

## Reititysprotokollien EIGRP ja OSPF vertailu virtuaaliympäristössä

Roni Selander



<b>Tekijä</b> Roni Selander	
<b>Koulutusohjelma</b> Tietojenkäsittelyn koulutusohjelma	
<b>Raportin/Opinnäytetyön nimi</b> Reititysprotokollien EIGRP ja OSPF vertailu virtuaaliympäristössä	<b>Sivu- ja liitesivumäärä</b> 25 + 10
<p>Nykyinen internet ei toimisi kuten se toimii ilman toimivia reititysprotokollia, eivätkä organisaatiot olisi niin tehokkaita kuin ne ovat ilman toimintakykyisiä verkkoyhteyksiä. Useissa tilanteissa siis suositaan dynaamista reititystä staattisen sijaan.</p> <p>Opinnäytetyön tarkoituksena on ottaa selvää kahdesta yleisesti käytetystä reititysprotokollasta EIGRP:sta ja OSPF:sta sekä luoda samankaltaiset toteutukset kahdessa eri virtuaaliympäristössä. Nämä protokollat valikoituivat, koska ne ovat yleisimmässä käytössä sekä niitä käsitellään tunnetussa Cisco Certified Network Associate (CCNA 200-125) sertifikaattikokeessa.</p> <p>Työ on jaettu kahteen osaan, teoriaan ja käytäntöön. Ensin käydään teorian kautta läpi yleisemmin tietoverkot ja reititys. Tarkemmin syvennytään näihin kahteen reititysprotokollaan toiminnallisuuksien ja konfiguraatioiden kautta.</p> <p>Käytännön osuudessa selvitetään, kuinka protokollat konfiguroidaan virtuaaliympäristöissä, joissa käytetään Cisco Packet Tracer -ohjelmaa. Ohjelma oli entuudestaan tuttu eikä ollut tarvetta käyttää kuin Ciscon laitteita. Ympäristöt toteutettiin niin, että eri sisäverkot ovat hypoteettisesti viisi eri kampusta. Lähtötilanteessa liikenne ei vielä kulje eri kampuksien välillä, mutta reititysprotokollien avulla tilanne korjaantuu.</p>	
<b>Asiasanat</b> Tietoliikenne, reititys, OSPF, EIGRP, Cisco	

## Sisällys

1	Johdanto .....	1
1.1	Keskeiset käsitteet .....	1
2	Tietoliikenneverkot .....	3
2.1	Viitemallit .....	3
3	Reititys .....	5
3.1	Exterior Gateway ja Interior Gateway protokollat.....	6
3.2	Etäisyysvektoriprotokollat.....	6
3.3	Linkkitilaprotokollat.....	7
4	Open Shortest Path First (OSPF)-protokolla .....	9
4.1	Toiminnallisuus .....	9
4.2	Konfigurointi.....	10
5	Enhanced Interior Gateway Routing Protocol (EIGRP)-protokolla .....	13
5.1	Toiminnallisuus .....	13
5.2	Konfigurointi.....	15
6	Protokollat käytännössä .....	17
6.1	Skenaario OSPF .....	18
6.2	Skenaario EIGRP.....	21
7	Pohdinta.....	25
	Lähteet .....	26
	Liitteet.....	27
	Liite 1. OSPF verkkokuva.....	27
	Liite 2. EIGRP verkkokuva .....	28
	Liite 3. OSPF skenaarion Helsinki reitittimen koko konfiguraatio .....	29
	Liite 4. OSPF skenaarion Porvoo reitittimen koko konfiguraatio .....	31
	Liite 5. EIGRP skenaarion Helsinki reitittimen koko konfiguraatio .....	33
	Liite 6. EIGRP skenaarion Porvoo reitittimen koko konfiguraatio .....	35

# 1 Johdanto

Opinnäytetyön tarkoitus sekä tutkimuskysymys on niin teorian kuin käytännön avulla selvittää, mitä ovat reititysprotokollat EIGRP ja OSPF. Tutkimuskysymys pitää sisällään myös sen, kuinka nämä reititysprotokollat konfiguroidaan reitittimiin sekä selvittää suurimmat erot konfiguroinneissa.

Lähdetään ensin tutkimaan näitä protokollia teorian kautta, joissa tärkeimpinä lähteinä toimivat niin Todd Lammlen kuin Wendell Odomin Ciscon CCNA 200-125 sertifiikaattiin valmistavat kirjat. Käydään ensin tietoverkkoja ja reititystä yleisemmin läpi, kuten kuinka protokollat usein jaetaan. Yksityiskohtaisemmin käydään läpi tutkimuskohteena olevat reititysprotokollat.

Käytännön osuudessa toteutetaan kaksi hyvin samanlaista esimerkkiverkkoa. Verkot suunnitellaan ja luodaan Ciscon Packet Tracer -ohjelmalla. Ensimmäisessä skenaariossa konfiguroidaan OSPF reitittimiin ja testauksien avulla huomataan, kuinka liikenne alkaa kulkemaan eri verkkojen välillä. Toisessa skenaariossa tehdään tarvittavat konfiguraatiot, että liikenne alkaa kulkemaan EIGRP reititysprotokollan avulla. Vaikka tähän toiseen skenaarioon on tuotu muutama asia lisää, ne ovat toiminnallisuuksiltaan samanlaisia ja pohjana toimii viisi eri sisäverkkoa, jotka yhdistyvät reititysprotokollien avulla toisiinsa. Skenaariot toteutettiin niin, että ne olisivat mahdollista toteuttaa myös fyysisillä laitteilla.

## 1.1 Keskeiset käsitteet

Aliverkonpeite	Jakaa IP-osoitteen aliverkon osoitteeseen ja isännän yksilöivään osoitteeseen
EIGRP	Enhanced Interior Gateway Routing Protocol, reititysprotokolla
BDR	Backup Designated Router, OSPF-protokollaa käytettäessä varalla oleva hallitseva reititin
Cisco	Yksi maailman johtavista verkkolaitteita valmistavista yrityksistä
DR	Designated Router, OSPF-protokollaa käytettäessä verkon hallitseva reititin
DROther	OSPF-protokollaa käytettäessä verkossa oleva reititin

LAN	Local area network eli lähiverkko. Rajoitetulla alueella toimiva liikenneverkko, esimerkiksi yrityksen verkko
Konvergenssi	Reititysprotokollien prosessi löytää uusi reitti, kun verkossa tapahtuu muutoksia
Loopback portti	Looginen portti reitittimellä
OSPF	Open Shortest Path First, reititysprotokolla
Packet Tracer	Ciscon kehittämä virtualisointiohjelma simuloimaan verkkoja
Ping	Työkalu, jolla voidaan testata eri laitteiden saatavuutta toisiinsa verkoissa
WAN	Wide Area Network eli laajaverkko. Yhdistää lähiverkot laajemmiksi verkoiksi
Wildcard aliverkonpeite	Käänteinen peite, jossa 1 ja 0 tarkoittavat päinvastaisia

## 2 Tietoliikenneverkot

Tietoliikenneverkkojen tarkoitus on siirtää tietoa paikasta toiseen, esimerkiksi lataamalla dokumentti palvelimelta tietokoneelle tai lähettää sähköposti tietokoneelta toiselle tietokoneelle. Näitä laitteita kutsutaan päätelaitteiksi. Kun verkolla halutaan tehdä muutakin kuin yhdistää kaksi päätelaitetta toisiinsa, tarvitaan myös tietoverkkolaitteita kuten kytkimiä ja reitittäjiä. Näin muodostuu lähiverkko eli LAN (Local Area Network). Lähiverkkoja on niin kouluissa, yrityksissä kuin yksityiskodeissakin. Lähiverkot yhdistyvät toisiin lähiverkkoihin palveluntarjoajien runkoverkkojen avulla sekä samalla yhdistytään ja kasvatetaan maailmanlaajuisia verkkoja eli internettiä.

### 2.1 Viitemallit

Helpottaakseen ihmisiä ymmärtämään tietoverkkoja paremmin on syntynyt malleja, jotka jakavat eri toiminnallisuudet eri kerroksiin ja niihin kuuluviin tiedonsiirtoprotokolliin. TCP/IP mallia voidaan pitää täsmällisempänä kuin OSI-mallia, koska lähes kaikki tietokoneet käyttävät TCP/IP protokollia. OSI-malli kuitenkin on edelleen hyvin yleisessä käytössä sen terminologian takia (Odom 2016, 21). Alapuolella olevassa taulukossa 1 vertaillaan ja havaitaan erot näiden kolmen eri mallin välillä.

Taulukko 1. OSI-malli verrattuna kahteen TCP/IP malliin (mukaillen Odom, 33)

	<b>OSI</b>	<b>TCP/IP (alkuperäinen)</b>	<b>TCP/IP (päivitetty)</b>
7	Application	Application	Application
6	Presentation		
5	Session		
4	Transport	Transport	Transport
3	Network	Internet	Network
2	Data Link	Link	Data Link
1	Physical		Physical

Verkkoteknologian terminologian ymmärtämiseksi OSI-malli pitää sisäistää. Verkkodokumenttaation yhteydessä etenkin kerrokset 2 ja 3 ovat hyvin käytettyjä. Myös LAN kytkimeen saatetaan viitata sanomalla "Layer 2 kytkin" nimenomaan peilaten OSI-malliin. Kerrokset ovat avattu sanallisesti tarkemmin alapuolella olevassa taulukossa 2.

Taulukko 2. OSI-mallin kerroksien kuvaukset (mukaillen Odom, 34)

Kerros	Toiminnallisuuden kuvaus
7	<b>Sovelluskerros.</b> Tarjoaa rajapinnan sovellukselta tietoverkkoon toimittamalla protokollan toiminnallisuudella, joka on sovelluksen ymmärrettävissä, esim. SMTP.
6	<b>Esitystapakerros.</b> Neuvottelee datan muotoilun, kuten ASCII teksti tai kuvan tiedostotyyppin, esim. JPEG.
5	<b>Istuntokerros.</b> Määrittelee, miten eri istunnot aloitetaan, prosessoidaan sekä lopetetaan.
4	<b>Kuljetuskerros.</b> Vastuussa datan siirtämisestä kahden eri päätelaitteen välillä sekä virheiden käsittelystä. Protokollina TCP ja UDP.
3	<b>Verkkokerros.</b> Loogiset osoitteet, reititys sekä parhaan reitin oppiminen ja valinta. Protokollana IP, laitteena esim. reititin.
2	<b>Siirtokerros.</b> Määrittellään mahdollisuudet kehysten ja pakettien lähetykseen. Luetaan MAC-osoite. Laitteina esim. kytkimet ja langattomat tukiasemat.
1	<b>Fyysinen kerros.</b> Viitataan niin kaapeleihin, liittimiin kuin sähkönsiirtoon.

