hyväksymispäivä arvosana

arvostelija

Multimediaprotokollat

Juha Pesonen

Helsinki 3. huhtikuuta 2001

Multimedia ja tietoliikenne-seminaari

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Multimediaprotokollat

Juha Pesonen

Multimedia ja tietoliikenne-seminaari
Tietojenkäsittelytieteen laitos
Helsingin yliopisto
3. huhtikuuta 2001, 14 sivua

Reaaliaikaisen datan ja sen myötä myös multimediapalvelujen tarjoaminen Internetissä on jatkuvasti lisääntymässä. Reaaliaikaisia palveluja ovat mm. web-kamerat, videokonferenssit tai internetin kautta tapahtuvat puhelut. Verrattuna perinteiseen verkkoliikenteeseen, em. palvelut asettavat huomattavasti tiukempia ajallisia vaatimuksia, mutta toisaalta taas heikompia laadullisia vaatimuksia. On selvää, että ne protokollat joita perinteinen verkkoliikenne käyttää, eivät sellaisenaan sovi reaaliaikaisen datan siirtoon. Tässä artikkelissa tarkastellaan muutamia vaihtoehtoja Internetissä käytettäviksi multimediaprotokolliksi.

Aiheluokat(Computing Reviews 1998): C.2.2, C.2.3, H.4.3 ,H.5.1

Avainsanat: reaaliaikaprotokolla, Internet, multimedia, RTP, RTSP, RSVP

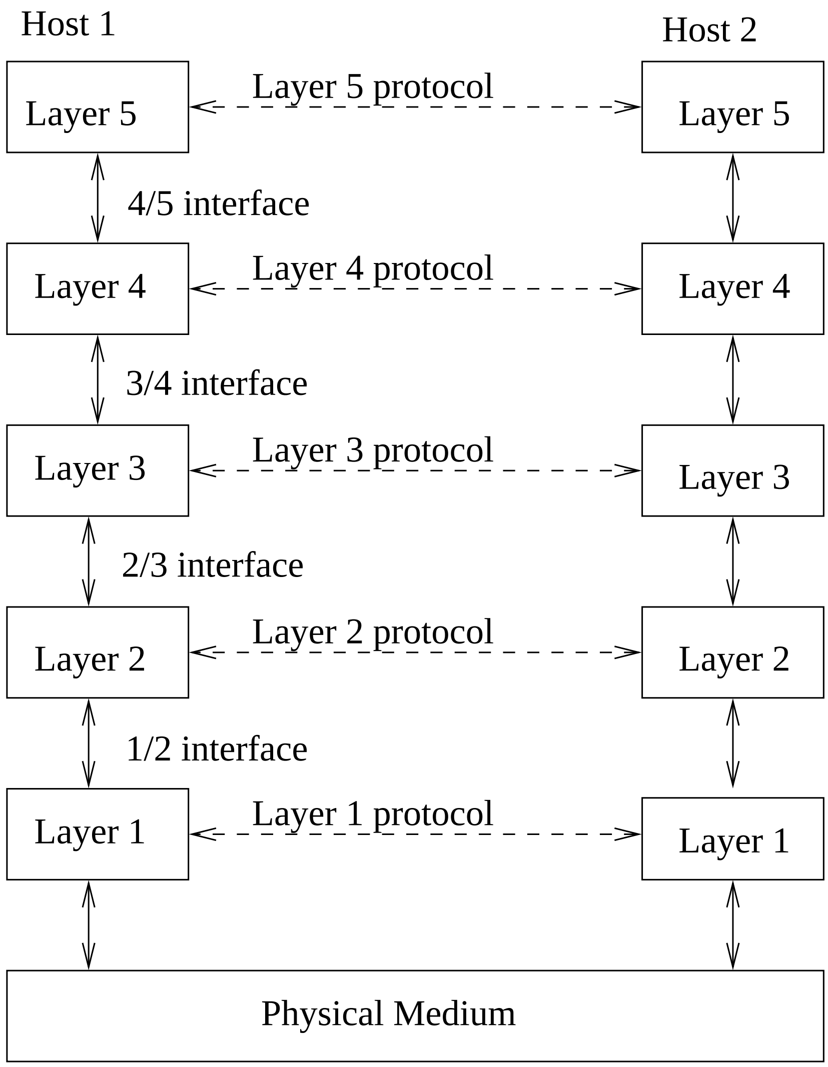
Sisältö

1 Johdanto

Protokolla on ennalta sovittu yhteyskäytäntö tiedon lähettäjän ja vastaanottajan välillä. Protokollalta vaadittava toiminnallisuus riippuu käytettävistä päätelaitteista, liikuteltavan tiedon luonteesta ja välissä olevasta siirtotiestä. Tyypillisesti yhteen kommunikaatiotapahtumaan osallistuu useita protokollia abstraktiotason noustessa lähestyttäessä tiedon loppukäyttäjää. Ylemmät tasot keskustelevat keskenään virtuaalisesti (katkoviivalla kuvassa 1), ja todellinen kommunikaatio osapuolten välillä tapahtuu vasta alimmalla tasolla käyttäen fyysistä mediaa. Arkimaailman esimerkki protokollasta on vaikka kahden ihmisen tapaaminen: henkilöstä ja tilanteesta riippuen vastapuolen kättä voi joko puristaa tai suudella. Tässäkin tapauksessa protokollan rikkominen tekee kommunikaatiosta hankalaa tai mahdotonta [Tan96]. Tässä artikkelissa käydään läpi yleisiä multimediaprotokollien vaatimuksia ja tarkastellaan joitakin todennäköisiä standardeja Internetin multimediaprotokolliksi. Käsiteltävät protokollat ovat käyttäjätason protokollia.

1.1 Historiaa

Aikaisemmin koneet olivat hyvin kalliita suhteessa saatuun tehoon, joten vaati suuria investointeja jotta saavutetaan siedettävä siirtonopeus suhteessa datan virheettömyyteen. Luonteva ratkaisu oli antaa verkon hoitaa suurin osa likaisesta työstä. Nykyiset verkot muuttuvat koko ajan fyysisesti luotettavammiksi, ja koneiden teho kasvaa, joten tehtäviä voidaan siirtää verkolta päätelaitteiden hoidettaviksi. Tyypillisiä protokollien tehtäviä ovat tiedon kuljetus, virheenkorjaus, vuonvalvonta, ja ruuhkautumisen esto.



