TEKNIIKKA JA LIIKENNE

Tietotekniikka

Tietoverkot

INSINÖÖRITYÖ

F5-SISÄLTÖKYTKIMIEN KÄYTTÖ REDUNDANTTISEN PALVELUYMPÄRISTÖN RAKENTAMISEEN

Työn tekijä: Jukka Laitinen
Työn ohjaajat: Janne Salonen

Työ hyväksytty: 27. 4. 2009

Janne Salonen
Yliopettaja

ALKULAUSE

Tämä insinöörityö tehtiin Metropolia Ammattikorkeakoululle. Kiitokset Metropolian Bulevardin toimipisteen tietotekniikkaosastolle mahdollisuudesta tehdä työ ja käytännön avusta työn loppuun saattamisessa.

Helsingissä 27.04.2009

Jukka Laitinen

Metropolia

TIIVISTELMÄ

Työn tekijä: Jukka Laitinen

Työn nimi: F5-Sisältökytkimien käyttö redundanttisen palveluympäristön rakentamiseen

Päivämäärä: 27.4.2009 **Sivumäärä:** 26 s. + 1 liite

Koulutusohjelma: Tietotekniikka

Suuntautumisvaihtoehto: Tietoverkot

Työn ohjaaja: Yliopettaja Janne Salonen

Työn ohjaaja: Yliopettaja Janne Salonen

Työn tarkoituksena oli ottaa käyttöön kaksi F5-sisältökytkintä, jonka avulla saatiin redundanttinen ympäristö ja palvelimet vikasietoisiksi. Työn tavoitteena oli tutkia kytkimien mahdollisuutta käyttää tuotantoympäristössä ja saada aikaiseksi toimiva ja vikasietoinen verkko. Työssä rakennettiin myös erillinen vikasietoinen verkko, joka toteutettiin HSRP-protokollaa hyödyntäen.

Teoriaosuudessa työssä keskitytään sisältökytkentään käsitteenä, kuormanjakoon HSRP-protokollaan ja perustietoliikennekäsitteisiin. Käytännön osuudessa rakennetaan vikasietoinen sisäverkko ja F5-kytkimillä varmistettu palvelinverkko, jonka toimivuus haluttiin selvittää.

Lopputuloksena saatiin toimiva vikasietoinen verkko ja varmuus palvelimien toiminnasta käyttäen sisältökytkimiä palvelimien suojaamiseen. Työtä on helppo käyttää pohjana jatkossa, jos laitteet halutaan ottaa tuotantokäytöön.

Avainsanat: F5, sisältökytkentä, sisältökytkin, HSRP, kuormantasaus

Metropolia

ABSTRACT

Name: Jukka Laitinen

Title: Usage of F5-Content Switches In Redundant Service Environment

Date: 27 April 2009 **Number of pages:** 26 + 1 appendix

Department: Information Technology **Study Programme:** Data Networks

Instructor: Janne Salonen, Principal lecturer

Supervisor: Janne Salonen, Principal lecturer

The purpose of the thesis was to utilize two F5-content switches to build a redundant service environment. The goal was to study the possibility of usage of devices in real production and to build a functional and redundant network. Another redundant network was also built using HSRP-protocol.

Content switching, load balancing, HSRP-protocol and basic telecommunications terms are introduced in theoretical part of the study. The redundant network and service environment secured with F5-content switches are introduced in the practical part. This was the main goal of the study.

As a result, services are secured with F5-switches and the redundant network was built. The study can be utilized for further studies if there is need to take devices in use.

Keywords: F5, content switching, content switch, HSRP, load balancing

SISÄLLYS

1 JOHDANTO

Tänä päivänä jokainen yritys mainostaa ja tarjoaa palvelujaan verkossa ja lähes jokaisella yrityksellä on omat palvelunsa ja resurssinsa verkossa. Erilaiset verkkoresurssit ja palvelut ovat tulleet yhä tärkeämmäksi osaksi yrityksen toimintaa, ja niiden saavutettavuus on siksi tärkeää. Esimerkiksi tietokantapalvelimien toimimattomuus vaikeuttaa tietojen saantia ja tallentamista. Siksi palvelimet olisi hyvä suojata vikatilanteelta kahdentamalla palvelut. Jos yrityksen verkkopalvelut eivät toimi, yritys voi pahimmassa tapauksessa menettää jopa asiakkaita tai kauppoja. Yrityksen sisäiset resurssit ja palvelut ja niiden toiminta on aivan yhtä tärkeä pitää kunnossa, koska kaikki tieto on nykyään verkkoon tallennettu. Tässä työssä rakennetaan ratkaisu, jolla pyritään saavuttamaan vikasietoinen verkko ja verkossa oleville palvelimille lisää turvaa ottamalla kuormantasaus käyttöön.

Työssä käytetään HSRP-verkkoa, joka luo vikasietoisen verkon työasemille. Verkossa on useita reittejä samaan paikkaan, joten ei haittaa vaikka jokin verkon reitittimistä vioittuisi, koska muut reitittimet hoitavat reitityksen. HSRP-verkko helpottaa myös verkon ylläpitoa ja näin vähentää konfiguroinnin määrää. HSRP-verkon lisäksi työssä otetaan käyttöön kaksi F5-sisältökytkintä. Näiden kytkimien avulla pystytään saavuttamaan kuormantasaus verkon palvelimille. Lisäksi sisältökytkimet mahdollistavat sisältökytkennän, jolla pystytään optimoimaan verkon toimintaa. Nämä kytkimet mahdollistavat usean palvelimen käytön samanaikaisesti ja kytkimet jakavat kuorman tasaisesti usealle palvelimelle. Kytkimet muodostavat redundanttisen parin, joten toisen laitteen vikaantuminen ei aiheuta katkosia palvelujen tai resurssien toimintaan. Toinen kytkimistä toimii aina aktiivisessa tilassa ja toinen odottaa mahdollista vikatilannetta passiivisessa tilassa.

Työn alussa tutustutaan protokolliin, TCP- ja IP-toiminnallisuuteen sekä OSI-malliin, joka kuvaa tiedonsiirtoprotokollien yhdistelmää. Sisältökytkentä osuudessa kerrotaan sisältökytkimistä ja verkkokerroksen perusteella tehtävästä sisältökytkennästä. Seuraavassa kappaleessa tutustutaan kuormantasauksen ja erilaisiin kuormantasausmenetelmiin. Kuormantasausta tarkastellaan sekä verkon että palvelinten kannalta. HSRP-kappaleessa tarkistellaan HSRP-protokollan toimintaa ja sen tuomia

etuja. Itse työosuudessa rakennetaan ja konfiguroidaan läpikäytyjen asioiden perusteella rakennettu verkko ja testataan sen toiminnallisuutta erilaisilla simuloinneilla vikatilanteista. Työ rakennettiin Metropolia Ammattikorkeakoulun tietotekniikka osaston laboratoriotiloissa ja käytetään hyväksi koulun tarjoamia resursseja.

2 TCP/IP

TCP/IP on kauan käytössä ollut tietoliikenneprotokollien yhdistelmä, jota käytetään Internet-liikennöinnissä. IP-protokolla toimii verkkokerroksella ja on tarkoitettu pakettien reitittämiseen verkossa ja päätelaitteiden osoitteistamiseen. IP-protokollan päällä kulkee muiden kerroksien protokollia, kuten kuljetuskerroksella toimiva TCP-protokolla. TCP-protokolla on vastuussa laitteiden välisestä tiedonsiirtoyhteydestä, pakettien uudelleenjärjestämisestä ja hukkuneiden pakettien uudelleen lähetyksestä. Protokolla sisältää myös muita protokollia, mutta koska suurin osa verkkoliikenteestä kulkee TCP/IP liikenteenä, siitä käytetään nimitystä TCP/IP. [1.]

2.1 OSI-malli

OSI-malli (Open System Interconnection reference model) kuvaa tietoliikenneprotokollien mallia seitsemässä kerroksessa. Jokainen protokolla käyttää yhtä alemman kerroksen protokollaa ja tarjoaa palveluja ylemmän kerroksen protokollille. OSI-malli on kehitetty jo 1980-luvun alussa ja on voimassa oleva kansainvälinen standardi. OSI-malli koostuu siis seitsemästä kerroksesta ja jokaiseen kerrokseen on kuvattu tiettyjä protokollia. Kuvasta 1 selviävät OSI-mallin fyysiset kerrokset ja mitä ne pitävät sisällään. [2.]

Kuva 1:OSI-malli [2.]

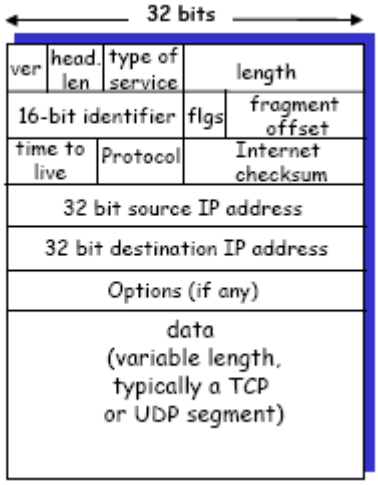
Ensimmäisellä kerroksella eli fyysisellä kerroksella, käsitellään sähköimpulssejä, valoa tai muuta fyysistä tekniikkaa. Toisella eli siirtoyhteyskerroksella hoidetaan paikallisessa lähiverkossa olevien laitteiden välillä tapahtuvan liikenteen. Kolmannella eli verkkokerroksella tapahtuu lähiverkon ulkopuolella tapahtuvan liikkeen reititys, eli se miten laite löytää haluamansa koneen Internetistä. Neljännellä eli kuljetuskerroksella pidetään huoli siitä, että paketti saapuu perille oikeassa järjestyksessä. Vuonhallinta kuuluu kuljetuskerroksen tehtäviin. Viidennellä eli istuntokerroksella huolehditaan useiden yhdessä yhteydessä kulkevien istuntojen multipleksoinnista. Kuudennella eli esitystapakerroksella huolehditaan, että tieto tulee käyttäjälle sopivassa muodossa, kuten vaikka Unicode-teksti muunnetaan kyriilisiksi merkeiksi. Seitsemäs eli sovelluskerros on se kerros, joka näkyy käyttäjälle sovelluksien muodossa. [2.]

2.2 IP

IP (Internet Protocol) toimii OSI-mallissa verkkokerroksella ja vastaa siitä, että IP-paketit löytävät tiensä perille pakettikytkentäisessä verkossa. IP-protokolla voisi myös kutsua Internetin ytimeksi, koska se on ainoa asia, mikä yhdistää kaikkia laitteita, jotka ovat Internettiin liitettyjä. Koska protokolla on yhteydetön, ei ole väliä, mitä kautta paketti kulkee laitteeltä toiselle tai tulee takaisin. [2.]

