Tuomas Virtanen

XMPP-pikaviestinprotokolla

Tietotekniikan
kandidaatintutkielma
19. tammikuuta 2012

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Tekijä: Tuomas Virtanen
Yhteystiedot: tuomas.virtanen@jyu.fi
Työn nimi: XMPP-pikaviestinprotokolla
Title in English: XMPP-Protocol
Työ: Tietotekniikan kandidaatintutkielma
Sivumäärä: 22
Tiivistelmä: XMPP on monipuolinen ja laajennettavissa oleva, avoimiin standardeihin perustuva pikaviestinprotokolla. Sitä voidaan käyttää paitsi puhtaiden tekstiviestien lähettämiseen ja olotilan tarkkailuun, myös esimerkiksi tiedostonsiirtoihin ja VoIP-puheluihin. Tässä tutkielmassa tutustutaan protokollan teknisiin ominaisuuksiin ja tehdään vertailuja muihin pikaviestinprotokolliin.
English abstract: XMPP is a versatile and extensible instant messaging protocol based on open standards. It can be used for eg. sending text messages and presence information, VoIP-calls and file transfers. This thesis examines the technical aspects of the protocol, and makes comparisons to other instant messaging protocols.
Avainsanat: XMPP, protokolla, pikaviestintä
Keywords: XMPP, Protocol, instant messaging

Sisältö

1 Johdanto

Pikaviestinnästä on tullut osa useiden tietokoneen, ja nykyään myös erilaisten mobiililaitteiden käyttäjien arkea. Pikaviestintää käytetään niin kotona, kuin myös koulussa [3] ja töissä [2]. Erilaisia pikaviestinsovelluksia on useita, ja viestintää voidaan harrastaa joko perinteisesti tekstiviestein, tai monipuolisemmin äänen ja videon avulla. WWW-ohjelmointitekniikoiden edistyessä pikaviestimiä löytyy jopa erilaisilta www-sivuilta, josta hyvänä esimerkkinä toimii mm. Facebook-palvelu [14].

Erilaisten pikaviestinsovellusten tullessa markkinoille, syntyy myös uusia viestintäprotokollia ja määritelmiä näiden sovellusten käyttöön. Tästä johtuukin nykymallin mukaisen pikaviestinnän suurin ongelma; eri pikaviestinverkkojen käyttöön tarvitaan usein eri sovellukset. Samalla käyttäjällä saattaakin olla käytössä useita ohjelmia, jotka toimivat omissa viestintäverkoissaan. Yrityksiä yhdistää eri pikaviestinverkkoja saman ohjelman käyttöön on olemassa, mutta kaikki näistä ovat eri syistä epätäydellisiä.

Usein ongelmana pikaviestintäprotokollien toteuttamisessa on se, että pikaviestinverkon protokolla on jollain tavalla suljettu. Protokolla voi olla suljettu esimerkiksi lisenssiehdoiltaan, joilla sen käyttöä rajoitetaan muiden kuin sille suunnitellun ohjelman käyttöön. Protokollan määritelmä voi olla suljettu ja itse protokolla voi olla salattu jollain salausalgoritmilla. Esimerkiksi MSN Messenger-ohjelman käyttämän MSNP-protokollan määritelmä ei ole virallisesti julkinen. Skype-pikaviestinohjelman käyttämä protokolla taas on täysin salattu [9] Rijndael-algoritmilla. Tällaisten ohjelmien käyttämiä protokollia ei voida toteuttaa muihin pikaviestimiin ilman protokollan purkua, joka esimerkiksi Skypen tapauksessa on toistaiseksi ollut vain osittain menestyksekästä [10].

Extensible Messaging and Presence Protocol (XMPP) pyrkii olemaan mahdollisimman avoin ja laajennettavissa oleva, yleiskäyttöinen protokolla kaikkeen pikaviestintään [1]. Se hallitsee laajennuksia käyttäen mm. tekstiviestit, VoIP-puhelut sekä tiedostojen siirron käyttäjien välillä. XMPP-verkko sallii myös esimerkiksi siltaukset muihin pikaviestinverkkoihin, kuten Internet Relay Chat (IRC). XMPP-protokollaa käyttääkin nykyään jo moni tunnettu yritys sovelluksissaan, kuten mm. Nokia (Nokia Ovi), Google (Google Talk, Google Wave), LiveJournal (LJ Talk). Myös muunmuassa Facebook paljastaa XMPP-pohjaisen rajapinnan palvelun ulkopuolisille käyttäjille [14].

Kandidaatintutkielma tarkastelee lyhyesti pikaviestintää yleisesti, ja keskittyy erityisesti XMPP-protokollaan. Luvussa 2 käsitellään pikaviestintää yleisesti, ja esitellään muutamia sen sovelluksia sekä protokollia. Luvussa 3 esitellään XMPP-protokollaa ja sen rakennetta. Luvussa 4 perehdytään XMPP-verkkoihin ja niiden eroihin muihin pikaviestinverkkoihin nähden. Luvussa 5 esitellään XMPP-protokollan pikaviestintään tarjoamia ominaisuuksia, kuten tekstiviestejä ja olotilan hallintaa.

2 Pikaviestintä ja sen protokollat

2.1 Pikaviestinnän käsitteitä

Pikaviestinnässä on Wikipedian määritelmän [8] mukaan kyse (lähes) reaaliaikaisesta, tekstipohjaisesta viestinnästä kahden tai useamman ihmisen välillä tietokoneiden tai muiden laitteiden välityksellä. Tiedon siirto tapahtuu Internetin tai muun verkon yli. Pikaviestintä voi monipuolisimmillaan olla myös videon tai äänen välityksellä tapahtuvaa viestintää. Pikaviestintään on kehitelty useita standardeja ja protokollia, kuten XMPP.

