

QUALITY OF SERVICE -ARKKITEHTUURIT

LAHDEN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma

Tietoliikennetekniikka

Opinnäytetyö

Kevät 2009

Timo Taavila

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

TAAVILA, TIMO: Quality of Service -arkkitehtuurit

Tietoliikennetekniikan opinnäytetyö, 73 sivua, 29 liitesivua

Kevät 2009

TIIVISTELMÄ

Opinnäytetyön aiheena on tietoliikenneverkkojen palvelunlaatu ja tietoliikenteen priorisointi Quality of Services -tekniikoilla. Tietoverkkojen siirtonopeus ja suorituskyky määrittävät usein suoraan niiden hyötyasteen. Koska IP-verkkojen fyysisiä ominaisuuksia ei voida loputtomasti, saati kustannustehokkaasti kasvattaa, ovat verkon suorituskykyä parantavat loogiset ratkaisut lisänneet merkitystään. Loogiset ratkaisut ovat usein reitittimien liityntäkohtaisten liikennevirtojen ohjaamista, jolla pyritään muuttamaan normaalia liikennevirtaa jonkin tietyn liikennetyypin eduksi. Tietoverkkojen tarkoituksenmukaista, tiettyjen liikennevirtojen kasvattamista, muokkaamista tai rajoittamista kutsutaan tietoliikenteessä yleisnimityksellä Quality of Service. Cisco Systems tarjoaa erilaisia laitekohtaisia Quality of Service -palveluja, joita hyödynnetään opinnäytetyön käytännön toteutuksessa.

Opinnäytetyössä selvitetään nykypäivän QoS-palvelutarpeet, kapasiteettivaatimukset, skaalautuvuus sekä toteutus jo olemassa olevaan verkkotopologiaan. Samalla käsitellään myös yleistä palvelun käytettävyyttä ja perustellaan miksi QoS on palveluratkaisuna varteenotettava verrattuna pelkän kaistakapasiteetin kasvattamiseen. Quality of Service -tutkimus rajataan Differentiated Services ja Integrated Services -arkkitehtuureihin. Edellämainittujen arkkitehtuurien lisäksi esitellään myös muita QoS-ratkaisuja kuten MPLS-verkkototeutukset, Cisco NBAR, sovellustason Policy Based QoS sekä päätelaitekohtaiset QoS-asetukset.

Opinnäytetyön käytännön toteutuksessa perehdytään sekä Differentiated Services - että Integrated Services -arkkitehtuurien verkkototeutuksiin Cisco-laitteistolla. QoS-arkkitehtuurien soveltuvuutta testilaitteistolle tutkitaan arkkitehtuurikohtaisesti sekä erikseen että yhdistetyssä verkossa. Arkkitehtuureja verrataan keskenään konfiguroinnin helppouden, tehokkuuden ja muokattavuuden näkökulmista. Testialustana toimii kuormitettu verkko jonka yli pyritään kuljettamaan jatkuvaa videokuvasiirtoa mahdollisimman hyvällä palvelunlaadulla. Testien perusteella voidaan todeta Differentiated Servicen parempi palvelunlaatu pienissä verkkokoonpanoissa ja tuodaan esille Integrated Servicen vahvuus suurten verkkojen palveluntoteuttajana. Differentiated Services voi toteuttaa koko verkon QoS-palvelun jopa yhden laitteen avulla. Integrated Services vaatii kaikkien verkon reitittimien konfiguroinnin, mutta laitesignaloinnin ansiosta sen ylläpito on muuttuvissa verkoissa helpompaa.

Avainsanat: QoS, Palvelunlaatu, Differentiated Services, Integrated Services

Lahti University of Applied Sciences
Degree programme in information technology

TAAVILA TIMO: Quality of Service -architectures

Bachelor's thesis in telecommunications, 73 pages, 29 attachments

Spring 2009

ABSTRACT

This bachelor's thesis presents the research of Quality of Service in data networks and shows how to overcome the physical capacity limitations. Bandwidth, efficiency, and the amount of dataflows often determines how efficient networks are. The physical attributes of the network are no longer the key feature in bettering the efficiency itself, rather logical solutions have drawn more attention in the field of telecommunications. These logical solutions are often referred as "Quality of Service" or "QoS". QoS is based on manipulating dataflows for the benefit of chosen applications. Cisco Systems offers both hardware- and software based solutions to QoS. Cisco manufactured devices are used in the conventional research part of this thesis. The main goal of this research is to investigate how quality of service can be configured and used in the consistence of the Cisco devices.

This thesis clarifies the needs of quality of service in today's data networks including capacity, scalability, and configurations. Both QoS-services usability and limitations are tested and also explained why QoS is a considerable solution when dealing with low bandwidth networks. The research is delimited in two main QoS-architectures, Differentiated Services and Integrated Services. Both of these architectures have gained a foothold as leading QoS-mechanisms in today's data networks. This thesis presents also QoS-knowledge related to MPLS-networks, Cisco NBAR-techniques, application policy Based QoS and QoS-settings for individual computers.

The conventional research of the architectures is done with the Cisco's devices. QoS-architectures suitability is tested both individually and together in the test network. The basics of the conventional research is to generate traffic in the test network increasing the congestion. The increased and harmful congestion is overcome with the QoS-service techniques that allow the critical dataflows such as videostreaming to use effectively the resources of the network. The Architectures are compared against each other with the characteristics such as easiness in configuration, efficiency and formability. The tests prove that Differentiated Services is better in small networks and the Integrated Services applies better in large networks. Differentiated Services can offer effective QoS-service for the whole network in only one device. Integrated Services demands configurations in each of the devices in the network, but can be more easily managed thanks to the signaling mechanisms in the devices.

Key words: Quality of Service, Differentiated Services, Integrated Services

SISÄLTÖ

1 JOHDANTO	1
1.1 Työn tausta	1
1.2 Työn tavoitteet	2
2 QUALITY OF SERVICE TIETOLIIKENTEESSÄ	4
2.1 Palvelun laatu	4
2.2 Quality of Servicen haasteet	7
2.3 VoIP ja multimediasovellukset	9
3 QUALITY OF SERVICEN PÄÄARKKITEHTUURIT	12
3.1 OSI-malli ja QoS	12
3.2 Differentiated Services	14
3.3 Integrated Services	17
3.4 Muut palvelunlaatutekniikat	21
4 CISCO JA QUALITY OF SERVICE	26
4.1 Cisco Systems ja QoS-sovellusten tuki	26
4.2 Cisco NBAR	28
4.3 Differentiated Services ja DSCP reitittimessä	30
4.4 Integrated Services ja RSVP reitittimessä	33
4.5 Class of Service (CoS) ja Auto-QoS	37
4.6 Verkkosuunnittelu ja laitevalinnat	41

5 QOS-TOTEUTUS CISCON LAITTEILLA	43
5.1 Testiympäristön kuvaus	43
5.2 Verkkotopologia	44
5.3 Differentiated Services -toteutus	49
5.3.1 Reititinkonfiguraatiot	49
5.3.2 Kytinkonfiguraatiot	53
5.4 Integrated Services -toteutus	60
5.5 Toteutusten vertailu	69
6 YHTEENVETO	71
LÄHTEET	74
LIITTEET	

LYHENTEET

ATM	Asynchronous Transfer Mode. Asynkroninen tiedonsiirtotapa, joka jakaa datan vakiomittaisiin 53 tavun soluihin. Käyttää kiinteää yhteyttä lähettäjän ja vastaanottajan välillä hyödyntäen verkkototeutuksissa erillisiä ATM-kytkimiä.
CoS	Class of Service. 3-bittinen kenttä siirtoyhteyserroksen Ethernet-kehyksen otsikkokentässä. Mahdollistaa kahdeksanportaisen prioriteettiarvoasteikon (0-7).
DSCP	Differentiated Services Code Point. IP-paketin otsikkokentässä sijaitseva 6-bittinen tietokenttä. Määrää QoS-palvelussa palveluluokkien jaottelun. Käytössä yleisesti Differentiated Services -arkkitehtuurin loogisissa operaatioissa.
Ethernet	Pakettipohjainen lähiverkkotekniikka. Ensimmäisenä laajasti hyväksytty ja nykyään yleisin lähiverkkotekniikka.
FTP	File Transfer Protocol. TCP-protokollaa käyttävä tiedostonsiirtomenetelmä kahden päätelaitteen välillä.
http	Hypertext Transfer Protocol. Protokolla, jota selaimet ja www-palvelimet käyttävät tiedonsiirtoon.
IP	Internet Protocol. TCP/IP-mallin verkkokerroksen protokolla, joka huolehtii tietoliikennepakettien toimittamisesta perille pakettikytkentäisessä verkossa
MOS	Mean Opinion Score. Subjektiiivinen, viisiportainen mittaustekniikka VoIP-äänienlaadulle, jossa käytetään testihenkilöjä verkkomittarien sijasta.
Multicast	Ryhmälähetys. Siirtotekniikka jossa lähetetään yhdeltä lähettäjältä samanaikaisesti monelle vastaanottajalle.
NBAR	Network Based Application Recognition. Ciscon reitittimissä ja kytkimissä käytettävä mekanismi, joka tunnistaa datavuon sen ensimmäisten vastaanotettujen pakettien perusteella.
OSI-malli	Open Systems Interconnection Reference Model. Esittää tiedonsiirtoprotokollat seitsemässä kerroksessa. Kukin kerroksista käyttää alla olevan kerroksen palveluja ja tarjoaa palveluja ylemmälle kerrokselle.
OSPF	Open Shortest Path First. Reititysprotokolla joka vertailee Dijkstran algoritmilla verkkoliityntöjen nopeuksia optimaalisimman reitin löytämiseksi.

RIP	Routing Information Protocol. Laajalti lähiverkkoreitityksessä käytetty sisäinen reititysprotokolla, joka käyttää reitin laskemiseen hyppymäärää laitteelta toiselle.
RSPEC	Request specification. Määrittää Integrated Service -toteutuksissa palveluluokitukset ja minimipalvelutason.
RSVP	Resource Reservation Protocol. Kuljetuskerrostason protokolla, jonka tehtävänä on varata verkkoresursseja priorisoidulle verkko- liikenteelle Integrated Services -arkkitehtuurissa.
RTSP	Real Time Streaming Protocol. Tiedonsiirto-protokolla multimedian suoratoistoa varten IP-verkoissa.
Striimaus	Videokuvan ja äänen reaaliaikainen siirto päätelaitteelta toiselle. Mahdollistaa videon katselun häiriöttä jo siirron aikana.
SQL	Structured Query Language. IBM:n kehittämä standardoitu kyselykieli, jolla relaatiotietokantaan voi tehdä erilaisia hakuja, muutoksia ja lisäyksiä.
TSPEC	Traffic specification. Määrittää Integrated Service -toteutuksissa kaistanvarauksen.
UDP	User Datagram Protocol. Yhteyksikäytäntö, jolla ohjelma tai sovellus voi lähettää viestejä verkon yli. Yhteydetön protokolla joka ei tue pakettien kohteeseen saapumisen tarkistusta.
Unicast	Siirtotekniikka, jossa lähetetään yhdeltä lähettäjältä yhdelle vastaanottajalle. Yleisin siirtomekanismi tietoverkoissa.
QoS	Quality of Service. Termi jolla yleisesti tarkoitetaan tietoliikenteen luokittelua, rajoittamista ja priorisointia
TCP	Transmission Control Protocol. Kuljetuskerroksen yhteydellinen protokolla, jolla luodaan yhteyksiä verkon päätelaitteiden välille. Mahdollistaa pakettien perillemenon tarkistukset ja uudelleenlähetykset.
TCP/IP	Transmission Control Protocol / Internet Protocol. Protokollapino, jota käytetään yleisesti kaikissa tietoverkoissa. Nelikerroksinen hierarkia tärkeimmistä tietoliikenneprotokollista.
ToS	Type of Service. 8-bittinen kenttä IP-paketin otsikkokentässä. Määrittää Quality of Servicen palveluluokat. Differentiated Services -arkkitehtuurissa korvattu suoraan DSCP-kentällä.
VoIP	Voice over IP. Yleistermi tekniikalle, jonka avulla ääntä ja videokuvaa voidaan siirtää reaaliaikaisesti tietoverkoissa.

1 JOHDANTO

1.1 Työn tausta

Tässä opinnäytetyössä tutkitaan tietoliikennepriorisoinnin ominaisuuksia, mahdollisuuksia ja toteutustyökaluja. Työn yksikäsitteiseksi aiheeksi valittiin Quality of Service, joka terminä käsittää hyvin moninaisia tietoverkkojen liikennepriorisointitekniikoita. Quality of Service Differentiated Services - sekä Integrated Services -arkkitehtuuritoteutukset ovat laajalti käytössä nykypäivän tietoverkoissa. Niiden avulla voidaan paitsi lisätä verkkojen tehokkuutta, myös saada aikaiseksi kustannussäästöjä vähentämällä verkon kriittisten pisteiden fyysisiä kapasiteettiinvestointeja. Operaattorit tarjoavat Quality of Service -palvelua parantamaan tiettyjen liikennetyyppien siirron sujuvuutta asiakasverkoissa. Palvelun laadun tasosta voidaan veloitaa sen mukaan, miten paljon kaistaa tai kapasiteettia kukin priorisoi liikenne vie. Yritysverkoissa QoS voi taata kriittiset palvelut takaamalla esimerkiksi toimivat VoIP-palvelut rakennuksessa tai toimistojen välillä. QoS tähänkin toimivien sovellusten ja sujuvan liikenteen takaamiseen tietoverkoissa.

Valitsin työn aiheen, koska Quality of Service on tekniikkana monimutkainen mutta tarjoaa lukuisia työkaluja niin verkkosuunnitteluun kuin toteutukseenkin. Verkon tehokkuutta voidaan lisätä fyysisten ominaisuuksien, kuten laitteiston ja kaapeloinnin osalta merkittävästikin, mutta palvelun laatuun tämä ei välttämättä vaikuta. Kaikkein mielenkiintoisin ominaisuus Quality of Service -tekniikoissa onkin verkon resurssien optimointi tasolle, joka voi vastata palvelutasoiltaan fyysisesti paljon tehokkaampia verkkoja. Tämä voidaan tehdä pelkillä laitteistokonfiguraatioilla, ilman merkittäviä rahallisia investointeja itse laitteistoon. Aihevalintaa tuki myös se, että Lahden ammattikorkeakoulun tietoverkkolaboratorion laiteresursseja oli mahdollista hyödyntää tämän opinnäytetyön tukena. Näin varmistuttiin siitä, että opinnäytetyö ei jäisi pelkästään teorialuokkimukseksi.

Ennen työn aloittamista selvitettiin tarkasti, minkälaisia toteutusmahdollisuuksia valituilla kahdella QoS-arkkitehtuurilla on Cisco-laitteistoilla. Oletusarvoisesti tietoliikennelaitevalmistajien laitteistosta tulisi löytyä mekanismeja sekä Differentiated Services - että Integrated Services -toteutukseen tai edes jonkintasoisen paikallisen QoS-palvelun toteutukseen. Koska Ciscon laitteistosta löytyi tuki DSCP-toteutukseen Differentiated Servicen tarpeisiin ja RSVP-protokollan tuki Integrated Servicen käyttöön, olivat lähtökohdat opinnäytetyön toteutukseen kunnossa. Suurin kysymys olikin, miten monimutkaisia saati toimivia QoS-ratkaisuja näillä laitteilla saataisiin aikaiseksi.

1.2 Työn tavoitteet

Tämän opinnäytetyön tavoitteena on perehdyttää lukija Quality of Servicen erilaisiin toteutustapoihin ja niiden merkitykseen nykyaikaisissa tietoverkoissa. Toisena tärkeänä tavoitteena on erilaisten QoS-toteutusmekanismien ja laitteistokohtaisten konfiguraatioiden toteutus ja esittely. Työssä keskitytään Quality of Servicen arkkitehtuuritoteutuksiin Ciscon (Cisco Systems) laitepohjalla. QoS-palvelun käsitteet, toiminta ja mahdollisuudet tullaan käsittelemään teoriapohjaisesti perehdyttämällä lukija tarkemmin verkkotason protokollatoteutukseen. Erilaiset liikennetyypit ja niiden kriittisyys sekä vaadittavat minimipalvelutasot liittyvät myös kiinteästi toimivan QoS-palvelun suunnitteluun. Quality of Servicen riittävä ymmärtäminen teoriatasolla on lukijan kannalta välttämätöntä käytännön toteutuksia ja konfiguraatioita esiteltäessä.

QoS-palveluratkaisuja lähestytään kahden eri arkkitehtuurin näkökulmasta. Nämä arkkitehtuurit ovat Differentiated Services ja Integrated Services, joista ensimmäinen on yleiskäyttöisempi lähiverkkototeutuksissa kun taas jälkimmäinen soveltuu paremmin laajoille reititinverkoille. Tässä opinnäytetyössä tullaan vertailemaan näitä kahta arkkitehtuuria toisiinsa ja tutkimaan, kumpiko niistä on käytökelpoisempi tehokkuuden, konfiguroinnin ja ylläpidon näkökulmasta. Työssä tuodaan esiin vertailukohtia eri arkkitehtuurien välillä sekä esitetään kriteerejä tekniikkakohtaisiin valintaperusteisiin mahdollista uutta QoS-toteutusta suunnitel-

taessa tai jo olemassa olevaa muokattaessa. Lisäksi tuodaan esille myös heikkouksia ja kehityskohteita arkkitehtuuritoteutusten osalta sekä tutkitaan tekniikoiden yhteensopivuutta ja toimintaa samassa verkkokokoonpanossa.

Opinnäytetyön keskeisenä osana on käytännön osuus, jossa esitellään Differentiated Services ja Integrated Services -arkkitehtuurien toteutusta testiverkossa. Testiverkko tulee käsittämään viiden reitittimen ja neljän kytkimen verkkokokoonpanon, jossa toteutetaan sekä Differentiated Services- ja Integrated Services -palvelu. Arkkitehtuurikohtaisen tutkimuksen tavoite on esittää verkkototeutus, jossa kuormitetun testiverkon häiriöt palvelunlaatuvaatimuksiltaan kriittiselle liikenteelle eliminoidaan QoS-palvelulla. Testeissä kriittisenä liikenteenä tullaan käyttämään reaaliaikaista videokuvansiirtoa (videostriimaus). Käytännössä verkkoa pyritään kuormittamaan FTP-liikenteellä pisteeseen, jossa sen läpi ei voida enää siirtää striimattua liikennettä. Verkon tukkeutumisesta pyritään toipumaan DSCP- ja RSVP-toteutusmekanismeilla ja mahdollistamaan striimatun liikenteen siirto syrjäyttämällä jatkuva FTP-liikenne verkossa. Toimiva testiverkko pyritään rakentamaan käyttäen pelkästään Ciscon laitteita.

Opinnäytetyön tavoitteena on todistaa, että ilman QoS-toteutusta testiverkko olisi käyttökelvoton minimipalvelunlaadun vaatimien liikennetyyppien osalta. Tukkeutuneessa testiverkossa videokuvan striimaus pyritään toteuttamaan laadullisesti hyvänä sekä Differentiated Services - että Integrated Services -verkkototeutuksissa. Tutkimuksen suurin haaste on verkkoa kuristavien pullonkaulojen tehokas eliminointi QoS-työkaluilla, jolloin on mahdollista taata hyvä suorituskyky verkon päästä päähän. Tähän haasteeseen pyritään vastaamaan molempien arkkitehtuurien osalta mahdollisimman tehokkaasti ja löytämään ne konfiguraatoratkaisut, jotka tuovat parhaat lopputulokset.

2 QUALITY OF SERVICE TIETOLIIKENTEESSÄ

2.1 Palvelun laatu

Quality of Service tai lyhyemmin ”QoS”, on tietoliikenteessä käytetty termi jolla viitataan tietoliikenneominaisuuksien kuten verkkoliikenteen ja palvelujen priorisointiin ja säännöstelyyn käyttäjien kesken. QoS:n tarkoituksena on usein taata siedettävä palvelutaso riittävän datasiirron osalta. QoS liittyy myös suoraan verkon käytön jouhevuuteen koska tilanteesta riippuen kaikkia verkon palveluja ei voida ajaa täydellä teholla kaistakapasiteetin rajallisuuden vuoksi. Jotkin palvelut kuten VoIP, videokonferenssisovellukset sekä striimaussovellukset vaativat tietyn minimikaistan toimiakseen. Käyttäjien kannalta QoS ei siis ota pelkästään kantaa käyttömukavuuteen tai verkon tehokkuuteen vaan takaa paitsi resurssit kaikille käyttäjille myös työkalut ylikuormituksen ja vikatilanteiden ennaltaehkäisemiseen. Kuitenkin jos verkon tukkeutumisesta ei ole vaaraa, QoS ei myöskään paranna sen suorituskykyä. Kyse on siis kaistan optimaalisen käytön takaamisesta ”turhan liikenteen” kustannuksella. (Cisco Systems Incorporated 2008a; Wikipedia 2008a.)

Quality of Service voidaan jakaa palvelutasoiltaan kahteen alakategoriaan: palveluluokkaan (Class of Service, CoS) sekä palvelutyypin (Type of Service, ToS). Palveluluokilla voidaan määrittää sovelluskohtaisesti kaistanleveyksiä ja viiveitä verkossa. Käytännössä tämä tehdään Ethernet-kehyksen otsikkokentän kolmen CoS-bitin avulla (7 erilaista palveluluokkaa). Eri liikennetyypit kategorisoidaan palveluluokkien sisään, jolloin tietoliikennevaatimuksiltaan samankaltainen liikenne voidaan käsitellä yhtenä ryhmänä. Palvelutyyppi (ToS) puolestaan sijaitsee loogisena muuttujana IP-otsikossa. Palvelutyyppi aktivoi palveluluokkaoperaatiot verkon laitteissa toimien ikäänkuin ohjaustyökaluna QoS-toteutuksissa. Palvelutyyppi pystyy määrittämään IP-pakettien tunnustuskentässä vähintään kahdeksan eri pakettityyppiryhmää eli palveluluokkaa kolmen bitin kentällä. Nykyään on kuitenkin yleisesti käytössä 6-bitin palvelutyyppikenttä, joka mahdollistaa huomattavasti laajemman ja tarkemman kategorisoinnin eri liikennetyypeille luokan sisäisen priorisoinnin avulla. (Davidson & Peters 2002, 189.)

Kuten edellisessä kappaleessa todettiin, Type of Service (ToS) toimii liikenteen ohjaustyökaluna QoS-verkkototeutuksissa. Tällöin se on vahvasti sidoksissa reititykseen, joka usein vaikuttaa myös suoraan palvelutasoon. ToS määrittää kaistanleveysvaateet, viiveiden minimoinnin ja pakettien perillemenon luotettavuuden minimoiden pakettien tiputukset reitittimissä. Reititysprotokollat sen sijaan reitittävät liikenteen lyhimmän reitin mukaan (RIP) tai suoraan liittymän nopeuden mukaan (OSPF). Ristiriitaisuuksia saattaa ilmetä tapauksissa joissa kriittisen liikenteen paketeilla on ToS-merkintä ”minimoi viive”. Tällöin paketit reititetään minimiviiveen omaavan linkin kautta, vaikka reititysprotokolla haluaisi siirtää paketit maksimaalisen kaistan omaavan liittymän kautta. Halutun QoS-palvelutason kannalta maksimikaistan omaava reitti saattaakin olla parempi vaihtoehto kriittisen liikenteen siirrolle, hieman suuremmasta viiveestä huolimatta. Minimiviiveen omaavat liittymät eivät välttämättä sovellu hyvin kaistanleveysiltään kriittisen liikenteen siirtoon. Tällöin onkin järkevää sivuuttaa ToS-kenttä reitityksessä asettamalla sen 3 bittiä arvoon 0. Esimerkki osoittaa kuinka tärkeää on tuntee itse verkkotopologian fyysiset siirto-ominaisuudet ja käytetyt reititysprotokollat QoS-palvelua suunniteltaessa. (Miller & Cummins 2000, 598-599.)

Quality of Service voidaan mitata verkkoliikennemittareilla tai protokollanalyysaattoreilla esimerkiksi suoraan sitä kontrolloivista reititinsolmukohdista. Useimmiten tuloksia verrataan suoraan laitekonfiguraatioissa määriteltyihin sääntöihin tai asetuksiin QoS-palvelun osalta. QoS:n tehokkuus voidaan siis suoraan todeta pelkästään ylläpitäjän näkökulmasta. QoE (Quality of Experience) puolestaan määrittää käyttäjäkohtaisen kokemuksen, joka voi olla hyvinkin subjektiivinen ja harhaanjohtava. Quality of Service -palvelu tulisi siis suunnitella ja perustella tarkkoihin mittaustuloksiin nojaten, keskitetyn hallinnon ja ylläpidon taholta. (Cisco Systems Incorporated 2008a; Wikipedia 2008a.)

QoS-palvelu toteutetaan verkon aktiivilaitteissa kuten reitittimissä, kytkimissä ja palvelimissa. Verkon solmukohdissa sijaitsevat QoS:stä huolehtivat laitteet takaavat tehokkaan ja keskitetysti ylläpidettävän palvelunhallinnan ylläpitäjän näkökulmasta. Jossakin tapauksissa voidaan ottaa kantaa suoraan palvelinresurssien priorisoimiseen esimerkiksi käytettävien sovellusten ja palveluiden osalta. Vapaan liikenteen ”best effort” -verkkoihin verrattuna QoS:ää käyttävät verkot ovat huo-

mattavasti tehokkaampia, varsinkin jos verkossa on paljon kuormapiikkejä. QoS vaatii kuitenkin jatkuvaa monitorointia ja ylläpitoa, varsinkin jos verkon jaettavat resurssit ja palvelut muuttuvat tai verkko kasvaa. (Cisco Systems Incorporated 2008a; Wikipedia 2008a.)

Quality of Servicen Differentiated Services - ja Integrated Services - arkkitehtuurien loogiset prioriteettisäännöt vaikuttavat pääasiassa verkon solmukohdissa. Usein solmukohdat sijaitsevat juuri verkon reitittimisissä, siellä missä varsinainen liikenteen reititys tapahtuu. Verkon reunalla, lähempänä asiakasta sijaitsevat kytkimet ja reitittimet voivat myös toteuttaa QoS-palvelua. Tällaiset aktiivilaitteet voivat olla asiakkaan vastuulla, jolloin QoS-toteutukset jäävät asiakkaan omalle vastuulle. Yritystapauksissa tehokkaita QoS-toteutustapauksia on usein mahdollista toteuttaa sekä palvelu- että liikennetyyppiakohtaisesti. Toisaalta reunaverkon laitteet voivat olla myös operaattorin ja asiakkaan palvelutasosopimuksen (SLA, Service Level Agreement) piirissä, jolloin asiakas maksaa tietystä palvelutasosta esimerkiksi VoIP-liikenteen suhteen. Tällöin esimerkiksi yrityksessä VoIP:ia käyttävä asiakas saa verkkoonsa sopimuksen tietyllä palvelutasolla takaamat IP-puheyhteydet ja maksaa tästä palvelusta erikseen lisähinnan. Palvelutasot voivat olla esimerkiksi ”Gold”, ”Silver” tai ”Bronze” joista ”Gold” takaa parhaimman tason. Quality of Service onkin usein lisäarvoa tuova palvelu, joka hinnoitellaan erikseen. (Nortel Networks 2008; Wustenhoff 2002.)

Jos operaattori tarjoaa QoS-palvelua, toteutetaan se usein pelkästään reititintasolla. Tämä johtuu yksinkertaisesti siitä, että palvelun looginen toteutus, integrointi ja toimivuus ovat usein parempia ja luotettavampia reitittimisessä, joka muutoinkin hoitaa verkkotason liikenteen käsittelyä. Kytkimiä voidaan puolestaan käyttää täydentämään tätä arkkitehtuuria ja lisäämään palvelun luotettavuutta jokaisessa verkon solmussa. Kytkimet voivat myös monipuolistaa priorisointisäännöstöä esimerkiksi käyttäen hyväksi Ethernetin peruserroksella porttikohtaisia säännöstöjä tai VLAN-tunnisteita. QoS-palvelun tuomat etuudet voidaan kuitenkin korvata kasvattamalla fyysisen verkon kapasiteettia. Ratkaiseva kysymys kuuluukin: tulee halvemmaksi lisätä liikennekapasiteettia vai implementoida QoS-palvelu jo olemassa olevaan verkkoon? (Nortel Networks 2008.)