Kuva 1: Protokollapino

2 Protokollat

2.1 Miksi useita protokollia

verkkojen perusrakenteesta johtuen (kerrosmalli) tarvitaan joka tapauksessa eri protokollia eri verkkokerroksille. Lisäksi eri tyyppisen datan siirtoon ei sovellu yksi ja sama protokolla johtuen sovellusten erilaisista tarpeista. Tärkeimmät erot ovat virheisiin suhtautuminen ja siirron jatkuvuus ja/tai nopeus. Multimedian siirto vaatii protokollalta ominaisuuksia jotka ovat tarpeettomia tai jopa haitallisia viestipohjaiselle tiedonsiirrolle ja päinvastoin. Esim. kaikkien virheiden korjaus ja puuttuvien pakettien uudelleen lähetys ovat videokuvan siirrossa turhia ominaisuuksia, mutta välttämättömiä esim. HTML-sivun latauksessa. Protokollia tarvitaan myös muihin tehtäviin kun varsinaiseen datasiirtoon. Erilaiset datasiirtoa tukevat toiminnot kuten resurssien varaus tai lähettäjän ja vastaanottajan välinen kommunikointi yhteyden aikana on järkevää antaa erikoistuneen protokollan tehtäväksi.

2.2 Ruuhkautuminen

Yleisin ongelma Internetissä tapahtuvassa tiedonsiirrossa on ruuhkan muodostuminen johonkin verkon solmuun. Ruuhkautuminen pahentaa itse itseään koko ajan, sillä ruuhkautunut reititin alkaa hävittää paketteja puskuritilan loppuessa. Kun lähettäjä ei koskaan saa kuittausta lähettämäänsä viestiin, se lähetetään uudelleen. Tämä tietenkin lisää pakettien määrää verkossa, ja siten itse ruuhkaa. Ruuhkautumisen välttämisessä ja poistamisessa on protokollilla ratkaisevan tärkeä osa. Nagle (1987) huomasi, että vaikka reitittimissä olisi äärettömän paljon muistia, ruuhkautuminen olisi itse asiassa pahempaa kuin pienemmällä muistin määrällä. Tämä johtuu siitä, että kun muistin määrää kasvatetaan, paketit viettävät reitittimissä entistä pidemmän ajan, ja jos lähettäjä ei saa kuittausta määrätyssä ajassa, se lähettää uuden paketin. Kun reitittimessä ollut ensimmäinen paketti pääsee käsiteltäväksi, se on jo vanhentunut, ja samoin käy lähettäjältä tulleille duplikaateille [Tan96]. Ruuhkaa tulee Internetissä olemaan varmasti tulevaisuudessakin, ja nykyisin käytetyt protokollat eivät siten sovellu kunnolla aikakriittisen tiedon välitykseen.

2.3 Protokolliin liittyviä käsitteitä

Yhteydetön (connectionless) protokolla on sellainen, joka ei tiedon siirtämistä varten edellytä alustustoimenpiteitä, ts. yhteyden muodostamista lähettäjän ja vastaanottajan välillä, eikä sillä ole samaan *istuntoon* (session) liittyvien viestien välitysaikana käytettävissään yhteyden tuomia ominaisuuksia kuten varmuutta pakettien saapumisjärjestyksestä.

