkiksi datan salausta, pakkaamista tai datan muodon muuttamista. Yksinkertaisesti sanottuna data muutetaan kerroksella siihen muotoon, jolla sovellus sen hyväksyy. (What is Layer 6 of The OSI Model: Presentation Layer? N.d.)

### 3.1.7 Application

Seitsemäs kerros eli Application on OSI-mallin ylin kerros. Kerros ei tarkoita käytettävää sovellusta, vaikka nimi niin antaisikin ymmärtää. Application kerroksen vastuulla on määrittää sovelluksen käyttämät protokollat datan lähettämiseen ja kertoa muille kerroksille sovelluksen asettamia määrittämiä. (Panek 2015.) Sovelluskerrokseen kuuluu protokollista sähköpostiviestinnässä käytetty POP3 ja selainkäytössä olevat HTTP ja HTTPS. (What is OSI Model? N.d)

## 4 IP-osoite

IP-osoite on fyysiseen MAC-Osoitteeseen verrattuna looginen, jota voidaan käyttöjärjestelmän sisällä monissa tapauksissa vaihtaa. IP-Osoite on uniikki ja 32-bittiä pitkä osoite. IP-Osoite tarvitaan kaikille laitteille, jotka osallistuvat viestintään verkossa ja pitää uniikisti tunnistaa. (Panek 2015.)

Esimerkkinä yleisesti kuluttajan kotisisäverkossa IP-osoitealue on binäärimuodossa 192.168.1.0/24 muotoinen verkko. Yhden ja kolmen pituiset numerosarjat ovat erotettu pisteillä, josta näkyy, että 32-bittiä on eroteltu neljään kahdeksan bitin osaan. Numerosarjan lopussa on vinoviivalla erotettu numero 24, joka kertoo verkon koon ja alueen. Numeroa kutsutaan nimellä prefiksi.

Laitteen IP-osoite tässä verkossa voi olla 192.168.1.5 eli viimeinen numero kertoo laitteen uniikin osoitteen. Numero /24 kertoo verkon olevan 0-255 välillä, joten verkossa laitteilla voi olla 2-254 päättyvä IP-Osoite. IP-osoite, jonka viimeinen osa on arvoltaan yksi, käytetään yleensä oletusyhdyskäytävänä, joka määrittää mistä osoitteesta päästään muihin verkkoihin. Oletusyhdyskäytävänä toimii yleensä reititin. IP-osoite, jonka viimeinen osa on arvoltaan 255, käytetään yleislähetysosoitteena. Verkko 192.168.1.0 on binäärimuodossa ilmaistuna 11000000.10101000.00000001.00000000.



## 5 VXLAN

Palvelin virtualisoinnin kasvava määrä on tuonut haasteita tietoverkkoihin konesaliympäristöissä. Virtual extensible Local Area Network – VXLAN on nykyaikainen suuriin konesaliympäristöihin kehitetty tekniikka ja kuten RFC 7348 (2014, 3) standardissa todetaan, vastaa ominaisuuksiltaan virtualisoinnin kasvavaan määrään. Tekniikan peruseriaate on laajentaa tavanomaisen VLAN-tekniikan ominaisuuksia skaalautuvuuden ja joustavuuden osalta. (RFC 7348 2014.)

Tekniikka on julkaistu vuonna 2014 Internet Engineer Task Forcen toimesta. Tekniikan IETF:n standardisoima virallinen dokumentti on RFC7348. Tekniikkaa on suunniteltu yhteistyössä ja VXLAN kehityksessä on ollut mukana monia suuria laite- ja sovelluskehittäjiä mm. Cisco, Broadcom, VMWARE ja Intel. (RFC 7348 2014.)

### 5.1 Ominaisuudet

VXLAN-verkko on virtualisoitu verkko, joka toimii OSI-mallin kerroksen kaksi verkon toimivuuden IP-verkon päällä. Tämä mahdollistaa liikennöinnin saman muista eristetyn virtuaaliverkon sisällä sijainnista riippumatta. Esim. Eri konesaleissa olevat virtuaalipalvelimet voivat liikennöidä samassa verkossa VXLAN-tekniikan avulla täysin eristetyksi muusta liikenteestä. (What is VXLAN? 2023.) VXLAN verkko toimii opinnäytetyössä OSPF-verkon päällä.

VXLAN mahdollistaa laajasti verkkojen segmentoinnin yli 16 miljoonaan omaan verkkoon uniikilla tunnisteella. VXLAN-tekniikka toimii teknisesti kapseloimalla kerroksen kaksi datakehysiksi kerroksen neljä UDP-paketeiksi. kapselointi ja sen purku tapahtuu VTEP:ssä eli päätepisteissä. (What is VXLAN? 2023.)

Tekniikan toimiessa IP-verkon päällä, ei 2.2 Spanning tree osiossa tarkasteltua Spanning Tree protokollaa tarvita. Protokolla on ongelmallinen kerroksen kaksi konesaliverkossa, koska se ajaisi kaikki linkkejä alas. Myös RF7348 dokumentissa (RFC 7348 2014, 5) todetaan STP-protokollan olevan rajoittava tekijä kerroksen kaksi konesaliverkossa.

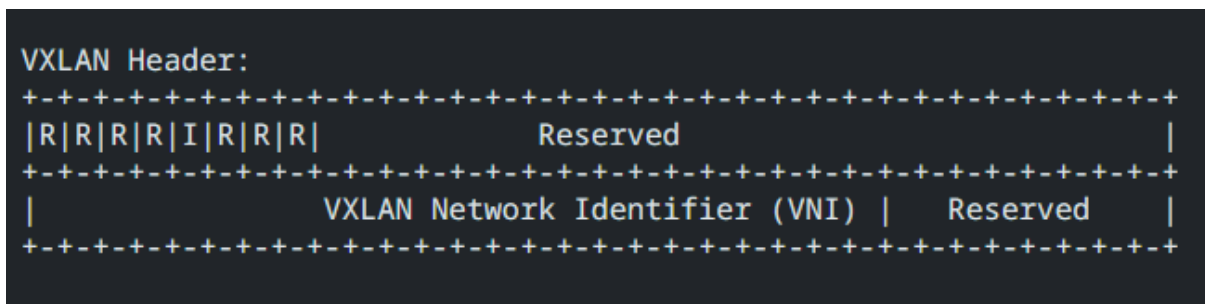
## 5.2 VXLAN paketti ja toiminta

VLAN:ssa käytetään 12-bittiä pitkää VLAN ID:tä, joka mahdollistaa 4094 uniikkia virtuaaliverkkoa. VXLAN-tekniikassa VXLAN-segmentissä on 24-bittiä pitkä tunnistekenttä, joka mahdollistaa yli 16 miljoonaa uniikkia virtuaaliverkkoa. Tätä tunnistekenttää kutsutaan lyhenteellä VNI – VXLAN Network Identifier. (RFC 7348 2014.) Täten skaalautuvuus ja segmentoinnin mahdollisuudet kasvavat suuresti siirryttäessä VLAN:sta VXLAN-tekniikkaan. VLAN ongelmat osiossa (VLAN ongelmat, 2.1.1) käsitellään VLAN ongelmia, jossa esitellään esimerkin avulla yli neljän tuhannen VLAN tunnisteen riittämättömyyttä konesalissa.

VTEP eli VXLAN Tunnel End Point on yksi VXLAN-tekniikan toiminnallisuuden mahdollistavista komponenteista. Virtual Tunnel End point on rajapinta, joka aloittaa tai lopettaa VXLAN tunneleita. VTEP:n tehtävä on kapseloida tuleva liikenne ja lähettää se kohde VTEP:lle, jonka tehtävä on purkaa kapseloitu liikenne. (RFC 7348 2014.) VTEP rajapinta voi sijaita fyysisillä kuin virtualisoiduillakin kytkimillä ja reitittimillä. (What is VXLAN? 2023).

Käyttöön otossa VTEP rajapinnat konfiguroidaan käyttämään toistensa löytämiseen ja kommunikointiin Multicast ryhmiä. Multicast ryhmän avulla rajapinnat pitävät yllä tietoja toisistaan ja muodostavat tunneleita liikenteen kuljettamiseksi. Rajapinnat kuuntelevat Multicast ryhmää ja lähettävät toisillensa verkossa herätteitä, jotta ne tietävät toisen rajapinnan tilasta ja olemassaolosta.

VTEP vastaanottaessa kehyksen, VTEP liittää kehykseen kahdeksan tavun kokoisen VXLAN otsikon, joka sisältää 8-bittisen lipukkeen ja 24-bittiä pitkän VXLAN tunnisteen VXLAN ID/VNI. VXLAN otsikon lipukkeen I- bitti pitää olla arvoltaan 1, jotta VNI todetaan todelliseksi. Lopussa olevat 24 -ja 8-bittiä pitkät kentät ovat varattuina tulevaisuutta varten ja niiden arvo on nolla. (RFC 7348 2014, 5.)



Kuvio 3 VXLAN otsikko

VXLAN otsikon lisäksi kehykseen liitetään ulkoinen UDP-otsikko, IP-otsikko ja Ethernet-otsikko. UDP-otsikko sisältää tietoja lähde- ja kohdeporteista. Kohdeportti on yleensä vakio ja IANI on asettanut VXLAN UDP kohde portin arvoksi 4789. Arvoa on kuitenkin mahdollista vaihtaa. Lähdeportti on VTEP määrittämä. UDP otsikko pitää sisällään myös UDP Checksum tarkistussumma kentän, jonka arvo pitäisi olla nolla. Arvon ollessa nolla, vastaanottavat VTEP:n tehtävänä on purkaa kapse-loitu paketti. Arvon ollessa eriävä nollasta, VTEP suorittaa virheentarkistuksen paketille ja mahdol-lisesti jopa pudottaa paketin. (RFC 7348 2014.)

## 6 OSPF

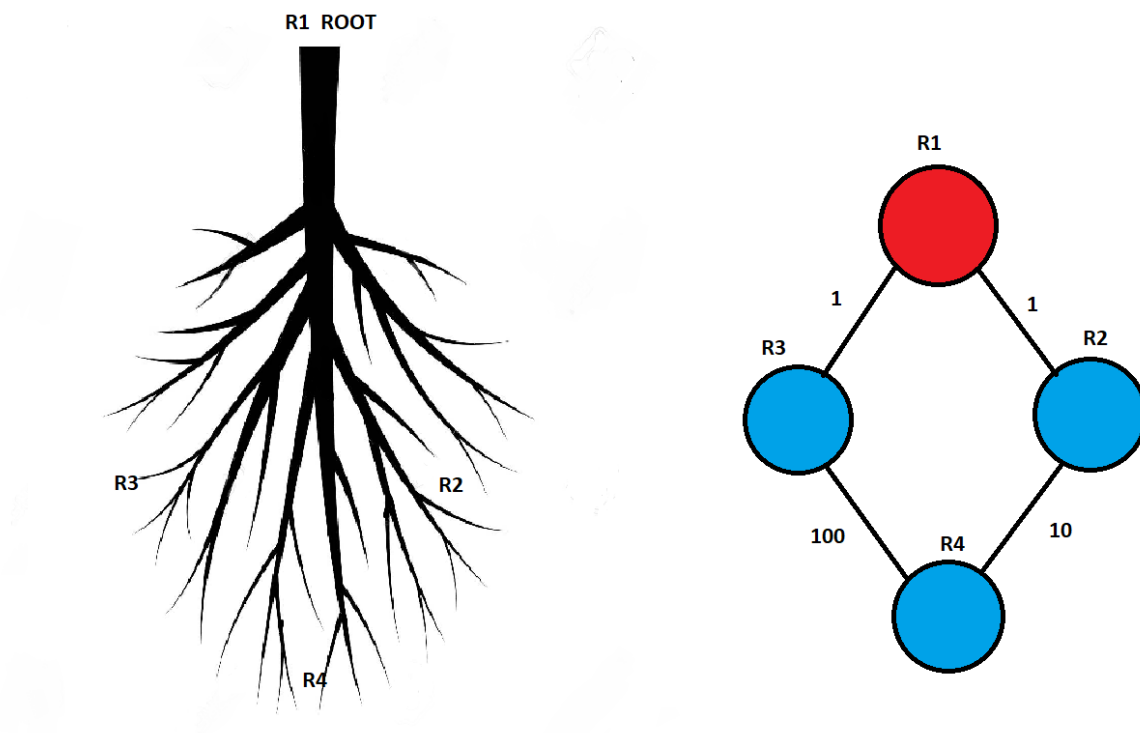
### 6.1 Perusteet

Open Shortest Path First on TCP/IP verkkoihin suunniteltu avoimen standardin reititysprotokolla. Protokollan on kehittänyt OSPF työryhmä Internet Engineer Task Forcesta. Ensimmäinen versio protokollasta julkaistiin vuonna 1989, mutta OSPFv2 esiteltiin jo vuonna 1991 sisältäen merkittäviä parannuksia ja virheen korjauksia, kuten RFC 1247 dokumentissa mainitaan. (RFC1247 1991, 181.)

OSPF toiminta perustuu ympäristön luomiseen reitittimien välillä. OSPF on linkkilallinen reititys-protokolla, joka tarkoittaa reitittimet lähettävät järjestelmän sisällä tietoja linkkien tilasta toisil-leen ja rakentavat näistä tiedoista tietokannan. OSPF käyttää Dijkstran lyhimmän reitin algoritmia laskeakseen verkkoliikenteelle kaikista parhaan reitin kulkea. Reititysprotokolla luokitellaan IGP järjestelmäksi, koska reititys tapahtuu yhden autonomisen järjestelmän sisällä. (RFC2328 1998, 5.)

## 6.2 Reititys

Verkossa, jossa reititys tapahtuu OSPF-protokollan avulla, jokainen reititin käyttää Dijkstran algoritmia laskeakseen linkkitila tietokannastaan lyhimmat reitit muihin kohteisiin. Jokainen reititin rakentaa niin sanotun puumallin, jossa algoritmia käyttävä reititin laskee itsestään parhaat reitit verkossa oleviin kohteisiin. Puumallissa kohteet ovat kuvattu lehtinä. Puumallia kuvataan kuviossa neljä verrattuna verkkotopologiaan neljän reitittimen avulla.



Kuvio 4 OSPF puumalli

Dijkstran algoritmin on kehittänyt vuonna 1959 Edsger W. Dijkstran ja se on luotu lyhimmän polun tai reitin etsimiseen verkosta eri solmujen välillä. Käytännössä algoritmia käytetään tietoliikenneyhteyksien reitityksessä, jotta verkosta ja siihen liittyvästä reitityksestä saadaan mahdollisimman nopea mahdollisimman pienillä kustannuksilla. Algoritmi luotiin aluksi nimenomaan etsimään lyhin reitti kahden solmun välillä, mutta uusissa versioissa algoritmi on rakennettu niin, että se käy kaikki reitit läpi annetusta niin sanotusta lähdesolmusta.