Protokolla määrittelee toimenpiteet ja rakenteet, joita käytetään tiedon siirtämiseen, tallentamiseen ja lataamiseen. *Tietoliikenneprotokollassa* pääasiana ovat tavat joilla viestinnän eri tahot siirtävät tietoa toisilleen, toimivat virhetilanteissa, autentikoituvat ja salaavat siirrettävän tiedon. *Autentikoinnilla* tarkoitetaan käyttäjän todentamista hänen yrittäessä päästä suojattuihin resursseihin. Tiedon *salauksessa* taas on kyse tiedon muuttamisesta jollain algorimilla sellaiseen muotoon, että vain tarkoitettu vastaanottaja pystyy sen lukemaan selkokielisenä. Tiedon salaukseen on useita standardeja, joista esimerkkinä suosittu TLS [7].

2.2 Sovellutukset

Pikaviestintää käytetään nykyään moneen eri tarkoitukseen. Tunnetuinta pikaviestintä on luultavasti nuorison keskuudessa, jotka ovat jo pitkään käyttäneet erilaisia ohjelmia sosiaalisten ryhmiensä väliseen viestintään. Nuorison keskuudessa suosituimpia sovelluksia nykyään ovat muunmuassa facebook, MSN Messenger, Skype sekä puhelimilla tapahtuva tekstiviestintä. Myös muita ohjelmia käytetään.

Työpaikoilla pikaviestintää käytetään useing projektien koordinointiin eri työryhmien välillä sekä viestintään asiakkaiden kanssa. Pikaviestinnän tärkeimpiä osa-alueita näissä käyttötarkoituksissa ovatkin esimerkiksi puhe- sekä videopohjainen pikaviestintä, sekä ryhmässä että kahdenkeskisesti. Koulumaailmassa pikaviestintää käytetään useimmiten opettajien ja oppilaiden sekä vanhempien väliseen viestintään. Myös kursseja tai tunteja saatetaan järjestää etänä käyttäen apuna ääni- video- ja tekstipohjaista pikaviestintää.

2.3 Muita protokollia

Tässä tutkielmassa tarkasteltavan XMPP-protokollan lisäksi on olemassa myös muita pikaviestinprotokollia, kuten muunmuassa *MSNP*, *Skype*, *IRC*, *SIMPLE* ja useita muita

vähemmän käytettyjä. Näistä SIMPLE suunniteltiin myös mahdollisimman yleiskäyttöiseksi ja avoimeksi, ja on käytössä esimerkiksi joissain VoIP-sovelluksissa.

Skype-protokolla on täysin suljettu ja salattu [9]. Protokollasta on onnistuttu selvittämään osia, mutta ainakin toistaiseksi sen toiminta on suurilta osin tuntematon. Skype-verkko on rakenteeltaan hajautettu, mutta verkkoon kirjautuminen pitää tehdä erillisen Skype Limited-yhtiön kirjautumispalvelimen kautta.

MSNP (Microsoft Notification Protocol) on Microsoft-yhtiön protokolla pikaviestintään. Sitä käytetään muunmuassa yhtiön omissa Windows Messenger, MSN Messenger ja Windows Live Messenger-sovelluksissa. Myös esimerkiksi avoimeen lähdekoodiin perustuvat pikaviestinsovellukset Pidgin ja Trillian taitavat viestinnän MSNP-protokollaa käyttäen. MSNP-verkko on rakenteeltaan keskitetty [11], ja kirjautumispalvelimet ovat Microsoftin hallinnassa. Microsoft ei julkaise protokollan määritelmää, vaan jokainen protokollan versio on käyttäjien toimesta purettu ja määritelmät julkaistu epävirallisesti. Tällä hetkellä MSNP:n viimeisin julkaistu versio on 19, vaikka kirjautumispalvelimet tukevatkin kaikkia protokollia versiosta 8 lähtien.

IRC (Internet Relay Chat) on vuonna 1988 Suomessa kehitetty tekstipohjainen pikaviestinprotokolla. Protokolla on avoin, ja siitä on useita versioita sekä toteutuksia. Rakenteeltaan IRC on hajautettu. Palvelimet voidaan yhdistää kokonaisuuksiksi eli keskusteluverkoiksi. Eräitä suosittuja keskusteluverkkoja on muunmuassa IRCNet ja Quakenet. Keskusteluverkossa voi olla useita keskustelukanavia, joille käyttäjä voi liittyä [12]. Käyttäjillä voi olla keskustelukanavilla erilaisia oikeuksia, joista esimerkkinä toisten käyttäjien poistaminen kanavalta sekä kanavan aiheen vaihtaminen.

3 XMPP-Protokolla

Extensible Messaging and Presence Protocol (XMPP) on avoin, reaaliaikaiseen pikaviestintään ja olotilan julistamiseen tarkoitettu XML-pohjainen standardi. Protokollaa kehittiin alun perin nimellä Jabber, ja se oli tarkoitettu saman nimiseen, vapaaseen lähdekoodiin perustuvaan pikaviestinohjelmaan. Vuonna 2002 perustettiin XMPP työryhmä (XMPP WG), jonka tehtäväksi tuli Jabber-protokollaan perustuvan pikaviestinprotokollan kehitys. Uuden XMPP-nimisen protokollan tarkoitus oli soveltua Internet Engineering Task Forcen (IETF) tukemaksi protokollaksi pikaviestintään (Instant Messaging) ja tilan (Presence status) hallintaan. Kehitystyön tuloksena oli lopulta dokumentti, jossa määriteltiin protokollan perusominaisuudet ja laajennukset pikaviestintään. Nykyään tätä perustoiminnallisuutta määritellään RFC-dokumenteissa 3920 ja 3921. Myöhemmin protokollaan on luotu laajennuksia esimerkiksi tiedostojen siirtämiseen sekä VoIP-puheluihin.

3.1 Versiot ja ydinstandardi

XMPP-protokollan kehityksen voidaan katsoa alkaneen vuonna 2000, jolloin julkaistiin XMPP:n edeltäjän, IMPP:n suunnitelma avoimeksi pikaviestintäprotokollaksi. Tämän jälkeen julkistettiin seuraavina vuosina useita suunnitelmadokumentteja muunmuassa salauksen soveltamisesta pikaviestintään, merkkijonojen kuljettamisesta XML-virroissa sekä olotilan julkistamisesta.