### 3 Reititys

Reititys viittaa datapaketin lähettämiseen laitteelta toiselle laitteelle, joka on toisessa verkossa. Reitittimet eivät juuri välitä päätelaitteista vaan toisista verkoista sekä parhaista reiteistä näihin verkkoihin. Reititin tarvitsee vähintään seuraavat tiedot reitittääkseen paketit tehokkaasti:

- Kohdeosoitteen
- Naapurireitittimet, joilta se oppii muista verkoista
- Mahdolliset reitit muihin verkkoihin
- Parhaan reitin eri verkkoihin
- Kuinka ylläpitää ja varmistaa reititysinformaatio

Reititin oppii eri verkoista joko naapurireitittimeltä tai laitteen ylläpitäjältä. Näiden tietojen pohjalta reititin rakentaa reititystaulun, joka on käytännössä kartta eri verkoista ja kuinka nämä verkot löytyvät (Lammle 2016, 359). Kuvasta 1 selviää, miltä reititystaulu voi näyttää.

```
Porvoo>ena
Porvoo#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 11 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
O       10.1.2.0/24 [110/2] via 10.1.1.3, 00:00:50, GigabitEthernet0/0
O       10.1.3.0/24 [110/65] via 10.1.1.3, 00:00:50, GigabitEthernet0/0
O       10.1.4.0/24 [110/65] via 10.1.1.3, 00:00:50, GigabitEthernet0/0
O IA    10.1.20.0/24 [110/3] via 10.1.1.3, 00:00:40, GigabitEthernet0/0
O       10.1.30.0/24 [110/66] via 10.1.1.3, 00:00:50, GigabitEthernet0/0
O       10.1.40.0/24 [110/66] via 10.1.1.3, 00:00:50, GigabitEthernet0/0
C       10.1.50.0/24 is directly connected, GigabitEthernet0/1
L       10.1.50.1/32 is directly connected, GigabitEthernet0/1
O       10.1.60.0/24 [110/2] via 10.1.1.2, 00:00:50, GigabitEthernet0/0
```

Kuva 1. Esimerkki reititystaulusta

Jos reititin ei ole liitettyä suoraan toiseen reitittimeen, on kaksi tapaa oppia toisista verkoista. Ensimmäinen tapa on määrittää reitti staattisesti komennolla *ip route*. Tähän tarvitaan tiedot vähintään näistä kahdesta asiasta, eli kohdeverkon osoite sekä suunta, miten sinne päästään. Suunta voidaan määrittää kahdella eri tavalla, joko antamalla seuraavan hypyn IP-osoite (next hop address) tarkoittaen seuraavan reitittimen IP-osoitetta reitillä kohdeverkkoon tai antamalla poistumisportti (Lammle T. 2016, 383). Alla esimerkki, kuinka komento annetaan sekä komento avattuna:

**ip route 172.16.3.0 255.255.255.0 192.168.2.4**

ip route [kohteen IP-osoite] [kohteen aliverkon maski] [seuraavan hypyn IP-osoite/poistumisportti]

Staattinen reititys ei kuitenkaan skaalaudu hyvin sekä sitä olisi hyvin työlästä ylläpitää. Tästä syystä sitä käytetään vain pienissä verkoissa tai erikoistilanteissa.

Toinen tapa on dynaaminen reititys. Tällä tarkoitetaan reititystä, jossa käytetään protokollia löytämään verkkoja sekä päivittääkseen reitittimen reititystauluja. Tätä pidetään huomattavasti vähemmän työläänä kuin staattista reititystä. Reititysprotokolla määrittää säännösten, jota reititin noudattaa kommunikoidakseen naapurireitittimien kanssa vaihtaakseen informaatioita reitityksestä. Reititysprotokollat voidaan jaotella eri ryhmiin niin käyttötarkoituksen kuin toimintaperiaatteen perusteella (Lammle T. 2016, 390).

### **3.1 Exterior Gateway ja Interior Gateway protokollat**

Käyttötarkoituksen mukaan protokollat voidaan karkeasti jakaa niin, että Interior Gateway protokollia (IGP) käytetään lähiverkon sisällä ja Exterior Gateway protokollia (EGP) käytetään yhdistämään lähiverkkoja toisiinsa. Tässä yhteydessä lähiverkkoja kutsutaan usein myös autonomiseksi järjestelmäksi (Autonomous System, AS). AS on joko yksittäinen verkko tai kokoelma verkkoja, joita hallitsee sitä käyttävä organisaatio. Käytännössä tämä tarkoittaa, että samaan AS:n kuuluvat reitittimet jakavat reititystaulutiedot keskenään (Lammle T. 2016, 392).

Johtava EGP, jota myös internetin reititysprotokollaksi kutsutaan, on Border Gateway protokolla (BGP). Routing Information Protocol (RIP), Intermediate System to Intermediate System (IS-IS), Open Shortest Path First (OSPF) sekä Enhanced Interior Gateway Routing Protocol (EIGRP) ovat tunnettuja Interior Gateway protokollia. RIP on yksi vanhimmista reititysprotokollista ja käytännössä jo historiaa. IS-IS alun perin kehitettiin, kun oletettiin OSI-pinosta tulevan hallitseva TCP/IP-pinon sijaan, IS-IS protokollasta on tehty yhteensopiva versio myös TCP/IP-pinon kanssa (Hunt, C.1997, luku 6). Myöhemmin tässä työssä käyn tarkemmin läpi OSPF:n ja EIGRP:n.

### **3.2 Etäisyysvektori-protokollat**

Etäisyysvektori-protokollien (distance vector) toimintaperiaate on yksinkertainen. Jokainen reitittävä laite verkossa ylläpitää tietoa etäisyydestä (distance) tai kustannuksesta (cost) jokaiseen tiedettyyn kohdeosoitteeseen. Reitti, jolla on pienempi kustannus on parempi

kuin reitti, jolla on suurempi kustannus. Pienimmän kustannuksen omaavasta reitistä tulee näin ollen ensisijainen reitti. Protokollat määrittävät suunnan (vector) ja etäisyyden (distance) jokaiseen linkkiin verkon sisällä (Brit D. ym. 2006, 176).

Informaatiota ylläpidetään etäisyysvektoritaulukossa (distance vector table). Taulukkoa mainostetaan säännöllisesti naapurireitittimille. Jokainen reititin prosessoi nämä ilmoitukset ja sitä kautta tekee päätöksen parhaasta mahdollisesta reitistä (Moy 2000, 35).

### 3.3 Linkkitilaprotokollat

Tietoverkkojen kasvaminen ja monimutkaistuminen loi vahvan tarpeen luoda uusia reititys algoritmeja. Näiden algoritmien tarkoitus oli korjata etäisyysvektoriprotokollissa havaitut puutteet. Linkkitila on reitittimen rajapinnan kuvaus (esimerkiksi IP-osoite, aliverkon peite, verkon tyyppi) ja niiden suhde naapurireitittimiin. Kokoelma näistä muodostaa linkkitilatie-tokannan (Britt, D. ym. 2006, 177).

Linkkitila-algoitmien käyttämä prosessi verkon määrittämiseksi on melko suoraviivainen:

1. Jokainen reititin tunnistaa kaikki muut verkkoon kytketyissä reitittämissä laitteissa.
2. Jokainen reititin mainostaa luettelon kaikista suoraan kytketyistä verkkolinkeistä ja kunkin linkin liittyvät kustannukset. Tämä suoritetaan linkin tilan vaihdon kautta (link state advertisements, LSA) muiden verkon reitittimien kanssa.
3. Näitä mainostuksia hyväksikäyttämällä jokainen reititin luo tietokannan, jossa ovat yksityiskohtaiset tiedot nykyisestä verkon topologiasta. Kunkin reitittimen topologiatietokanta on identtinen.
4. Jokainen reititin käyttää topologiatietokannan tietoja laskeakseen parhaimmat reitit jokaiseen kohdeverkkoon. Näitä tietoja käytetään päivittämään reititystaulua (Moy 2000, 39).

Taulukko 3. Reititysprotokollien luokittelu

	Etäisyysvektoriprotokollat	Linkkitilaprotokollat
IGP	RIP	OSPF
	IGRP	IS-IS
	EIGRP	
EGP	BGP	

Nämä lähiverkkojen reititysprotokollat voitaisiin jakaa yksinkertaistettuna niin, että etäisyysvektoriprotokollat oppivat tiedot vain naapurireitittimistään ja käyttävät linkkien mittauksen matkaa. Linkkitilaprotokollia käyttävät reitittimet taas tietävät koko verkon rakenteen. Näiden lisäksi on vielä olemassa kolmas luokka hybridiprotokollat. Merkittävimpänä esimerkkinä voidaan pitää EIGRP-protokollaa (Odom 2017, 175), josta kerron tarkemmin luvussa viisi. Taulukosta 3 on selvemmin nähtävillä reititysprotokollien luokittelu.

## 4 Open Shortest Path First (OSPF)-protokolla

OSPF-protokollan kehitys aloitettiin vuonna 1987 ja yleiseen käyttöön tullut versio valmistui vuonna 1991. On hyvä ymmärtää, mitä internet siihen aikaan oli ja mitä ongelmia OSPF-protokollalla haluttiin korjata. Suuri osa internetistä käytti staattista reititystä. Lähiverkoissa käytetyin dynaaminen reititysprotokolla oli RIP. Koska lähiverkkojen sekä reititustaulujen koko kasvoi, myös RIP:n lähettämien päivityksien vaatima kaistanleveys kasvoi (Moy 2000, 43–44).

Työryhmä, joka muodostettiin Internet Engineering Task Forcen (IETF) toimesta korvaamaan RIP asetettiin suunnitteluvaiheessa tavoitteita. OSPF:n haluttiin kuluttavan vähemmän verkon resursseja sekä haluttiin, että se saavuttaisi konvergenssin nopeammin kuin verkossa tapahtuu muutoksia. Esimerkkeinä verkon muutoksista voidaan pitää, kun linkki, reitittimen portti tai kokonainen reititin hajoaa. Kaikilla reititysprotokollilla on määrättyä löytää uusi reitti haluttuun kohteeseen. Tätä löytämisprosessia kutsutaan konvergenssiksi. OSPF-protokolla haluttiin myös ominaisuudet, jotka parantaisivat protokolla metriikkaa. Nämä saavutettiin ottamalla huomioon reittivaihtoehtojen nopeudet sekä mahdollisuudella määrittää reitille arvo (Moy 2000, 45).

Haluttiin myös ominaisuus nimeltä monipolkureititys (equal cost multipath). Tämä mahdollistaa OSPF:n löytämään useita parhaita reittejä tiettyyn kohteeseen, kun ne ovat olemassa (Moy 2000, 45). Tämän myötä tuli myös mahdolliseksi kuormantasaus, kun on useita samanarvoisia reittejä samaan kohteeseen ja pystytään lähettämään paketteja kahta eri reittiä pitkin (Moy 2000, 98).

OSPF on myös avoimen standardin reititysprotokolla, joten se ei ole sidottu tietyn laitteen merkkiin kuten Ciscoon. Tämä myös selittää OSPF:n suosiota sekä joustavuutta. Yksi tärkeimmistä ominaisuuksista on, että protokolla on tarkoitettu hierarkkiseen käyttöön. Tarkoittaen, että se mahdollistaa isompien verkkojen jakamisen pienimpiin verkkoihin käyttäen alueita (area) (Lammle 2016, 746–747).