Reitittimet jakavat tietoja linkkien tiloista toisillensa, jonka avulla ne päivittävät ja ylläpitävät tietokantaansa. Reitittimet lähettävät toisillensa paketteja, jotka sisältävät erilaisia tietoja. Pakettien tyyppejä on viisi erilaista.

Taulukko 4 OSPF-paketit

Paketin tyyppi	Selitys
Hello	Selvittää naapurit tai pitää yhteyttä yllä naapureihin.
Database Description	Tietoja tietokannan sisällöstä
Link State Request	Pyytää tietoja toisen reitittimen tietokannan sisällöstä
Link State Update	Lähetää tietoja toiselle reitittimelle tietokannan sisällöstä
Link State Ack	Kertoo vastaanottaneensa tietoja

Reitittimien tavoite OSPF-verkossa on pitää samanlainen tietokanta toistensa kanssa koko ajan.

Tämä tiedonvaihto ja tietokannan ajankohtaisena pysyminen mahdollistaa nopean reitityksen myös tilanteissa, joissa verkosta menee linkki alas.

Hello paketteja lähetetään jokaisesta toiminnallisesta reitittimen liitännästä ulospäin. Hello pakettilla reitittimet ylläpitävät ja löytävät naapuruussuhteita muihin reitittimiin. Pakettien avulla myös äänestetään Designated ja Backup Designated reitittimet yleislähetysverkossa. Sisällössä on aina ilmoitettu nykyinen päätös DR ja BDR reitittimien arvoista. Arvo on 0.0.0.0, jos päätöstä DR ja BDR reitittimistä ei ole tehty.

Äänestysprosessissa reitittimet käyttävät algoritmia ja tietoja muista reitittimistä Hello pakettien avulla. Korkeimman prioriteetin eli Router priority ID:n omaava reititin voittaa prosessin ja siitä tulee verkon DR reititin. Vakiona Router priority ID arvo on yksi, joten prosessissa voi hyvinkin tulla vastaan tilanne, jossa kaikilla reitittimillä on sama arvo. Tällöin algoritmi ottaa prosessiin mukaan

reitittimien ID:n eli Router ID:n, joista se valitsee korkeimman arvon. Router ID voi olla konfiguroimatta, jolloin algoritmi ottaa mukaan reitittimien virtuaalisen liitännän IP-Osoitteen arvon. Arvon ollessa konfiguroimatta prosessiin otetaan mukaan vielä fyysisen liitännä.

Designated Reitittimen tehtävänä on ylläpitää ajankohtaista tietokantaa verkon linkkien tiloista. Linkin mennessä verkossa alas, reititin lähettää naapureilleen viestin asiasta. Muut reitittimet paitsi DR reititin hylkäävät viestin. DR reititin päivittää tietokantaansa linkin menneen alas ja lähettää tiedon naapurireitittimille, jolloin reitittimet hyväksyvät viestin, koska se on tullut DR reitittimeltä.

No.	Time	Source	Destination	Protocol	Length	Info
7	10.283163	192.168.35.1	224.0.0.5	OSPF	114	Hello Packet
8	10.284248	192.168.35.2	192.168.35.1	OSPF	114	Hello Packet
9	10.564609	192.168.35.2	224.0.0.5	OSPF	114	Hello Packet
10	19.301584	192.168.35.1	224.0.0.5	OSPF	114	Hello Packet
11	19.733122	192.168.35.2	224.0.0.5	OSPF	114	Hello Packet
12	29.042279	192.168.35.2	224.0.0.5	OSPF	114	Hello Packet
13	29.212171	192.168.35.1	224.0.0.5	OSPF	114	Hello Packet
14	38.599481	192.168.35.2	224.0.0.5	OSPF	114	Hello Packet
15	38.895282	192.168.35.1	224.0.0.5	OSPF	114	Hello Packet
16	40.983743	192.168.35.1	192.168.35.2	OSPF	78	DB Description
17	41.281269	192.168.35.2	192.168.35.1	OSPF	78	DB Description
18	41.282904	192.168.35.1	192.168.35.2	OSPF	138	DB Description
19	41.283588	192.168.35.2	192.168.35.1	OSPF	138	DB Description
20	41.284040	192.168.35.1	192.168.35.2	OSPF	94	LS Request
21	41.284055	192.168.35.1	192.168.35.2	OSPF	78	DB Description
22	41.284794	192.168.35.2	192.168.35.1	OSPF	190	LS Update
23	41.284821	192.168.35.2	192.168.35.1	OSPF	94	LS Request
24	41.285740	192.168.35.1	192.168.35.2	OSPF	190	LS Update
25	41.289271	192.168.35.2	224.0.0.5	OSPF	94	LS Update
26	41.322218	192.168.35.2	224.0.0.5	OSPF	158	LS Update
27	41.338653	192.168.35.1	224.0.0.6	OSPF	110	LS Update
28	41.355156	192.168.35.2	224.0.0.5	OSPF	110	LS Update
29	43.580403	192.168.35.1	224.0.0.6	OSPF	94	LS Update
30	43.580796	192.168.35.2	224.0.0.5	OSPF	94	LS Update
31	43.613396	192.168.35.1	224.0.0.6	OSPF	158	LS Update
32	43.615283	192.168.35.2	224.0.0.5	OSPF	158	LS Update
33	43.786806	192.168.35.2	224.0.0.5	OSPF	138	LS Acknowledge
34	43.787474	192.168.35.1	224.0.0.6	OSPF	178	LS Acknowledge
35	45.806247	192.168.35.2	224.0.0.5	OSPF	110	LS Update

Kuvio 5 OSPF käyttö Wireshark pakettianalysaattorissa

### 6.2.1 Point-to-point verkot

Point-to-point verkolla kuvataan verkkoa, jossa on kaksi reititintä yhdistettyinä suoraan toisiinsa muodostaen naapuruuden toistensa kanssa. OSPF reititysprotokollaa käytettäessä Point-to-Point verkossa reitittimet jakavat naapuruuden toistensa kanssa. Point-To-Point verkko on tunnistettavissa siitä, että verkossa paketilla on aina yksi ja sama vastaanottaja. (RFC 2328 1998.)

### 6.2.2 Broadcast networks

Yleislähetysverkoissa on enemmän kuin kaksi reititintä. Verkossa on mahdollista lähettää yleislähetysviestejä ja saavuttaa jokainen verkossa oleva reititin. OSPF yleislähetysverkoissa on tarve DR ja BDR tyyppisille reitittimille. DR ja BDR reitittimet ovat vastuussa verkossa olevien reitittimien naapuruussuhteiden ylläpidosta ja OSPF päivitysviestien hallinnointi. (RFC 2328 1998.)

### 6.2.3 Non-Broadcast networks

Non-Broadcast verkoissa on monta tai enemmän kuin kaksi reititintä. Verkossa reitittimillä ei ole kuitenkaan yleislähetys kykyä. Toisiinsa liitetyt reitittimet pitävät naapuruussuhteitaan yllä OSPF-Hello pakettien avulla. OSPF-reititysprotokolla voi toimia kahdessa tilassa Non-Broadcast verkossa NBMA tai Point-to-MultiPoint. (RFC 2328 1998.)

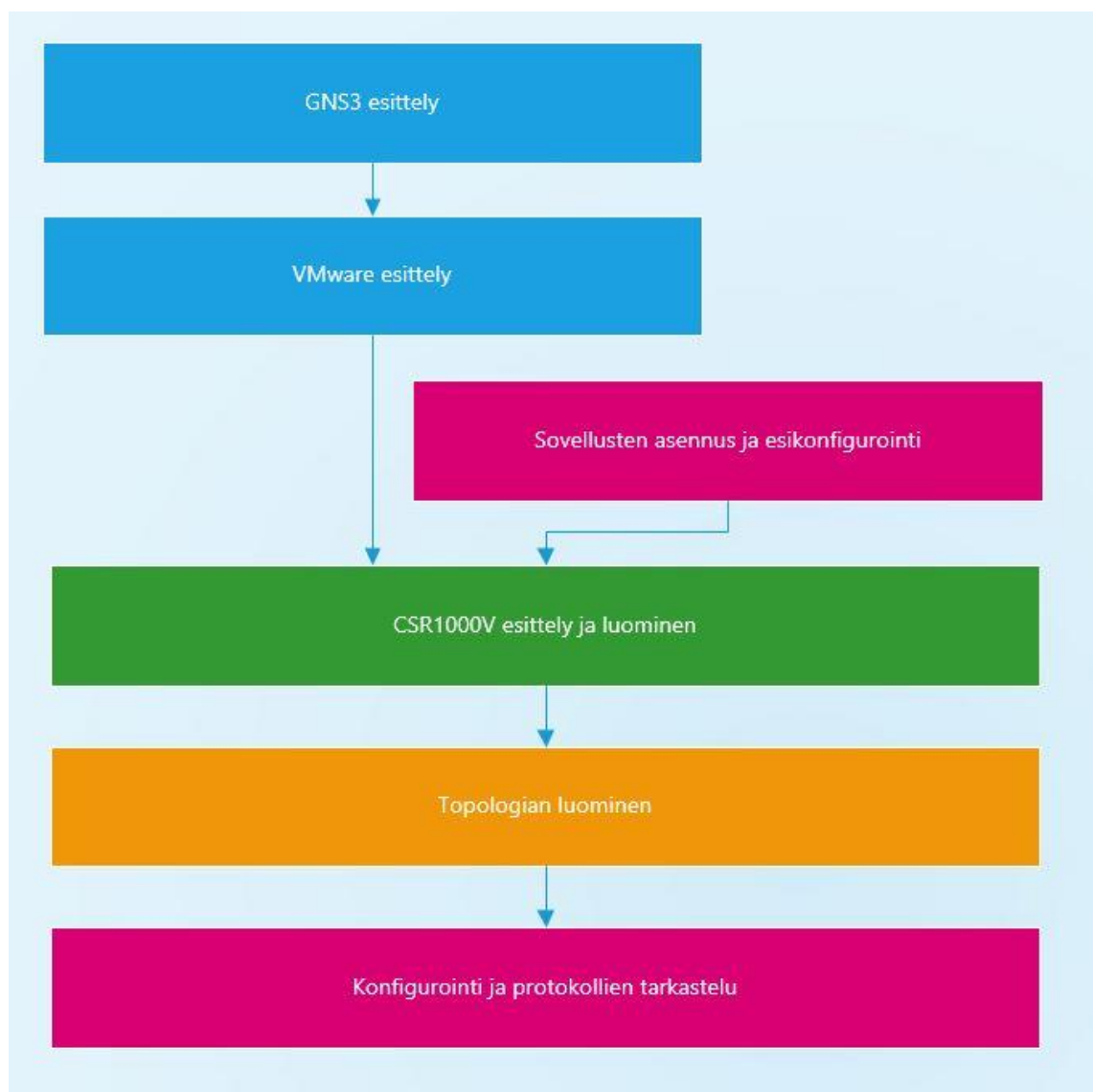
## 6.3 Autonominen järjestelmä

Tietoverkkotekniikassa puhutaan autonomisista järjestelmistä. Nämä järjestelmät ovat verkkoja, joiden sisällä tai välillä reititys tapahtuu. Autonomisissa järjestelmissä käytettävät reititysprotokollat jaetaan kahteen osaan Internal Gateway Protocol ja Exterior Gateway Protocol. IGP tehtävä on hoitaa reititys autonomisen järjestelmän sisällä, kun EGP tehtävä on hoitaa reititys autonomisten järjestelmien välillä. Internal Gateway protokolliin kuuluu mm. IS-IS ja EIGRP reititysprotokollat ja opinnäytetyön toteutusosiossa käytettävä OSPF.

## 7 Käyttöönotto

Opinnäytetyön toteutuksessa käytetään GNS3 virtualisointialustaa, jonka avulla saadaan rakennettua verkkotopologia reitittimistä. Reitittimille tullaan konfiguroimaan OSPF- ja VXLAN-tekniikat ja näitä tullaan tarkastelemaan pakettianalysaattorilla. Reitittiminä käytetään Cisco laitevalmistajan CSR1000V-virtuaalireitittimiä ja reitittimet suoritetaan VMware Workstation sovelluksessa.

Käyttöönotossa edetään kuvion seitsemän mukaisesti. Sovellukset esitellään ja asennetaan omissa osiossaan, jonka jälkeen siirrytään CSR1000V-virtuaalireitittimen esittelyyn ja luomiseen sovellusten avulla.



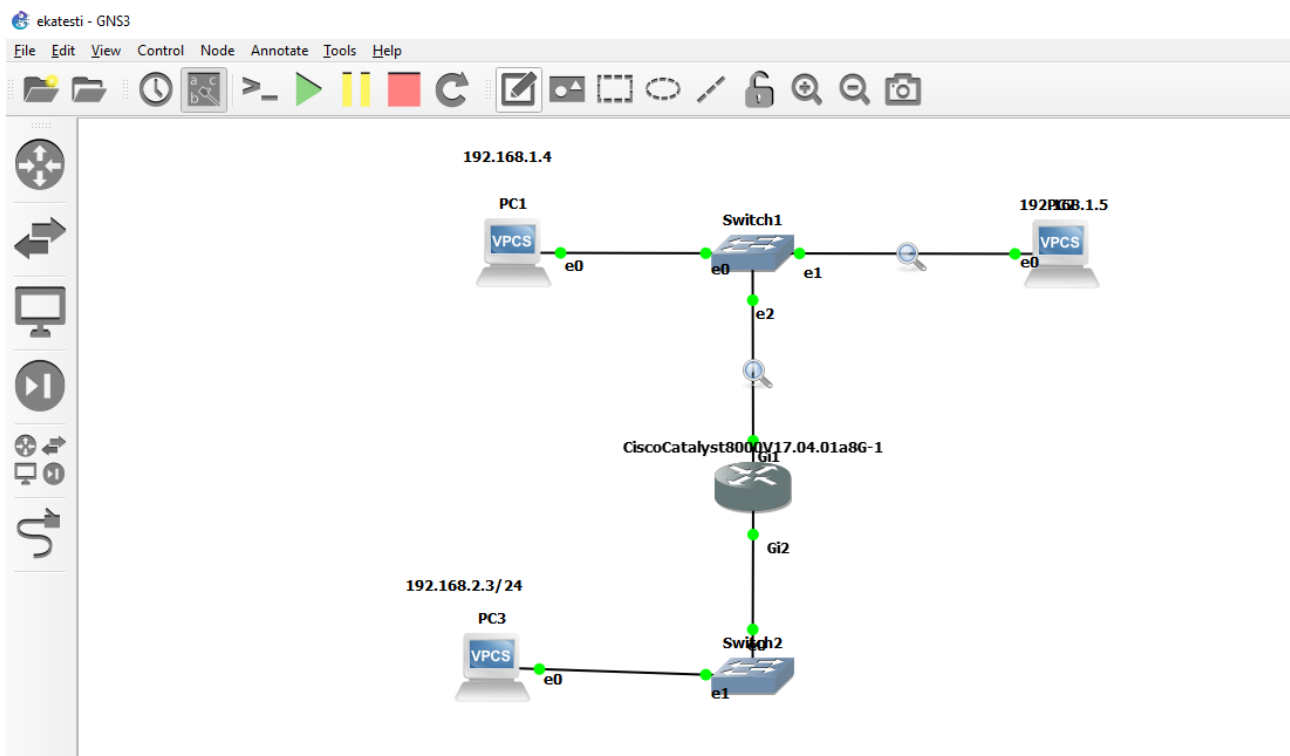
Kuvio 6 Prosessikaavio



## 7.1 Virtualisointialusta GNS3

Graphical Network Simulator 3 on avoimen lähdekoodin emulaattori- ja simulointialusta, joka mahdollistaa opinnäytetyön toteutusvaiheessa verkkolaitteiden virtualisoinnin, verkon luonnin ja laitteiden liittäminen GNS3-sovelluksen sisällä. Ohjelmisto mahdollistaa monen eri laitevalmistajan verkkolaitteiden emuloinnin, eli käyttäjän on mahdollista konfiguroida todellista verkkolaitteen käyttöjärjestelmää. Alusta tukee myös servereitä, kontteja ja hybridiliitäntöjä fyysisiin laitteisiin.