Nykysin käytössä olevassa IP-versiossa, IPv4, IP-paketin pituus on 32 bittiä. IP-osoite muodostuu neljästä kahdeksan bitin ryhmästä. Nämä kahdeksan bitin ryhmät muutetaan desimaaliluvuiksi, yksi ryhmä voi saada arvokseen luvun väliltä 0-255. IP-osoite voisi olla vaikka muotoa 192.168.98.1. Kehitteillä oleva uusi IP-versio, IPv6, kattaa moninkertaisen määrän osoitteita verrattuna IPv4:ään. Siinä IP-paketin pituus on 128 bittiä. [3.]

IP-protokollan kehyksen pituus on 32 bittiä, kuten kuvasta 2 selviää. Kaikki kentät ovat tärkeitä, mutta oleellisimmat ovat lähettäjän IP-osoite, vastaanottajan IP-osoite ja se mitä kehys kuljettaa, esimerkiksi TCP. Kuvassa 2 on esitetty kehys kokonaisuudessaan. Ensimmäisellä tasolla on esitetty protokollan versio numero, otsikon pituus, palvelun tyyppi ja pituuskentät. Toisella tasolta löytyy tunniste, mahdolliset liput ja lohkon sijaintikentät. Kolmannelta tasolta löytyy paketin elinaika, protokolla ja tarkistus kentät. Neljäs taso on lähdeosoite ja viides kohde osoitekenttä. Kuudes kenttä on valinnainen. Tähän kenttään voidaan määritellä turvallisuuteen tai reitittämiseen tarkoitettuja tietoja. Seitsemäs kenttä sisältää itse datan.



Kuva 2: IP:n protokollakehys [3.]

2.3 TCP

TCP (Transfer Control Protocol) on tietoliikenneprotokolla, jonka avulla luodaan yhteys Internetissä olevien koneiden välille. TCP on luotettava protokolla, koska se pitää huolen siitä, paketit saapuvat perille oikeassa järjestyksessä, ja jos paketteja on jäänyt pois lähetyksestä, se lähettää ne uudelleen. TCP pyrkii siirtämään paketteja mahdollisimman tehokkaasti

lähettämällä viestejä maksimi määrän, mitä vastaanottaja ja siirtotie sallivat. Protokollassa olevat menetelmät, vuonohjaus (flow control) ja ruuhkanhallinta (congestion control), rajoittavat liian suurien tietomäärien lähettämisen. TCP toimii OSI-mallissa kuljetuskerroksella. [4.]

TCP-protokolla päällä voi kuljettaa useita ylemmän tason protokollia jotka valitaan porttinumeron perusteella, esimerkiksi ftp-palvelua käytetään selaimella laittamalla osoiteriville IP-osoitteen minne ollaan menossa ja perään kaksoispisteellä erotettuna numeron 21. Palveluista esimerkkinä seuraavat, käytettävä porttinumero perässä:

- HTTP (www-sivut, portti 80)
- SMTP (sähköpostin välitys, portti 25)
- Telnet (pääteyhteys verkon yli, portti 23)
- SSH (salattu pääteyhteys verkon yli ja tiedoston siirto, portti 22)
- FTP (tiedoston siirto verkon yli, portti 21).

3 SISÄLTÖKYTKENTÄ

Verkkosivustoista on tullut monimutkaisia järjestelmiä, jotka koostuvat erikoistuneista tekniikoista. Järjestelmä voi koostua palomuurista, reitittimestä, kerroksen 2 tai 3 kytkimistä, kuormantasaus laitteista, välityspalveimista ja itse web-palvelimista. Tämän kaiken voi rakentaa itse tai vuokrata osia palveluista palveluntarjoajilta, mutta se kuinka kaikki tämä on rakennettu, vaikuttaa suoraan sivuston toimivuuteen. Jos sivustolle halutaan saada suuri käyttäjäkunta, on palvelun silloin toimittava moitteettomasti; käyttäjä menettää kärsivällisyytensä, jos sivujen latautuminen vie liian kauan aikaa. [5.]

3.1 Sisältökytkin

Yksi tapa parantaa suorituskykyä on liikenteen kontrolloiminen. Sisältökytkimet pystyvät tarjoamaan parhaimman tavan hallita sisääntulevaa liikennettä. Tarkistelemalla HTTP-kehystä, sisältökytkimet pystyvät

itsenäisesti tekemään päätöksiä kuormantasauksesta ja siitä, mistä yksittäiset sivut tai kuvat latautuvat. Tämän tason liikenteen ohjaus on käytännöllinen, jos tietyt palvelimet ovat optimoitu erityisille palveluille, kuten kuvien jakamiselle, SSL-istunnoille tai tietokantaliikenteelle. [5.]

Suorityskyky, skaalautuvuus ja toipumiskyky ovat syitä, minkä vuoksi kannattaa harkita sisältökytkimen hankkimista. Jos on olemassa vain yksi palvelin, josta tarjotaan kaikki palvelut, sen toimivuus ja suorituskyky ei välttämättä riitä tarjoamaan vaadittavaa tasoa, eikä se myöskään ole vikasietoinen. Siksi tarvitaan useampia palvelimia, jotka kaikki pystyvät tarjoamaan saman sisällön ja palvelut käyttäjille, mutta näkyvät käyttäjälle vain yhtenä laitteena. Käyttäjät näkevät vain virtuaalisen osoitteen johon he ottavat yhteyttä ja joka näkyy DNS-palvelimelta ulospäin verkkoon, mutta jokaisella palvelimella on oma osoite määriteltynä, joka on ainoastaan sisältökytkimen tiedossa. Tämän vuoksi, jos yksi palvelimista kaatuu, sisältökytkin ei lähetä kyseiselle palvelimelle liikennettä eikä käyttäjät havaitse palvelimen rikkoontumista. Myös uuden palvelimen lisäys on mahdollista helposti; ei tarvitse muuta, kuin määrittää sisältökytkimeen, kuinka jakaa data käyttäjille. [6.]

3.2 Layer 4 / Layer 7 -kytkentä

OSI-mallin seitsemännellä kerroksella tapahtuva sisällökytkentä, esimerkiksi URL:ään perustuva kytkentä, tarjoaa paremman tavan hallita palvelin pohjaisia ohjelmia ja parantaa verkkopalveluiden luotettavuutta sekä suorituskykyä. Sisältökytkimet optimoivat liikennettä käyttäen URL-osoitteita IP-osoitteiden lisäksi tehdessään kytkentä päätöksiä. URL:n lisäksi myös evästeet voivat parantaa hallittavuutta ja joustavuutta verkkoliikenteen kontrolloinnissa. URL-pohjainen kytkentä mahdollistaa tehokkaamman palvelin farmin rakentamisen. URL:ään perustuvat kytkimet voivat ohjata kyselyt suoraan eteenpäin sisällön perusteella palvelimelle, esimerkiksi kuva tai video, joka on optimoitu siihen tarkoitukseen. Kytkimet voivat myös ohjata kyselyjä palvelimille jonka sisältö on muuttuvaa, esimerkiksi live-palvelimille, tai palvelimille jossa tieto on pysyvää, kuten web-cache-palvelimiin. Kuormanjaon vuoksi palvelinten sisältö on kopioitu toisiin palvelimiin palvelin farmeissa. [7.]

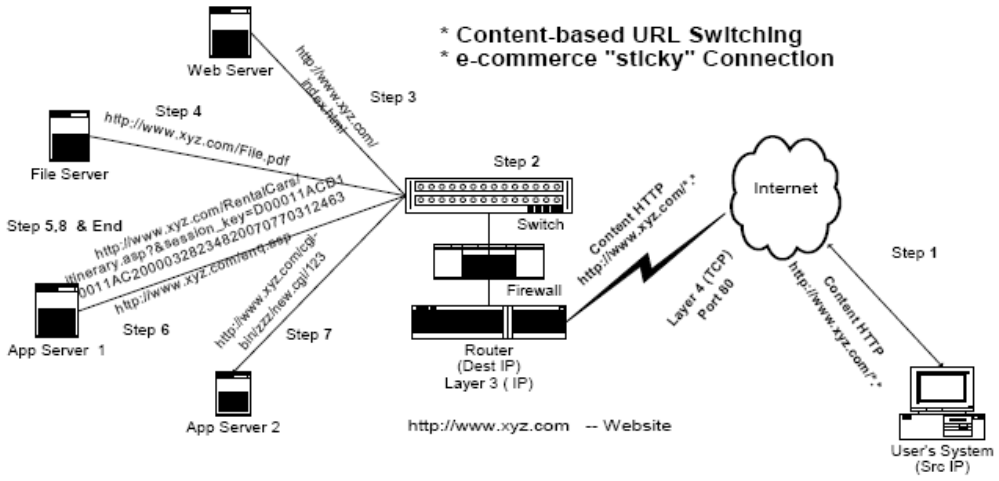
Sisältöön perustuvan kytkennän käyttö on kasvamassa tärkeämmäksi tekijäksi samalla kun palvelinfarmit leviävät maailmalla. Laitteet, jotka tukevat vain neljännen kerroksen palveluja, käyttävät TCP:n ja UDP:n kuljetuskerros informaatiota, esimerkiksi porttinumeroita, lähettääkseen paketin eteenpäin. Sisältöön perustuvassa kytkennässä voidaan käyttää sovelluskerroksen informaatiota lähetykseen, esimerkiksi tietosisältöä. Esimerkiksi kuljetuskerroksella toimiva kytkin saattaa ohjata kaiken liikenteen HTTP-portille, johonkin tiettyyn porttiin kytkimessä, johon web-cache-palvelin on yhdistetty. Tämän kaltaisella ratkaisulla näkymättömään web-cachingiin. Sivistyneemmät kuljetuskerroksen kytkimet saattavat tarjota lisäominaisuuksia. Kuten verkko-osoitten muunnoksen (NAT), kuormanjaon palvelinten välillä, vikasietoisuutta ja konfiguroitavan palvelunlaatu ominaisuuden erilaiselle liikenteelle. Perinteiset kuljetuskerroksen kytkimet, jotka tarjoavat mahdollisuuden kuormanjakoon ja näkymättömän välimuistin käyttämiseen, suorittavat TCP-liikenteen uudelleenohjausta uudelleenohjaamalla ensimmäisen SYN-paketin asiakkaalta valittuun kohteeseen ja uudelleenohjaamalla yhteyden aikana kaikki seuraavat paketit samaan kohteeseen. Tämän onnistumiseksi kytkimen ei tarvitse muuta kuin etsiä IP- ja TCP-otsikosta IP-osoitteen ja porttinumeron. Perinteinen kuljetuskerroksen kytkin ohjaa liikennettä tarkistelemalla SYN-pakettia. TCP-liikenteen uudelleenohjaaminen sisällön tai sovelluskerroksen informaation perusteella ei ole helppoa. TCP-tapahtumille sovelluskerroksen informaatio ei ole saatavilla ennen kuin TCP-yhteyden aloitus vaihe on suoritettu. [7.]