### 4.1 Toiminnallisuus

OSPF:n toiminnallisuus voidaan jakaa karkeasti kolmeen osaan:

- Naapurin ja naapuruuden käyttöönotto
- LSA tietojen vaihto
- SPF reitin laskeminen

Naapuruuden muodostamisen aloittaminen on iso osa koko operaatiota. Kun OSPF otetaan käyttöön reitittimellä, se allokoii muistia sille ylläpitääkseen naapuri- ja topologia- tauluja. Kun reititin on arvioinut, mitkä portit on konfiguroitu OSPF:tä varten se tarkistaa ovatko ne aktiivisina ja alkaa lähettämään Hello-paketteja. Näiden pakettien tarkoitus on saada selville naapurit, muodostaa naapuruus sekä ylläpitää näitä suhteita muiden reitittimien kanssa. Paketteja lähetetään säännöllisin väliajoin kaikissa reitittimien porteissa, joissa OSPF on aktivoituna. IP-osoite, joka on tarkoitettu tähän käyttöön, on 224.0.0.5 (Lammle 2016, 751–752).

LSA (Link State Advertisement) on OSPF datapaketti, joka sisältää linkkitila- sekä reititys- informaation, joita jaetaan OSPF reitittimien kesken (Lammle 2016, 750). LSU (Link State Updates) pakettien avulla luodaan verkkotopologia. Lähettämällä näitä paketteja säännöllisin väliajoin myös varmistetaan, että kaikilla OSPF reitittimillä se täsmää. Näiden tietojen avulla reitittimet tekevät tarvittavat laskelmat löytääkseen lyhyimmän reitin. Jokaisen paketin vastaanottanut reititin myös vahvistaa LSA päivityksen (Lammle 2016, 752).

Rakentaakseen reitit OSPF, kuten muutkin linkkitilaprotokollat, reititin laskee lyhyimmän reitin käyttäen Dijkstra Shortest Path First (SPF) algoritmia. Tämän prosessin tehtyä reititin muokkaa linkkitil tietokantaansa (LSDB), rakentaa reitit ja lisää lyhyimmät reitit reititys- tauluunsa (Odom 2017, 175).

## 4.2 Konfigurointi

OSPF:n konfigurointia alkaa aktivoimalla OSPF antamalla komento:

```
Helsinki(config)#router ospf ?  
<1-65535> Process ID
```

Kysymysmerkin tilalle annetaan arvo väliltä 1–65535, joka määrittää OSPF prosessi ID:n. Eri OSPF reitittimien ei tarvitse kuulua saman prosessi ID:n alle kommunikoidakseen tois- tensa kanssa. Se on vain paikallinen uniikki numero, jonka alle OSPF konfiguraatiot mää- ritetään. Yhdellä reitittimellä voi olla samaan aikaan käynnissä useita OSPF prosesseja. (Lammle 2016, 753).

Seuraavaksi kannattaa määrittää reititin ID (RID). Ilman sen määrittämistä OSPF valitsee korkeimman IP-osoitteen sen aktiivisessa portissa ja tämä voi aiheuttaa ympäristössä ta- pahtuvan muutoksen myötä ongelmia. RID:t ovat 32 bittisiä numeroita kuten IP-osoitteita. (Odom 2017, 181). Samalla kannattaa määrittää portit, joissa halutaan aktivoida OSPF

kommunikointi. Tämä komento myös määrittää verkon, jota halutaan mainostettavan muille:

```
Helsinki(config)#router ospf 1
Helsinki(config-router)#router-id ?
A.B.C.D OSPF router-id in IP address format
Helsinki(config-router)# router-id 1.1.1.1
Helsinki(config-router)#network 10.1.1.0 0.0.0.255 area ?
<0-4294967295> OSPF area ID as a decimal value
A.B.C.D OSPF area ID in IP address format
Helsinki(config-router)# network 10.1.1.0 0.0.0.255 area 0
```

Network-sanan jälkeen tulee ensin verkon IP-osoite, tämän jälkeen aliverkonmaski wild-card muodossa sekä määritetään alue (Lammle 2016, 754).

On hyvä tietää myös muutamia show-komentoja liittyen OSPF:n tutkimiseen ja vian selvitykseen. Ensimmäinen on *show ip ospf*, se kertoo tietoa RID:stä, alueista, SPF statistiikkaa sekä LSA ajastuksesta. Tätä komentoa voi käyttää myös portti kohtaisesti lisäämällä portin nimi komennon perään, kuten *show ip ospf int g0/0* (Lammle 2016, 766). Linkkitilastietokannan saa näkyviin komennolla *show ip ospf database*. Kuvasta 2 selviää, minkä syötteen komento antaa.

```
Vierumaki#show ip ospf database
      OSPF Router with ID (2.2.2.2) (Process ID 1)

      Router Link States (Area 0)

Link ID      ADV Router   Age         Seq#         Checksum
Link count
2.2.2.2      2.2.2.2     521        0x80000006  0x00b317 2
5.5.5.5      5.5.5.5     556        0x80000006  0x00d652 3
9.9.9.9      9.9.9.9     555        0x80000006  0x003ac2 3
1.1.1.1      1.1.1.1     523        0x80000006  0x001fbc 2
6.6.6.6      6.6.6.6     522        0x8000000b  0x0059c5 6
4.4.4.4      4.4.4.4     522        0x80000006  0x00ce43 1

      Net Link States (Area 0)

Link ID      ADV Router   Age         Seq#         Checksum
10.1.2.1     6.6.6.6     522        0x80000006  0x000689
10.1.1.3     6.6.6.6     522        0x80000007  0x00b254

      Summary Net Link States (Area 0)

Link ID      ADV Router   Age         Seq#         Checksum
10.1.20.0    4.4.4.4     516        0x80000003  0x00ab81
```

Kuva 2. Kuvankaappaus komennon *show ip ospf database* jälkeen

Komento kertoo reitittimien määrän lähiverkossa sekä naapurireitittimien RID:t. Kolmas hyödyllinen komento on *show ip ospf neighbor*. Se tekee yhteenvedon olennaisista tiedoista liittyen naapurireitittimistä, naapuruuksista ja niiden arvojärjestyksestä (DR, BDR tai

DROTHER). Lisäksi kertoo missä ajassa yhteys katsotaan katkenneeksi, jos ei tule Hello-  
viestiä. Sekä naapureiden IP-osoitteen ja poistumisportin (Lammle 2016, 769). Alapuolella  
olevassa kuvassa 3 on annettu komento ja sen antama syöte.

```
Helsinki#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
9.9.9.9	0	FULL/ -	00:00:37	10.1.4.2	Serial0/1/1
4.4.4.4	1	FULL/BDR	00:00:38	10.1.2.2	GigabitEthernet0/1
5.5.5.5	0	FULL/ -	00:00:36	10.1.3.2	Serial0/1/0
1.1.1.1	1	FULL/DROTHER	00:00:39	10.1.1.1	GigabitEthernet0/0
2.2.2.2	1	FULL/BDR	00:00:38	10.1.1.2	GigabitEthernet0/0

Kuva 3. Komennon *show ip ospf neighbor* antama syöte

## 5 Enhanced Interior Gateway Routing Protocol (EIGRP)-protokolla

EIGRP-protokollaa aloitettiin kehittämään samoihin aikoihin kuin OSPF-protokollaa. Tätä kutsutaankin reititysprotokollien toiseksi aalloksi ja lopullisesti se korvasi edeltäjänsä IGRP-protokollan vuonna 1993. EIGRP-protokollassa yhdistyy niin linkkitilaprotokollan kuin etäisyysvektoriprotokollan ominaisuuksia ja siksi sitä kutsutaan usein hybridiprotokollaksi tai kehittyneeksi etäisyysvektoriprotokollaksi. Haluttiin parempaa metriikkaa ja konvergenssia, mitä 80-luvun protokollissa oli. Syyt tämän protokollan kehittämiseen olivat siis hyvin samanlaiset kuin OSPF-protokollan taustalla (Odom 2017, 226).

Perinteisesti yhtenä huonona ominaisuutena pidetään sitä, että vuoteen 2013 asti protokollan pystyi implementoimaan vain Ciscon reitittimiin. Protokolla oli Ciscon kehittämä, eikä sitä haluttu yleiseen jakeluun. Tämä kuitenkin muuttui vuoden 2013 jälkeen, mutta protokollan levinneisyyden näkökulmasta tämä ei ollut paras mahdollinen päätös. Ennen vuotta 2013 moni yritys ei halunnut ottaa käyttöön EIGRP-protokollaa ja näin ollen olla sidottuna Ciscon reitittimiin vaan säilyttää mahdollisuuden olla riippumaton reititinvalmistajasta (Odom 2017, 227).

Muista reititysprotokollista poiketen EIGRP käyttää reittien valintaan omaa DUAL-algoritmiaan (*Diffusion Update Algorithm*). Tämän algoritmin ansiosta reitityssilmukoita ei synny. Käyttääkseen DUAL-algoritmia reitittimen pitää pystyä selvittämään, mitä laitteita siihen on suoraan kytketty. Hello-viestien avulla EIGRP selvittää ovatko naapurilaitteet reitittimiä vai eivät (Albrightson ym. 2).

EIGRP lähettää päivitystietoja verkosta vain, jos siinä tapahtuu muutoksia, kuten yhteys katkeaa tai laite hajoaa. Protokolla tukee myös luokatonta reititystä (CIDR) sekä kuormantasausta. Sitä myös pidetään tehokkaana kuormittamatta liikaa reitittimen resursseja (Lammle 2016, 683).

### 5.1 Toiminnallisuus

EIGRP eroaa OSPF protokollasta joillain merkittäväillä tavoilla, mutta niissä on myös hyvin paljon samaa. EIGRP voidaan jakaa kolmeen kohtaan, kun reititin liittyy verkkoon:

- Naapuruuden löytäminen
- Topologia tietojen vaihtaminen
- Reittien valinta

Löytääkseen naapurit EIGRP reitittimet lähettävät *Hello* viestejä IP-osoitteeseen 224.0.0.10 löytääkseen potentiaaliset viereiset naapuri EIGRP reitittimet ja suorittaa parametri tarkistukset tarkastaakseen, mistä reitittimistä pitäisi tulla naapureita. Nämä tarkistukset sisältävät neljä kohtaa.

1. Sen pitää läpäistä autentikointi prosessi
2. Pitää olla konfiguroitu käyttämään samaa autonomisen systeemin (AS) numeroa
3. Naapurireitittimen lähde IP-osoitteen *Hello* viestin pitää olla samassa aliverkossa kuin lokaalin reitittimen IP-osoite ja maski
4. Reitittimien K-arvojen pitää täsmätä (Cisco ei suosittele näiden vaihtamista). K-arvot avattu taulukossa 4.

Taulukko 4. K-arvojen selitykset (mukaillen Lammle, 699)

K-arvo	Osa	Kuvaus
K1	Kaistanleveys	Reitin alhaisin kaistanleveys
K2	Kuorma	Reitin huonoin kuormitus pakettinopeuden perusteella
K3	Viive	Reitin kumulatiivinen rajapinnan viive
K4	Luotettavuus	Luotettavuus perustuen reitin hengissä pysymiseen
K5	MTU	Pienin MTU reitissä (ei käytetä reitin laskemisessa)

Kun kahdesta reitittimestä on tullut naapureita, on naapurisuus suhde hyvin yksinkertainen. Ne siirtyvät suoraan tilaan, jossa ne alkavat vaihtamaan topologiatietoja käyttämällä EIGRP päivitysviestejä (Odom 2017, 234–235).

Vaihtaakseen topologiatietoja reitittimien välillä, EIGRP lähettää päivitysinformaatiot IP-osoitteeseen 224.0.0.10. Näitä viestejä lähettäessä ei käytetä yleisempiä protokollia UDP tai TCP, vaan käytetään protokollaa nimeltä RTP (Reliable Transport Protocol). Käyttämällä tätä protokollaa vältetään reitityssilmukoilta, koska reititin tietää varmasti, että naapurireititin on vastaanottanut lähetetyn informaation (Odom 2017, 236).

Oletuksena EIGRP käyttää metriikassaan kahta osaa, kaistanleveyttä ja viivettä. Muitakin asetuksia on mahdollista käyttää suoraan peilaten K-arvoihin, kuormaan ja luotettavuuteen. Alla kaava oletusasetuksilla (Lammle 2016, 699).

$$\text{Metriikka} = ((10^7/\text{kaistanleveys})+\text{viive})*256$$

## 5.2 Konfigurointi

Ensimmäiseksi aktivoidaan EIGRP antamalla komento:

