

TEKNILLINEN KORKEAKOULU

Sähkö- ja tietoliikennetekniikan osasto

Jan Wagner

REAALIAIKA-PITKÄKANTAINTERFEROMETRIAN KEHITTÄMINEN  
METSÄHOVIN RADIODIAGNOSTISILLA LAITTEILLA

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi diplomi-insinöörin  
tutkintoa varten Espoossa 19. marraskuuta 2007

Työn valvoja

Professori Martti Hallikainen

Työn ohjaaja

DI Jouko Ritakari

HELSINKI UNIVERSITY  
OF TECHNOLOGYAbstract of the  
Master's thesis

<b>Author:</b>	Jan Wagner	
<b>Name of the thesis:</b>	Development of Real-time Very Long Baseline Interferometry Systems at the Metsähovi Radio Observatory	
<b>Date:</b>	November 19, 2007	<b>Number of pages:</b> 74
<b>Department:</b>	Dep. of Electrical and Communications Engineering	
<b>Professorship:</b>	S-92 Space Technology	
<b>Supervisor:</b>	Professor Martti Hallikainen	
<b>Instructor:</b>	MSc Jouko Ritakari	
<p>Very long baseline interferometry (VLBI) allows radio astronomers to resolve the structure of stellar sources with high sensitivity and unprecedented angular resolution. Even geodesy, astrometry and spacecraft tracking often employ VLBI for their measurements. In traditional VLBI the results are unfortunately available only after a very long delay. However, with the advent of high-speed networks it has now become possible to significantly shorten the turn-around time and ultimately provide results in real-time. Real-time results are a major goal of the European Union EXPReS development project. Metsähovi is responsible for the data acquisition system.</p> <p>This master's thesis describes the EXPReS development work I carried out at Metsähovi. The thesis covers the development of real-time data acquisition and a corresponding suitable data transfer protocol. In addition, it describes a successful international real-time demonstration given to the EU.</p> <p>This work has received financial support from the European Commission (DGINFSO), within the Sixth Framework Programme (Integrated Infrastructure Initiative contract number 026642, EXPReS).</p>		
Keywords: e-VLBI, DAQ, real-time data transfer, real-time network protocols		

## TEKNILLINEN KORKEAKOULU

## Diplomityön tiivistelmä

<b>Tekijä:</b>	Jan Wagner	
<b>Työn nimi:</b>	Reaaliaika-pitkäkantainterferometrian kehittäminen Metsähovin Radiotutkimusasemalla	
<b>Päiväys:</b>	19. marraskuuta 2007	<b>Sivumäärä:</b> 74
<b>Osasto:</b>	Sähkö- ja tietoliikennetekniikan osasto	
<b>Professori:</b>	S-92 Avaruustekniikka	
<b>Valvoja:</b>	Professori Martti Hallikainen	
<b>Ohjaaja:</b>	DI Jouko Ritakari	
<p>Radioastronomiassa voidaan pitkäkantainterferometrian (VLBI) avulla kuvata tähti- taivaan kohteiden rakenteita erinomaisella herkkyydellä ja kulmaresoluutiolla. VLBI on myös suosittu mittausmenetelmä geodesiassa, astrometriassa ja avaruusluotainten seuraamisessa. Valitettavasti kohteiden havainnoinnin ja lopputulosten saamisen vä- lillä pitkä, jopa viikkojen mittainen viive, joka johtuu asemien tietojen lähettämisestä postitse sekä korrelaattoriaseman rajoitteista. Nopeiden tietoliikenneverkkojen kehitty- misen myötä on kuitenkin mahdollista lyhentää viiveitä huomattavasti – viime kädessä VLBI-havainnointi tulee antamaan tuloksia jopa reaaliajassa. Reaaliaikainen VLBI on tärkein päämäärä Euroopan Unionin EXPReS -kehityshankkessa. Metsähovin tehtävänä on kehittää tähän sopiva reaaliaikainen tiedonkeruujärjestelmä.</p> <p>Tässä diplomityössä kuvataan Metsähovissa EXPReS-projektia varten tekemäni työt. Eryteisesti käsittelen toimivan toimivan reaaliaikaisen tiedonkeruun ja tiedonsiirron vaa- timat kehitystyöt. Lopuksi esitän tulokset menestyksekkästä kansainvälisestä VLBI- demonstraatiosta, jossa EU:lle osoitettiin kehittämäni järjestelmän toimivuus.</p>		
Avainsanat: e-VLBI, tiedonkeruu, reaaliaikainen tiedonsiirto, tiedonsiirtoproto- kollat		

## Alkusanat

Tämä diplomityö on tehty Metsähovin radiotutkimuslaitoksella osana Euroopan Unionin EXPReS-projektia. Kiitän Euroopan Unionia tämän diplomityön rahoittamisesta FP6-kehityksen Integrated Infrastructure Initiative sopimuksen 026642 alla.

Diplomityön ohjaajaa Jouko Ritakaria, Ari Mujusta ja Guifré Moleraa haluan kiittää hienosta yhteistyöstä ja aina asiantuntevista neuvoista. Kiitän myös professori Martti Hallikaista tämän työn valvomisesta.

Lukuisten muiden radio-observatorioiden teknillinen henkilökunta ansaitsee myös kiitokset ystävällisestä avustuksesta reaaliaikaisissa siirtotesteissä sekä kehitystöihin liittyvästä rakentavasta palautteesta.

Lopuksi tahdon kiittää vanhempiani ja siskoani heidän tuestaan ja kirjoitustöihin kannustamisesta.

Vielen Dank auch an Ulla, Oma und HJ für alle Unterstützung!

27. marraskuuta 2007 Jan Wagner

## Sisältö

<b>Abstract</b>	<b>ii</b>
<b>Tiivistelmä</b>	<b>iii</b>
<b>Alkusanat</b>	<b>iv</b>
<b>Lyhenteet ja symbolit</b>	<b>vii</b>
<b>1 Johdanto</b>	<b>1</b>
<b>2 Reaaliaikainen e-VLBI</b>	<b>7</b>
2.1 <i>Metsähovin PC-EVN -järjestelmä</i>	10
2.2 <i>Näytteenottojärjestelmät</i>	14
<b>3 Reaaliaikaisen e-VLBI:n kehitystyöt</b>	<b>17</b>
3.1 <i>VSIB-ajurin lähdekoodin päivitys</i>	17
3.2 <i>10 Gbit/s verkkoyhteyden käyttöönotto</i>	19
3.3 <i>Tsunami UDP -protokolla</i>	22
3.3.1 <i>Lähetysten uudelleenaloitus</i>	27
3.3.2 <i>Rinnakkaisten Tsunami-asiakkaiden ajo</i>	28
3.3.3 <i>Käyttäjäraportointi</i>	29
3.3.4 <i>Muut toiminnalliset lisäykset</i>	29
3.4 <i>Reaaliaikainen Tsunami-protokolla</i>	30
3.4.1 <i>Yksittäisen VLBI-siirron asetusten välittäminen reaaliaikapalvelimelle</i>	34
3.4.2 <i>Aikarajoitetut uudelleenlähetykset</i>	35
3.4.3 <i>896 Mbit/s e-VLBI moodi</i>	39
3.5 <i>Reaaliaikaisen e-VLBI -havainnoinnin automatisointi</i>	41
<b>4 Kehitystyön tuloksia</b>	<b>44</b>
4.1 <i>Metsähovin sisäinen e-VLBI -testaus ja 1 Gbit/s kuitutesti</i>	44
4.2 <i>Kolmen aseman e-VLBI Month7 EU-demonstratio</i>	48
4.3 <i>896 Mbit/s siirtotestin tulokset</i>	53

4.4	<i>Tuotantokäyttö geodesiassa</i>	55
5	<b>Yhteenveto ja tulevaisuudennäkymä</b>	<b>57</b>
A	<b>Liitteet</b>	<b>60</b>

## Lyhenteet ja symbolit

- 1PPS** 1 pulssi per sekunti, aikareferenssin antama sähköinen signaali
- ADC** Analog to Digital Converter, analogisen signaalin näytteistys
- ASIC** Application Specific Integrated Circuit, käyttötarpeisiin räätälöity IC
- BBC** Baseband Converter, kaksi kaistaa perustaaajuudelle sekoittava laite
- COTS** Commercial off-the-shelve, kaupalliset laitteet ja komponentit
- DAQ** Data Acquisition, tiedonkeruu
- DAS** Data Acquisition System, tiedonkeruujärjestelmä
- e-VLBI** Electronic VLBI, tietoliikenneverkkoja hyödyntävä VLBI
- ESA** European Space Agency, Euroopan Avaruusjärjestö
- EVN** European VLBI Network, Euroopan VLBI-asemien verkosto
- EXPRoS** Express Production Real-Time e-VLBI Service, kehityshanke
- FLOP** Floating Point Operations per Second, liukulukulaskentanopeus
- FPGA** Field Programmable Gate Array, ohjelmoitava logiikkaverkko
- JIVE** Joint Institute for Very Long Baseline Interferometry in Europe
- Jy** Jansky,  $10^{-26}$  W/m<sup>2</sup>Hz
- LVDS** Low Voltage Differential Signaling, sarjaliikenneväylä
- arcsec** Kaarisekunti
- marcsec** Millikaarisekunti
- Mbit/s** Megabittiä sekunnissa
- MRO** Metsähovi Radio Observatory, Metsähovin Radiotutkimusasema
- MTU** Maximum Transmission Unit, Ethernetkehityksen kokorajoitus
- NASA** National Aeronautics and Space Administration
- RAID** Redundant Array of Independent Disks
- SATA** Serial-AT Attachment, kovalevyliitäntä
- SNR** Signal to noise ration, signaalikohinasuhde

**TCP/IP** Transmission Control Protocol siirtoprotokolla Internet Protocol -joukosta  
**UDP/IP** User Datagram Protocol yhteydetön protokolla Internet Protocol -joukosta  
**VLBI** Very Long Baseline Interferometry, pitkäkantainterferometria  
**VSI** VLBI Standard Interface, siirtoväylä näytteille  
**VSIB** VSI Board, I/O-kortti tietokoneille  
**VSIC** VSI Converter, toisen väylän VSI-väylälle sovittava laite

*B* Järjestysnumero, lohkonumero

$\lambda$  Aallonpituus

*L* Tiedon pituus, tavuja

$\tau$  Aika

*R* Bittinopeus, siirtonopeus

$\nu$  Taajuus, havaintotaajuus

$\Delta\nu$  Taajuuskaista, kaistanleveys

*x* Koordinaatti



## Luku 1

### Johdanto

Kuten klassisen optisen astronomian tutkimusalalla tutkitaan viime vuosisadan alkupuolella Grote Reberin innoittamalla uudella radioastronomian tutkimusalalla tähtitaivaan ilmiöitä ja kohteita seuraamalla näiden lähettämää sähkömagneettista säteilyä. Optisessa astronomiassa käytetään radioastronomiaan verrattuna huomattavasti korkeampia taajuuksia nanometriä aallonpituuksilla. Näkyvä valo esimerkiksi sijaitsee taajuusalueella 400—800 THz. Näin korkeilla taajuuksilla voidaan astronomiassa soveltaa klassisen optiikan menetelmiä ja tekniikkaa. Radioastronomiassa taas tällä hetkellä tyypilliset taajuudet sijoittuvat välille 10 MHz—300 GHz, eli alkavat kymmenien metrien aallonpituuksista ja etenevät aina mikro- ja millimetriaaltoalueille asti. Radioastronomian suhteellisen matalien taajuuksien takia havainnoinnissa sovelletaan alan nimen mukaisesti radiotekniikan menetelmiä ja käytetään radiotekniikan vastaanotinlaitteistoa. Nykyään maailmalla on jo useita kymmeniä eri kokoluokan radioteleskooppeja ja radio-observatorioita. Matalin taajuusalue on 10—150 MHz, jota tullaan käyttämään Hollannissa rakenteilla olevassa Low Frequency Active Array -verkostossa (LOFAR). Suurin havainnointitaajuus taas tulee olemaan Chileen valmistuvassa Atacama Large Millimeter Array alimillimetriaaltoteleskoopissa (ALMA).

Radioastronomiassa käytetään optiseen astronomiaan verrattuna matalampia taajuuksia, eli pitempiä aallonpituuksia. Tämä heijastuu myös radioteleskoopin rakenteiden koossa. Reflektorirakenteisen teleskoopin kuten tavallisen Cassegrain-teleskoopin erotuskyky  $\theta$ , joka vastaa antennin -3dB -keilanleveyttä, on kaavan 1.1 mukaisesti suoraan verrannollinen käytetyn aallonpituuden  $\lambda$  sekä teleskoopin apertuurin eli aukon läpimitan  $D$  suhteeseen.

Sama pätee myös vapaisiin antennirakenteisiin, joissa ei käytetä heijastinelementtejä.

$$\theta \propto \frac{c}{\nu} = \frac{\lambda}{D} \quad (1.1)$$

Käytetyn aallonpituuden kasvaessa radioteleskooppien erottelukyky huononee ja on käytettävä suurempaa apertuuria  $D$  aiemman erottelukyvyn säilyttämiseksi. Optisella alueella 10 cm:n apertuuri mahdollistaa noin kaarisekunnin resoluution, kun taas 22 GHz:n taajuudella toimivassa radioteleskoopissa sama erottelukyky saavutetaan vasta noin 3 km:n apertuurilla. Ekstragalaktisten kohteiden hienorakenteet ovat alle 1 millikaarisekunnin suuruusluokkaa. Vastaava apertuuri 22 GHz radioteleskoopilla pitäisi olla vähintään 3000 km. Käytännössä rakennustekniikka, kustannukset ja teleskooppirakenteiden toleranssivaatimukset rajoittavat toteutuskelpoisten reflektoriteleskooppien apertuurikokoja niin paljon, että 100 metriä suurempia teleskooppeja ei enää ole kustannustehokasta eikä järkevää rakentaa. Näiden rajoitusten takia on radioteleskoopilla saavutettava resoluutio yleensä kaariminuutin luokkaa.

Radioteleskooppien heikon erottelukyvyn takia radioastronomit ovat aiemmin seuranneet pääasiassa auringon pintaa, avaruudesta pilvinä löytyvien alkuaineiden ja molekyylien kuten esimerkiksi vedyn ja metanolin absorptio- ja emissiospektrejä, pulsareita, kvasaareita, sekä muita radioäänekkäitä tai ison suuruusluokan kohteita. Yksittäistä radioteleskooppia huomattavasti suurempi erottelukyky saavutettiin 1967 (Algonquin Radio Observatory, Dominion Radio Astrophysical Observatory), kun nobeloidut A. Hewish ja M. Ryle ensimmäistä kertaa sovelsivat silloin jo tutkakniikassa käytettyä vaiheistetun antenniryhmän menetelmää yhteenkytkettyihin radioteleskooppiryhmiin. Tässä uudessa sovelluksessa antennista saadaan muodostettua alkukantainen interferometri summaamalla yhteen kahden tai useamman, samaa taivaan kohdetta seuraavan teleskoopin antennisignaalia ja säätämällä signaalien keskinäistä viivettä (Tho91). Interferometriin kytkettyjen antennien muodostamia antennipareja kutsutaan kannoiksi ja kannan antennien välistä euklidista etäisyyttä kannan pituudeksi. Antenneihin saapuva eli pisteittäin havainnointu kohteesta peräisin oleva tasoaalto muodostaa interferoinnissa interferenssikuvion sille ominaisilla vyöhykkeillä (fringes). Radioastronomiassa kiinnostavimman keskimmäisen vyöhykkeen leveys  $x$  riippuu yhden kannan tapauksessa kaavan 1.2 mukaisesti käytetystä aallonpituudesta  $\lambda$ , kannan  $B$  projektiosta  $B'$  lähteen emittoiman aaltorintaman tasolle,

ja  $B'$  sisältämän tason etäisyydestä  $r$  kohteeseen.

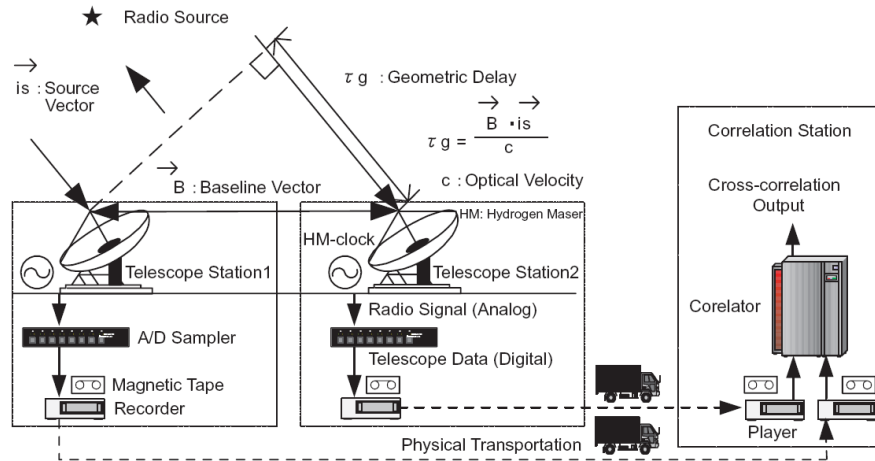
$$\frac{x}{r} = \frac{\lambda}{B'} \quad (1.2)$$

Vyöhykkeen leveys  $x$  on suoraan verrannollinen interferometrin kaariasteissa lausuttuun erottelukykyyn. Vakiotaaajuudella keskimmäinen vyöhyke kapenee ja interferometrin erottelukyky paranee mitä kauempana teleskoopit sijaitsevat toisistaan. Esimerkiksi millimetriaaltoalueella saavutetaan erinomainen resoluutio jo parin sadan metrin pituisilla kannoilla.

Radioastronomiassa tavallisesti käytetyillä pitkillä aallonpituuksilla myös kannat ovat pitempiä, usein tuhansia kilometrejä. Näillä aallonpituuksilla interferometriä kutsutaan pitkäkantainterferometriaksi (Very Long Baseline Interferometry, VLBI). Samasta VLBI-tekniikasta on joitakin variaatioita, kuten geodeettinen geo-VLBI, tai interferenssikuvion muitakin vyöhykkeitä hyödyntävä, laskennallisesti raskaampi laajan kentän wide-field-VLBI.

Maapallon pyörimisliikkeen ja kiertoradan takia kantaprojektio  $B'$  ja siten myös interferenssikuvio muuttuu jatkuvasti, jolloin saadaan radiotekniikan apertuurisynteesin tapaan interferenssikuvioden aikasarja. Radioastronomiassa apertuurisynteesin laskennallista synteesi- eli interferointivaihetta kutsutaan korreloinniksi, viittauksena laskennassa käytettyihin signaalien auto- ja ristikorrelaatioihin. Maailmassa on joitakin harvoja korrelointipalveluja tarjoavia laitoksia. Euroopassa näistä sijaitsevat ainoastaan tähtitieteilijöihin keskittyvä hollantilainen instituutti Joint Institute for Very Long Baseline Interferometry in Europe (JIVE) sekä pääasiassa geodeettisia VLBI-kokeita korreloiva saksalainen Max-Planck-Institut für Radioastronomie (MPIfR).

Eräs ongelma VLBI:ssä on antennien vastaanottamien signaalien häiriötön ja häviötön siirto korrelaattorikeskukseen. Lyhyillä välimatkoilla voidaan käyttää aaltojohtoja, kuten esimerkiksi tehdään yhä Westerborkin antenniryhmässä. Pidemmällä etäisyyksillä tämä ei kuitenkaan ole enää käytännöllistä, ja signaalit on tallennettava ja siirrettävä muilla menetelmillä, kuten esimerkiksi fyysisesti kuljettamalla tallennusmediat korrelaattorikeskukseen kuten kuvan 1.1 VLBI-asetelmassa. Toinen ongelma on tallennusjärjestelmien rajallinen kaistanleveys ja tallennusnopeus. Tämän takia aiemmin laajakaistainen alkupektri harvennettiin samalla heikentyvän VLBI-herkkyyden kustannuksella rajattuun  $2^n$ -



Kuva 1.1. VLBI-havainnoinnin toimintaperiaate, tiedonkeruu ja tallenteiden vanha fyysinen siirtotapa korrelaattoriasemalle (HHK03).

kokoiseen kanavajoukkoon kapeita, esimerkiksi alle 32 MHz levyisiä kaistoja, jotka perustajuudelle sekoittamisen jälkeen tallennettiin alun perin analogimuodossa magneettinauhoille. Analogi-digitaalimuuntimien (ADC) kehittyessä siirryttiin tallentamaan kaistoja digitaalisessa muodossa VLBI-tarkoituksiin täysin riittävällä 1-4 bitin resoluutiolla. Vuosia myöhemmin, kuluttajätietokoneiden kehittyessä ja tavallisten kovalevyjen ohittaessa magneettinauhat hintaan nähden suuremmalla tallennuskapasiteetilla, tallennusnopeudella ja luotettavuudella, VLBI:ssä siirryttiin käyttämään kovalevytallennusta (Whi99). Lisää kompaktia historiatietoa USA:n ja sieltä myös Eurooppaan rantautuneen pitkäkantainterferometrian teknillisestä kehityksestä löytyy Preussin tiivistelmästä (Pre02).