Sisältökytkentä viittaa kytkimen, joka on sijoitettu asiakkaiden tai palvelimien eteen, kykyyn uudelleenohjata HTTP-pyynnöt palvelimille perustuen URL:ään, jonka asiakas on määritellyt. Kun asiakas kirjoittaa selaimen osoiteriville URL-osoitteen, selain lähettää GET-pyynnön, joka sisältää URL:n ja muita HTTP-otsikon tietoja. Kytkin, joka on sijoitettu reitille asiakkaalta ja palvelimelle, keskeyttää GET-pyynnön ja tekee päätöksen, mihin palvelimelle pyyntö ohjataan. Asiakkaan TCP-yhteyspyynnöt päätetään kytkimelle ja TCP-yhteys pitää olla muodostettu ennen kuin mitään sovelluskerroksen informaatiota pystytään vastaanottamaan. Kun sovelluskerroksen informaatio on vastaanotettu, se järjestetään, jotta pystytään tekemään päätös mille palvelimelle lähetetään ennen kuin pyyntö uudelleenohjataan. Toinen TCP-yhteys muodostetaan kytkimeltä palvelimelle ja asiakkaan pyynnöt palvelimelle muodostuvat tämän yhteyden

läpi. Vastauspyynnöt lähetetään palvelimelta kytkimelle jälkimmäistä yhteyttä pitkin ja kulkee kytkimen läpi ensimmäistä yhteyttä pitkin asiakkaalle. Kaikki nämä ovat näkymättömiä asiakkaalle. [7.]

Normaalisti, kun asiakas tekee HTTP-kyselyä, TCP-yhteys muodostuu palvelimeen. Asiakas lähettää informaatiota liittyen kysyttyvään objektiin osana GET-kyselyä. Palvelin jäsentää tämän kyselyn ja palauttaa objektin asiakkaalle. Kun sisältökytkin liitetään tähän olettaen, että kohteen portti on 80, se näkymättömästi tulkkaa yhteyskyselyn paketit asiakkaalta ja lähettää ne sovelluskerroksen välityspalvelimille, joka ymmärtää HTTP-protokollaa. TCP-yhteys on nyt muodostettu asiakkaan ja palvelimen välillä. Välityspalvelin naamioi itsentä alkuperäiseksi palvelimeksi, jolle asiakas kyselyn teki. Asiakas lähettää GET-kyselyn välityspalvelimelle ja valityspalvelin päättää kohteen lähetetyn kyselyn perusteella. Tämän jälkeen välityspalvelin muodostaa TCP-yhteyden kohteen kanssa, välittää GET-kyselyn tähän yhteyteen, vastaanottaa palautteen ja siirtaa sen takaisin asiakkaalle. [7.]

Kuvassa 3 on kuvattu periaate, kuinka kytkentä tapahtuu. Kuvasta selviää, kuinka erilaisen sisällön sisältävät HTTP-kyselyt ohjataan eri palvelimille. Käyttäjä kirjoittaa selaimen osoite kenttään xyz.com. Asiakas yhdistetään palvelimelle, ja pyyntö kytketään web-palvelimelle. Asiakas tekee pyynnön esitteestä ja pyyntö ohjataan oikealle palvelimelle. Palvelin tarjoaa asiakkaalle halutun tiedoston ja siirtää tiedoston asiakkaalle. Siirto suoritetaan loppuun ja istunto suljetaan.



Kuva 3: Sisältöön perustuva kytkentä (Layer 7) [7.]

Sisältöön perustuvassa kytkennässä kytkimet käyttävät informaatiota, joka löytyy paketin hyötykuormasta, jotta pystyttäisiin käyttämään vieläkin hienostuneempia tapoja tehdä kytkentää. Esimerkiksi URL-osoitteesta, joka löytyy HTTP GET -kyselyssä, pystytään tutkimalla päättelemään, onko kyseessä vaikka kuva. Jos kyseessä on kuva, kaikki tämän TCP-yhteyden paketit, jotka liittyvät tähän kyselyyn, voidaan kytkeä palvelimelle, joka on optimoitu kuvien julkaisemiseen. Välimuistin tehokkuutta pystytään parantamaan jäsentämällä HTTP-kyselyiden sisältö jo kytkimellä. Kyselyt, jotka eivät sisällä välimuistissa olevaa tietoa, esimerkiksi CGI-skriptit, lähetetään web-palvelimelle välimuistipalvelimen sijaan, näin eliminoidaan välimuistipalvelimelle menevä turha kuorma. [7.]

Sisältökytkimen tehtävä on sidottu yleensä tiettyihin sovelluksiin, joten kytkimen mukauduttava niiden mukana ja siihen, miten sovelluksia käytetään. Kyky suorittaa useita ohjelmallisia ja ehdollisia luokitus jaksoja, on äärimmäisen tärkeää. Monimutkaisempien ja nopeasti kehittyvien sisältökytkimien tarpeen tuloksena uusia kytkinarkkitehtuureja on syntynyt ja niiden mukana tehokkaat prosessorit ovat löytäneet tiensä kytkimiin. Joissain uusissa kytkimissä on jopa oma dedikoitu prosessori fyysisille porteille, jotta pakettien manipulointi saataisiin tehokkaammaksi, oma prosessori ylläpitoon ja konfigurointiin ja lukuisia muita komponentteja, joilla kytkimen toiminteita saadaan nopeutettua. [7.]

4 KUORMANJAKO

Kuormanjako ei ole uusi konsepti, kun puhutaan palvelimista tai verkoista. Useat tuotteet suorittavat erilaisia tehtäviä liittyen kuormanjakoon. Esimerkiksi reitittimet ohjaavat liikennettä eri reittejä pitkin samaan kohteeseen. Palvelin kuormanjakajat toisaalta taas jakavat liikenteen eri palvelimien kesken. Alussa kuormanjakajat tekivät yksinkertaista kuormanjakoa, mutta nykyään niillä on useita tehtäviä, kuten kuormanjako, traffic engineering ja älykäs liikenteen kytkentä. Koska kuormanjakajat ovat useiten sijoitettu palvelimien eteen, ne estävät väärinkäyttöä ja parantavat turvallisuutta. Kuormanjako perustuu IP-pakettiin tai sovelluksen tekemään kyselyyn ja sen perusteella pystytään tekemään päätöksiä mihin paketti tai

kysely ohjataan, palvelimelle, palomuuriin tai vaikka välimuistipalvelimelle. [8, s. 1.]

Kuormantasaajan suurimmat käyttökohteet ovat verkot ja palvelimet. Intran ja Internetin käyttö, palvelimen ja työaseman yhdistäminen toisiinsa verkon kautta tai asiakkaan tarpeet ovat totuttaneet ihmiset tiettyyn palvelun laatuun. Verkko ei voi olla alhaalla tai sovellukset eivät saa toimia hitaasti, koska se saattaa lamauttaa koko liiketoiminnan. Kaupallisen verkkosivuston rakentaminen vaatii useita komponentteja, joihin pitää kiinnittää huomiota: palomuurit, reitittimet, web-palvelimet, tietokantapalvelimet ja kytkimet. Palvelimien lisääntyminen useille sovelluksille on luonut tietokeskuksia, jotka ovat täyttyneet palvelinfarmeista. Monimutkaisuus, haasteet skaalautuvuudessa, hallittavuus ja palvelinfarmien tavoitettavuus ovat yksi syy minkä vuoksi älykästä kytkentää tarvitaan. Skaalautuvuus ja korkea tavoitettavuus on taattava kaikille komponenteille ensimmäisestä reitittimestä, joka yhdistää Internettiin, takimmaiseen tietokantapalvelimeen. Kuormanjakajat ovat tehokkaita laitteita tähän tarkoitukseen. [8, s. 2.]

4.1 Kuormanjako ja palvelimet

Palvelimien määrä on kasvanut yrityksissä ja palvelintarjoajilla ainakin kahdesta eri syystä. Ensimmäinen syy on palvelut ja sovellukset, joita nykyään tarvitaan, esimerkiksi Web, FTP, DNS, sähköposti jne. Toiseksi, useat sovellukset tarvitsevat useita palvelimia, koska yksi palvelin ei yksinkertaisesti tarjoa riittävästi kapasiteettiä tai tehoa. Esimerkiksi jos sähköpostipalvelin ei pysty käsittelemään kasvavaa määrää käyttäjiä, uusi palvelin on lisättävä, jotta ongelma poistuu. Uuden palvelimen lisäyksessä on myös ajateltava sitä, kuinka kuinka tieto jaetaan kahden palvelimen välillä. Jos toinen palvelimista kaatuu, on toisen palvelimen hoidettava tehtävä kunnes rikkoontunut palvelin on taas kunnossa. [8, s. 2-3.]

4.2 Kuormanjako ja verkko

Perinteiset kytkimet ja reitittimet käyttävät IP-osoitteita tai MAC-osoitteita päättääkseen, minne paketti menee. Ne eivät kuitenkaan pysty täyttämään monimutkaisen palvelinfarmin tarpeita. Perinteiset kytkimet ja reitittimet eivat pysty lähettämään liikennettä älykkäästi tietylle sovellukselle tai tietylle palvelimelle. Jos kohde palvelin on kaatunut, perinteiset kytkimet jatkavat liikenteen lähettämistä kaatuneelle palvelimelle. Jotta pystyy ymmärtämään

perinteisen kytkimen toiminnan ja kuinka web-kytkentä täytyy tuntea OSI-malli, johon tutustuttiin kappaleessa 2.1. [8, s. 4-5.]