Protokollan ensimmäinen virallinen määritelmä julkaistiin vuonna 2004 RFC-dokumenteissa 3920-3923 [4] [5] [6]. Seuraavien vuosien aikana kehiteltiin useita laajennuksia mm. puheen ja videokuvan kuljettamiseen XMPP-verkossa, kunnes vuonna 2011 julkaistiin protokollan seuraava versio RFC-dokumenteissa 6120-6122. Uusimmat määritelmädokumentit ovat käytännössä vielä tarkastelun alla. Tässä dokumentissa käsitelläänkin protokollaa vielä vanhemman, käytetymmän ja tuetumman version pohjalta.

XMPP-protokollan ydinmääritelmä on hyvin yksinkertainen. Ytimessä ei käsitellä esimerkiksi pikaviestintää tai tilanhallintaa. Sen sijaan pikaviestintä, tilanhallinta, puheviestintä sekä esimerkiksi videon ja tiedostojen välitys ovat laajennuksia. Laajennukset määritellään tarkemmin XEP-koodilla varustetuissa XMPP-organisaation sisäisissä määrittelydokumenteissa.

Laajennusten määritelmädokumentit käyvät aina läpi tarkan seulonnan. Ne laajennukset jotka palvelevat hyvin asiaansa, voivat myöhemmin päästä mukaan itse protokollan ydinmääritelmään. Suurin osa laajennuksista on tällä hetkellä kuitenkin vielä suunnitelma- tai testausasteella, ja onpa olemassa kaksi virallista, lähinnä huumorin kannalta kehitettyä laajennustakin. Asiakasohjelmien ei teoriassa ole pakko tukea kaik-

kia laajennuksia, mutta käytännössä asiakasohjelmat odottavat että ainakin tärkeimmät toimivat.

Muutamia huomionarvoisia XMPP-protokollan laajennuksia:

- XEP-0001: XMPP Extension Protocols. Määrittelee muiden laajennusten vaatimukset.
- XEP-0095: Stream Initiation. Laajentaa xml-streamien käyttötapoja, jotta niissä voidaan siirtää mm. tiedostoja.
- XEP-0096: SI File Transfer. Tiedostonsiirto-ominaisuudet protokollaan lisäävä laajennus. Rakentuu pitkälti edellisen laajennuksen lisäysten päälle.
- XEP-0166: Jingle. Laajennus videon, äänen ja muun datan siirtoa varten.

3.2 XML-tietovirrat

XMPP-standardissa liikenne kuljetetaan yleensä XML-tietovirrassa. XML-tietovirtaa voidaan ajatella säiliönä, jota käytetään XML-elementtien kuljettamiseen verkkoentiteettien välillä. Virran alku ilmoitetaan aina XML-elementillä *stream*, jossa voidaan määritellä attribuutteina myös esimerkiksi käytettävä nimiavaruus ja nerkistö. Niin kauan kuin XML-virta on olemassa, sen luonut entiteetti voi lähettää määrittämättömän määrän XML-elementtejä vastaanottajalle. XML-tietovirta on aina yksisuuntainen; mikäli vastaanottava pää haluaa lähettää viestejä, pitää sen tätä varten erikseen neuvotella erillinen virta (Response Stream).

XML-tietovirrassa voidaan lähettää XMPP-standardin mukaan lähettää joko XML-säkeistöjä tai yhteyden neuvottelemiseen tarvittavia XML-elementtejä. Mikäli entiteetti saa vääränmuotoisen paketin, se hylätään. Yhteyden ja virran neuvottelee yleensä asiakas- tai palvelinentiteetti vastaanottavaan entiteettiin, joka on yleensä palvelin. Asiakas-palvelin yhteyksien lisäksi palvelin-palvelin yhteydet ovat myös mahdollisia.

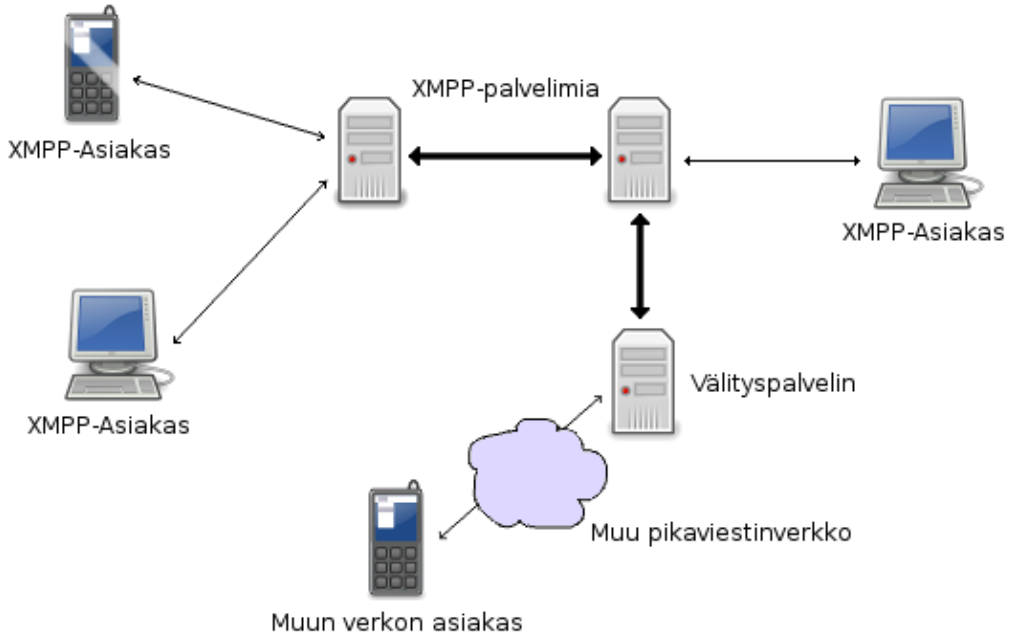
3.3 Säkeistöt

XMPP-protokollassa säkeistö (Stanza) on xml-elementti sisältöineen, joka voidaan lähettää entiteetiltä toiselle XML-virran yli. Säkeistö kuljetetaan aina suoraan XML-virran *stream*-juurielementin alla, ja voi olla nimeltään *message*, *presence* tai *iq*. Säkeistöjen nimiavaruuden on oltava *jabber:client* tai *jabber:server*. Mikään muu tietovirrassa lähetetty elementti ei ole säkeistö. Säkeistö sisältää yleensä yhden tai useampia lapsielementtejä.