Laajakaistaisen spektrin harventaminen kapeisiin osakaistoihin heikentää VLBI-havainnoinnin herkkyyttä. Tämän takia on aina tallennustekniikan salliessa pyritty siirtymään leveämpien osakaistojen käyttöön. Esimerkiksi NRAO-Cornellin vuonna 1967 kehittämä Mark I VLBI-järjestelmä käytti 300 kHz kaistanleveyttä ja sen ajan ensimmäisiä videokasetteja, mutta neljä vuotta myöhemmin Mark II -järjestelmässä käytettiin jo 2 MHz kaistanleveyttä. MIT Haystack Observatory -laitoksen kehittämään Mark IV VLBI-järjestelmään kytkeytyvät, maailmalla nykyään laajasti käytössä olevat Haystack Mark5A, A+, B, B+ ja C tallennusjärjestelmät ovat vanhan Mark VLBI-järjestelmän ominaisuuksien takia käytännössä rajoittuneet 32 tai 64 MHz kaistanleveyksiin 64 tai 32 kanavalla, kun taas japanilainen PC-VSI ja ADS-3000 VLBI-järjestelmä käyttää jo 1024 MHz kais-

Data rate	Frequency (GHz)				
	1.4	2.4	5.0	8.4	20
128 Mbps (2×16 MHz)	27	29	80	30	105
1 Gbps (2×128 MHz)	9.4	10	28	10	37
4 Gbps (2×512 MHz)	4.7	5.1	14	5.2	19
8 Gbps (2×1 GHz)		3.6	10	3.7	13
16 Gbps (2×2 GHz)			7.1	2.6	9.3

Kuva 1.2. Esimerkki kaistanleveyden vaikutuksesta VLBI-kuvan RMS signaali-kohinasuhteeseen ( $\mu\text{Jy/keila}$ ) eri havainnointitaajuuksilla yhden kannan 12 tunnin kokeessa. Yhdistetty tiedonsiirtonopeus vastaa kahta aseman tuottamaa 2-bittisten näytteiden virtaa näytteenottotaajuuudella (Phi04).

tanleveyksiä yhdellä tai usemmalla kanavalla (Whi02; TKN06). Metsähovissa 2001 kehitetty PC-EVN tiedonkeruujärjestelmä vuorostaan tukee standardoitua VSI-H väyläformaattia ja myös useita vanhoja formaatteja ja sijoittuu tällä hetkellä edellisten välimaastoon yksittäisellä PC-EVN -koneella mahdollisella 512 tai 1024 Mbit/s kokonaistallennusnopeudella (RiM04; DTW04). Seuraava askel tehdään EXPRoS -projektissa, jonka eräs tavoite on vähintään 2048 MHz kokonaiskaistanleveyttä vastaavan toimivan 4 Gbit/s tiedonkeruun laajamittainen käyttö Euroopan radio-observatorioissa sekä vuosikymmenen lopussa 30 Gbit/s nopeuskapasiteettiin siirtyminen (1, 5. C.1.1.1.).

Leveämmät taajuuskaistat parantavat VLBI:n herkkyyttä ja mahdollistavat tarkempien ja vähäkohinaisempien karttojen laskemisen. Kuvassa 1.2 on esimerkki VLBI-kuvan odotetusta RMS signaali-kohinasuhteesta ( $\mu\text{Jy/keila}$ ) eräiden kaistanleveyksien  $\Delta\nu$  ja näitä vastaavien tiedonsiirtonopeuksien suhteen eri havainnointitaajuuksilla (Phi04). Kaistanleveyden  $\Delta\nu$  vaikutus VLBI:n herkkyyteen eli havaittavaan pienimpään tehotiheyteen  $S_\nu$  taajuudella  $\nu$  nähdään kaavasta 1.3 (Tho91).

$$\Delta S_\nu = \frac{\sqrt{2k_b T_{sys}}}{\eta_a \eta_c A \sqrt{\Delta t \Delta \nu} \sqrt{\frac{1}{2} N(N-1)}} \quad (1.3)$$

Herkkys  $S_\nu$  paranee myös pidentämällä taivaan kohteen havainnointiaikaa  $\Delta t$  tai kasvattamalla VLBI-mittaukseen osallistuvien asemien määrää  $N$ . Koska tavallisesti pyritään havaitsemaan mahdollisimman monta kohdetta – skannia – yhden VLBI-kokeen rajallisen

antenniajan aikana,  $\Delta t$  pidentäminen ei ole toivottava vaihtoehto. Lisäksi osallistuvien asemien määrää  $N$  ei käytännössä ole helppo kasvattaa. On helpompaa käyttää suurempaa  $\Delta\nu$ . Tällöin  $C$  kanavaa  $b$ -bittisesti näytteenottotaajuudella  $f_s = 2\Delta\nu$  näytteistävän tiedonkeruujärjestelmän on pystyttävä vähintään kaavan 1.4 mukaiseen  $R_{bit}$  (bit/s) tiedonkeruunopeuteen:

$$R_{bit} = C \cdot f_s \cdot b \quad (1.4)$$

Vielä vuonna 2006 tavallisimmin käytetyt tiedonkeruunopeudet olivat 128, 256 ja 512 Mbit/s. Nämä nopeudet ovat huomattavasti pienempiä kuin nykyisissä 1 Gbit/s lähiverkossa ja euroopanlaajuisessa GÉANT2 10 Gbit/s tutkimusverkossa saavutettavissa olevat tiedonsiirtonopeudet. Eräs EXRPeS-projektin tavoitteista onkin hyödyntää Euroopan VLBI-verkoston (EVN) asemien tietojen siirrossa nykyistä tietoliikennetekniikkaa, aiemmin käytetyn hitaan fyysisen kuljetuksen sijaan. Euroopan sisällä tämä on jo tarpeeksi haastavaa. Lisäksi EVN:ään kuuluu useita asemia muilla mantereilla, joiden tiedonsiirrossa on käytettävä kansainvälisiä verkkoyhteyksiä. Näytteiden keruu ja siirto korrelaattoriasemalle Euroopan sisäisten laajakaistaisten tutkimusverkkojen kautta on seuraavan kappaleen aiheena.

## Luku 2

### Reaaliaikainen e-VLBI

Tällä hetkellä Euroopan VLBI-verkoston observatorioissa käytettyjen yksittäisten tiedonkeruujärjestelmien suorituskykyraja on 1024 Mbit/s, eräitä erikoistapauksia kuten Jodrell Bank e-MERLIN -antenniryhmää lukuun ottamatta. 1024 Mbit/s nopeutta käytetään vielä harvoin, lähinnä millimetri-VLBI -kokeissa. Yleisimpiä ovat 512 Mbit/s nopeus astronomisessa VLBI:ssä sekä 128 tai 256 Mbit/s nopeudet geodeettisessä VLBI:ssä.

Metsähovin PC-EVN ja Haystackin useaan otteeseen jälkikorjattu Mark5 (Mark5A, Mark5A+, Mark5B, Mark5B+) ovat molemmat selvinneet näytevirtojen kirjoittamisesta kovalevyjärjestelmille näillä nopeuksilla; PC-EVN käyttämällä Metsähovissa 2001 kehitettyä VSI-liitännäistä PCI-korttia (VSIB) sekä ohjelmallista tai PC:n emolevyille integroitua laitetason RAID:ia, Mark5 käyttämällä Conduant-yrityksen valmistamaa suhteellisen kallista StreamStor tiedonkeruu- ja kovalevyohjainkorttia erityisessä PC-kokoonpanossa. Yhden PC-EVN -koneen 512 Mbit/s nopeusrajoitus voidaan kiertää jakamalla 1 tai 2 Gbit/s näytevirta kahdelle tai neljälle rinnakkain tallentavalle koneelle. Metsähovi demonstroi Japanin kanssa tehdyssä VLBI-testissä jo vuonna 2003, että tällä menetelmällä voidaan saavuttaa yhteensä 2 Gbit/s tiedonkeruunopeus paikallisille kovalevyjärjestelmille.

Euroopan VLBI-yhteisössä halutaan kuitenkin vähitellen luopua näytteiden paikallisesta kovalevytallennuksesta. EXPRoS-projekti ajaa myös tätä kehitystä. Jotta observatorion levytallennuksesta voidaan päästä, ensimmäinen ja tärkein tavoite on kaikkien observatorioiden näytevirtojen siirto korrelaattoriasemalle reaaliajassa verkon yli, eli tulee rakentaa toimiva reaaliaikainen e-VLBI -havainnointimoodi. Näytteet kulkevat asemalta Euroopan nopeiden tutkimusverkkojen ja Internetin yli ja voidaan lopuksi tallentaa korrelaattoriaseman omille tiedostopalvelimille.

Seuraava kehitysaskel on reaaliaikaisten näytevirtojen korrelointi reaaliajassa, jolloin korrelaattoriasemalla ei enää tarvita suuren tallennuskapasiteetin tiedostopalvelimia. Ainoat tiedot jotka tallennetaan ovat korrelointiprosessin lopputulokset, jotka muodostavat murto-osan alkuperäisestä yhteenlasketusta asemien tietomäärästä. Koska asemien tietoja ei missään vaiheessa tallenneta, reaaliaikainen korrelointi soveltuu lähinnä niihin VLBI-kokeisiin, joissa korrelointiprosessia ei haluta suorittaa useaan kertaan eri parametreilla.

Täysin reaaliajassa toimivalla e-VLBI:llä on useita teknisiä ja tieteellisiä etuja. Eduista ehkä tärkein on se, että korrelointiprosessi antaa väliaikaistuloksia reaaliajassa. Näistä pystytään havaitsemaan mahdolliset ongelmat jonkin aseman laitteistossa tai laitteiston asetuksissa, joilloin ongelmista voidaan ilmoittaa asemalle jo VLBI-kokeen aikana. Jos taas perinteisellä tavalla lähetettäisiin aseman kovalevyypaketteja postissa, vastaavia ilmoituksia jouduttaisiin odottamaan viikkoja. Pahimmassa tapauksessa asema on tällä välin jo osallistunut toisiin VLBI-kokeisiin, ilman että ollaan tietoisia laitteisto-ongelmasta. Ongelman takia aseman VLBI-kokeiden tiedot saattavat olla vain rajallisesti käyttökelpoisia. Reaaliaikaisen e-VLBI:n ansiosta ongelmat voidaan teoriassa yrittää korjata jo VLBI-kokeen aikana, jolloin aseman tiedot pysyvät käyttökelpoisina, eikä VLBI-kokeen lopullinen herkkyys välttämättä huonone. Toinen reaaliaikaisen e-VLBI:n etu on, että aina yhden tähtitaivaan kohteen havainnoinnin eli skannin jälkeen korrelointitulokset ovat valmiita ja saatavilla, joten tähtitieteilijät voivat halutessaan aloittaa jo kokeen aikana tulosten jatkoprosessoinnin tarkaksi kohteen kuvaksi tai kartaksi. Kolmanneksi e-VLBI:n eduksi yleensä luvataan, että tähtitieteilijä pystyvät seuraamaan aiempaa paremmin tähtitaivaan äkillisiä ja lyhytkestoisia ilmiöitä, esimerkiksi supernovia ja gammapurkauksia. Aiemmin tämä oli epäkäytännöllistä VLBI-kokeiden pitkäkestoisen aikataulusuunnittelun ja asemien tiedontallennusjärjestelmissä vaadittujen valmistelujen takia.

Euroopan VLBI-yhteisössä reaaliaikaiseen e-VLBI:hin siirtyminen tapahtuu välivaiheitten kautta. Ensimmäinen vaihe on off-line e-VLBI, jossa tiedot tallennetaan paikallisesti ja siirretään VLBI-kokeen päädyttyä verkon yli korrelaattorille. Esimerkiksi vuoden 2006 alkupuolella niillä harvoilla EVN:n asemilla joilla oli jo nopea verkkoyhteys, yleisin e-VLBI -havainnointitapa oli tallentaa tiedot ensin Mark5-järjestelmällä erikoisrakenteisille Mark5-levyapaikoille, tai PC-EVN:llä tavallisille kovalevyille. VLBI-kokeen jälkeen skannit siirrettiin verkon yli JIVE:n korrelaattoriasemalle siellä olevien Mark5-koneiden kovale-



vyille.

Mark5-järjestelmän tapauksessa off-line e-VLBI ja skannien siirto verkon yli on erityisen hankalaa, koska Conduant StreamStor -ohjaimen kytkettyä levypakkaa, jolta skannit tulee lukea ulos, ei voida käyttää tavallisena Linux-tiedostojärjestelmänä. Levypakalla olevat skannit on käsiteltävä yksi kerrallaan, purkamalla ensin erikoisohjelmalla skanni levypakalta tavalliseksi tiedostoksi Mark5:en Linux-järjestelmälevylle. Sieltä tiedosto siirretään tavallisilla FTP- tai bbFTP-ohjelmilla tai Haystackin EGAE-tiedostonsiirto-ohjelmalla verkon yli JIVE:ssä sijaitsevien Mark5-palvelimien järjestelmä- tai RAID-levylle (bbFTP; FTP; EGAE). Periaatteessa ei ole vaikeaa ottaa jonkin tiedostonsiirto-ohjelman kuten FTP:n lähdekoodi ja muokata sitä siten, että ohjelma käyttäisi StreamStor-kutsurajapintaa ja osaisi lukea skannit suoraan levypakalta. Kuitenkaan kukaan ei ole ymmärtänyt tehdä tätä ohjelmointityötä. Lopputuloksena on, että skannit puretaan aina ensin järjestelmälevylle. Skannien siirtonopeus verkon yli ei ole kovin suuri. Vaikka JIVE on yksittäisissä testeissä saavuttanut 400 Mbit/s siirtonopeuksia, on EVN:n asemilla nopeus käytännössä vain 10–100 Mbit/s. Tämä johtuu käytetystä tiedonsiirto-ohjelmasta, lähettävän aseman hitaasta järjestelmälevystä sekä Mark5-koneiden varsin vanhasta emolevystä ja prosessorista. Kun skannit on siirretty JIVE:en, JIVE:n henkilökunta tavallisesti kopioi skannitiedostot omalta RAID- tai järjestelmälevyltä asemakohtaisille Mark5-levypakoille. Usein nämä pakat poistetaan verkkoon kytketyistä Mark5-koneista ja siirretään opiskelijatyövoimalla ensin varastohuoneeseen odottamaan MarkIV-korrelaattorin vapautumista tätä VLBI-koetta varten. Korrelaattorin vapauduttua pakat täytyy siirtää korrelaattoria syöttäviin Mark5-koneisiin, jonka jälkeen varsinainen korrelointi voidaan viimein aloittaa.

PC-EVN -järjestelmän tapauksessa skannien siirto observatoriosta korrelaattoriasemalle on yksinkertaisempaa. Havainnoinnin aikana eräs apuohjelma kirjoittaa VSIB-kortilla kaapattua näytevirtaa RAID-levylle tavallisiin tiedostoihin. Tiedostot voidaan myöhemmin siirtää korrelaattoriaseman RAID-levylle esimerkiksi FTP:llä. Siirto on huomattavasti nopeampaa käyttämällä Tsunami UDP -protokollaa, joka on vapaasti saatavilla ylläpitämästäni SourceForge-projektista (Wag06-1). Tsunamilla voidaan helposti saavuttaa 800 Mbit/s siirtonopeuksia 1 Gbit/s Internet-yhteyden yli. Tiedonsiirron päätyttyä vaikeudet ovat tässäkin tapauksessa korrelaattoriasemalla, jonka on kopioitava vastaanotetut tiedostot RAID-levyltä levypakoille.

Toimivan off-line e-VLBI -moodin jälkeen seuraava vaihe e-VLBI:n kehityksessä on tiedon siirto observatorioista korrelaattoriasemalle reaaliajassa ja siis ilman välitallennusta. Tämä vaatii korrelaattoriasemalla hyvät verkkoyhteydet ja riittävän määrän vastaanottavia tietokoneita. Jos korrelaattori on vielä liian hidas reaaliaikaiseen prosessointiin, asemien reaaliajassa vastaanotetut tiedot voidaan tallentaa samoin kuin off-line -moodissa ensin korrelaattoriaseman RAID-levyille ja prosessoida myöhemmin. Todella alusta loppuun reaaliaikainen e-VLBI vaatii tässä tapauksessa vielä korrelaattorin lisäkehitystöitä. Esimerkiksi EVN:ssä JIVE on EXPRoS-projektin takia päivittämässä laitteistokorrelaattoria ja tietoliikenneyhteyksiä, jotta verkko ja korrelaattori selviäisivät reaaliajassa 16 observatoriosta vähintään 512 Mbit/s asemakohtaisella nopeudella, eli itse korrelaattorin tulee kyetä käsittelemään sisääntulevaa tietoa 8 Gbit/s kokonaisnopeudella.

Vuoden 2008 lopussa korrelaattorin tulee lisäksi selvitä 4 Gbit/s nopeudella lähetävistä observatorioista. Tämä vaatii jatkossa myös observatorioissa lisää tiedonkeruujärjestelmien kehitystöitä. Diplomityöni aikana päätavoite kuitenkin oli reaaliaikasiirtojen kehittäminen 512 Mbit/s nopeuksille. Vuoden 2006 vaihteessa useat asemat olivat jo joidenkin VLBI-kokeiden osalta siirtyneet reaaliaikaiseen tiedonsiirtoon korrelaattoriaseman levyille käyttäen Mark5- tai jatkokehitettyä PC-EVN -järjestelmää. Vaikka elokuussa 2007 JIVE:n päivitystyö oli edennyt jo kuuden aseman 512 Mbit/s reaaliaikakorrelointiin, käytetään tavallisissa VLBI-kokeissa yhä kovalevytallennusta ja off-line- tai reaaliaikasiirtoja. Seuraavassa kahdessa kappaleessa esitellään tarkemmin Metsähovin PC-EVN -järjestelmän rakenne ja toiminta, ja esitetään tavallisen EVN:ssä käytetyn näytteenottojärjestelmän rakenne yksinkertaistetussa muodossa. Tämä antaa tarpeellisen teknisen pohjan seuraavassa luvussa 3 kuvatuille kehityksille, jotka tein EXPRoS-projektissa ja joiden kautta sekä automatisoitu off-line- että luotettava reaaliaikainen e-VLBI -havainnointi ovat tulleet mahdolliseksi PC-EVN -järjestelmällä.

## 2.1 Metsähovin PC-EVN -järjestelmä

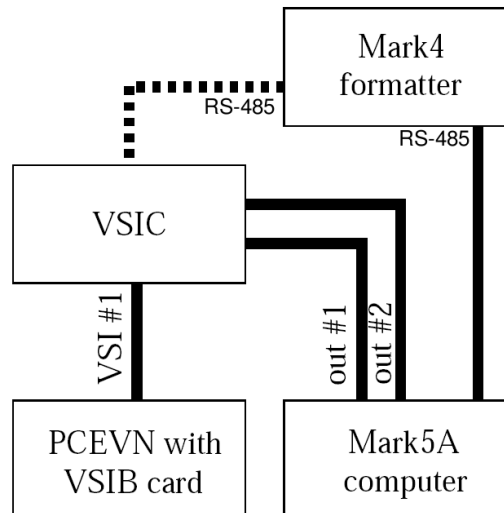
Metsähovin PC-EVN -järjestelmä perustuu kuluttajätietokoneisiin, jotka on rakennettu tavallisista suoraan kaupan hyllystä saatavista (COTS) komponenteista. Tietokone kerää tietoja VLBI:ssä käytetyltä VSI-väylältä koneen keskusmuistiin Metsähovin VSIB-

korttin avulla. Nämä käsitteet selitetään alempana tarkemmin. Keskusmuistista eri VSIB-apuohjelmat voivat kirjoittaa tietoja esimerkiksi tiedostoon kovalevyille.

Kovalevytallennusta varten PC-EVN -koneista löytyy vähintään neljä yhdysrakenteista serial-ATA -kovalevyliitäntää (SATA) sekä paikka yhdelle parallel-ATA Linux-järjestelmälevylle. Linuxissa koneen SATA-levyistä luodaan tason 0 ohjelmisto-RAID. RAID 0 tarkoittaa levyjen raidoittamista (striping) ja raitojen lomittamista sekä yhdistämistä yhdeksi loogiseksi levyksi. Käyttämällä  $N$  kappaletta  $C$  gigatavun kokoisia levyjä saadaan loogiselle RAID-levylle tallennustilaa yhteensä  $N \cdot C$ . Raitojen lomittamisen kautta saavutetaan samalla  $N$ -kertainen luku- ja kirjoitusnopeus yhteen levyyn verrattuna. Neljällä halvalla 250 GB tai 500 GB levyllä saadaan näin 1—2 TB kokonaiskapasiteetti. Kapasiteetti vastaa 512 Mbit/s VLBI-kokeessa 16—32 tunnin kokonaishavainnointiaikaa.

Metsähovissa tehtyjen suorituskykytestien perusteella neljän levyn ohjelmistopohjainen RAID 0 tavallisella kesän 2006 kuluttajätietokoneella selviytyy helposti 1—2 Gbit/s siirtonopeuksista, joten myös 2048 Mbit/s VLBI-kokeen kovalevytallennus on mahdollista kuluttajätietokoneella (Mol06).

Ulkoisen järjestelmän näytteistämä antennisignaali tai tarkemmin eri taajuuskanavien digitaaliset näytevirrat luetaan VSI Board -kortin (VSIB) avulla PC-EVN -koneeseen. VSIB on Metsähovissa 2002 suunniteltu digitaalinen I/O-kortti joka asennetaan tietokoneen 32-bittiseen PCI-väylään (RiM04) ja kytketään ulkoiseen VLBI Standard Interface -väylään (VSI). Kortin ulkoinen I/O-rajapinta täyttää kansainvälisesti sovitun VSI Hardware Interface Specification -standardin vaatimukset. Tämä standardi kuvaa käytettävät liittimet, VSI-väylän signaalit ja väylän sähköisen rajapinnan (Whi00). Väylä kuljettaa 40 matalan jännitteen differentiaalista sarjaliikennekanavaa (low voltage differential signaling, LVDS). Liittiminä käytetään tiheitä 80 pinnin MDR-liittimiä. LVDS-sarjaliikennelinjoista 32 siirtävät näytebittejä. Muista linjoista yksi on varattu näytebittien datakellolle, yksi tarkasti joka sekunti lähetettävälle 1PPS -signaalille ja muut eräille lisäsignaaleille. Kortti itse kykenee 1 Gbit/s luku- ja kirjoitusnopeuteen. Tietokoneen 32-bittinen 32 MHz PCI-väylä sallii periaatteessa 1024 Mbit/s siirtonopeuden, mutta PCI-väylän protokollan lisäkustannusten kuten muistiosoitteiden siirron takia todellisuudessa saavutettava siirtonopeus sijoittuu 512—768 Mbit/s välille. Kortti ei tue 64 MHz PCI-väylää. Tiedonkeruu on kuitenkin mahdollista myös 1 Gbit/s nopeudella kelloitetulta väylältä kahden PC-EVN:n avulla, joista toisen VSIB-kortti on alustettu keräämään näytejonosta vain



Kuva 2.1. Vanhan Mark4 Formatter -järjestelmän liittäminen PC-EVN -koneeseen VSIC-konvertterin avulla joko suoraan (katkoviivat) tai ketjuttamalla Mark5-liitäntälevyn kautta (out#1, out#2). Tietoja voidaan kaapata samanaikaisesti Mark5- ja PC-EVN -järjestelmillä (HMW07).

kaikki parilliset ja toinen kaikki parittomat näytteet.