```
Haaga(config)#router eigrp ?  
<1-65535> Autonomous system number
```

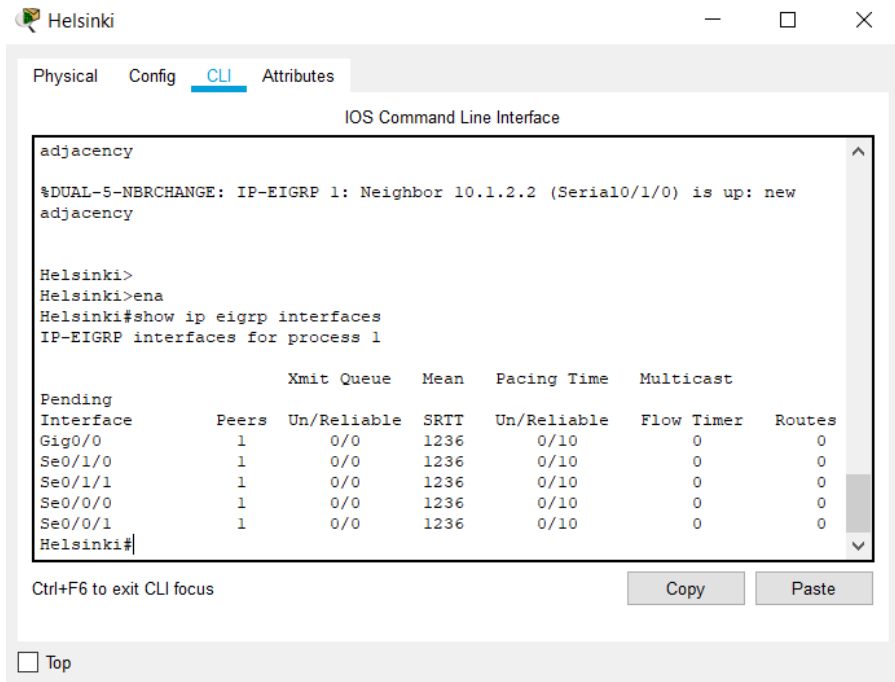
Kysymysmerkin tilalle annetaan arvo väliltä 1-65535, joka määrittää autonomisen järjestelmän numeron. Tämä toimii eri tapaan kuin OSPF:n reititin ID, toisistaan eroavat numerot eivät voi kommunikoida toistensa kanssa (Odom 2017, 249).

Protokollan aktivoinnin jälkeen annetaan komento, joilla mainostetaan verkkoja:

```
Haaga(config-router)#network ?  
A.B.C.D Network number  
Haaga(config-router)#network 10.1.2.0 ?  
A.B.C.D EIGRP wild card bits  
<cr>  
Haaga(config-router)#network 10.1.2.0 0.0.0.255
```

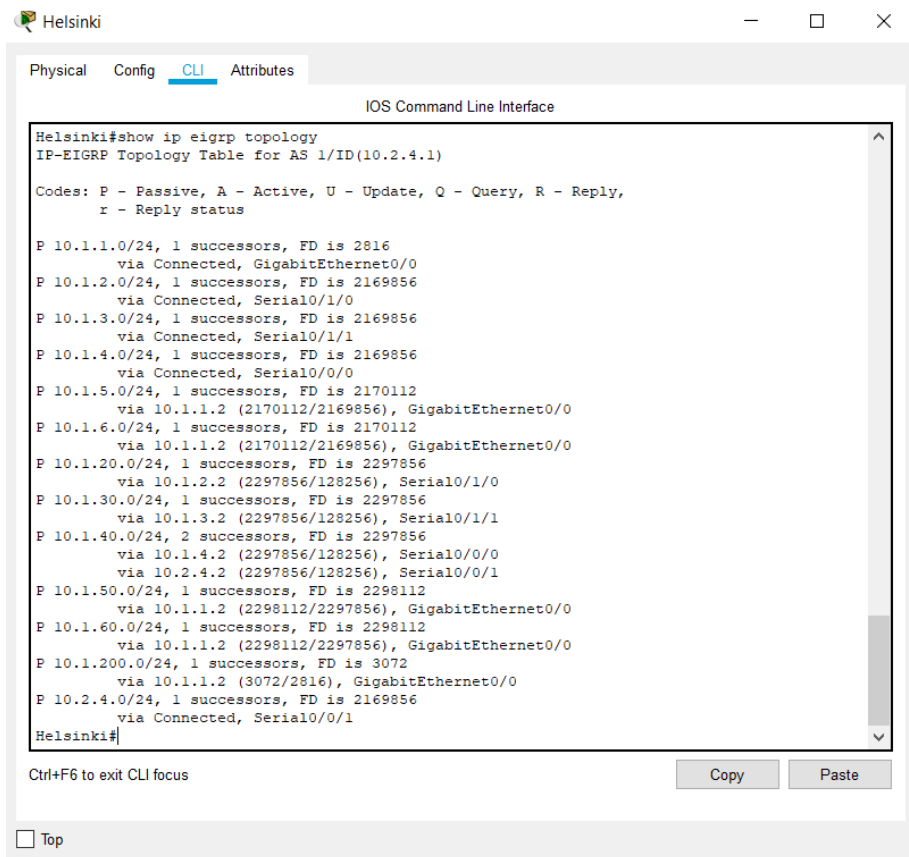
Network-sanan jälkeen tulee ensin verkon IP-osoite, jonka jälkeen aliverkonmaski wild-card muodossa. Jätettiin siis pois alue ja reititin ID:n määrittäminen, jos verrataan suoraan OSPF-protokollaan. (Lammle 2016, 692).

Tärkeimmät show-komennot liittyen EIGRP-protokollaan ovat *show ip eigrp interfaces* sekä *show ip eigrp topology*. Ensimmäinen listaa portit, joissa protokolla on aktivoituna sekä, minne reititin tällä hetkellä lähettää Hello-viestejä löytääkseen lisää reitittämiä. Kuvassa 4 komento on annettu sekä miltä syöte näyttää.



Kuva 4. Komennon *show ip eigrp interfaces* antama syöte

Toinen komento antaa tietoja verkkojen autonomisista järjestelmistä ja seuraajista (successor). Seuraajalla tarkoitetaan parasta reittiä kohteeseen (Lammle 2016, 713). Kuvassa 5 näkyy, miltä komennon antama syöte näyttää.

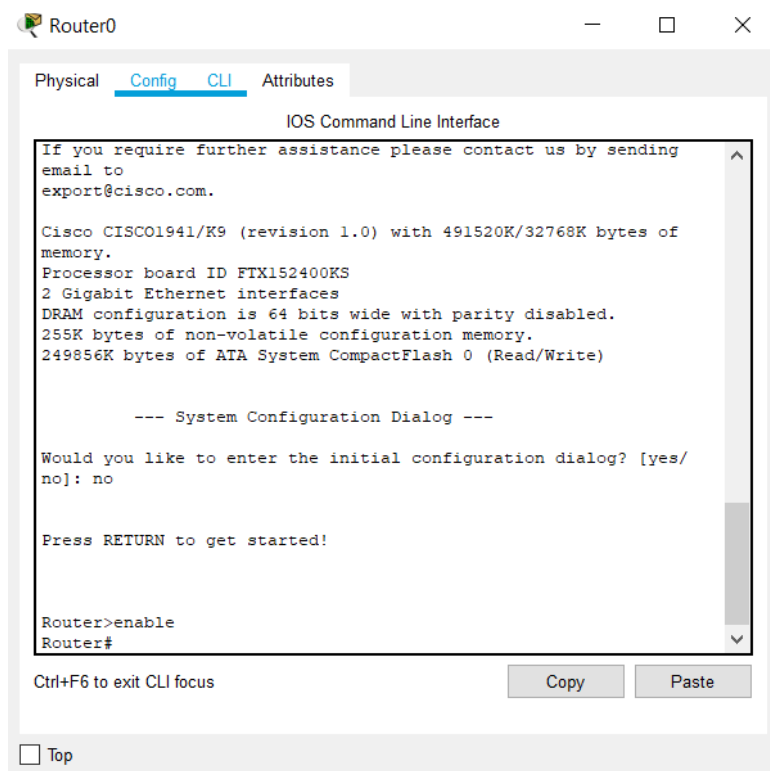


Kuva 5. Komennon *show ip eigrp topology* antama syöte

## 6 Protokollat käytännössä

Tässä työssä luodaan kaksi hyvin samanlaista verkkoa, joissa molemmissa toimii viisi eri sisäverkkoa, ja ne yhdistyvät toisiinsa WAN-verkkojen avulla. Ympäristöjen tekemiseksi ja verkkokuvien luomiseksi käytettiin Ciscon Packet Tracer -ohjelmaa. Virtuaaliympäristöjen luomiseksi olisi ollut muitakin vaihtoehtoja kuten GNS3 ja VIRL. GNS3-ohjelmaa varten olisi tarvinnut kuitenkin verkkolaitteiden kuvakkeet, joiden hankkiminen olisi ollut haastavaa. VIRL-ohjelman lisenssi maksaa jopa 199 dollaria vuodessa.

Packet Tracer on Ciscon kehittämä ja tarjoama tietoliikenneverkkojen virtualisointiohjelma, joka on opiskelijoille ilmainen. Ohjelmalla pystyy konfiguroimaan laitteita komentokehötteen avulla ja kaapeloimaan laitteet toisiinsa aivan kuten oikeillakin laitteilla. Alapuoalla olevassa kuvassa 6 on näkymä, kun painaa ohjelman sisällä verkkolaitetta ja komentokehote avautuu. Oikeilla laitteilla samaan tilaan päästään liittämällä tietokone verkkolaitteeseen kiinni konsolikaapelilla sekä terminaaliemulaattorihjelmalla kuten PuTTY.



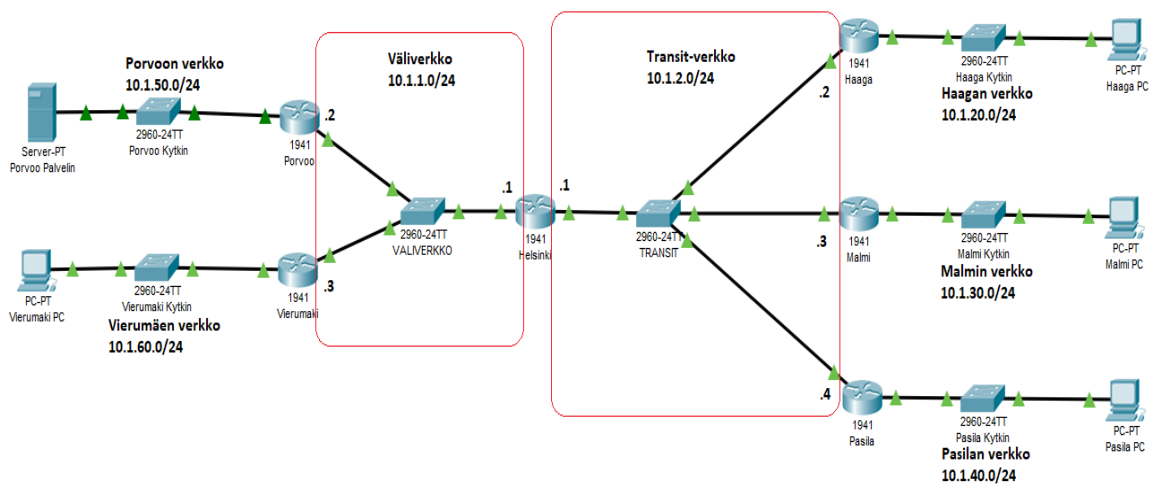
Kuva 6. Näkymä mallin 1941 komentokehotteesta.

Packet Tracer on sidonnainen Cisco laitteisiin, joten muiden, kuten Huawei tai Juniperin verkkolaitteita, sillä ei pysty simuloimaan. Tässä työssä on tarkoitus keskittyä reititysprotokoliin Ciscon laitteilla, joten se sopii tähän mainiosti.

Pienen kokeilujakson jälkeen Packet Tracer -ohjelmalla valitsin käytettäviksi reitittimiksi C1900 –sarjan mallit 1941 ja kytkimiksi Catalyst 2960 -sarjan mallit 24TT. Nämä mallit ovat hinnoiltaan myös maltillisia, jos toteutuksen haluaa tehdä oikeilla laitteilla sekä sopivat tämän työn simulaatioihin.

## 6.1 Skenaario OSPF

Verkkolaitteet on asetettu Packet Tracer -ohjelmaan kuvan 7 mukaisesti. Ympäristö koostuu viidestä eri sisäverkosta, joissa on yksi kytkin, reititin ja päätelaite jokaista sisäverkkoa kohden. Näiden lisäksi verkot yhdistyvät väliverkon ja transit-verkon avulla toisiinsa. Taulukosta 5 selviää, mitkä IP-osoitteet määritetään eri verkkolaitteiden porteille.



Kuva 7. Verkon topologia

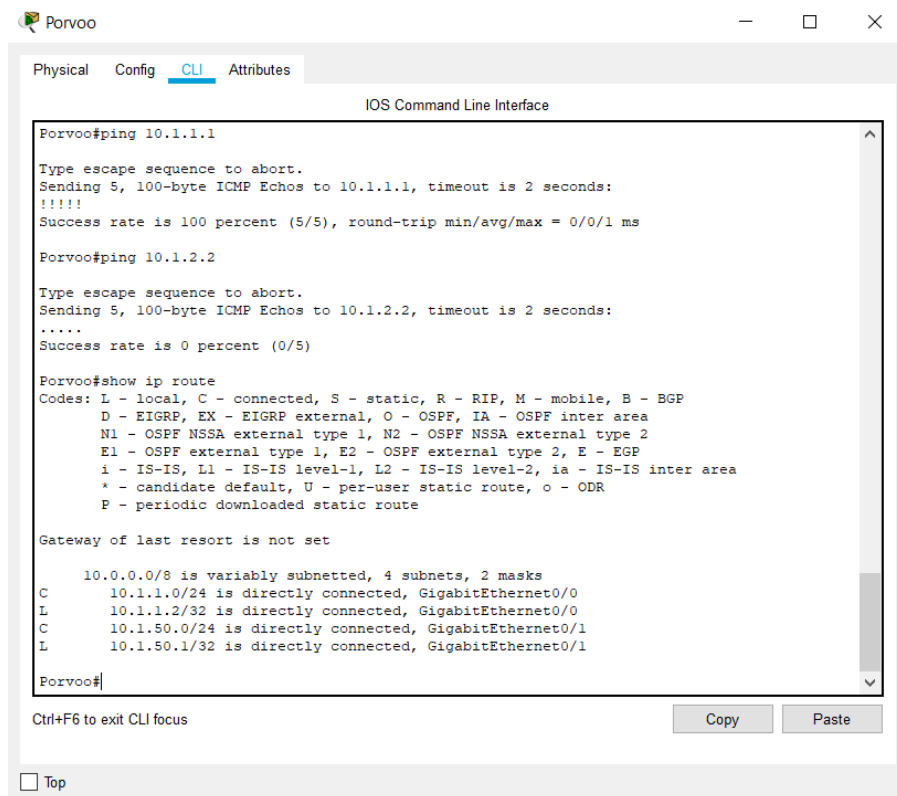
Taulukko 5. Esimerkkiverkon reitittimien porttien IP-osoitteet ja aliverkkojen peitteet

Reititin	Portti	IP-osoite ja aliverkonpeite
Helsinki	G0/0	10.1.1.1 255.255.255.0
	G0/1	10.1.2.1 255.255.255.0
Haaga	G0/0	10.1.2.2 255.255.255.0
	G0/1	10.1.20.1 255.255.255.0
Malmi	G0/0	10.1.2.3 255.255.255.0
	G0/1	10.1.30.1 255.255.255.0
Pasila	G0/0	10.1.2.4 255.255.255.0
	G0/1	10.1.40.1 255.255.255.0
Porvoo	G0/0	10.1.1.2 255.255.255.0
	G0/1	10.1.50.1 255.255.255.0
Vierumäki	G0/0	10.1.1.3 255.255.255.0
	G0/1	10.1.60.1 255.255.255.0

Koska verkkolaitteet ovat kokonaan tyhjiä, konfiguroidaan niille nimet sekä IP-osoitteet sekä sopivat kuvaukset porteille. Oletuksena portit ovat suljettuina niin annetaan myös *no shutdown* komento, joka nostaa ne ylös. Käyn konfiguraation eri vaiheet läpi käyttämällä esimerkkinä Helsinki -nimistä reititintä.