2.2 Quality of Servicen haasteet

Nykyisissä pakettikytkentäisissä verkoissa kuljetetaan yhä moninaisempaa data-liikennettä eri protokollien, kuten IP, TCP, FTP, HTTP, VoIP ja RTSP toimesta. Uudet, tietoliikennekaistaa runsaasti vievät palvelut, kuten IPTV ja Internetin reaaliaikaiset TV- ja videopalvelut datastriimauksineen nostavat paikoitellen verkon käyttöastetta jopa niin paljon, että verkon tukkeutuminen on mahdollista. Myös yleistyneet Peer-to-Peer verkkoyhteydet käyttävät usein hetkellisesti huomattavat määrät siirtoyhteykskapasiteettia muun liikenteen kustannuksella. Internetin kompleksisuus, laajat verkkokokonaisuudet sekä operaattorimisteiset verkot vaikeuttavat tietoliikenteen tehokasta kontrollointia, joten kapasiteetin kasvattaminen itse siirtojohdoissa on usein helpoin ratkaisu. Sen sijaan pienemmissä verkoissa kuten yritysverkoissa, voidaan saada suuriakin säästöjä aikaiseksi pystyttämällä verkkoon loogisesti järkevä ja skaalautuva QoS-palvelu. Yritysten välisillä yhteyksillä QoS-toteutus on varsin mahdollista, mutta silloin lähettäjällä täytyy olla laitteisto tai työkalut palvelun toteutukseen sekä keinot varmistaa palvelutason säilyminen koko verkon matkalla. Luonnollisesti tämä on erittäin haastavaa internetin yli kulkevilla yhteyksillä, jossa liikennereitit eivät ole tiedossa. (Vonage Forum 2009; Wikipedia 2008a.)

Yritysverkon käyttötarpeet ovat usein hyvinkin tarkkaan tiedossa, joten QoS-säännöstön (QoS Policy) suunnittelu on paljon helpompaa kuin internetin kaltaisessa verkossa. Turhan tai ei kiireellisen liikenteen karsiminen johtaa usein siihen, että kriittinen liikenne kulkee häiriöttä. Tämä antaa helposti käyttäjänäkökulmasta tunteen, että verkon kapasiteetti olisi suurempi. Palvelut siis toimivat paremmin ja pienemmillä viiveillä, mikä voidaan todeta myös protokolla-analysaattoreilla tai verkkoliikennemittauksilla. Tärkeiden ja vähemmän tärkeiden palvelujen määrittäminen voi olla täysin subjektiivinen tai tapauskohtainen kysymys. Esimerkiksi perinteisesti QoS:n kriittisiin palveluihin yhdistetyt VoIP ja videokuvan siirto eivät välttämättä aina ole ne palvelut, jotka tarvitsevat suurimman prioriteetin. Vaihtoehtona mainittakoon erilaiset SSH-ylläpitoyhteydet tai tietoverkkoihin integroidut hälytys- ja valvontajärjestelmät. QoS:llä voidaan myös lisätä käyttömukavuutta erilaisissa reaaliaikaisissa, paljon verkkoresursseja käyttävissä online-pelisovelluksissa, joissa verkon viiveet hidastavat peliä ja alentavat pelikokemuk-

sen mielekkyyttä. Alan harrastajien keskuudessa viiveilmiöstä puhutaan myös termillä ”lagiminen”. (Vonage Forum 2009; Nortel Networks 2008; Wikipedia 2008a.)

Quality of Servicen peruspriorisointi on esitetty kuviossa 1 (KUVIO 1) Malli esittää QoS-priorisointia. Suurin luku 7 esittää suurinta prioriteettia, ja vastaavasti luku 0 pienintä. Tämä malli ei kuitenkaan päde kaikissa tapauksissa, ja siksi ylläpidon olisi tärkeää mitata ja analysoida verkon kuormitusta ja liikennetyyppejä, varsinkin verkon käyttötarpeiden muuttuessa. ”Best Effort”-liikenne pienimmällä ja ylläpito suurimmalla prioriteetilla ovat kuitenkin hyvä nyrkkisääntö QoS-priorisoinnille. Kaikki, mitä jää tämän väliin, on täysin vapaavalintaista ja tapauskohtaista. Hyvänä esimerkkinä tästä on VoIP-liikenne, joka tarvitsee suurta prioriteettia (prioriteetti 5). Jos mitään hyötykäyttöä VoIP-liikenteelle ei ole voidaan se sisällyttää Best-effort-liikenteeseen. Tämä tosin tarkoittaa sitä että tällaisessa verkossa VoIP on lähes käyttökelvoton sovellus. Jo rakennettua QoS-arkkitehtuuria tarvitsee harvoin muuttaa, jos käyttäjäkanta pysyy samana. Jos verkkoa kuitenkin kasvatetaan, olisi ensiarvoisen tärkeää tarkistaa uudelleen QoS-palvelutarpeet ja mahdollinen skaalautuvuus. (Sun Microsystems 2009; Wikipedia 2008a.)

Priority Level	Traffic Type
0	Best Effort
1	Background
2	Standard (Spare)
3	Excellent Load (Business Critical)
4	Controlled Load (Streaming Multimedia)
5	Voice and Video (Interactive Media and Voice) [Less than 100ms latency and jitter]
6	Layer 3 Network Control Reserved Traffic [Less than 10ms latency and jitter]
7	Layer 2 Network Control Reserved Traffic [Lowest latency and jitter]

KUVIO 1. QoS-liikenneprioriteetit (Wikipedia 2008a).

2.3 VoIP ja multimediasovellukset

Kuten jo aikaisemmin todettiin, tietoverkoissa on liikennetyyppejä, jotka tarvitsevat toimiakseen tietyn palvelutason ja kapasiteettivaruksen. Paras ja ehkä tunnetuin esimerkki tästä on VoIP, joka on yleistynyt niin lähiverkoissa kuin internetissäkin rajusti viime vuosikymmenen aikana. VoIP-palvelun laatua on vaikea mitata protokolla- tai pakettitasolla, koska viime kädessä palvelun laadun kokee käyttäjä. Kysymys kuuluukin: onko äänenlaatu kuulijan mielestä kelvollinen ja viesti ymmärrettävissä? Laatua voidaan mitata pelkästään käyttäjäkokemusten perusteella, MOS-mittauksella. Joukko kuuntelijoita arvioi äänenlaatua, joka on tuotettu tietyllä koodekillä. Äänen laadulle annetaan arvosana väliltä 1-5. Erilaiset MOS-arvot

ja niiden merkitykset on esitetty kuviossa 2 (KUVIO 2) Koska palvelun laadun heikkeneminen havaitaan hyvin herkästi jo käyttökokemuksen perusteella ja laadun heikkeneminen on suoraan sidoksissa palvelun toimimiseen, on VoIP erittäin kriittinen sovellus QoS-arkkitehtuureissa. (About.com 2008; Wikipedia 2008b.)

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

KUVIO 2. Mean Opinion Score-arvot (Wikipedia 2008g).

Jos verrataan VoIP-tekniikkaa FTP- ja HTTP-protokolliin, ovat jälkimmäisenä mainitut huomattavasti joustavampia. Nettisivut aukeavat viiveelläkin ja tiedostot siirtyvät niin kauan, kun verkko ei ole tukossa tai käyttökelvoton. Näiden palvelujen kustannuksella voidaan siis harjoittaa priorisointia tarpeen vaatiessa. Toinen kasvava palveluntarve dataverkoissa liittyy liikkuvan kuvan siirtämiseen internet- ja lähiverkoissa. Pienet videotiedostot voidaan siirtää verkon yli FTP- protokollaa käyttäen vaikkapa palvelimelta kotikoneelle. Nykyään kuitenkin tarve siirtää yhä pidempiä leikkeitä tai jopa elokuvia hyvällä kuvanlaadulla kasvattaa tiedostokokoja niin suuresti, ettei perinteinen FTP-tiedostosiirto ole käyttäjän kannalta kovin mielekästä odotusajan kasvaessa usein yli sietokyvyn. Striimaustekniikat, tarkemmin sanoen suoratoisto, sallivat videokuvan katselun samalla hetkellä, kun siirto käynnistetään. Videota voidaan siis katsella, vaikka koko tiedosto ei ole tullut palvelimelta kokonaisuudessaan käyttäjälle. Striimausta käytetään yhä enemmän web-sivujen videokomponenteissa ja internet-TV-sovelluksissa multicast-siirtotekniikalla. (Kennedy 1999.)

Yrityksissä käytetyt videokonferenssi-istunnot vaativat myös tehokkaan reaaliaikaisen videokuvansiirron. Jos yritysverkossa on verkkoliikennettä vähääkään hidastavia pullonkauloja, vaikkapa toimistoverkkojen välissä, ovat häiriöt todennäköisiä. Striimaus- ja videosiirtotekniikoita kehitetään toki jatkuvasti, mutta ne vaativat usein myös tehokkaan QoS-tuen varsinkin kapasiteetiltaan rajallisissa verkoissa. Kuten VoIP-sovelluksia, myös videokuvan laatua ja QoS-toimivuutta voidaan havainnoida suoraan käyttäjäkokemusten kautta. Äänen lisäksi videokuvan siirto voidaankin luokitella kriittiseksi palveluksi, koska tekniikka vaatii tietyn minimikapasiteetin ja minimiviiveen toimiakseen. (Dominach 2009.)

Quality of Servicen yhteydessä VoIP- ja videotekniikoissa on huomattava pieni tekniikkakohtainen eroavaisuus. Videokuvan siirto verkon yli voidaan toteuttaa takaamalla aktiivilaitteilla tasainen minimisiirtonopeus ilman varautumista piikkiarvoihin. Resurssinvaraus tehdään siis vaikkapa tietty kehymäärä sekunnissa -periaatteella. VoIP-sovelluksissa kehysten siirrot ovat purskeisia koska puheessa on yleensä satunnaisia katkoja. Vastaavasti verkon täytyy myös varautua tilanteisiin, joissa keskustelu puolin ja toisin on varsin aktiivista. Tällöin siirrettyjen kehysten määrä kasvaa rajusti aikayksikköä kohti. QoS-palvelulla voidaan siis varata tietty kapasiteetti, tietty kehymäärä per sekunti -säännöstöllä, mutta huomioon on otettava varaus myös piikkiarvoille. Piikkiarvoja varten varataan kaistaa ikäänkuin reserviin, josta sitä käytetään ruuhkatilanteissa. Sekä VoIP, että videosovellukset ovat niin sanotusti inelastisia palveluita. Tämä tarkoittaa QoS:n kannalta sitä, että niillä on minimivaateet niin kaistan, viiveiden kuin luotettavuudenkin suhteen datasiirroissa. Inelastisten palveluiden tarpeita Quality of Servicen osalta käsitellään tarkemmin luvussa 3., Integrated Services- arkkitehtuurin yhteydessä. (VoIP-Info.org 2008; WiredRed 2008.)

3 QUALITY OF SERVICEN PÄÄARKKITEHTUURIT

3.1 OSI-malli ja QoS

Ennen tarkempaa perehtymistä Quality of Servicen pääarkkitehtuureihin ja pääprotokoliin, on hyvä tarkentaa palvelun kerrossuhteita OSI-malliin. Tietoverkot koostuvat usein monimutkaisista, useiden laitevalmistajien laitekoonpanoja sisältävistä solmupisteistä, jotka tarvitsevat toimiakseen yhtenevät säännöt. OSI-malli määrää tarkemmin protokollien ja laitteiden välisen kanssakäymisen mahdollistaen esimerkiksi uusien protokollien ja QoS-toteutusten liittämisen osaksi hierarkiaa. QoS-toteutusarkkitehtuurit ulottuvat mekanismeiltaan ja operaatioiltaan usein OSI-mallin ylimmälle kerrokselle, sovelluskerrokselle asti, joten priorisoidun datan kapselointi aina alimmalle kerrokselle käsittää koko OSI-mallikerrosten kirjon. Kuviossa 3 (KUVIO 3) on esitetty OSI-mallin seitsemän kerrosta. (Chappell 1999, 12.)



KUVIO 3. OSI-malli (Wikipedia 2009).

Reititys, joka sijaitsee OSI-viitemallin verkkokerroksella, on keskeisesti sidoksissa QoS-palvelun toteutukseen jättäen usein muut kerrokset toisarvoisiksi. On kuitenkin muistettava että pakettiprosessoinnin ja itse palveluun osallistuvien protokollien osalta myös kuljetus- ja siirtoyhteyskerrosten rooli on huomattava. MAC-taso (OSI-2) sisältää usein Quality of Serviceen osallistuvien kytkinten ToS (Type of Service) tietokentät, jotka yhdistävät liikennetyypit ylempään verkkokerrokseen. Kytkintason ToS erottelee liikenteen jonoihin, joilla on tietty kaistanvaraus. Näin ollen se sisältää usein vain QoS-perustason operaatiot. Integrated Services -arkkitehtuurin yhteydessä käytettävä RSVP-protokolla nojaa toimintaansa verkkokerroksen lisäksi myös kuljetuskerroksen palveluihin. NBAR-protokolla puolestaan priorisoi liikennettä ylimmän, sovelluskerroksen protokolla- ja liikennetunnistusmekanismien avulla. (Chappell 1999, 12; Cisco Systems Incorporated 2008d; Cisco Systems Incorporated 2008e.)

Tietoliikennetoimintojen jakamista monikerroksiseen hierarkiaan on perusteltavissa myös QoS:n näkökulmasta. Suurempien verkkokokonaisuuksien jakaminen pienempiin helpommin hallittaviin osiin on usein avain QoS:n tehokkaaseen hallintaan. Laitevalmistajariippumattomuus on myös välttämätöntä jo pelkästään palvelun skaalautuvuuden mahdollistamiseksi. Lisäksi kerrosten välinen riippumattomuus helpottaa QoS-kehitystyötä, koska palvelun osia voidaan kehittää ja lisätä kerroskohtaisesti joutumatta muuttamaan alempien kerrosten toteutusta. Quality of Service sisältää palvelutoteutuksena samoja verkko-operaatioita kuin mitkä tahansa muut verkon palvelut, siitäkin huolimatta, että QoS sijaitsee usein itse verkkoon integroituna. Tästä johtuen QoS-protokollat, signaalointi, merkinanto ja reititys noudattavat kaikessa toiminnassaan OSI-mallin kerrosjakoa. (Chappell 1999, 12.)

3.2 Differentiated Services

Differentiated Services tai lyhyemmin DiffServ, on yleisesti käytössä oleva QoS-palvelua toteuttava arkkitehtuuri moderneissa IP-verkoissa. Sen päätehtävät QoS:n osalta ovat eliminoida verkkoviihteitä, taata kriittiset palvelut ja niiden laatu, sekä tarjota pääsy verkkoon myös ei tärkeille sovelluksille ”best effort”- palvelumallilla. Differentiated Servicesin ominaispiirteitä ovat skaalautuvat, ”koruttomat” verkkototeutukset. Tällä tarkoitetaan arkkitehtuurin kykyä toteuttaa koko verkon palvelua yhdessä verkon solmupisteessä ilman signaalointimekanismeja muiden laitteiden kanssa. Differentiated Services -protokollat toimivat kiinteästi reitityksestä vastaavalla verkkokerroksella, OSI-mallin 3. kerroksella. Nykyisin yhä useammat reitittimet tukevat arkkitehtuurin protokollia sekä tarjoavat alustan sen toimintamekanismien toteutukseen. QoS-palvelu toteutetaan usein reitittimeen, mutta erilaisia kytkinkonfiguraatioitakin on myös olemassa reitityksen tueksi. (Cisco Systems Incorporated 2008b; Wikipedia 2008c.)

Differentiated Services toteuttaa liikenteen erottelua jakamalla ja merkitsemällä IP-paketteja ennaltamäärättyihin prioriteetti-tiloihin. Se ei siis ota kantaa itse liikennevuon priorisointiin vaan tekee sen pakettitasolla tehden näin palvelun erittäin hyvin integroitavaksi juuri reitittimeen ja reititykseen. Lisäksi jokainen reitin toimii arkkitehtuurissa ikäänkuin itsenäisenä solmukohtana ottamatta kantaa naapurireitittimen toimintaan. Differentiated Services toteutuu verkossa hyppyhypyltä -periaatteella (per-hop basis), jolloin priorisointi tehdään jokaisessa verkon laitteessa erikseen. Usein DiffServ-verkon reitittimissä ja kytkimissä on käytössä yhtenäinen priorisointisäännöstö, vaikka useinkaan tämä ei ole toteutuslogiikoiden kannalta välttämätöntä. Differentiated Services -toteutus voidaan tehdä monella tavalla, mutta esimerkiksi Cisco tarjoaa siihen varsin tehokkaan tavan käyttäen hyväksi DSCP (Differentiated Services Code Point)-tekniikkaa. DSCP on yleisin DiffServ-toteutustekniikka. Tekniikkaan ja sen toteutustapoihin syvennytään tarkemmin opinnäytetyön käytännön osuudessa sekä Ciscon Quality of Service -tekniikoita käsittelevässä luvussa 4.3. (Cisco Systems Incorporated 2008b; Wikipedia 2008c.)

Differentiated Services -arkkitehtuuri nojaa IP-pakettien priorisoimiseen merkitsemällä niiden otsikkokentät ja jakamalla ne edelleen prioriteettiluokkiin. Prioriteettiluokkiin jaettavat paketit käsitellään reitittimessä suoraan luokkakohtaisesti, mikä yhtenäistää ja selkeyttää itse liikenteen priorisointia. Prioriteettiluokkien kirjo voi olla hyvinkin moninainen, varsinkin jos verkossa on paljon palveluja ja vähän kaistaa. Jokaista prioriteettiluokkaa voidaan muokata tarpeen tullen yksilöllisesti, joko vaihtamalla prioriteettia tai pakettikohtaisia luokituksia. Reitittimet käsittelevät paketteja niiden luokituksen mukaan. Tämä tarkoittaa sitä, että jokainen paketti on jossain luokassa. Luokkajaoissa samassa luokassa olevat paketit saavat saman prioriteetin lähetyksessä. Reitittimissä prosessi on nimeltään ”Per-Hop Behavior” (PHB). Samaan luokkaan kuuluvat paketit, ”Behavior Aggregate” (BA), noudattavat samaa luokitusta ja prioriteettia. Per-Hop Behavior -prosessilla voidaan IP-pakettitasolla taata moninainen ja tehokas liikennejako. Käytännössä jopa 64 erilaista liikennetyyppiä voidaan osoittaa omaan prioriteettiluokkaansa. Luokkajako itsessään voi perustua esimerkiksi paketteihin, joiden viive verkossa on minimoitava, paketteihin, joiden häviöt matkalla ja uudelleenlähetykset on eliminoitava, sekä vapaasti jäljelle jäävää kapasiteettia käyttävään liikenteeseen. Vapaasti, ilman erityisvaatimuksia siirrettävä liikenne ja sen paketit merkitäänkin usein heikoimman prioriteetin luokkaan. (Cisco Systems Incorporated 2008b; Wikipedia 2008c.)

Differentiated Services -palvelua käyttävien reitittimien QoS-palvelua voidaan helposti kontrolloida pelkästään luokkaprioriteetteja vaihtamalla. Yksinkertaisimmillaan luokkia voi olla kaksi, Expedited Forwarding (EF) ja Assured Forwarding (AF). Expedited Forwarding on priorisoitua liikennettä jolla on minimaalinen viive ja häviöt. Tällöin pakettien perille pääsy taataan äärimmillään jopa muun liikenteen kustannuksella. Assured Forwarding on muuta liikennettä, jota voidaan edelleen jakaa alempiin luokkiin asettaen uusia hierarkisia prioriteetteja näiden liikennetyyppien kesken. Usein vain muutama liikennetyyppi kuten esimerkiksi VoIP-protokollaan nojaavat sovellukset, tarvitsevat EF-luokituksen. AF-luokkaan ei useinkaan priorisointia tarvitse edes soveltaa, vaikkakin se saattaa sisältää kymmeniäkin erilaisia liikennetyyppejä ja niiden protokollia. Differentiated Services -arkkitehtuuria ajatellen, olennaista onkin toteuttaa verkkosuunnittelua ja ylläpitoa kriittisten palvelujen näkökulmasta. (Yourdictionary.com 2008.)

Differentiated Services-arkkitehtuurissa ei tarvita erikseen signalointiprotokollia. Tällöin ei tarvita erillistä kaistakapasiteetin varaamista itse signalointiliikenteelle. Lähiverkoissa ja pienen kokoluokan yritysverkoissa Differentiated Services on helposti hallittava ja ylläpidollisesti kannattava ratkaisu, mutta juuri signaloinnin puuttumisen vuoksi palvelua on hyvinkin vaikea toteuttaa internetin kaltaisessa massiivisessa verkossa. Harvoin tähän on kuitenkaan tarvetta. Differentiated Services -arkkitehtuurin suurin heikkous onkin juuri skaalautuvuuden puuttuminen, koska verkon laitteiden välillä ei ole minkäänlaista kommunikointia signaloinnin puuttumisen vuoksi. Verkon laitteet toimivat siis itsenäisinä solmupisteinä jolloin kaistanvarausta ei aina pystytä takaamaan kattavasti koko verkon osalta. Toisaalta suunnittelussa etua tuo se, että verkon kriittisissä solmukohtissa vain yksi Differentiated Serviceä toteuttava laite voi tarjota toimivan QoS-palvelun koko verkolle. Tämä helpottaa huomattavasti koko verkon QoS-hallintaa ja valvontaa. (Cisco Systems Incorporated 2008b; Jarmo Siltanen 2007; Wikipedia 2008c.)

DiffServ-laitteiden autonomisuus helpottaa palvelun ylläpitoa koska pienilläkin laitekohtaisilla konfiguraatiomuutoksilla saadaan luotua koko verkon liikennekäyttäytymiseen vaikuttavia säännöstöjä. Tämä ilmenee varsinkin reitityksen kannalta olennaisissa verkon solmukohtissa, jonne liikennevirrat on keskitetty. Toisaalta tässä piilee myös DiffServ-toteutusten heikkous. Ylläpitäjän on usein tehtävä tarkkoja palvelutasomittauksia. Näin varmistetaan että liikenne ei kierrä tai muutoin ohita palvelua toteuttavia laitteita. Vaikka palvelu toteutuu laitteessa, ei se välttämättä toteudu itse verkossa. Lisäksi Differentiated Services -arkkitehtuurin suunnitteluvaiheessa ja ylläpidossa on muistettava, että vaikka palvelun toteutus verkon osalta olisikin yhtenäinen kokonaisuus, laitekonfiguraatiot ovat aina laitekohtaisia ja sitä kautta yksilöllisiä. Yksikin heikko ”DiffServ-lenkki” verkossa saattaa laskea palvelutasoja merkittävästi. (Cisco Systems Incorporated 2008b; Wikipedia 2008c; Jarmo Siltanen 2007.)

Käytännön elämässä Differentiated Servicen suurin ongelma on End-to-End-yhteyksien takaaminen laajoissa verkoissa kuten, internetin reunaverkoissa tai paljon solmukohtia sisältävissä asiakasverkoissa. Tällaisissa verkoissa ei pystytä takaamaan toimivaa End-to-End DiffServ-toteutusta, koska palvelu ei tue polkuvarauksia päästä päähän. Tällöin kokonaisten DiffServ-toteutusten sijaan on käytettävä jotain muuta QoS-palveluratkaisua, kuten Integrated Serviceä. Toimiva vaihtoehto on asettaa DiffServ-laitteet runkoverkkoihin tai palomuurien yhteyteen, mistä liikenne ei pääse kiertämään muun verkon kautta. Runkoverkkojen päässä olevat asiakasverkot varustetaan usein Integrated Services -toteutuksella jossa palvelunlaatu taataan aina runkoverkkoon liikennöivälle reitittimelle asti. Tästä eteenpäin runkoverkko voikin sisältää useitakin palvelunlaadultaan erilaisia DiffServ-verkkoja (region). Runkoverkon takana sijaitseva verkko voi toteuttaa jälleen Integrated Serviceä reunareitittimeltä asiakasverkkoon. QoS-arkkitehtuurin vaihtuessa tarvitaan usein esimerkiksi DiffServ-runkoon QoS-kontrollipalvelin (policy server), joka pitää runkoverkon puolella kirjaa asiakasverkkojen kokonaisvarauksista. Näin varmistetaan että QoS-tekniikan vaihtuessa asiakas ei ylitä hänelle määriteltyjä palvelutasoja tai kokonaiskaistavarauksia. (Wang 2001, 115-118.)

3.3 Integrated Services

Integrated Services tai lyhyemmin IntServ-arkkitehtuuri on QoS-palvelutoteutus, joka on päätehtäviltään verrattavissa Differentiated Services -arkkitehtuuriin. Tavoitteina tietoliikennepalvelujen priorisointi ja kriittisen liikenteen päästä päähän kuljettaminen häiriöttä ovat samat, mutta toteutuslogiikassa on huomattavia eroja. Integrated Servicen olennaisin piirre on toteuttaa palvelunlaatua kaikkien verkon laitteiden osalta signalointia hyödyntäen. Integrated Service- palvelutoteutukset ovatkin yleensä varsin laajoja ja yksityiskohtaisia verkkointegraatioita mahdollistaen yhtenäisen QoS-mallin monien liikennetyyppien tarpeisiin. Arkkitehtuuri toimii kiinteästi verkkokerroksella (OSI-3) reitityksen yhteydessä sekä osittain myös kuljetuskerroksen (OSI-4) palvelujen osana. Protokollat itsessään eivät kuitenkaan ota minkäänlaista kantaa itse reititykseen. Integrated Services -malli pe-

rustuu IP-pakettien ja datavuon hallintaan ja seurantaan liikennetyyppikohtaisesti, käyttäen hyväksi laitteiden välistä signalointia ja kaistanvarausta. (Cisco Systems Incorporated 2008c; Wikipedia 2008d.)

Signalointiprotokollana toimii yleisimmin RSVP (Resource Reservation Protocol), jonka päätehtävä verkossa on Integrated Services -palvelun kontrollointi ja looginen toteutus. Protokollan tämänhetkinen kehitysversio on RSVP-TE (RSVP-Traffic Engineering), jonka parannetut ominaisuudet tulevat jatkossakin tukemaan IntServ-toteutuksia tietoliikenneverkoissa. Protokolla varaa vastaanottajapäässä verkkoresursseja sekä unicast- että multicast-liikenteelle riippuen siitä, mitkä tarpeet sovelluksilla on minkäkinlaiselle liikenteelle. Liikennepriorisointi tehdään siis lähtökohtaisesti sovellustason liikenteen tarpeisiin. IntServ-malli pyrkii reaaliaikaiseen QoS-arkkitehtuuritoteutukseen. Priorisointi elää muuttuvien tarpeiden mukaan sekä verkon aktiivilaitteissa reitityksen yhteydessä, että verkon reunalla. Näin ollen myös verkon käyttäjät määrittävät omalta osaltaan QoS-liikennekuormituksen verkon päästä päähän. (Cisco Systems Incorporated 2008c; Wikipedia 2008d.)

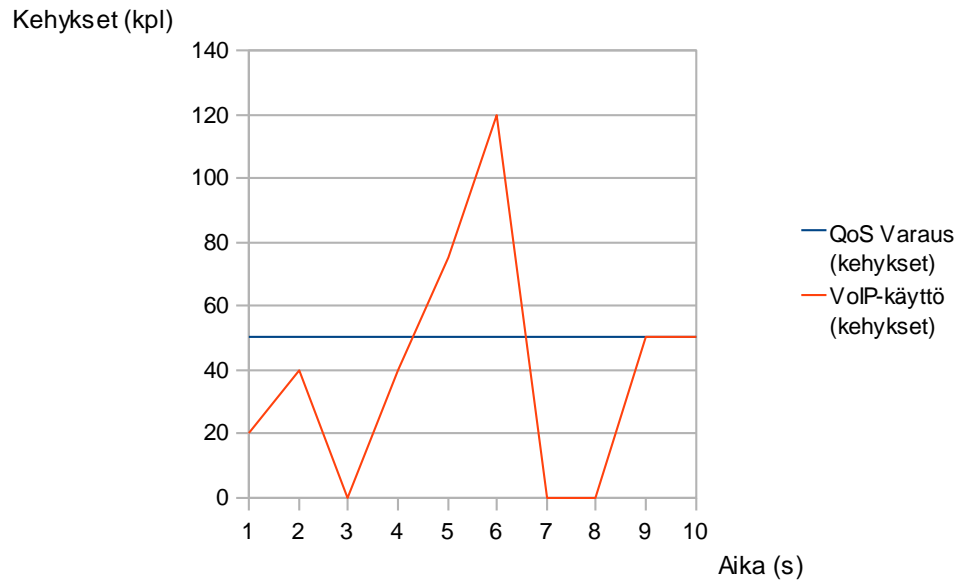
Integrated Services -arkkitehtuurin laitteet toteuttavat kahta operaatiota liikenteen seurannassaan. Nämä ovat nimeltään TSPEC (Traffic specification) ja RSPEC (Request specification). TSPEC-operaatiossa määritellään liikennetyyppi (FTP, VoIP, HTTP, IPTV jne.). Kullekin liikennetyypille annetaan tietynkoinen varaus verkkolinjaan, jota ne voivat käyttää joko puskurittain tai tasaisella kaistanleveydellä. Resurssinvaraus voidaan tehdä esimerkiksi periaatteella 50 kehystä sekunnissa. Jos liikenteessä on odotettavissa piikkejä, voidaan säästyneitä resursseja varata ikäänkuin varastoon tai puskuriin (buffer) ja käyttää tätä varastoa liikennepiikkien kohdalla kasvattamaan hetkellistä kaistaa. RSPEC puolestaan määrittää liikenteen käsittelyprioriteetit. Liikenne jaotellaan sen mukaan, varataanko sille kaistaa vai kuljetetaanko paketit verkon yli vasta, kun kukaan muu ei resursseja tarvitse. Reaaliaikaisuutta vaativat liikennetyypit, kuten VoIP ja videokuva saavat yleensä korkeimman RSPEC-prioriteetin. Tämän ansiosta paketit toimitetaan mahdollisimman pienellä viiveellä, ja niiden perillepääsyä valvotaan tiukasti minimoimalla pudotukset ja hylkäykset reitittimissä. Usein puhutaankin minimipalvelutason takaamisesta, mikä onkin yksi RSPEC:n päätehtävistä. Näiden liikenne-

tyyppien kohdalla TSPEC puolestaan takaa riittävän vuon ja sopivankokoiset puskurit mahdollisille piikkiarvoille. (Cisco Systems Incorporated 2008c; Wikipedia 2008d.)