Koska VSIB-kortti on yhteensopiva VSI-H -standardin kanssa, se voidaan liittää kaikkiin VSI-väylää käyttäviin järjestelmiin. Näistä näytteenottojärjestelmiä ovat ainakin amerikkalainen Digital Backend (DBE) sekä italialainen Digital Baseband Converter (dBBC).

Monet vanhat mutta yhä laajasti käytössä olevat näytteenotto- tai tiedonkeruujärjestelmät eivät tue VSI-väylää, vaan käyttävät omia RS-485 tai RS-232 -sarjaliikenneväyliä. Vanhoja väyliä on muun muassa Mark4 Formatter -järjestelmässä, jota käyttävät Very Long Baseline Array (VLBA) Amerikassa sekä lähes kaikki asemaa Euroopan VLBI-verkostossa. Myös Metsähovissa on Mark4 -järjestelmä. Vanhojen sarjaliikenneväylien VSI-yhteensopivuutta varten Metsähovissa kehitettiin 2002 VSIB-kortin ohelle VSI-väylämuunninkortti (VSI Converter, VSIC). VSIC sovittaa sekä väylien signalointitasot että kanavien ja niiden näytteiden esitystavan VSI-väylälle sopivaksi. Myös konversio toiseen suuntaan eli VSI-väylältä vanhalle väylälle onnistuu. Yhteensä VSIC tukee noin kymmentä eri formaattia.

Vanhat järjestelmät voidaan siis VSIC-kortin kanssa liittää VSI-väylään. VSIC-kortin

VSI-ulostuloliittimiin voidaan kytkeä esimerkiksi PC-EVN. Vanhaan väylään voidaan yhä liittää aikaisempi tiedonkeruujärjestelmä eli Mark5. Näin tiedonkeruu onnistuu samanaikaisesti vanhalla Mark5- ja uuteen VSI-väylään liitettävillä järjestelmillä. Periaate on esitetty kuvassa 2.1. Diplomityön kehitystöiden jälkeen useat asemat tekevät näin VLBI-kokeissa; varmuuskopiota tallennetaan vanhalle Mark5-järjestelmälle, ja samanaikaisesti tehdään PC-EVN:llä reaaliaikasiirtoa tai myös kovalevytallennusta.

VSIB-korttia käytetään Linux-käyttöjärjestelmälle kirjoitettujen ohjelmien avulla. Metsähovin A. Mujunen oli kehittänyt kortille ajurin Linux-kernelin versiolle 2.4. Ajuri käyttää M. Welchin luomaa BigPhysArea kernel-patchia ja varaa sen avulla ajurin omaan käyttöön suuren alueen keskusmuistista, jota käyttää rengaspuskurina (Wel96). VSIB-ajuri alustaa kortin tekemään ketjutettuja DMA-siirtoja VSIB-kortin ja tämän rengaspuskurin välillä. DMA-siirtojen käynnistämisen jälkeen VSIB-kortilla oleva PLX Technology PCI-siltauspiiri (PCI9054) hoitaa itsenäisesti 32-bittisten sanojen siirron VSI-väylältä rengaspuskuriin tai toisin päin. Tavallisin siirtosuunta on VSI-väylältä keskusmuistiin.

VSIB-ajuri toteuttaa lisäksi Linux-merkkioheislaitteen rajapinnan (character device). Ajuri esittää tämän rajapinnan läpi sisäisen rengaspuskurin muille ohjelmille jatkuvana tiedostona */dev/vsib*, joka tukee tavallisia POSIX *open*, *close*, *lseek* -tiedostofunktiota. Tämä yksinkertaistaa huomattavasti VSIB-kortin käyttöä eri ohjelmissa.

Käyttäjä voi ajurin parametreissa asettaa rengaspuskurille maksimikoon  $L_{VSIBuf}$  tavuina. Koko voi olla enintään BigPhysArea:n ennakkovaraaman muistin määrä. Tämä taas on korkeintaan 740 MB ja riippuu tietokoneen keskusmuistin määrästä ja Linux-kernelin omasta muistinjaosta. Koska aina uuden tiedon saapuessa rengaspuskuriin puskurin kaikkien vanhin tieto ylikirjoittuu, puskurin sisältämän näyتهistorian aikaikkuna on korkeintaan kaavan 2.1 pituinen  $\tau_{VSIBuf}$ :

$$\tau_{VSIBuf} = \frac{8 \cdot L_{VSIBuf}}{R_{VSI}} \quad (2.1)$$

Tässä  $R_{VSI}$  on VSI-väylän näytteiden bittinopeus, esimerkiksi 512 Mbit/s. Usein käytetty puskurikoko on 144 MB eli kyseisellä nopeudella noin 2 sekuntia. Puskurin kattama rajallinen näyتهistoria  $\tau_{VSIBuf}$  on eräs reaaliaikaisessa e-VLBI:ssä esiintyvä rajoite, joka muiden viiveiden ohella vaikuttaa siirron luotettavuuteen, tiedonhäviöön ja vikasietoisuuteen. Tätä rajoitetta tarkastellaan myöhemmässä reaaliaikaisen Tsunami-siirtoprotokollan

jatkokehitystä kuvaavassa osiossa 3.3.

Nykyinen VSIB-kortti on ominaisuuksiltaan riittävä 512–1024 Mbit/s tiedonkeruuseen ja reaaliaikasiirtoihin. Tähän liittyvä kehitystyö esitetään seuraavassa luvussa. Suurempia nopeuksia varten Metsähovi tulee vuoden 2007 loppupuolella kehittämään täysin verkko-pohjaisen 10 Gbit/s tiedonkeruujärjestelmän. Siihen mennessä EXPReS-projektissa pää-osa Metsähovin e-VLBI -kehitystyöstä on tapahtunut nykyisen PC-EVN -järjestelmän parissa.

## 2.2 Näytteenottojärjestelmät

Ulkoisen järjestelmän antamia näytteitä keräävien Mark5- ja PC-EVN -järjestelmien kanalta ei ole kovin olennaista, mistä lähteestä nämä näytteet ovat peräisin. Tärkeämpää on näytevirtojen formaatti käytetyllä väylällä. Koska eräs diplomityössä kokeiltu uusi e-VLBI -siirtomoodi riippuu näytevirran formaatista ja näytteenottojärjestelmän analogisen puolen ominaisuuksista, kuvataan lyhyesti näytteenottojärjestelmä, joka on käytössä monessa EVN:n observatoriossa ja myös Metsähovissa.

Tämä EVN:ssä yleinen näytteenottojärjestelmä on peräisin NRAO:n kymmenen aseman Very Long Baseline Array -teleskooppiryhmästä (VLBA). Järjestelmään kuuluvia laitteita ovat muun muassa A/D-muunninosa Mark4 Sampler sekä näytevirtojen esitystavan asettava Mark4 Formatter. Järjestelmä on dokumentoitu tarkasti Mark4 Memo Series -raporttisarjassa (Whi90).

Järjestelmän analoginen signaaliketju alkaa antennista ja sen syöttämästä RF-signaalista. Valitulta taajuudelta sekoitetaan kapea 512 MHz kaista IF-välitaajuudelle. 512–1024 MHz kaistalle sijoittuva IF-signaali käsitellään edelleen useassa rinnakkaisessa perustaa-juudelle sekoittavaa ja alipäästösuodattavaa yksikköä. Näitä ovat joko noin 20 vuotta vanhat baseband converter -yksiköt (BBC) tai hieman uudemmat, toiminnoiltaan BBC:eita vastaavat videokonvertteri-yksiköt (video converter, VC). Sekä BBC:ssa että VC:ssa on oma taajuusgeneraattori, joka on lukittu aseman maser-kellon antamaan tarkkaan taajuusreferenssiin. Taajuusgeneraattori voidaan virittää käytännössä jollekin 550–980 MHz välillä olevalle keskitaajuudelle. Asetetun keskitaajuuden sivukaistat (upper sideband, USB, lower sideband, LSB) sekoitetaan BF-perustaa-juudelle. Valituista asetuksista riip-puen BBC kaistarajoittaa näitä molempia kaistoja passiivisilla alipäästösuotimilla, joiden

leveys on joko 1, 2, 4, 8 tai 16 MHz. Muita BBC:n toteuttamia toimintoja ovat automaattinen vahvistuksen säätö ja kaistojen kokonaistehon mittaaminen.

VLBA-järjestelmässä voidaan ohjata korkeintaan 16 perustaaajuudelle sekoitettavaa laiteyksikköä, joista kaikista saadaan USB- ja LSB-kaistat eli yhteensä 32 perustaaajuista kaistaa.

Analogisen ketjun päässä on Mark4 Sampler, joka näytteistää kaikki 32 kaistaa rinnakkaisilla A/D-muuntimilla kahden bitin resoluutiolla ja maksimissaan 32 MHz näytteenottotaajuudella. Tämä tarkoittaa enimmillään 2048 Mbit/s kokonaisnopeutta.

Jokaisen ajanhetken näytteet kaikilta kaistoilta syötetään Mark4 Formatter -laitteelle, jonka tehtävä on valita halutut kaistat ja järjestää näiden näytebitit jollain tavalla 32-bittiselle sarjamuotoiselle ulostuloväylälle, joilta ne voidaan lukea tiedonkeruujärjestelmällä kuten nauha-aseamalla tai Mark5- tai PC-EVN -koneella. Näytebittien järjestämistapoja on valitettavasti useita, koska Mark4 ottaa huomioon jo antiikkien nauha-asemien urakohtaiset luotettavuus- ja tallennusnopeusrajoitukset. Tämän takia Mark4 osaa formatoida sarjamuotoiset näytteet eri tavalla sekä rinnakkais- että sarjamuotoon. Tiedot voidaan myös levittää nauhan eri urille aikajakoisesti uraa vaihdellen. Lopputuloksena on, että samaan näytteeseen kuuluvat bitit voivat olla ajallisesti erillään ja sijaita aina vaihtelevissa kohdissa väylällä. Lisäksi jokaisen 1PPS -sekuntisignaalin kohdalla formatterisi lisää 32-bittisen ulostuloväylän jokaiseen bittivirtaan otsaketietoja. Näihin kuuluvat synkronointisignaali, aikaleima ja kunkin kanavan numero. Formatterisi voi lisätä sarjamuotoiseen näytevirtaan myös muita tietoja kuten pariteettibittejä.

Tämä Mark4:n nykyään täysin turha nauhayhteensopivuus on johtanut siihen, että VLBA-näytteenottojärjestelmän antama tietovirta on sinänsä helppo kaapata Mark5- tai PC-EVN -järjestelmällä, mutta erittäin epäedullisesti formatoitujen näytetietojen jatkokäsittely esimerkiksi ohjelmistopohjaisissa korrelaattoreissa, apuohjelmissa tai muissa analyysityökaluissa on varsin hankalaa.

VSIC-kortti osaa tulkita useita formaatteja, mukaanlukien Mark4:n käyttämiä formaatteja. Jos sisääntulevalla 32-bit väylällä on vain 8 tai 16 hyötybittiä, VSIC voi yhdistää näitä vierekkäin 32-bittiselle VSI-väylälle. Bittien formatointia VSIC ei kuitenkaan muuta, joten vaikeusaste yksittäisten kanavien purkamiselle VSI-datasta riippuu yhä suuresti formaatista. Varsin yleisesti käytetyssä formaatissa kanavien purku on kuitenkin suoraviivaista, sillä 16 kanavan 2-bittiset näytteet sijaitsevat 32-bit sanassa joko vierekkäin, tai

näiden ensimmäiset bitit 32-bittisen sanan ensimmäisessä ja toiset toisessa puoliskossa.

Tätä helpompaa formaattia käytetään hyväksi myöhemmin esiteltävässä reaaliaikaisessa e-VLBI -siirrossa, jossa verkon yli lähetettävien kanavien määrää karsitaan ja päästään näin hyödyntämään verkkoyhteyden kapasiteetti tietyissä tapauksissa optimaalisesti.



## Luku 3

### Reaaliaikaisen e-VLBI:n kehitystyöt

Tässä luvussa esitellään ne Metsähovissa tekemäni kehitystyöt, joita vaadittiin EU EXPReS -projektissa demonstroimaan PC-EVN e-VLBI:n toimivuus lokakuun 2006 Month7 Demo -testipäivänä. Näitä töitä olivat erityisesti VSIB-ajurin lähdekoodin päivittäminen Linuxin uuteen 2.6 ytimeen, valmiiksi asennetun VSIB-ajurin ja PC-EVN -apuohjelmia sisältävän Metsähovi PC-EVN Debian Linux -jakelun luonti, Tsunami-tiedonsiirto-ohjelmiston reaaliaikaversioon jatkokehittäminen, sekä apuskriptien luominen joilla voidaan generoida automaattiset e-VLBI -havainnointiskriptit. Kehitystyöstä saatuja testituloksia käsitellään seuraavassa luvussa 4, pääasiassa Month7-demonstraation lopputuloksia sekä joitakin esimerkkejä PC-EVN e-VLBI -havainnoinnin tuotantokäytöstä. Seuraavassa kuvataan suorittamani kehitystyötä tarkemmin.

#### 3.1 VSIB-ajurin lähdekoodin päivitys

Ensimmäinen Linux-ytimen versio eli kernel 2.6 julkaistiin joulukuussa 2003. Sitä seuraavan vuoden aikana 2.6-sarjan ytimet syrjäyttivät vanhat 2.4-kernelit ja siirtyivät peruserneleiksi kaikissa suuremmissa Linux-jakeluissa. Metsähovissa A. Mujusen kirjoittama VSIB-ajuri oli kirjoitettu 2.4-kerneleille. Ytimen versiovaihdoksen yhteydessä kernelin sisäiseen rajapintaan tuli useita muutoksia, joiden takia VSIB-ajuri ei enää kääntynyt uudessa kernelissä. Jotta VSIB olisi käyttökelpoinen muissa observatoriossa asennettavissa uusissa koneissa ja uusissa Linux-jakeluissa, minun oli ensi työksi portattava VSIB-ajurimoduulin lähdekoodi 2.6-kernelsarjaan. Työn alkuvaiheessa en tehnyt ajuriin toiminnallisia lisäyksiä, vaan tein ainoastaan kääntymiseen tarvittavat muutokset. Tätä varten

oli muun muassa päivitettävä moduulin headeri- ja parametritietojen määrittely vastamaan uuden kernelin vaatimaa muotoa, kuten *module\_parm* eikä *MODULE\_PARM*, ja niin edelleen. Myös ISA-väylän laitteistokeskeytyksiä väliaikaisesti sammuttavat ohjelmarivit oli poistettava. Lisäksi kernelin kutsurajapinta PCI-väylän osalta oli muuttunut. Tarvittavat VSIB-lähdekoodimuutokset olivat kuitenkin suoraviivaisesti toteutettavissa, eivätkä vaatineet ajurin uudelleenkirjoittamista. Nämä muutokset on esitetty taulukossa 3.1.

Taulukko 3.1. Uuteen 2.6 kernel-versioon portattuun VSIB-ajuriin tehdyt muutokset PCI-väylän rajapinnan osalta.

Kernel 2.4 API	Kernel 2.6 API
pcibios_present	-
pcibios_find_device(bus, devfn)	pci_get_device(device), configure_device pci_dev_put
unsigned char bus, devfn	struct pci_dev* device
pcibios_read_config_byte	pci_read_config_byte
pcibios_read_config_dword	pci_read_config_dword
pcibios_write_config_byte	pci_write_config_byte
pcibios_write_config_dword	pci_write_config_dword
check_region, request_region	request_region

Kun ajuri oli saatu toimimaan 2.6-kernelissä tein myös joitakin toiminnallisia lisäyksiä. Alun perin käyttäjän oli manuaalisesti luotava käynnistetylle ajurille Linux-laitenoodi, esimerkiksi */dev/vsib*. Laitenoodin kautta VSIB-korttia pystyy käyttämään kuin tavallista tiedostoa. Yleensä Linux-kernelin ajureissa on kuitenkin tapana, että ajurit luovat tarvittavat laitenoodit itse. Tähän on periaatteessa kaksi tapaa. Ensimmäinen on Device File System -rajapinta (Devfs). Viime vuoden 2006 virallisessa kernel-versiossa 2.6.17 alkaen tämä rajapinta tosin poistettiin, koska kernel-ylläpitäjät päättivät Devfs:n jo vanhentuneen. Toinen, aiemmin Devfs:n kanssa rinnakkainen mutta nyt siis Devfs:n täysin korvaava tapa on käyttää Sysfs-rajapintaa, jonka kautta ajuri voi rekisteröidä itsensä kuuluvan johonkin uuteen virtuaaliseen laiteluokkaan ja voi kuvata ominaisuuksiaan. Laiteluokan rekisteröimisen jälkeen laitetiedot ilmestyvät käyttäjätasoon virtuaalisen hakemiston */sys* alle. Täältä erillinen *udev*-palvelinohjelma luo automaattisesti kaikki tarvittavat laite-

noodit käyttäen järjestelmän ylläpitäjän konfiguroitavissa olevaa *udev-rules* -säännöstöä. Säännöstöön voidaan lisätä VSIB-ajuriluokka, jolle annettujen sääntöjen pohjalta *udev* luo laitenuodin */dev/vsib* ja asettaa sille sopivat käyttöoikeudet.

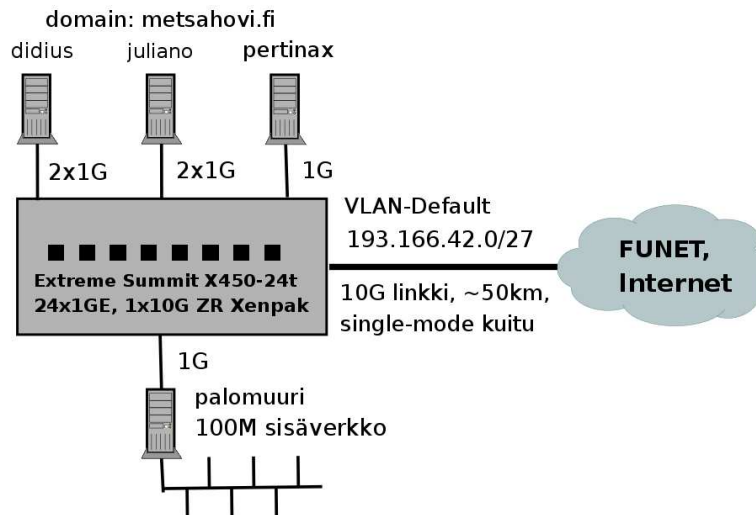
Devfs:ään verrattuna muutos koostuu siis kahdesta osasta – ensin ajuri on päivitettävä käyttämään Sysfs-rajapintaa ja lopuksi *udev*-säännöstöön on lisättävä uusi rivi tämän ajurin laiteluokalle. Aluksi Sysfs-rajapintaan päivitettäväni VSIB-ajuri luo nyt VSIB-moduulinimen avulla uuden luokan (*class\_create()*) ja lisää tähän uuden laitteen (*class\_device\_create()*). Vanhan VSIB-ajurin tapaan laite käyttää major-numeroa 254 ja minor-numeroa 0. Kun ajuri ladataan esimerkiksi tietokoneen käynnistyksen yhteydessä, säännöstöön */etc/udev/permissions.rules* tehdyn lisäyksen *KERNEL=="vsib", MODE=="0666", GROUP="root"* avulla *udev* luo automaattisesti */dev/vsib* noodin, johon kaikilla järjestelmän käyttäjillä on luku- ja kirjoitusoikeus. Koska nyt VSIB-laitenuodi luodaan aina automaattisesti, on VSIB-kortin käyttöönotto uudessa järjestelmässä aiempaa yksinkertaisempaa.

Toinen toiminnallinen lisäys oli uusi *ioctl()*-kutsu, jolla käyttäjä voi kysyä VSIB-ajurilta sen käyttämän rengaspuskurin kokoa. Jo alkuperäisessä versiossa käyttäjä pystyi toisella *ioctl()*-kutsulla kysymään rengaspuskurin täyttöastetta eli siinä olevien tavujen määrää. Lisäyksen jälkeen käyttäjä pystyy laskemaan puskurin prosentuaalisen täyttöasteen. Tätä käytetään pääasiassa Australian Long Baseline Array -observatorioissa (LBA) heidän omissa VSIB-apuohjelmissaan (Phi06).

Uuteen 2.6-kerneliin portatun VSIB-ajurin toimivuus testattiin muun muassa kuormituskokeessa, jossa kahdella PC-EVN -koneella kerättiin tietoja RAID-levylle 60 tunnin ajan. Kokeessa testattiin uuden ajurin lisäksi myös RAID-levyjä ja sähköistä häiriöalttiutta suojaamattoman, suojatun ja eri päistä maadoitetun suojatun VSI-kaapelin välillä, sekä näiden vaikutusta VSI-väylältä kaapatun testidatan oikeellisuuteen (Mol06).

### 3.2 10 Gbit/s verkkoyhteyden käyttöönotto

Metsähovin observatoriossa oli vuoden 2006 alussa vielä käytössä 2 Mbit/s ADSL-verkkoyhteys. Metsähovissa aikaisemmat e-VLBI -siirtotestit A. Mujunen ja J. Ritakari tekivät Tieteen tietotekniikan keskuksen (CSC) konehuoneesta käsin. Havainnoinnin



Kuva 3.1. Verkkotopologia Metsähovin 10 Gbit/s yhteydelle

tiedot tallennettiin Metsähovissa RAID-levyille, jonka jälkeen kuljetettiin koko tietokoneet autolla CSC:n konehuoneeseen ja kytkettiin runkoverkkoon. Muita siirtotestejä he ajoivat etänä toisten observatorioiden tietokoneilta.