```
Router(config)# hostname Helsinki
Helsinki(config)# int g0/0
Helsinki(config-if)# description VALIVERKKO
Helsinki(config-if)# ip address 10.1.1.1 255.255.255.0
Helsinki(config-if)# no shut
Helsinki(config-if)# int g0/1
Helsinki(config-if)# description TRANSIT
Helsinki(config-if)# ip address 10.1.2.1 255.255.255.0
```

Muutkin reitittimet konfiguroidaan omilla asetuksillaan verkkotopologian sekä taulukon viisi mukaisesti. Kun reitittimien käyttämille porteille on konfiguroitu IP-osoitteet, testataan yhteyksiä Porvoo -nimiseltä reitittimeltä ja tarkistetaan, miltä reititystaulu tällä hetkellä näyttää.



```
Porvoo
Physical Config CLI Attributes
IOS Command Line Interface
Porvoo#ping 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms

Porvoo#ping 10.1.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Porvoo#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.2/32 is directly connected, GigabitEthernet0/0
C       10.1.50.0/24 is directly connected, GigabitEthernet0/1
L       10.1.50.1/32 is directly connected, GigabitEthernet0/1

Porvoo#
```

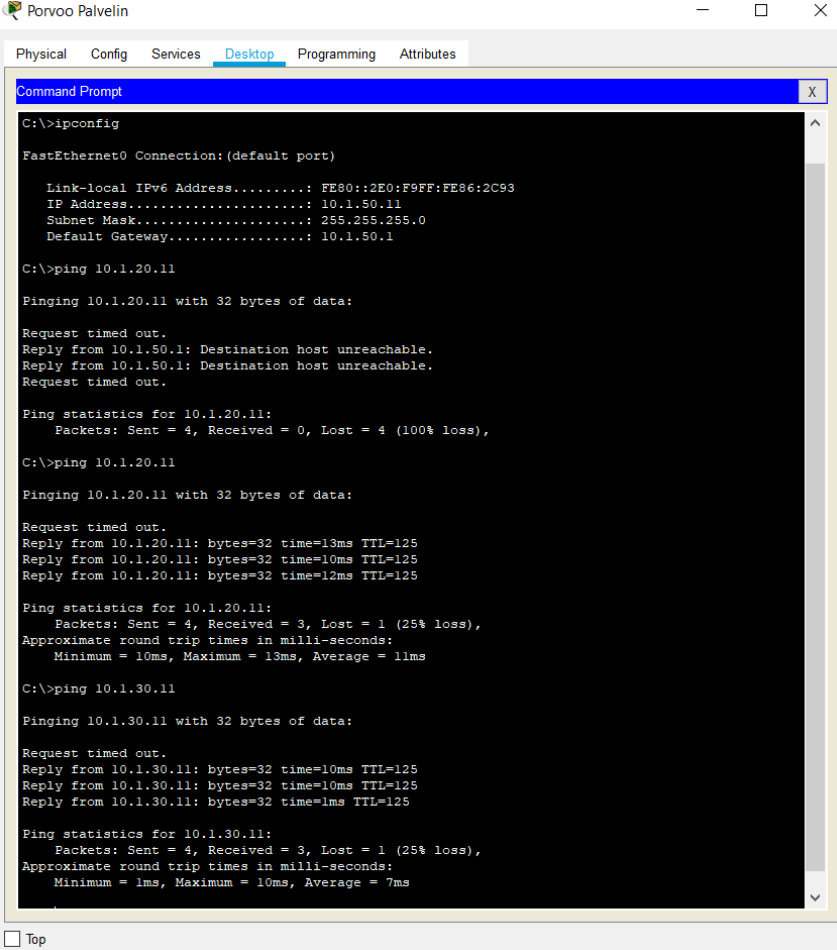
Kuva 8. Reititystaulu ja yhteyksien testaukset

Kuten kuvasta 8 selviää reititin tietää vain verkot, jotka siihen on suoraan liitetty. Ping-komento antamalla selviää, että yhteys onnistuu Helsingin reitittimen porttiin G0/0, mutta ei onnistu Haagan reitittimen porttiin G0/1.

Määritetään jokaiselle reitittimelle verkot, joita ne mainostavat, aktivoidaan OSPF ja määritetään reititin ID:t. Käytetään prosessi ID:nä numeroa yksi kaikissa reitittimissä, alue arvot käyttäen nollaa sekä reitin id:tä, jotka alkavat 1.1.1.1 Helsinki- ja loppuvat 6.6.6.6 Vierumäki-reitittimellä.