Integrated Services TCPEC- ja RSPEC-operaatiologiikkaa voidaan havainnollistaa esimerkiksi VoIP-liikentessä seuraavasti. Taulukko 1 (TAULUKKO 1) ja kuvio 4 (KUVIO 4) toimivat esimerkkinä kuvitteelliselle tilanteelle, jossa kaistaa varataan 50 Ethernet-kehystä sekunnissa VoIP-liikenteen käyttöön. Kuvion 4 käyrän alle jäävä pinta-ala kertoo varattujen ja käytettyjen kehysten määrän ajan funktiona. Koska varattu pinta-ala on suurempi 10 sekunnin ajalta, on QoS-taso riittävä huolimatta piikkiarvosta. Varaus voidaan tosin toteuttaa piikkiarvon mukaan eli 120 kehystä sekunnissa, mutta silloin tuhlattaisiin kaistaa muulta liikenteeltä. Videokuvan siirrossa varaus kannattaa tosin tehdä piikkiarvon mukaan, koska videokuvan IP-pakettien lähetystiheydet ja datavuo ovat usein varsin stabiileja.

TAULUKKO 1. QoS kehysvaraus VoIP-sovellukselle.

Aika (s)	QoS kehysvaraus (kpl)	VoIP kehukset (kpl)
1	50	20
2	50	40
3	50	0
4	50	40
5	50	75
6	50	120
7	50	0
8	50	0
9	50	50
10	50	50



KUVIO 4. Graafinen esitys QoS-kehysvarauksesta VoIP-sovellukselle.

Kuviota 4 voidaan havainnollistaa RPSEC:n osalta seuraavasti. RSPEC määrittää kuviossa 4 esitetyle VoIP-liikenteelle kaksi palveluluokitusta: Pakettien viiveet minimoidaan siirtotiellä. Pakettien pudotuksia laitteissa kontrolloidaan ja minimoidaan erittäin tarkasti. Näin luodaan pohja kuviossa 4 esitetyn liikennevarauksen toimivuudelle, sillä pakettiviiveet ja pudotukset häiritsisivät pahasti paitsi VoIP:n toimivuutta, myös TSPEC-operaatiota. TSPEC tekee siis varsinaisen kaistan fyysisen varauksen varaamalla kehyksiä siirtotielle. Käytännössä TSPEC määrittää maksimisiirtokaistan VoIP:lle, mutta RSPEC viime kädessä pitää huolta siitä, että varattua kaistaa käytetään tehokkaasti sovelluksen tarpeisiin, varmistamalla Ethernet-kehysten estottoman perillepääsyn.

Integrated Services -palvelu toimii useimmiten verkoissa, joissa on paljon solmu-kohtia ja joissa tarvitaan tehokasta reititystä. Verkon läpi kulkee usein monia Int-Serv-polkuja, joissa priorisoitu liikenne virtaa. Suurin ongelma IntServ-toteutuksissa on arkkitehtuurista puuttuva palvelun sisäinen reititys. Usein Int-Serv-polut noudattavat esimerkiksi RIP-reititysprotokollan käyttäytymistä, jossa lyhin reitti mahdollisimman pienillä hyppyillä on paras. QoS:n kannalta tämä logiikka ei kuitenkaan aina toimi. RIP saattaa määrittää lyhimmäksi reitiksi linkki-välin, jossa kaistanleveys on huonoin. RIP ei siis ota millään tavalla kantaa liiken-

teen sujuvuuteen verkossa. Usein IntServ-toteutuksissa tällaiset linkkivälit on irrottava palvelusta tai tehtävä staattisia reitityksiä RIP:n sijasta. Heikkojen linkkiväliden korvaaminen suoraan tehokkaammilla saattaa osoittautua helpoimmaksi ratkaisuksi, jolloin reititysprotokollaa ei tarvitse muuttaa tai korvata. Joissain tapauksissa myös erityisen IntServ-palvelua tukevan reitityspalvelimen käyttöönotto tai verkkotopologian muuttaminen saattaa ratkaista ongelman. (Wang 2001, 23-24.)

3.4 Muut palvelunlaatu-tekniikat

Policy based QoS on yleiskäyttöinen nimitys sovellustason priorisointitekniikoille. Esimerkiksi NBAR- ja RSVP-protokollia voidaan hyödyntää policy based -palvelutoteutuksissa ja datavuon tunnistuksessa. Vaikka QoS-mekaniikka nojaa osittain pakettien DSCP- ja ToS-merkkaukseen verkko- ja siirtoyhteyskerroksella, säännösten (policy) luontia kontrolloivat päätelaitteet. Isoimmissa verkkokokonaisuuksissa käytetään usein palvelinratkaisua kontrolloimaan koko verkon QoS-säännösten. Keskitetyn hallinnan ja päätelaitteiden välisten End-to-End-säännösten ansiosta verkon kytkinten ja reitittimien konfigurointiin ei useinkaan policy based QoS -toteutuksissa tarvitse erikseen varautua. Tällöin tosin tarvitaan erillinen kontrollilaitte, joka korvaa reitittimien ja kytkinten roolin palvelun fyysisenä ja loogisena toteuttajana. Merkittävin ero policy based -palvelutoteutuksissa verrattuna Integrated Services- ja Differentiated Services- arkkitehtuureihin on verkkotason QoS-mekanismien piilottaminen sovellustasorajapinnan taakse. (Extreme Networks Incorporated 2000.)

QoS-palvelun ylläpito on pyritty yksinkertaistamaan, sillä mekanismi toimii keskitetysti päätelaitteiden tai erillisen QoS-verkonhallintasovelluksen kontrolloidessa säännösten. Säännösten kirjo on usein varsin suppea, koska policy based QoS takaa usein vain sovellus- tai käyttäjätason kriittisen liikenteen. QoS-säännösten voidaan tosin edelleen jaotella erityisiin profiileihin, joko käyttäjien, liikennetyyppien tai sovellusten mukaan. Tämä ominaisuus tuokin mahdollisuuden luoda erilaisia yksilöllisiä palveluluokkia eri käyttäjäryhmille. Policy based

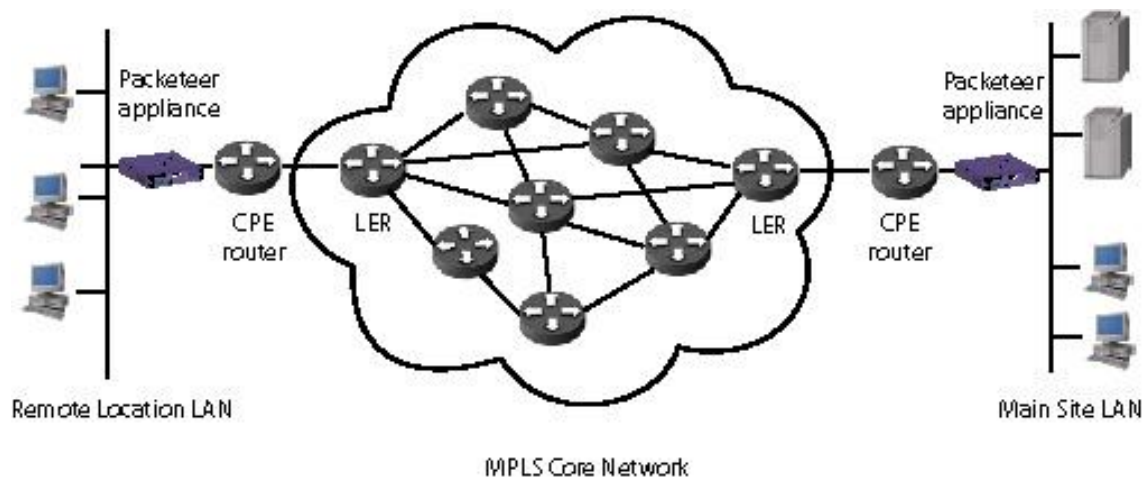
QoS lähestyykin QoS-palvelutoteutusta tarjoten vain välttämättömimmät perusmekanismit sovellustason tarpeisiin. Policy based QoS soveltuu usein ylläpidollisesti järkeväksi ratkaisuksi esimerkiksi yritysverkkoihin. (Extreme Networks Incorporated 2000.)

Quality of Service -palvelua ei aina tarvitse toteuttaa verkon reitittimisissä ja kytkimissä. Palvelun toteutus voidaan tehdä myös paikallisesti ilman verkon tai sen laitteiston erillistä tukea. Tästä hyvänä esimerkkinä ovat erilaiset käyttöjärjestelmät ja serverit, jotka usein huolehtivat vain omasta QoS-palvelutoteutuksestaan. Usein tällaiset laitteet käsittelevät IP- paketteja ja liikennettä ”best effort”-mallisesti, jolloin liikennetyyppeihin ei oteta kantaa. Tällä on suora vaikutus sovellustason toimintaan. QoS-asetukset toteutetaan verkkokorttirajapinnassa. Saapuva liikenne vastaanotetaan usein verkon QoS-säännösten mukaan, koska vastaanottajakone ei voi enää QoS-palvelun tasoon juurikaan vaikuttaa. Sen sijaan liikenteen lähetyksessä verkkokorttikohtaisilla QoS-asetuksilla on paljonkin merkitystä varsinkin palvelinten osalta. Liikenne kategorisoidaan joko IP-osoitteiden, protokollien, porttien tai sovellusnimien ja polkujen perusteella. Lähiverkoissa host-koneet voivatkin muodostaa yhtenäisen QoS-verkon liikennetyyppisäätöihin nojaten. QoS-operaatioita kontrolloidaan ohjelmistollisesti käyttöliittymästä riippuen. Esimerkiksi windows-ympäristössä liikennettä kontrolloi erillinen ohjelmistorajapinta, ”traffic control application programming interface” (TC API/QoS API). Protokollatasolla DSCP ja RSVP suorittavat usein paketti- ja datavuomerkintää, mutta varsinaisesta QoS-verkkototeutuksesta saati arkkitehtuurista ei kuitenkaan voida puhua. (Davies 2007.)

Tulevaisuuden QoS-verkkonäkymänä voidaan pitää tehokasta, integroitua MPLS ja QoS-palvelutoteutusta IP-verkoille. Tekniikka ei ole uusi, ja sitä on jo menestyksekkäästi käytetty runkoverkon QoS-toteutuksissa. MPLS (Multi Protocol Label Switching) on runkoverkon tekniikka jossa IP- paketit ”merkitään” verkon reunareitittimillä (Label Edge Router) kulkemaan jokin tietty reitti reunalta reunalle. Tätä reittiä kutsutaan nimellä Label Swithed Path (LSP). Tämän jälkeen liikenne siirtyy CPE (customer premise)-reitittimelle, joka on usein WAN-reititin asiakasverkon reunalla. Reunareitittimien sijoittelu MPLS- runkoverkkoon on esi-

tetty yksityiskohtaisemmin kuviossa 5 (KUVIO 5). Vaikka MPLS perustuu osittain vanhempaan ATM-tekniikkaan (Asynchronous Transfer Mode), se ei tue ennalta-reititettyjä virtuaaliyhteyksiä vaihtoehtoisten reittien kautta kuten ATM. MPLS saattaa kuormittaa joitain linkkivälejä raskaasti ja jättää vaihtoehtoina potentiaaliset reitit hyödyntämättä. Tähän tehokkaaseen verkkoresurssien käytön ongelmaan on vastattu kehittämällä MPLS-TE (Traffic Engineering). MPLS-TE tekee LPS-reittien ohien vaihtoehtoisille reiteille liikennetunneleita (traffic trunks), jotka toimivat aivan samalla ideologialla kuin ATM-tekniikan vaihtoehtoiset virtuaalireitit. Näin voidaan tehokkaasti valjastaa vanhan tekniikan hyvät puolet uuden verkkotekniikan käyttöön. (Cisco Press 2009; Cisco Systems Incorporated 2009e; Blue Coat Systems Incorporated 2009.)

Label Edge Routerin käyttö MPLS-runkoverkossa on sukua siirtoyhteyserroksen tag switching -tekniikalle, jossa pakettien Ethernet-kehysiin merkitään erillinen tieto (tag) paketin reitistä runkoverkon läpi. Erona on kuitenkin se, että MPLS-kontrollointi soveltuu paremmin verkkokerroksen IP-pakettien käyttöön. MPLS-tekniikka toteuttaa nopeaa, ennaltamäärättyjen reittien liikenteen siirtoa liikenteenasauksella (load balancing) ilman runkoverkon solmukohtaista reititystä. Pyrkimykset tehokkaampaan tiedonsiirtoon eliminoimalla viiveitä ja hyödyntämällä kapasiteettia parhaalla mahdollisella tavalla ovat analogiassa QoS-palvelun ideologian kanssa. Erona on kuitenkin se että QoS-tekniikat MPLS-runkoverkoissa usein sotkevat ennaltamäärättyjä verkon kuormantasausta toteutavia reittejä. Tästä huolimatta Quality of Service on tekniikkana hyvin integroitava MPLS-runkoon. (Cisco Systems Incorporated 2009e; Blue Coat Systems Incorporated 2009.)



KUVIO 5. MPLS-runkoverkko ja reunareitittimet (LER) (Blue Coat Systems Incorporated 2009).

Kuten useimmat QoS-tekniikat, MPLS-QoS käyttää liikenteen merkintään ja priorisointimekanismeihin joko RSVP-signaaloitinta tai DSCP-pakettimerkintää. MPLS-runko takaa jo sinällään tehokkaan liikenteensiirron reititinsolmujen läpi, mutta varsinaista luokkajakoa tai minimidatavuontakausta kriittisen liikenteen tarpeisiin tekniikka ei sinällään tue. Koska QoS yleensä korvaa nämä puutteet jo peruspalvelutasoillaan, saadaan aikaiseksi erittäin tehokas ja palveluiltaan toimiva runkoverkko. RSVP soveltuu hyvin MPLS-runkoverkkoon, koska sen varaus ja signaaloitimekanismi perustuu pitkälti vain lähde- ja kohdeosoitteisiin. Nämä osoitteet viittaavat suoraan verkon päätelaitteisiin, joten erillistä pakettien käsittelyä MPLS-runkoverkon reitittimisessä ei tarvita. Samaan ideologiaan perustuu myös koko MPLS-runkoverkon toiminta, sillä hop-by-hop -reititys ja pakettiprosessointi halutaan eliminoida solmupisteistä verkon toiminnan nopeuttamiseksi. Myös DSCP-prosesseja käyttävä QoS hyötyy MPLS-verkon toiminnasta. Kun MPLS-reunareititin (LER) saa DSCP-merkityn paketin, jonka on tarkoitus matkata runkoverkkoon, se merkitsee paketin DSCP-kenttää vastaavalla MPLS-kentällä (MPLS label). MPLS label sisältää erikseen tiedot QoS-palvelunlaadusta ja priorisoinnista. Paketti kulkee runkoverkon läpi, ja saavuttuaan vastakkaiselle LER-reitittimelle MPLS label riisutaan ja siirrytään normaaliin DSCP-prosessiin, jossa pakettia kuljetetaan reitittämällä (hop-by-hop basis). Ainoa uhka tehokkaalle

QoS-prosessille MPLS-reitityksessä on kuormantasaus, joka saattaa QoS-prosesseissa muuttaa ennaltamäärättyjä MPLS-reittejä. Tällöin lopputuloksena saattaaakin syntyä kuormallisesti epätasapainoinen MPLS-runko. (Blue Coat Systems Incorporated 2009; Cisco Systems Incorporated 2009e.)

Edellämainittujen tekniikoiden ja integroitujen toteutustapojen lisäksi Quality of Serviceksi voidaan mieltää verkkototeutukset, jotka eivät toteuta mitään varsinaista QoS-protokollaa tai mekanismia (CoS, ToS, DSCP, RSVP, Queueing). Näitä kutsutaan tässä kappaleessa nimellä ”keinotekoiset QoS-toteutukset”. Tällöin onkin kyse usein jo verkkosuunnittelussa fyysisten siirtoyhteyksien sijoitteluun liittyvistä kriittisten palveluiden takaamisesta riittävällä kaistanleveydellä. Sama voidaan toteuttaa esimerkiksi runkoverkon resurssien jakamisella lähiverkkoon, jolloin kytkinporttien lukumäärää tai nopeutta voidaan säädellä verkkosegmenteille. Myös muusta liikenteestä vaikkapa palomuuritoteutuksella erotettu siirtoyhteys takaa kyseisessä mediassa kuljetettavalle liikenteelle huomattavan prioriteettitason. Helposti ajatellaan, että pelkästään siirtoyhteyksien kaistanleveyttä kasvattamalla nostetaan myös Quality of Servicen tasoa. Tämä on harhaanjohtava käsitys, sillä kaistanleveys ei vaikuta priorisointiin. Turha liikenne vie prosentuaalisesti saman verran kaistaa kriittiseltä liikenteeltä siirtoyhteyden tehokkuudesta riippumatta ellei se saa vaatimaansa maksimikaistaa. Suuret liikennevirrat saattavat myös vaatia verkon pääte- ja aktiivilaitteilta huomattavat määrät prosessointitehoa, jolloin kriittisen liikenteen vastaanottaminenkin ja käsittely voi edelleen hidastua. Edellämainittuja ”keinotekoisia QoS- toteutuksia” ei tässä opinnäytetyössä tunnusteta Quality of Service -tekniikoiksi, ja ne ovatkin lähinnä ajatuksia herättäviä vertailukohtia tutkimuksen varsinaiselle sisällölle selkeyttämään Quality of Servicen tietoliikenneteknistä käsitettä.

4 CISCO JA QUALITY OF SERVICE

4.1 Cisco Systems ja QoS-sovellusten tuki

Cisco Systems on tietoliikenneverkkojen suunniteluun ja tietoverkkojen aktiivilaitteiden valmistukseen keskittynyt yhtiö. Sen päämaja on San Josessa, Kaliforniassa. Yhtiön perustivat Len Bosack ja Sandy Lerner, ja se on ollut toiminnassa vuodesta 1984 lähtien. Cisco suunnittelee ja valmistaa laitteita viiden brändin alaisuudessa: Cisco, Linksys, WedEx, IronPort ja Scientific Atlanta. Cisco ei ollut ensimmäinen kaupallisia reitittämiä valmistava yhtiö, mutta IP-verkkojen yleistyttyä 80- ja 90-lukujen kuluessa, se saavutti jalansijan johtavien laitevalmistajien keskuudessa. Innovaatiot, kuten moniprotokollatuki reitittimissä, sekä tiivis tekniikka- ja protokollakehitystyö ovat tehneet Ciscon reitittimistä ja verkkolaitteista suosittuja IP-verkoissa ja verkkoarkkitehtuureissa, jotka hyödyntävät MPLS-kehyskäsitteitä reitityksessä. (Cisco Systems Incorporated 2008f; Wikipedia 2008e.)

Kasvava laitetarve tietoverkkojen yleistyessä myös kotikäytössä, asettaa sekä haasteita että mahdollisuuksia laite- ja ohjelmistoyhtiöille. Cisco arvioi, että vuoteen 2010 mennessä 14 biljoonaa laitetta on kytketty Internetiin (arvio tehty vuonna 2008). Omalta osaltaan Cisco pyrkii tarjoamaan laajan laitepohjan niin yritys- kuin kotikäyttöönkin. Huomioon otetaan erilaiset palvelutarpeet niin datalle, äänelle, videolle kuin mobiiliratkaisuillekin. Cisco kehittää muunmuassa Linksys-brändin alla langattomia, ja etenkin kotiverkon tarpeisiin suunniteltuja laitteita. Kokonaisvaltainen kehitystyö, R&D (Research and Development), on yksi Ciscon liiketoiminnan kulmakivi johon sijoitetaan miljardeja dollareita vuositasolla. (Cisco Systems Incorporated 2008f.)

Opinnäytetyössä käsitelty QoS-palvelu ja sen tuki on otettu hyvin huomioon myös Ciscon laitteiden ohjelmistoissa. Cisco on tehnyt myös omaa tekniikka- ja protokollakehitystyötä QoS-palvelun osalta, tavoitteena sisällyttää palvelu hyvin skaalautuvaksi osaksi itse verkkosuunnittelua. Cisco on paitsi havainnoinut, myös vahvistanut loogisten tekniikoiden merkitystä verkkoresurssien kasvattamisessa ja

toimivuuden varmistamisessa. Kattava QoS-arkkitehtuurien tukeminen on hyvä esimerkki tästä kehityssuuntauksesta. Tietoliikenneverkkoihin valjastetut Quality of Service -palvelut ovat hyvin riippuvaisia verkon aktiivilaitteista ja etenkin reitityksestä vastaavista verkkokerrostason reitittimistä. Cisco on ollut mukana kehittämässä QoS-arkkitehtuuria aktiivisesti panostamalla omien QoS-protokollien tutkimustyöhön. Tätä skenaariota on toteutettu paitsi erilaisilla uusilla palveluarkkitehtuureilla, mutta myös tukemalla laitteissa jo olemassa olevia ohjelmistoratkaisuja. QoS-standardit laitevalmistajien kesken tähtäävät lähinnä liikennetyyppi- ja protokollakohtaiseen tukeen, mutta toteutustekniikat vaihtelevat. Tällä tähdätään mahdollisimman vapaaseen verkkosuunnitteluun ilman laitevalmistajakohtaisia sidonnaisuuksia tai rajoituksia. Ciscon reitittimet ja runkokytkimet tukevat seuraavia liikennetyyppejä ja protokollia:

- videostriimaus ja multimediasovellukset
- IPTV
- VoIP ja IP-pohjaiset äänisovellukset
- VTC (Video Conferencing)
- SSH-hallinta
- Interaktiiviset sovellukset (Online Gaming)
- RSVP (Integrated Services)
- DSCP-toteutus (Differentiated Services)
- Cisco NBAR-tekniikka
- IPv4 ja IPv6-yhteensopivuus
- APM CBR, IP CBR ja CIR, CR-LDP

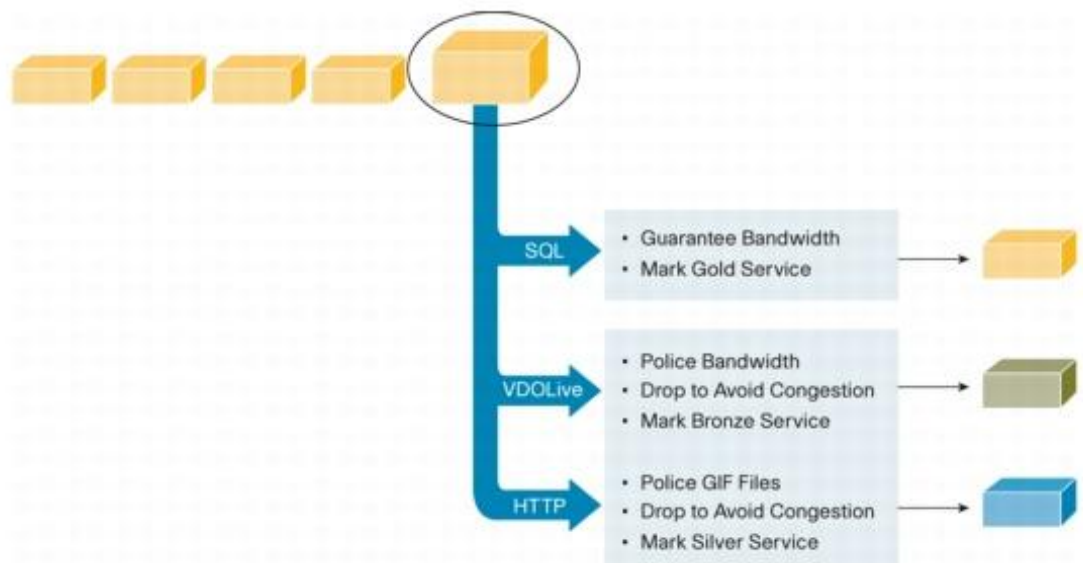
Näistä tärkeimmiksi ja keskeisimmiksi tämän tutkimuksen kannalta nousevat loogisilta toteutuksiltaan monipuolinen DSCP, videostriimaus ja Integrated Servicesen RSVP-protokolla. (TeraBitti 2008; Wikipedia 2008a.)

4.2 Cisco NBAR

NBAR (Network-Based Application Recognition) on Ciscon kehittämä QoS-toteutusmenetelmä reitittimissä ja kytkimissä. Ensimmäinen tekniikkaa tukeva Ciscon IOS-versio oli 12.0(5)XE2. Resurssien varaus tehdään sovellustason (OSI-7) liikenneominaisuuksilla. Itse prosessimekaniikka turvaa kuljetuskerroksen palveluihin (OSI-4). Liikenne kategorioidaan kuljetuskerroksella sen mukaan, mitä sovellusta se edustaa. Koska liikennetyyppien paketit tulevat yleensä tietyssä datavuossa tai ryppäässä, Ciscon laitteet tekevät QoS-priorisoinnin jo ensimmäisten pakettien perusteella. Näin päästä päähän -reititystä ei tarvitse tehdä jokaisen pake-tin tai kehyksen osalta erikseen. Jatkuvan QoS -prosessoinnin puuttuminen no-peuttaa hieman reititystä ja edesauttaa liikenteen sujuvuutta, mutta toisaalta väljä kategorisointi ei tue erityisen tiivistä protokollakohtaista QoS-politiikkaa. NBAR-mekanismiin käyttöä perustellaan usein sillä, että useimmissa tapauksissa Quality of Servicen tehokas toteutus käytännössä ei vaadi kovinkaan monimutkaisia prio-risointisäännöstyöjää, saati aktiivista pakettiprosessointia. (Cisco Systems Incorporated 2008d.)

NBAR takaa varatun End-to-End-yhteyden verkon päätteille ja niiden käyt-tämille sovelluksille. Sovellukset tunnistetaan niiden käyttämien TCP- tai UDP-porttien porttinumeroiden mukaan. QoS-periaatteiden mukaan sovellukset voi-daan jakaa prioriteetti- ja palveluluokkiin, vaikkapa Gold, Silver ja Bronze -prioriteetti- ja palveluluokituksella. Kuviossa 6 (KUVIO 6) VDOLive -videoliikenne, SQL-tietokannan resurssit ja HTTP-liikenne on priorisoitu edellämaitulla tavalla. Yk-sinkertaisimmillaan Gold-tason SQL-sovellusliikenne on taattu tietyllä kaistanle-veydellä, tarpeen vaatiessa muiden palveluiden kustannuksella. Tämä johtuu siitä että SQL-palvelimelle kohdistuvat pyynnöt vaativat lyhyitä vasteaikoja ja pitkälli-nen verkon ylikuormitus sattaa haitata merkittävästi palvelun toimivuutta. VDO-Live ja HTTP ovat palveluluokiltaan ja tyyplitään selvästi heikommat verrattuna SQL-liikenteeseen, mutta toisiinsa nähden lähes samanarvoiset. Usein Silver- ja Bronze-tason liikenne määritelläänkin selvästi toisarvoiseksi QoS-toteutuksissa. Kuvion 6 esimerkissä sekä VDOLive- että HTTP-liikenne saavat kukin tietyn kaistavarauksen. Varaus on Silver-liikenteellä mahdollisesti suurempi verrattuna

Bronze-liikenteeseen, mutta kummankaan prioriteettiluokan varausta ei taata, sillä SQL-liikenne aktivoituessaan vie aina tietyn, QoS-asetuksilla ennaltamääritetyn kaistan. Palvelutyypiltään VDOLive- ja HTTP-liikenne ovat kuitenkin samantarvoiset. Niiden pakettien perillemeno ei taata jos verkon kuormitus kasvaa yli QoS-palveluasetuksissa määritettyjen rajojen. (Cisco Systems Incorporated 2008d.)