Reaaliaikaista e-VLBI:tä varten Metsähoviin oli kuitenkin saatava suora, vähintään 630 Mbit/s verkkoyhteys. Tämä onnistui edullisesti EXPReS-projektin puitteissa – projektissa EU rahoittaa muun kehitystyön ja laitteiston ohella myös EVN:n teleskooppien kytkennän Internetiin nopealla kuituyhteydellä. Ritakarín ja Mujusen tekemän laitteistonselvityksen, avoimen tarjouskilpailun ja verkkosopimusten jälkeen he saivat hankittua Metsähoville kustannustehokkaan verkkokytkimen kuituoptiikkoineen ja 10 Gbit/s kuituyhteyden FUNET:in runkoverkkoon. Muilla observatorioilla oli samaan aikaan korkeintaan 2.5 Gbit/s yhteydet. Metsähovi oli näin maailman ensimmäinen radio-observatorio 10 Gbit/s yhteydellä (RiM06).

Kuituyhteys päättyy Metsähovin puolella Extreme Networks Summit X450-24t -kytkimen 10 Gbit/s ZR Xenpak -kuitumoduuliin. Kytkimessä on kahden 10G Xenpak-moduulipaikan lisäksi 24 1G Ethernet-porttia, joihin yhdistimme myöhemmin useita uusia PC-EVN -testikoneita. Kytkimen asentaminen ja asetusten tekeminen vei Metsähovin T. Lindforsilta ja minulta noin kaksi työpäivää, testikoneiden rakentaminen ja ohjelmistoa-sennukset vielä jonkin verran kauemmin. Kytkimeen aluksi tehdyssä konfiguraatiossa ja-  
oimme Metsähovin 32 IP-osoitetta sisältävän julkisen IP-osoitealueen *193.166.42.0/27*

pienempiin Virtual LAN (VLAN) -alueisiin. VLAN:eilla on joitakin tietoturvan ja reitityksen kannalta hyviä ominaisuuksia. Kytkimeen konfiguroitiin useita VLAN:eja eri näihin lisättävien koneiden eri käyttötarkoituksiin perustuvalla jaottelukriteerillä. Yksi kahden osoitteen VLAN luotiin kuitulinkille, toinen 14 osoitteen VLAN e-VLBI -testikoneille, kolmas VLAN sisälsi palomuurin lähiverkkoon, neljäs julkisen www-palvelimen. Myöhemmin kuitenkin jouduimme toteamaan, että testikone-VLAN:in oli kerääntynyt jo monta konetta ja sen 14 osoitetta olivat alkuperäisten odotusten vastaisesti käymässä pian vähiin. Tämän lisäksi useaan VLAN:iin jakamisen takia 32:sta julkisesta IP-osoitteita kahdeksaa ei voitu käyttää, koska nämä olivat VLAN:ien gateway- ja broadcast-osoitteita. Lopullinen ratkaisu oli luopua muista VLAN:eista ja koota kaikki osoitteet takaisin yhteen VLAN:iin, johon kuuluvat kaikki testikoneet ja muut palvelimet.

Verkon lopullinen topologia on esitetty kuvassa 3.1. Kuvassa merkintä 2x1G tarkoittaa, että kyseisellä koneella on kaksi 1 Gbit/s verkkokorttia. Kaikki kuvan PC-EVN -koneet ovat kuluttajätietokoneita, joissa molemmat verkkokortit ovat emolevyllä integroituja, joko osana piirisarjaa tai erillisenä piirinä. Kortit voidaan Linuxissa yhdistää yhdeksi virtuaaliseksi verkkokortiksi, jolla voidaan teoriassa saavuttaa lähes kaksinkertainen nopeus (link aggregation, IEEE 802.3ad). Toimiakseen tämä vaatii vastaavat asetukset myös X450-kytkimessä, jossa tietokoneen kytkimessä käyttämät portit on lisättävä omaan kuormatasapainoitusryhmään.

Verkkoyhteyden nopeus Metsähovin ja ulkomaailman välillä testattiin lyhyesti iperftestiohjelmalla. Iperf on ohjelma, jolla voidaan mitata kahden tietokoneen välisen verkkoyhteyden nopeus sekä TCP- että UDP-protokollille. Funetilla on oma iperf-palvelin, jota käytimme UDP iperf -testissä. Testi osoitti, että yhden Metsähovi-koneen ja Funet-testikoneen välillä saavutetaan vaivatta lähes 1 Gbit/s nopeus.

Vastaavat iperf-testit ajettiin myös niiden testikoneiden välillä, joilla oli kaksi verkkokorttia kytkimen kuormatasapainoitusryhmässä. UDP-siirrot toimivat noin 1500 Mbit/s nopeudella. Näin ainakin periaatteessa olisi mahdollista siirtää yhdestä tietokoneesta hie- man yli 1 Gbit/s nopeudella tietoja Internetiin.

Täyttä 10 Gbit/s kapasiteettia ei kuitenkaan testattu, koska ei ollut saatavilla muita etäkoneita vähintään 1 Gbit/s yhteydellä ja iperf-palvelulla. Funet oli myös ilmoittanut 2.5 Gbit/s pullonkaulasta ulkomaille ja antanut luvan korkeintaan 1.5 Gbit/s siirroille. Pullonkaulan luvattiin poistuvan lopullisesti vasta vuoden 2007 alussa, pari kuukautta

Metsähovin 10 Gbit/s verkon asentamisen jälkeen.

### 3.3 Tsunami UDP -protokolla

Tsunami on Indianan Yliopiston Pervasive Technology Labs -tutkimuskeskuksessa kehitetty nopea aggressiivinen tiedostonsiirtoprotokolla, jota voidaan tietyin rajoituksin käyttää myös tiedonsiirtoprotokollana. Tsunami toimii täysin käyttäjäympäristössä tavallisen käyttäjän oikeuksilla eikä tarvitse muutoksia Linux-kerneliin, toisin kuin TCP/IP-muunnelmat. Siirrettävät tiedot lähetetään UDP/IP-paketeissa, joiden perilletuloa verkko ei takaa, mutta joiden siirtonopeus on huomattavasti TCP-protokollaa suurempi, erityisesti pitkällä etäisyyksillä. Kaikille Tsunamin kaltaisille UDP-pohjaisille tiedonsiirtoprotokollille on ominaista, että siirtonopeudet ovat helposti yli kymmenen kertaa suurempia kuin TCP:llä, koska ne pystyvät hyödyntämään verkon vapaata kaistaa huomattavasti paremmin. TCP:llä on myös muita puutteita, jotka tekevät sen täysin sopimattomaksi mihin tahansa reaaliaikaiseen tiedonsiirtoon. Siirtonopeutta ei voida hallita eikä reaaliaikasiirtojen aikarajoissa pysyminenäkään ole mahdollista. Edellisten takia VLBI:ssä reaaliaikakäyttöön on valittava UDP-pohjainen siirtoprotokolla.

Indianan Yliopisto julkaisi Tsunami-ohjelmiston erittäin kokeellisen version vuonna 2002 lähdekoodimuodossa. Tsunamin etu muihin vastaavanlaisiin UDP-siirtoprotokolliin (QUANTA, SABUL, UDT) verrattuna ovat sen mukana tulevat palvelin- ja asiakasohjelmat, joilla on helppokäyttöinen FTP:n kaltainen käyttäjäkomentorivi.

Koska Indianan Yliopisto ei ensimmäisen julkaisun jälkeen enää päivittänyt Tsunamia, useat muut tahot ovat tehneet siihen omia lisäyksiä ja korjauksia. Suurimmat korjaukset tein osana tätä diplomityötä. Nyt ajan tasalla oleva Tsunami on vapaasti ladattavissa SourceForge-sivustolla ylläpitämästäni *Tsunami UDP* -projektista. Projekti sisältää sekä uusimman lähdekoodin että uudistetun dokumentaation (Wag06-1).

Seuraavaksi esitellään Tsunami-protokollan yleinen toiminta ja alkuperäisen version oleelliset puutteet. Alkuperäisen Tsunami-protokollan toiminnasta löytyy tarkempaa tietoa M. Meissin julkaisusta (Mei02). Tämän lisäksi joitakin Tsunami-protokollan perusominaisuuksia on lueteltu taulukossa 3.2, kiinnostunut lukija voi vertailla näitä muiden tuntemiensa protokollien ominaisuuksiin.

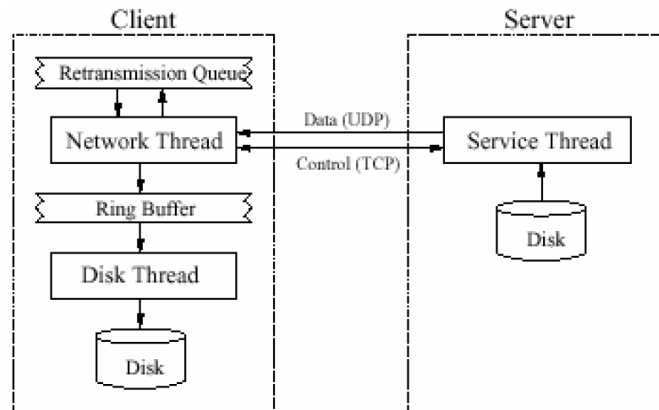
Taulukko 3.2. Tsunami-protokollan ominaisuudet.

Yhteys	TCP-hallintayhteys asiakkaalta palvelimelle
Säikeistys	asiakassäikeet palvelimessa
Autentikointi	MD5-tiiviste (salasana ja palvelimen salt)
Asetusten hallinta	asiakasohjelman määrittelee siirron asetukset
Tiedostonsiirto	yksisuuntainen GET, tiedosto siirtyy lohkoina
Tiedonsiirtomoodit	häviötön, aika rajoitettu lähes häviötön, häviöllinen
Tiedonsiirtonopeus	UDP-lähetysnopeus asiakkaan asetusten ja raporttien mukaan
Ruuhkautumisen hallinta	lähetyksenopeuden asymptoottinen hidastaminen
Sallittu pakettihäviö	käyttäjän asettama, oletusarvo 7%
Häviöikkuna	korkeintaan 2048 pakettia, aikaikkuna 50 pakettia tai 350ms
Yhteysraportit	yksisuuntainen, asiakas raportoi palvelimelle
Sietokyky	pakettihäviö, monistuneet paketit, uudelleenjärjestäminen
Käyttäjäraportointi	ruutuloki, bittikarttatiedosto

Tsunamin perusajatus on siirrettävän tiedoston pilkkominen isoihin, samankokoisiin lohkoihin. Lohkoille annetaan järjestysnumero  $B_{0...N}$ . Jokainen lohko lähetetään erillisen tunnistetietokentän kanssa asiakkaalle tietyllä tavoitenopeudella, UDP-pakettien muodossa. Oletusarvoisesti lohkojen koko  $L_{block}$  on 32 kB ja tavoitenopeus  $R_{target}$  650 Mbit/s. Suorituskyvyn kannalta on huomattava, että mikäli lohkon sisältävä UDP-paketti ylittää siirtoreitin suurimman sallitun yksikön (maximum transfer unit, MTU), lähetettävät UDP-paketit pilkkoutuvat verkkokortin tai kortin ajurin toimesta siirtoreitin sallimaan MTU-kokoon. Tämä saattaa kasvattaa prosessorikuormaa. Lohkon kokoasetus on rajoitettu noin 64 kB arvoon, sillä UDP/IP protokollan määritysten mukaan UDP-paketille suurin sallittu koko on 64kB.

Edellä mainittu Tsunamin UDP-paketeissa käyttämä tunnistetietokenttä koostuu paketin sisältämästä lohkon järjestysnumerosta ja lohkon tyypistä. Lohko voi olla tyypiltään joko palvelimen ensimmäistä kertaa lähettämä lohko, uudelleen lähetetty lohko tai erikoinen siirron päätelohko. Lohkon tyyppitietoa hyödynnetään erityisesti asiakasohjelman laatimissa siirtotilastoissa.

Tsunamin yleisrakenne on esitetty kuvassa 3.2. Tiedonsiirron eteneminen ja häviöttömyys ovat lähes täysin asiakkaan hallinnassa. Jos asiakas huomaa saapuvien peräkkäisten loh-



Kuva 3.2. Tsunami UDP -protokollan asiakas- ja palvelinohjelmien rakenne. Asiakas ja palvelin kommunikoivat TCP-yhteyden yli (Control). Palvelin lähettää kovalevyllä (Disk) olevan tiedoston numeroitujen UDP-pakettien sarjana (Data). Asiakas kirjaa kadonneet paketit (Retransmission Queue). Tietojen tallentaminen tapahtuu toisessa ohjelmasäikeessä (Disk Thread), erillään verkkoliikenteen käsittelystä (Network Thread) jaetun muistipuskuria (Ring Buffer) kautta (Mei02).

kojen joukosta UDP-siirrossa kadonneen lohkon eli puuttuvan järjestysnumeron, asiakas voi pyytää tai olla pyytämättä palvelinta lähettämään puuttuva lohko uudestaan. Toisin kuin TCP:ssä, siirrossa ei siis käytetä vastaanottovahvistusta. Näin palvelinohjelma on varsin kevyt eikä sisällä esimerkiksi ikkunointia, vuonohjausta tai ruuhkanhallintaa, ja siirron ominaisuudet voidaan sovittaa täysin käyttäjän tarpeisiin. Tämä ei ole mahdollista TCP-protokollalla.

Kuvataan seuraavaksi palvelinohjelman toiminta. Tsunamin asiakasohjelma avaa aluksi TCP-yhteyden palvelinohjelmaan, ja autentikoituu etukäteen tunnetulla salasanalla (oletusarvoisesti "kitten"). Palvelinohjelma tarkistaa salasanan ja vertaa oman ja asiakkaan protokollarevisioita. Onnistuneen yhdistämisen jälkeen asiakasohjelma voi aloittaa uuden tiedostosiirtosession GET-komennolla. Tässä asiakas välittää aluksi palvelimelle kaikki käyttäjäasetukset – lohkojen koon, siirtomoodin, tavoitenopeuden, ja muut asetukset – sekä lähettää halutun tiedoston nimen. Palvelin palauttaa asiakkaalle siirrettävän tiedoston pituuden tavuina.

TCP-yhteyden yli tapahtuneen alustuksen jälkeen liikenne palvelimelta asiakkaalle koostuu ainoastaan UDP-paketeista, joiden lähetyksen palvelin aloittaa heti. Tämä UDP-



pakettivirta voidaan mieltää kahdeksi eri virraksi, joista ensimmäisessä palvelin lähettää tiedoston kaikki lohkot järjestyksessä asiakkaalle. Toisessa virrassa on asiakkaan satunnaisesti uudelleen pyytämiä, ei välttämättä peräkkäisiä lohkoja. Ajanhetken  $t$  lähetyksenopeus  $R_{actual}(t)$  palvelimen puolella koostuu siis kahdesta osasta  $R_{sequential}(t)$  sekä  $R_{retransmissions}(t)$ :

$$R_{actual}(t) = R_{sequential}(t) + R_{retransmissions}(t) \quad (3.1)$$

Jokaisen lähtetyn lohkon jälkeen palvelin käsittelee TCP-yhteyden yli asiakkaalta mahdollisesti tulleet palautteet. Näitä ovat virheraportit, uudelleenlähetyspyynnöt ja siirron uudelleenaloituspyynnöt.

Asiakkaan virheraportti sisältää asiakkaan koostaman yksittäisen kertoimen, lohkojen ajallisesti suodatetun häviöprosentin, jonka perusteella palvelin laskee seuraavan lähetyksenopeuden säätöön käytetyn kertoimen  $r(t)$  (0—1.0). Kerroin pienenee eksponentiaalisesti, jos häviöprosentti kasvaa yli käyttäjän salliman häviöprosentin. Säätökerroin  $r(t)$  käyttäytyy kuitenkin maltillisemmin kuin TCP-protokollan vastaavat ruuhkanhallintamekanismit kuten esimerkiksi Reno, Cubic tai BIC. Sen avulla yritetään pitää häviöprosenttia korkeintaan asiakkaan asetuksen mukaisena säätämällä ajanhetken  $t$  kokonaissiirtonopeutta  $R_{actual}(t)$  hitaammaksi tai kasvattamalla sitä takaisin kohti tavoitenopeutta  $R_{target}$ :

$$R_{actual}(t) = R_{sequential}(t) + R_{retransmissions}(t) = r(t) \cdot R_{target} \quad (3.2)$$

Palvelin rajoittaa lähetyksenopeutta nopeuteen  $R_{actual}(t)$  odottamalla tarvittava aika  $\tau_{IPD}(t)$  (inter-packet delay, IPD) aiemman ja seuraavan lähetettävän lohkon välillä:

$$\tau_{IPD}(t) = \frac{L_{block} \cdot 8bit/byte}{R_{actual}(t)} \quad (3.3)$$

Uudelleenlähetyspyynnössä asiakkaan luetteloimat lohkot luetaan uudestaan tiedostosta ja lähetetään tyyppin ”uudelleenlähetetty”lohkona.

Uudelleenaloituspyynnössä palvelin taas peruuttaa lohkojärjestyksen mukaan etenevässä UDP-pakettivirrassa takaisin pyynnössä annettuun lohkonumeroon ja aloittaa koko lähetyksen uudestaan tästä lohkoista.

Kaikkien palautetietojen prosessoinnin jälkeen palvelin jatkaa tiedoston seuraavien lohkojen peräkkäistä lähettämistä. Lopussa, kun palvelin on lähettänyt viimeisen lohkon, se jää lähettämään erikoista lopetuslohkoa, kunnes asiakas lopettaa tiedostosiirtosession TCP:n yli annetulla lopetuspyynnöllä.

Asiakasohjelman puolella tiedostonsiirto etenee seuraavasti. Asiakkaassa on vastaanottosilmukka, jossa odotetaan ja käsitellään sisääntulevia UDP-paketteja. Vastaanotetut UDP-paketit prosessoidaan tuottaja-kuluttaja -periaatteella – pääprosessi tuottaa verkosta saadut UDP-paketit rengaspuskuriin, jota ylläpitää vastaanottosilmukasta erillään suorittuva I/O-säie. I/O-säie lukee rengaspuskuriinsa ilmestyvät UDP-paketit ja purkaa niistä lohkonumeron ja hyötydatan. Lohkon järjestysnumeron perusteella merkitään bittikarttaan, että lohko on jo vastaanotettu eikä enää puutu. Lopuksi lohkon tiedot kirjoitetaan ulostulotiedostoon, lohkon numeron määräämään kohtaan.

Asiakasohjelman on huolehdittava siirron häviöttömyydestä, ainakin mikäli häviötön siirto vastaa käyttäjän tarpeita. TCP-tiedonsiirto on aina häviötön, kun taas UDP-pakettien siirrossa voi ilmaantua tiettyjä ongelmia. Pakettien saapumisjärjestys voi olla toinen kuin lähetysjärjestys, tai UDP-paketit saattavat eri syistä kadota kokonaan. Tavallisesti UDP-pakettien katoamien johtuu odotetun saapumisajan aliarvioimisesta (paketti saapuu perille paljon myöhemmin kuin odotettu), verkon ylikuormittumisesta jossakin solmupisteessä, lähettävässä päässä olevan kytkimen vastaanottopuskurin ylivuodosta, johtuen lähettävän koneen mikropurskeisesta liikenteestä, tai jopa vastaanottokoneessa liian hitaan asiakasohjelman, tietokoneen tai IP-defragmentaation prosessorikuorman takia.

Jotta tavalla tai toisella kadonneet lohkot voidaan pyytää uudelleen, asiakas pitää kirjata viimeisestä lohkoista, johon asti tiedosto on jo aukottomasti vastaanotettu, ja odottaa tätä seuraavaa lohkoanumeroa *next\_block*. Jos vastaanotetun UDP-paketin sisältämä lohkonumero on tätä suurempi, väliin jäävien lohkojen numerot lisätään uudelleenlähetyslistaan. Lopuksi UDP-paketti siirretään prosessoitavaksi I/O-säikeen rengaspuskuriin.

Uudelleenlähetyslistan maksimipituus  $L_{retxlist}$  on oletusarvoisesti 2048 lohkoa. Vähintään jokaisen 50 vastaanotetun UDP-paketin mutta enintään 350ms aikavälin jälkeen asiakas käy läpi tähän mennessä kertynyttä listaa. 50 pakettia vastaa 32 kB pakettikoolla ja 512 Mbit/s nopeudella noin 3ms viivettä. Listan keräämisaikaa ennen sen lähettämistä palvelimelle merkitään myöhemmin  $\tau_{RP}$ . Jotta turhilta uudelleenlähetyksiltä voidaan välttyä, ennen listan lähettämistä siitä karsitaan väliaikana jo saapuneet lohkot, siis ne lohkot, jot-

ka eivät tosiasiasa kadonneet, vaan saapuivat vain väärässä järjestyksessä tai eri viiveellä. Karsinnan jälkeen listassa olevat lohkonumerot lähetetään uudelleenlähetyspyyntönä palvelimelle.

Jos kuitenkin lista ylittää maksimipituuden  $L_{retxlist}$ , asiakas välittää palvelimelle lähetyksen uudelleenaloituspyynnön.

Näillä menetelmillä asiakas pyrkii siis vastaanottamaan tiedosto siten, että se on tiettyyn lohkoon  $B_{gapless}$  asti aukoton. Tämän lohkon  $B_{gapless}$  ja viimeisen vastaanotetun lohkon välissä saa olla aukollista aluetta, jossa on yhteensä korkeintaan  $L_{retxlist}$  eli 2048 puuttuvaa lohkoa. Uudelleen lähetettävien UDP-pakettien ja etenevässä järjestyksessä lähetettyjen UDP-pakettien lukumäärien suhde pysyy tällä tavalla pienenä. Näin uudelleenlähetysten aiheuttama prosessorikuorma minimoituu ja tiedonsiirto etenee huomattavasti tehokkaammin.

Tsunami-protokollan alkuperäisessä versiossa oli useita puutteita. Osa näistä oli korjattu Jamil Zamanin (Australian CSIRO) levittämässä Tsunami-versiossa, joka tosin käyttää liikaa muistia, sisältää tarpeettoman paljon debug-tulostuksia ja lukuisia uusia virheitä. Ritakari oli luonut 2005 molemmista versioista reaaliaikaiset asiakas- ja palvelinohjelmat tekemällä näihin kaksi lisäystä. Ensiksi aidon tiedoston sijaan käytetään VSIB-kortin merkkioheislaitetta `/dev/vsib` lohkojen lukemiseen tai kirjoittamiseen. Toiseksi asiakas voi tiedostonimessä kertoa palvelimelle muun muassa oikean aloitusajankohdan näytteiden keruulle.