Säkeistöllä on kolme tyyppiä, joilla määritetään viestintämekanismi. *Push*-mekanismi on tarkoitettu yleiseen viestintään, *publish-subscribe* [16] yleiseen palvelimen ja verkon tilan viestittämiseen asiakkaille, ja *request-response* taas tiedon vaihtamiseen entiteettien kesken. *Push*-mekanismissa kyse on siitä, että entiteetti työntää toiselle entiteetille tietoa, eikä välttämättä odota vastausta. Tästä esimerkkinä on esimerkiksi *message*-elementti. *Publish-subscribe* -mekanismissa tilaava entiteetti ilmoittaa julkaisevalle entiteetille haluavansa tietoja tietyntyyppisistä asioista eli tekee tilauksen, ja julkaiseva entiteetti lähettää tällaisten asioiden ilmestymisestä tiedon tilaajalle. *Request-response* -tyyppisessä viestinnässä entiteetti pyytää tietoa toiselta entiteetiltä. Esimerkiksi Iq-paketti on tyypiltään tällainen.

4 Verkko

XMPP-verkko on hajautettu, ja palvelimia voi olla useita. Palvelimet voivat olla yhteydessä toisiinsa suoraan, tai välipalvelimien välityksellä. Asiakkaat voivat halutessaan ottaa yhteyden mihin tahansa verkon palvelimista. Kuvaajassa 1 kuvataan verkon rakennetta.



Kuva 1: XMPP-verkon rakenne

4.1 Palvelin

Vaikka XMPP:n määritelmässä ei varsinaisesti määritellä miten viestejä lähetetään, käytetään protokollaa yleensä asiakas-palvelin-arkkitehtuurilla. Erona tässä useisiin muihin tunnettuihin pikaviestinprotokolliin, kuten OSCAR ja MSNP verrattuna on kuitenkin se, että XMPP-verkolla ei ole yhtä keskitettyä autentikaatiopalvelinta. Käytännössä siis kuka tahansa voi ajaa omaa palvelintaan, ja liittää sen osaksi suurempaa verkkoa.

Palvelimen tehtävänä XMPP-verkossa on TCP-yhteyksien vastaanotto sekä liikenteen välittäminen muille autentikoiduille käyttäjille, palvelimille ja muille kohteille. Palvelin myös reitittää asiakkaan lähettämät paketit oikeisiin kohteisiin. Palvelimet voivat myös tallentaa käyttäjäkohtaisia tietoja, kuten kontaktilistan, joka voi sisältää muita käyttäjiä mistä tahansa liitetystä XMPP-verkosta.

4.2 Asiakas

Asiakas luo yhteyden palvelimeen TCP-protokollan yli tiettyyn palvelimen porttiin. Internet Assigned Numbers Authority (IANA) on määritellyt XMPP-protokollan käyttöön portin 5222, mutta muitakin voi olla käytössä. Asiakkaan tehtävänä on lähettää viestejä ja tilatietoja palvelimelle, sekä vastaanottaa niitä palvelimelta. Asiakkaan on osattava vähintään avata yhteys XML-viestien lähettämistä ja vastaanottamista varten.

XMPP-verkossa jokaista asiakasta tai päätepistettä kutsutaan entiteetiksi, jolla on aina oma tunnisteensa nimeltään *JID*. Tunniste on kolmiosainen, esimerkiksi muotoa *solmu@toimialue/resurssi*. Tunnisteessa toimialue-kenttä määrittelee käytetyn palvelimen osoitteen, solmu-kenttä käyttäjän nimen ja resurssi-kenttä käyttäjälle kuuluvan toisen asiakaslaitteen. Protokolla salliikin useampien asiakasohjelmien kirjautumisen samalle palvelimelle samalla käyttäjätunnuksella, kunhan eri ohjelmille määritellään oma kotiosoitteensa.

Entiteeteille voidaan määrittää myös tärkeystasoja, jolloin esimerkiksi osoitteeseen *solmu@toimialue/matkapuhelin* lähetetty viesti menee käyttäjän matkapuhelimeen, mutta osoitteeseen *solmu@toimialue* lähetetty viesti ohjautuu tärkeimmäksi määriteltyyn asiakasohjelmaan tai laitteeseen.

4.3 Välipalvelin

Koska XMPP-sallii useampien palvelinten toimimisen yhdessä, voidaan XMPP-verkkoa laajentaa niinsanotuilla välipalvelimilla (Gateway). Välipalvelimen tehtävänä on muuntaa XMPP-viestejä sopivaksi välipalvelimen toisella puolella toimivaan pikaviestinverkkoon, ja toisaalta toisesta verkosta tulevia viestejä XMPP-verkkoon sopiviksi. Tunnettuja välipalvelimia on olemassa esimerkiksi SMTP, Internet Relay Chat, SIMPLE sekä SMS -verkkoja varten.

Suurimpana ongelmana välipalvelinten käytössä on se, että protokollat eivät aina ole täysin yhteensopivia. Tästä seuraa, että osa viesteissä välitetystä tiedosta ei välttämättä ole esitettävissä toisella protokollalla. Ongelmia tulee myös turvallisuuden kanssa, sillä lähetetyn viestin tai muun tiedon siirtyessä toiseen pikaviestinverkkoon, sen turvallisuutta ei voida enää taata.

4.4 Yhteyden suojaus

XMPP-verkon kaikkien asiakkaiden ja palvelinten on tuettava TLS- ja SASL-metodeja tiedon salaamiseen. Salauksen käyttö ei ole protokollan määritelmädokumentissa vaa-

dittua, mutta sitä suositellaan vahvasti. Myös palvelinten välisten tietoliikenneyhteyksien välinen salaus on suositeltua, muttei vaadittua.

XMPP-verkossa määritellään myös palvelinten välinen takaisinsoitto (Server dialback), jota voidaan käyttää varmistamaan että palvelimeen yhdistävä toinen palvelin on olemassa. Protokollan määritelmädokumenteissa takaisinsoitto määritelläänkin hyvin heikoksi suojaukseksi, ja sen käyttöä ei enää suositella. Palvelinten ei odoteta enää tukevan kyseistä metodia.