Yhteydellinen (connection-oriented) protokolla taas vaatii datan siirtoa varten erillisen alustus (ja purku) vaiheen, jolloin yhteyden muodostaja lähettää yhteyspyynnön vastaanottajalle, saa vastauksen, ja jonka jälkeen yhteys joko muodostetaan tai ollaan muodostamatta. Kaikki lähetettävä data kuuluu tämän jälkeen samaan yhteyteen, ja jos yhteys jossakin vaiheessa katkeaa, joudutaan muodostamaan uusi yhteys. Yhteydellisessä siirrossa järjestelmä takaa viestien saapumisen oikeassa järjestyksessä vastaanottajalle. Yhteydelliseen siirtoon liittyy usein käsite synkroninen ja vastaavasti yhteydettömään asynkroninen. Esimerkkinä vaikka puhelinkeskustelu vs. tekstiviestin lähetys.

Tietovirta (stream) taas voidaan määritellä seuraavasti: jos vastaanottaja ei voi purkaa vastaanottamaansa dataa erillisiin vastaanottajan tarkoittamiin viesteihin, ts. vastaanotettu data on vain virtaavaa tietoa joka alkaa jollakin hetkellä ja loppuu ennalta määräämättömän ajan T kuluttua, on kyseessä tietovirta. Usein tämä aika tosin kerrotaan käyttäjälle ennalta mukavuussyistä, jotta hän esimerkiksi voi selvittää elokuvan etenemiskohdan aikalaskurista. Tietovirtakäsite mahdollistaa sen, että voimme aloittaa esimerkiksi videon katsomisen alusta vaikka se ei ole vielä kokonaisuudessaan siirtynyt esittävälle koneelle.

Esimerkkejä yhteydettömistä palveluista ovat vaikka sähköposti ja tietokantakysely ja yhteydellisistä ftp-siirto tai tietovirta [Tan96].

Ajatellaan sitten tilannetta, jossa yksi vastaanottaja lähettää tietoa useille vastaanottajille. Tiedonsiirto voidaan järjestää siten, että jokaiseen vastaanottajaan muodostetaan oma yhteys ja lähettäjä lähettää jokaiselle vastaanottajalle saman datan sisältävän paketin, jossa vain osoite muuttuu. Tällaisen toimintamallia kutsutaan *yksittäislähetykseksi* (unicast). Tämä aiheuttaa kuitenkin tarpeetonta kuormaa verkossa. Lähettämällä data vain kerran ja antamalla verkon huolehtia tiedon jakamisesta vasta siinä pisteessä kun eri vastaanottajille ei enää löydy yhteistä reittiä, saavutetaan huomattavia kuormitussäästöjä. Tällaisen mekanismin nimi on *monilähetys* (multicast), ja se toteutetaan liittämällä vastaanottajat samaan *monilähetysryhmään* (multicast group), johon voidaan viitata yhdellä osoitteella. Verkon reitittimet huolehtivat tämän jälkeen pakettien ohjaamisesta oikeille vastaanottajille.