Tsunamin jatkokehitystyöissä käytin pohjana näitä reaaliaikaversioita ja Tsunamin alkuperäistä Indianan Yliopiston versiota. Tekemäni korjaukset ja toiminnalliset lisäykset on dokumentoitu tietyssä mielessä tarkemmin SourceForge-projektisivun ChangeLog-tiedostossa ja CVS lähdekoodin historiatiedoissa. Lukuisat korjaukset olivat enimmäkseen lyhyitä, joten niitä ei esitellä tässä työssä sen tarkemmin. Alla kuvataan laajimmat ja tärkeimmät Tsunami-ohjelmiin tekemäni muutokset.

### 3.3.1 Lähetysten uudelleenaloitus

Eräs korjattava ongelma oli Tsunamin lähetyksen uudelleenaloitusmenetelmän periaatteellinen toimimattomuus. Metsähovin 10G testiverkossa tämä ilmeni 950 Mbit/s lähes tyvillä lähetyksnopeuksilla. Tässä koneiden välillä koko tiedostonsiirto saattoi jo pienellä pakettihäviöllä pysähtyä niin, että siirron valmistumisaste jäi joko värähtelemään kahden

tason välille tai ei edennyt ollenkaan. Tämä ilmeni myös huomattavasti hitaammilla siirtonepeuksilla tekemissäni pidemmän reitin siirtokokeissa muun muassa Kashiman (Japani) ja Jodrell Bankin (UK) kanssa kesällä 2006. Syyksi paljastui, että siirtoreitille Tsunami-yhteyden kiertoviiveen  $\tau_{RTT}$  (round-trip time, RTT) aikana mahtuvien lohkojen määrä ylitti uudelleenlähetyslistan maksimipituuden puolikkaan  $\frac{1}{2}L_{retxlist}$ , toisin sanoen:

$$\frac{R_{actual}(t) * \tau_{RTT}}{L_{block} * 8bit/byte} > \frac{1}{2}L_{retxlist} \quad (3.4)$$

Ennen kuin palvelin vastaanottaa ja toteuttaa uudelleenaloituspyynnön, verkossa kulkee joka tapauksessa uudelleenaloituspyynnön aiheuttaneen lohkon perässä useita mahdollisesti vielä perille saapuvia lohkoja, joilla on suurempi lohkonumero. Alkuperäisessä Tsunamissa jokainen näistä lohkoista aiheutti edeltä yhä puuttuvien lohkojen takia toistetun uudelleenaloituspyynnön. Tämä tukki palvelimen turhilla uudelleenaloituspyynnöillä ja siirto joko hidastui huomattavasti tai pysähtyi kokonaan.

Luotettavasti toimiva korjaus onnistui vain kirjoittamalla uudelleenaloitustoteutus osittain uusiksi. Kun uudelleenaloituspyyntö laaditaan asiakaspuolella (*protocol.c:tcp\_repeat\_retransmit*), asetetaan *restart\_pending* -lippu ja kirjataan viimeisen puuttuvan lohkon numero *restart\_lastidx*. Vastaanottosilmukassa (*command.c:command\_get*) *restart\_pending* -moodissa rajoitutaan vastaanottamaan vain tiedoston ensimmäisen ja *restart\_lastidx* -lohkon välinen alue täydellisesti. Tätä uudemmat linjalla vielä kulkevat lohkot jätetään prosessoimatta – suoraviivaisin joskaan ei kaikkein optimaalisin toteutus. Kun viimeinen puuttuva lohko *restart\_lastidx* on vastaanotettu, lippu resetoitetaan ja siirto jatkuu tavalliseen tapaan, ilman vastaanottoalueen rajoitusta.

### 3.3.2 Rinnakkaisten Tsunami-asiakkaiden ajo

Toinen ongelma Tsunamissa oli, että asiakas merkitsi UDP-socketin uudelleenkäytettäväksi (*SO\_REUSEADDR* socket-optio, *network.c:create\_udp\_socket*). Tämä esti usean rinnakkaisen asiakkaan käynnistämisen samalla koneella. VLBI-tiedostonsiirrossa on usein kätevää hakea tiedostot rinnakkain eri observatorioilta, jos vastaanottavan tietokoneen RAID-levyt ja verkkoyhteyden kapasiteetti tämän sallivat.

Jotta rinnakkainen siirto toimisi, oli ensiksi poistettava *SO\_REUSEADDR* optio. Ensimmäinen käynnistetty asiakasohjelma varaa vastaanottavan UDP-portin, joka oletusar-

voisesti on 46224. Seuraavan asiakasohjelman on käytettävä eri porttia ja aiemmin käyttäjä joutui valitsemaan tämän manuaalisesti. Nyt oletusportin avaamisen epäonnistuessa asiakasohjelma yrittää varata seuraavasta kuudestatoista portista ensimmäisen vapaan ja tulostaa varoituksen että useita muita, ehkä eri käyttäjän käynnistämää asiakasohjelmaa oli jo käynnissä. Jos käytössä on palomuri, sen asetuksissa on otettava huomioon Tsunami-asiakkaan kokeilema laajempi porttialue.

### 3.3.3 Käyttäjäraportointi

Asiakasohjelma myös laski esimerkiksi käyttäjäraporttiin tulostuvat tilastot osittain väärin. Tämä vaikutti jonkin verran virhekertoimeen, jonka asiakas säännöllisesti lähettää palvelimelle virheraportin sisällä. Asiakkaan virheellisen tilastoinnin vaikutus palvelimen tekemään siirtonopeuden säätöön oli kuitenkin marginaalinen. Virheelliset käyttäjäraportit sen sijaan olivat suurempi haitta, myös koska laatimallani Matlab-ohjelmalla tehdyt tiedonsiirron eri kuvaajat olivat tästä johtuen virheellisiä. Korjausta varten kirjoitin käytännössä koko tilastoinnin uudestaan (*protocol.c:ttp\_update\_stats*).

Laajensin raportointia myös uudella *blockdump* -asetuksella, jolla siirron päätteeksi kirjoitetaan erilliseen tiedostoon aiemmin mainittu bittikartta, jolla on pidetty kirjaa asiakkaan vastaanottamista lohkoista. Tavallisessa häviöttömässä siirrosta tallennettavasta bittikartasta ei ole paljon hyötyä, koska se sisältää vain 1-bittejä. Jos kuitenkin käytetään kapaleen 3.4.2 potentiaalisesti häviöllistä siirtoa, skanneja jatkokäsittelevät ohjelmat voivat tallennetusta bittikartasta tarkistaa helposti, mitkä osat tiedostosta puuttuvat. Tämä on huomattavasti nopeampaa kuin giga- tai teratavun kokoisen tiedoston sisällön tutkiminen.

Toiminnolle ei ole vielä löytynyt käyttöä, koska lähes kaikki VLBI-tiedostojen siirrot ja reaaliaikasiirrot on tähän mennessä suoritettu häviöttöminä. Lisäksi korrelaattorin yläpitäjien on vielä päästävä yksimielisyyteen siitä, millä tavalla hävinneet tiedot tulisi parhaiten korvata. Bittikartta tulee siis hyötykäyttöön vasta tulevaisuudessa.

### 3.3.4 Muut toiminnalliset lisäykset

VLBI-siirroissa asiakasohjelmaa käytetään mieluusti automaattisten skriptien kautta, mutta alkuperäistä asiakasta pystyttiin käyttämään ainoastaan sen oma sisäänrakennetun komentorivin yli eikä sitä siis voitu millään kovin kätevällä tavalla kutsua

skripteistä. Tämän uuden toiminnallisuuden lisääminen oli suoraviivaista. Nyt samat Tsunami-komentorivin komennot voidaan välittää ohjelmalle myös kutsuparametreissa (*main.c:main*). Tätä käytetään esimerkiksi reaaliaikaisen e-VLBI -havainnoinnin skripteissä, joita kuvataan tarkemmin kappaleessa 3.5.

Aiemmin salasana jolla palvelin todentaa asiakkaan oli joko kovakoodattu asiakkaaseen tai asiakasohjelma kysyi salasanaa käyttäjältä. Tsunami-palvelimen komentoriviparametreissa on mahdollista vaihtaa oletussalasana, joskaan mikään asema ei vielä ole yrittänyt vaihtaakaan salasanaa. Lisäsin asiakasohjelmaan kuitenkin tulevaisuuden varalle uuden asetuksen, jonka avulla asiakkaan kokeilema salasana voidaan asettaa myös esimerkiksi automatisointiskripteistä käsin.

Tiedostopalvelimessa uusi toiminto ovat komentorivillä annettavat eksplisiittisesti jaettavat tiedostot. Asiakasohjelma voi vuorostaan suorittaa *GET \** -käslyn ja ladata peräjälkeen kaikki näin jaetut tiedostot. Jo J. Zamanin muokkaamassa Tsunami-versiossa oli vastaava toiminto, mutta se sisälsi useita virheitä ja rajoituksia. Esimerkiksi Bonnin korrelaattoriasema huomasi, että tavallisen VLBI-kokeen yli tuhannelle skannitiedostolle yritetty *GET \** kaatoi asiakkaan. Näiden takia kirjoitin toiminnoille täysin uuden toteutuksen. Lisäsin asiakkaaseen myös uuden käslyn *DIR*, jolla voidaan luetteloida palvelimen jakamat tiedostot ja näiden koot.

### 3.4 Reaaliaikainen Tsunami-protokolla

Tsunamiasta on olemassa aiemmin mainittu Ritakaran 2005 laatima versio, joka soveltuu reaaliaikaiseen VSI-näytteiden siirtoon. Reaaliaikainen palvelin ja asiakasohjelma eroavat tiedostoversioista kahdessa kohtaa. Ensinnäkin reaaliaikaversiot käyttävät kovalevyllä olevan tiedoston sijaan VSIB-kortin rengaspuskuria. Toisekseen vaikka siirtopyynnöissä käytetään yhä tiedostonimiä, varsinaista lähetettävää tiedostoa ei ole olemassa levyllä. Sen sijaan palvelinohjelma voi tulkita tiedostonimestä reaaliaikaisiirrolle toivotun aloitusajankohdan ja keston.

Protokolla jota reaaliaikainen ja tavallinen Tsunami käyttävät on sama, minkä takia on mahdollista käyttää ja yleensä käytetäänkin molempia versioita ristikkäin. Eri käyttötavat on havainnollistettu taulukossa 3.3. Tällä hetkellä (2007) yleisimmät käyttötavat ovat

ei-reaaliaikainen off-line VLBI ja reaaliaikainen e-VLBI tiedonkeruumoodi, jossa asiakas kirjoittaa reaaliaikaiset näytteet tiedostoon. Täysin reaaliaikainen e-VLBI siirto aseman VSI-väylältä verkon yli korrelaatoriaseman VSI-väylälle saattaa yleistyä, kun laitteistopohjaiset korrelaattorit joutuvat tukemaan VSI-väylää paremmin Haystack Mark5 VSI-laiteversioiden yleistyessä. Tässä ajetaan molemmissa päässä Tsunamin reaaliaikaversiota. Ohjelmistopohjaisissa korrelaattoreissa taas riittää Tsunamin reaaliaikainen palvelin-versio.

Taulukko 3.3. Tavallisen ja reaaliaikaisen Tsunamin yhteiskäyttö ja vastaavat VLBI-tiedonsiirtomoodit.

Asiakas	Palvelin: tavallinen	reaaliaika
tavallinen	tiedostonsiirto, off-line e-VLBI	reaaliaikainen e-VLBI tiedonkeruu
reaaliaika	e-VLBI tiedontoisto	reaaliaikainen VSI-VSI siltaus

Tavallisesti reaaliaikaisuudesta puhuttaessa määritellään mitkä ovat aikarajoitteet reaaliaikaisuudelle. e-VLBI:ssä tällaisia reaaliaikakriteerejä ei kuitenkaan ole suoraan annettu. Nämä määräytyvät epäsuorasti vastaanottopuolella suoritettavan laskennan eli korreloinnin kautta, jossa prosessoidaan aina saman ajanhetken näytteet asemilta tulevista tietovirroista. Asemien tiedonkeruujärjestelmien ja erityisesti verkon tiedonsiirron viiveet on siis sovitettava asemavirtojen kesken samoiksi joko lähetys- tai vastaanottopuolella. Viiveet ovat kymmenien tai satojen millisekuntien luokkaa. Yleensä sovitusta toteutetaan nopeilla muistipuskureilla korrelointiaseman puolella.

Koska korrelaattoriasemassa puskurointikapasiteetti on aina rajallinen ja tietyn aseman puskurin ylivuoto tarkoittaa aseman tippumista VLBI-kokeesta, e-VLBI on implisiittisesti kovan reaaliajan järjestelmä. Toisaalta koska viivesovitusta ei toteuteta observatorioissa tai tiedonsiirtoprotokollissa, ei reaaliaikaisessa Tsunami-versiossa ainakaan tällä hetkellä jouduta ottamaan huomioon viiveiden sovitusta ja sen ongelmia.

Tsunamin reaaliaikaohjelmista yleisimmin käytetty on reaaliaikapalvelin. Asiakasohjelmalle on ollut konseptidemonstraatioiden jälkeen tuskin lainkaan käyttöä, koska laitteistokorrelaattoreita ei tähän mennessä vielä pystytä liittämään VSI-väylään ja observatorioissa tarvitaan tiedonkeruuta eikä -toistoa. Myös reaaliaikaisen asiakasohjelman tarvitsema toistava versio VSIB-kortista, jossa piirilevyllä VSI-väylän LVDS-vastaanotto piirit on korvattu LVDS-ajureilla, on harvinainen korttimalli.

EXPreS-töissä on käytetty vain reaaliaikaista palvelinta, ei asiakasta, joten tämän kapaleen kuvaukset liittyvät vain palvelimeen ja tavalliseen VSI-väylältä lukevaan VSIB-korttiversioon.

Toisin kuin kaavan 3.2 tiedostosta luettaessa vaihteleva  $R_{sequential}(t)$ , reaaliaikaisessa Tsunami-palvelimessa näiden uusien UDP-pakettien peräkkäinen lähetysnopeus rajoittuu automaattisesti VSI-väylältä tulevien näytteiden omaan vakionopeuteen  $R_{VSI}$ , koska VSIB-kortin tiedonkeruu on väylän kellottama eikä näytteitä pystytä lukemaan nopeammin kuin niitä saapuu väylältä.

Tsunamin ruuhkanhallinta- ja nopeudensäätömekanismi taas yrittää pitää UDP-pakettien lähetysnopeutta asiakkaan määräämässä tavoitenopeudessa  $R_{target}$ . Reaaliaikaisuuden perusteella on valittava asiakkaassa  $R_{target}$ -asetukseksi jokin sopiva ja riittävästi marginaalia sisältävä  $R_{target}$ ,  $R_{target} > R_{VSI}$ , esimerkiksi asettamalla tavoite 100 Mbit/s verran korkeammaksi kuin  $R_{VSI}$ . Nopeusmarginaali sisältää uudelleenlähetysten UDP-pakettivirran ja Tsunamin oletusarvoisen 650 Mbit/s tavoitenopeuden kanssa marginaali on ilman lisäasetuksia riittävä myös 512 Mbit/s  $R_{VSI}$  siirroille. Vain suuremmilla nopeuksilla täytyy muistaa muuttaa oletusarvoa.

Seuraava ero tavallisen tiedostosiiirtoon käytetyn Tsunami-palvelimen ja reaaliaikaisen palvelimen välillä on tarve välittää palvelimelle oikea tiedonkeruun aloitusajankohta ja kesto. Nämä ovat etukäteen tunnettuja VLBI-kokeen yhteisestä aikataulusta luoduista asemakohtaisista havainnointiaikatauluista. Aseman aikataulu sisältää VLBI-kokeen kohdeluettelon, jonka jokaiselle kohteelle eli skannille on annettu havainnoinnin aloitusajankohta päivämääränä ja UTC-aikana sekä skannin havainnointiajan pituus. Kaikki ajat on annettu sekunnin tarkkuudella.

Tsunami-tiedostonsiirtoprotokollassa määritellään ainoastaan siirrettävän tiedoston nimi ja sen pituus tavuina, suoraa protokollatason tukea skannien aloitusajankohdille ja pituuksille ei ole. Puuttuva skedulitieto voidaan välittää reaaliaikaiselle Tsunami-palvelimelle useilla tavoilla.

Näistä suoraviivaisin on skannin tietojen lisääminen palvelimelta pyydetyn tiedoston nimeen. Tällöin itse protokollaa eikä mitakaan ohjelmia tarvitse muuttaa, lukuun ottamatta reaaliaikaista palvelinohjelmaa. Palvelinohjelmassa tiedot voidaan purkaa helposti tiedostonimestä tiedostonimitulkilla, jonka ensimmäisen version oli kirjoittanut Timo Lindfors.



Laajemman yhteensopivuuden VLBI:ssä takaa se, että EVN:ssä on jo olemassa standardoitu nimeämiskäytäntö skannitiedostoille. Standardin mukaan tiedostonimi saa sisältää skannin aloitusajankohdan, ja se sallii myös käyttäjäkohtaiset lisätietokentät osana tiedostonimeä (HKR04). Esimerkki tyypillisestä EVN-tiedostonimestä on *euro84\_mh\_184-0150b\_2006-10-17T12:25:30\_dl=1536000000.vsi*. Tiedosto vastaa EURO84 VLBI-kokeessa asemalla Mh (Metsähovi) tallennettua skannia tähtitaivaan kohteesta 184-0150b, jonka havainnointi alkoi annettuna ajankohtana. Lisätietokentässä *dl=1536000000* reaaliaika-palvelimelle välittyy tieto skannin pituudesta tavuina – tämä on sama kuin tiedoston pituus Tsunami-siirron lopussa. EVN-tiedostonimeen voidaan siis liittää kaikki reaaliaikaisen Tsunamin tarvitsemat tiedot.

Hieman ennen haluttua tiedonkeruun aloitusajankohtaa reaaliaikainen Tsunami-palvelin käynnistää VSIB-kortin tiedonkeruutilaan. Kortti aloittaa jatkuvan tiedonkeruun VSI-väylän seuraavalla 1PPS pulssilla, jonka jälkeen palvelinohjelma saa kortin ajurilta näytteitä verkkoon lähetettäväksi. Näin tiedonsiirto tapahtuu reaaliajassa. Reaaliaikaisten näytteiden siirron lisäksi palvelin voi asiakkaan halutessa kirjoittaa lähtevän liikenteen paikallisen tiedostoon, eli voi luoda varmuuskopion kerätyistä näytteistä. Varmuuskopios- ta voi olla hyötyä muun muassa jos asiakaskoneen RAID-levy hajoaa. Kopiota voidaan myös käyttää, mikäli asiakas on tahallisesti valinnut häviöllisen tiedonsiirtomoodin eikä verkon kapasiteetti ajoittain riittänyt reaaliaikasiirrolle ja siirrossa on kadonnut enemmän kuin hyväksyttävä osa näytteistä. Paikalliskopio voidaan siirtää myöhemmin korrelaatio- rille Tsunamilla off-line e-VLBI -tiedostosiirtona.

Kun reaaliaikainen palvelin saa asiakkaalta tiedostonlähetyspyynnön ja pyyntöön sisälty- vän tiedostonimen, EVN-tiedostonimitulkki yrittää ensin purkaa nimestä skannin UTC- aloitusajankohdan ja pituuden. Jos nimi ei vastaa EVN-standardia, skanni valitaan alka- maan heti ja sille asetetaan vakiopituus.

Seuraavaksi palvelin odottaa kunnes tietokoneen kello on puoli sekuntia ennen skannille määrättyä UTC-aloitusajankohtaa. Tässä kohtaa palvelin asettaa VSIB-kortin tiedonke- ruutilaan, jossa kortti aloittaa tiedonkeruun heti seuraavalla VSI-väylän 1PPS signaalil- la. 1PPS signaali on johdettu observatorion tarkasta kellosta, joka on tavallisesti GPS- lukittu vetymaseri. Tämä takaa huomattavasti alle millisekunnin tarkkuuden tiedonke- ruun aloitusajankohdalle jokaisella sekuntirajalla, mutta palvelimen on vielä odotettava oikeaa aloitussekuntia. Tämän onnistuu täysin riittävällä tarkkuudella, jos tietokone pide-

tään verkon yli NTP-synkronoituna NTP-palvelimeen, kuten Mittaustekniikan keskuksen ja Funet:in tarjoamaan NTP-aikastandardiin. Kun VSIB-tiedonkeruun on lopuksi käynnistynyt tarkasti määrättyinä aloitusajankohtana, palvelinohjelma aloittaa VSIB-ajurin rengaspuskuriin ilmestyvien näytteiden reaaliaikaisen siirron Tsunami-asiakkaalle.

Reaaliaikainen Tsunami toimi jo vuonna 2005 konseptitoteutuksena suhteellisen hyvin. Kesällä 2006 Metsähovissa paikallisesti ja verkon yli muiden asemien kanssa tekemiä reaaliaikaisten siirtokokeiden aikana ilmeni kuitenkin useita virheitä ja puutteita, jotka oli korjattava ennen EXPreS-projektin marraskuista Month7 Demo -esittelytilaisuutta. Tuotantokäyttöä varten oli myös joidenkin muiden asemien antamia palautteita ja kehitystoiveita, joista suurimman osan toteutin vuodenvaihteessa.

### 3.4.1 Yksittäisen VLBI-siirron asetusten välittäminen reaaliaikapalvelimelle

Alkuperäinen EVN-tiedostonimitulkki ei selvinnyt joidenkin asemien käyttämistä valideista EVN-tiedostonimistä, mahdollisesti koska tulkin yksikkötestauksen testitapauksena oli vain standardissa annettu yksi esimerkki. Suurin ongelma tulkille olivatkin EVN-standardin muut aika- ja päivämääräformaattit. Tulkissa oli lisäksi joissakin kohdissa käytetty *assert()* kutsuja virheenkäsitteilyn sijaan. Testi- ja loppukäännöksissä nämä tietyissä tilanteissa lopettivat tai kaatoivat Tsunami-palvelinohjelman. Korvasin kaikki *assert()* kutsut ja loin lisää testitapauksia, jotka kattavat mahdolliset kymmenen eri aikaformaattia ja nimeämistapaa (HKR04).