```
Helsinki(config)#router ospf 1
Helsinki(config-router)# router-id 1.1.1.1
Helsinki(config-router)# network 10.1.1.0 0.0.0.255 area 0
Helsinki(config-router)# network 10.1.2.0 0.0.0.255 area 0
```

Kun tämä on tehty kaikille reitittimille omilla asetuksillaan, konfiguroidaan eri verkkojen päätelaitteet käyttämään omassa verkossaan IP-osoitteita, joiden viimeinen oktetti on 11. Kokeillaan myös toimiiko liikenne eri verkkojen välillä.



The screenshot shows a Windows Command Prompt window titled 'Command Prompt' with a close button (X) in the top right corner. The window content displays the following text:

```
C:\>ipconfig

FastEthernet0 Connection: (default port)

    Link-local IPv6 Address . . . . . : FE80::2E0:F9FF:FE86:2C93
    IP Address . . . . . : 10.1.50.11
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.1.50.1

C:\>ping 10.1.20.11

Pinging 10.1.20.11 with 32 bytes of data:

Request timed out.
Reply from 10.1.50.1: Destination host unreachable.
Reply from 10.1.50.1: Destination host unreachable.
Request timed out.

Ping statistics for 10.1.20.11:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 10.1.20.11

Pinging 10.1.20.11 with 32 bytes of data:

Request timed out.
Reply from 10.1.20.11: bytes=32 time=13ms TTL=125
Reply from 10.1.20.11: bytes=32 time=10ms TTL=125
Reply from 10.1.20.11: bytes=32 time=12ms TTL=125

Ping statistics for 10.1.20.11:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 10ms, Maximum = 13ms, Average = 11ms

C:\>ping 10.1.30.11

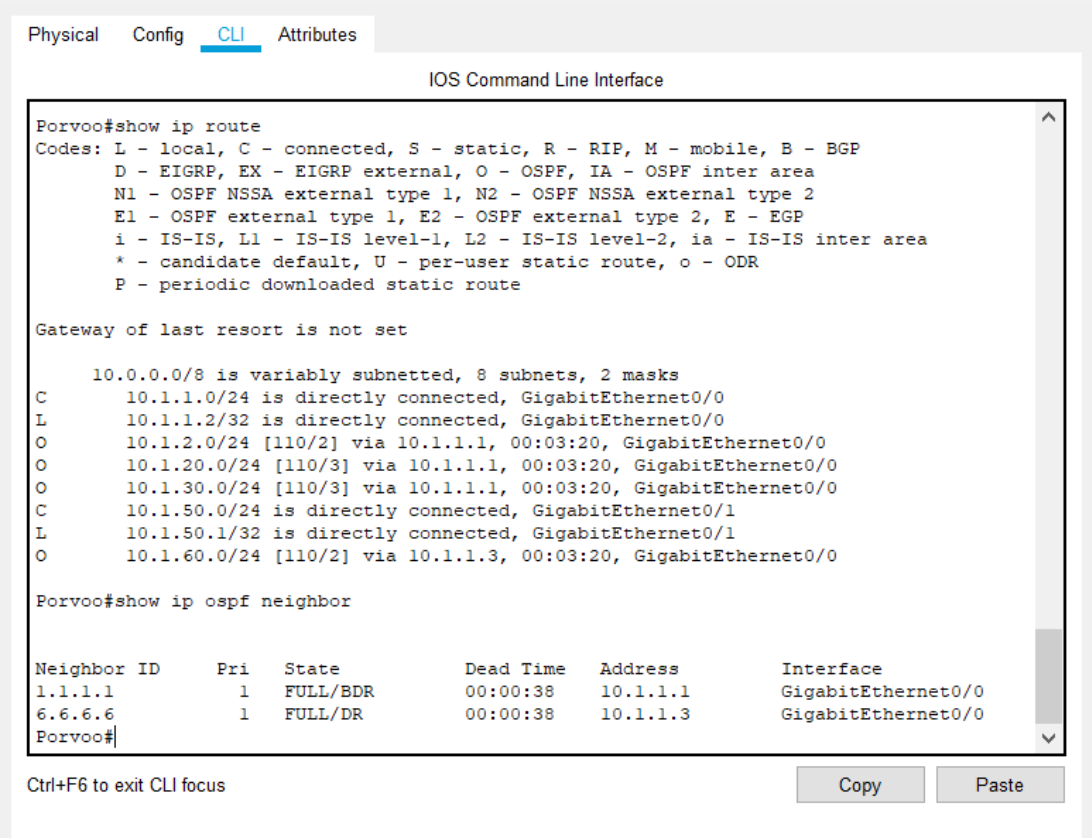
Pinging 10.1.30.11 with 32 bytes of data:

Request timed out.
Reply from 10.1.30.11: bytes=32 time=10ms TTL=125
Reply from 10.1.30.11: bytes=32 time=10ms TTL=125
Reply from 10.1.30.11: bytes=32 time=1ms TTL=125

Ping statistics for 10.1.30.11:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 10ms, Average = 7ms
```

Kuva 9. Verkkoasetukset ja yhteyden kokeilu

Kuvasta 9 selviää, että yhteydet onnistuvat Porvoon päätelaitteelta Haagan sekä Malmin päätelaitteisiin. Tarkistetaan vielä, miten Porvoon reitittimen reititystaulu on muuttunut sekä miltä OSPF naapuruudet näyttävät kuvan 10 avulla.



```
Porvoo#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

 10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C    10.1.1.0/24 is directly connected, GigabitEthernet0/0
L    10.1.1.2/32 is directly connected, GigabitEthernet0/0
O    10.1.2.0/24 [110/2] via 10.1.1.1, 00:03:20, GigabitEthernet0/0
O    10.1.20.0/24 [110/3] via 10.1.1.1, 00:03:20, GigabitEthernet0/0
O    10.1.30.0/24 [110/3] via 10.1.1.1, 00:03:20, GigabitEthernet0/0
C    10.1.50.0/24 is directly connected, GigabitEthernet0/1
L    10.1.50.1/32 is directly connected, GigabitEthernet0/1
O    10.1.60.0/24 [110/2] via 10.1.1.3, 00:03:20, GigabitEthernet0/0

Porvoo#show ip ospf neighbor

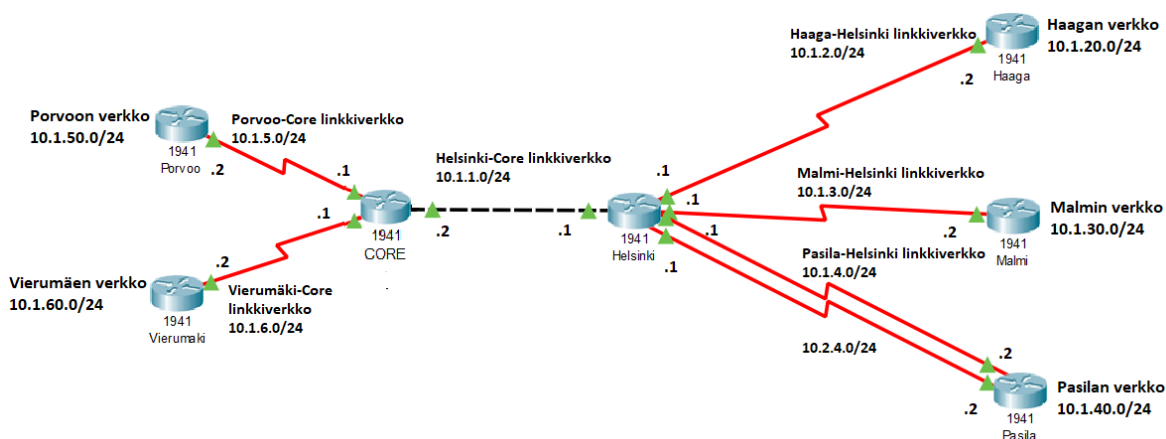
Neighbor ID      Pri   State           Dead Time   Address        Interface
1.1.1.1          1     FULL/BDR        00:00:38   10.1.1.1       GigabitEthernet0/0
6.6.6.6          1     FULL/DR         00:00:38   10.1.1.3       GigabitEthernet0/0
Porvoo#
```

Kuva 10. Reititystaulu sekä OSPF naapuruudet

Reitittäulu on kasvanut kahdesta reitistä kuuteen ja neljä on opittu OSPF:n avulla. Reitti Pasilan sisäverkkoon puuttuu, koska se jätettiin konfiguroimatta simuloidakseen, ettei reittiä opita ilman reititysprotokollaa. Nähdään myös, että suurimman reitin id:n omaava reititin Vierumäki on nimetty reititin (designated router) ja Helsinki on varanimetty reititin (backup designated router).

## 6.2 Skenaario EIGRP

Packet Tracer -ohjelmaan on asetettu verkkolaitteet hyvin samaan tapaan kuin ensimmäisessä skenaariossa alapuolella olevan kuvan 11 mukaan. Verkosta jätettiin päätelaitteet pois verraten ensimmäiseen skenaarioon, mutta näitä simuloitiin kuitenkin loopback-porttien avulla. Tämä tehtiin tuodakseen esille, että simulaatio on mahdollista toteuttaa myös tällä tavalla. Tästä skenaariosta jäi pois myös kytkimet, koska ne eivät tuo mitään lisäarvoa simulaatioon.

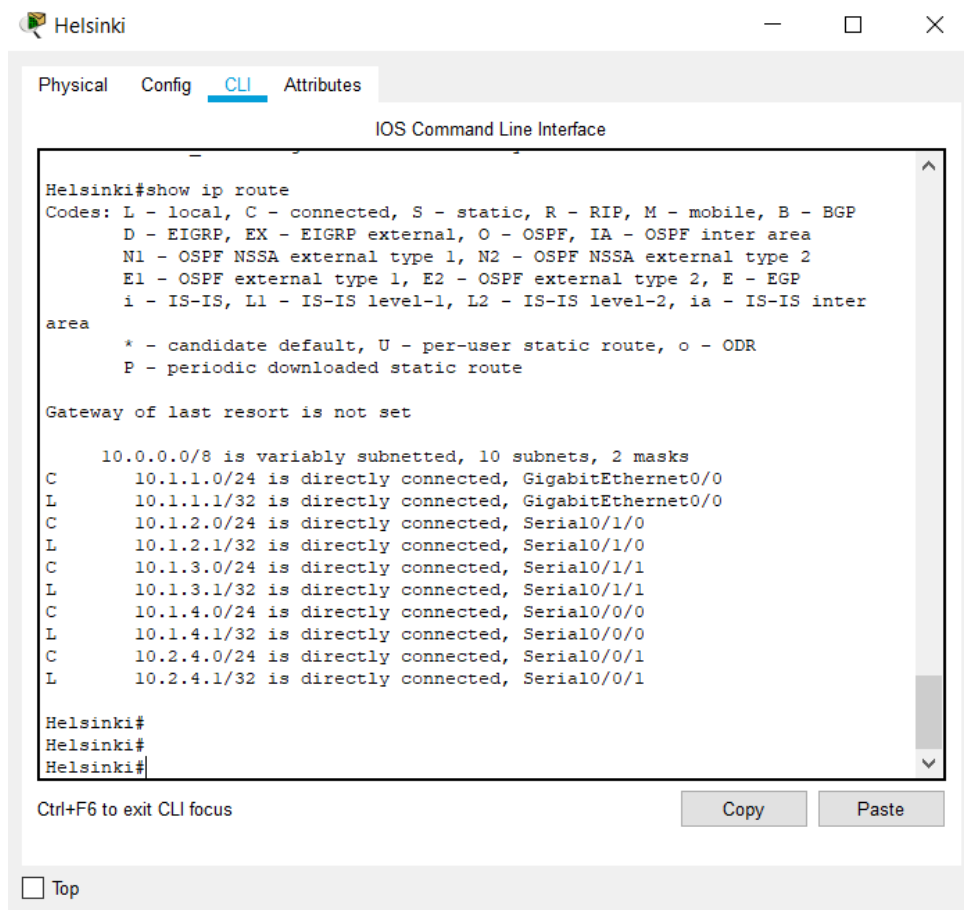


Kuva 11. Verkon topologia

Taulukko 6. Esimerkkiverkon reitittimien porttien IP-osoitteet ja aliverkkojen peitteet

Reititin	Portti	IP-osoite ja aliverkkojen peite
Helsinki	G0/0	10.1.1.1 255.255.255.0
	S0/1/0	10.1.2.1 255.255.255.0
	S0/1/1	10.1.3.1 255.255.255.0
	S0/0/0	10.1.4.1 255.255.255.0
	S0/0/1	10.2.4.1 255.255.255.0
Haaga	S0/0/0	10.1.2.2 255.255.255.0
	Lo0	10.1.20.1 255.255.255.0
Malmi	G0/0	10.1.2.3 255.255.255.0
	Lo0	10.1.30.1 255.255.255.0
Pasila	G0/0	10.1.2.4 255.255.255.0
	Lo0	10.1.40.1 255.255.255.0
Porvoo	G0/0	10.1.1.2 255.255.255.0
	Lo0	10.1.50.1 255.255.255.0
Vierumäki	G0/0	10.1.1.3 255.255.255.0
	Lo0	10.1.60.1 255.255.255.0

Laitteille on konfiguroitu yläpuolella olevien taulukon 6 ja kuvan 11 mukaiset IP-osoitteet porteille ja ne on nostettu ylös. Tarkistetaan, miltä Helsingin reitittimen reititystaulu tällä hetkellä näyttää.



Kuva 12. Reititystaulu skenaarion alussa

Kuvasta 12 näkyy, että reititin tietää vain verkot, jotka ovat siinä suoraan kiinni. Aktivoi-  
daan EIGRP ja määritetään mainostettavat verkot Helsingin reitittimelle.