4.5 Asiakkaan autentikointi

XMPP-protokollan mukainen yhteys asiakas- ja palvelinsovelluksen välillä aloitetaan luomalla XML-virta. Ensimmäiseksi virrassa suoritetaan mahdollinen salauksen neuvottelu sopivin xml-säkein. Salauksen epäonnistuessa palvelinsovellus voi vastata erilaisilla virheilmoituksilla [4].

Kun haluttu yhteyden salauksen neuvottelu on suoritettu, aloitetaan virallinen yhteys lähettämällä protokollamääritelmän mukainen stream-elementti. Palvelin vastaa elementtiin omalla *stream*-elementillään sekä *stream:features*-elementillä, jossa määritellään autentikointiin käytettävät suojausmetodit. Tavallisesti käytässä on ainakin MD5-tiivistealgoritmi sekä puhdas teksti. Asiakasohjelma valitsee suojausmetodin, ja ilmoittaa siitä *auth*-elementissä palvelimelle.

Kun autentikaatiometodin valinta on suoritettu onnistuneesti, aloitetaan itse autentikaatio. Kaikki autentikaatiopaketit kuljetetaan asiakkaan ja palvelimen välillä *response* ja *challenge*-elementeissä base64-enkoodattuna. Virhetilanteessa palvelin voi lähettää myös *failure*-elementin virhetietojen kera. Lähetettävät elementit riippuvat valitusta autentikaatiometodista. Kun asiakas on vastannut oikein kaikkiin palvelimen lähettämiin haasteisiin, saa asiakas palvelimelta *success*-elementin. Tämän jälkeen asiakas neuvottelee palvelimenkanssa uuden yhteyden *stream*-elementeillä, ja palvelin lähettää asiakkaalle listan palvelimen osaamista toiminnoista *stream:features*-elementissä. Näitä voi olla mm. sessionhallinta, pikaviestintä ja resurssien sidonta.

4.6 Sessio ja resurssin sidonta

Kun yhteys on luotu ja autentikointi on suoritettu, suoritetaan käyttäjän resurssin sidonta palvelimelle. Sidonta on tarpeellista, mikäli palvelin aiemmassa vaiheessa ilmoitti sen ominaisuutena asiakkaalle. Sidonnan tarkoituksena on yhdistää asiakkaan resurssi, esim. *kotikone* tai *kannettava* palvelimella toimivaksi JID-tunnisteeksi. Tätä varten asiakas lähettää heti autentikoinnin onnistuttua palvelimelle iq-säkeen, jossa lähetetään sen sisäisessä *resource*-elementissä haluttu resurssi. Asiakas voi antaa myös

palvelimen suorittaa resurssinimen valinnan automaattisesti jättämällä kentän ryhjäksi. Palvelimen ei myöskään ole pakko hyväksyä haluttua resurssia, vaan voi vaihtaa sen toiseen. Resurssin sidonnan onnistuttua palvelin lähettää asiakkaalle iq-säkeen, jonka sisäisessä *jid*-elementissä on asiakkaan täysi JID-tunniste. Tätä tunnistetta käytetään asiakkaan tunnistamiseen palvelimella loppuyhteyden aikana.

Mikäli palvelin mainostaa asiakkaalle *stream:features*-elementissä sessio-ominaisuutta, ja mikäli asiakasohjelma haluaa käyttää palvelimen pikaviestintä- ja tilanhallintaominaisuuksia, on asiakkaan neuvoteltava palvelimen kanssa sessio. Tämä tehdään lähettämällä palvelimelle sopiva iq-säe. Palvelin vastaa tähän joko onnistumista merkkaavalla tyhjällä iq-säkeellä, tai *error*-elementillä varustetulla virhettä merkkaavalla iq-säkeellä. Mikäli session luonti epäonnistuu, voi se olla merkki siitä, että asiakkaalla ei ole oikeuksia pikaviestintään palvelimella. Kun session luonti on suoritettu, sanotaan resurssin XMPP-terminologiassa olevan aktiivinen.

4.7 Tekstiviestien välitys

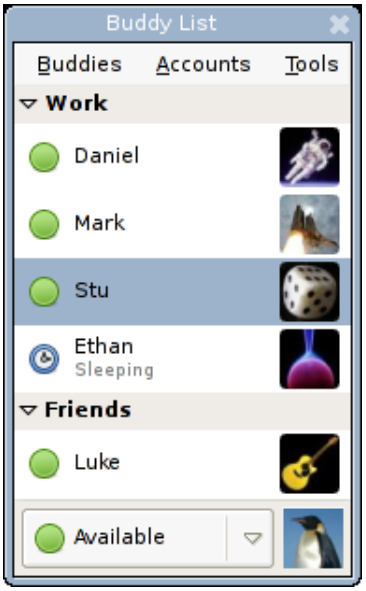
Viestien lähetystä varten asiakkaalla on oltava täysi JID. Viestipakettina käytetään message-säettä, jolla on oltava aina vähintään *to*-parametri viestin vastaanottajan JID:tä varten. Vastaanottajan JID voi sisältää resurssin, mikäli viesti on suunnattu vastaanottajan tietylle resurssille. Mikäli resurssia ei määritetä, suuntautuu viesti vastaanottajan kaikkiin resursseihin. Elementissä voidaan määritellä myös *from*-parametri, johon sijoitetaan viestin lähettäjän JID. Mikäli lähettäjän JID sisältää resurssinimen, suunnataan takaisin tulevat viestit tähän resurssiin.

Message-säe sisältää aina *body*-elementin, jossa on kuljetettava viesti. Viestin pituudelle ei teoriassa ole määritetty rajaa, mutta asiakasohjelmat saattavat itse lyhentää viestiä. Säe voi sisältää myös *thread*-elementin, jolla ilmoitetaan keskustelu, johon viesti liittyy. Mikäli viestiin tai keskusteluun halutaan määrittää aihe, se voidaan lähettää *subject*-elementissä. Kun säe on rakennettu, se voidaan lähettää palvelimelle, ja palvelin hoitaa säkeen siirron oikealle vastaanottajalle.