KUVIO 6. NBAR-priorisointi (Cisco Systems Incorporated 2008d).

NBAR-toteutus mukailee monissa kohdin Integrated Servicesin RSVP-toteutusta. Koska kyseisen QoS-tekniikan kehitystä tuetaan pelkästään Ciscon laitteissa, sitä ei voida lukea osaksi Integrated Services -arkkitehtuuria. Toisaalta itse toteutusmekaniikka ja logiikka nojaavat pitkälti DSCP-konfiguraatioihin, joten voidaan todeta että NBAR pyrkii hyödyntämään molempien QoS-arkkitehtuurien hyviä puolia. NBAR-tekniikka ei sisällä signalointia, vaikka mekaniikka pyrkii End-to-End palvelutoteutukseen, yhtenäisesti verkon päästä päähän. Toisaalta laitekohmainen konfiguraatio voidaan tehdä liikenteen tunnistuksen ja luokkiin jaon suhteen suoraan protokollatasolla, jolloin vaikkapa VoIP-protokollan priorisointi voidaan tehdä erittäin tehokkaasti. Vaikka Cisco mahdollistaa NBAR-tekniikassa ko-

ko verkon kattavan QoS-toteutuksen, myös kriittisten solmukohtien yksittäinen merkitseminen on mahdollista mukaillen Differentiated Servicesin konfiguraatiomalleja. (Cisco Systems Incorporated 2008d; Cisco Systems Incorporated 2009f.)

4.3 Differentiated Services ja DSCP reitittimessä

Ciscon reitittimissä käytetään QoS-palvelun toteuttamiseen IP-kehysten muokkautusta bittitasolla. IP-kehys sisältää ToS (Type of Service)-kentän, joka korvataan Differentiated Services -arkkitehtuuria toteutettaessa DiffServ-kentällä. ToS- ja DiffServ-kentät on pituudeltaan samat, mutta bittien merkitys muuttuu. Ciscon Differentiated Services -toteutuksissa käytetään hyväksi IP-kehysten ToS (Type of Service) -kentän korvaamista suoraan DiffServ-kentän DSCP (Differentiated Services Code Point)-biteillä. ToS-kenttä tosin säilyy IP-paketin kehyyksessä, mutta sitä ei huomioida, ellei DSCP-operaatio sitä erikseen velvoita. Käytännössä DiffServ-kenttä on taaksepäin yhteensopiva tekniikaltaan vanhemman ToS-kentän kanssa. (Cisco Systems Incorporated 2008e; IETF 2008.)

ToS määrittää alkuperäisessä IP-kehyyksessä muutamia reititykseen vaikuttavia ominaisuuksia, kuten paketin viive, läpimeno, ja hylkäysprioriteetti. DSCP on analogiassa näiden määreiden suhteen ToS-kentän kanssa mutta määrittää prioriteetit tarkemmin tuoden mukaan tarkoin määritetyn pakettien luokkajaon. Käytännössä reitittimet käyttävät pakettiprioriteettien uudelleenjärjestelyprosesseissaan joko ToS-kentän prioriteettibittejä (3 kpl) tai sisällyttävät prioriteettitiedon suoraan DiffServ-kentän DSCP-biteihin. Jos kaikissa verkon laitteissa on Differentiated Services -arkkitehtuuri käytössä, kannattaa käyttää priorisointiin pelkästään DSCP-bittejä QoS-tehokkuuden ja luotettavuuden takaamiseksi. ToS-kentän ja DiffServ-kentän analogia on esitetty kuviossa 7 (KUVIO 7). (Cisco Systems Incorporated 2008e; IETF 2008.)

ToS Byte

P2	P1	P0	T2	T1	T0	CU1	CU0
----	----	----	----	----	----	-----	-----

- IP precedence—three bits (P2 to P0)
- Delay, Throughput and Reliability—three bits (T2 to T0)
- CU (Currently Unused)—two bits(CU1-CU0)

DiffServ Field

DS5	DS4	DS3	DS2	DS1	DS0	ECN	ECN
-----	-----	-----	-----	-----	-----	-----	-----

- DSCP—six bits (DS5-DS0)
- ECN—two bits

KUVIO 7. ToS- ja DiffServ-kentät (Cisco Systems Incorporated 2008e).

ToS ja DiffServ-kentät koostuvat molemmat kahdeksasta bitistä. QoS-prosessien ja toteutuksen helpottamiseksi ne ovat hyvin samankaltaiset. 6 eniten merkitsevää bittiä (bitit alkaen vasemmalta) vaikuttavat priorisointiin. 2 vähiten merkitsevää bittiä eivät ole käytössä, koska niille ei ole määritelty tehtävää. 6-bitin kenttä jaetaan vielä kahteen osaan ToS-kentässä. Nämä ovat IP precedence ja Delay. Tämä jaottelu koskee myös DiffServ-kenttää, jossa 3 bittiä varataan luokkaa varten ja 3 bittiä pakettien hylkäysprioriteeteille. Käytännössä 3:lla bitillä voidaan toteuttaa 8 erilaista arvoa (0-7). DiffServ-kentässä voidaan muodostaa 3 bitillä (DS5, DS4, DS3) 8 erilaista luokkaa joiden hylkäysprioriteetti kasvaa bittien (DS2, DS1, DS0) muodostaman desimaaliluvun kasvaessa. ToS-kentässä 3 ensimmäistä bittiä (P2, P1, P0) määrittävät liikenteen priorisoinnin, jossa desimaaliluku 7 merkitsee korkeinta prioriteettia. ToS-bitit T2, T1, T0 määrittävät sellaisinaan palvelutyypin vaatimukset. T2 vaatii viiveen minimointia, T1 suurempaa kaistaa ja T0 pakettitiputusten minimointia. Bitti asetetaan arvoon 1 jos vaatimus palvelutyypistä halu-

taan esittää. Yhden bitin määräyslogiikka on sekä yksinkertainen että tehokas, jos erilaisia liikennetyyppejä on vähän. Useampitasoista liikennepriorisointia ei kuitenkaan voida tehokkaasti toteuttaa. ToS-kentän CU1- ja CU0-bitit eivät ole käytössä. DiffServ-kentän viimeiset kaksi ECN (Explicit Congestion Notification) -bittiä toteuttavat tavallisesti IP-verkoissa liikenteen tukkeutumisenestoprosesseja, mutta DiffServ-operaatioissa niitä ei käytetä. Kyseessä onkin lähinnä optio tulevaisuuden DSCP-prosessikehitystä varten. Priorisoinnin toteutus ja PHB-prosessi reitittimessä on DiffServ-kentässä paitsi moninaisempi, myös tehokkaampi. Tämä johtuu siitä että mukaan tulee kokonaan uusi luokkajako. Kuviossa 8 (KUVIO 8) on esitetty luokka- ja prioriteettijako DiffServ-arkkitehtuurille käyttämällä DSCP-kentän bittejä. (Cisco Systems Incorporated 2008e.)

Drop	Class 1	Class 2	Class 3	Class 4
Low	001010	010010	011010	100010
	AF11	AF21	AF31	AF41
	DSCP 10	DSCP 18	DSCP 26	DSCP 34
Medium	001100	010100	011100	100100
	AF12	AF 22	AF32	AF42
	DSCP 12	DSCP 20	DSCP 28	DSCP 36
High	001110	010110	011110	100110
	AF13	AF23	AF33	AF43
	DSCP 14	DSCP 22	DSCP 30	DSCP 38

KUVIO 8. DiffServ-luokkajako (Cisco Systems Incorporated 2008e).

Laitekonfiguraatioissa usein käytetty kymmenjärjestelmään perustuva DSCP- luku on suoraan käännetty kentän binääriarvosta. Edelleen AF-arvot viittaavat suoraan kronologisesti DSCP-arvoihin määrittäen loogisesti luokkajaon. AF-arvojen konfigurointi suoraan reitittimeen helpottaa luokkajaon tekemistä ja samalla varmistetaan, että DSCP-arvot menevät oikein. Cisco asettaa AF- ja DSCP-arvot ole-

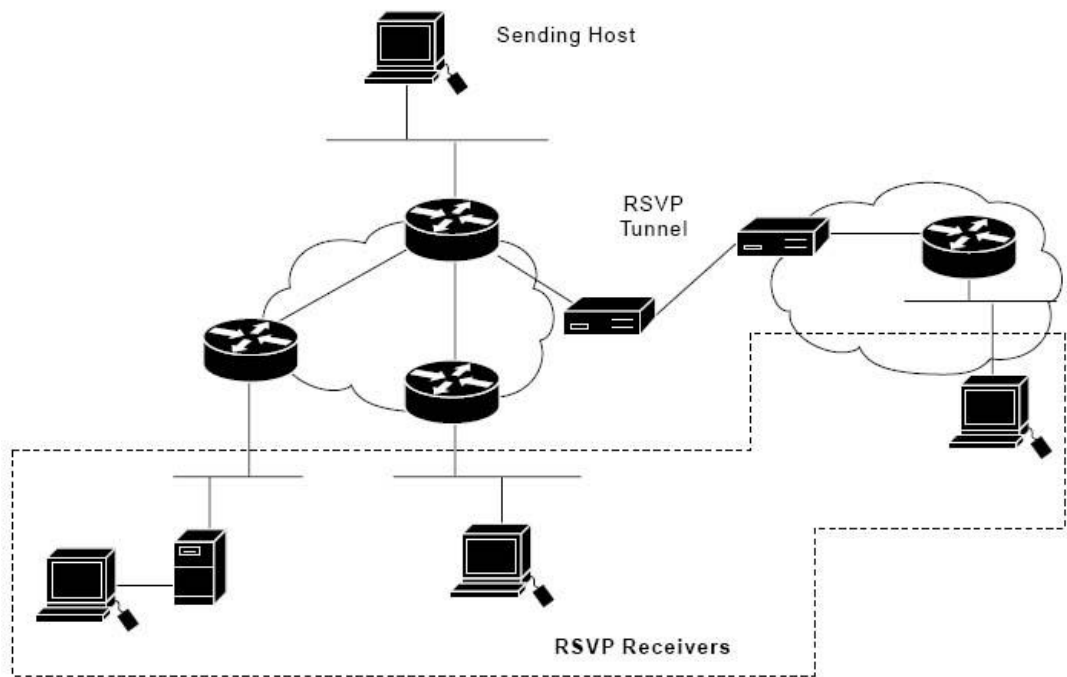
tusarvoiksi eri liikennetyypeille. Esimerkkinä mainittakoon reaaliaikainen videokuvansiirto, joka saa arvon DSCP-arvon 34. Tarpeen vaatiessa luokkahierarkia on manipuloitavissa asettamalla DCSP-arvoja uudelleen eri liikennetyypeille. Tällöin liikennekohtainen luokkajako ja priorisointi muuntuu tarpeen mukaan. Ennalta määriteltyjen sääntöjen muokkaus onkin yksi tärkeimmistä ominaisuuksista Cisccon laitekoonpanoissa, koska harvoin intressit priorisoinnin suhteen ovat muuttumattomia, saati globaalilla tasolla yhteneviä. (Cisco Systems Incorporated 2008e.)

4.4 Integrated Services ja RSVP reitittimessä

Quality of Servicen Integrated Services -arkkitehtuuri Cisccon laitekoonpanolla toteutetaan pitkälti luottaen RSVP (Resource Reservation Protocol) -protokollaan. RSVP liittyy kiinteästi reitittimien toimintaan koska usein RSVP-käytävät ajetaan niiden läpi. Protokolla ei kuitenkaan ota kantaa itse reitittämiseen, vaikka sen toiminta on kiinteästi sidoksissa verkkokerroksen operaatioihin, kuten IP-pakettien ohjaukseen ja liikenteen kontrollointiin. RSVP asetetaan OSI-mallissa kuljetuskerrokselle (OSI 4, Transport layer). RSVP:n toiminta perustuu erilaisten dynaamisten pääsyylojien (access list) generointiin, joita reititysprotokollat laskevat reitittäessään liikennettä. Priorisointi tehdään siis pääsyylojien sääntöjen mukaan, mikä on ominaista myös Cisccon Differentiated Services -arkkitehtuurin QoS-toteutuksessa. Internet Engineering Task Force (IETF) pyrkii kehittämään ja yhtenäistämään RSVP-protokollan kehitysorganisaatioiden avulla protokollan standardeja. Koska Integrated Services -arkkitehtuuri on paremmin soveltuva suurille MAN- ja WAN-verkoille, on RSVP sitä palvelevana protokollana tärkeä kehityskohde. (SDSU Communications Network Laboratory 2008; Cisco Systems Incorporated 2009a.)

Tyypillisesti RSVP aktivoidaan toimimaan kaikissa verkoissa, joissa QoS-arkkitehtuurin sisällä olevat päätelaitteet toimivat (Sending Host, RSVP Receivers). Tämä tarkoittaa sitä, että myös jokainen verkon reititin ja kytkin täytyy konfiguroida tukemaan protokollaa ja siis olemaan tietoisia priorisointisäännös-

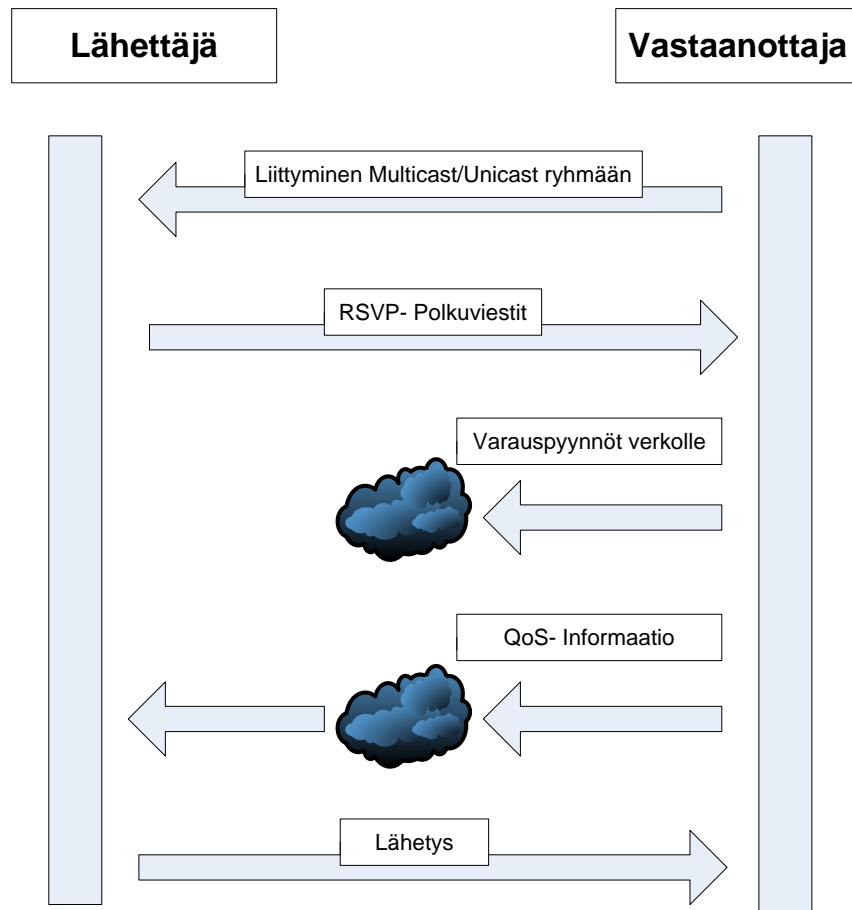
töistä. Poikkeuksena mainittakoon oletusyhdyskäytävä-reitittimien (gateway) ja host-koneiden väliin jäävät kytkimet jotka eivät vaadi RSVP-asetuksia. Yhtenäisen RSVP-ketjutus on erityisen tärkeää verkoissa, joissa esiintyy paljon ”pullonkauloja”. Kuviossa 9 (KUVIO 9) on esitetty esimerkki RSVP:tä tukevasta verkkokokonaisuudesta, jossa yksi kone (Sending Host) generoi prioriteetiltaan tärkeää liikennettä usealle kohdekoneelle (RSVP Receivers). Ilman RSVP-tukea tämä liikenne juuttuisi todennäköisesti verkon sarjalinkkeihin tai verkkojen yhdyskäytäviin. (SDSU Communications Network Laboratory 2008; Cisco Systems Incorporated 2009a.)



KUVIO 9. RSVP-verkkoympäristö (SDSU Communications Network Laboratory 2008).

Integrated Services -toteutus hyödyntää datavuotunnistusta kontrolloidessaan liikennepriorisointia. RSVP tunnistaa datavuot samanlaisiksi lähde- ja kohdeosoitteen ja QoS-määritysten perusteella. Samankaltaiseksi tunnistettua liikennettä voidaan siis käsitellä samalla tavalla taaten näin juuri tietynlaiselle liikenteelle täysin samat prioriteetit. Laitteet keskustelevat keskenään tietynlaiset vuospesifikaatiot, jotka tarkemmin määrittävät QoS-arvot liikenteelle. Samalla voidaan tehdä myös verkon kaistavarauksia minimiresursseja vaativille sovelluksille ja taata yhtenäinen sekä toimiva prioriteettihierarkia. Karkeasti liikenneprioriteetit ovat ”best-effort”, ”rate-sensitive” ja delay-sensitive”. Integrated Services -toteutuksissa määrätään ne datavuot jotka käyttävät edellämainittuja prioriteetteja. Useimmiten äänisovellukset asetetaan ”rate-sensitive”, videokuva ”delay-sensitive” ja kaikki muu liikenne ”best-effort”-luokkaan. (SDSU Communications Network Laboratory 2008.)

RSVP-protokolla toimii siten, että vastaanottajakoneet liittyvät tiettyyn multicastryhmään tietyllä kohde IP-osoitteella. Kohdeosoite määrää ryhmän. Tämän osoitteen määrää IGMP-protokolla (Internet Group-Membership Protocol) yhdessä PIM-protokollan kanssa (Protocol-Independent Multicast). Unicast-liikenteessä voidaan käyttää pelkkää unicast-reititystä. Potentiaaliset liikenteen lähettäjät lähettävät RSVP-polkuviestejä liikennettä vastaanottavan koneen osoitteeseen. Nämä viestit sisältävät tarkempaa tietoa minkälaista liikennettä olisi tarkoitus verkon yli kuljettaa. Vastaanottaja reagoi tähän lähettämällä verkkoon aktiivilaitteille varauspyyntöjä määrittäen samalla QoS-spesifikaatioita tulevalle liikenteelle. Nämä pyynnöt valuvat verkon läpi aina lähettäjälle, joka prosessin aloitti. Kun varauspyyntö on saapunut lähettäjälle, on vastaanottajalla sekä kaikilla siirtoon osallistuvilla verkon laitteilla tieto mitä liikennettä verkkoon lähetetään ja mitkä sen prioriteetit ovat. Tätä prosessia kuvataan tarkemmin kuviossa 10 (KUVIO 10). Huomattavaa on, että RSVP-prosessissa vastaanottaja käynnistää varausneuvottelut eikä lähettäjä. Tämä vaikuttaa myös Ciscon laitekonfiguraatioon merkittävästi. (SDSU Communications Network Laboratory 2008.)



KUVIO 10. RSVP-varausneuvottelu.

RSVP erottelee eri liikenneluokkiin kuuluvat IP-paketit erilaisia liikennesiirtosessioita analysoimalla. Liikenne voi olla joko unicast- tai multicast-liikennettä jolla on tietty lähde- ja kohdeosoite tietyllä porttiarvoilla (TCP tai UDP). Samaa liikennettä kuljettavissa siirtosessioissa nämä arvot ovat samat, mikä johtaa siihen, että QoS:ää toteuttava laite siirtää sen tiettyyn prioriteettiluokkaan. Samassa verkossa voi siis olla useaa erilaista liikennettä samaan aikaan, mutta jokaisella liikennetyypillä on aina IP-paketissa yksilöllinen reititystieto. Tämä tieto yksilöi siitä RSVP-protokollalle tunnistettavan. RSVP:n ideana onkin, että sekä host-koneet että reitityksen aktiivilaitteet osallistuvat aktiivisesti QoS-määrittelyyn ja toimintaan. Host-kone siis pyytää varausta tietylle sovellukselleen. Reititin taas ottaa pyynnön vastaan joko hyväksymällä tai hylkäämällä sen. Jos pyyntö hyväksytään, reitittimen tehtävä on siirtää tieto muiden reitittimien välityksellä koko verkolle.

Tästä johtuen RSVP:n on katettava koko verkko. Jos välillä on laitteita, jotka eivät tunnista tai osaa käyttää protokollaa, on niiden välille tehtävä RSVP-tunneli (RSVP Tunnel), jonka avulla RSVP-tieto ikäänkuin hyppää prosessiin osallistumattomien laitteiden yli jatkaen niiden takana oleville RSVP-laitteille. (SDSU Communications Network Laboratory 2008.)

RSVP-liikenteen kuljetus suuremmissa verkoissa voi kuormittaa reitittäjiä. Tämä johtuu tavallisesti siitä että reitittimet joutuvat prosessoimaan useita eri liikennevirtoja jotka sisältävät kukin erilaisia priorisoitua liikennettä. Tällöin usein myös verkon skaalautuvuus asetetaan suureen haasteeseen. Ciscon RSVP-toteutuksessa varautuminen prosessorikuormitukseen ja skaalautuvuuden haasteisiin on otettu jo ennalta huomioon ohjelmistollisesti. Ciscon reitittimet pystyvät suuremmissa kokoonpanoissa käsittelemään jopa 10000:tta erilaista RSVP-liikennevuota verkon siirtä kärsimättä. Ciscon 3600- ja 7200- sarjan reitittimillä liikenneviiveet reitittimeltä toiselle ovat myös varsin pienet. Viive reitittimeltä toiselle on luokkaa 10ms. Näin ollen 10 reititinhypyn päässä olevien IP-puhelimien puhelunaloitus veisi vain sekunnin kymmenyksen. Tämän jälkeen QoS-priorisoitu linja on auki ennalta määritetyllä priorisointitasolla. (Cisco Systems Incorporated 2009a.)

4.5 Class of Service (CoS) ja Auto-QoS

Siirtoyhteyskerroksen (OSI-2) IEEE 802.1Q-kehyksessä esiintyvä Tag-kontrolli-informaatiokenttä määrittää liikenneprioriteetteja kytkinverkoissa. Kehystasolla tämä tehdään muuttamalla CoS (Class of Service)-arvoa. Prosessi toteutetaan kolmen eniten merkitsevän bitin prioriteetikombinaatioilla. Näitä on yhteensä 7 erilaista, joista numeron 7 omaava kehys on prioriteetiltaan suurin eli liikennetyypiltään tärkein. CoS-arvoa käytetään yleisimmin kytkimien Quality of Service-toteutuksissa koska data enkapsuloidaan ylempältä verkkokerrokselta mentäessä kehystasolle. Tällöin se ei vaikuta itse reitittimien QoS-politiikkaan, mutta saattaa vaikuttaa QoS:n toimivuuteen varsinkin verkossa, jossa on paljon kytkimiä. (Cisco Systems Incorporated 2008j.)

Quality of Service voidaan Ciscon VoIP-toteutuksissa (Cisco softphone) konfiguroida oletusasetuksena, jolloin vaihtoehdot palvelulle ovat yksinkertaistaen joko ”päällä” tai ”pois”. Tällöinkin kytkimen sisääntuloportteihin määritellään CoS-to-DSCP-sidos, jolloin verkkokerroksen DSCP-priorisointitieto voidaan siirtää suoraan siirtoyhteyserroksen QoS-operaatioiden käyttöön. Kytkinporttikohmainen QoS sisältää rajoitetun määrän liikennetyyppejä sekä DSCP- että CoS-arvoille. Lisäksi DSCP ja CoS-arvojen keskinäiset suhteet ovat ennaltamäärätyt oletusasetuksina. Arvot on valmiiksi sidottu liikenteen sisään- ja ulostulojonoihin, joilla niilläkin on ennaltamäärätyt viive- ja puskuriarvot. Tästä johtuen Ciscon kytkimien yhteydessä puhutaan usein termistä, Auto-QoS, joka toteuttaa tiettyä oletussäännöstöä. Auto-QoS voi olla myös oheislaiteriippuvainen tukien esimerkiksi vain tiettyjä IP-puhelintyypppejä tai ohjelmistoja. Tällöin laite ilmoittaa suoraan liikennetyypin, jota kytkin ei muuten tunnista QoS-operaatioissaan. (Cisco Systems Incorporated 2008g.)

Useimmiten Auto-QoS tyydyttää tarvittavat QoS-tarpeet verkossa, koska sen säännöstö on asetettu palvelemaan kaikkein yleisimpiä verkkoliikenteen priorisointitarpeita. Näistä mainittakoon esimerkiksi aiemmin esitellyn VoIP-liikenteen priorisointi muun liikenteen edelle. Auto-QoS-palvelu on oletusasetuksena pois Ciscon laitteissa. Jos tätä optiota halutaan käyttää, astuu voimaan pakettien kategorisointi ja merkintä kuvion 11 (KUVIO 11) mukaisesti. CoS-arvojen suhteita DSCP-arvoihin voidaan muuttaa konfiguraatioissa, jolloin esimerkiksi CoS-arvon 5 yhteyteen voidaan siirtää mikä tahansa DSCP-arvo. Tällöin voidaan VoIP-liikenteen sijasta antaa jollekin muulle liikennetyypille parempi prioriteetti. Operaatio on hankala ja vaatii kaikkien Auto-QoS-oletusarvokomentojen uudelleenasettamisen kytkimeen. Ennaltamäärätyn arvotaulukon manipulointi on käytännössä usein viimeinen keino, jos Quality of Serviceä ei muuten voida toteuttaa. Suuri ongelma on myös se, että liikenteen tunnistusmekanismit perustuvat pelkästään DSCP-arvoihin rajatulla määrällä erilaisia liikennetyyppejä. Lisäksi kaistanvaraukset ja puskurit noudattavat aina neliportaista jonomekanismia, jossa arvot on pitkälti ennaltasidottu. (Cisco Systems Incorporated 2008g.)

	VoIP ¹ Data Traffic	VoIP Control Traffic	Routing Protocol Traffic	STP BPDU Traffic	Real-Time Video Traffic	All Other Traffic	
DSCP	46	24, 26	48	56	34	-	
CoS	5	3	6	7	4	-	
CoS-to- Ingress Queue Map	2, 3, 4, 5, 6, 7 (queue 2)					0, 1 (queue 1)	
CoS-to- Egress Queue Map	5 (queue 1)	3, 6, 7 (queue 2)			4 (queue 3)	2 (queue 3)	0, 1 (queue 4)

¹ VoIP = voice over IP

KUVIO 11. Liikennetyypit, pakettimerkintä ja jonot (Queue) (Cisco Systems Incorporated 2008g).

Kuviossa 11 (KUVIO 11) on esitetty CoS (Class of Service) arvoille määritellyt oletusarvolliset jonot (queue). Kehyksien liikennepriorisointi nojaa sille määritellyyn CoS-arvoon, joka kuvion 11 mukaan on vahvasti sidoksissa siihen, mihin jonoon (queue) kehykset menevät. DSCP-arvot ovat taas suoraan sidoksissa CoS-arvoihin. Liikkumavaraa priorisoinnin suhteen ei juurikaan ole. Esimerkiksi kriittinen VoIP-liikenne arvolla 5 menee aina Egress (systemistä lähtevä porttikohdainen liikenne asiakkaan tai käyttäjän päähän) -suunnassa 1. jonoon, kun taas toisaalta Ingress (systemin porttikohdainen sisääntuleva liikenne) -suunnassa se voi olla 2. jonossa. QoS-tehokkuuden kannalta tuloportit eivät ole yhtä kriittisiä kuin lähtöportit. Lähtöportit määräävät prosentuaalisen kaistanvarauksen, joka on yleensä priorisoidun liikenteen osalta merkittävä lähetettäessä liikennettä kohti verkon pullonkauloja. Kerran priorisoitu liikenne säilyttää QoS-arvonsa päätepisteeseensä saakka, ellei sitä priorisoida uudelleen matkalla. Huomattavaa on kuitenkin, että Auto-QoS-mekanismi asettaa videokuvan vasta 3:een jonoon Egress jonoissa, mikä ei sinänsä vastaa nykypäivän QoS-asetuksia ottaen huomioon multim mediasovellusten yleistymisen tietoverkoissa. (Cisco Systems Incorporated 2008g.)