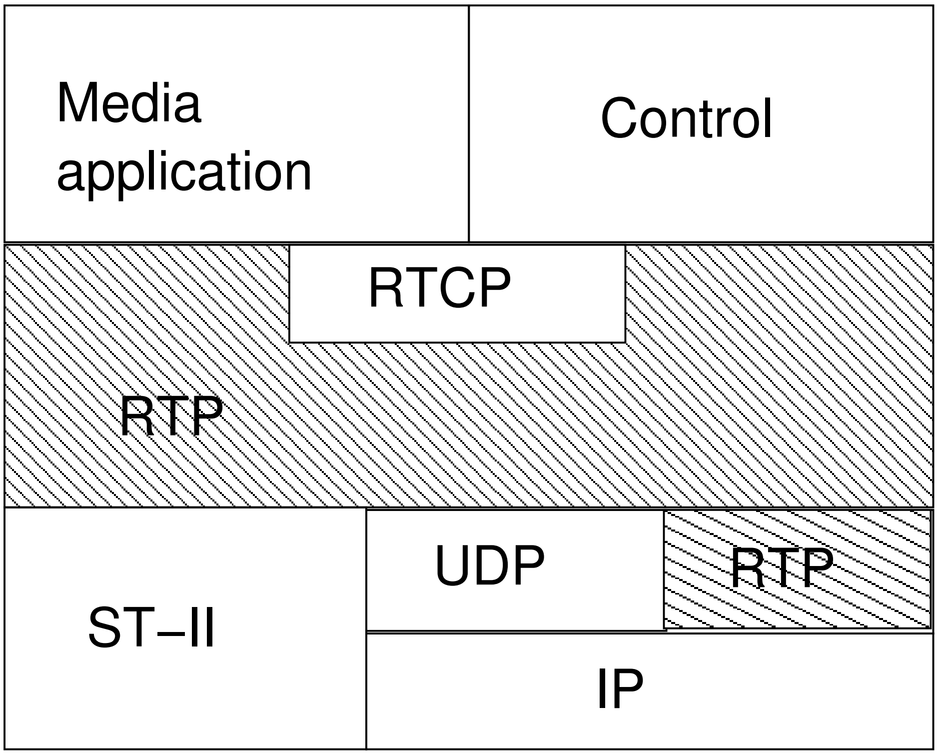
3 Reaaliaikaprotokollat

Reaaliaikaisuutta vaativaa kommunikointia ovat esimerkiksi kuvan, äänen, tai virtuaalisimulaation siirtäminen. Nykyisin yleisessä käytössä olevista protokollista UDP (user datagram protocol tai unreliable datagram protocol) sopisi parhaiten multimedian siirtoon tahokkuutensa ansiosta, mutta siinä on muutamia vakavia puutteita reaaliaikaisuutta ajatellen. Siitä puuttuvat mm. viiveiden hallinta (aikaleima), synkronointimahdollisuus eri mediatyyppien välillä sekä mediavirtojen välinen yhdistäminen ja erottaminen (multiplexing, demultiplexing). Näitä ominaisuuksia voitaisiin osittain siirtää sovelluksen tehtäväksi, mutta se olisi tehotonta ja rikkoo modulaarisuuden käsitettä sovellusten ja protokollien suunnittelussa. Tästä syystä tilannetta korjaamaan on kehitetty, ja kehitetään jatkuvasti, uusia protokollia. Seuraavassa tarkastellaan suhteellisen yleisellä tasolla muutamia reaaliaikaisen datan siirtoon liittyviä protokollia. Ensimmäisenä erästä varsinaista siirtoprotokollaa, sen jälkeen siirtoon tarvittavien resurssien varaamiseen tarkoitettua protokollaa ja lopuksi reaaliaikayhteyksien hallintaprotokollaa.

3.1 RTP

Seuraavassa kappaleessa viitattaessa sekä RTP-protokollaan, että RTCP-protokollaan, on käytetty kursivoitua merkintää *RTP*. Puhuttaessa protokollista erikseen on merkintä RTP tai RTCP.

RTP (Real Time Protocol) on ratkaisu korkeamman tason protokollaksi reaaliaikaisen datan siirtoon internetissä. *RTP* on epäluotettava protokolla, eli se ei takaa pakettien saapumista perille, niiden saapumisjärjestystä tai virheettömyyttä. Tämä on ymmärrettävä ratkaisu, koska *RTP* voi todennäköisesti pienemmin kustannuksin 'ostaa' palvelun alapuolella olevilta verkkokerroksilta. *RTP* toimii sovelluksen ja ylemmän tason siirtoprotokollan (kuten TCP tai UDP) välissä, ja sen tarjoamat lisäpalvelut tukevat reaaliaikadatan siirtoa. Näitä palveluja ovat mm. siirrettävän datan tyypin ilmaiseminen (payload type), siirtoviiveiden (delay) ja viiveen vaihteluiden (delay jitter) hallinta paketeissa olevien aikaleimojen avulla. *RTP* tukee sekä yhdeltä yhdelle (unicast), että yhdeltä monelle (multicast) tyyppistä tiedonsiirtoa.



Kuva 2: Esimerkki RTP-toteutus

3.2 RTP:n ominaisuudet

RTP toimii yhdessä RTCP (RTP Control Protocol) kontrolliprotokollan kanssa. RTP on yksisuuntainen protokolla varsinaisen datan siirtoon ja RTCP kaksisuuntainen ohjausprotokolla lähettäjän ja vastaanottajan väliseen viestintään. Suurin osa protokollan toiminnasta keskittyy RTCP:n ympärille. RTP:n voitaisiin ajatella olevan vain kanava jota pitkin tieto siirtyy ja jota ohjataan RTCP:llä. RTP kykenee paketissa olevan tunnisteen avulla kuljettamaan samassa yhteydessä useita eri mediatyyppejä, esim kuvaa ja ääntä. Erillisten virtojen synkronointi on taas jätetty RTCP:n harteille. *RTP* on kehitetty tuomaan reaaliaikaisen datan tuki olemassa olevien menetelmien päälle, eikä se ole sidottu mihinkään tiettyyn toteutusympäristöön. Alun alkaen *RTP* on suunniteltu ratkaisemaan Internetissä olevia ongelmia, mutta se voidaan yhtä hyvin ottaa käyttöön muissakin ympäristöissä. *RTP* takaa melko hyvän turvallisuuden, sillä jokainen paketti on mahdollista salata käyttäen jotakin tunnettua menetelmää. Oletuksena on DES (data encryption standard). Tulevaisuudessa alemman tason protokollat IPv6 (Internet protocol version 6 tai IPng, IP next generation), IPSEC (IP secure) tukevat salausta, joten sen suorittaminen muuttuu tarpeettomaksi RTP:lle. On kuitenkin katsottu, että vaikka RTP:n salaus ei olisikaan enää tarpeen, sen lisääminen RTP:n ominaisuuksiin ei vähennä protokollan tehokkuutta merkittävästi. Salausmahdollisuuden lisäksi jokaisen osallistujan tunniste on satunnaisesti generoitu arvo RTCP-paketissa. Käytännössä on siis hyvin vaikeaa päätellä pakettien lähettäjän tai vastaanottajan todellista henkilöllisyyttä [SCF96].