4.8 Ystävälistan hallinta

Ystävälista pitää sisällään tiedot käyttäjän listalle lisäämistä JID-kontakteista. Ystävälistan säilytystä hallitsee palvelin, ja asiakas voi pyytää sen erikseen palvelimelta. Ystävälistalle voidaan lisätä käyttäjiä vain, mikäli lisäyksen kohde sallii sen. Ystäväpyyntöä kutsutaan XMPP-terminologiassa tilaukseksi (subscription). Asiakas suorittaa tilauksen lähettämällä palvelimelle sisällöttömän presence-säkeen, jonka *type*-parametrina on *subscribe* ja *to*-parametrina lisäyksen kohteen JID ilman resurssia. Vastaanottaja vas-

taa tähän lähettämällä vastaavan paketin, jonka *type*-parametrina on joko *subscribed* (tilattu) tai *unsubscribed* (ei tilattu).



Kuva 2: Tyypillinen kontaktilista pikaviestinohjelmassa

4.9 Tilatiedot

Asiakas voi lähettää ystävälistallaan oleville käyttäjille tietoja omasta tilastaan, esimerkiksi "poissa" tai "älä häiritse". Asiakas lähettää tilanvaihtopyynnön palvelimelle, joka hoitaa sen eteenpäin ystävälistan kaikille kontakteille. Mikäli asiakas ei viestitä tilaa palvelimelle, voi palvelin myös erikseen pyytää sitä. Tilamuutokset lähetetään aina kaikkien kontaktien kaikille resursseille.

Tilanmuutospyyntö tehdään lähettämällä palvelimelle presence-säe. Tila ilmoitetaan säkeessä elementissä *show*. Asiakas voi myös lähettää lisätietoja tilasta *status*-elementissä. Lisätietoelementissä voi olla pitempikin teksti, mutta tilaelementin sisältö on yleensä lyhyt.

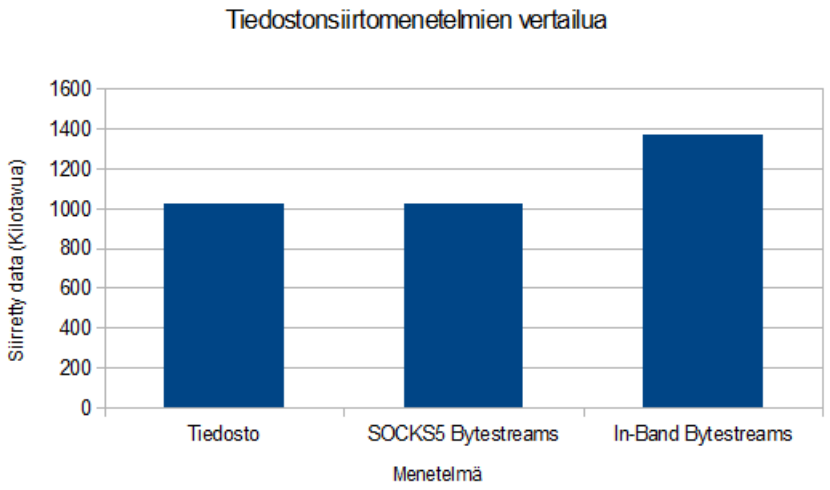
4.10 Tiedostonsiirto

XMPP-protokollan laajennuksissa määritellään monta erilaista tapaa siirtää vapaavalintaista binääristä tietoa yhteyden yli. Näistä vanhimmat, In-Band Bytestreams (IBB) [15], Out of Band Data (OOB) [17] sekä SOCKS5 Bytestreams [20] laajennukset tarjoavat tavan siirtää tietoa joko erillisen yhteyden tai jo muodostetun XML-virran yli.

OOB-menetelmässä asiakas ilmoittaa vastaanottajalle, että määritetystä paikasta on saatavilla tiedosto, joka vastaanottajan pitäisi hakea. Ongelmana tässä menetelmässä on se, että vastaanottajalle ei juurikaan tarjota tietoa tiedostosta, ja yhteyden tielle tulevat palomuurit tekevät yhteyden luomisen usein mahdottomaksi.

SOCKS5 Bytestreams-laajennus määrittelee laajemman tavan siirtää tietoa asiakkaiden välillä. Siirto voi olla joko asiakkaiden välinen (Peer to Peer) tai välitetty (Mediated). Välitetyn tiedonsiirron etuna onkin se, että sekä lähettäjä että vastaanottaja voivat välttää palomuurit luomalla yhteyden toistensa sijaan välityspalvelimeen.

IBB-menetelmässä tieto siirretään pienissä paketeissa Base64-menetelmällä enkoodattuna olemassaolevan xml-virran yli vastaanottajalle. Tämän menetelmän haittana onkin huono hyötysuhde sekä mahdollisesti välissä olevat palvelimet, jotka saattavat rajoittaa yhteyden nopeutta. Kuvassa 3 kuvataan IBB-menetelmän ja SOCKS5-menetelmän siirretyn tiedon määrää käytettäessä yhden megatavun hyötykuormaa. IBB-menetelmä sopii kuitenkin hyvin tapauksiin, joissa muut menetelmät eivät toimi esimerkiksi palomuurien tai muiden esteiden takia.



Kuva 3: Siirretyn tiedon kokonaismäärä tiedostonsiirrossa

Siinä missä OOB ja IBB-menetelmät ovat yleisiä tiedon siirtoon suunniteltuja menetelmiä, tarjoaa uudempi SI File Transfer-laajennus [18] nimenomaan käyttäjien tiedostojen siirtoon tarkoitetun menetelmän. SI File Transfer hyödyntää pääasiallisesti SOCKS5 Bytestreams-laajennusta, mutta voi myös tarvittaessa käyttää IBB-siirtoja. Laajennus ei tuokaan lisää tapoja siirtää tietoja, vaan määrittää vain siirron aloituksen, lopetuksen sekä toiminnan eri virhetilanteissa.

Uusin laajennus tiedostonsiirtoon XMPP-protokollassa on Jingle File Transfer [19]. Jingle File Transfer tarjoaa SI File Transfer-laajennuksesta parennetun tuen kaksisuun-

taiselle viestinnälle tiedostonsiirron aikana sekä paremman toteutuksen palomuurien ja NAT-ratkaisujen kiertämiseen. Jingle File Transfer on vielä suhteellisen tuore laajennus, mutta sen odotetaan tulevaisuudessa käytännössä korvaavan SI File Transfer-laajennuksen.