Perinteiset kytkimet ja reitittimet toimivat OSI-mallin 2/3 kerroksella. 2/3 kerroksen otsikon perusteella laitteet pystyvät tekemään päätöksiä, mihin paketti pitää lähettää ja kuinka sitä pitää prosessoida. Kytkimet periaatteessa tekevät hyödyllistä työtä, mutta paljon hyödyllistä informaatiota jää hyödyntämättä, koska laitteet eivät pysty tulkitsemaan kaikkea tietoa. Jotta kaikki tieto saataisiin otsikoista ulos, on käytettävä ylemmän kerroksen kytkimiä. Neljänneltä seitsemänteen kerrokseen tapahtuvalla kytkennällä kytkin katsoo 4-7 kerroksen otsikko tietoa ja tekee kytkentä päätökset sen perusteella.TCP ja UDP tärkeimmät tässä työssä käsiteltävät 4-kerroksen protokollat. TCP ja UDP sisältävät paljon hyödyllistä tietoa jonka perusteella voidaan tehdä älykkäitä kytkentä päätöksiä. Esimerkiksi, HTTP-protokolla, jota käytetään web-sivustoilla käyttää TCP:n porttia 80. Jos kytkin pystyy näkemään TCP porttinumeron, se saattaa pystyä priorisoimaan sen tai estämään käytön, tai ohjaamaan sen kokonaan toiselle palvelimelle. Ainoastaan katsomalla porttinumeroita, kytkimet pystyvät tunnistamaan monen yleisesti käytetyn protokollan liikenteen, kuten HTTP, FTP, DNS, SSL. TCP:n ja UDP:n tarjoaman informaation avulla, voidaan tehdä kuormanjakoa, jakamalla TCP- ja UDP-yhteydet useiden palvelinten kesken. [8, s. 5-6.]

4.3 Kuormanjakomenetelmät

Kuormantasaukseen on mahdollista käyttää useita eri menetelmiä. Tekijät, jotka vaikuttavat, mitä kuormantasausta käytetään, ovat laittestovaatimukset, käytettävissä olevat ominaisuudet, toteutuksen vaativuus ja hinta. Esimerkiksi jos käytetään erillistä laitteistoa kurmantasaukseen, on se huomattavasti kalliimpaa kuin ohjelmallisesti toteutettu. [11.]

Round Robin DNS-kuormanjako menetelmää käytetään DNS-palvelimissa ja tällä saadaan jaettua kuorma usean palvelimen kesken. Tämä yksi ensimmäisistä kuormanjako tekniikoista, jolla IP-osoitteita kierrätetään siten, että ensimmäinen saapuva pyyntö saa ensimmäisen osoitteen ja seuravaava pyyntö seuraavan. Riippuen palvelimella olevien osoitteiden määrästä, kun viimeinen osoite on annettu, palaa ympyrä taas alkuun ja

alkaa jakamaan osoitteita ensimmäisestä lähtien. Tässä menetelmässä hyviä puolia on yksinkertaisuus sekä helppo ja halpa toteutus. [11.]

Laitteistolla toteutettu kuormanjako pystyy reitittämään TCP/IP-paketteja useille palvelimille klusterissa. Tätä tapaa käytetään, jos halutaan korkea käytettävyys ja kestävä topologia. Tämän menetelmän huono puoli on korkea hinta. [11.]

Ohjelmallisesti toteutettu on yleisimmin käytetty menetelmä ja on usein integroitu komponentti palvelinohjelmisto paketissa. Tämä menetelmä on halvempi kuin laitteistolla toteutettu ja paremmin konfiguroitavissa. Huonona puolena voisi pitää erillisen laitteiston hankkimista, jotta kuormanjakaja saadaan eristettyä muusta laitteistosta. [11.]

5 HSRP

HSRP (Hot Standby Routing Protocol) on suunniteltu suojaamaan verkkoa häiriöiltä ja on rakennettu niin, että se on käyttäjälle näkymätön. Varmennetussa verkossa on päästävä tärkeimpiin verkkoresursseihin useamman reitin kautta. HSRP-protokollalla pystytään takaamaan pääsy verkkoresursseihin useampaa reittiä pitkin. Käytettäessä HSRP-protokollaa, useampi reititin toimii yhdessä ja näin luodaan illuusio yhdestä virtuaalireitittimestä käyttäjille, jotka ovet verkossa. Tästä ryhmästä käytetään nimeä HSRP-ryhmä tai valmiusryhmä. Käytännössä ainoastaan toinen reititin hoitaa pakettien reitityksen ja toinen reititin on käytössä varmistukseksi. Reititin, joka hoitaa reitityksen, on nimeltään aktiivinen reititin (active router) ja toinen varareititin(standby). Varareititin on määritelty korvaamaan aktiivireititin, jos häiriöitä tapahtuu. HSRP antaa sen edun, että jos toinen reitittimistä hajoaa tai kaatuu, verkon toiminta jatkuu eikä käyttäjä huomaa, että mitään olisi tapahtunut. Protokolla tarjoaa mekanismin, jolla pystytään määrittämään mukana olevat reitittimet. Ylimääräisen verkkoliikenteen minimoimiseksi, ainoastaan aktiivi- ja varareititin lähettävät jaksottaisia HSRP-viestejä sen jälkeen, kun valinta prosessi on tehty. Jos aktiivireititin vikaantuu, varareititin ottaa vastuun. Jos varareititin vikaantuu tai tulee aktiivireitittimeksi, joku toinen reititin valitaan varareitittimeksi. [9.]

HSRP:ssä käytetään konfiguroimiseen priorisointi menetelmää, jolla valitaan aktiivireititin. Jotta reititin saadaan konfiguroitua aktiivireitittimeksi, on prioriteetti määriteltävä korkeammaksi kuin muissa reitittimissä on määritelty. Oletus prioriteetti on 100, joten jos konfiguroidaan vain yksi reititin, jolla on korkeampi prioriteetti, siitä tulee aktiivireititin. HSRP toimii siten, että HSRP-protokollaa käyttävät reitittimet lähettävät multicast viestejä toisilleen ja mainostavat prioriteettiaan. Jos aktiivireititin ei lähetä tietttyyn konfiguroituun ajanjaksoon Hello-viestiä, varareititin alkaa toimimaan aktiivireitittimenä. [10.]

HSRP-protokollaa käyttävät reitittimet vaihtavat seuraavanlaisia viestejä:

- Hello – Kuljettaa toisille HSRP-reitittimille HSRP:n tilan ja prioriteetin. Oletuksena lähetetään kolmen sekunnin välein.
- Coup – Kun varareititin muuttaa tilansa aktiivireitittimeksi.
- Resign – Reititin, joka toimii aktiivireitittimenä lähettää tämän viestin jos se on sammumassa tai jos reititin, jolla on korkeampi prioriteetti, lähettää viestin. [10.]

HSRP-reitittimet ovat aina jossain seuraavista tiloista:

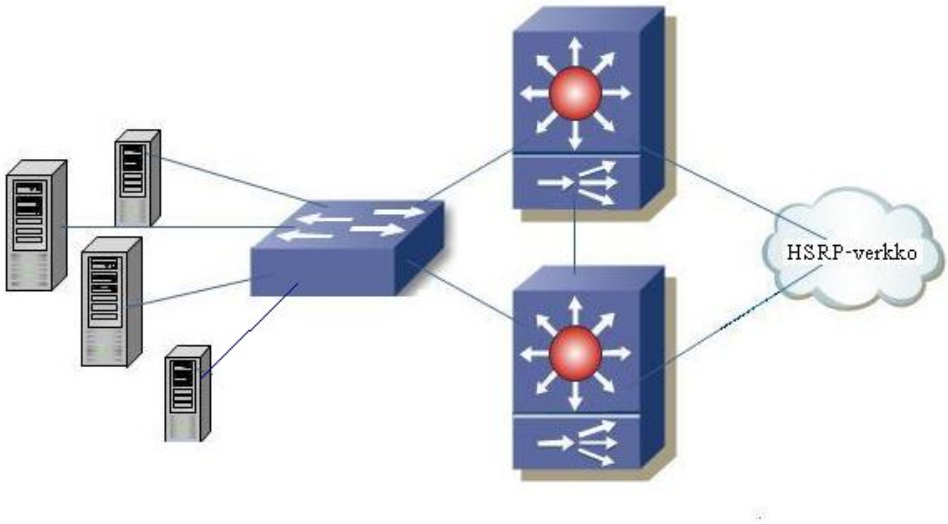
- Active – Reititin hoitaa reititystä.
- Standby – Reititin on valmiustilassa, jos aktiivireititin kaatuu.
- Speaking and listening – Reititin lähettää ja vastaanottaa viestejä.
- Listening – Reititin vastaanottaa viestejä. [10.]

6 TYÖ

Työ tehtiin kahdessa osuudessa, HSRP- ja F5-kokoonpano erikseen. Työn HSRP-osuudessa käytetään kahta Cisco 3560 layer 3-kytkintä ja kahta Cisco 2950 layer 2-kytkintä. F5-sisältökytkin osuudessa käytetään kahta F5 BigIP-kytkintä sekä yhtä Cisco 2950 layer 2-kytkintä. Kokoonpano tehtiin Metropolian laboratorioluokassa ja käytettiin hyväksi luokan työasemia, joihin asennettiin FTP sekä HTTP-palvelin ohjelmistot, joita työn toteuttamiseen tarvittiin. Näillä ohjelmistoilla testattiin kokoonpanon toimivuutta.

6.1 F5-kytkimien konfigurointi

F5-verkon osuus on kuvan 4 mukainen. F5-kytkimet kytketään toisiinsa fail over-porteista F5 laitevalmistajan omalla kaapelilla, joka tulee yleensä laitteen mukana. Tämä kaapeli on myös kohtalaisen helppo tehdä itse, jos sitä syystä tai toisesta ei ole saatavilla. Kaapelin avulla välitetään tilatietoja laitteiden tilasta. Jos toinen laite vikaantuu tai sammuu, toinen laite ottaa toiminnan itselleen.



Kuva 4: F5-verkon kuvaus

Työssä käytetään kahta F5-kytkintä, jotta saadaan palvelimet mahdollisimman vikasietoisiksi. Kumpaankin kytkimeen määritellään samat asetukset lukuunottamatta kummankin laitteen omaa hallinta-IP-osoitetta. Näistä kytkimistä tulee redundanttinen pari, joka mahdollistaa virhetilanteissa palvelimien toiminnan ilman käyttäjälle näkyvää haittaa tai katkosta. Jotta F5-kytkimien kautta saadaan palvelimien liikenne kulkemaan, tarvitaan lisäksi yksi kytkin, johon määritellään F5-kytkimeen menevät portit trunk-porteiksi. F5-kytkimien taakse tulevat palvelimet toimivat pareina, jotta saadaan varmistettua palvelinten redundanttisuus.