Kuviossa 12 (KUVIO 12) ja kuviossa 13 (KUVIO 13) on listattu jonokohtaisia prioriteettiominaisuuksia sekä kaistankäytössä että kaistanvarauksissa. Kuvioissa liikennepriorisoinnin näkökulmasta mielenkiintoisimpia tietoja ovat prosentuaalinen kaistankäyttö (Queue weight), sekä liikennevarausmäärät (Queue size).

Ingress Queue	Queue Number	CoS-to-Queue Map	Queue Weight (Bandwidth)	Queue (Buffer) Size
SRR shared	1	0, 1	81 percent	67 percent
Priority	2	2, 3, 4, 5, 6, 7	19 percent	33 percent

KUVIO 12. Ingress Queues (Cisco Systems Incorporated 2008g).

Egress Queue	Queue Number	CoS-to-Queue Map	Queue Weight (Bandwidth)	Queue (Buffer) Size for Gigabit-Capable Ports	Queue (Buffer) Size for 10/100 Ethernet Ports
Priority	1	5	up to 100 percent	16 percent	10 percent
SRR shared	2	3, 6, 7	10 percent	6 percent	10 percent
SRR shared	3	2, 4	60 percent	17 percent	26 percent
SRR shared	4	0, 1	20 percent	61 percent	54 percent

KUVIO 13. Egress Queues (Cisco Systems Incorporated 2008g).

Sekä tulo (ingress) - että lähtö (egress) -jonot sisältävät sekä ”priority”-arvon että ”SRR (shaped round robin) shared”-arvon. Oletusarvollisesti yksi liikennetyyppi (Queue 1) saa korkeimman prioriteetin, jolloin se käyttää nelkein kaiken saatavilla olevan kaistan. Loput liikennetyypit on jaettu jonoihin 2 (Ingress) ja 2-4 (Egress). Nämä jonot saavat käyttöönsä jäljelle jäävän kaistan. SRR-jonon sisäiset liikennetyypit kilpailevat jonolle asetetusta kokonaiskaistasta. Tästä syystä korkealle priorisoitu jono tulisi sisältää vain yhtä liikennetyyppiä. (Cisco Systems Incorporated 2008g.)

QoS-toimivuuden kannalta tärkeimmät jonot ovat egress-jonot (4 kpl), joiden liittymien takana verkon pullonkaulat usein sijaitsevat. Pääjono (priority) tulisi konfiguroida lähelle 100-prosenttia, jos se sisältää VoIP- tai videokuva-liikennettä. Tällöin vastoin kuvion 13 Auto-QoS-oletusasetuksia, SRR-jonot 2, 3 ja 4 saavat usein vain muutaman prosentin kaistasta käyttöönsä. Käytännön toteutuksissa tämä ei useinkaan haittaa, sillä priorisoitu liikenne käyttää harvoin koko kaistaa yhämittaisesti. Kaistan vapautuessa pääjonolta SRR-jonon varausprosentit muuttuvat painoarvoiksi kaistajaossa, jolloin tyhjä kaista käytetään täysimittaisesti hyödyksi. Hyvä käytäntö on asettaa SRR-jonon yhteenlasketut prosentuaaliset kaistanvaraukset 100 prosenttiin, jotta vapautuvat siirtoresurssit saadaan tehokkaasti ja tarkasti hyödynnettyä.

4.6 Verkkosuunnittelu ja laitevalinnat

Suunniteltaessa kahden arkkitehtuurin Quality of Service-toteutusta, on erittäin tärkeää suunnitella verkkotopologia huolella ja suorittaa laitevalinta siten, että halutut palvelut ovat tuetut. Lisäksi olisi hyvä varautua jo ennalta verkon tai palvelutoteutusten muutoksiin. Tällöin ennaltaehkäistään jo varhaisessa vaiheessa ylimääräistä työtaakkaa verkon ja QoS-palvelun ylläpidossa. Vaikka tässä työssä testiympäristöksi valittiin valmiin verkon osa, itse topologia ja palvelutoteutus oli suunniteltava edellämainittujen suositusten mukaisesti ottaen huomioon myös laitteistokohtaiset resurssit. Tärkeitä verkon ja laitteiden ominaisuuksia ovat muunmuassa toiminnot, luotettavuus, käytettävyys, joustavuus ja kustannukset. Näistä viimeisin oli testiverkon kannalta kuitenkin merkityksetön, koska kaikki testit tehtiin oppilaitoksen tiloissa. Sen sijaan toiminnot ja käytettävyys nousivat tärkeään asemaan. Alkuperäisen teorialueiden pohjalta rakennetun toteutussuunnitelman onnistuminen käytännössä oli pitkälti kiinni juuri edellämainituista kriteereistä. (Ballew 1998, 27-29.)

QoS-toteutuksissa reitittimien rooli on usein tärkein. Myöskään reititinmallin ja ohjelmiston valinta QoS-palvelutoteutuksessa ei ole yhdentekevää jos halutaan saada aikaiseksi toimiva palveluratkaisu. Reitittimien täytyy tukea ohjelmistollisesti QoS-toteutuksia niin kattavasti että tutkimustavoitteet voitiin saavuttaa. Sama vaatimus koski myös myöhemmässä vaiheessa suoritettua kytkinvalintaa. Yleisesti reitittimien valinnassa tärkeitä lähtökohtia ovat itse reititystuki ja protokollat, hallintatyökalut sekä pakettisuodatus. Vaikka Ciscon reitittimissä on paljon erilaisia ominaisuuksia, Quality of Service -toteutuksien yhteydessä kannattaa keskittyä edellämainittujen kriteerien lisäksi pääasiassa vain QoS-protokollatukeen. Lisäksi on hyvä selvittää, voiko jonkin puuttuvan ominaisuuden QoS-toteutuksessa korvata tai kokonaan ohittaa vaarantamatta itse toimivuutta. Yksinkertaistaminen saattaa myös helpottaa verkkosuunnittelua ja tehdä QoS-palvelun ylläpidosta yksinkertaisemmän. Jos jotain palvelua tai toimintoa ei tarvita nyt tai tulevaisuudessa, kannattaa se sivuuttaa jo alkuvaiheessa. (Ballew 1998, 86-89.)

5 QOS-TOTEUTUS CISCON LAITTEILLA

5.1 Testiympäristön kuvaus

Quality of Service- palvelun testiympäristö rakennettiin viidestä Ciscon 2801 -reitittimestä. Konfiguraatiot laitteisiin toteutettiin IOS-ohjelmistoversiolla: Cisco IOS Software, 2801 Software (C2801-ADVENTERPRISEK9-M), Version 12.4(15)T5, RELEASE SOFTWARE (fc4). Reitittimet muodostivat lähiverkon rungon. Keskusreitittimen roolina oli toimia vasemman ja oikean verkkopuolen solmukohtana. 2800-sarjan reitittimet valittiin siksi, että kyseisissä laitteissa on sekä tuki että valmiudet ohjelmistossa toteuttaa sekä Differentiated Services että Integrated Services -arkkitehtuuria. Reitittimien rooli QoS-palvelun toteutuksessa oli varsin keskeinen. Ne toimivat verkossa solmupisteinä kahden eri QoS-arkkitehtuuritoteutuksen sekä testikoneiden välillä. Reitittimissä sekä fyysiset liittytävät että loogiset konfiguraatiovajavaisuudet heijastuvat välittömästi itse QoS-palvelun toimivuuteen. Valittu reititinmalli mahdollisti myös kattavat konfiguraatio-optiot muokata ja varioida QoS-palvelun toteutusta. (Cisco Systems Incorporated 2008h.)

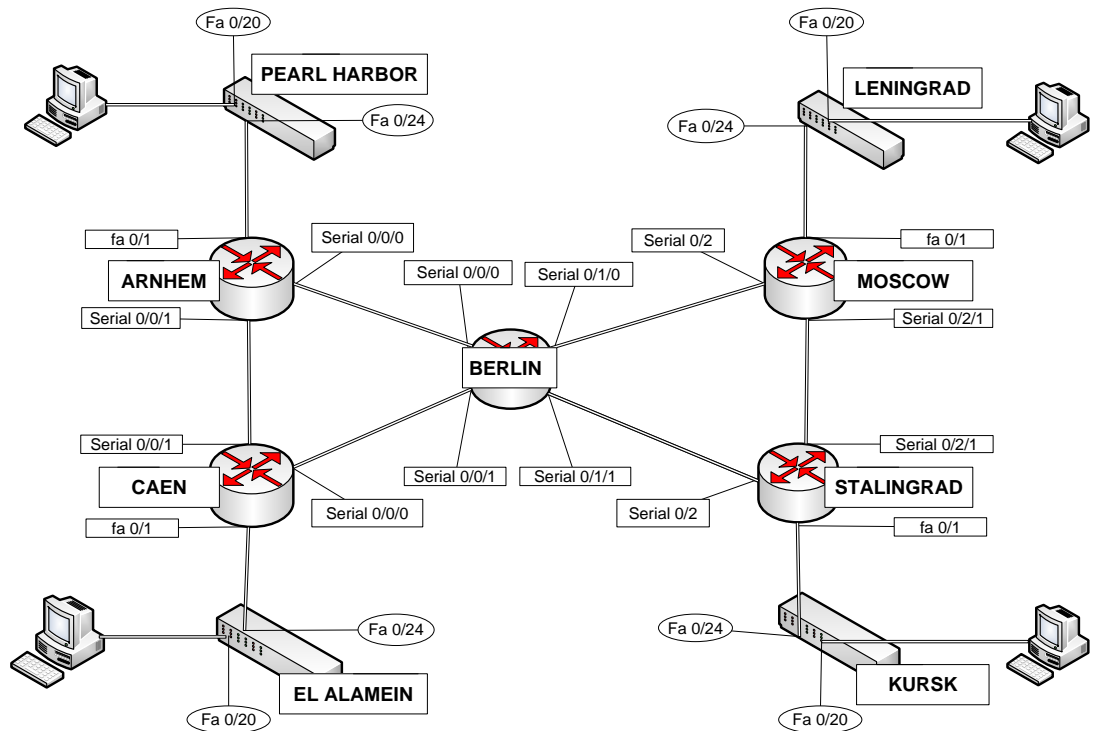
Verkon Differentiated Services -laidalle sijoitettiin kaksi kappaletta Ciscon kytkimiä mallia: Cisco Catalyst WS-C2960-24TT-L (PowerPC405) processor (revision D0). Kytkimien ohjelmistoversiona testien aikana käytettiin seuraavaa: Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 12.2(44)SE, RELEASE SOFTWARE (fc1). Integrated Services -verkkoon sijoitettiin kaksi Ciscon kytkintä mallia: Cisco Catalyst ME-3400-24TS-A ohjelmistoversiolla 12.2(44)SE2. Catalyst 2960 -kytkimet soveltuivat testikäyttöön molemmille Quality of Service -arkkitehtuureille, mutta syy miksi Catalyst ME 3400 -kytkimiä käytettiin, johtui halusta haastaa myös verkkotason reitittävä kytkin QoS-toteutukseen. Käytännössä ME-3400- kytkin täydensi Integrated Servicen laitaverkkoa vaihtoehtoisena toteutusratkaisuna reitittimien RSVP-toteutuksen rinnalle. (Cisco Systems Incorporated 2008i.)

Erityisen tärkeää kytkimien ominaisuuksissa oli tuki verkkotason konfiguraatiolle sekä protokollille siirtoyhteyskerroksen palvelujen lisäksi. Olennaisinta QoS-palvelua toteuttavalle kytkimelle oli lähtökohtaisesti kyky prosessoida IP-pakettien DSCP-tunnistekenttiä ja kuljettaa niiden tietoja edelleen siirtoyhteyskerrokselle. Ciscon 2960-sarjan kytkin mahdollisti QoS-palvelun loogisen implementoinnin tukemaan itse testirunkoverkon reititointimintaa. ME 3400-sarjan kytkimellä reititinominaisuudet olivat jo lähtökohtaisesti olemassa. Vaikkakin QoS-toteutus testiverkossa nojasi varsin pitkälle reititinkonfiguraatioon, oli tärkeää että itse kytkimissä oli myös valmiudet vaikuttaa verkon reunan QoS-prosessiin, ennen liikenteen ohjaamista verkon päätelaitteisiin tai vastaavasti päätelaitteilta pulonkauloina toimiviin sarjalinkkeihin. (Cisco Systems Incorporated 2008i.)

Ciscon laitteiston lisäksi verkon testaamiseen ja testiliikenteen generointiin valjastettiin kaksi DELL:in Pentium 4-prosessorilla varustettua pöytäkonetta joita käytettiin vain testiliikenteen luomiseen ja vastaanottamiseen. QoS-toteutuksessa testikoneet eivät kuuluneet itse palvelun loogiseen toteutukseen, koska jokaisella verkoon liiketyllä koneella (host) on jo oletusarvoisesti valmiudet lähettää ja vastaanottaa QoS-verkon liikennettä sellaisenaan, palvelua tukevan verkon kautta. Lähiverkkokaapelointina käytettiin 100 Mb Ethernet-kaapelointia sekä sarjalinkeissä 1,5 Mb sarjalinkkejä. (Cisco Systems Incorporated 2008i.)

5.2 Verkkotopologia

Testiverkko toteutettiin kahtena erillisenä verkkona, jotka yhdistettiin keskusreititimellä toisiinsa. Verkon laitteiden IP-osoitteet sekä verkko-osoitteet varattiin IP-osoiteavaruudesta 192.168.0.0 - 192.168.130.254. Kuviossa 14 (KUVIO 14) on esitetty testiverkko, jossa Quality of Service -palvelun testiliikennettä ajettiin.



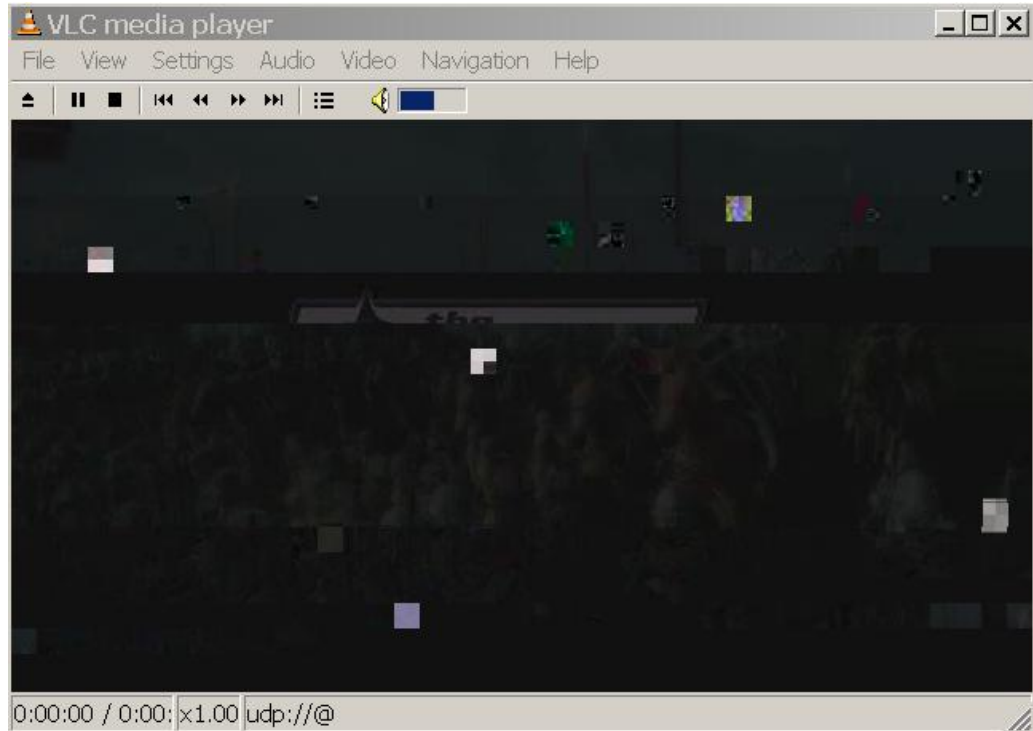
KUVIO 14. QoS-testiverkkotopologia.

Verkon oikeaa puolta käytettiin Differentiated Services-arkkitehtuurin testaukseen ja vasenta puolta Integrated Service -arkkitehtuurin testeihin. Näiden testien aikana keskusreitittimen sarjalinkit pidettiin suljettuina, jotta testiverkko olisi mahdollisimman eristetty häiriöiltä. Quality of Service -testien ja liikennepriorisoinnin osalta kiertoteiden sulkeminen oli välttämätöntä mahdollisimman oikeanlaisten tapauskohtaisten testitulosten saamiseksi. Reitittimiin konfiguroitiin valmiudet toteuttaa reititystä molempien QoS-arkkitehtuurien välillä sekä verkon päästä päähän. QoS-toteutus kahdella reitittimellä antaa testien valossa riittävän varmuuden konfiguraatioiden toimivuudesta, koska reitittäviä yhdistävä sarjalinkki toimii verkon kriittisenä liityntänä, liikenteen pullonkaulana.

Varsinaisten testien osalta lähtötilanteessa käytettiin peruskonfiguraatiota, jossa laitteille asetettiin IP-osoitteet ja reitittimille asetettiin reititystaulut käyttäen reitittävänä protokollana RIP-protokollaa. Testiverkossa striimattiin videokuvaa kahden host-koneen välillä niin, että toinen toimii striimauspalvelimena ja toinen vastaanottavana osapuolena. Striimauksen ajaksi verkkoon generoitiin kaistaa reilusti

vievää FTP-liikennettä. Striimaus toteutettiin unicast-tyyppisenä UDP-protokollan yli, mutta käytännössä samaa voidaan soveltaa multicast-liikennettä ja -protokollia tukevassa verkossa. UDP-kuljetus striimauksessa sopii hyvin varsinkin pienille verkoille, koska toteutus on yksinkertainen, eikä pakettien perillemeno tarvitse juurikaan valvoa. Tämä johtuu siitä, että pakettihäviöt ovat odotusarvoisesti pienet. Verkon sarjalinkit reitittimien välillä toimivat kokeellisina pullonkauloina, joissa striimatun liikenteen läpimeno laadulliset minimivaatimukset täyttäen onnistuttiin toteuttamaan vain QoS-palvelua hyväksikäyttäen. QoS-palvelun välttämättömyys tällaisessa verkossa johtui paljolti sarjalinkkien fyysisen kuljetuskapasiteetin vajavaisuudesta verrattuna striimaussiirtonopeuteen ja siitä aiheutuvaan verkon ylikuormitukseen. (Keränen 2009; Wikipedia 2008f.)

Videokuvan striimaus valittiin testejä varten puhtaasti sen tietoliikenneteknisesti haasteellisten ominaisuuksien vuoksi. Striimattu videoliikenne mahdollistaa myös välittömän tulosreflektoinnin ilman erillistä mittauskalustoa. Yleensä QoS palvelu haastetaan takaamaan VoIP-liikenne häiriöttömästi verkon läpi, mutta tarvittavien testitulosten saamiseksi ei ole juurikaan väliä, mitä liikennetyyppiä priorisoitava liikenne on. Perusidea on, että jollekin tai joillekin liikennetyypeille taataan palvelullisesti riittävä kaista ja loput kaistan kapasiteetista on ”best effort”- tyyppistä. Koska striimattu media ”avataan” samalla hetkellä, kun sitä vielä lähetetään, on riittävä kaistansaataavuus erittäin tärkeää koko lähetyksen ajan. Jos kaistan saataavuutta ei voida taata QoS-palvelun tai fyysisen kaistan avulla, videokuva pikselöityy pahasti. Kuviossa 15 (KUVIO 15) on esitetty striimattu videokuvaotos tukkoisessa verkossa.



KUVIO 15. Striimattu videokuva riittämättömällä kaistanleveydellä.

Häiriöinen videokuva voidaan korjata asettamalla QoS-palvelu verkon aktiivilaitteille. Kuviossa 16 (KUVIO 16) nähdään nyt samassa verkossa QoS-palvelun avulla striimattu täysin sama videolähetys.



KUVIO 16. Striimattu videokuva Quality of Serviceä tukevassa verkossa.

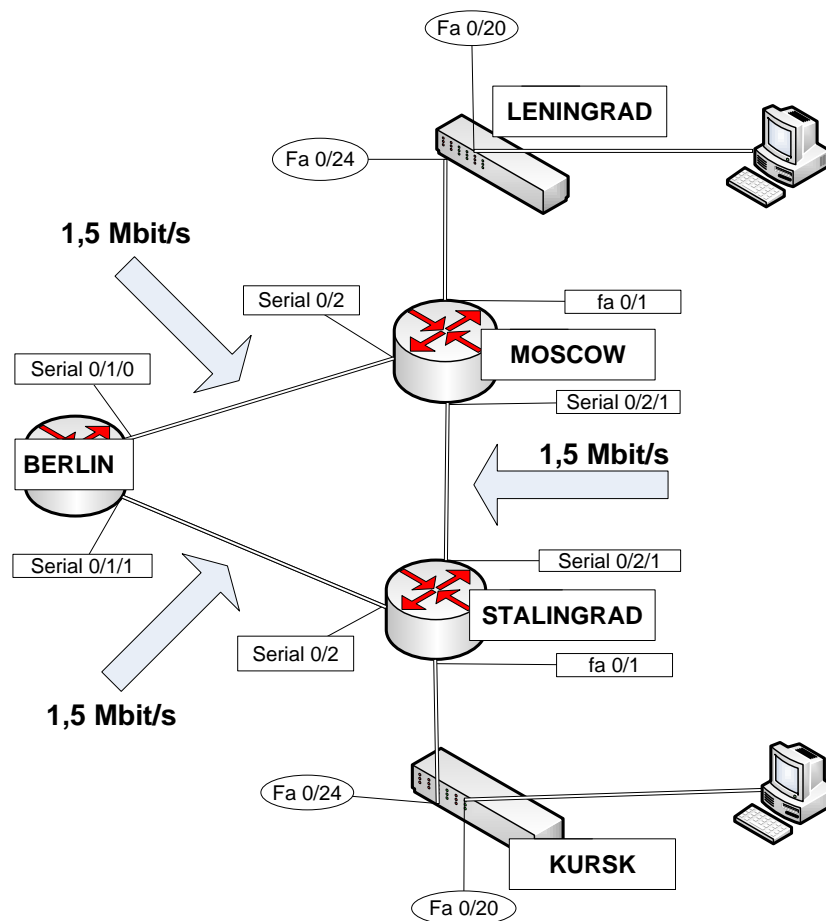
Video on MPEG-2-pakattu suhteellisen paljon yksityiskohtia sisältävä kolmen minuutin pituinen videoleike, jota ajettiin jatkuvalla toistolla verkon läpi. Jotta kuviossa 16 näkyvä kuvanlaatu saatiin aikaiseksi, vaadittiin täydellinen liikenne-priorisointi suhteessa verkon kapasiteettiin. Ilman Quality of Serviceä videoleike paitsi pikselöityi ja sumeni, pysäytti striimauksen joko kokonaan tai osittain aiheuttaen useiden sekuntien katkoksia. Myöhemmissä luvuissa esitetään yksityiskohdaisesti, miten testiverkossa toteutettiin Quality of Servicen arkkitehtuurit niin, että lopputuloksena on kuvion 16 esittämä videokuvanlaatu.

5.3 Differentiated Services -toteutus

5.3.1 Reititinkonfiguraatiot

Differentiated Services -arkkitehtuurin testaamiseen varattiin testiverkon (KUVIO 14.) oikea puoli. Testireitittiminä käytettiin reitittimiä ”MOSCOW” ja STA-LINGRAD”. Kyseisten reitittimien konfiguraatiot on tarkemmin esitelty liitteissä 1 ja 2 (LIITE 1, LIITE 2). Lisäksi keskusreitittimenä toimiva ”BERLIN”, joka yhdistää Integrated Services- ja Differentiated Services -arkkitehtuurit, osallistuu myös DSCP-operaatioihin. BERLIN-reitittimen konfiguraatio on esitelty liitteessä 3 (LIITE 3).

Differentiated Servicen toteutus Ciscon 2801-reitittimeen nojaa OSI-3-tason palveluihin ja DSCP:n käyttöön liikenteen kategorisoimiseksi. DiffServ-palvelu toteutetaan testiverkossa päästä päähän -liikenteenä, testikoneelta testikoneelle striimattuna videokuvana. Palvelun toteutus nojaa reitittimien kykyyn havaita ja tunnistaa videokuva muusta liikenteestä, joka testiolosuhteissa oli satunnaista verkon yli siirrettyä dataa, ei kuitenkaan videokuvaa. Työssä käytetty Differentiated Services -testiverkko on esitetty kuviossa 17 (KUVIO 17).



KUVIO 17. Differentiated Services –testiverkko.

Riippumatta striimaussuunnasta reitittimien konfiguraatiot olivat hyvin samankaltaiset. Tämä johtuu siitä, että lähettävän host-koneen puolella oleva reititin suodattaa liikenteen ja muut verkon aktiivilaitteet vain siirtävät sen suodatettuna eteenpäin. QoS-luokitus tapahtuu siis vain yhdessä verkon laitteessa eikä sitä tarvitse toistaa muissa. Jos siirtosuunta olisi aina yksisuuntaista, tarvittaisiin testiverkon tapauksessa vain yksi reititin ja yksi konfiguraatio. Testeissä liikennettä kuitenkin generoitiin molempiin suuntiin. Koska striimaus toteutettiin unicast-liikenteenä aina UDP-portin 1234 kautta, ei testiverkon konfiguraatiota voida missään nimessä käyttää kaikissa verkkototeutuksissa. Periaate on kuitenkin sama, mutta konfiguraatioparametrit muuttuvat tapauskohtaisesti. Seuraavassa tarkempi katselmus tärkeimpiin konfiguraatioparametreihin.

Reititinkonfiguraatiossa esiintyy seuraava komentorivikokonaisuus, jota konfiguraation ymmärtämiseksi on tarkisteltava lähemmin:

```
class-map match-all Video
  match access-group 101
class-map match-all premium
  match ip dscp ef
!
!
policy-map RTSP
  class premium
    bandwidth 1152
policy-map SetDSCP
  class Video
    set ip dscp ef

access-list 101 permit udp any any eq 1234
```

”Class-map match-all”-komennon tarkoituksena on ohjata kaikki siihen kuuluvat luokat (class) tunnistautumaan reitittimen pääsyylistoja (access-list) hyväksi käyttäen. Tässä tapauksessa viitataan access-listaan 101. Access-listat on asetettu tiettyin parametrein tunnistamaan videoliikenne muusta liikenteestä. Tässä tapauksessa havaitaan siis UDP-portin 1234 liikenne, jota videostriimaus käyttää kohdeportinaan. ”Policy-map”-komento sisältää puolestaan tiedon mitä luokan sisältävälle liikenteelle tehdään kun se on tunnistettu. Konfiguraatioesimerkissä asetetaan luokan ”Video” DSCP-arvo. Tämä arvo on ”ef” (DSCP 46) ja sillä tarkoitetaan tässä yhteydessä termiä ”expedited forwarding”. Yleensä ef-arvolla viitataan aina VoIP-liikenteeseen, mutta tässä yhteydessä sen valintaa perustellaan esimerkkiluontoisena todisteena siitä, että DSCP-arvoja voidaan valjastaa minkä tahansa liikennetyypin käyttöön missä tahansa yhteydessä. Tämä ominaisuus tekee DSCP-pohjaisesta QoS-toteutuksesta erittäin muuntautumiskykyisen. (Cisco Systems Incorporated 2008k.)

Edellistä konfiguraatioesimerkkiä mukailten voidaan havainnoida prosessia askel askeleelta:

1. Pääsyylistaa 101 koskettava liikenne saa luokka-arvon "Video".
2. Policy-map muuttaa "Video"- luokan DSCP arvon arvoon ef (46).
3. Class-map premium kerää kaiken liikenteen jolla on DSCP arvo ef (46).
4. Policy-map antaa premium-liikenteelle varauksen 1152 bit/s.

Lopputuloksena varataan siis 1,5 Mbit/s kaistasta 1152 bit/s. QoS-toteutus ei siis salli koko kaistan varausta. Kyseessä on Ciscon ohjelmistollinen rajoite johon täytyy varautua myös verkkosuunnittelussa. Class-map- ja policy-map-parametrit voivat olla mitä tahansa määreitä, tunnuksia tai tekstiä, kunhan ne esiintyvät samanlaisina molemmissa komennoissa. Tässä tapauksessa käytettiin siis parametreja "Video", "premium", "RTSP" ja "SetDSCP", mutta parametrit voidaan valita täysin vapaasti. (Cisco Systems Incorporated 2008k.)