5 XMPP-protokollan edut ja heikkoudet

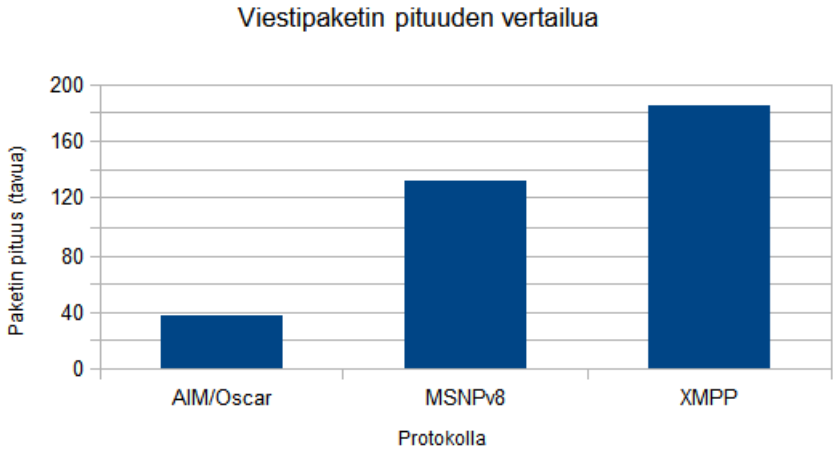
5.1 Avoimuus

XMPP-protokolla on täysin avoin, ja kuka tahansa voi käyttää sitä ilman maksuja millekään taholle. Monet palvelut käyttävätkin protokollaa jo suoraan tai tarjoavat siltapalvelimen johon XMPP-verkon asiakkaat voivat liittyä. Siltapalvelimia tarjoavista palveluista löytyy hyvänä esimerkkinä Facebook [14].

Suhteessa esimerkiksi MSNP- tai OSCAR-protokolliin, on XMPP helpompi ottaa käyttöön omassa sovelluksessa. Sen koko dokumentaatio on julkinen, ja kehitysprosessi on täysin näkyvä kaikille asiasta kiinnostuneille. Suljetut, niin sanotut yhden yhtiön protokollat kuten Skypen protokolla, MSNP ja OSCAR ovat vain osittain julkisesti dokumentoituja, ja kaikissa tapauksissa dokumentaatio ei ole protokollan kehittäneen yhtiön julkaisemaa. Lisäksi ei ole mitään takeita siitä, että protokollan määritelmä pysyy samanlaisena aina, eikä sen kehitykseen useinkaan voi vaikuttaa kehittäjäyhtiön ulkopuolelta. Esimerkiksi MSNP- ja Skype-protokollat päivittyvätkin jatkuvasti. Päivitysten syynä on usein ominaisuuksien lisäämisen lisäksi protokollan purkajien edellä pysyminen.

5.2 Hyötysuhde

XML-pohjaisten protokollien heikkoutena on pakettien suuri koko ja suhteellisen pieni hyötysuhde. XMPP ei ole poikkeus. Viestipakettien koko voi pahimmillaan olla moninkertainen itse paketissa lähetettävään tietoon. Kuvassa 4 kuvataan standardin viestipaketin kokoa, kun viestin koko on kolme merkkiä ja paketissa on protokollan kannalta vain kaikkein olennaisin tieto.



Kuva 4: Viestipaketin koko eri protokollissa

Vaikka XMPP-protokollan viestielementtien koko saattaakin olla kohtalaisen suuri, ei se välttämättä ole ongelma nykyaikaisilla Internet-yhteyksillä. Esimerkiksi binääritiedon siirtoon protokollasta löytyy hyvälläkin hyötysuhteella varustettuja menetelmiä. Hitaammilla yhteyksillä (esimerkiksi heikkolaatuinen GPRS), saattaa esimerkiksi siirtojen vasteajassa näkyä eroja parempiin yhteyksiin verrattuna.

5.3 Suojaus

TLS (Transport Layer Security) on salausprotokolla, jolla voidaan salata Internet-sovellusten tietoliikenne, joka määritellään dokumentissa RFC5246 [7]. Aiemmin TLS tunnettiin nimellä SSL (Secure Sockets Layer). TLS onkin paranneltu versio SSL-protokollasta. Yleisimpiä käyttötarkoituksia TLS:lle on muunmuassa WWW-sivujen suojattu siirto HTTPS-protokollalla.

Tiedon salaus on nykyään tärkeässä osassa erilaisessa viestinnässä. XMPP-protokolla sisältääkin jo määritelmässään tuen yhteyden salaukseen TLS-protokollan yli [4]. Joihinkin vanhempiin protokolliin salausmenetelmät on lisätty jälkikäteen, toisiin taas ei. Esimerkiksi yritysten sisäisessä viestinnässä tiedon salaus on usein hyvinkin tärkeää, mikäli verkossa siirretään arkaluontoista tietoa. Taulukossa 1 luetellaan muutamien protokollien tilanne yhteyden salaukselle.

Protokolla	Tuki salaukselle
IRC	Kyllä, tuki riippuu palvelimesta
MSNP	Ei
OSCAR	Kyllä, TLS
Skype	Kyllä, patentoitu
XMPP	Kyllä, TLS

Taulukko 1: Eräiden protokollien tuki yhteyden salaukselle.

6 Yhteenveto

Pikaviestintä on nykyään arkea jo työpaikoilla, kouluissa ja muussakin arkielämässä. Erilaisten sovellusten myötä vaihtoehtoja pikaviestintään on useita. Vaikka onkin hyvä että jokaiselle löytyy varmasti sopiva sovellus, on ongelmana kuitenkin se, että useimmiten eri sovellukset käyttävät eri verkkoja viestinnän välittämiseen, joista useat ovat tavalla tai toisella aidattuja ("Walled garden"). Niinpä käyttäjät usein tarvitsevatkin useita ohjelmia viestiäkseen kaikkien eri verkkoja käyttävien tuttujensa kanssa.