3.3 RTCP

RTCP-viesti koostuu useista eri pakettityypeistä, joita voidaan samaan viestiin pinota useita. Kaksi pakettityyppiä joista toisen on aina oltava mukana lähetetyssä RTCP-viestissä ovat *lähettäjän raportti* (sender report, SR) ja *vastaanottajan raportti* (receiver report, RR). Nämä raportit sisältävät mm. sen lähettäjän tunnisteen ja QoS:n hallintaan liittyvää tietoa, joka onkin RTCP:n tärkein tehtävä.

Kun sovellus on lähettänyt dataa vastaanottajalle tai vastaanottajille, se lähettää perään lähettäjän raportin. Tämän jälkeen vastaanottajat lähettävät palautteena vastaanottajan raportin kaikille yhteyden osapuolille. Kun datan lähettäjä vastaanottaa raportit, se kykenee päättelemään näissä olevista tiedoista sillä hetkellä toteutuvan QoS:n. Samaten kaikki muut yhteyteen osallistuvat ovat tietoisia QoS:stä. Tämä mahdollistaa QoS:n muokkaamisen yhteyden aikana ennalta sovittujen parametrien mukaisesti. Verkko voidaan jakaa erilaisiin toiminnallisiin luokkiin, kuten vapaa, kuormitettu ja ruuhkainen. Esimerkiksi jos 20% vastaanottajilla on ongelmia yhteydessä, ei välttämättä kannata tehdä vielä mitään. Jos sitävastoin 50% vastaanottajista saa vain osan tarkoitetusta datasta, tehdään demokraattinen päätös ja ruvetaan käyttämään (esim) kuormitetulle verkolle tarkoitettua QoS:eä. Tällä tavoin palvelun laatua kyetään valvomaan dynaamisesti yhteyden aikana. Tämä mekanismi toimii hyvin, jos verkon kyky tarjota tietty kaistanleveys vaihtelee järjellisissä rajoissa. Lähetettävän datan laatua ei tietenkään voida loputtomasti huonontaa, eikä RTP:lla ole paljoa tehtävissä jos jokin reitti osoittautuu laadultaan kelvottomaksi. Tällaisessa tilanteessa yhteys todennäköisesti menetetään [BDS96].

RTCP:n tehtäviin kuuluu QoS:n valvonnan lisäksi mediavirtojen synkronointi pakettien aikaleimojen perusteella, ja riippumatta koneiden todellisesta kellonajasta (wallclock time). Tämä on mahdollista, koska paketissa on kaksi aikaleimaa, toinen kertoo ajan suhteessa edeltävään pakettiin ja toinen todellisen kellonajan. Lisäksi RTCP:n avulla tunnistetaan yhteyden osapuolet.

RTCP huolehtii myös omien lähetystensä rajoittamisesta tarvittaessa. Muualla paitsi valtion byrokratiassa, hallintakulujen nouseminen suhteettoman suuriksi varsinaiseen hyötyyn nähden ei ole hyväksyttävää. Tämän vuoksi valitaan jokin arvo, esimerkiksi 5%, joka on suurin osuus mitä RTP-yhteydessä saa liikkua RTCP viestejä nähden RTP:n välittämään dataan. Ongelmia saattaa syntyä silloin, kun yhteyttä muodostetaan, ja RTCP-paketteja lähtee verkkoon suuria määriä yhtä aikaa [SCF96].

3.4 RSVP

RSVP (Resource ReserVation Protocol) on suhteellisen monimutkainen protokolla, ja seuraavassa pyritään antamaan vain pintapuolinen kuvaus sen toiminnallisuudesta.