Ohjelmiston alkuperäinen kehittäjä Jeremy Grossman loi ohjelman lopputyönään CCNA ja CCNP-sertifikaattien opiskelun avuksi. Ensimmäinen versio ohjelmistosta julkaistiin 2007 vuoden puolivälissä. Alustan avulla on kätevää harjoitella verkkolaittekonfiguraatioita tekemistä ja verkkojen luontia oli tarkoitus sertifikaattiin opiskelu tai tuotantotasaisen verkon testaus ennen tuotantoon pänemistä. GNS3 yhteisöllä onkin yli 800 tuhatta jäsentä opiskelijoista arkkitehteihin. (Getting Started with GNS3. N.d.)



Kuvio 7 GNS3 testi verkkotopologia

### 7.1.1 GNS3 asennus ja konfigurointi

GNS3 ohjelmisto ladataan gns3.com viralliselta verkkosivustolta. Lataaminen vaatii yhteisöön liittymistä käyttäjätilin luomisella sähköpostiosoitteen avulla. Lataaminen ja asentaminen on suoraviivaista. On syytä seurata sovellusvalmistajan virallista ohjetta, joka löytyy seuraavasta osoitteesta <https://docs.gns3.com/docs/getting-started/installation/windows/> . Verkkosivustolla esitellään isäntäkoneen komponentti- ja käyttöjärjestelmävaatimukset. Asennusvaiheessa on mahdollista valita asennettavaksi erilaisia hyödyllisiä lisäosia. (Getting Started with GNS3.) Opinnäytetyön käyttöönottovaiheessa käytetään Wireshark pakettianalysaattoria.

GNS3 sovellus koostuu kahdesta komponentista. The GNS3-all-in-one software (GUI) on komponentti, joka toimii käyttäjän graafisena rajapintana. The GNS3 virtual machine (VM) on palvelin-komponentti, jonka avulla on mahdollista suorittaa laitteita ja palvelin prosesseja. Palvelinkomponenttia on mahdollista suorittaa paikallisena, paikallisena virtuaalikoneena tai etä-virtuaalikoneena. Opinnäytetyössä GNS3 virtuaalikonetta suoritetaan virallisen dokumentaation suosittelemalla VMware Workstation Pro Hypervisorilla. (Getting Started with GNS3.)

## 7.2 VMware Workstation

VMware Workstation on Hypervisor ohjelmisto Windows- ja Linux käyttöjärjestelmille. Hypervisorin avulla on mahdollista suorittaa virtuaalikoneita fyysisen tietokoneen sisällä. Hypervisor sovellusrajapinnan avulla virtuaalikoneille on mahdollista asettaa eri asetuksia ja määrittää mm. suoritintimien, muistin ja tallennustilan määrää. Virtuaalikoneille on myös mahdollista lisätä verkkokortteja, joka on hyvin oleellista opinnäytetyön käyttöönottoaiheessa.

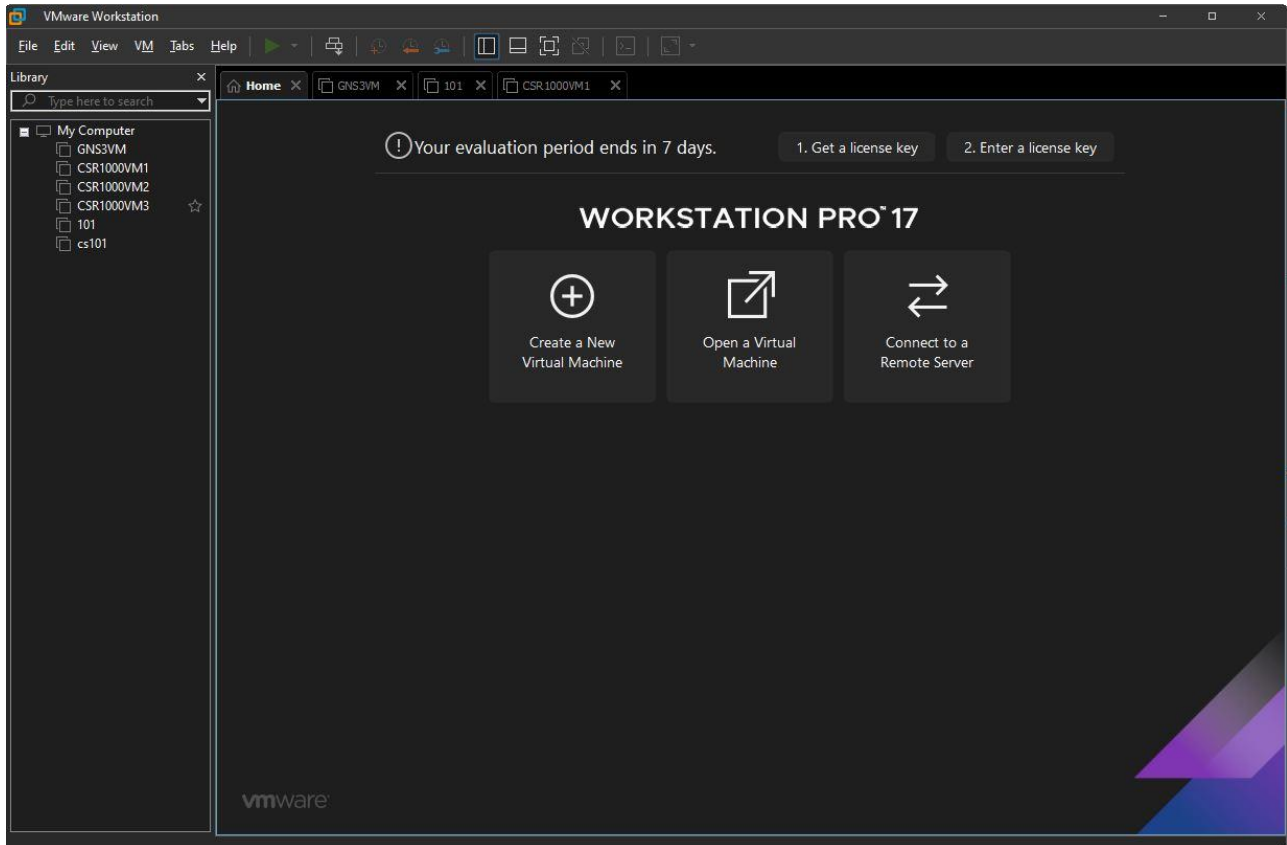
VMware workstation on tyypin kaksi Hypervisor, joka tarkoittaa Hypervisorin toimivan käyttöjärjestelmän päällä. Tyypin yksi Hypervisor suoritetaan suoraan fyysisen laitteiston päällä. Tyypin yksi Hypervisoreita ovat mm. Microsoft Hyper-V, VMwaren tuotteet vSphere ja ESX. Tyypin kaksi Hypervisoreita ovat mm. VMware Workstation ja Virtualbox.



Kuvio 8 Hypervisor eroavaisuudet

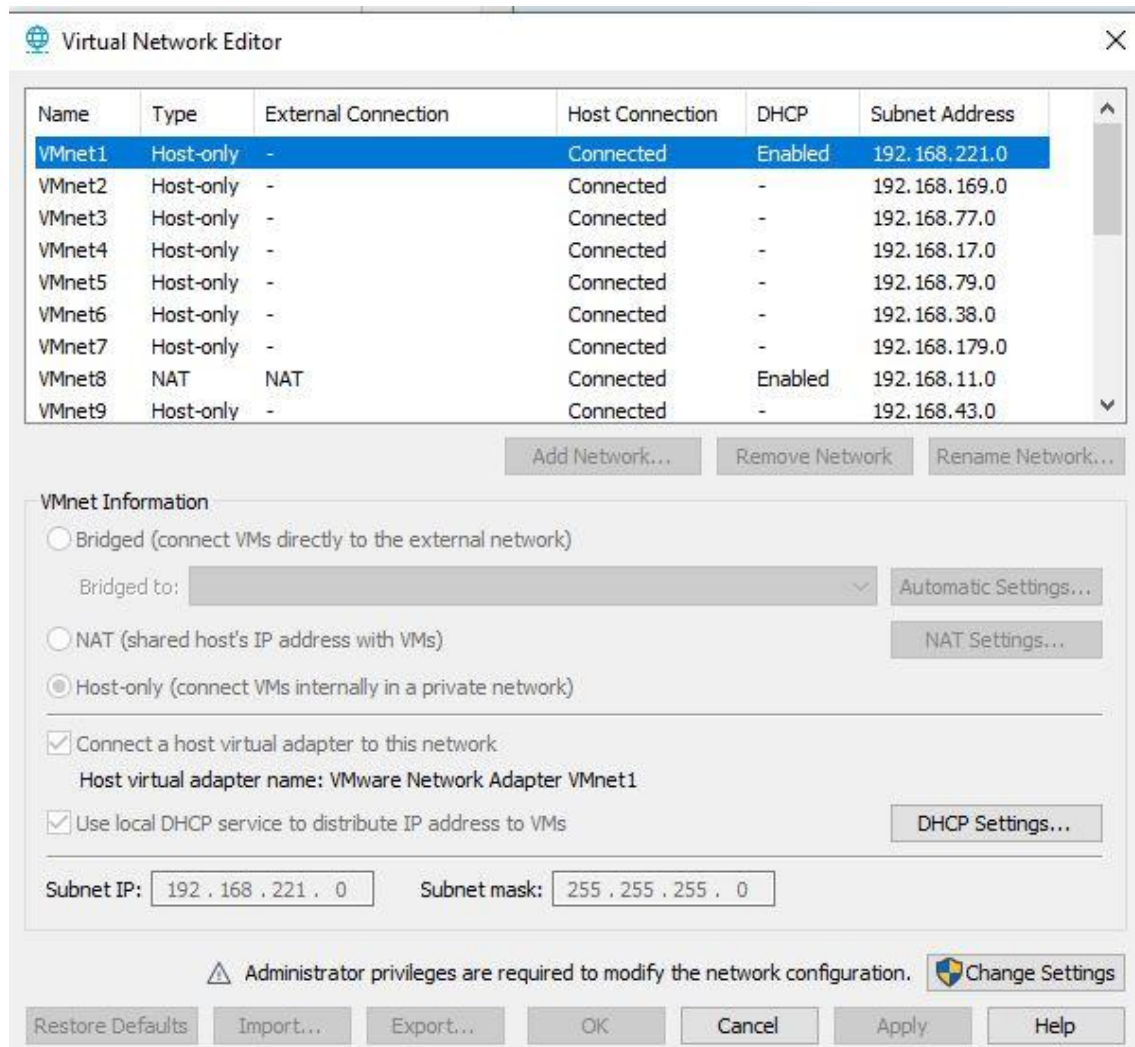
### 7.2.1 VMware Workstation asennus ja konfigurointi

VMware Workstation Pro sovellus ladataan VMware.com viralliselta verkkosivustolta ja suoritetaan ladattu asennuspaketti. Asennus on hyvin suoraviivainen ja sovellus avautuu asennuksen jälkeen Kuvion 9 mukaisesti.



Kuvio 9 VMware Workstation Pro

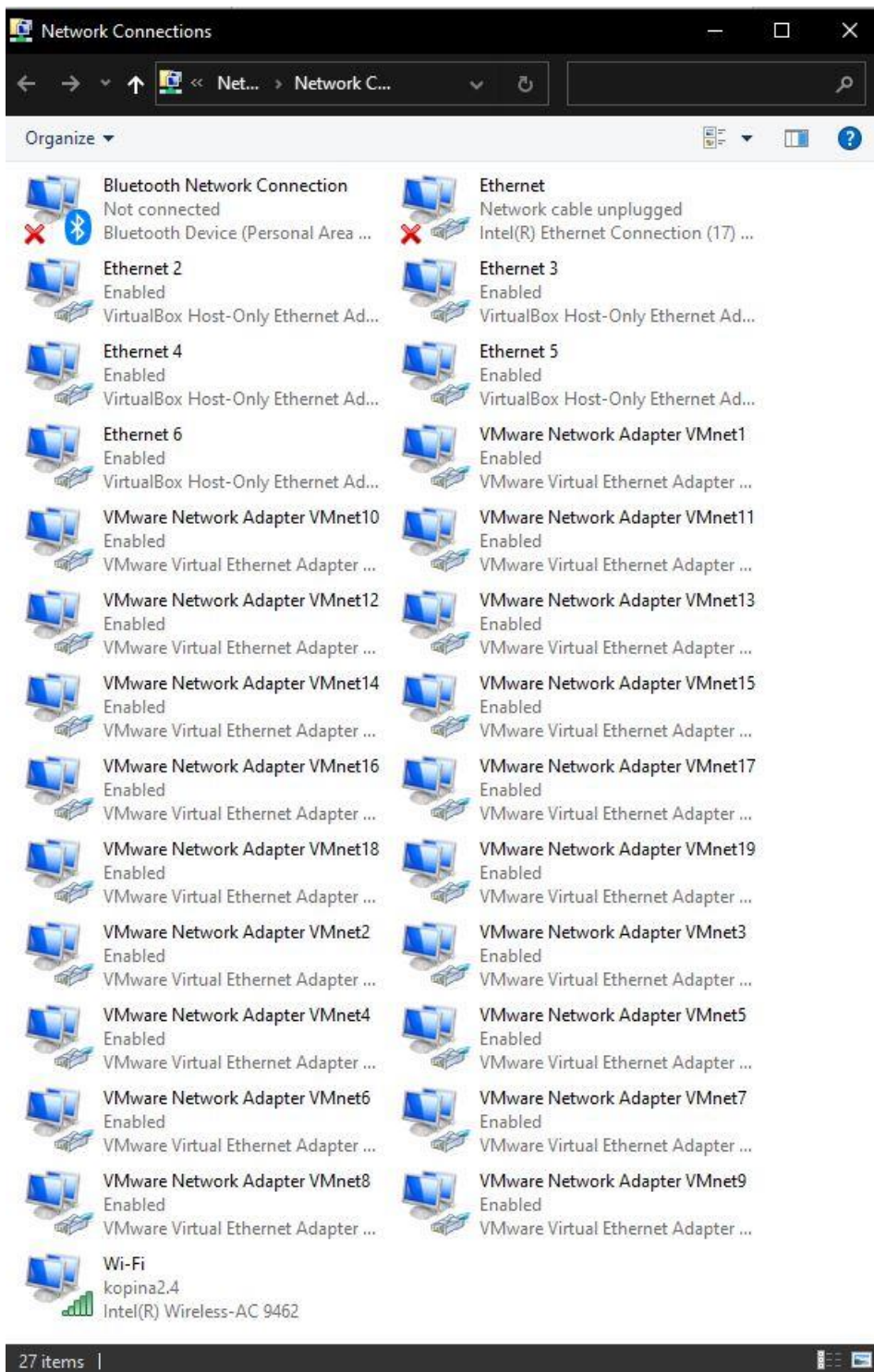
Verkkolaitteet tarvitsevat käyttöönotossa verkkoliitännöjä, joten niitä pitää lisätä sovelluksesta. VMware Workstation Pro sovelluksen valikosta valitaan Edit ja painetaan Virtual Network Editor painiketta.