Korjattu tiedostonimitulkki tukee kaikkia aikaformaatteja, desimaalimuotoista esitystapaa lukuun ottamatta. Uusi versio pystyy lisäksi käyttämään osittain puutteellisten päivämäärätietojen kohdalla oletusarvona nykyisen päivämäärän vastaavia tietoja. Käyttäjärjestelmässä voi nyt käyttää muuta kuin UTC aikavyöhykettä, ilman vaikutusta tiedostonimen UTC-ajan tulkintaan.

Alkuperäisessä reaaliaikapalvelimessa oli siirrettävän tiedon pituudeksi asetettu aina 56 GB. Tämä vastaa oletetulla 512 Mbit/s tiedonkeruunopeudella 15 minuutin havainnointiaikaa. Kuitenkin EVN:n VLBI-kokeissa tiedonkeruunopeudet vaihtelevat. Käytetään myös yhä skanneja ja näiden välisiä taukoja, joiden pituudet vaihtelevat kokeen tyypistä riippuen parin kymmenen sekunnin ja kymmenien minuuttien välillä. Taukojen aikana antenni voi siirtyä kohteesta toiseen. Vaikka sekä Mark5 StreamStor-kortti että PC-EVN soveltuisivat hyvin jatkuvaan keskeyttämättömään tiedonkeruuseen, taukojen ajalta ei tal-

lenneta turhia näytteitä. Näin tehdään pääasiassa säästösyistä, koska kovalevykapasiteettia vapautuu noin 10-20% hyötykäyttöön. Toisin kuin Mark5-järjestelmässä, PC-EVN ja ohjelmistokorrelaattorien tapauksessa lisäetuna on myös, että havainnoituja tähtitaivaan eri kohteita pystytään erillisiin tiedostoihin jaettuna käsittelemään paljon kätevämmiin.

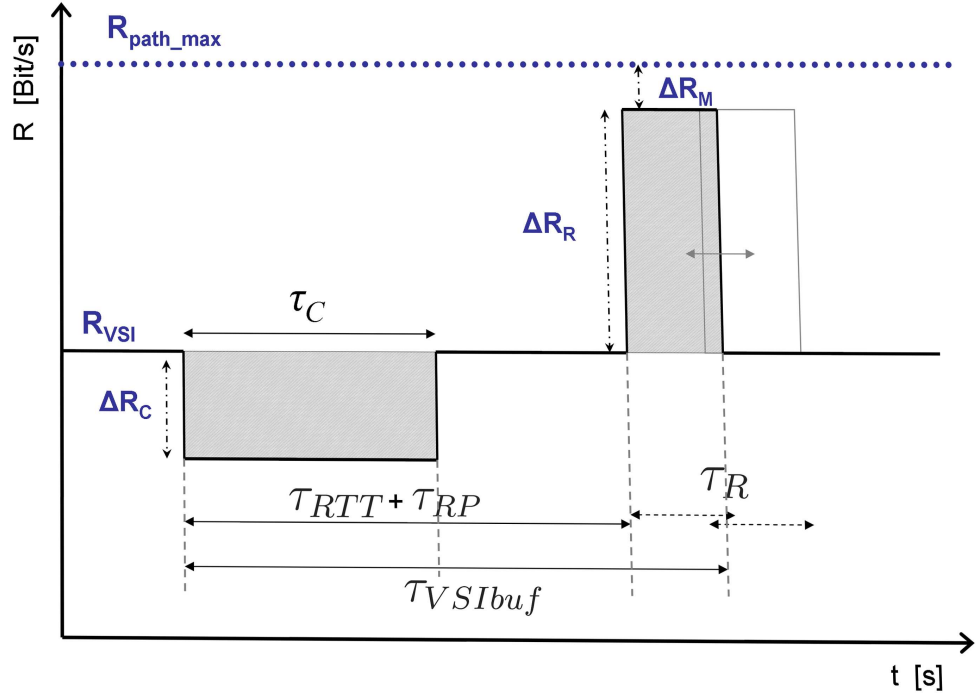
Suoraviivaisin tapa välittää palvelimelle tieto skedulien perusteella kerättävän vaihtelevan pituisen tiedon määrästä on EVN-tiedostonimistandardin vapaavalinnaisissa lisätietokentissä. Lievä hankaluus on, että VSIB-kortti kerää näytteitä VSI-väylän kellon mukaan eikä VSIB-ajurille tai reaaliaikaiselle Tsunami-palvelinohjelmalle välity tietoa tästä kellosta eli ulkoisen järjestelmän käyttämästä näytteenottonopeudesta. Skedulissa sekuntiyksiköissä annetut skannien pituudet kerrotaan siis samasta skedulista a priori -tietona saatavalla skannin ajankohtana käytetyllä näytteenottotaajuudella, ja saatu tavumäärä välitetään asiakkaalta palvelimelle osana tiedostonimeä.

### 3.4.2 Aikarajoitetut uudelleenlähetykset

Kaikentyypisessä reaaliaikaisessa tiedonsiirrossa voi syntyä ongelmatilanteita, varsinkin jos vakio reaaliaikainen siirtonopeus yhdessä verkon muun tietoliikenteen kanssa on lähellä runkoverkon polun maksimikapasiteettia. Tämä johtaa hetkittäin polun ylikuormittumiseen, jolloin reaaliaikaiselle UDP-siirrolle saatava kaista voi olla liian pieni ja verkko tiputtaa kaistanleveyden ylittävää liikennettä. Tiputtamiseen vaikuttaa eri liikenneluokkien ja protokollien priorisointi verkossa. Usein kaikki liikenneluokat kärsivät yhtäläisesti.

Jos reaaliaikasiirrolta puuttuva kaistanleveys ruuhkautumisen keston yli integroituna ylittää lähetyspuolen puskurointikapasiteetin, ruuhkatilanteen alussa kadonneita tietoja ei voida enää pyytää uudelleen siirrettäviksi ja tiedot kadotetaan peruuttamattomasti. Vanhin tieto katoaa ensimmäisenä, koska lähetyspuolella käytetään rengaspuskuria. Jos taas ruuhkautuminen on tarpeeksi lievä tai lyhytkestoinen eikä lähetyspuolen puskuritila ylitä, palvelin voi yrittää nostaa lähetysnopeutta ja siirtää vakionopeuksisen reaaliaikaisen tiedon lisäksi aiemmin kadonneet tiedot uudestaan. Tämä siirto vaatii polulta enemmän vapaata kaistaa kuin ennen ruuhkautumista. Siksi ruuhkien aiheuttama tiedonhäviö on korjattavissa luotettavasti vain, jos siirtoa häiritsevä muu liikenne on Pareto- tai Poisson-jakautunutta ja siten myös ruuhkautumiset purskeisia, jolloin ruuhkautumisten ulkopuolella polun vapaa kaistamarginaali on yhä riittävän suuri.

Verkon ruuhkautumisen lisäksi myös palvelin- tai asiakaskoneilla voi tapahtua ylikuor-



Kuva 3.3. Uudelleenlähetyksen aikarajoitukset. Palvelimen näytehistoria  $\tau_{\text{VSIbuf}}$ , polun kiertoviive  $\tau_{\text{RTT}}$  ja uudelleenlähetykspyyntöjen keräämisaika  $\tau_{\text{RP}}$  määräävät pisimmän ajan  $\tau_R$ , jonka kuluessa puuttuvan paketin uudelleentoimitus on onnistuttava, jotta siirto olisi häviötön. Ruuhkatilanteessa nopeus putoaa  $\Delta R_C$  ja uudelleenlähetyksien vaatima nopeutumisen on  $\Delta R_R$ .

mittumista. Mikäli palvelintietokoneella on konetta huomattavasti kuormittavia prosesseja, käyttöjärjestelmän UDP-lähetyspuskuri voi harvinaisissa tilanteissa vuotaa yli ja aiheuttaa UDP-pakettien häviötä. Asiakaspuolella sama pätee käyttöjärjestelmän UDP-vastaanottopuskuriin. Jos asiakasohjelma lukee vastaanottopuskuria liian hitaasti, esimerkiksi kun tiedostoon kirjoittaminen jostain syystä hidastuu, puskuuri voi vuotaa yli ja UDP-paketteja katoaa.

Tsunamin tiedostosiirroissa kaikki kadonneet UDP-paketit eli tiedoston lohkot voidaan aina pyytää lähetettäväksi uudestaan, koska lähetettävä tiedosto ei katoa eikä vanhene. Reaaliaikaisessa Tsunami-siirrossa siirron häviöttömyyttä ruuhkatilanteissa eniten rajoittavat tekijät ovat polun kiertoviive  $\tau_{\text{RTT}}$  ja asiakkaan uudelleenlähetykspyyntöjen keräämisaika  $\tau_{\text{RP}}$ . Jos näiden summa ylittää kaavassa 2.1 esitetyn palvelimen VSIB-puskurin

näytehistorian ajallisen pituuden  $\tau_{VSIB_{buf}}$ , palvelin käsittelisi puuttuvan paketin tietoja koskevaa uudelleenlähetyspyyntöä vasta tietojen jo kadottua näytehistoriasta. Näin tapahtuu erityisesti, jos myös uudelleenlähetettyjä paketteja katoaa ja niitä pyydetään uudestaan.

Reaaliaikasiirtoon vaikuttavat aikarajoitukset on havainnollistettu kuvassa 3.3, jossa oletetaan yksinkertaistettu hetkellinen ruuhkautuminen ja vakionopeuden putoaminen hetkeksi matalammalle vakionopeudelle. Joko palvelimen tai asiakkaan on huomattava aikarajoitusten mahdollinen ylittyminen eli validien tietojen katoaminen ja on käsiteltävä tämä tilanne jollain sopivalla tavalla. VLBI:ssä ehdotetaan usein puuttuvan tiedon korvaamista esimerkiksi täysin satunnaisella ja korreloimattomalla kohinalla, pelkästään nollatavuilla tai myös jollakin vakiolla tavukuviolla, jonka korrelaattori voi tunnistaa. Tsunamissa puuttuvat tiedosto-osat sisältävät nollatavuja, mutta *blockdump*-asetuksen luoman bittikarttatiedoston kautta ulkoiset ohjelmat voivat helposti soveltaa puuttuviksi merkittyihin osiin vaihtoehtoisia näytteiden korvaamistapoja.

Ruuhkatilanteen jälkeen tarpeellinen lähetysnopeuden hetkellinen nosto, kuten kuvassa 3.3, toimii Tsunamissa jo itsestään, sillä  $R_{target}$  tavoitenopeuden oletusarvo on noin 100 Mbit/s korkeampi kuin reaaliaikainen VSIB-lukunopeus 512 Mbit/s. Siksi VSIB-kortilta luetut vanhat eli uudelleenlähetettävät tiedot yritetään siirtää nopeudella  $R_{target}$ ,

Koska alun perin Tsunami oli tarkoitettu tiedostojen häviöttömään siirtoon, Tsunamissa ei ollut reaaliaikaisen siirron vaatimia aikarajoja uudelleenlähetyksille. Ideaalitapauksessa tietysti verkko, palvelin- ja asiakaskone eivät koskaan kokisi ruuhkaa tai ylikuormituksia. Käytännössä näin ei usein ole, kuten myöhemmissä aitojen e-VLBI -siirtojen testituloksia esittelevissä kappaleissa todetaan. Yllättäen jopa verkkojen valopolkupalveluja ja taattua kapasiteettia käyttävissä siirroissa esiintyy samoja ongelmia.

Alustavasti hyvä kohta toteuttaa aikarajat on Tsunamin palvelinohjelmassa – ohjelma tuntee VSIB-rengaspuskurin koon ja tätä kautta puskurista vielä löytyvän vanhimman lohkon järjestysnumeron. Jos asiakkaan uudelleenlähetyspyynnöt sisältävät tätä pienempiä lohkonumeroita, vastaavien aitoja tietoja sisältävien lohkojen sijaan voidaan asiakkaalle palauttaa vakiosisältöinen lohko. Toteutuksen etu on, että se on täysin läpinäkyvä asiakasohjelmalle ja vaatii siis muutoksia vain reaaliaikaiseen palvelinohjelmaan eikä itse protokollaan. Jos protokollaa muutettaisiin, voidaan paikkauslohkojen vaatima lähetyskaista minimoida lisäämällä protokollaan yksinkertaisimmillaan mukaan uusi UDP-

paikkauslohkotyyppi, nykyisten kolmen lohkotyyppin eli alkuperäisen, uudelleenlähetetyn tai lopetuslohkon ohelle. Paikkauslohko sisältäisi ainoastaan otsaketiedot eikä näytetietoja.

Lopuksi lisäksi kuitenkin aikarajoitetun siirtomoodin palvelimen sijaan asiakasohjelmaan. Näin voidaan muun muassa välttää verkon kuormittaminen turhilla uudelleenlähetetyillä eikä protokollaa tarvitse muuttaa näiden vaatiman kaistan minimoimiseksi. Toisekseen Tsunami-protokollan periaate säilyy ja käyttäjä sekä asiakasohjelma yhä hallitsevat koko tiedonsiirron kulkua. Lisäksi palvelimeen yhdistävä käyttäjä voi nyt asettaa omat aikarajansa tai täysin sammuttaa uudelleenlähetetykset ja siten myös ruuhkanhallinnan. Koska Tsunami-protokollasta oli lisäksi asennettu kehitystyön kuluessa jo useita aiempia versioita eri observatorioiden koneille, toteutuksen yhteensopivuus kaikkien aiempien palvelin- ja asiakasohjelmaversioiden kanssa oli toivottavaa.

Asiakas voidaan nyt yhden uuden komentoriviasetuksen avulla siirtää tavallisesta häviötömästä siirtomoodista aikarajoitettuun moodiin. Aikarajan  $\tau_{window}$  käyttäjä voi muuttaa 15 sekunnin oletusarvosta millisekunnin tarkkuudella joksikin järkevämmäksi arvoksi. Sopiva arvo jää pitkälti käyttäjän estimoitavaksi. Erittäin hyvä lähtökohta  $\tau_{window}$  arvolle on VSIB-puskurin  $\tau_{VSIBuf}$  käyttäjän tuntemalla siirtonopeudella, vähennettynä jokin kieroviiveen  $\tau_{RTT}$  moninkerta. Kiertoviive  $\tau_{RTT}$  voidaan joko mitata Linuxin *ping*-verkkotyökalulla tai sen tilalle voidaan yksinkertaisesti sijoittaa erittäin pessimistinen arvo 750 ms.

Kun Tsunami-asiakas vastaanottaa lohkon  $B_{newest}$  eivätkä sitä edeltäneet lohkot saapuneet perille, asiakas tarkastaa jokaisen puuttuvan lohkon  $B_j \in \{B_{gapless+1} \dots B_{newest-1}\}$  kohdalla, täyttyykö sille käyttäjän määrittelemä aikaraja, eli voidaanko lohko vielä pyytää uudelleenlähetettäväksi. Tähän käytetään epäyhtälön 3.5 heuristista kriteeriä. Lohko lisätään uudelleenlähetyslistaan, jos seuraava epäyhtälö on tosi:

$$(B_{newest} - B_j) \cdot L_{block} \cdot 8 \frac{bit}{byte} \leq 0.8 \cdot R_{actual}(t) \tau_{window} \quad (3.5)$$

Tässä oletetaan yhteyden kapasiteetin olevan jatkossa vain 80% nykyisestä  $R_{actual}(t)$  kokonaissiirtonopeudesta, ja muunnetaan tämä aikaikkunaan  $\tau_{window}$  mahtuvaksi bittimääräksi. Uusimman vastaanotetun ja puuttuvan  $B_j$  -lohkon välinen bittimäärä on oltava tätä pienempi. Ohjelmassa sama tarkistus on toteutettu siten, että lasketaan  $B_{gapless}$  ja

$B_{newest}$  välille sijoittuva vanhin lohko  $B_v$  joka voidaan vielä lähettää uudestaan. Kaikki lohkot tästä lohkosta  $B_v$  nykyiseen  $B_{newest}$  asti lisätään uudelleenlähetettäväksi. Lopuksi hypätään aikakriteerin mukaan kadotettujen lohkojen yli ja siirretään  $B_{gapless}$  arvoa eteenpäin  $B_v$  -lohkoon.

Tämä menetelmä yhdessä Tsunami-ohjelmiin lisättävien uusien asetusten kanssa vaati vain noin 15 ohjelmariviä, mutta toimii yllättävän hyvin. Paikallisverkossa kahden koneen välillä tehtiin kuormituskoe, jossa siirrettiin reaaliaikaista testidataa 512 Mbit/s nopeudella ja lisättiin rinnalle häiritsevää UDP- ja TCP-liikennettä ajamalla *iperf*-testiohjelmaa samojen koneiden välillä. Häviöttömässä siirtomoodissa nopeus hidastui alle reaaliaikanopeuden ja yli 10% oikeista testinäytetiedoista menetettiin rengaspuskurin ylivuodoissa. Pitkällä  $\tau_{window} = 5000ms$  aikaikkunalla lähetysnopeus vaihteli yhä suuren uudelleenlähetyksen määrän takia palvelimen paljon reaaliaikanopeuden ympärillä, mutta tietoja katosi selvästi alle 1%. Pienentämällä aikaikkunaa häviö kasvoi uudelleenlähetysspyyntöjen vähentyessä, mutta palvelimen lähetysnopeus pysyi paljon vakaampana.

Jos siis huonolla yhteydellä käytetään reaaliaikaisessa siirrossa aikarajoitusta eikä tiedostosiirtoihin tarkoitettua häviötöntä siirtoa, siirto voi pysyä yhä reaaliaikanopeudessa ja lopullinen hävinneiden tietojen määrä voi olla pienempi.

Tähän mennessä käytännön yhteydet eivät kuitenkaan ole olleet siinä määrin ruuhkaisia, että aikarajoitetun ja häviöttömän siirtomoodin välillä olisi ollut mitään käytännön eroja.

Siirtonopeuksien kasvaessa tämä uusi toiminto tulee ajankohtaisemmaksi, kuten myös suuremman etäisyyden kansainvälisissä siirroissa, esimerkiksi Kiinan kautta. Näissä runkoverkkojen kapasiteetti on varsin rajallinen, yhteydet kuljettavat paljon liikennettä eikä usean eri kansallisen verkon läpi kulkevalle VLBI-liikenteelle ole saatavissa valopolkupalveluita päästä päähän asemien välille. Aikarajoitetut siirrot voivat siis tulevaisuudessa osoittautua varsin hyödyllisiksi.

### 3.4.3 896 Mbit/s e-VLBI moodi

Tiedonkeruu Mark5-järjestelmillä tapahtuu vain Mbit/s nopeuksilla, jotka ovat kahden potensseja, esimerkiksi 512 tai 1024 Mbit/s. Tietoja voidaan teoriassa kerätä kaikkien 16 BBC-yksiköiden maksimissaan 16 MHz levyisiltä USB ja LSB kaistoilta. Tavallisesti VLBI:ssä käytetään vain 8 BBC-yksikköä, jolloin 2-bittisten näytteiden virta on nopeim-

millaan 1024 Mbit/s 16 MHz kaistanleveyksillä tai 512 Mbit/s 8 MHz kaistanleveyksillä. Metsähovin 14 BBC-yksikön kanssa vastaavat maksiminopeudet ovat 1792 Mbit/s ja 896 Mbit/s.

Ennen kuin 10 Gbit/s yhteydet yleistyivät vuonna 2007, EVN:n asemilla oli korkeintaan yksi 1 Gbit/s kuituyhteys. Mark5 voi siirtää 512 Mbit/s tietoja suhteellisen helposti tällaisen yhteyden oli. Noin puolet kuidun kapasiteetista jää käyttämättä. Oli kiinnostava nähdä, voidaanko kuidun kapasiteetti hyödyntää vielä paremmin PC-EVN -järjestelmän avulla.

Koska tietoliikenteen yhteyskäytäntö vaatii aina oman osan kuidun kapasiteetista, täyttää 1 Gbit/s nopeutta ei voida saavuttaa 1 Gbit/s kuidulla. Aiemmin mainituista tiedonkeruunopeuksista 896 Mbit/s on lähimpänä 1 Gbit/s kuidun todellista kapasiteettia. Aidoilla näytteillä tämä nopeus voidaan saavuttaa kahdella eri tavalla. 1024 Mbit/s näytevirrasta voidaan poistaa yksittäisiä kanavia ja jatkaa muiden kanavien lähettämistä tasaisella nopeudella. Voidaan myös lähettää kaikki kanavat yhteensä 1024 Mbit/s nopeudella ja aikajakoisesti tiputtaa aina pätkä näytteistä. Tässä luotetaan siihen, että ohjelmien tietoliikenne puskuroidaan lyhyellä aikavälillä käyttöjärjestelmän protokollapinossa ja puskuroinnin jälkeen verkkokortilta lähtevä liikenne olisi ajallisesti keskiarvoistunut hitaamalle nopeudelle.

Lähetysnopeuden ajalliseen keskiarvoistumiseen luottamisessa on omat ongelmansa. Yksinkertaisin tapa saada 896 Mbit/s kokonaisnopeus oli kanavien poistaminen. Tämä toteutettiin reaaliaikaisessa Tsunami-palvelinohjelmassa. Tässä 896 Mbit/s saadaan käyttämällä 8 MHz kaistanleveyttä ja yhdistämällä näytteet kahdeksan BBC:n (512 Mbit/s, 32 näytebittiä) ja toisen kuuden BBC:n (384 Mbit/s, 24 näytebittiä) antamilta kanavilta. Kaikkia neljäätoista BBC:tä varten tarvitaan kaksi VSI-väylää. Kuhunkin VSI-väylään on liitetty yksi PC-EVN tietokone. Koneet ovat kiinni samassa 1 Gbit/s kytkimessä, joka on liitetty esimerkiksi 1 Gbit/s valokuidun yli läheisimpään Internet-palveluntarjoajaan.