```

Helsinki(config)#router eigrp 1
Helsinki(config-router)# network 10.1.2.0 0.0.0.255
Helsinki(config-router)# network 10.1.3.0 0.0.0.255
Helsinki(config-router)# network 10.1.4.0 0.0.0.255
Helsinki(config-router)# network 10.1.1.0 0.0.0.255
Helsinki(config-router)# network 10.2.4.0 0.0.0.255

```

Tehdään samat EIGRP aktivoinnit ja verkkojen mainostukset kaikilla reitittimillä. Sekä kon-  
figuroidaan tarvittavat loopback portit. Alla esimerkkinä Haagan reititin.

```

Haaga(config)#interface Loopback0
Haaga(config-if)# ip address 10.1.20.1 255.255.255.0
Haaga(config)#router eigrp 1
Haaga(config-router)#network 10.1.2.0 0.0.0.255
Haaga(config-router)#network 10.1.20.0 0.0.0.255
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 10.1.2.1 (Serial0/0/0) is
up: new adjacency

```

Kuten viimeisestä rivistä huomataan, naapuruussuhde muodostuu välittömästi. Kun reitittimet ovat konfiguroitu on hyvä tarkistaa yhteyksien toimivuus ja reititystaulu. Tehdään tämä Helsingin reitittimeltä.



The screenshot shows a network device's CLI window titled "Helsinki". The window has tabs for "Physical", "Config", "CLI", and "Attributes", with "CLI" selected. The main content area is titled "IOS Command Line Interface" and displays the output of the command "show ip route". The output shows a routing table with various entries, including directly connected networks and routes learned via EIGRP. Below the routing table, two ping tests are performed: one to 10.1.20.1 and another to 10.1.50.1, both showing a 100% success rate. At the bottom of the window, there are buttons for "Copy" and "Paste", and a "Top" button.

```
Helsinki#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 18 subnets, 2 masks
C    10.1.1.0/24 is directly connected, GigabitEthernet0/0
L    10.1.1.1/32 is directly connected, GigabitEthernet0/0
C    10.1.2.0/24 is directly connected, Serial0/1/0
L    10.1.2.1/32 is directly connected, Serial0/1/0
C    10.1.3.0/24 is directly connected, Serial0/1/1
L    10.1.3.1/32 is directly connected, Serial0/1/1
C    10.1.4.0/24 is directly connected, Serial0/0/0
L    10.1.4.1/32 is directly connected, Serial0/0/0
D    10.1.5.0/24 [90/2170112] via 10.1.1.2, 00:03:44, GigabitEthernet0/0
D    10.1.6.0/24 [90/2170112] via 10.1.1.2, 00:03:47, GigabitEthernet0/0
D    10.1.20.0/24 [90/2297856] via 10.1.2.2, 00:03:44, Serial0/1/0
D    10.1.30.0/24 [90/2297856] via 10.1.3.2, 00:03:44, Serial0/1/1
D    10.1.40.0/24 [90/2297856] via 10.2.4.2, 00:03:45, Serial0/0/1
    [90/2297856] via 10.1.4.2, 00:03:42, Serial0/0/0
D    10.1.50.0/24 [90/2298112] via 10.1.1.2, 00:03:44, GigabitEthernet0/0
D    10.1.60.0/24 [90/2298112] via 10.1.1.2, 00:03:47, GigabitEthernet0/0
D    10.1.200.0/24 [90/3072] via 10.1.1.2, 00:03:51, GigabitEthernet0/0
C    10.2.4.0/24 is directly connected, Serial0/0/1
L    10.2.4.1/32 is directly connected, Serial0/0/1

Helsinki#ping 10.1.20.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.20.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/5 ms

Helsinki#ping 10.1.50.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.50.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