Kuvio 10 Virtual Network Editor

Jokaisella reitittimellä on käyttöönotossa vähintään kolme yhteyttä muihin laitteisiin, joten tarvitaan vähintään yhdeksän uutta verkkoliitännää. Verkkoliitännän lisääminen käy Kuviossa 10

näkyvästä Add Network painikkeesta. Virtual Network Editor ikkunan adapterit näkyvät myös Windows-käyttöjärjestelmän Network Connections ikkunassa.



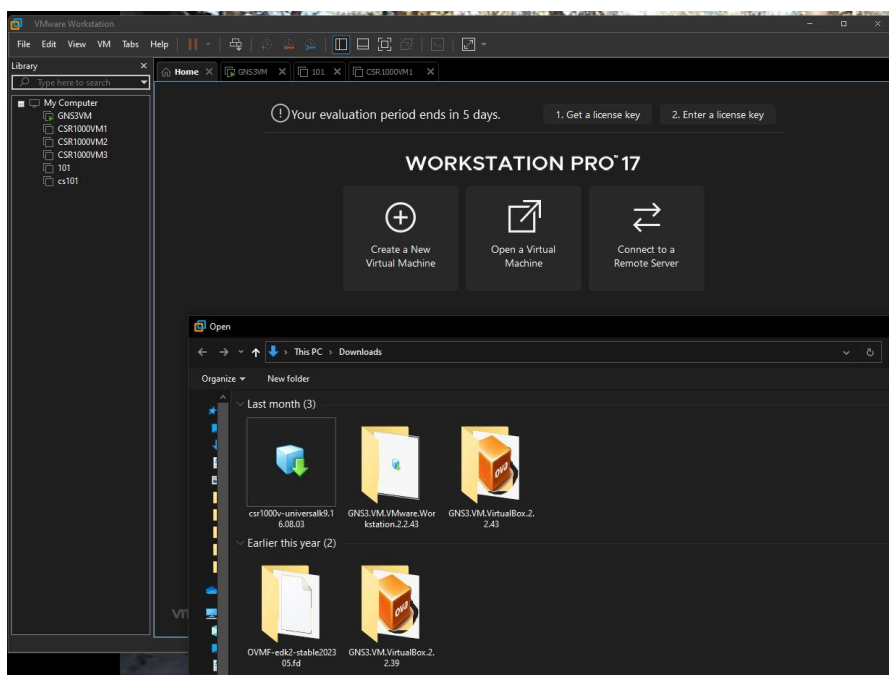
Kuvio 11 Windows Network Connections

## 7.3 Cisco CSR1000V

Cisco CSR1000V on virtuaalinen pilvipalveluihin suunniteltu reititin. Laite on suunniteltu ja sopii ominaisuuksiltaan reunareitittimeksi virtuaalisiin- ja pilviympäristöihin. Reitittimellä on mahdollista luoda erilaisia VPN-tunneleita ja ottaa käyttöön monia dynaamisia reititysprotokollia. Ominaisuuksiin kuuluu myös DHCP-, DNS-, NAT- ja VXLAN-tekniikat, joita voidaan tarvita isoissakin ympäristöissä.

### 7.3.1 CSR1000V käyttöönotto ja esikonfigurointi

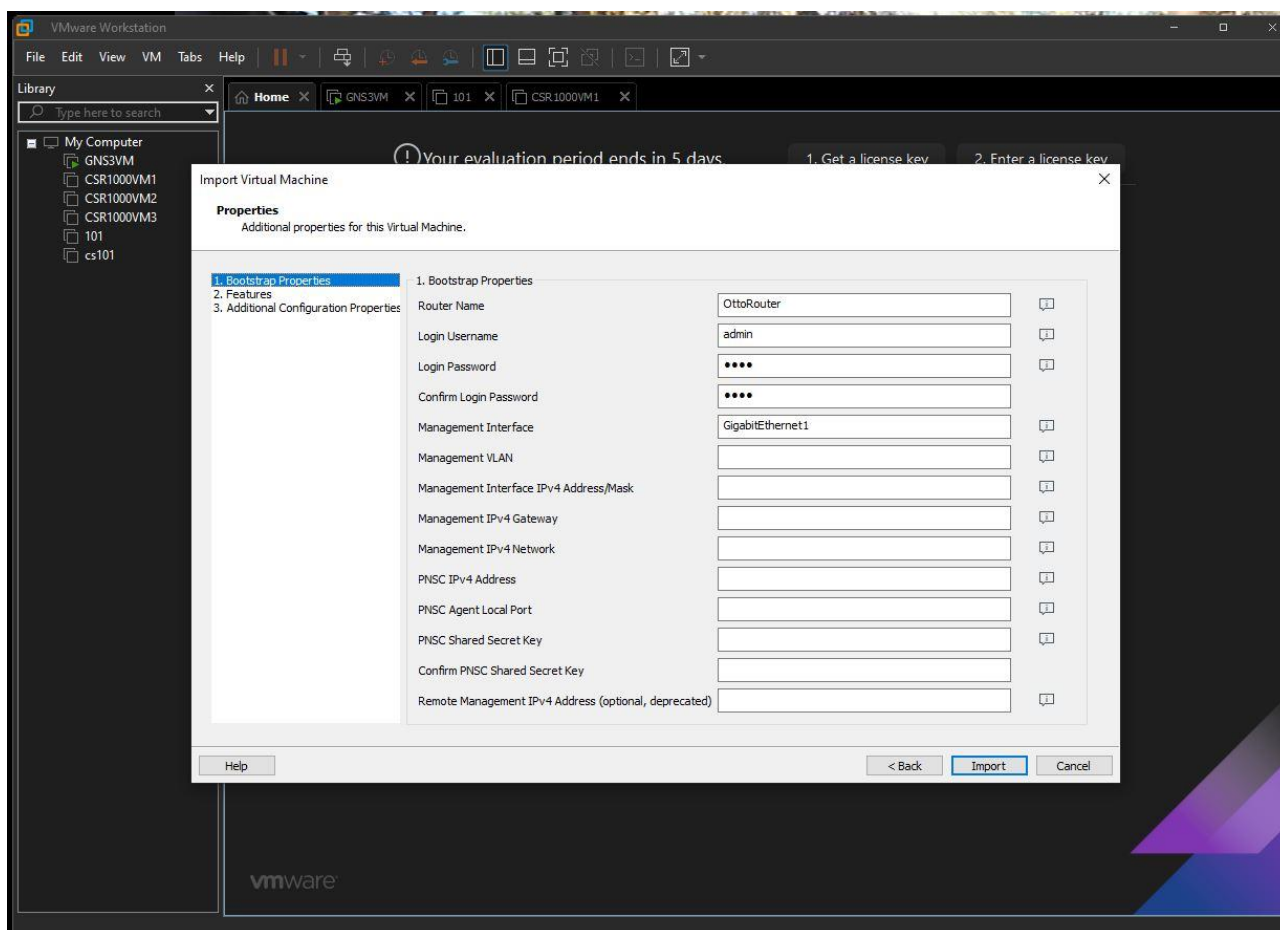
CSR1000v jakelupaketti ladataan Cisco-laitevalmistajan virallisilta tukisivustoilta. Tämä vaatii käyttäjätunnuksen sivustolle ja oikeudet ladata jakelupaketin. Käyttöönotossa käytetään csr1000v-universalk9.16.08.03.ova nimistä jakelupakettia, joka todettiin toimivaksi monien jakelupakettien testauksen takia.



Kuvio 12 CSR1000V tuonti VMware virtuaalisovellukseen

Jakelupaketti universalk9.16.08.03.ova tuodaan VMware virtualisointisovellukseen painamalla "Create a Virtual Machine" painiketta, jonka jälkeen jakelupaketti haetaan sen tallennetusta hake- mista kuvion 12 mukaisesti.





Kuvio 13 CSR1000V-virtuaalikoneen määrytykset

Jakelupaketin valitsemisen jälkeen avautuu ikkuna, josta on mahdollista määrittää asetuksia reitittimelle. Ensimmäisessä valikossa määritetään virtuaalikoneelle haluttu resurssimäärä. Käyttöön-otossa reitittimelle riittää pienet resurssit eli yksi prosessori ja neljä gigatavua muistia. Kuvion 13 mukaisesti Virtuaalikoneelle valitaan Asetetaan nimi, käyttäjätunnus ja salasana. Asetusten määrittämisen jälkeen painetaan import-painiketta ja virtuaalireititin käynnistyy.

Virtuaalikoneen käynnistymisen jälkeen reitittimelle päästään kirjautumaan. Jakelupaketissa pitäisi olla valmiiksi määrytyksiä virtuaalikoneelle. Tarkistetaan, että reititin on saanut valmiiksi jakelupaketin määrittämät verkkokortit käyttöönsä. Syötetään komennot:

```
>enable
```

Enable komento antaa oikeudet etuoikeutettuun tilaan. Etuoikeutetun tilan näkee komentokehotteen kirjoituskentän alkavan reitittimen nimen jälkeisestä risuaidasta.



```
#show platform software vnic-if interface-mapping
```

```
#show ip interface brief
```

Komennoilla saadaan näkyviin tietoja reitittimellä olevista verkkokorteista ja mitä liityntää mikäkin verkkokortti käyttää kuvion 14 mukaisesti. MAC-Osoitteen avulla on mahdollista paikantaa verkkokortti isäntäkoneelta, jos esimerkiksi tarvitsee toteuttaa määrittämiä.

```

rooter#show platform software vnic-if interface-mapping
-----
Interface Name      Driver Name      Mac Addr
-----
GigabitEthernet3    net_vmxnet3      000c.2972.c6f8
GigabitEthernet2    net_vmxnet3      000c.2972.c6ee
GigabitEthernet1    net_vmxnet3      000c.2972.c6e4
-----

rooter#show ip interface brief
Interface      IP-Address      OK? Method Status Protocol
GigabitEthernet1  unassigned      YES unset  administratively down down
GigabitEthernet2  unassigned      YES unset  administratively down down
GigabitEthernet3  unassigned      YES unset  administratively down down
rooter#

```

Kuvio 14 CSR1000V liitännä komentoja

```
(config)# platform console serial
```

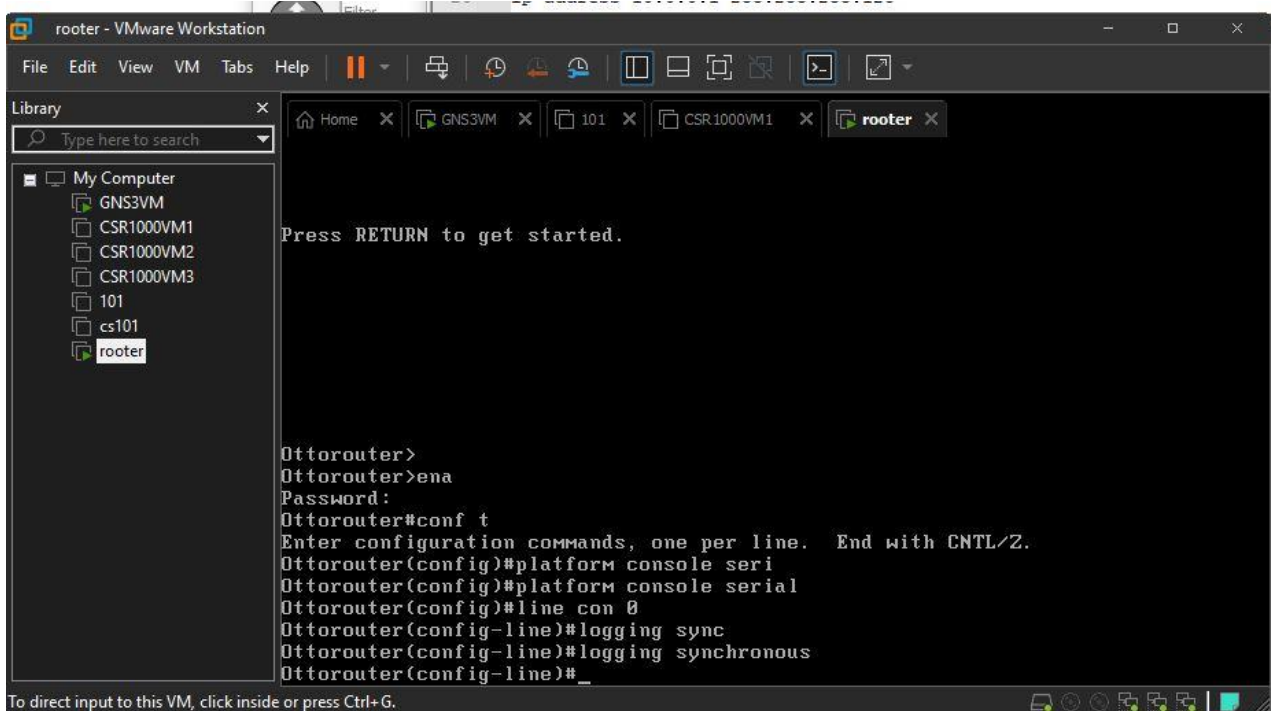
*Platform console serial* komento asettaa Cisco käyttöjärjestelmän komentokehottiin ulostuloksi serial-yhteyden. Reititin osaa myös havaita ja yhdistää ulostulon automaattisesti.

```
(config)#line console 0
```

```
(config-line)#logging synchronous
```

*Line console 0* komennolla päästään konsoliliitännän konfigurointi tilaan, joka voidaan todeta *config-line* etuliitteestä. Käyttöjärjestelmän ulostulo konfiguroidaan synkronoiduksi komennolla *logging synchronous*, joten reititin suodattaa osan lokikirjaan tulevista viesteistä pois kuvion 16

mukaisesti. Tällöin lokikirjasta mm. Debug-viestit eivät tulostu käyttöjärjestelmän komentokehoteelle, joten konfigurointi on selkeämpää.