Jotta vaaditun QoS:n tarjoaminen olisi mahdollista, edellytetään joko sellaista verkkoa, jonka käyttäytyminen ja käyttöaste tunnetaan ennalta (PPP-linkki, suljettu lähiverkko), tai vastaavan tilanteen luomista julkiseen verkkoon muilla keinoilla. Ongelma on merkityksellisin Internet-ympäristössä, ja ratkaisu on suhteellisen luonteva Internetin rakenteen huomioon ottaen. Jos kyetään luomaan tilanne, jossa Internetin solmukoneet (reitittimet) sisältävät jollekin tietovirralle etuoikeutetun väylän, jonka läpimeno kyetään takaamaan normaaleissa olosuhteissa, voidaan myös QoS:n säilyminen taata. Tällainen ratkaisu edellyttää tietenkin väylän säilymistä kaikissa reitin varrella olevissa solmuissa. RSVP on protokolla, joka varaa pyydetyn kokoisen (mikäli mahdollista) osan kaistaa kaikista solmuista lähettäjien ja vastaanottajien väliltä mukaanlukien lähettävät ja vastaanottavat koneet. RSVP:n toiminta perustuu vastaanottajien lähettämien varauspyyntöjen kuljettamiseen samaa reittiä jota varsinainen data tulee kulkemaan, ja resurssien varaamiseen matkan varrelta. Yksi RSVP:n suurimmista haasteista on selviytyä suuresta samanaikaisesta vastaanottajien lukumäärästä kuitenkaan hukkaamatta resursseja tarpeettomasti [BEB97].

3.5 Varaustapahtuma

Vastaanottajan lähettäessä varauspyynnön lähettäjälle varauspyyntö etenee puumaisesti solmusta toiseen kohti datan lähettäjää. Kun saavutaan ensimmäiseen sellaiseen solmuun, jossa on jo olemassa oleva varaus samalle yhteydelle, varauspyynnön edelleenlähettäminen kyseisen vastaanottajan osalta keskeytetään. Tämän jälkeen reitittimessä olemassa oleva varaus ja juuri saapunut varaus yhdistetään, ja suurempi edellä mainituista jatkaa matkaansa kohti lähettäjää. Mikäli varaus onnistuu, on tuloksena tietyn QoS:n tarjoava reitti lähettäjältä (tai lähettäjiltä) vastaanottajille. RSVP huolehtii yhteyden aikana myös siitä, että käyttämättömät varaukset vapautetaan tarvittaessa. RSVP on riippuvainen reitityksestä, mutta ei itsessään ole reititysmekanismi. Periaate on se, että erillinen reititysalgoritmi kertoo mihin paketit reititetään, ja RSVP ottaa kantaa siihen mitä niille reitittimessä tehdään. RSVP edellyttää, että lähettäjä ja vastaanottaja vaihtavat jatkuvasti RSVP:n edellyttämiä ohjaustietoja. Resurssien alustus ja yhteyden aikana vaihdetut viestit eivät eroa toisistaan millään tavalla. Tämä mahdollistaa RSVP:n sopeutumisen reitityksen muutoksiin, vastaanottajien ja lähettäjien kokoonpanossa tapahtuviin muutoksiin ja QoS-parametrien yhteyden aikana tapahtuviin muutoksiin läpinäkyvästi. Jos ohjaustietoja ei kulje reitin varrella olevien solmujen kautta, varatut resurssit vapautetaan. RSVP-reititin huolehtii myös liikenteestä, jolle ei ole pyydetty resursseja. Kun tällainen paketti saapuu reitittimeen, se sijoitetaan sellaiseen QoS-ryhmään jossa on vapaata tilaa. Näin pystytään takaamaan myös priorisoimattoman datan kuljetus. RSVP tuo mukanaan myös yhden kenties epätoivotun ominaisuuden: sellainen vastaanottaja joka ei ole varannut resursseja, liittyy esimerkiksi videokonferenssiin sellaisen reitittimen takana jossa on kyseiselle yhteydelle varaus valmiina. Tämä vastaanottaja saa taatun QoS:n vaikka ei ole siitä esimerkiksi maksanut. Tämän seikan on kuitenkin katsottu olevan välttämätön paha, ja sen poistaminen edellyttäisi suuria muutoksia arkkitehtuurissa [BEB97].

3.6 Vaiheet

Ensimmäiseksi suoritetaan yhteyden alustus, jolloin vastaanottajat siis lähettävät varauspyynnön lähettäjälle. Seuraavassa oletamme, että varauspyyntö on saapunut ensimmäiseen reitin varrella olevaan RSVP-reitittimeen. Seuraavaksi tarkastetaan voidaanko pyydetty QoS tarjota. Mikäli näin ei ole, pyytäjälle lähetetään virheilmoitus. RSVP käyttää tämän selvittämiseen käyttöjärjestelmän palvelua. Tämän jälkeen RSVP tutkii, onko varauksen pyytäjällä oikeutta varauksen tekemiseen. Tähän vaiheeseen liittyy paljon sellaisia näkökohtia, jotka eivät ole automaattisesti pääteltävissä, kuten laskutus tai erilaiset kieltolistat. RSVP siis pyytää niinikään jotakin muuta sovellusta avukseen. Mikäli varauspyyntö on edennyt näin pitkälle, voidaan reitittimeen pakettien priorisoijaa pyytää lisäämään reitittimeen uusi tila pyydettyä QoS:eä varten. Tämän jälkeen pakettien luokittelija vastaa siitä, että sisään tulevat paketit yhdistetään oikein niille kuuluvaan QoS-ryhmään yhdessä ilman QoS-varausta olevien pakettien kanssa.