Reititinkomennot on vielä asetettava liityntöihin (interface). Tässä kohtaa on otettava huomioon striimattavan liikenteen sisään- ja ulosmenosuunnat reitittimessä. Kuten aikaisemmin mainittiin, yksi reititin pystyy hoitamaan QoS-valvonnan vain yhteen suuntaan. Tämä johtuu siitä että DiffServ-reitittimen on oltava lähettäjän puolella jo sen vuoksi että välissä voi olla pullonkauloja muodostavia sarjalinkkejä. Seuraavassa esitellään liityntäkohtaiset konfiguraatiot, jotka viittaavat reitittimen policy-map-parametreihin. Liityntäparametrit viittaavat siis suoraan policy-map-parametreihin, jotka esitetään edelleen "service-policy"- komentoina:

```
interface FastEthernet0/1
  service-policy input SetDSCP

interface Serial0/0/0
  service-policy output RTSP
```

Ensimmäinen rivi asettaa halutun DSCP-arvon videoliikenteelle. Tämä parametri on asetettava sisäänmenoliityntään, koska sen tehtävänä on tunnistaa liikenne ja määrittää DSCP-arvot. Jälkimmäinen rivi viittaa komentoon ”policy-map RTSP”, joka asettaa kaistanvarauksen ja sitä kautta prioriteetit tunnistetulle liikenteelle. (Cisco Systems Incorporated 2008k.)

Konfiguroinnissa on syytä kiinnittää huomiota reititinkohtaisesti siihen, mistä liitynnästä tuleva liikenne otetaan vastaan. Mikäli mahdollista, liityntöihin kannattaa tehdä varmistava kaksoiskonfiguraatio niin, että kaikki liitynnät toimivat sekä sisään- että ulostulo-liityntöinä samanaikaisesti. Tärkeintä on kuitenkin huomata että sisääntuloliitynnät määräävät aina DSCP-arvot ja ulkotuloliitynnät kaistanvarauksen. Lisäksi on syytä muistaa, että pullonkaulan takana oleva reititin ei voi QoS-määrittelyä enää aktivoida tai jälkikäteen toteuttaa, koska liikenne on ennen sille tuloa jo kuristunut. Voidaankin sanoa että DiffServ-toteutus Ciscon reitittimissä on aina ”per-hop”-tyyppistä, ei niinkään topologiakohtaista. Suuremmissa verkkokokonaisuuksissa toimivan, koko verkon kattavan konfiguraation luominen on haastavaa, koska jokainen reititin toimii yksilönä eikä ota kantaa naapurireitittimen konfiguraatioon. Laitteiden välillä ei siis ole minkäänlaista viestintää tai signalointia verkon muutoksista tai QoS-säännöistä. Vastaavasti pienissä verkoissa DiffServ-toteutukset ovat varsin helppoja, koska konfiguraatio voidaan toteuttaa laitekohtaisesti vain kriittisiin verkon solmupisteisiin (Cisco Systems Incorporated 2008k).

5.3.2 Kytkinkonfiguraatiot

Kuten jo aiemmin todettiin, kytkinkonfiguraatiot molempiin Quality of Service -arkkitehtuureihin toteutettiin kahdella kytkinmallilla, Catalyst 2960 ja Catalyst ME 3400. Näistä ensimmäinen on puhtaasti siirtoyhteyserroksen kytkin joka ei osallistu IP-tason reititykseen. ME 3400-sarjan kytkin puolestaan on sekä siirtoyhteyserroksen, että verkkokerroksen operaatioihin valmiudet omaava laite. Arkkitehtuurijaossa kaksi Catalyst 2960 -kytkintä sijoitettiin Differentiated Services -puolelle (Kts. KUVIO 17) työnimillä ”LENINGRAD” ja ”KURSK”. Kytkinten

konfiguraatiot on esitetty liitteissä 6 ja 7 (LIITE 6, LIITE 7). Vastaavasti Integrated Services -puolelle (Kts. KUVIO 19) asettiin verkon reunoille kaksi ME 3400-kytkintä turvaamaan Quality of Service RSVP-verkon vikatilanteissa. Kytkinten konfiguraatiot on esitetty liitteissä 8 ja 9 (LIITE 8, LIITE 9) työnimillä ”PEARL HARBOR” ja ”EL ALAMEIN”. ME 3400 -kytkimet toteuttivat käytännössä samaa palvelutasoa kuin RSVP- reitittimet, mutta toteutusmekaniikka nojasi DSCP-operaatioihin. Tämä siksi, että testiverkon kytkimet eivät tukeneet RSVP-protokollaa. Syy tähän on se, että kytkimet harvoin tarjoavat oletusyhdykskäytävän verkon laitteille ja juuri oletusyhdykskäytäväliitynnän omaava laite aloittaa tai päättää RSVP -ketjun. Suurin syy ME 3400 -kytkinten valintaan ja testiverkkoon sijoittamiseen oli tutkia ja vertailla niiden QoS-ominaisuuksia Catalyst 2960-kytkimiin.

Kuten luvussa 4.5 todettiin, ainoa keino toteuttaa Quality of Service kytkintasolla on Auto-QoS ja sen operaatiot. Sinällään Auto-QoS ei tue testiverkossa ajettua striimausta, koska se ei tunnista videoliikennettä muusta liikenteestä (kuten esimerkiksi FTP-liikenteestä). Näin ollen Ciscon määrittämää ”Real-Time Video Traffic” -liikennetyyppiä DSCP-arvolla 34 ei voida hyödyntää valmiina asetuksena. Käytännössä ainoaksi keinoksi jää Auto-QoS-palvelun manipulointi niin, että videoliikenne ottaa VoIP-liikenteen paikan priorisointitauluissa. Tämän jälkeen palvelunlaatua voidaan edelleen parantaa muuttamalla jonojen ja puskurien arvoja. Tällöin saadaan videoliikenteelle varattua lähes maksimaalinen kaistavaraus. (Cisco Systems Incorporated 2008g.)

Seuraavat konfiguraatorivit asettavat testiliikenteen (videostriimaus) kaiken muun liikenteen edelle VoIP-liikenteen sijaan. Konfiguraatorivit on poimittu LENINGRAD-kytkimestä (LIITE 6):

```
mls qos map policed-dscp 24 26 34 46 to 0
mls qos map cos-dscp 0 4 8 16 24 32 46 48
mls qos srr-queue input bandwidth 90 10
```

Ensimmäinen rivi vapauttaa valitut DSCP-arvot muokattaviksi policy-map-parametreilla (huomaa arvo 34, ”Real-Time Video Traffic”). Seuraava rivi listaa käytettävät CoS-arvot (huomaa arvo 4, ”Real-Time Video Traffic”). Oletusarvoisesti CoS-arvot ovat aivan jotain muuta, kuin mitä yleisesti käytetään (0-7). Tämä johtuu siitä, että jos arvo menee yli 7, DSCP-arvokin käy, jolloin se muutetaan vastaavaksi CoS-arvoksi. Konfiguroinnissa kannattaa kuitenkin käyttää suoraan CoS-arvoa selkeyden vuoksi. Kolmas rivi määrittää prosentuaaliset kaistanleveydet ingress-jonoille (2 kpl).

```

mls qos

class-map match-all Video
  match access-group 101
class-map match-all premium
  match ip dscp af41
!
!
policy-map RTSP
  class premium
    set dscp af41
policy-map SetDSCP
  class Video
    set dscp af41

access-list 101 permit udp any any eq 1234

```

Parametri ”mls-qos” aktivoi Auto-QoS-palvelun. ”Class-map”- ja ”policy-map”-parametrit toimivat samalla logiikalla kuin reitittimen DSCP-toteutus. ”Class-map match-all”-komennon tarkoituksena on ohjata kaikki siihen kuuluvat luokat (class) tunnistautumaan reitittimen pääsilystoja (access-list) hyväksi käyttäen. Tässä tapauksessa viitataan access-listaan 101. Access-listat on asetettu tietyin parametrein tunnistamaan videoliikenne muusta liikenteestä. Tässä tapauksessa havaitaan siis UDP-portin 1234 liikenne, jota videostriimaus käyttää kohdeportinaan. ”Policy-map”- komento sisältää puolestaan tiedon mitä luokan sisältävälle liikenteelle tehdään, kun se on tunnistettu. Konfiguraatioesimerkissä asetetaan luokan ”Video” DSCP-arvo. Tämä arvo on ”af41” (DSCP 34) joka viittaa liikennetyyppiin ”Real-Time Video Traffic”. ”Policy-map” parametrit asetetaan kytkimen sisääntuloportteihin. ”Policy-map” parametrit eivät kuitenkaan määritä kais-

tanleveyttä kuten reitittimen esimerkissä nähtiin, vaan ne asettavat liikenteelle DSCP-arvot joita mls-parametrit ohjaavat edelleen jonoihin. Jonoissa puolestaan on määrittämät kaistavarauksista:

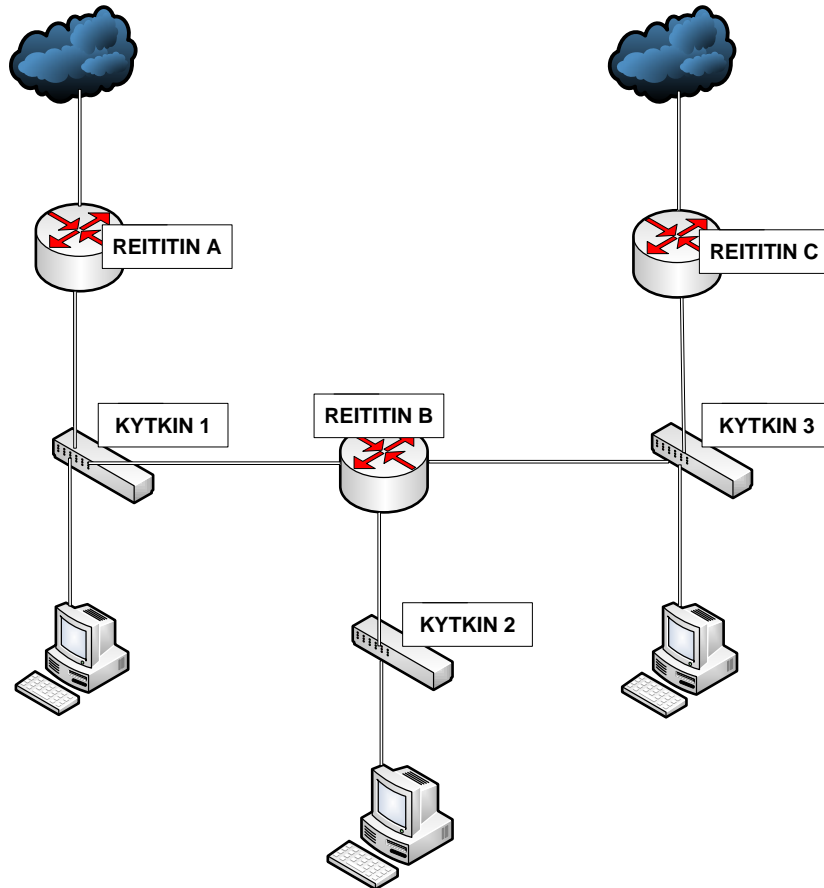
```
interface FastEthernet0/20
  srr-queue bandwidth share 90 2 6 2
  mls qos trust cos
  service-policy input SetDSCP

interface FastEthernet0/24
  srr-queue bandwidth share 90 2 6 2
  priority-queue out
  mls qos trust cos
```

Liityntäkohtaiset asetukset määrittävät jonokohtaiset, prosentuaaliset kaistanvaraukset. 1. jono saa 90%, 2. jono 2% 3. jono 6% jne. ”mls qos trust cos” -komento määrittää että saapuvan liikenteen CoS-arvoja käytetään QoS-operaatioissa. Näin ollen luotetaan edellämääriteltyn CoS-to-DSCP-suhteeseen. Tämä on ensiarvoisen tärkeää, koska kytkimen QoS-parametreissa käytetään sekä CoS- että DSCP-arvoja. Sisäänmenoliityntään määritetään ”service-policy”, tässä tapauksessa ”SetDSCP”. ”Priority-queue out” komento aktivoi 1. egress-jonon pääjonoksi. Tämä tarkoittaa sitä, että pääjonon kaistavaatimukset täytetään aina. Jos liikenne pääjonossa laskee, SRR-jonot voivat käyttää pääjonosta vapautuvaa kaistaa.

QoS-toteutuksen kannalta kytkimen välttämätön ominaisuus on pääsyylojen (access-list) tukeminen. Aivan kuten 2800-sarjan reitittimet, kytkimet voidaan konfiguroida tunnistamaan videoliikenne pääsyylojen avulla. Pääsyylistat keräävät ”class-map match-all Video” -luokkaan video-liikenteen, jolle asetetaan ”policy-map SetDSCP” -komennolla DSCP-arvo 34 (af41). Kun liikenne on tunnistettu CoS-arvoon 4 ja DSCP-arvoon 34, voidaan hyödyntää Auto-QoS-tauluja (KUVIO 11, KUVIO 12, KUVIO 13) Testikonfiguraatio toteutettiin käytämällä VoIP-liikenteen oletusasetuksia sekä liikenteen sisään-, että ulostulojonoissa. Lisäksi jonojen kaistanvarauksia muutettiin paremmiksi. Tässä tapauksessa kaikki muu liikenne, myös VoIP, jäi tunnistamattomaksi ”best-effort”-liikenteeksi. (Cisco Systems Incorporated 2008g.)

Koska kytkinten kannalta CoS-to-DSCP- ja Auto-QoS-operaatiot ovat hankalasti ymmärrettäviä, on tilannetta syytä selkeyttää kuvion 18 avulla (KUVIO 18).



KUVIO 18. Auto-QoS -esimerkkiverkko.

Kuviossa 18 on esitetty kuvitteellinen tilanne, jossa kytkimet 1, 2 ja 3 toteuttavat Differentiated Servicen Auto-QoS-operaatioita. Reitittimet A, B ja C toteuttavat DiffServ-palvelua DSCP-prosesseilla (reitittimet A ja C ovat yhteydessä ulko-verkkoon). Esimerkissä tietokoneet lähettävät sekä toisilleen että ulko-verkkoon DSCP-merkittyjä paketteja. Koska matkalla liikennettä käsitellään sekä kytkimis-
sä kehystasolla (CoS-prosesseilla), että reitittimissä pakettitasolla (DSCP-
prosesseilla), täytyy aktiivilaitteiden osalta toteuttaa QoS-operaatioita samalla ta-
valla OSI-kerroserosta huolimatta. Käytännössä reititin toteuttaa vain DSCP-

merkintää, jota kytkin ei QoS-prosesseissaan käytä. CoS-to-DSCP merkintä toimii kytkinten Auto-QoS-operaatioissa molempiin suuntiin (CoS-to-DSCP, DSCP-to-CoS). Reitittimeltä priorisointitieto tulee aina DSCP-kentässä. Käytännössä siirtoyhteyskerroksen kytkimet pystyvät tunnistamaan paketin mukana tulleen DSCP arvon, mutta eivät voi sitä suoraan priorisointijonojen suhteen hyödyntää. Tämän vuoksi kytkimen on aina muodostettava liikenteen yksilöivä CoS-arvo. (Cisco Systems Incorporated 2008g.)

DSCP- ja sitä vastaava CoS-arvo sisältävät samat priorisointitiedot, koska ne viittaavat samaan liikennetyypiin. Kun paketti saapuu reitittimeltä kytkimelle, on kytkimen käytettävä hyväksi konfiguraatioissa ennaltamääriteltyä DSCP-to-CoS-määrittystä. Näin kytkin saa tiedon liikennetyypin CoS-arvosta, jolla varsinainen kytkintason priorisointi suoritetaan. Kun kehystasolla on tieto CoS-arvosta, säilyy tieto oletusarvoisesti Ethernet-kehyksessä, ellei mikään laite sitä uudelleen aseta. DSCP-kenttä puolestaan säilyy IP-kehyksessä arvoiltaan samana verkon solmupisteestä solmupisteeseen. Jos liikennepriorisointi aloitetaan kytkimestä (asetetaan CoS-arvo) ja siirretään reititinverkkoon, on kytkimen tehtävä CoS-to-DSCP-muunnos, koska reitittimet eivät CoS-arvoja ymmärrä. Auto-QoS-operaatioissa kytkimet ovatkin keskeisessä roolissa, sillä niiden täytyy paitsi muuntaa DSCP-arvoista CoS-arvoja, myös asettaa ja yksilöidä DSCP-arvoja reitittimien tarpeisiin. DSCP on taaksepäin yhteensopiva ToS-kentän kanssa, mutta kytkinconfiguraatioiden kannalta on tärkeää että priorisointitieto kulkee nimenomaan DSCP-muodossa. (Cisco Systems Incorporated 2008g.)

Cisco ME 3400 -kytkimen konfiguraatiot perustuivat täysin luvussa 5.3.1 esitettyyn DSCP-tekniikkaan luottaen pääsilystoihin perustuvaan liikenteentunnistukseen ja DSCP-arvojen asettamiseen. Vaikka kytkimet toimivat sekä siirtoyhteyskerroksella että verkkokerroksella, ne toteuttivat täydellisesti DSCP-operaatioita aivan kuten reitittimet. Suurin ero reitittimiin verrattuna olivat liitynnät ja niiden varaukset. Kytkimen porteissa käytettiin 100Mbit/s yhteyttä sekä verkkoon että testikoneen suuntaan kaistanvarauksella 1,152Mbit/s. Kytkin ei ollut suorassa yhteydessä mihinkään verkon pullonkaulaan, joten varaus on kytkimen sijoittelun näkökulmasta vähintäänkin riittämätön. Kytkintesti todistikin, että solmupistevaraus kantaa koko verkon läpi, joten varaus voidaan tehdä pienimmän kaistakapasite-

teetin omaavan linkin mukaan. Suurempi varaus on toki myös mahdollinen, mutta silloin sarjalinkkien fyysiset siirto-ominaisuudet estävät paremman palvelulaadun saavuttamisen. (Cisco Systems Incorporated 2008g.)

Esimerkkinä ME 3400 -kytkimen konfiguraatiosta esitetään testiverkon PEARL HARBOR -kytkimen pääkonfiguraatiot (LIITE 8):

```
class-map match-all Video
  match access-group 101
class-map match-all premium
  match ip dscp ef
!
!
policy-map RTSP
  class premium
    bandwidth 1152
policy-map SetDSCP
  class Video
    set ip dscp ef

interface FastEthernet0/20
  bandwidth 100000
  service-policy input SetDSCP
  service-policy output RTSP

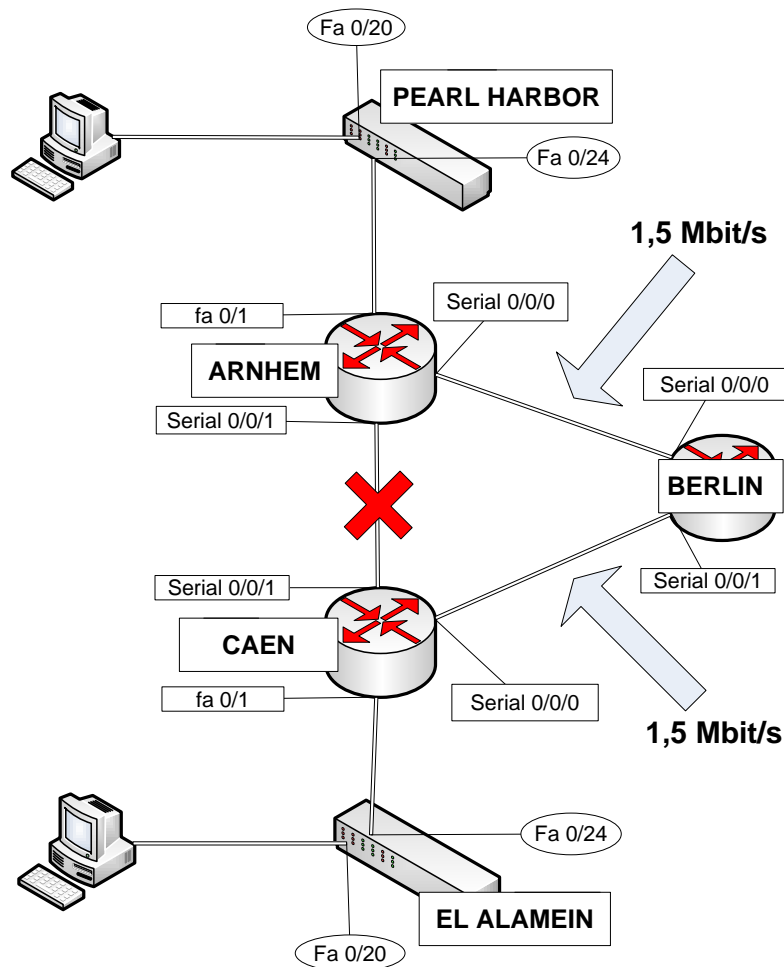
interface FastEthernet0/24
  port-type nni
  bandwidth 100000
  service-policy input SetDSCP
  service-policy output RTSP

access-list 101 permit udp any any eq 1234
```

ME 3400 -kytkin toimii QoS-operaatioissa täysin samalla tavalla kuin 2081-reititin. Verkkokerrostason reitittävä kytkin onkin huomattavasti helpommin konfiguroitavissa DiffServ-operaatioihin kuin Catalyst 2960 -kytkin, joka toimii ainoastaan siirtoyhteyskerroksella (OSI-2).

5.4 Integrated Services -toteutus

Integrated Services -toteutus nojaa aikaisemmin esiteltyyn RSVP-protokollaan ja sen loogisiin operaatioihin. Reitittimet ”BERLIN”, ”ARNHEM” ja ”CAEN” osallistuvat RSVP-operaatioihin. Reitittimien konfiguraatiot on esitetty liitteissä 3, 4, ja 5 (LIITE 3, LIITE 4 ja LIITE 5). Liitteessä 3 esitelty ”BERLIN”-reitittimen konfiguraatio osallistuu sekä Differentiated Services että Integrated Services -palvelutoteutukseen. On huomattava, että RSVP:n Integrated Services -toteutukseen eivät osallistu testikoneiden ja reitittimien väliset kytkimet, vaan RSVP-polku päättyy oletusyhdyntäväliiityntöihin fa0/1-liitynnän omaavissa reitittimissä ”ARNHEM” ja ”CAEN”. Toisin kuin edellä esitettyssä DSCP-toteutuksessa, jokaisen RSVP-protokollaa tukevan reitittimen on signaloitava viestejä verkon läpi. Testiolosuhteissa reitittimien ”ARNHEM” ja ”CAEN” välinen sarjalinkki suljettiin, jotta mahdollisimman monta reititintä saatiin mukaan signalointi-ketjuun. On huomattavaa, että RSVP ei vaikuta testiverkon oikealla puolella, koska Differentiated Serviceä toteuttavat reitittimet eivät sitä tue. Opinnäytetyössä käytetty Integrated Services -testiverkko on esitetty kuviossa 19 (KUVIO 19).



KUVIO 19. Integrated Services -testiverkko.

Lähtökohtana RSVP-signalointiin ja verkonvaraukseen osallistuvissa reitittimissä on oltava liityntäkohtainen RSVP-kaistanvaraus. Viimeinen varausliityntä on siis oletusyhdykäytäviliityntä verkon laidoilla. Konfiguraatioista voidaan huomata että kaistanvaraus on 1150Bit/s, konfiguraatiokomentona: ”ip rsvp bandwidth 1150”. On huomattava, että varaus käsittää maksimissaan 75% kokonaiskaistasta, mikä on Ciscon asettama rajoite. Osittain tätä perustellaan signaloitiliikenteelle varatulla 25% kaistalla, joka pienemmissä verkoissa on turhankin suuri. Liityntäkohtaiset RSVP-varaukset voidaan suorittaa seuraavalla komennolla:

```
BERLIN(config-if)# ip RSVP bandwidth 1150
```

“Show running-config” -komennolla voidaan helposti käydä läpi, missä liittynöissä RSVP-varaus vaikuttaa. S0/0/0-liittynnällä on siis tässä esimerkissä kokonaisvaraus 1150Kbit/s kaikelle RSVP-liikenteelle:

```
interface Serial0/0/0
 ip address 192.168.100.1 255.255.255.0
 no fair-queue
 clock rate 2000000
 ip rsvp bandwidth 1150
```

Kokonaisvarausta ei voida ylittää, joten useaa liikennetyyppiä priorisoitaessa kokonaiskaistavaraus pitää jakaa näiden liikennetyyppien kesken. Jos kaikissa RSVP:tä tukevissa reitittimissä on varausparametri, kulkee liikenne lyhintä mahdollista reittiä. Tästä syystä onkin kompleksisimmissä verkkokokoonpanoissa syytä hyödyntää reitittimen ”show”-komentoparametreja RSVP-polkuvarausten verifiointiseksi. Seuraavassa listassa on esitetty tärkeimmät näistä. Komennot viittaavat CAEN-reitittimen konfiguraatioon (LIITE 5):

CAEN# show ip rsvp interface

interface	allocated	i/f max	flow max	sub max
Fa0/1	1150K	1150K	1150K	0
Se0/0/0	1150K	1150K	1150K	0
Se0/0/1	0	1150K	1150K	0

”Show ip rsvp interface” -komento listaa liittymäkohtaiset RSVP-varaustiedot. Komento on tehokas tapa tarkistaa reititinkohtaisesti, mitkä linkit tukevat RSVP-varausta. On huomioitava, että kyseiset liittytiedot liittyvät reservation (varauspolku)-komentomäärittelyyn. Sender (lähetyspolku)-tietoja komento ei kerro. (Cisco Systems Incorporated 2009b.)

Tärkeimmät RSVP-konfiguraatioparametrit liittyvät polkuvarauksiin lähettäjältä, (sender) ja verkonvarauksiin (reservation). Riippumatta siitä missä kohtaa verkkoa RSVP-reitittimet ovat, osoittavat komentojen kohde- ja lähdeosoitteet aina verkon päätelaitteisiin, useimmiten liikennettä generoiviin tietokoneisiin. Komentojen pituuden ja monimutkaisuuden vuoksi niiden parametrikohmainen tarkempi esittely toteutetaan seuraavassa kappaleessa. Cisco on rajannut sender- ja reservation-komentojen asettamisen ainoastaan reunareitittimiin suurten verkkokonfiguraatitoteutusten helpottamiseksi. Verkon muut reitittimet saavat tarvittavat RSVP-tiedot reunareitittimiltä ja lähettävät ne edelleen naapurireitittimiinsä. Sender-tiedot kulkevat kaikkien reitittimien läpi mutta reservation-komennot eivät näy muualla kuin reunareitittimissä. Tämä johtuu siitä että ainoastaan reunareitittimet suorittavat koko verkon läpi menevän RSVP-varauksen. Tämä varaus on kuitenkin riippuvainen jokaisesta verkon reitittimien liityntäkohtaisesta varauksesta. (Cisco Systems Incorporated 2009b; Cisco Systems Incorporated 2009c.)

Seuraavaksi perehdytään sender- ja reservation-komentojen parametrikohlaiseen tulkintaan. Ensimmäiseksi on esitetty koko komento ja sen alapuolella tulkinta. Esimerkkikonfiguraatorivit löytyvät CAEN-reitittimen konfiguraatiosta (LIITE 5):

```
ip rsvp sender 192.168.44.2 192.168.55.2 UDP 1234 1234
192.168.55.2 FastEthernet0/1 1150 1150
```

```
ip rsvp sender [kohdeosoite] [lähdeosoite] [protokolla] [lähdeportti] [kohdeportti]
[viestin saapumisosoite] [viestin saapumisliityntä reitittimessä] [kaistanvaraus
(bit/s)] [maksimilähetyspurske (bit/s)]
```

```
ip rsvp reservation 192.168.44.2 192.168.55.2 UDP 1234
1234 192.168.55.2 FastEthernet0/1 FF RATE 1150 1150
```

ip rsvp reservation [kohdeosoite] [lähdeosoite] [protokolla] [lähdeportti] [kohdeportti] [varauksen kohdeosoite] [varauksen lähetysliityntä reitittimessä] [varaus-tyyppi (yksittäisvaraus {FF}, jaettu ja rajattu varaus {SE}, jaettu ja rajaamaton varaus {WF})] [kaistanvaraus (bit/s)] [maksimilähetyksurke (bit/s)]

Huomattavaa on, että kohde- ja lähdeosoitteet ovat täysin samat molemmissa kommennoissa. Lähde- ja kohdeosoitteet ovat siis varsinaisen datan lähetys- ja vastaanotto-osoitteet, eivät RSVP-polkuviestien ja kaistanvarausten osoitteet, jotka ovat vastakkaiset suhteessa toisiinsa. Yksittäisvaraus "FF" tarkoittaa, että varaus tehdään vain yhden lähettäjän käyttöön. Jaettu "SE"- ja "WF"-varaus taas jakavat varauksen useamman lähettäjän hyödynnettäväksi. Lähettäjien määrä voidaan myös rajata tai pitää rajaamattomana. (Cisco Systems Incorporated 2009b; Cisco Systems Incorporated 2009c.)