Laitteisiin täytyi aluksi määritellä IP-osoite hallintakonsolista, joka löytyy laitteen kyljestä. Tästä määritellään management-portille IP-osoite. Laitteella on oletuksena osoite 192.168.1.3. F5-kytkimien konfigurointi aloitetaan ottamalla yhteys laitteisiin management-portin kautta. F5-Kytkimeen on määriteltävä alkuasetukset, joita pääsee määrittämään etusivun welcome-

valikon alta. Valikosta löytyy ”run setup”, johon täytyy määritellä IP-osoitteet, joilla laitteeseen saadaan yhteys ulko- ja sisäverkosta. Näistä asetuksista voi myös määritellä, että laitteita on kaksi ja ne toimivat redundanttisena parina. Laitteille on myös annettava yksikkö tunnus, jotta laitteet tunnistetaan eivätkä ne sekoitu toistensa kanssa. Alkuasetuksissa määritellään laitteille ”kelluva” IP-osoite ja fyysinen osoite. Koska laitteet toimivat redundanttisena parina, toinen laitteista on aina aktiivisessa ja toinen laite on passiivisessa tilassa. Kun laitteeseen otetaan yhteyttä verkon kautta, otetaan yhteys ”kelluvaan” IP-osoitteeseen, joka on kummallekin laitteelle määritetty samaksi. ”Kelluva” IP-osoite on aina aktiivisen laitteen käytössä, joten aina, kun yhteyttä otetaan, otetaan yhteys aktiiviseen laitteeseen. Fyysiseksi osoitteeksi määritellään ensimmäiseen laitteeseen 172.16.50.101 ja toiseen 172.16.50.102. ”Kelluvaksi” osoitteeksi määritellään kumpaankin laitteeseen 172.16.50.100. Nämä osoitteet määritellään käyttämään porttia 1.1.

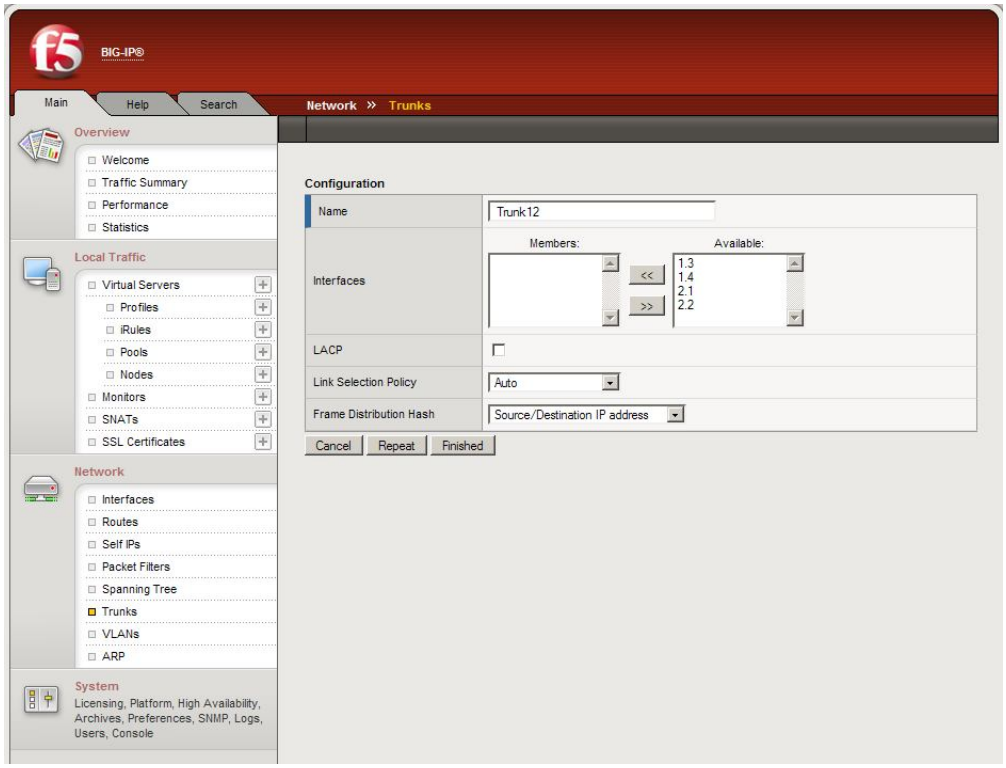
Kun laitteeseen on saatu yhteys, kannattaa ennen enempiä määrityksiä konfiguroida Ciscon kytkin. Konfiguroinnissa ei tehdä muuta kuin määritellään F5-kytkimeen menevät portit trunk-porteiksi ja halutuille porteille pääsy haluttuihin VLAN:hin. Seuraavanlaiset kuvan 5 mukaiset konfiguraatiot tehtiin kytkimelle.

```
hostname F5-Layer2
interface FastEthernet0/1
 switchport access vlan 2
!
interface FastEthernet0/2
 switchport access vlan 2
!
interface FastEthernet0/3
 switchport access vlan 3
!
interface FastEthernet0/4
 switchport access vlan 3
!
interface FastEthernet0/5
 switchport access vlan 4
!
interface FastEthernet0/6
 switchport access vlan 4
!
interface FastEthernet0/7
 switchport access vlan 5
!
interface FastEthernet0/8
 switchport access vlan 5
!
interface FastEthernet0/10
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/11
 switchport trunk encapsulation dot1q
 switchport mode trunk
```

Kuva 5: Kytkimen konfiguraatio

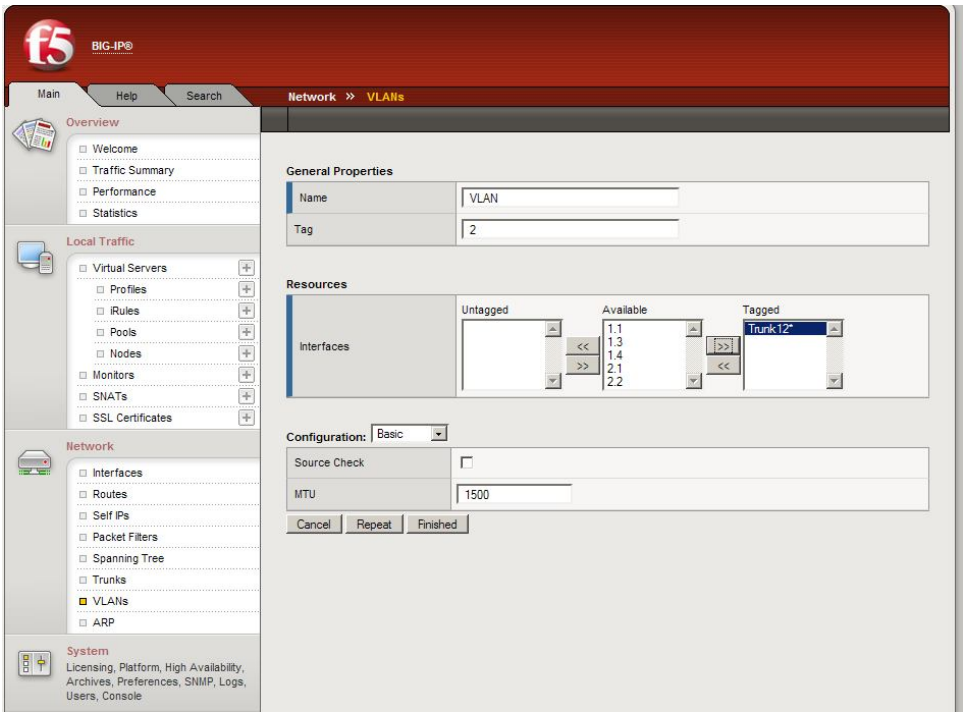
Nyt kun laitteeseen on saatu yhteys, päästään laitetta hallitsemaan verkon kautta. Aluksi laitteelle on konfiguroitava trunk-portti, joka liitetään porttiin 1.2. Luonti tapahtuu valitsemalla vasemman puoleisesta valikosta Network

ja sen alapuolelta Trunks. Valitsemalla create pääsee luomaan itse porttia. Portille on annettava nimi. Se on vapaavalintainen, mutta pakollinen. Alla olevasta valikosta valitaan haluttu rajapinta, joka on työssä 1.2. Se liitetään members-kategoriaan. Muita valintoja ei tarvitse tehdä, vaan oletusasetukset ovat riittävät. Kuvasta 6 näkee laitteen käyttöliittymän ja se kuinka asetukset tehdään.



Kuva 6: Hallintaliittymä ja trunk-portin asettaminen

Kun trunk-portit on luotu, voidaan luoda yksittäiset VLAN:it. Vasemmassa reunassa olevassa konfigurointivalikossa valitaan Network ja alhaalle aukeavasta valikosta VLANs. Oikeasta laidasta pääsee luomaan VLAN:in painamalla create-nappia. VLAN:in luonti aloitetaan nimeämisellä; nimellä ei ole väliä, kunhan sen itse muistaa. Tag-kohdassa VLAN:ille voidaan asettaa esimerkiksi samaa VLAN:ia osoittava numero, mutta tämä ei ole pakollista. VLAN on vielä liitettävä trunk-porttiin ja tämä tapahtuu valitsemalla jo luodun trunk-rajapinnan valikosta ja siirtämällä sen tagged-kenttään. Samalla tavoin luodaan myös muut VLAN:it VLAN2:sta eteenpäin. Kuvasta 7 näkyvät täytettävät kohdat.



Kuva 7: VLAN:ien luonti

VLAN:ien luonnin jälkeen VLAN:lle määritellään IP-osoitteet. Valikosta Network valikon alta löytyy Self IPs, josta pääsee create-napista määrittelemään IP-osoitteet. Täällä annetaan IP-osoite, aliverkon peite ja määritetään IP-osoite haluttuun VLAN:iin. Jos IP-osoitteiden määrittelyssä tapahtuu virhe, pitää se poistaa ja lisätä uudestaan. Osoitetta ei voi muuttaa sen jälkeen, kun se on kerran määritelty. Tässä työssä käytetään seuraavanlaisia IP-osoitteita:

- VLAN2 192.168.12.1
- VLAN3 192.168.13.1
- VLAN4 192.168.14.1
- VLAN5 192.168.15.1.

Jos IP-osoitteiden määrittelyssä tehdään virhe, on se aluksi poistettava ja määriteltävä oikea IP-osoite uudestaan. Kerran määriteltyä osoitetta ei voi muokata luonnin jälkeen. VLAN:in osoite toimii oletusyhdyskäytävänä VLAN:iin liitetyille koneille.