Kahden VSI-väylän näytteet vaativat ennen verkkoon lähettämistä eri käsittelyn. Ensimmäisen kahdeksan BBC:n näytetiedot kulkevat suoraan ensimmäisellä 32-bittisellä VSI-väylällä, jolta ne voidaan lähettää suoraan verkkoon tavallisella reaaliaikaisella Tsunami-palvelinohjelmalla.

Toiselta VSI-väylältä tulevien näytteiden käsittely on hieman vaikeampaa. VSIB-kortti



lukee VSI-väylää aina täydeltä 32 bitin leveydeltä, joten käytettäessä vain viimeistä kuutta BBC:tä väylän 8 ylintä bittiä eivät sisällä aitoja näytteitä ja ne on poistettava ohjelmallisesti. Tätä varten lisäsin palvelinohjelmaan turhia VSI-tavuja poistavan 3/4-moodin, joka säilyttää yhdestä 32-bittisestä sanasta eli neljästä tavusta vain 3 ensimmäistä käyttökelpoista tavua. Karsitun 512 Mbit/s näytevirran nopeus tippuu 384 Mbit/s nopeuteen. Muokattu palvelinohjelma lukee yhä VSIB-korttia rengaspuskurin kautta tavallisen tiedoston tapaan, mutta kun uusi UDP-pakettia kootaan jostain kohtaa tätä tiedostoa, on tiedostokohtaan osoittava tavuosoitin ensin korjattava kertoimella  $1 + 1/3$ . Korjaus antaa aidon osoitteen rengaspuskurissa. Korjatun tavuosoittimen määräämästä kohdasta luetaan kaksinkertainen määrä VSIB-tavuja väliaikaiseen puskuriiin. Hyötytavuja kopioidaan puskurista lopulliseen UDP-pakettiin ottamalla aina korjatun tavuosoittimen 32-bittisesti alignoiduista kohdista kolme tavua ja jättämällä neljäs tavu kopioimatta. Silmukassa suoritettu kopiointi lisää palvelinohjelman prosessorikuormitusta jonkin verran, mutta ensimmäisten toimivien 384 Mbit/s ohjelmatestien perusteella suurempi kuorma ei vaikuta haitallisesti siirtonopeuteen.

896 Mbit/s kokonaisnopeus 1 Gbit/s pullonkaulan läpi saadaan siis ajamalla observatorion puolella yhdessä PC-EVN-koneessa tavallista reaaliaikapalvelinta ja toisessa koneessa uutta 3/4-moodin reaaliaikapalvelinta. Vastaanottopuolella ajetaan kahta tavallista Tsunami-asiakasta samassa koneessa tai eri koneissa. Jodrell Bankin kanssa testasimme onnistuneesti 896 Mbit/s siirtomoodin toimivuutta käytännössä. Koeasettelu ja tulokset on esitetty myöhemmässä kappaleessa 4.3.

### 3.5 Reaaliaikaisen e-VLBI -havainnoinnin automatisointi

Toteutettu menetelmä, jolla yksittäisen reaaliaikaisen skannin siirtoa pyydetään palvelimelta on reaaliaikaisen e-VLBI -havainnoinnin kannalta välttämätön, mutta sillä ei saada pyydettyä kokonaisen VLBI-kokeen kaikkeja skanneja helposti. Jotta VLBI-kokeen havainnointi olisi mahdollisimman helppoa, kirjoitin asemakohtaisen aikataulutiedoston havainnointiskriptiksi muuntavat apuskriptit. Näitä generointiskriptejä on kaksi, ensimmäinen luo levytallennukseen tarkoitettuja skriptejä ja toinen Tsunami-asiakasskriptejä reaaliaikaiselle e-VLBI -havainnoinnille. Ajan tasalla olevat skriptit löytyvät muun muassa Linux-järjestelmän ja VSIB-työkalujen asennusta PC-EVN -tietokoneelle tarkemmin

kuvaavasta verkkojulkaisusta (Wag06-2).

Laatimani automaattisten havainnointiskriptien generointiskriptit lukevat skannien nimet, kohteet, aloitusajat ja kestot VLBI-koetta vastaavasta FieldSystem-järjestelmän .snap-tekstitiedosta. FieldSystem on vanha Haystackissä kehitetty ohjelmistopaketti, joka sisältää VLBI-hallintaohjelman ja apuohjelmia. Hallintaohjelma pitää huolen havainnointiaikataulun suorittamisesta VLBI-kokeen aikana, ja välittää aseman muulle laitteistoille kuten antenniohjauksella ja tiedonkeruujärjestelmille oikeat asetukset. Hallintaohjelma käyttää VLBI-kokeen yleisaikataulusta apuohjelmalla luotuja asemakohtaisia .snap ja muita tiedostoja. Luomani generointiskriptit voivat siis helposti käyttää näitä olemassaolevia FieldSystem-tiedostoja. Snap-tiedostoissa aloitusajaksi on annettu muodossa vuosi, päivä ja kellonaika. Generointiskriptit tallentavat nämä tiedot samassa muodossa generoituun skriptiin ja muuntavat lisäksi ajat ja kohteet EVN-tiedostonimiksi.

Generointiskriptin luomat levytallennusskriptit ovat rakenteeltaan yksinkertaisia. Skriptit kutsuvat vuorotellen Mujusen aiemmin kirjoittamaa *dstart* ohjelmaa, joka palaa vastasille kutsuparametreissa välitettynä skannin UTC-aloitusajanhetkenä, sekä *wr* kovalevytallennusohjelmaa, joka lukee annetun tavumäärän tietoja VSIB-kortilta ja kirjoittaa ne yhteen tai useaan tiedostoon.

e-VLBI -kokeille generoitu Tsunami-asiakasskripti kutsui ensimmäisessä toteutusversiossa yksinkertaisesti Tsunami-asiakasohjelmaa peräjälkeen – aluksi ensimmäiselle skannille, skannin siirron päädyttyä seuraavalle skannille, ja niin edelleen. Siirtojen skriptityksen mahdollistaa aiemmin mainittu Tsunami-asiakasohjelmaan tekemäni lisäys, jolla yksittäisiä tai useita peräkkäisiä Tsunami-komentoja voidaan välittää myös ohjelman kutsuparametreissa. Jokaisessa asiakasskriptin tekemässä Tsunami-asiakasohjelman kutsussa asiakkaalle annetaan komennot, joilla asiakas yhdistää annettuun palvelimeen, asettaa maksimisiirtonopeuden, pyytää skannitiedostoa, ja sulkee yhteyden kun tiedosto on vastaanotettu.

e-VLBI -skriptin yksinkertaisessa toteutuksessa oli joitakin ongelmia. Esimerkiksi käyttäjän keskeyttämää asiakasskriptiä ei voinut suoraan käynnistää uudestaan, muokkaamalla sitä ensin manuaalisesti. Muutoin skripti olisi aloittanut siirtopyynnöt uudestaan ensimmäisestä jo vanhentuneesta skannista eikä seuraavasta ajankohtaisesta skannista. Useista syistä asiakasohjelma saattaa myös pysähtyä odottamaan tietoja, joita palvelimelta ei kuitenkaan saada, jolloin koko skripti pysähtyy kyseisen skannin kohdalle. Näin voi tapahtua

pääasiassa verkkokatkon tai tietystä kohdasta manuaalisesti keskeytetyn palvelinohjelman takia.

Nämä ongelmakohdat on pyritty ratkaisemaan generoitujen asiakasskriptien uudessa toteutuksessa. Nyt asiakasskripti hyppää kaikkien menneisyydessä alkavien skannien yli ja odottaa oletusarvoisesti 15 sekuntia ennen seuraavan skannin aloitusajankohtaa. Tässä kohtaa skripti tarkistaa, onko aikaisemman skannin pyytänyt Tsunami-asiakasohjelmainsi vielä käynnissä, ja tarvittaessa lopettaa sen. Seuraavaksi skripti käynnistää taustalle uuden Tsunami-siirron, ja jää itse odottamaan skannin määrätyn keston ajaksi. Lopuksi skripti palaa taas odottamaan 15 sekuntia ennen seuraavan skannin aloitusajankohtaa.

Myöhemmissä testeissä ja VLBI-kokeissa skriptien uusi versio ja tämän uudet toiminnallisuudet toimivat luotettavasti. Ainoa muutosten kautta tullut ohjelmallinen lisävaatimus oli, että nyt PC-EVN-palvelinkoneen lisäksi myös asiakaskone tulee olla NTP-synkronoitu.

## Luku 4

### Kehitystyön tuloksia

Seuraavassa esitetään Metsähovissa tekemäni e-VLBI -kehitystyön tuloksia. Tuloksiin sisältyvät kehitettyjen tai päivitettyjen ohjelmien toiminnallisuus- ja suorituskykytestit, sekä tulokset julkisista kansainvälisistä demonstraatioista. Tässä rajoitutaan vain tuotantokäytön kannalta kaikkein tärkeimpiin testeihin ja näistä saatuihin tuloksiin.

Tsunami-ohjelmiin lisäämäni korjaukset ja useat uudet toiminnot testasin kehitystyön aikana useissa siirtokokeissa Onsalan, Jodrell Bankin, Kashiman ja muiden asemien ystävällisellä avulla. Lisäykset ovat toimivia, mutta rajoitetun tilan ja luettavuuden vuoksi näihin testituloksiin ei tässä paneuduta tarkemmin. Sama koskee useassa asemassa nyt jo käytössä olevien e-VLBI -automatisaatiokriptien testausta.

Aluksi kehitystyön tuloksista esitetään päivitetyn VSIB-ajurin kuormitus- ja tavalisten toimistokoneiden nopeustestien tulokset. Tämän jälkeen kuvataan ensimmäinen kolmen Eurooppalaisen aseman aidon PC-EVN e-VLBI -demonstraation koearkitys ja demonstraation tulokset. Lopuksi esitellään 896 Mbit/s siirtokokeen Metsähovi-maailmanennätys, ja kuvataan lyhyesti, kuinka tässä työssä kehitetty järjestelmä on nyt siirtymässä tuotantokäyttöä geodeettisessa VLBI:ssä. Kaikki viitatuksi kuvaajat löytyvät liitteestä A.

#### 4.1 Metsähovin sisäinen e-VLBI -testaus ja 1 Gbit/s kuitutesti

Kehitystöiden alussa, ennen kuin Metsähovi sai 10 Gbit/s kuituyhteyden, testasimme Tsunami-ohjelmien suorituskykyä paikallisesti kolmella tavallisella pöytäkoneella ja myös

etänä JIVE:ssä ja Jodrell Bankissa sijaitsevien, ystävällisesti meidänkin käyttöön annettujen testikoneitten välillä. Tarkoituksena oli selvittää, kuinka luotettavasti Tsunami-siirrot toimivat emolevyjen yhdysrakenteisten 1 Gbit/s verkkokorttien yli. Testitulokset myös näyttäisivät, kuinka käyttökelpoisia tavalliset toimistotietokoneet olisivat eVLBI-siirroissa.

Paikallisissa siirtokokeissa koneet kytkettiin halpaan SMC EZ Switch 8508T 1 Gbit/s kytkimeen, jota ei liitetty Metsähovin lähiverkkoon. Etätesteissä liikenne Jodrell Bankin UKLight-verkkotarjoajan ja JIVE:n Surfnet-verkkotarjoajan välillä siirtyi Jodrell Bankin hankkiman valopolun yli (lightpath). Valopolku on tutkimusverkkojen usein tarjoama maksullinen lisäpalvelu, jonka tilaajalle taataan reititys johonkin päätepisteeseen vain tilaajan käyttöön tulevia kuituja pitkin. Tilaaajan liikenne ei tällä reitillä joudu jakamaan kapasiteettia muun verkon liikenteen kanssa. Siirtotestien aikana polulla ei ollut muuta liikennettä. Jodrell Bankin kone voitiin kaapelointia muuttamalla siirtää heidän 630 Mbit/s kuidusta toiseen 1 Gbit/s kuituun.

Kirjoitin Tsunamiin pienen lisäyksen, jolla palvelimen levyltä luku tai asiakkaan levyille kirjoitus voidaan sammuttaa. Pelkkänä muistista muistiin siirtona eli ilman levy-I/O:ta tuloksista voidaan nähdä verkkokortin suorituskyky. Tässä verkon yli ei siirry hyötydataa, vaan ainoastaan alustamattoman muistipuskurin sisältö. Toinen lisäys oli, että asiakkaassa voidaan sen omalta komentoriviltä määritellä siirrettävien UDP-pakettien koko.

Koesiirrot olivat aluksi muistista muistiin siirtoja, toisessa vaiheessa muistista levyille siirtoja. Siirrettävän tiedon määrä oli aina joko 6.4GB tai 25.6GB. Testikoneet olivat vähintään Intel 945 tai nForce4-tason koneita. Metsähovissa käytettiin kahta Dell'in tavallista Optiplex GX620 -konetta, joiden hinta oli noin 500eur, ja yhtä itse kasattua konetta nForce4 emolevyllä. Verkkokorttien MTU-asetuksena oli tavallinen 1500 tavua, jumbokehysä ei käytetty. Etäkoneet taas kukin asema oli kasannut itse. Jodrell Bank käytti 4470 tavun jumbokehysä. Tarkempi kuvaus testikoneiden kokoonpanosta löytyy koetulosten verkkoraportista (MoW06).

Aluksi testasimme Tsunamin UDP-pakettikoon vaikutusta levyttömään siirtonopeuteen. Tulokset on esitetty taulussa 4.1. Pakettikoon kasvattaminen ei huomattavasti hidastanut siirtoja, mikä johtuu 0% pakettihäviöstä kaikilla pakettikooilla. Huonolla siirtoreitillä pakettihäviö kasvaisi pakettikokoon verrannollisesti - sama Ethernet-kehysien häviämistodennäköisyys aiheuttaa IP-lohkomisen (IP fragmentation) jälkeen suurilla eli useam-

Taulukko 4.1. Pakettikoon vaikutus paikallisessa siirrossa Dell-koneelta toiselle ilman levyjä. Nopeudet on annettu Mbit/s yksiköissä ja pakettihäviö prosentteina.

tavoitenopeus	15 kB	32 kB	45 kB	62 kB
512	520.9, 0%	-	-	-
1024	956.9, 0%	947.8, 0%	957.2, 0%	955.1, 0%

Taulukko 4.2. Tsunami-siirto palvelimen muistista asiakkaan RAID-levylle eri tavoite-nopeuksilla ja UDP-pistokkeen puskurikooilla. Nopeudet on annettu Mbit/s yksiköissä, CPU-kuormat prosentteina.

suunta	tavoitenopeus	puskuri	nopeus	pakettihäviö	palvelin	asiakas
Dell->nForce4	512	20MB	552.9	0%	99.9%	21%
Dell->nForce4	1000	20MB	862.2	11.7%	99.9%	87%
Dell->nForce4	1000	100MB	850.0	12.0%	99.9%	87%
Dell->nForce4	1000	8MB	850.0	9.0%	99.9%	86%
nForce4->Dell	512	20MB	551.0	0%	70%	22%
nForce4->Dell	1000	20MB	955.2	0.1%	75%	49%

mista kehyksistä koostuvilla UDP-paketeilla isomman häviön kuin käyttäessä pienempiä UDP-paketteja.

Tsunami-ohjelmat kuormittavat prosessoria suhteellisen paljon. Etenkin palvelinohjelmassa lähtevien pakettien aikavälin tarkka säätö kiertokyselyä käyttävällä odotuksella yrittää omia koko prosessorin. Tämä voi vaikuttaa samaan aikaan ajettujen muiden ohjelmien suorituskykyyn. Tekemällä Tsunami-siirtoja SMC 1G kytkimen yli palvelimen muistista asiakkaan RAID-levylle testasimme ohjelmistollisen RAID:in ja Tsunami-siirtojen yhteistoimintaa. Molemmissa koneissa oli neljän SATA-levyn RAID 0, jonka raitojen koko oli vain 16kB (chunksizet). RAID:in suorituskykyä olisi voinut kasvattaa käyttämällä suurempaa raitakokoa. Kokeessa vertailtiin lisäksi valmiin Dell-koneen ja itse kasatun nForce4-koneen suorituskykyeroja. Myös UDP-pistokkeen puskurikokoa muutettiin 20MB oletusarvosta. Tulokset on esitetty taulussa 4.2.

Yllättäen halpa Dell-toimistokone toimi vastaanotossa RAID-levylle paremmin kuin itse kasattu nForce4. Kummankin siirtosuunnan tulokset osoittivat molemman koneen hyvän

toimivuuden yli 512 Mbit/s e-VLBI -siirtoja varten. Dell-koneen kokoonpano oli halvemmasta päästä, mikä osoittaisi että mikä tahansa tuorehko kotitietokone kelpaa e-VLBI -siirtoihin.

Lopuksi teimme pidemmän etäisyyden siirtokokeen Jodrell Bankin ja JIVE:n koneiden välillä 1 Gbit/s kuituyhteyden yli. Aiemmin ajamani koe samojen koneiden välillä mutta Jodrell Bankin hitaamman 630 Mbit/s kuidun yli oli jo toiminut 560 Mbit/s nopeudella täysin ilman pakettihäviötä. Tämä osoitti jo 630 Mbit/s kuitujen olevan riittäviä 512 Mbit/s reaaliaikasiirroille, mikäli reitillä ei ole muuta liikennettä.

1 Gbit/s kuitutestin alussa ilmeni ongelmia MTU-asetusten kanssa, joita emme puuttuvien oikeuksien takia pystyneet päivittämään. Kuituyhteys käytti tavallista 4470 tavun MTU:ta, mutta JIVE:n päässä verkkokortin MTU oli 1500 tavua. Verkon ja käyttöjärjestelmien käyttämä polun pienimmän MTU:n etsimistekniikka (PMTU, path MTU discovery) ja siten oikea IP-lohkominen eivät jostain syystä toimineet tällä polulla. Ainoa mahdollinen lähetysuunta oli JIVE:stä Jodrell Bankiin, koska JIVE:ssä MTU oli pienin. Kokeissa siirrettiin taas 6.4GB tai 25.6GB tietomääriä muistista muistiin, ilman kovalevytallennusta. Tsunami-asiakkaassa muutettiin vain tavoitenopeutta, muut asetukset kuten UDP-puskurikoko (20MB) käyttivät oletusarvoja. Siirtojen tilastot eri siirtonopeuksilla on esitetty kuvaajassa A.1.

Kokeesta saatiin useita uusia tietoja. Esimerkiksi Tsunamin nopeuden säätö alaspäin ruuhkatilanteissa A.1 toimii vain rajallisesti - jos asiakkaassa valitaan tavoitenopeus, joka on huomattavasti suurempi kuin mitä yhteys tai reitillä oleva pullonkaula pystyvät ylläpitämään, aiheuttaa palvelimen lähettämä UDP-pakettivirta myös TCP-kontrollikanavan liikenteen tukkeutumisen. Palvelin ei saa kaikkia asiakkaan siirtoilistoraportteja eikä hidasta lähetysnopeutta alaspäin, joten ei myöskään saavuteta asiakkaan antamaa, oletusarvoisesti 7.5% kohdepakettihäviötä. Nopeutus- ja hidastumiskertoimia voi muuttaa asiakkaan komentorivin kautta, ja esimerkiksi valita nopeutuskerroin loivemmaksi. Tämän jälkeen lähetysnopeus lähestyy hitaammin pullonkaulanopeutta ja virheraportit voivat vielä päästä palvelimelle. Kokeilujen jälkeen kuitenkin osoittautui helpommaksi testata uuden siirtoyhteyden tuntematonta maksimikapasiteettia suoraan joko *iperf*-työkalulla tai kokeilemalla peräkkäisiä Tsunami-siirtoja kasvavalla tavoitenopeudella. Kun maksimikapasiteetti on selvitetty, riittää jatkossa valita tavoitenopeus siten, että se ei ylitä ainakaan kovin suuresti tätä nopeutta.

1 Gbit/s kuidun yli oli kuitenkin helppo saavuttaa 900 Mbit/s nopeus JIVE:en vain 5% pakettihäviöllä. Nopeus on täysin riittävä 512 Mbit/s e-VLBI:lle. Suuremmat nopeudet ovat helposti mahdollisia yhdistämällä useita 1 Gbit/s kuituja tai hankkimalla 10 Gbit/s kuitu, ja käyttämällä useita tietokoneita 1 Gbit/s verkkokortilla. Myös 10 Gbit/s PCI-express verkkokortit ovat mahdollisia.

Toinen kokeista saatu tulos oli, että tavalliset toimistokoneet RAID:iksi yhdistetyillä SATA-levyillä toimivat erittäin hyvin jopa 955 Mbit/s siirtonopeuksiin asti pelkällä yhdysrakenteisella 1 Gbit/s verkkokortilla ja tavallisella 1500 tavun MTU-asetuksella, ilman jumbokehyskiä. Tsunami-ohjelmien aiheuttama prosessorikuorma ei hidastanut ohjelmistollista RAID:ia haittaavasti.

Kokeessa käytettyjä nForce4 ja Intel 9x5-sarjan emolevyjä ja kokoonpanoja suositeltiin myöhemmin myös usealle toiselle observatoriolle, jotka olivat hankkineen VSIB-kortin ja halusivat kasata PC-EVN koneen.

Reaaliaikainen Tsunami jäi tässä kokeessa testaamatta, mutta kokeen tuloksista pystyttiin jo päättelemään, että siirto VSIB-ajurin muistipuskurista Tsunamilla asiakaskoneen kovalevyille pitäisi muistista levyille -siirtojen tapaan toimia täysin vaivatta vähintään 512 Mbit/s nopeuksilla. Myöhemmissä siirtokokeissa ja alla kuvattavassa Month7 demonstraatiossa tämä päätelmä osoittautui oikeaksi.