Integrated Services -testiverkossa käytettiin seuraavia sender- ja reservation-konfiguraatioparametreja toimivan RSVP-polun luomiseksi IntServ-verkon päästä päähän. ARNHEM-reitittimen (LIITE 4) konfiguraatio:

```
ip rsvp sender 192.168.44.2 192.168.55.2 UDP 1234 1234
192.168.100.254 Serial0/0/0 1150 1150
```

```
ip rsvp sender 192.168.55.2 192.168.44.2 UDP 1234 1234
192.168.44.2 FastEthernet0/1 1150 1150
```

```
ip rsvp reservation 192.168.44.2 192.168.55.2 UDP 1234
1234 192.168.100.254 Serial0/0/0 FF RATE 1150 1150
```

```
ip rsvp reservation 192.168.55.2 192.168.44.2 UDP 1234
1234 192.168.44.2 FastEthernet0/1 FF RATE 1150 1150
```

Seuraavksi verkon vastapäisellä reunalla sijaitsevan CAEN-reitittimen (LIITE 5)

sender- ja reservation-parametrit:

```
ip rsvp sender 192.168.44.2 192.168.55.2 UDP 1234 1234
192.168.55.2 FastEthernet0/1 1150 1150
```

```
ip rsvp sender 192.168.55.2 192.168.44.2 UDP 1234 1234
192.168.110.254 Serial0/0/0 1150 1150
```

```
ip rsvp reservation 192.168.44.2 192.168.55.2 UDP 1234
1234 192.168.55.2 FastEthernet0/1 FF RATE 1150 1150
```

```
ip rsvp reservation 192.168.55.2 192.168.44.2 UDP 1234
1234 192.168.110.254 Serial0/0/0 FF RATE 1150 1150
```

BERLIN-reititin ei sender tai reservation-komentoja tarvitse, koska se on keskusreititin eikä omaa oletusyhdykäytäväliityntöjä. Suurissa verkoissa tämä ominaisuus säästää huomattavasti konfigurointityötä, koska verkon keskellä sijaitsevat reitittimet vaativat ainoastaan liityntäkohtaiset varaukset, ” ip rsvp bandwidth”. RSVP-polkujen toimivuus on kuitenkin syytä tarkistaa show-komennoilla. Seuraavassa ARNHEM-reitittimen (LIITE 4) hyödyllisimmät show-komennot:

```
-----
ARNHEM# show ip rsvp installed
```

```
RSVP: FastEthernet0/1
BPS   To           From           Protoc  DPort  Sport
1150K 192.168.55.2   192.168.44.2   UDP     1234   1234
```

```
RSVP: Serial0/0/0
BPS   To           From           Protoc  DPort  Sport
1150K 192.168.44.2   192.168.55.2   UDP     1234   1234
```

```
RSVP: Serial0/0/1 has no installed reservations
```

```
-----
ARNHEM# show ip rsvp sender
```

```
To           From           Pro  DPort  Sport  Prev Hop           I/F     BPS
192.168.44.2 192.168.55.2  UDP 1234 1234 192.168.100.254   Se0/0/0 1150K
192.168.55.2 192.168.44.2  UDP 1234 1234 192.168.44.2     Fa0/1   1150K
```

```
-----
```



```
-----
ARNHEM# show ip rsvp reservation
```

To	From	Pro	DPort	Sport	Next Hop	I/F	Fi	Serv	BPS
192.168.44.2	192.168.55.2	UDP	1234	1234	192.168.100.254	Se0/0/0	FF	RATE	1150K
192.168.55.2	192.168.44.2	UDP	1234	1234	192.168.44.2	Fa0/1	FF	RATE	1150K

```
-----
```

Ensimmäinen osio kertoo onko liittymälle konfiguroitu RSVP-varauksia (reservation). Toinen osio listaa RSVP-polkuvaraukset (liikenteen) lähettäjiltä. Viimeinen osio listaa verkonvarauspyynnöt (liikenteen) vastaanottajilta. Kuten aikaisemmin RSVP-protokollan esittelyn yhteydessä todettiin, liikenteen vastaanottaja suorittaa kaistanvarauspyynnöt (reservation) saatuaan lähettäjältä polkuvarausviestin (sender).

Seuraavaksi CAEN-reitittimen (LIITE 5) show-komennot:

```
-----
CAEN# show ip rsvp installed
```

```
RSVP: FastEthernet0/1
BPS   To           From           Protoc  DPort  Sport
1150K 192.168.44.2    192.168.55.2  UDP    1234   1234

RSVP: Serial0/0/0
BPS   To           From           Protoc  DPort  Sport
1150K 192.168.55.2    192.168.44.2  UDP    1234   1234
```

```
RSVP: Serial0/0/1 has no installed reservations
```

```
-----
CAEN# show ip rsvp sender
```

To	From	Pro	DPort	Sport	Prev Hop	I/F	BPS
192.168.44.2	192.168.55.2	UDP	1234	1234	192.168.55.2	Fa0/1	1150K
192.168.55.2	192.168.44.2	UDP	1234	1234	192.168.110.254	Se0/0/0	1150K

```
-----
```

```
CAEN# show ip rsvp reservation
```

To	From	Pro	DPort	Sport	Next Hop	I/F	Fi	Serv	BPS
192.168.44.2	192.168.55.2	UDP	1234	1234	192.168.55.2	Fa0/1	FF	RATE	1150K
192.168.55.2	192.168.44.2	UDP	1234	1234	192.168.110.254	Se0/0/0	FF	RATE	1150K

```
-----
```

Lopuksi BERLIN-keskusreitittimen (LIITE 3) show-komennot:

```
BERLIN# show ip rsvp installed
```

```
RSVP: Serial0/0/0 has no installed reservations
RSVP: Serial0/0/1 has no installed reservations
RSVP: Serial0/1/0 has no installed reservations
RSVP: Serial0/1/1 has no installed reservations
```

```
BERLIN# show ip rsvp sender
```

To	From	Pro	DPort	Sport	Prev Hop	I/F	BPS
192.168.44.2	192.168.55.2	UDP	1234	1234	192.168.110.1	Se0/0/1	1150K
192.168.55.2	192.168.44.2	UDP	1234	1234	192.168.100.1	Se0/0/0	1150K

```
BERLIN# show ip rsvp reservation
```

To	From	Pro	DPort	Sport	Next Hop	I/F	Fi	Serv	BPS
----	------	-----	-------	-------	----------	-----	----	------	-----

Kuten komentojen tulosteista nähdään, BERLIN-reititin ei tee RSVP-varauksia. Sen sijaan BERLIN saa sender-polkutiedot naapurireitittimiltään. Tämän lisäksi kannattaa vielä tarkistaa BERLIN-reitittimestä ”show ip rsvp interface”-komennolla liityntöjen tila, sillä BERLIN on RSVP-polkutietojen osalta täysin muiden reitittimien varassa. Testimielessä liityntöihin Se0/1/0 ja Se0/1/1 on asetettu varausparametrit, vaikka niiden takana ei enää ole RSVP-tukea. Show-komennosta nähdäänkin, että ne eivät osallistu RSVP-operaatioihin:

```
BERLIN# show ip rsvp interface
```

interface	allocated	i/f max	flow max	sub max
Se0/0/0	1150K	1150K	1150K	0
Se0/0/1	1150K	1150K	1150K	0
Se0/1/0	0	1150K	1150K	0
Se0/1/1	0	1150K	1150K	0

Sender- ja reservation-komennoissa esiintyvät itse liikenteen kaistanvaraus sekä lähde- ja kohdeosoitteet. RSVP-varaukset ovat ikäänkuin polkuja reititinverkoissa, joten pelkästään IP-osoitetiedot ja porttimääritykset liikenteelle eivät määrää yksiselitteisesti RSVP-varausreittiä. Kuten jo aiemmin todettiin, RSVP käyttää lyhintä mahdollista reittiä, joka ei kuitenkaan aina tapauskohtaisesti ole paras mahdollinen. Tällöin on kiinnitettävä huomiota erityisesti liityntäkohtaisiin varauksiin. Liitteissä 3, 4 ja 5 (LIITE 3, LIITE 4 ja LIITE 5) esitetyt reitittimien sender- ja reservation-parametrit kattavat liikennevarauksen molempiin suuntiin. Tätä huomataankin että yhteen suuntaan varattava liikenne vaatii aina sekä sender-, että reservation- parametrit. Sekä sender- ja reservation- parametrit konfiguroitiin reitittimiin ”ARNHEM” ja ”CAEN”, kun taas reitittimeen ”BERLIN” asetettiin vain liityntäkohtaiset RSVP-varaukset (IP RSVP bandwidth 1150). Tämä siksi että ”ARNHEM” ja ”CAEN” ovat oletusyhdykäytäväreitittimiä ja ”BERLIN” on verkon sisäinen reititin. Komentojen lopussa olevat hyppyparametrit, ja niiden suunta ovat myös huomattavan tärkeitä konfiguraation toimivuuden kannalta. Sender-viesteissä osoitetaan IP-osoitteeseen, josta viesti tuli (previous-hop-IP-address), jotta kukin reititin tietää mistä osoitteesta polkuviesti on lähtöisin. Analogiassa tähän reservation-viesteissä viitataan seuraavan hypyn takana olevaan osoitteeseen (next-hop-IP-address), joka on sama kuin sender-viestissä viitattu osoite. Tämä johtuu siitä, että reservation-viesti menee vastakkaiseen suuntaa täysin samaa varauspolkua pitkin. (Cisco Systems Incorporated 2009b; Cisco Systems Incorporated 2009c.)

5.5 Toteutusten vertailu

Differentiated Services -toteutus nojasi pitkälle DSCP- ja CoS-toteutusmekanismien käyttöön. Konfiguraatioiden osalta toteutus oli yksinkertaisempi kuin Integrated Servicessä, sillä palvelu voitiin toteuttaa laitekohtaisesti. Samalla voitiin myös helposti todeta palvelun toimivuus ja toimimattomuus, sillä yhden laitteen käsittely palvelun toteuttajana on helpompaa kuin kokonaisen verkkokonaisuuden. Yllättävää oli huomata, kuinka samanlaisia DSCP-konfiguraatiot olivat laitteiden välillä. Sekä 2081-reititin että ME 3400 -kytkin toteuttivat täysin samaa mekaniikkaa DiffServin osalta.

Liikennetyyppien tunnistus pääsilystoilla osoittautui DiffServ-toteutuksissa toimivaksi ratkaisuksi. Laitteet eivät oletusasetuksilla kyenneet tunnistamaan liikennettä, eikä niistä löytynyt minkäänlaisia automaattisia tunnistusmekanismeja QoS-palvelun käyttöön. IntServin osalta tämä on suurempi puute, sillä palvelu nojaa pitkälti moninaisen sovellustason liikenteen käsittelyyn ja liikenteentunnistussisällytetään suoraan monimutkaisiin RSVP-komentoihin. Pääsilystojen hyödyntäminen liikenteen tunnistamisessa lisää mekaanista työtä DiffServ-konfiguroinnin osalta, mutta lopputuloksena saadaan hyvinkin tarkasti määriteltyjä liikennetyyppejä DSCP-operaatioiden käyttöön. Sama pätee myös RSVP-komentoihin, mutta mekaanisen työn osuus on huomattavasti suurempi vaikeuttaen näin myös konfiguraatioiden hallintaa. DiffServin DSCP-operaatioiden suurimmat edut olivat nopea reagointi palvelutoteutuksissa sekä palvelutason säilyttäminen koko verkon läpi kriittisten liikennepisteiden kautta. Siirtoyhteyskerrostason Catalyst 2960-kytkin osoittautui toteutusten kannalta huonoksi valinnaksi. Konfiguraatioiden vaikeus ja valjastaminen eri liikennetyyppien käyttöön tekee kytkimestä hankalasti käsiteltävän DiffServ-operaatioihin. Vaikka lopulta toimivat konfiguraatiot saatiin aikaiseksi, Catalyst 2960 -kytkintä tuskin voi suositella QoS-palvelua toteuttavaksi laitteeksi. Suositeltavampi valinta on ME 3400 -kytkin.

Tulokset Integrated Servicen RSVP-protokollatoteutuksen suhteen olivat heikommat verrattuna Differentiated Serviceen. RSVP:n täyttä potentiaalia suuren verkon palvelutoteuttajana ei pystytty todistamaan pienen testiverkon vuoksi, mutta itse QoS-palvelutason tehokkuus voitiin todeta. Tutkimustulosten mukaan

RSVP tekee verkkovarauksen yllättävän hitaasti pienessäkin verkossa. Täydellinen videostriimauslaatu saavutettiin vasta n. 60-90 sekunnin kuluessa kolmella reitittimellä. Sama tulos saatiin myös kahdella reitittimellä. Protokollan hitaus tuskin vaikuttaa, jos verkkoa käytetään esimerkiksi 1,5 tunnin mittaisen elokuvan katsomiseen, sillä hitaan palvelun aktivoimisen jälkeen kuva on laadullisesti hyvä, niin kauan kuin yhteyttä ei suljeta.

RSVP:n konfigurointi verkon reunareitittämiin säästää paljon aikaa, mutta itse konfiguraatiot ovat haastavia. RSVP:n konfigurointiin kannattaakin tutustua huolella ennenkuin sitä lähdetään verkkoon toteuttamaan. Vaikka pieni testiverkko rajoittikin tuloksien tehokasta analysointia, RSVP:n edut huomattiin jo konfiguraatiovaiheessa. Reunareitittimien takana olevat reitittimet ovat erittäin helposti hallittavia QoS-palvelun osalta. Aiemmin mainitun skaalautumisen puuttuminen huomattiin ongelmalliseksi siinä tapauksessa, että verkon reunalle lisätään reitittämiä. Tällöin verkon QoS-konfiguraatiot täytyy tehdä uudelleen. Mutta jos verkon sisälle lisätään laitteistoa, ongelma ei ole kovinkaan suuri. Tällöin verkkoon tulee vain uusia signaalintisolmuja, joiden konfigurointi QoS-palvelun osalta on vaivatonta.

6 YHTEENVETO

Opinnäytetyön tavoitteena oli tutkia Quality of Service -palvelua kahden eri arkkitehtuurin näkökulmasta. Nämä arkkitehtuurit olivat Differentiated Services ja Integrated Services. Työssä perehdyttiin QoS-tekniikoiden tarjoamiin suunnittelu- ja toteutusratkaisuihin nykyajan tietoliikenneverkoissa. Differentiated Services huomattiin tehokkaaksi toteutusarkkitehtuuriksi muuttuviin tai kasvaviin verkkoihin. Differentiated Servicen DSCP-toteutus on tehokas vain verkon solmupisteissä, joten sen tehokas verkkosijoittelu on ensiarvoisen tärkeää. Arkkitehtuuri on hyvin skaalautuva koko verkon tarpeisiin mutta usein sen toteutus on karkea. Jokaiselle verkon osalle ei pystytä tekemään yksityiskohtaisesti määriteltyjä palvelutasoja, varsinkin jos palvelua toteuttavia laitteita on niukasti. Usein Differentiated Services soveltuu parhaiten pienille, moninaista liikennettä käsitteleville verkoille, joissa palvelun vaatima tehokas ylläpito on mahdollista toteuttaa.

Integrated Services on QoS-arkkitehtuureista hienojakoisempi, sillä sen pääprotokolla RSVP toteuttaa tehokasta signalointia laitteiden välillä. Tämän ansiosta eri palvelutasojen ja liikennetyyppien määrää voidaan verkossa paitsi helposti kasvat-
taa myös tehokkaasti hallita. Laitekonfiguraatiot ovat helposti toteutettavissa suurempiinkin verkkoihin juuri laitteiden välisen signaloinnin ansiosta. Toisaalta Integrated Services ei mukaudu muuttuviin verkkotopologioihin kovinkaan hyvin, sillä jokaisen palvelua toteuttavan laitteen on oltava tietoinen siitä. Skaalautuvuuden ongelmiin voidaan ylläpidollisesti vaikuttaa hyvällä verkkosuunnittelulla ja keskitetyllä laitehallinnalla. Integrated Services soveltuu hyvin topologioiltaan monimutkaisiin verkkoihin, joissa reititys kiertää useiden solmujen kautta. Arkkitehtuuri soveltuu hyvin reititin- ja runkoverkkoihin, mutta lähi- ja kytkinverkoissa sen tehokkuus heikkenee merkittävästi. Tämä johtuu RSVP:n huonosta tuesta sekä päätelaitteissa että kytkimissä.

Työn tärkeimpänä tuloksena voidaan pitää onnistunutta testiverkkototeutusta sekä Differentiated Servicen että Integrated Servicen osalta. Differentiated Servicen DSCP- ja CoS-toteutus sekä reitittimisessä että kytkimissä onnistui yli odotusten. Palvelun tehokkuus videostriimaustesteissä osoittautuikin Integrated Serviceä te-

hokkaammaksi ja varmemmaksi. Koko testiverkon osalta voitiin todeta myös arkkitehtuurien hyvä yhteensopivuus. Arkkitehtuurit toteuttivat omaa palvelutasoaan omilla verkkopuoliskoillaan eivätkä vaikuttaneet toisiinsa. Arkkitehtuurien risteyskohdassa toiminut keskusreititin konfiguroitiin onnistuneesti toteuttamaan molempia arkkitehtuureja. Testeissä huomattiin, että koko verkon osalta DiffServ-toteutus on dominoivampi. Tämä johtuu siitä että DiffServ-verkosta siirryttäessä IntServ-verkkoon, RSVP-operaatiot eivät aktivoidu, sillä ketju ei ylety DiffServ-puolelle. Lisäksi liikenteen jättäessä IntServ-verkon, DiffServ määrää palvelutasot uudelleen, elleivät ne ole samat kuin IntServ-verkossa. Integrated Services määrää kuitenkin täysin palvelutasot RSVP-ketjujen päästä päähän. IntServ RSVP:n tehokkuutta ei pystytty kattavasti todistamaan testiverkon pienuuden vuoksi. Testiverkon koko vaikutti merkittävästi myös Differentiated Servicesin paremmuuteen testitulosten osalta.

Mielenkiintoisimmaksi haasteeksi Differentiated Services -testeissä osoittautuivat kytkimet, joiden ensisijainen tarkoitus oli toteuttaa QoS-palvelua tilanteessa, jossa runkoverkko (reitittimet) ei QoS-palvelua toteuttanut. Tällöin huomattiin myös palvelun haavoittuvuus liikenteen vastaanottajapäässä. Pullonkaulana toimineen sarjalinkin läpi tullutta liikennettä ei voitu enää QoS-palvelulla taata siedettäväksi vastaanottajapäässä. Näin ollen DSCP- ja CoS-operaatiot on aina suoritettava ennen siirtymistä tukkoiseen verkkoon. Lisäksi Catalyst 2960-kytkin, jolla ei ollut minkäänlaisia reitittäviä ominaisuuksia, osoittautui CoS-toteutuksen suhteen erittäin vaikeakäyttöiseksi. Tästä huolimatta kytkimeen pystyttiin palvelu onnistuneesti luomaan, joskin itse laitteen merkitys palvelun toteuttajana jäi toisarvoiseksi.

Tässä opinnäytetyössä esitetyt Quality of Servicesin pääarkkitehtuurit ja niiden toteutusmekaniikat luovat pohjan myös tulevaisuuden QoS-toteutusarkkitehtuureille. Quality of Servicesin protokollat ovat monikäyttöisiä, ja niitä voidaan hyödyntää monissa eri tekniikoissa, tästä hyvänä esimerkkinä Ciscon NBAR. Quality of Service -tekniikat eivät tästä huolimatta ole kiinteä osa nykyajan tietoverkkoja vaan pikemminkin lisäarvoa tuova palveluratkaisu. Päätelaitteiden (tietokoneiden) tai modeemien porttikohtaisten QoS-toteutusten tehokkuus ei vastaa läheskään sitä palvelutasoa, mitä verkkoarkkitehtuurit tarjoavat. Quality of

Servicen tulevaisuus ja laajamittaisempi käyttö kuluttajaverkoissa onkin pääasias-
sa operaattorien halukkuuden varassa. On huomattavasti helpompaa lisätä fyysistä
laajakaistakapasiteettia kuin lisätä erikseen koko verkkoa kattava palvelu. Yritys-
verkoissa tilanne on kuitenkin valoisampi. Usein yritysverkkojen palvelutarpeet
tunnetaan hyvin ja sisäverkkojen QoS-palvelu on edullinen ja helppo toteuttaa.
Tällöin säästetään paitsi rahaa, myös parannetaan verkon toimivuutta kaikilla
käyttötasoilla.

Opinnäytetyö keskittyi Differentiated Services- ja Integrated Services -
tekniikoihin runkoverkoissa. Päätelaitekohtaista tai kotikäyttöön soveltuvaa QoS-
palvelua ei tässä työssä pintaa syvemältä käsitelty. Kuluttajille suunnatun QoS-
palvelun tutkiminen ja edelleen kehittäminen parantaisi paitsi palvelun tunnetta-
vuutta, myös edistäisi sen juurtumista yleiseiskäyttöiseksi palveluksi. Lähiverkko-
jen helppokäyttöisten ja tehokkaiden QoS-tekniikoiden tutkimusta pitäisikin lisätä
ja kehitystyötä kannustaa. Tätä tukee osittain myös se tosiasia, että minimipalve-
lutason vaativat sovellukset ovat lisääntyneet nykyajan tietoverkoissa. Tästä hy-
vänä esimerkkinä työssäkin käsitelty VoIP ja videostriimaus. Koska verkkoli-
kenne on ja tulee olemaan yhä monipuolisempaa, käyttäjäkohtaisten palvelu-
tasojen hyödyntäminen lisää tulevaisuudessakin tietoverkkojen käyttöarvoa. Qua-
lity of Service ei ole vain verkon tehokkuutta lisäävä palvelu. Pelkällä siirtokais-
tan maksimoinnilla pitkitetään vain tehokkuuden ja tarjonnan kilpajuoksua, jossa
siirtoyhteydet täytetään yhä vain raskaammilla liikennekuormilla. Määrä ei ole
koskaan korvannut laatua, ei edes tietoverkoissa. Tästä syystä Quality of Service
on tutkimus- ja kehityskohteenä kannattava, tulevaisuuden nouseva tekniikka.

LÄHTEET

About.com 2008. Voice Over IP. About.com tietokantajulkaisut [viitattu 27.10.2008].

Saatavissa: <http://voip.about.com/od/voipbasics/a/qos.htm>

Ballew, S. 1998. IP-verkkojen hallinta Ciscon reitittimillä.

Espoo:Suomen ATK- kustannus Oy.

Blue Coat Systems Incorporated 2009. Enhance MPLS QoS. Blue Coat Systems verkkojulkaisut [viitattu 24.3.2009].

Saatavissa:

<https://hypersonic.bluecoat.com/packetguide/7.2.0/solutions/app-control/enhance-mpls-qos.htm>

Chappell, L. 1999. Cisco-reitittimet. Helsinki: IT Press.

Cisco Systems Incorporated 2008a. Internetworking Technology Handbook - Quality of Service. Cisco verkkojulkaisut [viitattu 15.10.2008].

Saatavissa:

<http://www.cisco.com/en/US/docs/internetworking/technology/handbook/QoS.html>

Cisco Systems Incorporated 2008b. Quality of Service - The Differentiated Services Model. Cisco verkkojulkaisut [viitattu 20.10.2008].

Saatavissa:

http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6558/ps6610/product_data_sheet0900aecd8031b36d.html

Cisco Systems Incorporated 2008c. Quality of Service - Integrated Services. Cisco verkkojulkaisut [viitattu 22.10.2008].

Saatavissa:

http://www.cisco.com/en/US/products/ps6611/products_ios_protocol_group_home.html

Cisco Systems Incorporated 2008d. Network Based Application Recognition (NBAR). Cisco verkkojulkaisut [viitattu 25.10.2008].

Saatavissa:

http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6558/ps6616/product_case_study09186a00800ad0ca.html

Cisco Systems Incorporated 2008e. Implementing Quality of Service Policies with DSCP. Cisco verkkojulkaisut [viitattu 26.10.2008].

Saatavissa:

http://www.cisco.com/en/US/tech/tk543/tk757/technologies_tech_note09186a00800949f2.shtml

Cisco Systems Incorporated 2008f. Corporate overview. Cisco verkkojulkaisut [viitattu 2.11.2008].

Saatavissa: http://newsroom.cisco.com/dlls/corpinfo/corporate_overview.html

Cisco Systems Incorporated 2008g. Catalyst 2960 Switch Software Configuration Guide. Cisco verkkojulkaisut [viitattu 10.11.2008].

Saatavissa:

http://www.cisco.com/en/US/docs/switches/lan/catalyst2960/software/release/12.2_44_se/configuration/guide/swqos.html#wp1231112

Cisco Systems Incorporated 2008h. Cisco 2801 Integrated Services Router. Cisco verkkojulkaisut [viitattu 20.11.2008].

Saatavissa: <http://www.cisco.com/en/US/products/ps6018/index.html>

Cisco Systems Incorporated 2008i. Cisco Catalyst 2960 Series Switches. Cisco verkkojulkaisut [viitattu 24.11.2008].

Saatavissa: <http://www.cisco.com/en/US/products/ps6406/>

Cisco Systems Incorporated 2008j. Cisco Catalyst Switch 2960 Software Configuration Guide. Cisco verkkojulkaisut [viitattu 26.11.2008].

Saatavissa:

http://www.cisco.com/en/US/docs/switches/lan/catalyst2960/software/release/12.2_44_se/configuration/guide/swqos.html#wp1657487

Cisco Systems Incorporated 2008k. Overview of DiffServ for QoS. Cisco verkkojulkaisut [viitattu 1.12.2008].

Saatavissa:

http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfdfrsv_ps1835_TSD_Products_Configuration_Guide_Chapter.html#wp998822

Cisco Systems Incorporated 2009a. Signaled QoS (Using RSVP). Cisco verkkojulkaisut [viitattu 3.3.2009].

Saatavissa:

http://www.cisco.com/en/US/products/ps6611/products_white_paper09186a00800ade1a.shtml#wp34608

Cisco Systems Incorporated 2009b. Configuring RSVP. Cisco verkkojulkaisut [viitattu 6.3.2009].

Saatavissa:

http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfrsvp_ps1835_TSD_Products_Configuration_Guide_Chapter.html

Cisco Systems Incorporated 2009c. Commands: ip rsvp reservation-host -- match ip dscp. Cisco verkkojulkaisut [viitattu 11.3.2009].

Saatavissa:

http://www.cisco.com/en/US/docs/ios/12_2/qos/command/reference/qrfcmd4.html

Cisco Systems Incorporated 2009d. Cisco ME 3400 Switch Software Configuration Guide, Rel. 12.2(25)SEG. Cisco verkkojulkaisut [viitattu 15.3.2009].

Saatavissa:

http://www.cisco.com/en/US/docs/switches/metro/me3400/software/release/12.2_25_seg_seg1/configuration/guide/ME3400CG.pdf

Cisco Systems Incorporated 2009e. MPLS, Quality of Service and Traffic Engineering. Cisco verkkojulkaisut [viitattu 24.3.2009].

Saatavissa:

http://www.cisco.com/en/US/docs/internetworking/technology/handbook/MPLS_Tag-Switching.html#wp1020701

Cisco Systems Incorporated 2009f. Configuring Network-Based Application Recognition. Cisco verkkojulkaisut [viitattu 26.3.2009].

Saatavissa:

http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfnbar.pdf

Cisco Press 2009. What Is MPLS TE? Cisco Press verkkojulkaisut [viitattu 7.4.2009].

Saatavissa: <http://www.ciscopress.com/articles/article.asp?p=30436&seqNum=2>

Davidson, J. & Peters, J. 2002. Voice over IP.

Helsinki: Edita Publishing Oy, IT Press.

Edward Wustenhoff 2002. Service Level Agreement in the Data Center. Sun Microsystems, Sun BluePrints™ OnLine verkkojulkaisut [viitattu 6.4.2009].

Saatavissa: <http://www.sun.com/blueprints/0402/sla.pdf>

Extreme Networks Incorporated 2000. Policy Based Quality of Service for Enterprise LANs. Extreme Networks verkkojulkaisut [viitattu 23.3.2009].

Saatavissa: <http://www.csd.uoc.gr/~hy536/PB.pdf>

IETF 2008. RFC2598, An Expedited Forwarding PHB. IETF (Internet Engineering Task Force) RFC-dokumentti julkaisut [viitattu 10.12.2008].

Saatavissa: <http://www.ietf.org/rfc/rfc2598.txt>

Jarmo Siltanen 2007. Quality of Service and Dynamic Scheduling for Traffic Engineering in Next Generation Networks. Jyväskylän Yliopiston verkkojulkaisu [viitattu 26.10.2008].

Saatavissa: <http://dissertations.jyu.fi/studcomp/9789513928117.pdf>

Joseph Davies 2007. The Cable Guy: QoS Support in Windows. TechNet Magazine artikkeliverkkojulkaisut [viitattu 23.3.2009].