Kuvio 15 Komentokehotteen ulostulon konfigurointi

### *#write memory*

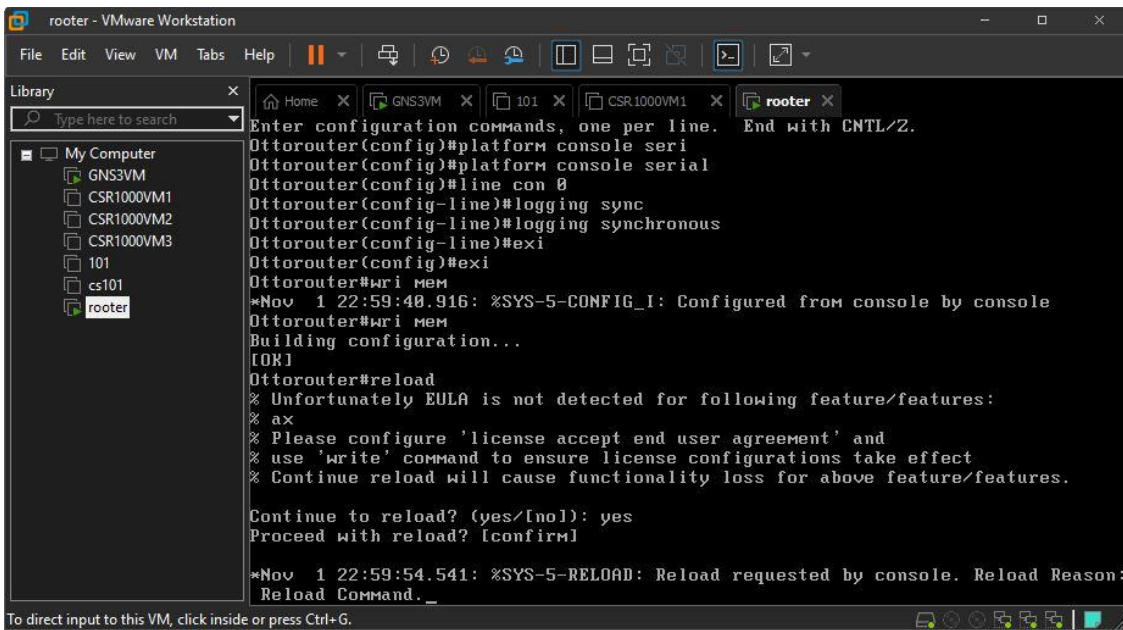
Reitittimellä käytössä oleva konfiguraatio tallennetaan komennolla *write memory* käynnistyskonfiguraatioon. Ilman *write memory* komentoa tehdyt konfiguraatiot katoaisivat reitittimen uudelleenkäynnistämisen yhteydessä.

### *#reload*

### *#yes*

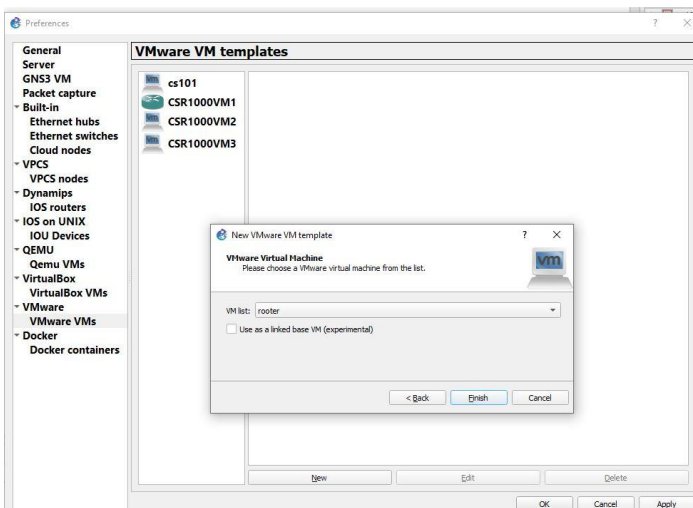
Reititin käynnistetään uudelleen komennolla *reload*. Kuviossa 17 nähdään reitittimen varmistavan vielä käyttäjän olevan varma uudelleenkäynnistämisestä. Reititin sammutetaan

uudelleenkäynnistämisen jälkeen suoraan VMware sovelluksesta valitsemalla virtuaalikone aktiiviseksi ja valitsemalla virtapainikkeesta ”Power off”.



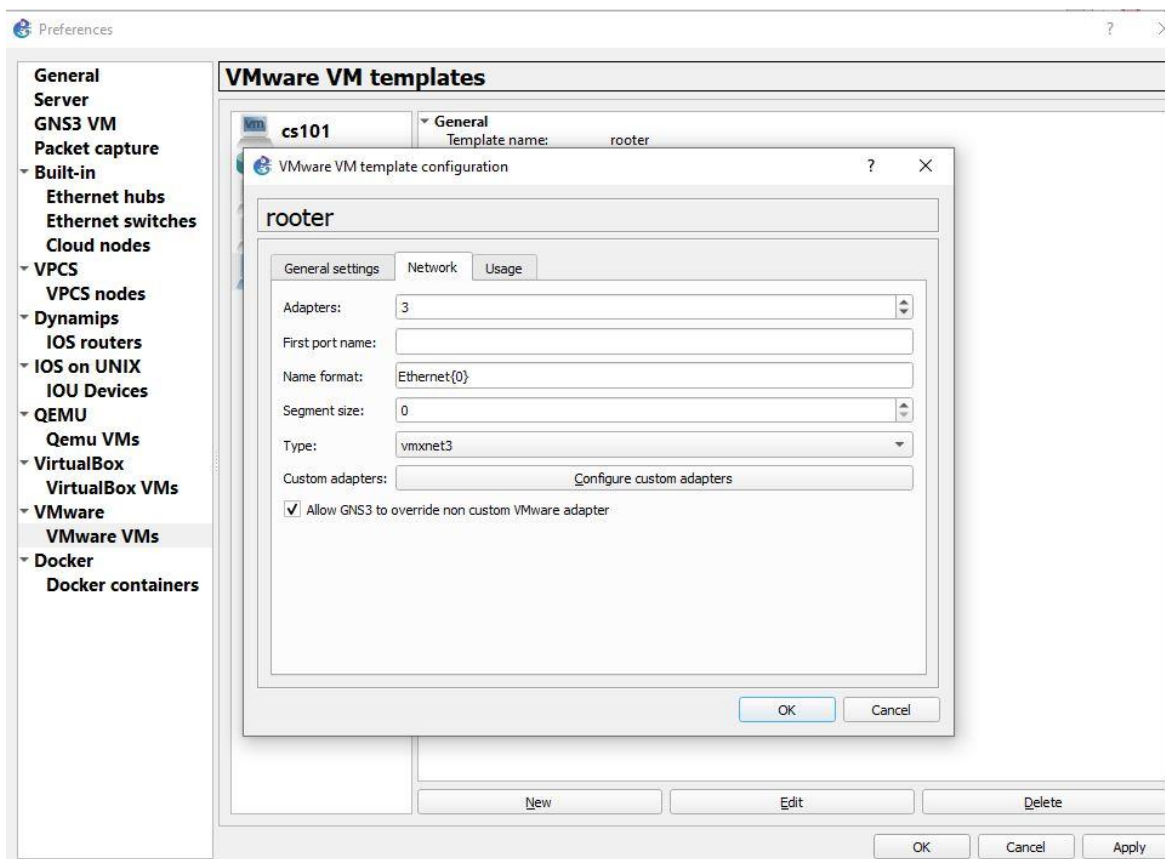
Kuvio 16 CSR1000V uudelleenkäynnistäminen

Seuraavassa vaiheessa siirrytään viemään CSR1000v-virtuaalireititin GNS3 verkkosimulaattorisovellukseen. Sovelluksen ylävalikosta avataan asetukset painikkeilla Edit -> Preferences. Valitaan VMware valikko ja valitaan Add-painike, jolloin avautuu kuvion 18 mukainen virtuaalikonelistaus. Listassa näkyy VMware-sovelluksessa olevat virtuaalikoneet. Valitaan nimetty CSR1000V-virtuaalireititin.



### Kuvio 17 GNS3 VMware virtuaalikoneen lisäys

CSR1000V-virtuaalikoneen lisäyksen jälkeen muutetaan vielä asetuksia valitsemalla edit painike. Avautuu asetussivu, josta valitaan laatikko *Start VM in headless mode* ja muutetaan reitittimen kategoria *Routers* ryhmään. Siirrytään kuviossa 19 näkyvään *Network* välilehteen ja valitaan adapterien määräksi kolme ja asetetaan niiden tyyppiä vmxnet3. Tärkeää on valita aktiiviseksi *Allow GNS3 to override non custom VMware adapter*. Tämän avulla GNS3 voi lisätä serial liitännän asetuksiin VMware virtuaalikoneen määrittäisiin.

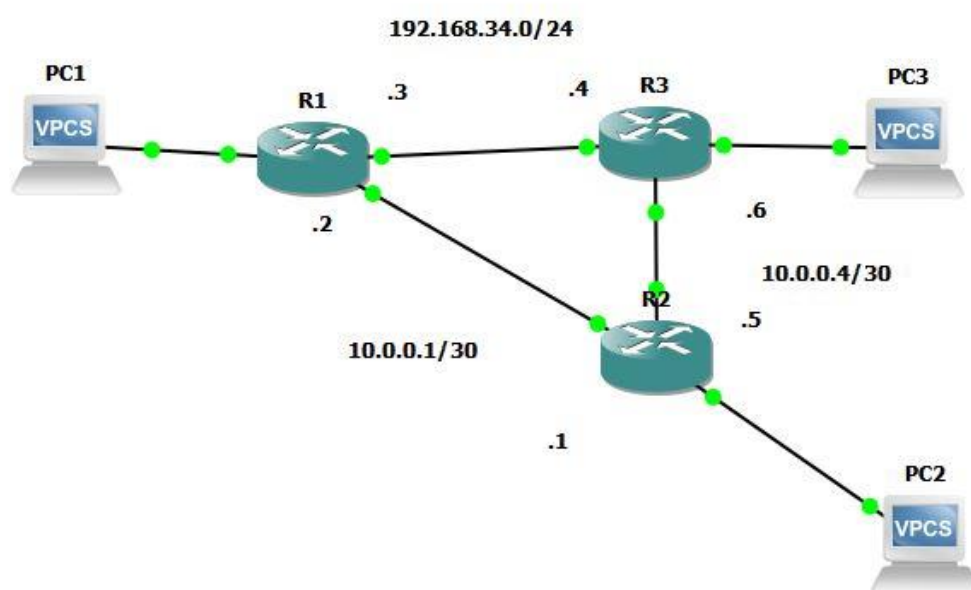


### Kuvio 18 Asetuksien määrittäminen

Virtuaalireitin on nyt valmis käytettäväksi GNS3 verkkosimulaattorissa. Laite sijaitsee ylimmässä *Routers* valikossa ja on valmis tuotavaksi jo valmiiseen tai uuteen projektiin.

## 7.4 Topologia ja konfigurointi

Projektin luominen tapahtuu valitsemalla yläkulmasta *File* ja *New blank project*. Projektin nimetään ja luodaan. Laitteita voidaan tuoda GNS3 simulaattorin vasemmasta reunasta laitevalikosta käyttööliittymään valitsemalla laitteen ja raahaamalla.



Kuvio 19 Käyttöönoton topologia

Käyttöönotossa käytetään kolmea CSR1000V-virtuaalireititintä, jotka ovat liitetty toisiinsa muodostaen silmukan. Reitittimien ovat kytketty kolmella linkillä ja näillä linkeillä on kuviossa kaksikymmentä esitetyt IP-osoitealueet. Jokaiseen reitittimeen on kytketty GNS3 simulaattorin tarjoaman VPCS virtuaalitietokone, joiden avulla voidaan lähettää ICMP, TCP ja UDP viestejä verkossa. Reitittimen ja virtuaalitietokoneen yhdistävään linkkiin reitittimen puolelle konfiguroidaan VTEP-raja-pinta. Virtuaalitietokoneet ovat myös omassa sisäverkossaan, jossa käytetään omaa IP-osoitealuetta.

Taulukko 5 Käyttöönoton IP-osoitteet ja -alueet

Kohde	IP-osoitealue	Käytettävät IP-osoitteet	Yleisjakelu IP-osoite
R1(Gi1)-R2(Gi1)	10.0.0.0/30	R1 10.0.0.2 R2 10.0.0.2	10.0.0.3
R2(Gi2)-R3(Gi1)	10.0.0.4/30	R2 10.0.0.5 R3 10.0.0.5	10.0.0.7
R1(Gi2)-R3(Gi2)	192.168.34.0/24	R1 192.168.34.3 R3 192.168.34.4	192.168.34.255
VPCS LAN	192.168.2.0/24	192.168.2.1-.254	192.168.2.255

Käyttöönotto aloitetaan konfiguroimalla reitittimien välisiin liitäntöihin IP-osoitteet ja alueet taulukon mukaisesti. Reitittimien portit yksi ja kaksi ovat tarkoitettu yhdistettävän muihin reitittämiin ja kolmas portti kytketään virtuaalitietokoneisiin.

Konfiguroidaan reitittimeltä R1 liitäntä, joka johtaa reitittimelle R2.

*R1>enable*

*R1#configure terminal* – komennolla siirrytään konfigurointi tilaan.

*R1(config)#interface gigabitEthernet 1* – komento valitsee liitännän.

*R1(config-if)#ip address 10.0.0.2 255.255.255.252* – komento asettaa IP-osoitteen ja alueen.

*R1(config-if)#no shutdown* - komento siirtää liitännän reitittimen puolelta toiminnalliseen tilaan.