3.7 RTSP

RTSP (Real Time Streamig Protocol), on tarkoitettu verkossa olevien reaaliaikadataa tuottavien palvelinkoneiden 'kauko-ohjaimeksi'. RTSP-protokolla käyttää *pyyntö-*

vastaus (request-reply) tyyppistä tiedonsiirtoa asiakkaan ja palvelimen välillä, eli se on yhteydetön. RTSP mahdollistaa esimerkiksi verkossa eri paikassa sijaitsevilta palvelimilta saatavan tietovirran yhdistämisen multimediaesitykseksi vastaanottajakoneella. RTSP ei itse siirrä varsinaista dataa, vaan ohjaa jotakin toista protokollaa kuten RTP:aa. RTSP on syntaksin ja toiminnan osalta samanlainen kuin HTTP (hyper text transport protocol). Tällä saavutetaan se etu, että RTSP on jäsennettävissä nykyisillä ja tulevilla HTTP-jäsentäjillä. HTTP:n mekanismien uudelleenkäyttö helpottaa myös kahden muun ongelma ratkaisua: turvallisuus (tunnistus, salaus jne.) ja *välityspalvelimet* (proxy). Koska HTTP:n ja RTSP:n vaatimukset ovat hyvin samanlaisia, niihin voidaan soveltaa samanlaisia ratkaisuja em. kohdissa. HTTP:n kanssa yhteensopivuutta puoltaa myös se seikka, että yhteyspyyntö reaaliaikaista dataa lähettävälle palvelimelle tehdään usein käyttäen www-selainta. HTTP ei ole kuitenkaan edellytys RTSP:n käytölle; on siis mahdollista toteuttaa pelkästään RTSP:aa käyttävä sovellus. Joissain kohdissa RTSP kuitenkin eroaa HTTP:sta, seuraavassa tärkeimmät:

- Oma protokollatunnus ja useita uusia toimintoja
- Sekä palvelin että asiakas voivat suorittaa palvelupyyntöjä
- Varsinainen datan siirto tapahtuu (yleensä) toisen protokollan avulla
- Tilansa säilyttävä protokolla. RTSP kutsu voi asettaa yhteyteen liittyviä parametreja jotka jäävät voimaan pyynnön kuittauksen jälkeenkin.

RTSP:n toiminta edellyttää, että joltakin verkon palvelimelta on haettu *esitystapakuvaus* (presentation description) Esitystapakuvaus voidaan pyytää käyttäen HTTP-protokollaa, RTSP-protokollaa tai vaikkapa sähköpostia, eikä sen tarvitse välttämättä sijaita varsinaista dataa lähettävällä palvelimella. Esitystapakuvauksesa määritellään esityksessä käytettävät formaatit, niiden koodaustapa ja tietoa esityksen sisällöstä. Esitystapakuvaus sisältää myös dataa lähettävien palvelinten sijainnin verkossa RTSP URL-viitteinä. Tämä mahdollistaa sen, että esitys voi koostua useilta eri palvelimilta saatavista tietovirroista. Esimerkki RTSP URL-viitteestä seuraavassa:

RTSP URL on muotoa:

```
rtsp://media.example.com:554/twister/audiotrack
```

, missä

- rtsp:// tarkoittaa, että käytetään RTSP-protokollaa TCP-yhteyden kautta. Voitaisiin valita myös rtspu://, jolloin alemman tason protokolla on UDP tai rtsps:// suojatulle yhteydelle (secure socket layer, SSL).
- 554 on RTSP:n käyttämä (IP) portti.
- twister on haettavan esityksen nimi.
- audiotrack on yhden tietovirran nimi esityksessä. Tätä ei ole välttämätöntä määritellä.