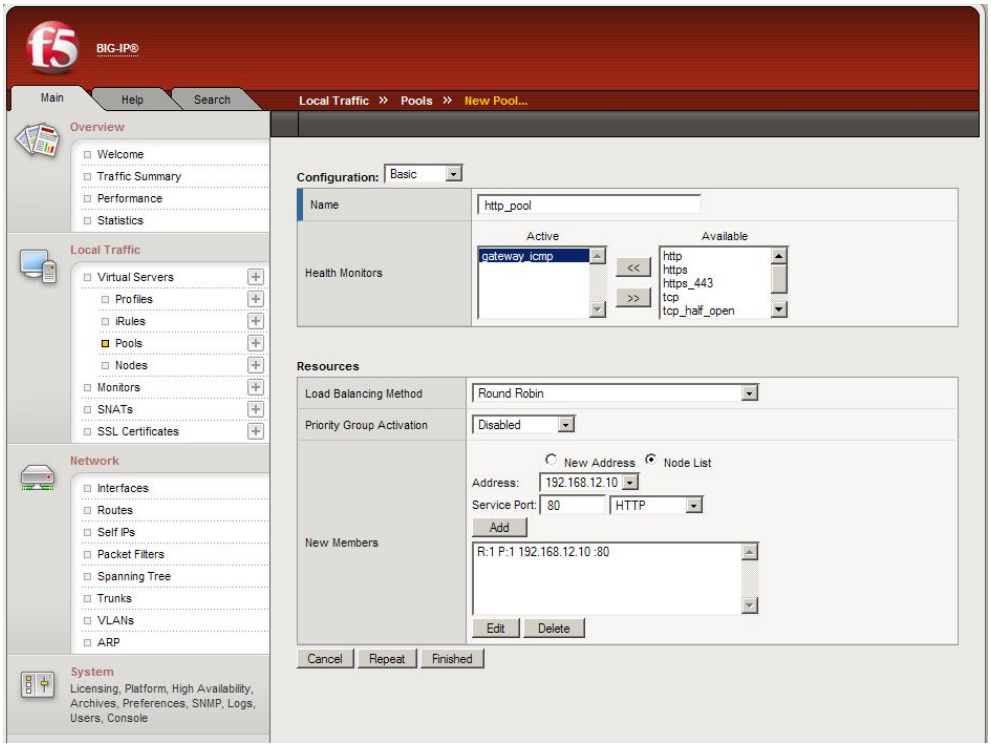
Seuraavaksi on luotava altaat F5-kytkimeen. Tässä vaiheessa voidaan määritellä, mitkä koneet ovat käytössä, mutta fyysisten koneiden ei tässä

vaiheessa tarvitse olla kiinnitettynä mihinkään. Koneiden luonti aloitetaan valitsemalla Local Traffic, Nodes. Koneet voi nimetä haluamallaan tavalla. Tässä työssä käytetään nimiä Node1 - Node5. Aktiiviseksi monitorointi välineeksi voidaan valita tässä vaiheessa ylävalikosta välilehdeltä Default Monitor, ICMP. Koneiden luonti alkaa Create-napista; jokaiselle annetaan IP-osoite ja nimi. Koneille annettiin seuraavan taulukon mukaiset nimet ja IP-osoitteet:

Taulukko 1: Koneiden nimeäminen ja osoitteistus

VLAN 2	VLAN 3	VLAN 4
192.168.12.10 Node1	192.168.13.10 Node3	192.168.14.10 Node5
192.168.12.11 Node2	192.168.13.11 Node4	

Koneiden luonnin jälkeen voidaan alkaa luomaan altaita. Hallintasivun vasemmasta laidasta valitaan Local Traffic, Pools. Create-napilla aloitetaan altaiden luonti. Annetaan altaalle nimi, valitaan aktiiviseksi monitoroinniksi gateway_icmp ja tässä vaiheessa kuormantasausmenetelmäksi Round Robin, jonka tarkoitus on varmistaa, että palvelimia kuormitetaan tasapuolisesti, eikä jouduta tilanteeseen, jossa yksi palvelimista joutuisi ylikuormitus tilanteeseen ja sen vuoksi aiheuttaisi palvelimen kaatumisen. Myöhemmässä vaiheessa voidaan testata muita F5-kytkimen tarjoamia kuormantasaus menetelmiä, jos halutaan selvittää, mitä erilaiset kuormantasausmenetelmät tekevät. Kuvasta 8 näkyvät HTTP-poolin luontiin tarvittavat määritykset. Muut altaat luodaan samalla tekniikalla.



Kuva 8: Altaiden luonti

Http_pool:iin määritellään service port HTTP, ftp_pool FTP ja cache_pool HTTP. Add-napillla lisätään koneet listaan. Työssä luotiin seuraaavanlaiset altaat:

- http_pool: node1 ja node2
- ftp_pool: node3 ja node4
- cache_pool: node5.

Kun altaat on luotu, voidaan siirtyä luomaan virtuaali palvelimia. Luonti aloitetaan valitsemalla vasemmasta reunasta Local Traffic, Virtual Servers. Uuden virtuaalipalvelimen luonti aloitettaan Create-painikkeesta. Palvelimelle annetaan nimi, IP-osoite ja service port. Näiden lisäksi palvelimelle määritellään oletusallas. Kun luodaan HTTP-palvelinta, valitaan profiiliksi HTTP ja FTP-palvelimen luonnissa käytetään profiilia FTP.

Luotiin seuraavanlaiset palvelimet:

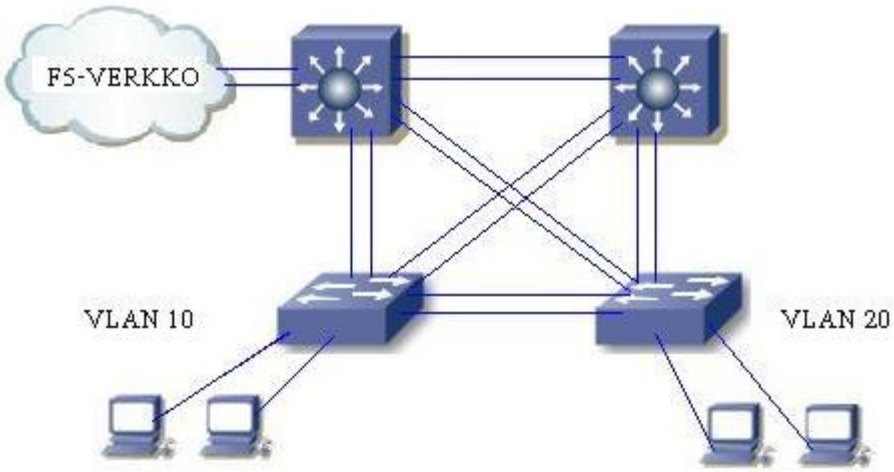
- HTTP_Server 172.16.50.150, Service Port 80 HTTP, default pool: http_pool

- FTP_Server 172.16.50.151, Service Port 21 FTP, Default Pool: ftp_pool.

Kun kaikki asetukset on tehty, voidaan asentaa työasemille FTP ja HTTP-palvelin ohjelmistot sekä määrittää työasemille IP-osoitteet. Node 1 ja 2 ovat HTTP-palvelimia ja Node 3 ja 4 FTP-palvelimia.

6.2 HSRP-verkon konfigurointi

HSRP-verkon tekeminen alkaa kuvan 9 mukaan. Kuvassa olevat kytkimet DLS1, DLS2, ALS1 ja ALS2 kytketään toisiinsa verkkojohdoilla kuvan mukaisesti. Kun laitteet on kytketty toisiinsa, kytketään virrat päälle ja aloitetaan kofigurointi ottamalla konsoliyhteys kytkimiin. Aluksi on tarkistettava, että kytkimiin ei ole jäänyt aiempia kongurointeja, jotka saattaisivat estää uuden konfiguraation toiminnan. Kun on varmistettu, että laitteessa ei ole vanhoja konfigurointeja, voidaan aloittaa konfiguraation määritteleminen. Lopulliset konfiguraatiot ovat liitteenä.



Kuva 9: HSRP-verkko kuvaus.

Konfigurointi aloitetaan DLS-kytkimistä. Aluksi voidaan nimetä kytkimet halutuilla nimillä. Kytkimille voidaan myös halutessa määrittää salasana ja Telnet-yhteys. Koska kytkimet ovat yhdistetty toisiinsa aina kahdella kaapelillla, esimerkiksi DLS1-kytkin DLS-kytkimeen porteista fastethernet0/11 – fastethernet0/12, luodaan näiden välille ether-kanava. Tämä tarkoittaa sitä, että laitteiden väliset fyysiset yhteydet niputetaan yhdeksi loogiseksi yhteydeksi. Jokaiselle kytkimelle tehdä pitää tehdä sama

konfiguraatio. Samalla määritetään kytkimien väliset yhteydet trunk-linkeiksi, jotta saadaan välitettyä VLAN-tietoa muille verkon kytkimille. Alla olevassa kuvassa 10 konfiguraatio DLS1-kytkimelle.

```
interface FastEthernet0/7
  switchport trunk encapsulation dot1q
  switchport mode trunk
  channel-group 1 mode desirable
 !
 interface FastEthernet0/8
  switchport trunk encapsulation dot1q
  switchport mode trunk
  channel-group 1 mode desirable
 !
 interface FastEthernet0/9
  switchport trunk encapsulation dot1q
  switchport mode trunk
 channel-group 2 mode desirable
 !
 interface FastEthernet0/10
  switchport trunk encapsulation dot1q
  switchport mode trunk
  channel-group 2 mode desirable
 !
 interface FastEthernet0/11
  switchport trunk encapsulation dot1q
  switchport mode trunk
  channel-group 3 mode desirable
 !
 interface FastEthernet0/12
  switchport trunk encapsulation dot1q
  switchport mode trunk
  channel-group 3 mode desirable
```

Kuva 10: DLS1-kytkimen konfiguraatiota.

Kun konfiguraatio on tehty, voidaan trunk-linkin tila tarkistaa komennolla show interface trunk. Kuvan 11 mukaan DLS1-kytkimellä on trunk on päällä, ja se on toiminnassa.

```
DLS1# show interface trunk

Port        Mode         Encapsulation  Status        Native vlan
Po1         on           802.1q         trunking      1
Po2         on           802.1q         trunking      1
Po3         on           802.1q         trunking      1

Port      Vlans allowed on trunk
Po1         1-4094
Po2         1-4094
Po3         1-4094

Port        Vlans allowed and active in management domain
Po1         1
Po2         1
Po3         1

Port        Vlans in spanning tree forwarding state and not pruned
Po1         1
Po2         1
Po3         1
```

Kuva 11: Trunk-toiminnon tarkistus

Komennolla ”show etherchannel summary” voidaan tarkistaa ether-kanavien toiminta. Kuvasta 12 näkyy ether-kanavien määrä, minkä porttien välille kanavat on kytketty ja mitä protokollaa kanavassa käytetään.

```
ALS1# show etherchannel summary
Flags:  D - down        P - in port-channel

    I - stand-alone s - suspended
    H - Hot-standby (LACP only)
    R - Layer3      S - Layer2
    U - in use      f - failed to allocate aggregator
    u - unsuitable for bundling
    w - waiting to be aggregated
    d - default port

Number of channel-groups in use: 3
Number of aggregators:           3

Group  Port-channel  Protocol    Ports
------+-------------+-----------+--------------------
--
1      Po1(SU)          PAgP      Fa0/7(P)    Fa0/8(P)
2      Po2(SU)          PAgP      Fa0/9(P)    Fa0/10(P)
3      Po3(SU)          PAgP      Fa0/11(P)   Fa0/12(P)
```

Kuva 12: Ether-kanavien tarkistus

Seuraavaksi voidaan siirtyä VTP:n (VLAN Trunking Protocol) tekemiseen. VTP:llä hallitaan helposti VLAN-verkkoa. Jos VTP-palvelimelle lisätään uusi VLAN, jaetaan tieto verkon muille kytkimille. Näin vältytään turhalta työltä ja säästetään aikaa konfiguroinnissa. ALS-kytkimet määritellä ”asiakkaiksi”,

jotka vastaanottavat tietoa VTP-palvelimelta. Komento ”vtp mode client” määrittää kytkimet tilaan, jossa ne vastaanottavat muutos tietoja. Kytkimet ovat oletuksena palvelintilassa, joten DLS-kytkimille ei tarvitse tehdä muutoksia tämän suhteen. DLS1-kytkimelle määritellään VTP-toimialue ja sen nimi. Tässä työssä toimialueen nimi on INSSI.