*R1#write memory* – komento tallentaa konfiguraation

Konfiguroidaan reitittimeltä R1 liitäntä, joka johtaa reitittimelle R3

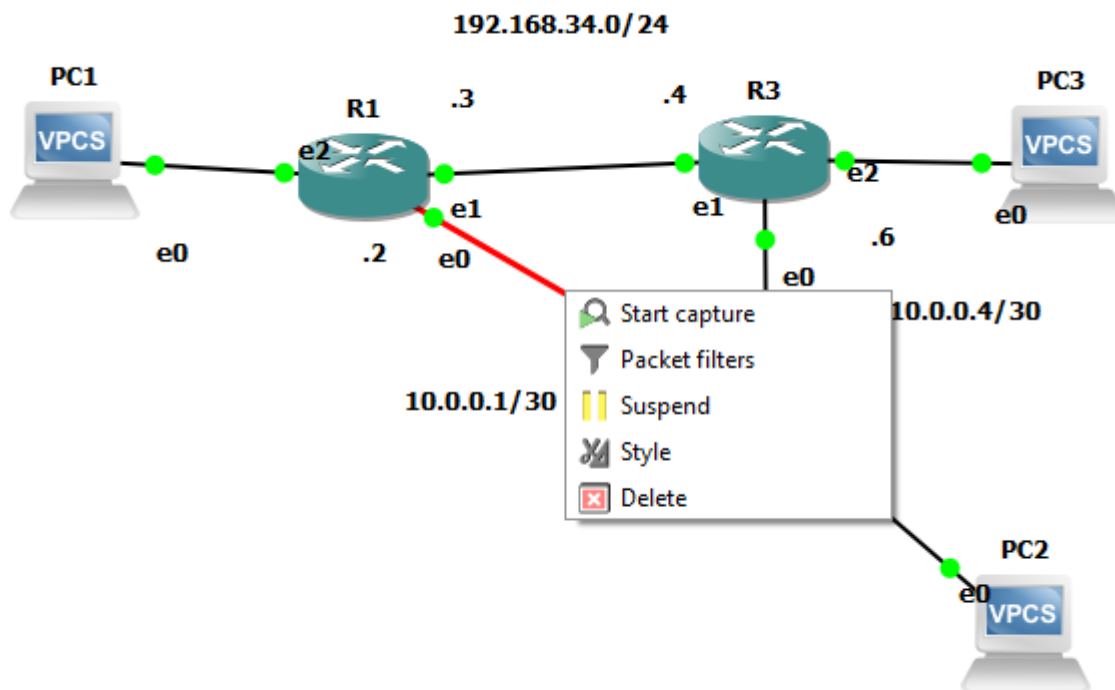
*R1(config)#interface gigabitEthernet 2*

*R1(config-if)#ip address 192.168.34.3 255.255.255.0*

```
R1(config-if)#no shutdown
```

```
R1#write memory
```

Konfiguroidaan myös R2 ja R3 reitittimien liitännät omilla IP-osoitteillaan. Tarkastellaan liikennettä Wireshark-paketti analysaattorin avulla painamalla reitittimien välisestä liitännästä *start capture* painiketta kuvion 21 mukaan.



Kuvio 20 Pakettianalysoijan käynnistys

Varmistetaan reitittimeltä R1 *ping* komennon avulla yhteyden toimivuus.

```
R1#ping 10.0.0.1 – komento ICMP kyselyn lähettämiseen R2 reitittimelle
```

Reititin tulostaa kyselyn onnistuttua seuraavan viestin.

```
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
```

```
!!!!
```

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/3/5 ms

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.2	10.0.0.1	ICMP	114	Echo (ping) request id=0x0009, seq=0/0, ttl=255 (reply in 2)
2	0.001523	10.0.0.1	10.0.0.2	ICMP	114	Echo (ping) reply id=0x0009, seq=0/0, ttl=255 (request in 1)
3	0.012147	10.0.0.2	10.0.0.1	ICMP	114	Echo (ping) request id=0x0009, seq=1/256, ttl=255 (reply in 4)
4	0.013495	10.0.0.1	10.0.0.2	ICMP	114	Echo (ping) reply id=0x0009, seq=1/256, ttl=255 (request in 3)
5	0.028486	10.0.0.2	10.0.0.1	ICMP	114	Echo (ping) request id=0x0009, seq=2/512, ttl=255 (reply in 6)
6	0.029521	10.0.0.1	10.0.0.2	ICMP	114	Echo (ping) reply id=0x0009, seq=2/512, ttl=255 (request in 5)
7	0.044479	10.0.0.2	10.0.0.1	ICMP	114	Echo (ping) request id=0x0009, seq=3/768, ttl=255 (reply in 8)
8	0.046091	10.0.0.1	10.0.0.2	ICMP	114	Echo (ping) reply id=0x0009, seq=3/768, ttl=255 (request in 7)
9	0.060289	10.0.0.2	10.0.0.1	ICMP	114	Echo (ping) request id=0x0009, seq=4/1024, ttl=255 (reply in 10)
10	0.061345	10.0.0.1	10.0.0.2	ICMP	114	Echo (ping) reply id=0x0009, seq=4/1024, ttl=255 (request in 9)

<p>&gt; Frame 1: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface ~, id 0</p> <p>&gt; Ethernet II, Src: VMware_a7:20:e7 (00:0c:29:a7:20:e7), Dst: VMware_75:bd:21 (00:0c:29:75:bd:21)</p> <p>&gt; Internet Protocol Version 4, Src: 10.0.0.2, Dst: 10.0.0.1</p> <p>0100 .... = Version: 4</p> <p>.... 0101 = Header Length: 20 bytes (5)</p> <p>&gt; Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)</p> <p>Total Length: 100</p> <p>Identification: 0x002d (45)</p> <p>0000 .... = Flags: 0x0</p> <p>...0 0000 0000 0000 = Fragment Offset: 0</p> <p>Time to Live: 255</p> <p>Protocol: ICMP (1)</p> <p>Header Checksum: 0xa769 [validation disabled]</p> <p>[Header checksum status: Unverified]</p> <p>Source Address: 10.0.0.2</p> <p>Destination Address: 10.0.0.1</p> <p>&gt; Internet Control Message Protocol</p>	<pre> 0000 00 0c 29 75 bd 21 00 0c 29 a7 20 e7 08 00 45 00  ..ju1... 0010 00 64 00 2d 00 00 ff 01 a7 69 0a 00 00 02 0a 00  ..d..... 0020 00 01 08 00 2c 56 00 09 00 00 00 00 00 00 00 e1  ....V... 0030 51 0a ab cd ab cd ab cd ab cd ab cd ab cd ab cd  Q..... 0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd  .... 0050 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd  .... 0060 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd  .... 0070 ab cd .. </pre>
---	---

## Kuvio 21 ICMP kysely paketti analyysaattorissa

Pakettianalyysaattorin avulla huomataan R1 reitittimen lähettäneen viisi ICMP kyselyä ja R2 vastaan-  
neen niihin onnistuneesti, kuten reitittimen *ping* komennon tulosteessakin lukee. Reitittimen tun-  
temat reitit saa tulostettua reitittimellä syöttäen komennon *show ip route*. Huomataan, että R2 ja  
R3 osoitteet näkyvät R1 reittitaulussa.

10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks

C 10.0.0.0/30 is directly connected, GigabitEthernet1

L 10.0.0.2/32 is directly connected, GigabitEthernet1

192.168.34.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.34.0/24 is directly connected, GigabitEthernet2

L 192.168.34.3/32 is directly connected, GigabitEthernet2

Reittitaulusta huomataan, ettei R1 reitittimellä ole reittiä R2 ja R3 reitittimien väliseen 10.0.0.4/30  
verkkoon. Konfiguroidaan OSPF reititysprotokolla reitittimien välille, jonka avulla pääsy myös R2 ja  
R3 väliseen verkkoon saadaan. OSPF toimii samalla VXLAN-tekniikan alaverkkona eli kuljettaa  
VXLAN liikenteen.



OSPF-verkot ovat käyttöönnotossa yleislähetysverkkoja, joten reitittimet äänestävät DR ja BDR reitittimet keskenään. Reitittimille konfiguroidaan virtuaalinen liitäntä omalla IP-osoitteellaan, joka määrittää äänestysprosessissa suurimman IP-osoitteen DR-tilaan. Virtuaalisen liitännän luomisen avulla mahdollistetaan reititin luomaan tunneleita VXLAN konfiguraatiossa. Konfiguroidaan R1 reititin aloittamalla virtuaalisen liitännän luomisella.

```
R1#configure terminal
```

```
R1(config)#interface loopback 0 – komento luo virtuaalisen liitännän
```

```
R1(config-if)#ip address 4.4.4.4 255.255.255.255 – komento asettaa IP-osoitteen ja alueen
```

```
router ospf 1 – Komento ottaa OSPF protokollan käyttöön ja siirtyy router ospf tilaan
```

```
Network "ip-osoite" area 0 komennolla määritetään verkkoalueen wildcardin avulla, area 0 kertoo OSPF alueen olevan nolla.
```

Reitittimille konfiguroidaan verkot, jotka ovat liitännöissä. Muiden reitittimien verkot mainostuvat OSPF-tekniikan avulla.

```
network 4.4.4.4 0.0.0.0 area 0
```

```
network 10.0.0.0 0.0.0.3 area 0
```

```
network 192.168.34.0 0.0.0.255 area 0
```

R2 Konfiguraatio

```
R2(config)#interface loopback 0
```

```
R2(config-if)#ip address 2.2.2.2 255.255.255.255
```

```
router ospf 1
```

```
network 2.2.2.2 0.0.0.0 area 0
```

```
network 10.0.0.0 0.0.0.3 area 0
```

```
network 10.0.0.4 0.0.0.3 area 0
```

R3 Konfiguraatio

```
R3(config)#interface loopback 0
```

```
R3(config-if)#ip address 6.6.6.6 255.255.255.255
```

```
router ospf 1
```

```
network 6.6.6.6 0.0.0.0 area 0
```

```
network 10.0.0.4 0.0.0.3 area 0
```

```
network 192.168.34.0 0.0.0.255 area 0
```

Tarkastellaan pakettianalysaattorin kanssa OSPF paketteja R1 ja R2 reitittimien välillä. Suoritetaan myös komento *clear ip ospf process* komento reitittimelle R2. Komento tyhjentää reitittaulun ei staattisista ja liitetystä reiteistä ja nollaa OSPF prosessin. Kuviossa 22 R2 reitittimen OSPF prosessi on käynnistynyt ja reititin lähettää Hello-paketin AllSPFRouters 224.0.0.5 multicast osoitteeseen, jonka avulla se pyrkii löytämään naapuruussuhteita. Huomataan Hello-paketin sisällöstä, että Designated router ja Backup Designated Router arvot ovat 0.0.0.0. Seuraavalla rivillä R1 reititin vastaa Hello paketilla ja reitittimet aloittavat tietokantojen synkronoinnin vaihtamalla Database Description paketteja ja tätä tilaa kutsutaan Database Exchange Process nimikkeellä. Exchange prosessista siirrytään Loading tilaan, jossa reitittimet pyytävät vielä uudempia tietoja linkkien ti-  
loista Link State Request ja Link State Update pakettien avulla. Full tila saavutetaan LSACK viestien jälkeen, kun reitittimien tiedot ovat ajan tasalla.

3187	14389.849776	10.0.0.1	224.0.0.5	OSPF	122	LS Update
3188	14389.883986	10.0.0.1	224.0.0.5	OSPF	110	Hello Packet
3189	14389.890630	10.0.0.2	10.0.0.1	OSPF	114	Hello Packet
3190	14389.891473	10.0.0.1	10.0.0.2	OSPF	78	DB Description
3191	14389.891517	10.0.0.1	10.0.0.2	OSPF	114	Hello Packet
3192	14389.906194	10.0.0.2	10.0.0.1	OSPF	78	DB Description
3193	14389.906919	10.0.0.1	10.0.0.2	OSPF	198	DB Description
3194	14389.921746	10.0.0.2	10.0.0.1	OSPF	78	DB Description
3195	14389.922827	10.0.0.1	10.0.0.2	OSPF	78	DB Description
3196	14389.924297	10.0.0.2	10.0.0.1	OSPF	70	LS Request
3197	14389.924830	10.0.0.1	10.0.0.2	OSPF	122	LS Update
3198	14389.939356	10.0.0.1	224.0.0.5	OSPF	122	LS Update
3199	14389.968243	10.0.0.2	224.0.0.5	OSPF	154	LS Update
3200	14392.370612	10.0.0.2	224.0.0.5	OSPF	78	LS Acknowledge
3201	14392.469724	10.0.0.1	224.0.0.5	OSPF	98	LS Acknowledge
3202	14393.842405	10.0.0.2	224.0.0.5	OSPF	114	Hello Packet
3203	14393.896213	10.0.0.1	224.0.0.5	OSPF	122	LS Update
3204	14394.001675	10.0.0.1	224.0.0.5	OSPF	154	LS Update
3205	14394.012735	10.0.0.2	224.0.0.5	OSPF	154	LS Update
3206	14395.992224	10.0.0.1	224.0.0.5	OSPF	122	LS Update
3207	14396.415450	10.0.0.2	224.0.0.5	OSPF	98	LS Acknowledge
3208	14396.514222	10.0.0.1	224.0.0.5	OSPF	98	LS Acknowledge
3209	14398.899709	10.0.0.1	224.0.0.5	OSPF	114	Hello Packet

```

Protocol: OSPF IGP (89)
Header Checksum: 0xbaac [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.0.0.1
Destination Address: 224.0.0.5
Open Shortest Path First
  OSPF Header
    Version: 2
    Message Type: Hello Packet (1)
    Packet Length: 44
    Source OSPF Router: 2.2.2.2
    Area ID: 0.0.0.0 (Backbone)
    Checksum: 0xe79e [correct]
    Auth Type: Null (0)
    Auth Data (none): 0000000000000000
  OSPF Hello Packet
    Network Mask: 255.255.255.252
    Hello Interval [sec]: 10
    Options: 0x12, (L) LLS Data block, (E) External Routing
    Router Priority: 1
    Router Dead Interval [sec]: 40
    Designated Router: 0.0.0.0
    Backup Designated Router: 0.0.0.0
  OSPF LLS Data Block

```

Kuvio 22 OSPF pakettianalysaattorissa

Syötetään reitittimelle R1 komento *show ip route*, joka tulostaa reittitaulun.

```

2.0.0.0/32 is subnetted, 1 subnets
O   2.2.2.2 [110/2] via 10.0.0.1, 00:16:06, GigabitEthernet1
    4.0.0.0/32 is subnetted, 1 subnets
C   4.4.4.4 is directly connected, Loopback0
    6.0.0.0/32 is subnetted, 1 subnets
O   6.6.6.6 [110/2] via 192.168.34.4, 00:16:04, GigabitEthernet2
    10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C   10.0.0.0/30 is directly connected, GigabitEthernet1
L   10.0.0.2/32 is directly connected, GigabitEthernet1
O   10.0.0.4/30 [110/2] via 192.168.34.4, 00:16:04, GigabitEthernet2

```

Reittitaulua tarkastellessa huomataan, että OSPF-reititysprotokollan avulla R1 reitittimellä on nyt reitit R2 ja R3 reitittimien väliseen verkkoon ja näiden reitittimien virtuaalisiin rajapintoihin.