XMPP on eräs ratkaisu verkkojen määrään. Sen tarkoituksena on luoda yksi tarkkaan määritelty ja vapaasti käytettävissä oleva standardi, jota kuka tahansa voi käyttää ja joka soveltuu useimmille pikaviestinnän osa-alueille. Vaikka protokollaa ei nykyään vielä käytetäkään suurimmissa pikaviestinohjelmista, leviää se silti pienempien palveluiden kautta ja monien suurempien yritysten tuella eteenpäin. Protokollan suurin hyöty onkin sen avoin dokumentaatio, hyvin suunniteltu määritelmä, yhteensopivuus olemassaolevien järjestelmien kanssa välipalvelinjärjestelmää soveltamalla sekä monet eri lisenssein saatavilla olevat ohjelmistototeutukset protokollasta.

Koska XMPP on alusta lähtien suunniteltu laajennettavaksi, on se sovitettavissa hyvinkin erilaisiin tarkoituksiin. Protokolla onkin jo saanut useita laajennuksia. Protokollan ydinmääritelmästä taas on juuri julkaistu uusi versio, johon on sisällytetty monia tärkeimmistä laajennuksista. XMPP onkin avoimuuden, jatkuvan kehityksen ja parantuvan ohjelmatuen ansiosta harkinnanarvoinen ratkaisu uusiin pikaviestinsovelluksiin.

Lähteet

[1] The XMPP Standards foundation, *"About The Extensible Messaging and Presence Protocol"*, Saatavilla osoitteesta <URL: `http://xmpp.org/about-xmpp/`>, viitattu 17.1.2012.

[2] Ann Frances Cameron, Jane Webster, *"Unintended consequences of emerging communication technologies: Instant Messaging in the workplace"*, Saatavilla osoitteesta <URL: `http://www.sciencedirect.com/science/article/pii/S0747563203001109`>, School of Business, Queenś University, 24.1.2004, Kanada.

[3] Andrew J. Flanagin, *"IM Online: Instant Messaging Use Among College Students"*, Saatavilla osoitteesta <URL: `http://www.tandfonline.com/doi/abs/10.1080/00036810500206966`>, University of California, 16.8.2006, Yhdysvallat.

[4] P. Saint-Andre, *"Extensible Messaging and Presence Protocol (XMPP): Core"*, Saatavilla osoitteesta <URL: `http://www.xmpp.org/rfcs/rfc3920.html`>, lokakuu 2004, RFC-3920.

[5] P. Saint-Andre, *"Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence"*, Saatavilla osoitteesta <URL: `http://www.xmpp.org/rfcs/rfc3921.html`>, lokakuu 2004, RFC-3921.

[6] P. Saint-Andre, *"Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM)"*, Saatavilla osoitteesta <URL: `http://tools.ietf.org/html/rfc3922`>, lokakuu 2004, RFC-3922.

[7] T. Dierks, E. Rescorla, *"The Transport Layer Security (TLS) Protocol"*, Saatavilla osoitteesta <URL: `http://tools.ietf.org/html/rfc5246`>, elokuu 2008.

[8] Wikipedia, *"Instant messaging"*, Saatavilla osoitteesta <URL: `http://en.wikipedia.org/wiki/Instant_messaging`>, viitattu 27.11.2010.

[9] Salman A. Baset, Henning Schulzrinne, *"An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol"*, Saatavilla osoitteesta <URL: `http://arxiv.org/pdf/cs/0412017`>, Columbia University, New York, Yhdysvallat, 15.9.2004.

[10] Matthew Humphries, *"The Skype protocol has been reverse engineered "*, Saatavilla osoitteesta <URL: http://www.geek.com/articles/news/the-skype-protocol-has-been-reverse-engineered-2011062/>, viitattu 17.1.2012.

[11] MSNPiki, *"MSN Protocol Version 8"*, Saatavilla osoitteesta <URL: http://msnpiki.msnfanatic.com/index.php/Main_Page>, viitattu 10.1.2011.

[12] Jarkko Oikarinen, Darren Reed, *"RFC1459 - Internet Relay Chat Protocol"*, Saatavilla osoitteesta <URL: http://tools.ietf.org/html/rfc1459>, viitattu 31.10.2011.

[13] Peter Saint-Andre, *"XEP-0001: XMPP Extension Protocols"* Saatavilla osoitteesta <URL: http://xmpp.org/extensions/xep-0001.pdf>, 10.3.2010, XEP-0001.

[14] Facebook Developers, *"Facebook Chat API"* Saatavilla osoitteesta <URL: http://developers.facebook.com/docs/chat>, viitattu 16.11.2011.

[15] Justin Karneges, Peter Saint-Andre, *"XEP-0047: In-Band Bytestreams"* Saatavilla osoitteesta <URL: http://xmpp.org/extensions/xep-0047.html>, 1.3.2011, XEP-0047.

[16] Peter Millard, Peter Saint-Andre, Ralph Meijer, *"XEP-0060: Publish-Subscribe"* Saatavilla osoitteesta <URL: http://xmpp.org/extensions/xep-0060.html>, 11.7.2010, XEP-0060.

[17] Peter Saint-Andre, *"XEP-0066: Out of Band Data"* Saatavilla osoitteesta <URL: http://xmpp.org/extensions/xep-0066.html>, 16.8.2006, XEP-0066.

[18] Thomas Muldowney, Matthew Miller, Ryan Eatmon, Peter Saint-Andre, *"XEP-0096: SI File Transfer"* Saatavilla osoitteesta <URL: http://xmpp.org/extensions/xep-0096.html>, 13.4.2004, XEP-0096.

[19] Peter Saint-Andre, *"XEP-0234: Jingle File Transfer"* Saatavilla osoitteesta <URL: http://xmpp.org/extensions/xep-0234.html>, 29.6.2011, XEP-0234.

[20] Dave Smith, Matthew Miller, Peter Saint-Andre, Justin Karneges, *"XEP-0065: SOCKS5 Bytestreams"* Saatavilla osoitteesta <URL: http://xmpp.org/extensions/xep-0065.html>, 20.4.2011, XEP-0065.

Liite 1. Yhteyden hallinta