Toimialueen määrityksen lisäksi luodaan VLAN:it. Luodaan neljä uutta VLAN:ia ja määritellään niille nimet. Alla olevassa kuvassa 13 on konfiguroitu työssä käytettävät VLAN:it ja niiden nimet. Kuvassa 13 on myös VTP-toimialueen konfigurointi. Tämä konfigurointi tehdään ainostaan DLS1-kytkimelle.

```
DLS1(config)#vtp domain INSSI
DLS1(config)#vlan 10
DLS1(config-vlan)#name HOSTS
DLS1(config-vlan)#exit
DLS1(config)#vlan 20
DLS1(config-vlan)#name HOSTS2
DLS1(config-vlan)#exit
DLS1(config)#vlan 50
DLS1(config-vlan)#name Optional-Srv
DLS1(config-vlan)#exit
DLS1(config)#vlan 40
DLS1(config-vlan)#name F5-SRV
DLS1(config-vlan)#end
```

Kuva 13: VTP- ja VLAN-konfigurointi

Seuraavassa vaiheessa määritellään vapaassa käytössä olevat portit pääsy-porteiksi. Se tarkoittaa sitä, että näistä porteista voidaan päästä kyseiseen verkkoon. DLS1-kytkimeen määritellään pääsy porteille fastethernet0/5 ja fastethernet0/6, joihin kytketään F5-kytkimet. Näille porteille määritellään myös pääsy VLAN:iin 50. Muille kytkimille määritellään pääsy sen mukaan mitä VLAN-määrityksiä kytkimeen on tehty.

Seuraavassa vaiheessa siirrytään HSRP:n konfigurointiin. Ennen kuin aletaan määrittelemään VLAN:hin mitään, komennolla ”ip routing” määritetään kytkin reitittäväksi, jotta saadaan kytkimeen reititys ominaisuudet käyttöön. HSRP:n konfigurointi aloitetaan määrittelemällä VLAN:lle IP-osoite ja aliverkon peite. Tämä konfigurointi tehdään ainoastaan DLS-kytkimille. DLS1-kytkimelle VLAN10:een määritellään IP-osoite 172.16.10.3 ja DLS2-kytkimelle 172.16.10.4. Kumpaankin määritellään standby-osoitteeksi 172.16.10.1. Samalla määritellään kumpaankin kytkimeen prioriteetit. DLS1 saa prioriteetti arvoksi 150 ja DLS2 saa arvoksi

100. Tällä asetuksella määritellään se, kumpi kytkimistä saa standby-osoitteen ja toimii tälle VLAN:lle reittinä ulos. DLS1 toimii aktiivisena kytkimenä VLAN:lle 10 ja jos tämä kytkin vikaantuu, DLS2 alkaa toimimaan aktiivisena. VLAN20 määritellään samalla tavalla VLAN10 kanssa. VLAN 40 ja VLAN50 aktiivisena reitittimenä toimii DLS2 ja valmiusreitittimenä DLS1. Kuvassa 14 näkyy DLS1-kytkimeen tarvittava HSRP-konfiguraatio .

```
interface Vlan10
 ip address 172.16.10.3 255.255.255.0
 standby 1 ip 172.16.10.1
 standby 1 priority 150
 standby 1 preempt
!
interface Vlan20
 ip address 172.16.20.3 255.255.255.0
 standby 1 ip 172.16.20.1
 standby 1 priority 150
 standby 1 preempt
!
interface Vlan40
 ip address 172.16.40.3 255.255.255.0
 standby 1 ip 172.16.40.1
 standby 1 preempt
!
interface Vlan50
 ip address 172.16.50.3 255.255.255.0
 standby 1 ip 172.16.50.1
 standby 1 preempt
```

Kuva 14: DLS1-kytkimen HSRP-konfiguraatio

HSRP:n nykyisen tilan pystyy tarkistamaan komennolla ”show standby”. Tällä komennolla näkyy, mikä kytkimistä on aktiivisena millekin VLAN:lle, milloin on tapahtunut muutoksia, mikä on prioriteetti ja mikä on laitteen virtuaalinen IP-osoite eli standby-osoite.

6.3 Testaus

F5-laitteiden redundanttisuus testattiin kytkemällä virrat pois aktiivisen olevasta laitteesta. Passiivinen laite muutti tässä vaiheessa tilansa aktiiviseksi ja siirtyi hallitsemaan toimintoja. Kun toinen laite kytkettiin takaisin päälle, jäi se passiiviseen tilaan, joten tämä toimi niinkuin pitikin.

Palvelimien toimintaa testattiin FTP-palvelimilla, koska yhteydenotto yritykset näkyivät selkeästi ohjelmiston etusivun näytöllä. Kuormantasaukseksi oli määritelty round robin, joten kun FTP-palvelimeen otettiin yhteyttä kuorma jakautui tasaisesti kummallekin palvelimelle. FTP-yhteydenotot näkyivät tasaisesti kummankin palvelimen monitoroinnissa. Tämä toimi niin kuin pitikin.

HSRP-verkon toimivuutta testattiin irroittamalla DLS1-kytkimen ja ALS1-kytkimen välinen yhteys. Tällöin DLS2-kytkin otti yhteyden haltuunsa ja tätä kautta pääsi ulos verkosta. Kun yhteys palautettiin takaisin, DLS1-kytkin otti yhteyden takaisin haltuunsa. Sama testattiin myös muissa väleissä ja kaikki toimi niin kuin pitikin.

7 YHTEENVETO

Työn tarkoituksena oli saada toimintaan F5-sisältökytkimet ja niiden tuomat edut rakennettaessa redundanttista järjestelmää. Tällä hetkellä laitteet eivät ole tuotantokäytössä, mutta tämän työn avulla tutkittiin mahdollista käyttää sisältökytkimiä tulevaisuudessa.

Käytännön osuuden toteuttamiseen oli jo olemassa osittaiset ohjeet, mutta kahden laitteen käytöstä redundanttisena järjestelmänä ei ollut ohjeita ja laitteiden käytöönotto redundanttisena parina tuotti välillä hankaluuksia puuttuvan ohjeistuksen vuoksi. Laitteiden käyttöönottaminen jouduttiin turvautumaan välillä jopa keskustelupalstojen apuun, josta tietoja löytyi ripotellen, mutta näiden avulla päästiin toimivaan lopputulokseen. Myös laitteiden toiminnallisuuden kannalta tärkeä kaapeli jouduttiin tekemään itse, koska alkuperäinen oli kadonnut.

Haasteita tuotti myös joka kertainen kokoonpanon kasaus, johon kului oma aikansa. Loppujen lopuksi laitteiden konfigurointi oli kohtuullisen helppo töinen, kun vain kaikki tieto oli saatu kasattua. Konfigurointi ja laitteiden kytkeminen oli suoraviivaista toimintaa valmiiden konfiguraatio tietojen ollessa saatavilla.

Työlle asetetut vaatimukset saavutettiin ja työn pohjalta laitteiden käyttöönotto helpottuu valmiiden tietojen ollessa saatavilla. Työn loppuun saattaminen kesti suunniteltua pitempään, johtuen rajallisista mahdollisuuksista tehdä työtä koululla ja omien työkiireiden vuoksi. Työssä olisi voitu myös keskittyä enemmän itse sisältökytkentään, toisaalta palvelujen varmistaminen sisältökytkimien avulla on myös tärkeä osa, mitä kytkimillä pystyy tekemään.

VIITELUETTELO

[1] TCP/IP, Wikipedia vapaa tietosanakirja, [verkkodokumentti, viitattu 17.10.2008] Saatavissa: http://fi.wikipedia.org/wiki/TCP/IP.

[2] OSI-malli, Wikipedia vapaa tietosanakirja, [verkkodokumentti, viitattu 17.10.2008] Saatavissa: http://fi.wikipedia.org/wiki/Sovelluskerros.

[3] Päättötyö, [verkkodokumentti, viitattu 24.10.2008] Saatavissa: http://koti.mbnet.fi/mrin/paattotyo/tcp_ip.html.

[4] TCP, Wikipedia vapaa tietosanakirja, [verkkodokumentti, viitattu 25.10.2008] Saatavissa http://fi.wikipedia.org/wiki/TCP.

[5] Network Computing – Web content switching, [verkkoartikkeli, viitattu 31.10.2008] Saatavissa: http://www.networkcomputing.com/1115/1115f1.html.

[6] Techworld – RTFM: Content Switching, [verkkoartikkeli, viitattu 3.11.2008] Saatavissa: http://www.techworld.com/networking/features/index.cfm?featureid=472.

[7] PMC-Sierra – URL-based switching, [verkkodokumentti, viitattu 3.11.2008] Saatavissa: www.pmc-sierra.com/cgi-bin/document.pl?docnum=2002232.

[8] Kopparapu, Chandra (2002), Load Balancing Servers, Firewalls, and Caches. Yhdysvallat:John Wiley & Sons, Inc.

[9] Internet-protokollien standardoinnista vastaava organisaatio, [verkkodokumentti, viitattu 29.12.2008] Saatavissa: http://www.ietf.org/rfc/rfc2281.txt.

[10] Cisco Systems – Using HSRP for Fault-Tolerant IP Routing, [verkkodokumentti, viitattu 29.12.2008] Saatavissa: http://www.cisco.com/en/US/docs/internetworking/case/studies/cs009.html.

[11] Website Gear – Server Load Balancing Methods, [verkkoartikkeli, viitattu 29.12.2008] Saatavissa: http://content.websitegear.com/article/load_balance_methods.htm.