Varmistetaan vielä pääsy 10.0.0.4/30 verkkoon.

```

R1#ping 10.0.0.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.6, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/4/7 ms

```

Aloitetaan VXLAN konfigurointi reitittimestä R1. Konfiguroidaan ensimmäisenä Multicast käyttöön. VXLAN VTEPit eli virtuaaliset päätepisteet käyttävät Multicastia löytääkseen toisensa.

*R1(config)#ip multicast-routing distributed* – komennolla otetaan käyttöön Multicast  
*R1(config)#ip pim rp-address 4.4.4.4* – komennolla asetetaan rendezvous piste käyttöön. Rendezvous pisteinä käyttöönotossa toimii R1 reititin.

*R1(config)#int range gi1-2,loopback 0* – valitaan liitännät GigabitEtherne 1, GigabitEthernet 2 ja loopback 0.

*R1(config-if)#ip pim sparse-mode* – komennolla asetetaan Multicast käyttöön liitännään sparse tilassa.

Luodaan virtuaalinen päätepiste.

*R1(config)#interface nve1* – Komento luo network virtualization endpoint liitännän

*R1(config-if)#source-interface Loopback0* – Komento osoittaa aikasemmin tehdyn loopback liitännän nve1 liitännään

*R1(config-if)#member vni 5000 mcast-group 239.1.1.1* – komento luo VNI arvoltaan 5000 joka mainostuu multicast ryhmään 239.1.1.1

*R1(config-if)#no shutdown* komento asettaa liitännän toiminnalliseen tilaan.

MTU eli maximum transmission unit arvo asetetaan 9216. Tämä sallii liitännästä PDU:n arvon olevan enintään 9216 bittiä. VXLAN kehykset ovat huomattavasti suurempia normaaliin verkkoon verrattuna, joten jokaisessa VXLAN verkon liitännässä pitäisi olla määritettynä suurehko MTU arvo.

*R1(config)#interface range interface gigabitEthernet1-3*

*R1(config-if)#mtu 9216*

*R1(config)#interface interface gigabitEthernet3*

Komennoilla luodaan rajapinta bridge domainin ja fyysisen portin välille. Kapselointi tapa asetetaan untagged tilaan eli porttiin on niin sanotussa access tilassa.

*R1(config-if)#service instance 10 ethernet*

*R1(config-if)#encapsulation untagged*

Luodaan bridge domain, joka mahdollistaa liikenteen L3 ja L2 tasojen välillä. Bridge domainissa määritetään VNI arvo ja virtuaalitietokoneelle menevä liitäntä ja service-instance

*R1(config)#bridge-domain 10*

*R1(config-bdomain)# member vni 5000*

*R1(config-bdomain)# member interface gigabitEthernet3 service-instance 10*

Tehty VXLAN konfiguraatio reitittimelle R1 toistetaan reitittimille R2 ja R3. Tarkastellaan luotua virtuaalista päätepistettä VTEP komennolla *R1#show nve interface nve1*.

*Interface: nve1, State: Admin Up, Oper Up, Encapsulation: Vxlan,*

*BGP host reachability: Disable, VxLAN dport: 4789*

*VNI number: L3CP 0 L2CP 0 L2DP 1*

*source-interface: Loopback0 (primary:4.4.4.4 vrf:0)*

Tulosteesta voidaan päätellä VTEP olevan toiminnassa. Kapselointi tila on VXLAN. VTEP saa osoitteensa äskettäin luodusta virtuaalisesta Loopback 0 liitännästä.

Syötetään komento *show nve vni*.

R1#show nve vni

*Interface VNI Multicast-group VNI state Mode BD cfg vrf*

*nve1 5000 239.1.1.1 Up L2DP 10 CLI N/A*

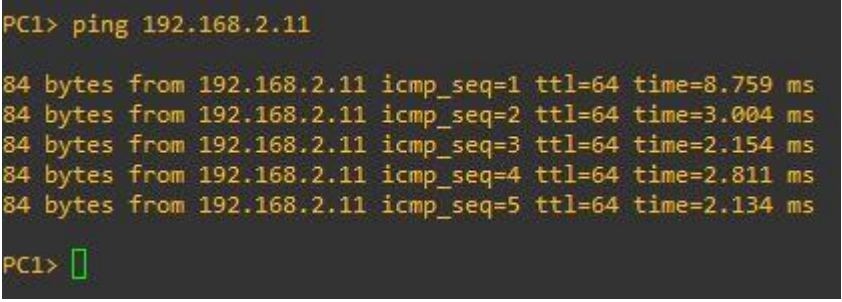
Reititin tulostaa tietoja *nve1* liitännästä ja kertoo VNI olevan 5000, kuten aikaisemmin konfiguroitiin bridge domainiin. Loopback 0 liitäntään konfiguroitu Multicast ryhmä osoitteella *239.1.1.1* näkyy myös tulosteessa.

Tarkastellaan, onko reititin oppinut muita VTEP-liitäntöjä.

R1#sh nve peers

*Interface VNI Type Peer-IP Router-RMAC eVNI state flags UP time*

Tuloste on tyhjä. Tämä johtuu siitä, että Multicast ryhmään VNI arvolla 5000 ei ole lähetetty mitään. Lähetetään virtuaalikoneelta PC1 ICMP kysely koneelle PC3.



```
PC1> ping 192.168.2.11
84 bytes from 192.168.2.11 icmp_seq=1 ttl=64 time=8.759 ms
84 bytes from 192.168.2.11 icmp_seq=2 ttl=64 time=3.004 ms
84 bytes from 192.168.2.11 icmp_seq=3 ttl=64 time=2.154 ms
84 bytes from 192.168.2.11 icmp_seq=4 ttl=64 time=2.811 ms
84 bytes from 192.168.2.11 icmp_seq=5 ttl=64 time=2.134 ms
PC1> 
```

Kuvio 23 PC1 ping PC3 virtuaalikoneelle

Tarkastellaan uudelleen, onko reititin R1 oppinut muita VTEP liitäntöjä.

R1#show nve peers

*Interface VNI Type Peer-IP Router-RMAC eVNI state flags UP time*

*nve1 5000 L2DP 6.6.6.6*

Tarkastellaan tulostetta ja huomataan, että R1 VTEP on oppinut R3 VTEP. Tämä voidaan vahvistaa Peer-IP sarakeessa arvosta 6.6.6.6, joka on R3 Loopback 0 osoite.

```

> Frame 377: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface -, id 0
> Ethernet II, Src: VMware_a7:20:f1 (00:0c:29:a7:20:f1), Dst: IPv4mcast_01:01:01 (01:00:5e:01:01:01)
> Internet Protocol Version 4, Src: 4.4.4.4, Dst: 239.1.1.1
> User Datagram Protocol, Src Port: 31525, Dst Port: 4789
> Virtual eXtensible Local Area Network
> Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Address Resolution Protocol (request)

```

#### Kuvio 24 PC1 lähettämä ARP kysely VXLAN-verkossa

Tarkastellaan kuviossa 24 näkyvää pakettianalysaattorin dataa. Virtuaalikoneen PC1 lähettämä ARP kysely on kaapattu R1 ja R3 välisestä linkistä. Huomataan, että R1 VTEP on kapseloinut Ethernet kehyksen ja rakentanut siitä VXLAN segmentin. Tärkeä huomio on IP-otsikossa näkyvä kohde IP-osoite 239.1.1.1, jonka avulla R1 VTEP saavuttaa R3 VTEP rajapinnan. Samanlainen kuviossa 24 näkyvä viesti on mennyt myös R1 ja R2 välisestä linkistä mutta siihen ei ole tullut vastausta.

```

> Frame 378: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface -, id 0
> Ethernet II, Src: VMware_ab:5c:6a (00:0c:29:ab:5c:6a), Dst: VMware_a7:20:f1 (00:0c:29:a7:20:f1)
> Internet Protocol Version 4, Src: 6.6.6.6, Dst: 4.4.4.4
> User Datagram Protocol, Src Port: 2406, Dst Port: 4789
> Virtual eXtensible Local Area Network
> Ethernet II, Src: Private_66:68:02 (00:50:79:66:68:02), Dst: Private_66:68:00 (00:50:79:66:68:00)
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Private_66:68:02 (00:50:79:66:68:02)
  Sender IP address: 192.168.2.11
  Target MAC address: Private_66:68:00 (00:50:79:66:68:00)
  Target IP address: 192.168.2.10

```

#### Kuvio 25 PC3 virtuaalikoneen lähettämä vastaus

Kuviossa 25 on esitetty PC3 virtuaalikoneen lähettämä vastaus kaapattuna pakettianalysaattorilla R1 ja R3 reitittimien välistä. PC3 on vastannut PC1 virtuaalikoneen pyytämän MAC-Osoitteen. Tärkeä huomio on uudelleen IP-otsikon kohde IP-osoite, joka on R1 reitittimien Loopback 0 osoite.



Kuviot 24, 25 ja 26 todentavat VXLAN-tekniikan toimivuuden käyttöönotossa.

377	761.188761	Private_66:68:00	Broadcast	ARP	114 Who has 192.168.2.11? Tell 192.168.2.10
378	761.230066	Private_66:68:02	Private_66:68:00	ARP	114 192.168.2.11 is at 00:50:79:66:68:02
379	761.232659	192.168.2.10	192.168.2.11	ICMP	148 Echo (ping) request id=0x2609, seq=1/256, ttl=64 (reply in 381)
380	761.239440	192.168.34.4	224.0.0.13	PIMv2	68 Join/Prune
381	761.240194	192.168.2.11	192.168.2.10	ICMP	148 Echo (ping) reply id=0x2609, seq=1/256, ttl=64 (request in 379)
382	762.247602	192.168.2.10	192.168.2.11	ICMP	148 Echo (ping) request id=0x2709, seq=2/512, ttl=64 (reply in 383)
383	762.249304	192.168.2.11	192.168.2.10	ICMP	148 Echo (ping) reply id=0x2709, seq=2/512, ttl=64 (request in 382)
384	763.256620	192.168.2.10	192.168.2.11	ICMP	148 Echo (ping) request id=0x2809, seq=3/768, ttl=64 (reply in 385)
385	763.257620	192.168.2.11	192.168.2.10	ICMP	148 Echo (ping) reply id=0x2809, seq=3/768, ttl=64 (request in 384)
386	764.264138	192.168.2.10	192.168.2.11	ICMP	148 Echo (ping) request id=0x2909, seq=4/1024, ttl=64 (reply in 387)
387	764.265410	192.168.2.11	192.168.2.10	ICMP	148 Echo (ping) reply id=0x2909, seq=4/1024, ttl=64 (request in 386)
388	765.272588	192.168.2.10	192.168.2.11	ICMP	148 Echo (ping) request id=0x2a09, seq=5/1280, ttl=64 (reply in 389)
389	765.273609	192.168.2.11	192.168.2.10	ICMP	148 Echo (ping) reply id=0x2a09, seq=5/1280, ttl=64 (request in 388)

Kuvio 26 Pakettianalysaattorin tuloste PC1 virtuaalikoneen ICMP kyselystä

## 8 Tulokset ja johtopäätökset

Käyttöönotossa saatiin toteutettua toimiva verkkokokonaisuus OSPF- ja VXLAN-tekniikoilla. Verkko saatiin konfiguroitua, niin että virtualisoitujen tietokoneiden kesken saatiin lähetettyä ICMP-kyseilyjä. Todisteena OSI-mallin kerroksen kaksi tasolla tapahtuvasta liikenteestä voidaan tarkastella kuvion 26 riveiltä 377 ja 378. Riveillä nähdään tason kaksi sisäverkossa luonnollisesti esiintyvät ARP-kysely yleislähetysosoitteeseen ja ARP-vastaus kyselyn lähettäjän MAC-osoitteeseen.

Käyttöönoton ensimmäisessä vaiheessa saatiin reitittimet lähettämään ja vastaamaan ICMP-kyseilyihin suoraan liitettyjen verkkoliitännöiden kesken. Ensimmäisessä vaiheessa luotiin kolme virtuaali-reititintä ja liitettiin ne toisiinsa, joten verkko muodosti kolmiomaisen rakenteen. Reitittimien verkkoliitännöihin konfiguroitiin IP-osoitteet, tallennettiin konfiguraatio ja lähetettiin ICMP-kyseilyitä. Kyselyitä ja vastauksia pystyttiin seuraamaan kuviosta 21 ja todentamaan konfiguroinnin onnistuminen.

OSPF-reititysprotokollan konfigurointi aloitettiin virtuaalisen liitännän luomisella, jota päästiin hyödyntämään VXLAN-tekniikan konfiguroimisessa. Jokaisen reitittimen virtuaaliselle liitännälle asetettiin oma IP-osoite. OSPF-naapurisuuden syntymistä tarkasteltiin kuviossa 22 pakettianalysaattorin avulla. OSPF-reititysprotokollan toimivuus todistettiin tulostamalla reittitaulu ja lähettämällä ICMP-kyselyitä verkkoon, joka oli edellisessä vaiheessa saavuttamattomissa.

Käyttöönoton viimeisessä vaiheessa eli VXLAN-tekniikan konfiguroiminen aloitettiin konfiguroimalla Multicast, jonka avulla VTEP-rajapintojen oli mahdollista löytää toisensa. Multicast ryhmä

konfiguroitiin jokaisella VTEP-rajapinnalla olemaan IP-osoitteella 239.1.1.1. Viimeisessä vaiheessa VXLAN-tekniikan toimivuutta tarkastellessa todettiin konfiguraation toimivan halutulla tavalla. Yleislähetyspakettien kohdeosoite oli 239.1.1.1. Tarkastelun kohteena oli myös ICMP-kyselyjen ja -vastausten tarkastelu virtuaalitietokoneiden välillä VXLAN-verkossa. Saatiin todettua, että paketti ja liikenne kulkee VXLAN-verkossa ja on oikein muodostunutta RFC 7348 (2014, 7) standardin mukaisesti.

## 9 Pohdinta

Opinnäytetyössä käyttöönotettiin OSPF- ja VXLAN-tekniikat virtualisoimalla reitittämiä ja rakentamalla niistä verkko GNS3 verkkosimulaattorin avulla. Työssä tarkasteltiin verkkotekniikan perusteita standardien pohjalta. VLAN- ja STP-tekniikoista tarkasteltiin jo todettuja ongelmia ja soveltuvuuksia suurissa verkkoympäristöissä esimerkkien avulla. OSPF- ja VXLAN-tekniikoita tarkasteltiin enimmäkseen virallisten standardien kautta keskittyen tekniikoiden toimivuuteen. Käyttöönotossa dokumentointiin vaiheittain koko prosessi, jotta lukijan olisi se mahdollista toteuttaa.