Helsinki#
```

Kuva 13. Yhteyksien toimivuus ja reititystaulu

Kuten kuvasta 13 huomataan, niin kahdeksan uutta verkkoa on opittu EIGRP-protokollan avulla, ja yhteydet onnistuvat niin Haagan kuin Porvoon verkkoihin.

## 7 Pohdinta

EIGRP ja OSPF -protokollissa on hyvin paljon samaa, mitä selittää protokollien taustalla olevat syyt niiden kehitykseen. Konfiguraatiossa yhtenä merkittävänä erona pidän OSPF:n ominaisuutta käyttää alueita, joka piti konfiguroida, vaikka alueita olisi vain yksi. Tässä työssä käytettiin vain yhtä aluetta niin sen tuomat hyödyt eivät kuitenkaan tulleet esille. Suurimpana erona näen kuitenkin EIGRP:n AS:n ja OSPF:n reititin ID:n. Reititin ID:n eroavaisuuksilla ei ole toimivuuden kannalta eroa, mutta AS:ien on pakko täsmätä reitittimien välillä ja tämän kanssa ei saa tehdä virheitä.

Toiminnallisuuden puolesta mielestäni OSPF:n nimetty reititin oli selkeämpi kuin EIGRP:n seuraaja. Tähän varmasti vaikuttaa myös se, etten tässä työssä kovin syvällisesti tähän keskittynyt. OSPF on ollut koko sen elinkaaren ajan avoimen standardin reititysprotokolla, kun EIGRP on sitä ollut vasta vuodesta 2013. Tämä on vahvasti vaikuttanut protokollien levinneisyyteen ja sen myötä suosioon.

Cisco Packet Tracer -ohjelma soveltui näihin simulaatioihin erinomaisesti. Ainoana huonona puolena voidaan pitää, että kaikki komennot eivät toimineet. Esimerkkinä komento *show interface description*, joka oikealla laitteella näyttää reitittimen kaikkien porttien tilat ja kuvaukset.

Jos luotaisiin alusta asti kokonaan uusi ympäristö, niin tämän työn perusteella saaduilla tiedoilla valitsisin OSPF-protokollan.

## Lähteet

Albrightson, B., Boyle, J. & Garcia-Luna-Aceves J. J. EIGRP - a fast routing protocol based on distance vectors. Luettavissa: <http://vir.liu.se/~TDTS02/papers/eigrp.pdf>. Luettu 15.8.2020.

Britt, D., Davis C., Forrester J., Liu W., Matthews C., Parziale, L. & Rosselot N. 2006. TCP/IP tutorial and technical overview. IBM Redbooks. USA. Luettavissa: <https://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf>. Luettu 17.5.2020.

Hunt, C. 1997. TCP/IP Network Administration. Luettavissa: [http://web.deu.edu.tr/doc/oreily/networking/tcpip/ch07\\_05.htm](http://web.deu.edu.tr/doc/oreily/networking/tcpip/ch07_05.htm). Luettu: 26.4.2020.

Lammle, T. 2016. CCNA Routing and Switching Complete Study Guide. John Wiley & Sons, Inc. Indianapolis.

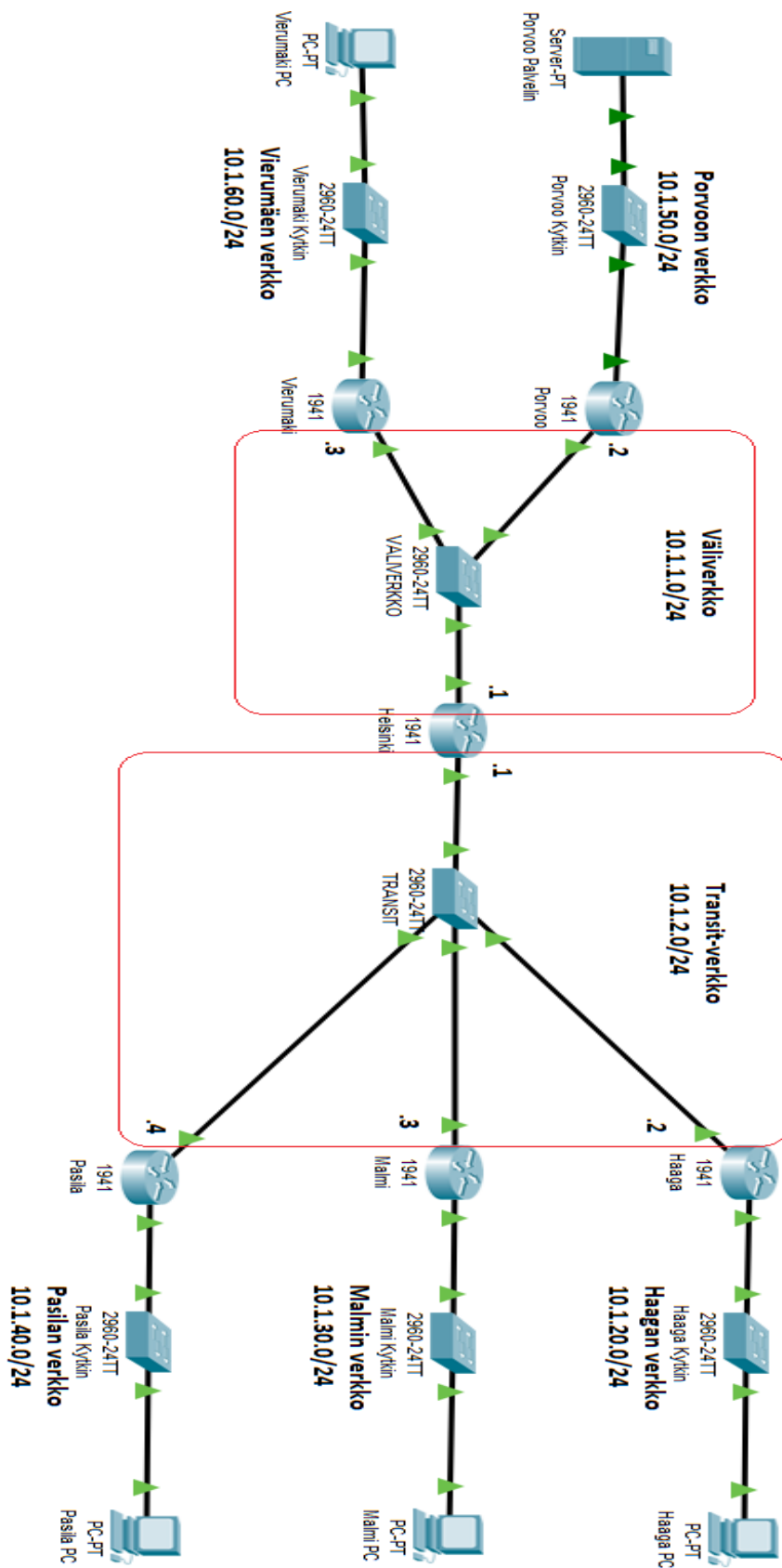
Moy, John T. 2000. OSPF: Anatomy of an Internet Routing Protocol. Addison-Wesley. New Jersey.

Odom, W. 2016. CCENT/CCNA Routing and Switching ICND1 100-105 Official Cert Guide. Cisco Systems, Inc. Indianapolis.

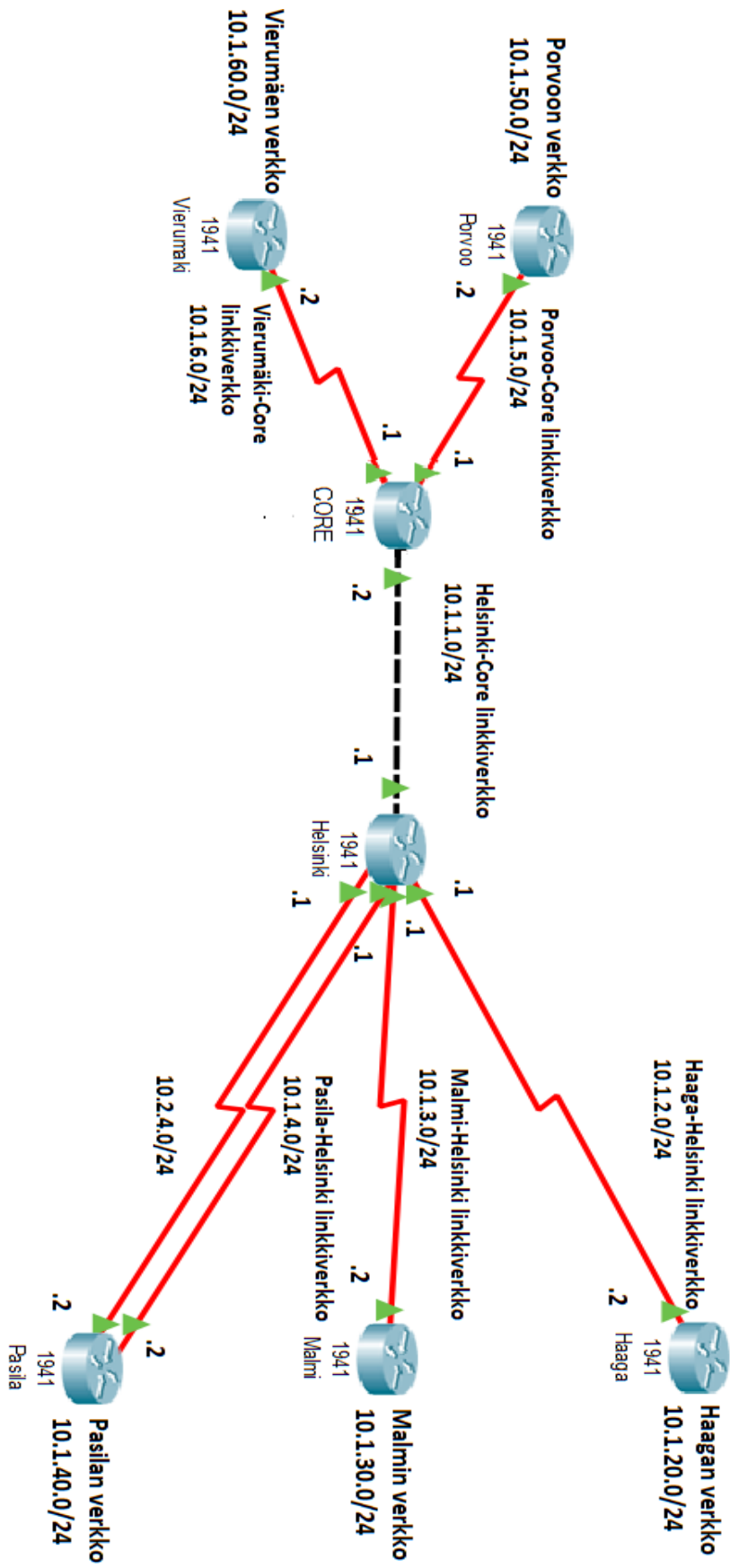
Odom, W. 2017. CCNA Routing and Switching ICND2 200-105 Official Cert Guide. Pearson Education, Inc. Indianapolis

# Liitteet

## Liite 1. OSPF verkkokuva



## Liite 2. EIGRP verkkokuva



### Liite 3. OSPF skenaarion Helsinki reitittimen koko konfiguraatio

```
Current configuration : 1005 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Helsinki
!
no ip cef
no ipv6 cef
!
license udi pid CISCO1941/K9 sn FTX152461IS-
!
spanning-tree mode pvst
!
interface GigabitEthernet0/0
description VALIVERKKO
ip address 10.1.1.1 255.255.255.0
duplex auto
speed auto
!
interface GigabitEthernet0/1
description TRANSIT
ip address 10.1.2.1 255.255.255.0
duplex auto
speed auto
!
interface Serial0/1/0
description MALMI
ip address 10.1.3.1 255.255.255.0
clock rate 2000000
!
interface Serial0/1/1
description PASILA
ip address 10.1.4.1 255.255.255.0
clock rate 2000000
!
interface Vlan1
no ip address
shutdown
!
router ospf 1
router-id 1.1.1.1
log-adjacency-changes
network 10.1.1.0 0.0.0.255 area 0
```

```
network 10.1.2.0 0.0.0.255 area 0
!  
ip classless  
!  
ip flow-export version 9  
!  
line con 0  
!  
line aux 0  
!  
line vty 0 4  
  login  
!  
end
```

#### Liite 4. OSPF skenaarion Porvoo reitittimen koko konfiguraatio

Current configuration : 814 bytes

```
!  
version 15.1  
no service timestamps log datetime msec  
no service timestamps debug datetime msec  
no service password-encryption  
!  
hostname Porvoo  
!  
no ip cef  
no ipv6 cef  
!  
license udi pid CISCO1941/K9 sn FTX15247XLL-  
!  
spanning-tree mode pvst  
!  
interface GigabitEthernet0/0  
description VALIVERKKO  
ip address 10.1.1.2 255.255.255.0  
duplex auto  
speed auto  
!  
interface GigabitEthernet0/1  
description PORVOO KAMPUS  
ip address 10.1.50.1 255.255.255.0  
duplex auto  
speed auto  
!  
interface Vlan1  
no ip address  
shutdown  
!  
router ospf 1  
router-id 5.5.5.5  
log-adjacency-changes  
network 10.1.1.0 0.0.0.255 area 0  
network 10.1.50.0 0.0.0.255 area 0  
!  
ip classless  
!  
ip flow-export version 9  
!  
line con 0  
!  
line aux 0  
!  
line vty 0 4
```

```
login  
!  
end
```

## Liite 5. EIGRP skenaarion Helsinki reitittimen koko konfiguraatio

```
Current configuration : 1232 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Helsinki
!
no ip cef
no ipv6 cef
!
license udi pid CISCO1941/K9 sn FTX1524C3W2-
!
spanning-tree mode pvst
!
interface GigabitEthernet0/0
description CORE
ip address 10.1.1.1 255.255.255.0
duplex auto
speed auto
!
interface GigabitEthernet0/1
no ip address
duplex auto
speed auto
shutdown
!
interface Serial0/0/0
description PASILA
ip address 10.1.4.1 255.255.255.0
clock rate 2000000
!
interface Serial0/0/1
description PASILA2
ip address 10.2.4.1 255.255.255.0
clock rate 2000000
!
interface Serial0/1/0
description HAAGA
ip address 10.1.2.1 255.255.255.0
clock rate 2000000
!
interface Serial0/1/1
description MALMI
ip address 10.1.3.1 255.255.255.0
clock rate 2000000
```

```
!  
interface Vlan1  
  no ip address  
  shutdown  
!  
router eigrp 1  
  network 10.1.2.0 0.0.0.255  
  network 10.1.3.0 0.0.0.255  
  network 10.1.4.0 0.0.0.255  
  network 10.1.1.0 0.0.0.255  
  network 10.2.4.0 0.0.0.255  
!  
ip classless  
ip route 0.0.0.0 0.0.0.0 10.1.1.2  
!  
ip flow-export version 9  
!  
line con 0  
!  
line aux 0  
!  
line vty 0 4  
  login  
!  
end
```

## Liite 6. EIGRP skenaarion Porvoo reitittimen koko konfiguraatio

Building configuration...

Current configuration : 1064 bytes

```
!  
version 15.1  
no service timestamps log datetime msec  
no service timestamps debug datetime msec  
no service password-encryption  
!  
hostname Porvoo  
!  
no ip cef  
no ipv6 cef  
!  
license udi pid CISCO1941/K9 sn FTX15243ENW-  
!  
spanning-tree mode pvst  
!  
interface Loopback0  
 ip address 10.1.50.1 255.255.255.0  
!  
interface GigabitEthernet0/0  
 no ip address  
 duplex auto  
 speed auto  
 shutdown  
!  
interface GigabitEthernet0/1  
 no ip address  
 duplex auto  
 speed auto  
 shutdown  
!  
interface Serial0/0/0  
 description CORE  
 ip address 10.1.5.2 255.255.255.0  
 clock rate 2000000  
!  
interface Serial0/0/1  
 no ip address  
 clock rate 2000000  
 shutdown  
!  
interface Serial0/1/0  
 no ip address  
 clock rate 2000000  
 shutdown
```

```
!  
interface Serial0/1/1  
  no ip address  
  clock rate 2000000  
  shutdown  
!  
interface Vlan1  
  no ip address  
  shutdown  
!  
router eigrp 1  
  network 10.1.5.0 0.0.0.255  
  network 10.1.50.0 0.0.0.255  
!  
ip classless  
!  
ip flow-export version 9  
!  
no cdp run  
!  
line con 0  
!  
line aux 0  
!  
line vty 0 4  
  login  
!  
end
```