Seuraavassa esimerkki tavasta yhdistää RTSP:n käyttö HTML-sivuun:

```
<EMBED SRC="http://lancs.ac.uk/webaudio.sdf" TYPE="application/x-audio">
```

1. HTTP:n kautta haetaan esitystapakuvaus
2. RTSP alustaa yhteyden ja ohjaa sitä
3. RSVP varaa resurssit
4. RTP huolehtii datan siirrosta

Itse RTSP-session ei tarvitse käyttää samaa alemman tason protokollaa kuin varsinaisen datan siirto; RTSP-viestejä voidaan siis siirtää käyttäen esimerkiksi TCP-protokollaa, ja datan lähetys voi tapahtua UDP:n tai RTP:n avulla. Tämä johtaa siihen, että vaikka RTSP-kyselyihin ei saatasi vastauksia, esitys voi silti jatkua normaalisti. Koska kyseessä on yhteydetön protokolla, esitykseen liittyvät RTSP-viestit voidaan lähettää kukin käyttäen esimerkiksi eri TCP-yhteyttä toiminnan häiriintymättä [SRL98].

RTSP:n määrittelee monia muitakin ominaisuuksia, mutta seuraavassa luetellaan multimediaesityksen kannalta oleellisimmat:

SETUP: Palvelin varaa resurssit datayhteydelle ja aloittaa RTSP-session.

PLAY ja RECORD: Aloittaa datan siirron SETUP-vaiheessa avattuun yhteyteen.

PAUSE: Pysäyttää datan lähettämisen palvelimelta ilman resurssien vapauttamista.

TEARDOWN: Vapauttaa varatut resurssit ja tuhoaa avatun RTSP-session.

RTSP on edelleen kehitysasteella, jossa sitä on ehdotettu Internet-standradiksi (proposed standard), mutta mitä todennäköisimmin tulee myöhemmin saavuttamaan vankan jalansijan Internetissä. Kun IPv6:n käyttö yleistyy, RTSP:n ominaisuuksia kyetään myös hyödyntämään tehokkaammin kuin nykyisin. RealNetworksin RealPlayer ja RealMedia SDK ovat ilmeisesti ainoat varteenotettavat sovellukset, jotka tukevat RTSP:aa tällä hetkellä.

4 Yhteenveto

Multimedian siirto tietoverkoissa vaatii laitteistolta ja protokollilta sellaisia ominaisuuksia, jotka ovat monelta osin suhteellisen varhaisessa kehityvaiheessa. Multimedian kasvava kysyntä kuitenkin asettaa paineita teknologialle, ja ongelmaksi muodostuu standardoinnin hitaus suhteessa teknologian kehitykseen. Kaikki uudet ratkaisut joudutaan jossakin määrin perustamaan vanhojen rakenteiden päälle koska yhteensopivuus alaspäin tulee säilyttää. On vaikeaa saada kaupalliset yritykset odottamaan standardin valmistumista juuri silloin kun innostus uusiin, kokeiluasteella oleviin ratkaisuihin olisi suurinta.

Onneksi näyttää siltä, että protokollien osalta vallitsee suhteellisen hyvä yhteisymmärrys siitä, mitä niistä tullaan tulevaisuudessa käyttämään. RTP ja RSVP ovatkin merkittäviä parannuksia juuri multimediaa ajatellen. Paljon ratkaisemattomia kysymyksiä liittyy vielä mm. siihen, kuka liikenteestä lopulta maksaa. Käytännössä kustannukset lopulta siirtyvät loppukäyttäjän maksettavaksi. Tarvitaan kuitenkin uusia menetelmiä, jotta avoimessa Internetissä voidaan seurata esimerkiksi siirrettyjä datamääriä käyttäjäkohtaisesti. Vaikka resursseja pystytäänkin varaamaan etukäteen, ovat ne kuitenkin aina rajallisia. Tuskin vielä pitkään aikaan päästään tilanteeseen, jossa kaistaa olisi niin paljon, että verkkoon ei koskaan muodostuisi ruuhkaa. Pidemmällä tähtäimellä tärkeintä onkin pyrkiä minimoimaan kustannuksia eli turhaa liikennettä mielummin kuin pyrkiä rakentamaan entistä leveämpiä siirtoteitä. Tässä kohdin suurin vastuu lankeaa juuri protokollien harteille.

Viitteet

BDS96 I. Busse, B. Deffner, and H. Schulzrinne. Dynamic qos control of multimedia applications based on rtp, computer communications, jan. 1996, January 1996. http://ftp.gaia.cs.umass.edu/pub/Buss9601:Dynamic.ps.gz.

BEB97 Robert Braden, Deborah Estrin, Steven Berson, Shai Herzog, and Daniel Zappala. The design of the rsvp protocol, May 1997. http://www.isi.edu/div7/rsvp/pub.html/rsvp1.final.reports.ps.

SCF96 Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 1889: RTP: A transport protocol for real-time applications, January 1996.

SRL98 H. Schulzrinne, A. Rao, and R. Lanphier. RFC 2326: Real time streaming protocol (RTSP), April 1998.

Tan96 Andrew S. Tanenbaum. *Computer Networks*. Prentice-Hall, Upper Saddle River, NJ 07458, USA, third edition, 1996.