Saatavissa:

[http://technet.microsoft.com/fi-fi/magazine/2007.02.cableguy\(en-us\).aspx](http://technet.microsoft.com/fi-fi/magazine/2007.02.cableguy(en-us).aspx)

Miller, P., Cummins M. 2000. LAN Technologies Explained. Boston: Digital-Press™.

Nortel Networks 2008. Introduction to Quality of Service (QoS). Nortel Networks verkkojulkaisu [viitattu 5.12.2008].

Saatavissa:

http://www.nortel.com/products/02/bstk/switches/bps/collateral/56058.25_022403.pdf

Office Manager Finland Ky kotisivut 2008. VoIPia PK-Yrityksille? Office Manager Finland Ky verkkojulkaisu [viitattu 15.11.2008].

Saatavissa: http://www.eofficemanager.net/voice_over_ip.htm

Richard Dominach 2009. QoS for IP Videoconferencing. findarticles.com, sähköinen artikkelikirjasto [viitattu 16.12.2008].

Saatavissa: http://findarticles.com/p/articles/mi_m0CMN/is_6_41/ai_n6080556

SDSU Communications Network Laboratory 2008. ReSource Reservation Protocol (RSVP). SDSU (San Diego State University) verkko-opetusmateriaalijulkaisu [viitattu 3.12.2008].

Saatavissa: <http://medusa.sdsu.edu/network/tcpip/documents/cisco-rsvp.pdf>

Sun Microsystems 2009. How to Define the Classes for Your QoS Policy (System Administration Guide: IP Services). Sun Microsystems verkkojulkaisu [viitattu 12.4.2009].

Saatavissa: <http://docs.sun.com/app/docs/doc/816-4554/ipqos-config-planning-14?a=view>

TeraBitti 2008. QoS Selvitys. TeraBitti-projektin verkkojulkaisu [viitattu 20.12.2008].

Saatavissa: http://tisu.it.jyu.fi/terabitti/Documents/QoS_Selvitys.pdf

Tim Kennedy 1999. What is streaming media? Streamingmediaworld.com sähköiset artikkelit [viitattu 22.10.2008].

Saatavissa: <http://streamingmediaworld.com/gen/tutor/whatis/>

Vesa Keränen 2009. Streaming media, kuva ja ääni virtaamaan. Tietokone.fi, sähköinen lehtiartikkeliarkisto [viitattu 7.3.2009].

Saatavissa:

http://www.tietokone.fi/lukusali/artikkelit/2002tk11/tekniikka_streaming.htm

VoIP-Info.org 2008. QoS. VoIP-Info.org sähköiset artikkelit [viitattu 2.11.2008].

Saatavissa: <http://www.voip-info.org/wiki/view/QoS>

Vonage Forum 2009. QoS ("How can I solve outbound audio problems caused by too much traffic?"). vonage.nmhoy.net, tutkimusartikkelit ja verkkojulkaisu [viitattu 25.2.2009]

Saatavissa: <http://vonage.nmhoy.net/qos.html>

Wang, Z. 2001. Internet QoS, Architectures and Mechanisms for Quality of Service. San Francisco: Morgan Kaufmann Publishers.

Wikipedia 2008a. Quality of Service. Wikipedia, sähköinen tietosanakirja [viitattu 15.10.2008].

Saatavissa: http://en.wikipedia.org/wiki/Quality_of_service

Wikipedia 2008b. Mean opinion score. Wikipedia, sähköinen tietosanakirja [viitattu 17.10.2008].

Saatavissa: http://en.wikipedia.org/wiki/Mean_opinion_score

Wikipedia 2008c. Differentiated Services. Wikipedia, sähköinen tietosanakirja [viitattu 23.10.2008].

Saatavissa: <http://en.wikipedia.org/wiki/DiffServ>

Wikipedia 2008d. Integrated Services. Wikipedia, sähköinen tietosanakirja [viitattu 26.10.2008].

Saatavissa: http://en.wikipedia.org/wiki/Integrated_services

Wikipedia 2008e. Cisco Systems. Wikipedia, sähköinen tietosanakirja [viitattu 1.11.2008].

Saatavissa: http://en.wikipedia.org/wiki/Cisco_Systems

Wikipedia 2008f. Streaming Media. Wikipedia, sähköinen tietosanakirja [viitattu 3.11.2008].

Saatavissa: http://en.wikipedia.org/wiki/Streaming_media

Wikipedia 2008g. Mean Opinion Score. Wikipedia, sähköinen tietosanakirja [viitattu 6.11.2008].

Saatavissa: http://en.wikipedia.org/wiki/Mean_Opinion_Score

Wikipedia 2009. OSI-malli. Wikipedia, sähköinen tietosanakirja [viitattu 10.3.2009].

Saatavissa: <http://fi.wikipedia.org/wiki/Sovelluskerros>

WiredRed 2008. Video Conferencing Bandwidth & Quality of Service (QoS). WiredRed yrityksen verkkojulkaisu [viitattu 24.11.2008].

Saatavissa:

<http://www.wiredred.com/video-conferencing/video-bandwidth-qos.html>

Yourdictionary.com 2008. Diffserv (tietosanakirja-artikkeli). yourdictionary.com sähköiset tietosanakirja-artikkelit [viitattu 20.10.2008].

Saatavissa: <http://www.yourdictionary.com/diffserv>

LIITTEET

LIITE 1 MOSCOW

LIITE 2 STALINGRAD

LIITE 3 BERLIN

LIITE 4 ARNHEM

LIITE 5 CAEN

LIITE 6 LENINGRAD

LIITE 7 KURSK

LIITE 8 PEARL HARBOR

LIITE 9 EL ALAMEIN

```
MOSCOW(config)#do show run
Building configuration...

Current configuration : 1694 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname MOSCOW
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
dot11 syslog
ip cef
!
!
!
!
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
multilink bundle-name authenticated
!
!
voice-card 0
!
!
!
!
!
!
!
!
!
!
archive
  log config
  hidekeys
!
!
!
class-map match-all Video
  match access-group 101
class-map match-all premium
  match ip dscp ef
!
!
policy-map RTSP
  class premium
    bandwidth 1152
policy-map SetDSCP
  class Video
    set ip dscp ef
!
!
```

```
!  
!  
!  
interface FastEthernet0/0  
  no ip address  
  shutdown  
  duplex auto  
  speed auto  
!  
interface FastEthernet0/1  
  ip address 192.168.33.254 255.255.255.0  
  duplex auto  
  speed auto  
  service-policy input SetDSCP  
  service-policy output RTSP  
!  
interface Serial0/1/0  
  no ip address  
  no fair-queue  
  clock rate 2000000  
!  
interface Serial0/1/1  
  no ip address  
  shutdown  
  clock rate 2000000  
!  
interface Serial0/2/0  
  ip address 192.168.120.1 255.255.255.0  
  clock rate 2000000  
!  
interface Serial0/2/1  
  ip address 192.168.68.1 255.255.255.0  
  clock rate 2000000  
  service-policy input SetDSCP  
  service-policy output RTSP  
!  
interface Serial0/3/0  
  no ip address  
  shutdown  
!  
interface Serial0/3/1  
  no ip address  
  shutdown  
  clock rate 2000000  
!  
router rip  
  version 2  
  network 192.168.33.0  
  network 192.168.68.0  
!  
ip forward-protocol nd  
!  
!  
ip http server  
no ip http secure-server  
!  
access-list 101 permit tcp any any eq 554  
access-list 101 permit udp any any eq 1234  
!  
!  
!
```



```
!  
!  
!  
control-plane  
!  
!  
!  
!  
!  
!  
!  
!  
!  
line con 0  
line aux 0  
line vty 0 4  
  login  
!  
scheduler allocate 20000 1000  
end
```



```
policy-map RTSP
  class premium bandwidth 1152
policy-map SetDSCP
  class Video
    set ip dscp ef
!
!
!
!
!
interface FastEthernet0/0
  no ip address
  shutdown
  duplex auto
  speed auto
!
interface FastEthernet0/1
  ip address 192.168.66.254 255.255.255.0
  duplex auto
  speed auto
  service-policy input SetDSCP
  service-policy output RTSP
!
interface Serial0/1/0
  no ip address
  shutdown
  no fair-queue
!
interface Serial0/1/1
  no ip address
  shutdown
  clock rate 2000000
!
interface Serial0/2/0
  ip address 192.168.130.1 255.255.255.0
  clock rate 2000000
!
interface Serial0/2/1
  ip address 192.168.68.254 255.255.255.0
  service-policy input SetDSCP
  service-policy output RTSP
!
interface Serial0/3/0
  no ip address
  shutdown
!
interface Serial0/3/1
  no ip address
  shutdown
  clock rate 2000000
!
router rip
  version 2
  network 192.168.66.0
  network 192.168.68.0
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
```

```
!  
access-list 101 permit udp any any eq 1234  
!  
!  
!  
!  
!  
control-plane  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
line con 0  
line aux 0  
line vty 0 4  
  login  
!  
scheduler allocate 20000 1000  
end
```

```

BERLIN(config)#do show run
Building configuration...

Current configuration : 2183 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname BERLIN
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
!
ip cef
!
!
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
multilink bundle-name authenticated
!
!
voice-card 0
  no dspfarm
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
archive
  log config
  hidekeys
!
!
!
!
class-map match-all Video
  match access-group 101
class-map match-all premium
  match ip dscp ef
!
!

```

```
policy-map RTSP
  class premium
    bandwidth 1152policy-map SetDSCP
  class Video
    set ip dscp ef
!
!
!
!
!
interface FastEthernet0/0
  no ip address
  shutdown
  duplex auto
  speed auto
!
interface FastEthernet0/1
  no ip address
  shutdown
  duplex auto
  speed auto
!
interface Serial0/0/0
  ip address 192.168.100.254 255.255.255.0
  service-policy output RTSP
  ip rsvp bandwidth 1150
!
interface Serial0/0/1
  ip address 192.168.110.254 255.255.255.0
  service-policy output RTSP
  ip rsvp bandwidth 1150
!
interface Serial0/1/0
  ip address 192.168.120.254 255.255.255.0
  shutdown
  service-policy input SetDSCP
  service-policy output RTSP
  ip rsvp bandwidth 1150
!
interface Serial0/1/1
  ip address 192.168.130.254 255.255.255.0
  shutdown
  service-policy input SetDSCP
  service-policy output RTSP
  ip rsvp bandwidth 1150
!
interface ATM0/3/0
  no ip address
  shutdown
  no atm ilmi-keepalive
  dsl operating-mode auto
!
router rip
  version 2
  network 192.168.100.0
  network 192.168.110.0
  network 192.168.120.0
  network 192.168.130.0
!
ip forward-protocol nd
!
```

```
!  
ip http serverno ip http secure-server  
!  
access-list 101 permit udp any any eq 1234  
!  
!  
!  
!  
!  
control-plane  
!  
!  
!  
!  
!  
!  
!  
!  
!  
line con 0  
line aux 0  
line vty 0 4  
  login  
!  
scheduler allocate 20000 1000  
!  
end
```



```
duplex auto
speed auto
!
interface FastEthernet0/1
 ip address 192.168.44.254 255.255.255.0
 duplex auto
 speed auto
 ip rsvp bandwidth 1150
!
interface Serial0/0/0
 ip address 192.168.100.1 255.255.255.0
 no fair-queue
 clock rate 2000000
 ip rsvp bandwidth 1150
!
interface Serial0/0/1
 ip address 192.168.77.1 255.255.255.0
 shutdown
 no fair-queue
 clock rate 2000000
 ip rsvp bandwidth 1150
!
interface Serial0/1/0
 no ip address
 shutdown
!
interface Serial0/1/1
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/2/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/2/1
 no ip address
 shutdown
 clock rate 2000000
!
interface ATM0/3/0
 no ip address
 shutdown
 no atm ilmi-keepalive
 dsl operating-mode auto
!
router rip
 version 2
 network 192.168.44.0
 network 192.168.77.0
 network 192.168.100.0
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
```

```
ip rsvp sender 192.168.44.2 192.168.55.2 UDP 1234 1234
192.168.100.254 Serial0/0/0 1150 1150ip rsvp sender 192.168.55.2
192.168.44.2 UDP 1234 1234 192.168.44.2 FastEthernet0/1 1150 1150
ip rsvp reservation 192.168.44.2 192.168.55.2 UDP 1234 1234
192.168.100.254 Serial0/0/0 FF RATE 1150 1150
ip rsvp reservation 192.168.55.2 192.168.44.2 UDP 1234 1234
192.168.44.2 FastEthernet0/1 FF RATE 1150 1150
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
  login
!
scheduler allocate 20000 1000
!
end
```



```
interface FastEthernet0/0
  no ip address
  shutdown
  duplex auto
  speed auto
!
interface FastEthernet0/1
  ip address 192.168.55.254 255.255.255.0
  duplex auto
  speed auto
  ip rsvp bandwidth 1150
!
interface Serial0/0/0
  ip address 192.168.110.1 255.255.255.0
  no fair-queue
  clock rate 2000000
  ip rsvp bandwidth 1150
!
interface Serial0/0/1
  ip address 192.168.77.254 255.255.255.0
  shutdown
  no fair-queue
  ip rsvp bandwidth 1150
!
interface Serial0/1/0
  no ip address
  shutdown
  no fair-queue
!
interface Serial0/1/1
  no ip address
  shutdown
  clock rate 2000000
!
interface Serial0/2/0
  no ip address
  shutdown
  clock rate 2000000
!
interface Serial0/2/1
  no ip address
  shutdown
  clock rate 2000000
!
interface ATM0/3/0
  no ip address
  shutdown
  no atm ilmi-keepalive
  dsl operating-mode auto
!
router rip
  version 2
  network 192.168.55.0
  network 192.168.77.0
  network 192.168.110.0
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
```

```
ip rsvp sender 192.168.44.2 192.168.55.2 UDP 1234 1234
192.168.55.2 FastEthernet0/1 1150 1150
ip rsvp sender 192.168.55.2 192.168.44.2 UDP 1234 1234
192.168.110.254 Serial0/0/0 1150 1150
ip rsvp reservation 192.168.44.2 192.168.55.2 UDP 1234 1234
192.168.55.2 FastEthernet0/1 FF RATE 1150 1150
ip rsvp reservation 192.168.55.2 192.168.44.2 UDP 1234 1234
192.168.110.254 Serial0/0/0 FF RATE 1150 1150
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
  login
!
scheduler allocate 20000 1000
!
end
```

```

LENINGRAD(config)#do show run
Building configuration...

```

```

Current configuration : 4617 bytes

```

```

!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Leningrad
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
system mtu routing 1500
ip subnet-zero
!
!
mls qos map policed-dscp 24 26 34 46 to 0
mls qos map cos-dscp 0 4 8 16 24 32 46 48
mls qos srr-queue input bandwidth 90 10
mls qos srr-queue input threshold 1 8 16
mls qos srr-queue input threshold 2 34 66
mls qos srr-queue input buffers 67 33
mls qos srr-queue input cos-map queue 1 threshold 2 1
mls qos srr-queue input cos-map queue 1 threshold 3 0
mls qos srr-queue input cos-map queue 2 threshold 1 2
mls qos srr-queue input cos-map queue 2 threshold 2 5 6 7
mls qos srr-queue input cos-map queue 2 threshold 3 3 4
mls qos srr-queue input dscp-map queue 1 threshold 2 9 10 11 12
13 14 15
mls qos srr-queue input dscp-map queue 1 threshold 3 0 1 2 3 4 5
6 7
mls qos srr-queue input dscp-map queue 1 threshold 3 32
mls qos srr-queue input dscp-map queue 2 threshold 1 16 17 18 19
20 21 22 23
mls qos srr-queue input dscp-map queue 2 threshold 2 33 35 36 37
38 39 46 48
mls qos srr-queue input dscp-map queue 2 threshold 2 49 50 51 52
53 54 55 56
mls qos srr-queue input dscp-map queue 2 threshold 2 57 58 59 60
61 62 63
mls qos srr-queue input dscp-map queue 2 threshold 3 24 25 26 27
28 29 30 31
mls qos srr-queue input dscp-map queue 2 threshold 3 34 40 41 42
43 44 45 47
mls qos srr-queue output cos-map queue 1 threshold 3 4
mls qos srr-queue output cos-map queue 2 threshold 3 3 6 7
mls qos srr-queue output cos-map queue 3 threshold 3 2 5
mls qos srr-queue output cos-map queue 4 threshold 2 1
mls qos srr-queue output cos-map queue 4 threshold 3 0
mls qos srr-queue output dscp-map queue 1 threshold 3 34 40 41 42
43 44 45 47
mls qos srr-queue output dscp-map queue 2 threshold 3 24 25 26 27
28 29 30 31
mls qos srr-queue output dscp-map queue 2 threshold 3 48 49 50 51
52 53 54 55

```

```
mls qos srr-queue output dscp-map queue 2 threshold 3 56 57 58 59
60 61 62 63
mls qos srr-queue output dscp-map queue 3 threshold 3 16 17 18 19
20 21 22 23
mls qos srr-queue output dscp-map queue 3 threshold 3 32 33 35 36
37 38 39 46
mls qos srr-queue output dscp-map queue 4 threshold 1 8
mls qos srr-queue output dscp-map queue 4 threshold 2 9 10 11 12
13 14 15
mls qos srr-queue output dscp-map queue 4 threshold 3 0 1 2 3 4 5
6 7
mls qos queue-set output 1 threshold 1 138 138 92 138
mls qos queue-set output 1 threshold 2 138 138 92 400
mls qos queue-set output 1 threshold 3 36 77 100 318
mls qos queue-set output 1 threshold 4 20 50 67 400
mls qos queue-set output 2 threshold 1 149 149 100 149
mls qos queue-set output 2 threshold 2 118 118 100 235
mls qos queue-set output 2 threshold 3 41 68 100 272
mls qos queue-set output 2 threshold 4 42 72 100 242
mls qos queue-set output 1 buffers 10 10 26 54
mls qos queue-set output 2 buffers 16 6 17 61
mls qos
!
crypto pki trustpoint TP-self-signed-3104291840
  enrollment selfsigned
  subject-name cn=IOS-Self-Signed-Certificate-3104291840
  revocation-check none
  rsakeypair TP-self-signed-3104291840
!
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
class-map match-all Video
  match access-group 101
class-map match-all premium
  match ip dscp af41
!
!
policy-map RTSP
  class premium
    set dscp af41
policy-map SetDSCP
  class Video
    set dscp af41
!
!
!
!
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
```

```
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
!
interface FastEthernet0/13
!
interface FastEthernet0/14
!
interface FastEthernet0/15
!
interface FastEthernet0/16
!
interface FastEthernet0/17
!
interface FastEthernet0/18
!
interface FastEthernet0/19
!
interface FastEthernet0/20
  srr-queue bandwidth share 90 2 6 2
  mls qos trust cos
  service-policy input SetDSCP
!
interface FastEthernet0/21
!
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
  srr-queue bandwidth share 90 2 6 2
  priority-queue out
  mls qos trust cos
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
  ip address 192.168.33.1 255.255.255.0
  no ip route-cache
  shutdown
!
ip http server
ip http secure-server
```



```
access-list 101 permit udp any any eq 1234
!
control-plane
!
!
line con 0
line vty 0 4
  login
  length 0
line vty 5 15
  login
!
end
```

```
KURSK(config)#do show run
Building configuration...
```

```
Current configuration : 4617 bytes
```

```
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname KURSK
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
system mtu routing 1500
ip subnet-zero
!
!
mls qos map policed-dscp 24 26 34 46 to 0
mls qos map cos-dscp 0 4 8 16 24 32 46 48
mls qos srr-queue input bandwidth 90 10
mls qos srr-queue input threshold 1 8 16
mls qos srr-queue input threshold 2 34 66
mls qos srr-queue input buffers 67 33
mls qos srr-queue input cos-map queue 1 threshold 2 1
mls qos srr-queue input cos-map queue 1 threshold 3 0
mls qos srr-queue input cos-map queue 2 threshold 1 2
mls qos srr-queue input cos-map queue 2 threshold 2 5 6 7
mls qos srr-queue input cos-map queue 2 threshold 3 3 4
mls qos srr-queue input dscp-map queue 1 threshold 2 9 10 11 12
13 14 15
mls qos srr-queue input dscp-map queue 1 threshold 3 0 1 2 3 4 5
6 7
mls qos srr-queue input dscp-map queue 1 threshold 3 32
mls qos srr-queue input dscp-map queue 2 threshold 1 16 17 18 19
20 21 22 23
mls qos srr-queue input dscp-map queue 2 threshold 2 33 35 36 37
38 39 46 48
mls qos srr-queue input dscp-map queue 2 threshold 2 49 50 51 52
53 54 55 56
mls qos srr-queue input dscp-map queue 2 threshold 2 57 58 59 60
61 62 63
mls qos srr-queue input dscp-map queue 2 threshold 3 24 25 26 27
28 29 30 31
mls qos srr-queue input dscp-map queue 2 threshold 3 34 40 41 42
43 44 45 47
mls qos srr-queue output cos-map queue 1 threshold 3 4
mls qos srr-queue output cos-map queue 2 threshold 3 3 6 7
mls qos srr-queue output cos-map queue 3 threshold 3 2 5
mls qos srr-queue output cos-map queue 4 threshold 2 1
mls qos srr-queue output cos-map queue 4 threshold 3 0
mls qos srr-queue output dscp-map queue 1 threshold 3 34 40 41 42
43 44 45 47
mls qos srr-queue output dscp-map queue 2 threshold 3 24 25 26 27
28 29 30 31
mls qos srr-queue output dscp-map queue 2 threshold 3 48 49 50 51
52 53 54 55
```

```
mls qos srr-queue output dscp-map queue 2 threshold 3 56 57 58 59
60 61 62 63
mls qos srr-queue output dscp-map queue 3 threshold 3 16 17 18 19
20 21 22 23
mls qos srr-queue output dscp-map queue 3 threshold 3 32 33 35 36
37 38 39 46
mls qos srr-queue output dscp-map queue 4 threshold 1 8
mls qos srr-queue output dscp-map queue 4 threshold 2 9 10 11 12
13 14 15
mls qos srr-queue output dscp-map queue 4 threshold 3 0 1 2 3 4 5
6 7
mls qos queue-set output 1 threshold 1 138 138 92 138
mls qos queue-set output 1 threshold 2 138 138 92 400
mls qos queue-set output 1 threshold 3 36 77 100 318
mls qos queue-set output 1 threshold 4 20 50 67 400
mls qos queue-set output 2 threshold 1 149 149 100 149
mls qos queue-set output 2 threshold 2 118 118 100 235
mls qos queue-set output 2 threshold 3 41 68 100 272
mls qos queue-set output 2 threshold 4 42 72 100 242
mls qos queue-set output 1 buffers 10 10 26 54
mls qos queue-set output 2 buffers 16 6 17 61
mls qos
!
crypto pki trustpoint TP-self-signed-3104291840
  enrollment selfsigned
  subject-name cn=IOS-Self-Signed-Certificate-3104291840
  revocation-check none
  rsakeypair TP-self-signed-3104291840
!
!
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
class-map match-all Video
  match access-group 101
class-map match-all premium
  match ip dscp af41
!
!
policy-map RTSP
  class premium
    set dscp af41
policy-map SetDSCP
  class Video
    set dscp af41
!
!
!
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
```

```
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
!
interface FastEthernet0/13
!
interface FastEthernet0/14
!
interface FastEthernet0/15
!
interface FastEthernet0/16
!
interface FastEthernet0/17
!
interface FastEthernet0/18
!
interface FastEthernet0/19
!
interface FastEthernet0/20
  srr-queue bandwidth share 90 2 6 2
  mls qos trust cos
  service-policy input SetDSCP
!
interface FastEthernet0/21
!
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
  srr-queue bandwidth share 90 2 6 2
  priority-queue out
  mls qos trust cos
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
  ip address 192.168.66.1 255.255.255.0
  no ip route-cache
  shutdown
!
ip http server
ip http secure-server
```

```
access-list 101 permit udp any any eq 1234
!  
control-plane  
!  
!line con 0  
line vty 0 4  
  login  
  length 0  
line vty 5 15  
  login  
!  
end
```

```
PEARL_HARBOR(config)#do show run
Building configuration...

Current configuration : 1982 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PEARL_HARBOR
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
system mtu routing 1500
ip subnet-zero
ip routing
!
!
!
!
!
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
!
!
vlan internal allocation policy ascending
!
vlan 11
!
!
class-map match-all Video
  match access-group 101
class-map match-all premium
  match ip dscp ef
!
!
policy-map RTSP
  class premium
    bandwidth 1152
policy-map SetDSCP
  class Video
    set ip dscp ef
!
!
!
interface FastEthernet0/1
  shutdown
!
interface FastEthernet0/2
  shutdown
!
interface FastEthernet0/3
  shutdown
```

```
!  
interface FastEthernet0/4  
  shutdown  
!  
interface FastEthernet0/5  
  shutdown  
!  
interface FastEthernet0/6  
  shutdown  
!  
interface FastEthernet0/7  
  shutdown  
!  
interface FastEthernet0/8  
  shutdown  
!  
interface FastEthernet0/9  
  shutdown  
!  
interface FastEthernet0/10  
  shutdown  
!  
interface FastEthernet0/11  
  shutdown  
!  
interface FastEthernet0/12  
  shutdown  
!  
interface FastEthernet0/13  
  shutdown  
!  
interface FastEthernet0/14  
  shutdown  
!  
interface FastEthernet0/15  
  shutdown  
!  
interface FastEthernet0/16  
  shutdown  
!  
interface FastEthernet0/17  
  shutdown  
!  
interface FastEthernet0/18  
  shutdown  
!  
interface FastEthernet0/19  
  shutdown  
!  
interface FastEthernet0/20  
  bandwidth 100000  
  service-policy input SetDSCP  
  service-policy output RTSP  
!  
interface FastEthernet0/21  
  shutdown  
!  
interface FastEthernet0/22  
  shutdown  
!  
interface FastEthernet0/23
```

```
shutdown
!
interface FastEthernet0/24
  port-type nni
  bandwidth 100000
  service-policy input SetDSCP
  service-policy output RTSP
!
interface GigabitEthernet0/1
  port-type nni
!
interface GigabitEthernet0/2
  port-type nni
!
interface Vlan1
  ip address 192.168.44.1 255.255.255.0
!
no ip http server
ip http secure-server
ip classless
!
!
access-list 101 permit udp any any eq 1234
!
!
control-plane
!
!
line con 0
line vty 5 15
!
end
```



```
EL_ALAMEIN(config)#do show run
Building configuration...

Current configuration : 1983 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname EL_ALAMEIN
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
system mtu routing 1500
ip subnet-zero
ip routing
!
!
!
!
!
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
!
!
vlan internal allocation policy ascending
!
vlan 11
!
!
class-map match-all Video
  match access-group 101
class-map match-all premium
  match ip dscp ef
!
!
policy-map RTSP
  class premium
    bandwidth 1152
policy-map SetDSCP
  class Video
    set ip dscp ef
!
!
!
interface FastEthernet0/1
  shutdown
!
interface FastEthernet0/2
  shutdown
!
interface FastEthernet0/3
  shutdown
```

```
!  
interface FastEthernet0/4  
  shutdown  
!  
interface FastEthernet0/5  
  shutdown  
!  
interface FastEthernet0/6  
  shutdown  
!  
interface FastEthernet0/7  
  shutdown  
!  
interface FastEthernet0/8  
  shutdown  
!  
interface FastEthernet0/9  
  shutdown  
!  
interface FastEthernet0/10  
  shutdown  
!  
interface FastEthernet0/11  
  shutdown  
!  
interface FastEthernet0/12  
  shutdown  
!  
interface FastEthernet0/13  
  shutdown  
!  
interface FastEthernet0/14  
  shutdown  
!  
interface FastEthernet0/15  
  shutdown  
!  
interface FastEthernet0/16  
  shutdown  
!  
interface FastEthernet0/17  
  shutdown  
!  
interface FastEthernet0/18  
  shutdown  
!  
interface FastEthernet0/19  
  shutdown  
!  
interface FastEthernet0/20  
  bandwidth 100000  
  service-policy input SetDSCP  
  service-policy output RTSP  
!  
interface FastEthernet0/21  
  shutdown  
!  
interface FastEthernet0/22  
  shutdown  
!  
interface FastEthernet0/23
```

```
shutdown
!
interface FastEthernet0/24
  port-type nni
  bandwidth 100000
  service-policy input SetDSCP
  service-policy output RTSP
!
interface GigabitEthernet0/1
  port-type nni
!
interface GigabitEthernet0/2
  port-type nni
!
interface Vlan1
  ip address 192.168.55.1 255.255.255.0
!
no ip http server
ip http secure-server
ip classless
!
!
access-list 101 permit udp any any eq 1234
!
!
control-plane
!
!
line con 0
line vty 5 15
!
end
```