Cisco kytkimien konfiguraatiot HSRP-verkossa.

```
hostname ALS1
!
no aaa new-model
system mtu routing 1500
ip subnet-zero
!
no file verify auto
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
interface Port-channel1
 switchport mode trunk
!
interface Port-channel2
 switchport mode trunk
!
interface Port-channel3
 switchport mode trunk
!
interface FastEthernet0/6
 switchport access vlan 10
 switchport mode access
!
interface FastEthernet0/7
 switchport mode trunk
 channel-group 1 mode desirable
!
interface FastEthernet0/8
 switchport mode trunk
 channel-group 1 mode desirable
!
interface FastEthernet0/9
 switchport mode trunk
 channel-group 2 mode desirable
!
interface FastEthernet0/10
 switchport mode trunk
 channel-group 2 mode desirable
!
interface FastEthernet0/11
 switchport mode trunk
 channel-group 3 mode desirable
!
interface FastEthernet0/12
 switchport mode trunk
 channel-group 3 mode desirable
!
interface Vlan1
 ip address 172.16.1.101 255.255.255.0
 no ip route-cache
!
```

```
ip default-gateway 172.16.1.1
ip http server
!
control-plane
!
end
```

```
hostname ALS2
!
no aaa new-model
system mtu routing 1500
ip subnet-zero
!
no file verify auto
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
interface Port-channel1
 switchport mode trunk
!
interface Port-channel2
 switchport mode trunk
!
interface Port-channel3
 switchport mode trunk
!
interface FastEthernet0/6
 switchport access vlan 20
 switchport mode access
!
interface FastEthernet0/7
 switchport mode trunk
 channel-group 1 mode desirable
!
interface FastEthernet0/8
 switchport mode trunk
 channel-group 1 mode desirable
!
interface FastEthernet0/9
 switchport mode trunk
 channel-group 2 mode desirable
!
interface FastEthernet0/10
 switchport mode trunk
 channel-group 2 mode desirable
!
interface FastEthernet0/11
 switchport mode trunk
 channel-group 3 mode desirable
!
interface FastEthernet0/12
```

```
 switchport mode trunk
 channel-group 3 mode desirable
!
interface Vlan1
 ip address 172.16.1.102 255.255.255.0
 no ip route-cache
!
ip default-gateway 172.16.1.1
ip http server
ip http secure-server
!
control-plane
!
end
```

```
hostname DLS1
!
no aaa new-model
system mtu routing 1500
ip subnet-zero
ip routing
!
no file verify auto
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
interface Port-channel1
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface Port-channel2
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface Port-channel3
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/5
 switchport access vlan 50
 switchport mode access
!
interface FastEthernet0/6
 switchport access vlan 50
 switchport mode access
!
interface FastEthernet0/7
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 1 mode desirable
```

```
!
interface FastEthernet0/8
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 1 mode desirable
!
interface FastEthernet0/9
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 2 mode desirable
!
interface FastEthernet0/10
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 2 mode desirable
!
interface FastEthernet0/11
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 3 mode desirable
!
interface FastEthernet0/12
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 3 mode desirable
!
interface Vlan1
 ip address 172.16.1.3 255.255.255.0
 standby 1 ip 172.16.1.1
 standby 1 priority 150
 standby 1 preempt
!
interface Vlan10
 ip address 172.16.10.3 255.255.255.0
 standby 1 ip 172.16.10.1
 standby 1 priority 150
 standby 1 preempt
!
interface Vlan20
 ip address 172.16.20.3 255.255.255.0
 standby 1 ip 172.16.20.1
 standby 1 priority 150
 standby 1 preempt
!
interface Vlan40
 ip address 172.16.40.3 255.255.255.0
 standby 1 ip 172.16.40.1
 standby 1 preempt
!
interface Vlan50
 ip address 172.16.50.3 255.255.255.0
 standby 1 ip 172.16.50.1
 standby 1 preempt
!
router eigrp 1234
```

```
 network 172.16.0.0
 network 192.168.0.0 0.0.255.255
 auto-summary
!
vtp domain INSSI
!
vlan 10
!
name HOSTS
!
exit
!
vlan 20
!
name HOSTS2
!
exit
!
vlan 50
!
name Optional-Srv
!
exit
!
vlan 40
!
name F5-SRV
!
end
!
ip classless
ip http server
ip http secure-server
!
!
!
control-plane
!
end
```

```
hostname DLS2
!
no aaa new-model
system mtu routing 1500
ip subnet-zero
ip routing
!
no file verify auto
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
```

```
interface Port-channel1
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface Port-channel2
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface Port-channel3
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/6
 switchport access vlan 40
 switchport mode access
!
interface FastEthernet0/7
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 1 mode desirable
!
interface FastEthernet0/8
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 1 mode desirable
!
interface FastEthernet0/9
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 2 mode desirable
!
interface FastEthernet0/10
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 2 mode desirable
!
interface FastEthernet0/11
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 3 mode desirable
!
interface FastEthernet0/12
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 3 mode desirable
!
interface Vlan1
 ip address 172.16.1.4 255.255.255.0
 standby 1 ip 172.16.1.1
 standby 1 preempt
!
interface Vlan10
 ip address 172.16.10.4 255.255.255.0
 standby 1 ip 172.16.10.1
 standby 1 preempt
```

```
!
interface Vlan20
 ip address 172.16.20.4 255.255.255.0
 standby 1 ip 172.16.20.1
 standby 1 preempt
!
interface Vlan40
 ip address 172.16.40.4 255.255.255.0
 standby 1 ip 172.16.40.1
 standby 1 priority 150
 standby 1 preempt
!
interface Vlan50
 ip address 172.16.50.4 255.255.255.0
 standby 1 ip 172.16.50.1
 standby 1 priority 150
 standby 1 preempt
!
router eigrp 1234
 network 172.16.0.0
 network 192.168.0.0 0.0.255.255
 auto-summary
!
ip classless
ip http server
ip http secure-server
!
control-plane
!
!
end
```

F5-verkon Cisco kytkin

```
hostname F5-layer2
!
!
!
!
!
!
!
ip subnet-zero
!
!
!
interface FastEthernet0/10
 duplex full
 speed 100
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/11
 duplex full
```

```
 speed 100
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface VLAN1
 no ip address
 no ip directed-broadcast
 no ip route-cache
 shutdown
!
interface VLAN2
 no ip directed-broadcast
 no ip route-cache
!
interface VLAN3
 no ip directed-broadcast
 no ip route-cache
 shutdown
!
interface VLAN4
 no ip directed-broadcast
 no ip route-cache
 shutdown
!
interface VLAN5
 no ip directed-broadcast
 no ip route-cache
 shutdown
!
!
line con 0
 transport input none
 stopbits 1
line vty 5 15
!
end
```

F5-kytkimistä otetut konfiguraatiot UNIT 1

```
self 172.16.50.100 {
   netmask 255.255.255.0
   unit 1
   floating enable
   vlan VLAN50
   allow default
}
self 192.168.2.1 {
   netmask 255.255.255.0
   unit 1
   floating enable
   vlan internal
   allow tcp https
}
self 172.16.60.100 {
```

```
    netmask 255.255.255.0
    unit 1
    floating enable
    vlan external
    allow tcp https
}
partition Common {
    description "Repository for system objects and shared objects."
}
route default inet {
    vlan VLAN50
}
shell write partition Common
user admin {
    password crypt "$1$PTKcsTa0$.fWDg/BJS1Q4gJSu.XGoj/"
    description "Admin User"
    id 0
    group 500
    home "/home/admin"
    shell "/bin/false"
    role administrator in all
}
user f5emsvr {
    password crypt "!!"
    description "F5 EM Service Account"
    id 975
    group 975
    home "/root"
    shell "/bin/false"
    role guest in all
}
node 192.168.12.10 {
    screen Node1
}
node 192.168.12.11 {
    screen Node2
}
node 192.168.13.10 {
    screen Node3
}
node 192.168.13.11 {
    screen Node4
}
node 192.168.14.10 {
    screen Node5
}
pool http_pool {
    monitor all gateway_icmp
    members
        192.168.12.10:http
        192.168.12.11:http
}
pool ftp_pool {
    monitor all gateway_icmp
    members
```

```
      192.168.13.10:ftp
      192.168.13.11:ftp
}
pool cache_pool {
   monitor all gateway_icmp
   members 192.168.14.10:http
}
virtual HTTP_Server {
   pool http_pool
   destination 172.16.50.150:http
   ip protocol tcp
   profiles
      http
      tcp
}
virtual FTP_Server {
   pool ftp_pool
   destination 172.16.50.151:ftp
   ip protocol tcp
   profiles
      ftp
      tcp
}
node * monitor icmp
```

F5-kytkimistä otetut konfiguraatiot UNIT 2

```
self 192.168.2.1 {
   netmask 255.255.255.0
   unit 1
   floating enable
   vlan internal
   allow all
}
self 172.16.50.100 {
   netmask 255.255.255.0
   unit 1
   floating enable
   vlan VLAN50
   allow default
}
self 172.16.60.100 {
   netmask 255.255.255.0
   unit 1
   floating enable
   vlan external
   allow tcp https
}
partition Common {
   description "Repository for system objects and shared objects."
}
route default inet {
   vlan VLAN50
}
```

```
shell write partition Common
user admin {
   password crypt "$1$PTKcsTa0$.fWDg/BJS1Q4gJSu.XGoj/"
   description "Admin User"
   id 0
   group 500
   home "/home/admin"
   shell "/bin/false"
   role administrator in all
}
user f5emsvr {
   password crypt "!!"
   description "F5 EM Service Account"
   id 975
   group 975
   home "/root"
   shell "/bin/false"
   role guest in all
}
node 192.168.13.11 {
   screen Node4
}
node 192.168.13.10 {
   screen Node3
}
node 192.168.12.10 {
   screen Node1
}
node 192.168.14.10 {
   screen Node5
}
node 192.168.12.11 {
   screen Node2
}
pool http_pool {
   monitor all gateway_icmp
   members
      192.168.12.10:http
      192.168.12.11:http
}
pool ftp_pool {
   monitor all gateway_icmp
   members
      192.168.13.10:ftp
      192.168.13.11:ftp
}
pool cache_pool {
   monitor all gateway_icmp
   members 192.168.14.10:http
}
virtual HTTP_Server {
   pool http_pool
   destination 172.16.50.150:http
   ip protocol tcp
   profiles
```

```
      http
      tcp
}
virtual FTP_Server {
   pool ftp_pool
   destination 172.16.50.151:ftp
   ip protocol tcp
   profiles
      ftp
      tcp
}
node * monitor icmp
```