Käyttöönotto vaati erittäin paljon tutkintaa eri tavoista toteuttaa VXLAN-tekniikkaa virtualisoidussa ympäristössä. GNS3 ohjelmistona vaati opiskelua ja ongelmia tuli vastaan jo asennusvaiheessa. Tarkoituksena oli käyttää GNS3-virtualisointi sovellusta Manjaro Arch Linux-pohjaisen käyttöjärjestelmän päällä. Tästä koitui kuitenkin monenlaisia ongelmia ja lukemattomia "error"-viestejä lukuisista korjausyrityksistä huolimatta. Johtopäätöksenä vaihdettiin käyttöjärjestelmä Windows-10 N versioon.

Virtualisoitujen verkkolaitteiden valinnassa oli haasteita. Ehdotuksena tietoverkkotekniikan ammattilaiselta saatiin vihje Cisco C8000V virtuaalireitittimistä. Ominaisuuksia, joita VXLAN-tekniikka vaatii, ei näillä reitittimillä saatu toimimaan. Myös Hypervisorin valinnassa koitui ongelmia. Käyttöönottoa testattiin todella pitkään Oracle VM Virtualbox Hypervisorilla, mutta ongelmaksi koitui sovelluksen virtuaalisiin verkkoliitäntöihin liittyvät määrytykset ja asetukset. Virtuaaliselle reitittimelle ei ikinä saatu näkyviin verkkoliitäntöjä, kun käytössä oli Hypervisor Virtualbox.

Käyttöönotto saatiin onnistuneesti toteutettua ja liikennettä tarkasteltiin pakettitasolla vaihe vaiheelta konfiguroinnin edetessä. Käyttöönotto toimiva ja se on todistettu käyttöönoton kuvioissa ja teksteissä. Opinnäytetyössä on tarkasteltu paljon standardeista saatua tietoa ja käyttöönotto



toimii niiden mukaisesti. Käyttöönottoa suunnitellessa ja sitä tehdessä pohdittiin, onko käyttöönotto liian pieni tai yksinkertainen. Tultiin kuitenkin johtopäätökseen, että työ on nimenomaan käyttöönotto eli haluttiin saada tekniikat toimimaan standardien mukaisesti ja virtuaalitetokoneet keskustelemaan VXLAN-verkon yli. Käyttöönotto on jatkokehittävissä ja seuraavana verkkoon tullaan tuomaan kytkin ja muutama virtuaalitetokone. Verkkoon on tullaan luomaan lisää virtuaalisia lähiverkkoja omalla VNI-tunnisteella. Multicastia tullaan myös tarkastelemaan, koska Multicast-alue on tällä hetkellä todella laaja.

Käyttöönotto ei ole tietoturvallinen, jotta sitä voitaisiin käyttää muualla kuin sisäisessä virtuaalissa ympäristössä. OSPF-reititysprotokollassa on ominaisuutena autentikointi, jota ei ole otettu käyttöön ja tällä hetkellä reitittimien OSPF-protokolla pyrkii ominaisuuksiltaan etsimään uusia naapureita. VXLAN-tekniikassa ongelmaksi voi koitua Multicast ryhmään liittyminen, joka voisi hyvin olla tietoturvariski tässä toteutuksessa. Multicast ryhmään liittyminen aiheuttaa saavuttavuuden muihin VTEP-rajapintoihin. VXLAN-ympäristössä laitteiden pitäisi olla konfiguroitavissa ja hallittavissa vain järjestelmänvalvojan toimesta.

Laatua pyrittiin parantamaan käyttämällä mallina Cisco CSR1000v laitteen virallisesta VXLAN-konfigurointi dokumentaatiota. Laatu ja luotettavuus on esitetty käyttöönotossa, tuloksissa ja johtopäätöksissä. Esitetyistä tuloksista ja konfiguraatioista on viitattu tekniikoiden virallisiin standardeihin.

Käyttöönoton laatua ja luotettavuutta pyrittiin vahvistamaan myös esittelemällä käyttöönottoa tietoverkkojen parissa työskenteleville henkilöille. Esittelyssä todettiin redundanttisuuden toimivuus lähettämällä ICMP-kyselyjä virtuaalitetokoneilta toisille ja katkaisemalla linkki. Redundanttisuus todettiin toimivaksi ja paketteja ei kadonnut matkalla. Multicastin toimivuuden todentaminen ratkaisusta todettiin olevan haastavaa, koska yleensä todentaminen tapahtuu striimaamista käyttäen. PIM todettiin olevan oikein konfiguroitu Sparse-tilaan. Rendezvous piste ja multicast ryhmät olivat konfiguroitu ja reitittimet näkivät ne onnistuneesti. Kehitysideoita tarkastelijoilta tuli mm. Source-Specific multicastin ja ACL lisäämiseen reitittimien konfiguraatioon.

## 10 Lyhtenteet

ACL	Access Control List
AS	Autonomous system
IGP	Interior Gateway Protocol
EGP	Exterior Gateway Protocol
BGP	Border Gateway Protocol
IEEE	Institute of Electrical and Electronics Engineers
BPDU	Bridge Protocol Data Unit
LSA	Link State Advertisement
LSDB	Link State Database
LSAck	Link State Acknowledgement
LSU	Link State Update
LSR	Link State Request
RID	Router ID

## Lähteet

C, Panek. 2020. Network Fundamentals, Sybex. Viitattu 18.10.2022.

C. Paciello, G. 2016. Spanning Tree Protocol, from a feature CCNA's Perspective, by Gerald C. Paciello. Blogipostaus. Cisco.com. Viitattu 04.09.2023. <https://community.cisco.com/t5/networking-blogs/spanning-tree-protocol-from-a-feature-ccna-s-perspective-by/ba-p/3101592>

Cisco Cloud Services Router 1000V At-a-Glance. N.d. Cisco laitevalmistajan CSR1000V reitittimen dokumentaatio. Viitattu 13.11.2023. <https://www.cisco.com/c/en/us/products/collateral/routers/cloud-services-router-1000v-series/at-a-glance-c45-733686.html>

Cisco Meraki, 2023. Determining the RSTP/STP Root Bridge on an MS Switch Network. Cisco Meraki artikkeli. Viitattu 14.09.2023. [https://documentation.meraki.com/MS/Port\\_and\\_VLAN\\_Configuration/Determining\\_the\\_RSTP%2F%2F%2F%2FSTP\\_Root\\_Bridge\\_on\\_an\\_MS\\_Switch\\_network](https://documentation.meraki.com/MS/Port_and_VLAN_Configuration/Determining_the_RSTP%2F%2F%2F%2FSTP_Root_Bridge_on_an_MS_Switch_network)

Getting Started with GNS3, N.d. GNS3 sovelluksen virallinen ohjeistus. Viitattu 06.09.2023. <https://docs.gns3.com/docs/>

How Switches Work. N.d. Study CCNA verkkosivuston artikkeli. Viitattu 09.09.2023. <https://study-ccna.com/how-switches-work/>

Howard, 2023. VXLAN: the Future for Data Center Networks. Blogipostaus. Fs.com. Viitattu 04.09.2023. <https://community.fs.com/blog/vxlan-the-future-for-data-center-networks.html>

R, Phani. 2015. OSPF: A Network Routing Protocol. Tadimety. Viitattu 31.08.2023.

RFC 2328 - OSPF Version 2. 1998. Internet Engineering Task Force laatima standardi. Viitattu 30.10.2023. <https://datatracker.ietf.org/doc/html/rfc2328> OSPF

RFC 7348 - Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. 2014. Internet Engineering Task Force laatima standardi. Viitattu 8.10.2023. <https://datatracker.ietf.org/doc/html/rfc7348>

Russell, A. 2013. OSI:The Internet that wasn't. Spectrum IEEE. Viitattu 10.09.2023. <https://spectrum.ieee.org/osi-the-internet-that-wasnt>

Spanning Tree Protocol (STP) Overview. 2023. Cisco Meraki tuotesarjan virallinen dokumentaatio. Viitattu 29.10.2023. [https://documentation.meraki.com/MS/Port\\_and\\_VLAN\\_Configuration/Spanning\\_Tree\\_Protocol\\_\(STP\)\\_Overview](https://documentation.meraki.com/MS/Port_and_VLAN_Configuration/Spanning_Tree_Protocol_(STP)_Overview)

Transport Layer Explanation – Layer 4 of the OSI Model. N.d. Study CCNA verkkosivuston artikkeli. <https://study-ccna.com/transport-layer/>

What is an Open Systems Interconnection Model (OSI Model)? N.d. Infoblox sanasto. Viitattu 10.09.2023. <https://www.infoblox.com/glossary/open-systems-interconnection-model-osi-model/>

What is Layer 5 of the OSI Model: Session Layer? N.d. Infoblox sanasto. Viitattu 04.09.2023. <https://www.infoblox.com/glossary/layer-5-of-the-osi-model-session-layer/>

What is Layer 6 of The OSI Model: Presentation Layer? N.d. Infoblox sanasto. Viitattu 04.09.2023 <https://www.infoblox.com/glossary/layer-6-of-the-osi-model-presentation-layer/>

What is OSI Model? N.d. Amazon AWS pilvipalvelutuottajan artikkeli. Viitattu 30.11.2023. <https://aws.amazon.com/what-is/osi-model/>

What is VXLAN? 2023. Artikkeli Juniper laitevalmistajan sivustolla. Viitattu 8.10.2023. <https://www.juniper.net/us/en/research-topics/what-is-vxlan.html>

## Liitteet

### Liite 1. R1 reitittimen konfiguraatio

```
R1#sh run
Building configuration...

Current configuration : 1923 bytes
!
! Last configuration change at 09:03:05 UTC Thu Nov 30 2023
!
version 16.8
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
ip multicast-routing distributed
subscriber templating
multilink bundle-name authenticated
!
!
license boot level ax
no license smart enable
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
!
redundancy
bridge-domain 10
member vni 5000
member GigabitEthernet3 service-instance 10
!
cdp run
!
interface Loopback0
ip address 4.4.4.4 255.255.255.255
ip pim sparse-mode
!
interface GigabitEthernet1
mtu 9216
ip address 10.0.0.2 255.255.255.252
ip pim sparse-mode
```

```
negotiation auto
no mop enabled
no mop sysid
!
interface GigabitEthernet2
mtu 9216
ip address 192.168.34.3 255.255.255.0
ip pim sparse-mode
negotiation auto
no mop enabled
no mop sysid
!
interface GigabitEthernet3
mtu 9216
no ip address
negotiation auto
no mop enabled
no mop sysid
service instance 10 ethernet
encapsulation untagged
!
!
interface nve1
no ip address
source-interface Loopback0
member vni 5000 mcast-group 239.1.1.1
no mop enabled
no mop sysid
!
router ospf 1
network 4.4.4.4 0.0.0.0 area 0
network 10.0.0.0 0.0.0.3 area 0
network 192.168.34.0 0.0.0.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
ip pim rp-address 4.4.4.4
!
control-plane
!
line con 0
logging synchronous
stopbits 1
line vty 0
login
line vty 1
login
length 0
line vty 2 4
login
!
wsma agent exec
!
```

```
wsma agent config
!
wsma agent filesys
!
wsma agent notify
!
!
```

## Liite 2. R2 reitittimen konfiguraatio

```
2#sh run
Building configuration...

Current configuration : 3947 bytes
!
! Last configuration change at 09:03:11 UTC Thu Nov 30 2023
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
platform console serial
!
hostname R2
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
!
ip multicast-routing distributed
!
login on-success log
!

!
subscriber templating
!

!
multilink bundle-name authenticated
!

!
crypto pki trustpoint TP-self-signed-
enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-
revocation-check none
rsa-keypair TP-self-signed-
!
!
crypto pki certificate chain TP-self-signed-

!
!
```

```

license udi pid CSR1000V sn XXXXXXXXXXXX
no license smart enable
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
!
!
!
redundancy
bridge-domain 10
member vni 5000
member GigabitEthernet3 service-instance 10
!
!
interface Loopback0
ip address 2.2.2.2 255.255.255.255
ip pim sparse-mode
!
interface GigabitEthernet1
mtu 9216
ip address 10.0.0.1 255.255.255.252
ip pim sparse-mode
negotiation auto
no mop enabled
no mop sysid
!
interface GigabitEthernet2
mtu 9216
ip address 10.0.0.5 255.255.255.252
ip pim sparse-mode
negotiation auto
no mop enabled
no mop sysid
!
interface GigabitEthernet3
mtu 9216
no ip address
negotiation auto
no mop enabled
no mop sysid
service instance 10 ethernet
encapsulation untagged
!
!
interface nve1
no ip address
source-interface Loopback0
member vni 5000 mcast-group 239.1.1.1
no mop enabled
no mop sysid
!
router ospf 1
network 2.2.2.2 0.0.0.0 area 0
network 10.0.0.0 0.0.0.3 area 0
network 10.0.0.4 0.0.0.3 area 0
!
ip forward-protocol nd
no ip http server

```



```
no ip http secure-server
ip pim rp-address 4.4.4.4
!
```

```
!
control-plane
!
```

```
!
line con 0
stopbits 1
line vty 0 4
login
```

```
!
!
End
```

### **Liite 3. R3 reitittimen konfiguraatio**

```
R3#sh run
Building configuration...
```

```
Current configuration : 4029 bytes
!
! Last configuration change at 09:03:13 UTC Thu Nov 30 2023
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
platform console serial
!
hostname R3
!
boot-start-marker
boot-end-marker
!
!
logging console emergencies
!
no aaa new-model
!
ip multicast-routing distributed
!
login on-success log
!
!
subscriber templating
!
multilink bundle-name authenticated
!
crypto pki trustpoint TP-self-signed-
enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-
revocation-check
rsa-keypair TP-self-signed-
!
```

```

!
crypto pki certificate chain TP-self-signed-
certificate self-signed 01

license udi pid CSR1000V sn XXXXXXXX
no license smart enable
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
redundancy
bridge-domain 10
member vni 5000
member GigabitEthernet3 service-instance 10
!

!
cdp run
!

interface Loopback0
ip address 6.6.6.6 255.255.255.255
ip pim sparse-mode
!
interface GigabitEthernet1
mtu 9216
ip address 10.0.0.6 255.255.255.252
ip pim sparse-mode
negotiation auto
no mop enabled
no mop sysid
!
interface GigabitEthernet2
mtu 9216
ip address 192.168.34.4 255.255.255.0
ip pim sparse-mode
negotiation auto
no mop enabled
no mop sysid
!
interface GigabitEthernet3
mtu 9216
no ip address
negotiation auto
no mop enabled
no mop sysid
service instance 10 ethernet
encapsulation untagged
!
!
interface nve1
no ip address
source-interface Loopback0
member vni 5000 mcast-group 239.1.1.1
no mop enabled
no mop sysid
!
router ospf 1
network 6.6.6.6 0.0.0.0 area 0

```

```
network 10.0.0.4 0.0.0.3 area 0
network 192.168.34.0 0.0.0.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
ip pim rp-address 4.4.4.4
!

control-plane
!

line con 0
logging synchronous
stopbits 1
line vty 0 4
login
end
```