## 4.2 Kolmen aseman e-VLBI Month7 EU-demonstraatio

EXPreS-projektissa oli vuoden 2006 lokakuussa tarkistuspiste, jossa oli esitettävä Metsähovissa tehty e-VLBI kehitystyö ja demonstroitava COTS-komponenteista PC-EVN-järjestelmän käyttökelpoisuus reaaliaikaisessa e-VLBI:ssä. Alkujaan kokeen tavoitteena oli siirtää reaaliajassa usean aseman näytteitä 128 - 512 Mbit/s nopeuksilla JIVE:n korrelaattoriasemalle (Hollanti), jonka piti toimia siirtojen asiakkaana. JIVE:n verkkoyhteys ja käytössä olevat tallentavat tietokoneet olivat tosin riittämättömiä. Saatavilla oli vain yksi 1 Gbit/s kuituyhteyteen kytketty Mark5 tietokone usealla RAID 5 levyllä. Metsähovi taas oli maailman ensimmäinen observatorio, jolla oli 10 Gbit/s kuituyhteys. Lisäksi meillä oli 5 e-VLBI testikonetta RAID 0 levypakoilla. Month7 e-VLBI -siirtojen asiakkaan rooli päättyi tämän takia Metsähoville. Ottaen huomioon käytettävissä olevien asemien antennit



ja vastaanottimet, koe päätettiin tehdä 5 GHz havainnointitaajuudella Ruotsin Onsalan (On) ja Englannin Jodrell Bankin (Jb) radioteleskoopeilla. Asemien skannit korreloitiin JIVE:ssä. Korrelointia ei ollut mahdollista suorittaa reaaliajassa, koska JIVE:ssä oli vain yksi Mark5 tietokone RAID-kovalevyllä, eikä yksi kone riitä syöttämään JIVE:n laitteistopohjaista korrelaattoria kaikkien asemien skanneilla.

Month7-demonstraatio oli jaettu kolmeen havainnointiosaan (A-C). Ennen koetta näille osille suunniteltu aikataulu ja osien tavoitteet on luetteloitu alla. Kuten tyypillistä VLBI:ssä, päivä ennen varsinaista Month7-koetta ja myös kokeen aikana suunnitellut päätettiin toteuttaa osittain hieman eri tavalla, pääasiassa johtuen aliarvioidusta kapasiteetista ja ongelmista JIVE:n päässä. Yksityiskohtia lukuun ottamatta toteutettu koe kuitenkin eteni perussuunnitelman mukaisesti:

**Osa A** 0900 -1000 UT. Demonstroidaan off-line e-VLBI -moodi. Havainnoidaan kaksitoista 4 minuutin skannia 1 minuutin välitauoilla. On ja Jb tallentavat levyille 512Mbit/s nopeudella. Skannitiedostot siirretään myöhemmin Tsunami-protokollalla JIVE:n tietokoneelle ja sieltä korreloitavaksi.

**Osa B1** 1030 - 1110 UT. Demonstroidaan reaaliaikainen e-VLBI -moodi paikallisella varmuuskopiolla. Havainnoidaan kahdeksan 4 minuutin skannia 1 minuutin välitauoilla. Jb ja/tai On yrittävät reaaliaikaista Tsunami-e-VLBI -siirtoa Metsähovin levyille 256 Mbit/s nopeudella. Käytetään varmuuskopio-moodia nauhoittamalla skannit samanaikaisesti myös asemien levyille. Myöhemmin osoitetaan siirretyt tiedostot identtisiksi ja e-VLBI siirto häviöttömäksi. Seuraavana päivänä tiedostot siirretään hitaasti FTP:llä JIVE:en korreloitavaksi.

**Osa B2** 1120 - 1200 UT. Kuten osa B1, mutta 512Mbit/s nopeudella.

**Osa C** 1230 - 1330 UT. Jb ja/tai On yrittävät reaaliaikaista e-VLBI -siirtoa JIVE:n tietokoneelle. Muut asemat tallentavat skannit paikallisesti levyille ja siirtävät ne myöhemmin JIVE:en. Tiedot konvertoidaan JIVE:ssä korrelointia varten. Havainnoidaan kaksitoista 4 minuutin skannia 1 minuutin välitauoilla.

Onsala ja JIVE laativat havainnointiaikataulun ja valitsivat havainnoitavat tähtitieteelliset kohteet. Kokeen jokaisessa osassa havainnointi suoritettiin Tsunami- tai kovalevytallennusskriptein avulla automatisoituna. Nämä skriptit loin edellisessä kappaleessa 3.5 mainituilla generointiskripteillä.

Jotta PC-EVN -tiedonkeruu ja reaaliaikasiirto olisivat mahdollisia, Jb ja On varustet-

tiin tarpeellisilla VSIB-korteilla ja VSIC-konverttereilla. Asemilla koottiin omat PC-EVN-koneet hyvissä ajoin ennakkoon. Molemmille asemille lähetimme PATA-levyt, jotka sisälsivät kopion kesällä kokoamastani Debian 3.1 Linux -referenssijärjestelmästä (Wag06-2). Tämä järjestelmä, johon oli valmiiksi asennettu kaikki tarvittavat VSIB:iin liittyvät lisäohjelmat, sekä verkkodiagnoosissa hyödyllisimmät Linux-apuohjelmat, yksinkertaisti PC-EVN -koneiden käyttöönottoa asemilla. Month7-kokeessa kummallakin asemalla oli käytössä yksi PC-EVN. Molemmilla asemilla oli Internet-yhteytenä 1 Gbit/s valokuitu. Jodrell Bankin liikenne JIVE:en kulki 1 Gbit/s valopolun yli. Reitti Jodrell Bankista ja Onsalasta Metsähoviin kulki ilman valopolkupalveluja tavallisten tuotantolinkkien ja GEANT2:en tavallisten reittien yli. 10 Gbit/s GEANT2 tutkimusverkon reittien kuormitusaste oli myös vuonna 2006 tyypillisesti 5% - 20%, joten e-VLBI -liikenteelle löytyy yleensä enemmän kuin riittävästi vapaata kaistaa. Jodrell Bankin tapauksessa liikenne Metsähoviin kulki ensin Euroopan-laajuisen GEANT2 10 Gbit/s tutkimusverkon kautta Ruotsiin ja siirtyi sieltä yhdessä Onsalan liikenteen kanssa NORDUnet:in ja Funet:in kautta kulkevaa reittiä Metsähoviin. Liikenne Onsalasta JIVE:en ei myöskään käyttänyt valopolkua.

Metsähovissa käytimme vastaanottoon PC-EVN tietokoneita yhdysrakenteisilla 1 Gbit/s verkkoliitännöillä. Koneet oli kytketty ExtremeNetworks Summit X450 reitittimen kautta 10 Gbit/s valokuidun taakse. Oli etukäteen tiedossa, että TKK:n kampuksen ja Helsingin keskustan välillä oli 2.5 Gbit/s pullonkaula.

Asemien yhteenlaskettuna kaikissa viidessä tietokoneessa oli RAID -levypakat, joiden nopeus oli vähintäänkin riittävä tallentamaan skanneja kokeessa käytettävillä suurilla tiedonkeruunopeuksilla.

Pari päivää ennen varsinaista koetta varmistin muiden asemien VLBI-henkilökunnan ystävällisellä avulla, että kokeen tekniset edellytykset oli täytetty. Testasin JIVEN:n ja asemien välisiä verkkoreittejä ja niiden kapasiteettia *iperf*-testiohjelmalla. Tarkastin myös asemien PC-EVN -järjestelmien suorituskyvyn ajamalla Tsunami-tiedostosiirtoja palvelimen RAID-levyltä asiakkaan RAID-levylle eri siirtonopeuksilla. Lopuksi siirsin vielä reaaliajassa VSI-H -testidataa Metsähovin PC-EVN koneilta asemille reaaliaikaisella Tsunamiilla. Muiden asemien ja Metsähovin välillä ajetut yksittäiset Tsunami-tiedonsiirrot toimivat kaikki vähintään 720 Mbit/s nopeuksilla. 512 Mbit/s reaaliaikasiirroille oli siis mukavasti 200 Mbit/s lisämarginaalia yli tavoitenopeuden. Juuri ennen koepäivää varmis-

tin uudestaan Tsunamilla, että verkkoyhteyksissä ei ollut tapahtunut muutoksia.

Koska ensimmäisten esikokeiden siirrot toimivat hyvin yli 512 Mbit/s nopeuksilla ja Tsunami raportoi pakettihäviöksi 0.0%, päätimme Metsähovissa kokeilla lyhyttä siirtoa omille testikoneillemme kahdelta asemalta samanaikaisesti, yhteensä 1.4 - 2 Gbit/s kokonaisnopeudella, FUNET:ia etukäteen varoittamatta. Valitettavasti Helsingin 2.5 Gbit/s pullonkaula ei kestänyt tätä kuormaa, ja siirroissa oli jonkin verran pakettihäviötä (kuvan A.2 ylempi kuvaaja). Month7-kokeen osaa B2 vastaava yhteenlaskettu 1 Gbit/s liikenne siirtyi kuitenkin ilman pullonkaulaongelmia.

Esikokeissa ilmeni vielä joitakin korjaustarpeita PC-EVN:n asennusdokumentaatiossa ja ohjeissa. Esikokeiden jälkeen päivitin myös PC-EVN-tiedonsiirtoon liittyvät skriptit parempaan muotoon, ennen kuin annoin skriptit ja ohjeet eteenpäin Onsalan, Jodrell Bankin ja JIVE:n asemille, joissa skriptien oletusarvoiset asetukset sitten muutettiin asemakohtaisesti sopiviksi.

Metsähovin yhteyden 2.5 Gbit/s pullonkaula ei osoittautunut ongelmaksi 1 Gbit/s siirroille, mutta kerroimme FUNET:ille varmuuden vuoksi tulevista Month7 e-VLBI -siirroista etukäteen, jotta yhteys pysyisi koepäivänä mahdollisimman vapaana.

Asemia olimme jo etukäteen opastaneet VSI-kaapeloinnissa ja VSIC-konverttereiden oikeissa asetuksissa. Esikokeiden lyhyet reaaliaikasiirtotestit toimivat hyvin, joten asetukset vaikuttivat olevan kunnossa.

Varsinainen Month7-koe alkoi 20. lokakuuta 2006 ja eteni suunnitellun aikataulun ja osiojaon mukaisesti. Kokeen aikana asemat olivat yhteydessä Skype-konferenssin kautta. Osioiden suoritustapa ja tulokset on esitetty osioittain alla.

Osa A, off-line e-VLBI: ensimmäiset kuusi skannia tallentuivat onnistuneesti 512 Mbit/s, jonka jälkeen tallennus keskeytettiin ja asemien ensimmäinen skanni siirrettiin JIVE:en korreloitavaksi. Siirto Tsunamilla JIVE:en toimi yli 600 Mbit/s siirtonopeudella. Korrelointi ja ensimmäisten tulosten saaminen viivästyi JIVE:n paikallisverkon ja Mark5A-koneen ongelmien takia - JIVE:n tuli kopioida skannit asiakastietokoneeltaan paikallisverkon yli Mark5A-koneelle ja sieltä levypakalle.

Osat B1 ja B2, reaaliaikainen e-VLBI paikallisella varmuuskopiolla: ensimmäisen osan 256 Mbit/s reaaliaikasiirrot Metsähoviin toimivat tasaisella nopeudella ja täysin ongel-

mitta. Siirtonopeuskuvaajat B1-osan ensimmäiselle skannille on esitetty kuvissa A.3 ja A.4. B2-osan skannit siirtyivät myös hyvin, kuten Onsalan erään B2-skannin kuvaaja kuvassa A.6. Jodrell Bankin siirtonopeudessa oli kuitenkin yhden B2-skannin kuvaajan A.5 mukaisesti enemmän vaihtelua kuin B1-osassa. Kokeiden jälkeen selvisi, että VSIB-ajurin puskurikokoasetuksen arvosta puuttui yksi nolla, puskuri oli 14.4MB 144MB sijaan.

Osan B2 kolmannen skannin aikana Jodrell Bankin yhteyden keskimääräinen kapasiteetti tippui alle 512 Mbit/s nopeudelle. Siirtokuvaaja on esitetty kuvassa A.7. Huonon linkin takia skannin siirtoaika venyi VSIB-ajurin rengaspuskurin kokoon nähden liikaa ja skannin tietoja menetettiin ajoittain. Siirtoaika venyi lisäksi yli skannien välisen 60 sekunnin tauon, jolloin neljäs skanni jouduttiin jättämään väliin. Myöhemmin päivittämissäni siirtoskripteissä tämä mahdollisuus otetaan huomioon ja venynyt siirto katkaistaan ennen seuraavaa skannia.

Onsalan B2 512 Mbit/s -siirrot sujuivat kaikki ongelmitta.

Osa C, Jb ja On reaaliaikainen 512 Mbit/s siirto JIVE:n tietokoneelle: osa oli vähällä jäädä aloittamatta, koska osassa A JIVE:n paikallisesti aloittama skannien kopiointi heidän asiakaskoneeltaan Mark5A-koneelle kesti erilaisten ongelmien takia yllättävän kauan. Kopiointi saatiin kuitenkin päätökseen juuri ennen C-osaa. Osa päätettiin toteuttaa hieman toisin kuin alun perin suunniteltu - Jodrell Bankista ajettaisiin yhä reaaliaikasiirtoja JIVE:en, mutta lisäksi käynnistettäisiin toinen siirto Onsalasta Metsähoviin. Osa C onnistui valitettavasti vain osittain. JIVE:ssä oli muutettu A-osan ensimmäisen skannin paikallista kopiointia varten useita verkkoasetuksia eikä alkuperäisiä asetuksia muistettu palauttaa ennen C-osaa. JIVE:n korjausyritykset tunnin kestävän C-osan aikana eivät onnistuneet. Väärät asetukset johtivat siihen, että Jodrell Bankista tulevan UDP-liikenteen vastaanotto JIVE:ssä ei enää toiminut. Siirrot Onsalasta Metsähoviin toimivat kahta skannia lukuun ottamatta erittäin hyvin, mutta ainoastaan Onsalasta saadut skannit eivät riittäneet korrelointiin.

Osien A ja B skannien korrelointi JIVE:ssä onnistui menestyksekkäästi. JIVE:n useista tietokoneongelmista johtuen A-osassa ensimmäisen paikallisesti tallennetun 512 Mbit/s skannin korrelointitulos oli saatavilla vasta hieman ennen C-osan aloitusajankohtaa. Tuloksista löytyi interferenssikuvio, joten koe oli menestys ja asemien tiedonkeruu sekä Tsunami-siirto olivat kaikki todistetuksi toimineet kuten odotetusti oikein. Tarkemmat

korrelointitulokset löytyvät alkuperäisestä Month7 EU-raportista (CWK06). Osasta C korrelointituloksia ei pystytty laskemaan, koska skanneja saatiin siirrettyä vain Onsalan asemalta.

Month7-demonstraatiossa ylitettiin alkujaan asetetut tavoitteet. Demonstraation perustavoite, löytää kuluttajätietokoneilla tallennetuista skanneista korreloinnin jälkeen interferenssikuvio eli osoittaa PC-EVN-järjestelmän käyttökelpoisuus VLBI-tiedonkeruuseen, oli täytetty jo A-osan ensimmäisellä skannilla.

Molempien radio-observatorioiden ja näiden PC-EVN-järjestelmien sekä reaaliaikasiirtojen toimivuus vastasi vähintäänkin tuotantokäytössä olevaa vanhaa Mark5-järjestelmää. Kokeet todistivat UDP-pohjaisen Tsunami-siirron toimivan molemmissa moodeissa, offline VLBI tiedostonsiirrossa ja reaaliaikaisessa 256 Mbit/s siirrossa asemalta asiakkaan RAID-levylle. Osan B2 siirtokokeet osoittivat myös 512 Mbit/s reaaliaikasiirtojen periaatteellisen toimivuuden - ja itse asiassa 512 Mbit/s PC-EVN reaaliaikasiirto oli ensimmäinen tämän nopeusluokan reaaliaikainen e-VLBI -siirto EVN:ssä.

Kokeet osoittivat Tsunami-protokollalla tehdyn UDP-pohjaisen tiedonsiirron toimivan hyvin. Useat muut observatoriot testaavat yhä TCP-protokollaa ja siitä laadittuja muunnelmia TCP:n oletetun verkkoystävällisyyden takia. TCP:n verkkoystävällisyys on tosin ristiriidassa reaaliaikasiirron nopeus- ja aikarajoitusten kanssa. Tämän takia Metsähovissa seuraavaksi kehitettävässä  $> 1$  Gbit/s tiedonkeruujärjestelmässä tullaan käyttämään UDP-pohjaista tiedonsiirtoa. Myös Haystackin kehitteillä oleva Mark 5C -järjestelmä käyttää UDP-tiedonsiirtoa.

### 4.3 896 Mbit/s siirtotestin tulokset

Tämä koe suoritettiin hieman Month7-demonstraation jälkeen Jodrell Bankin avulla. P. Burgess ja S. Casey hoitivat Jodrell Bankin laitteiden asennukset ja peruskonfiguroinnit. He lainasivat testikäyttöön kaksi testikonetta (pcevn1, pcevn2), jotka molemmat oli kytketty kuidulle reitittävän 3Com 1740 -kytkimen taakse. Aseman 1 Gbit/s kuituyhteys toimi sopivana pullonkaulana, jonka läpi pystyi kokeilemaan kirjoittamaani 896 Mbit/s Tsunamisiirtoa ja hyödyntämään koko kuidun kapasiteettia e-VLBI -tiedonsiirroille.

Jodrell Bankissa siirtokokeeseen tarjotut tietokoneet olivat neljä vuotta vanha Supermic-

ro 1.8G ja uudempi, itse kasattu kone M2N-E emolevyllä ja nForce 570 -piirisarjalla. Molemmissa koneissa oli VSIB-kortti ajureineen. VSIB:it oli liitetty VSIC-konvertterin kautta Mark4-formatteriin. Vain yhdessä koneista oli RAID-järjestelmä. Toisen koneen puuttuvan RAID:in takia päätettiin käyttää Jodrell Bankin koneita reaaliaikapalvelimina ja Metsähovissa kahta RAID-levyllistä konetta asiakkaina (Didius, Juliano).

Metsähovissa lähiverkossa ajamissani ensimmäisissä testeissä kaksi palvelin-PC-EVN:ää oli kytketty 1 Gbit/s pullonkaulaa simuloivaan SMC-kytkimeen, joka vuorostaan oli kytketty yhteen Extreme Networks X450 10 Gbit/s -kytkimen vapaaseen 1 Gbit/s porttiin. Samaan X450-kytkimeen oli liitetty kaksi PC-EVN -asiakastietokonetta.

Näissä testissä ilmeni, että halpa SMC-kytkin ja myös Extreme Networks -kytkin, jota jopa mainostettiin linjanopeudella toimivaksi, eivät käytännössä selvinneet purskeisesta liikenteestä. Tietokoneiden verkkokortit lähettävät tietoja mikropurskeina, eivätkä kytkimet osanneet aggregoida näitä kahta purskeista UDP-pakettivirtaa samaan ulostuloporttiin ilman tämän portin puskurimuistin ylivuotoa. GEANT2-verkon Performance Emergency Response Team -ryhmä vahvisti, että tämä on tyypillinen ongelma kytkimissä. Parhaaksi ratkaisuksi he ehdottivat kalliimman laitteen hankkimista, jossa olisi enemmän puskurimuistia.

Kertaluontoista 896 Mbit/s siirtodemonstraatiota varten ratkaisin ongelman kuitenkin eri tavalla, reitittämällä Tsunami-palvelinkoneen liikenne läpi toisesta palvelinkoneesta, jossa oli kaksi verkkoliitäntää. 896 Mbit/s kokonaisliikenteen aggregoinnin suoritti siis tietokone eikä kytkin. Kaksi verkkoliitäntää sisältävä kone konfiguroitiin tekemään IP-osoitemuunnos (network address translation, NAT) ja porttiforwardointia. Ensimmäiseen verkkokortin taakse oli konfiguroitu oma yksityinen IP-aliverkko. Tähän liitettiin ristikytkenäkaapelilla ensimmäinen reaaliaikainen Tsunami-palvelinkone. NAT-kone siirsi tämän palvelinkoneen lähettämän 384 Mbit/s ja oman lähtevän 512 Mbit/s liikenteen toisen verkkokortin kautta yhtenä virtana 10 Gbit/s kytkimeen, joka reititti liikenteen eteenpäin Internetiin. Verkon rakenne on havainnollistettu kuvassa A.8. NAT-koneeseen määritetyn porttiforwardoinnin ansiosta asiakas voi avata yhteyden aliverkossa sijaitsevaan Tsunami-palvelinkoneeseen ja voi pyytää tiedonsiirron aloittamista.

Lähiverkkotesteissä yllä kuvattu toteutus toimi jo varsin hyvin, mutta oikeiden *iproute2* reititysasetusten tekeminen ja jälkikäteen muuttaminen oli suhteellisen hankalaa. Lopullisessa toteutuksessa valitsin tämän takia NAT-palomuurin sijaan tavallisen tuntumatto-

man Ethernet-siltauksen. Ensimmäisen palvelinkoneen yksityinen aliverkko-osoite vaihdettiin takaisin julkiseksi IP-osoitteeksi. Linuxin *bridge-utils* -paketin asentamisen jälkeen kolme komentoa riitti Ethernet-siltauksen pystyttämiseen. Siltauskone näkyi tämän jälkeen myös ulkoa julkisessa IP-osoitteessa. Ensimmäisen koneen IP-osoitteeseen tulevat tai koneelta lähtevät paketit reitittyivät siltauksen yli. Kaapelointi pysyi samana kuin kuvassa A.8.

Jodrell Bankissa henkilökunta kytki kaksi konetta aiemman kuvan A.8 mukaisesti ja suoritti lähettämäni konfiguraatioskriptit. Siirrot toimivat heti, ilman lisätoimenpiteitä. Yli minuutin kestäviä 896 Mbit/s reaaliaikasiirtoja Tsunamin häviöttömässä siirtomoodissa toistettiin useaan kertaan, jotta siirroista kertyisi tarpeeksi lokitiedostoja. Siirtojen jälkeen Jodrell Bankin palvelinten ja Metsähovin asiakaskoneiden Tsunami-lokitiedostot kopioitiin talteen.

Tyypillisiä yhden 896 Mbit/s -siirron osakuvaajia ovat kuvan A.9 Didius-koneen 512 Mbit/s ja kuvan A.10 384 Mbit/s Juliano-koneen siirtokuvaajat. Yhdistetyt UDP-pakettivirrat siirtyivät 1 Gbit/s kuitulinkin yli varsin pienellä UDP-pakettien uudelleenlähetysmäärällä. Testitulokset osoittavat, että reaaliaikaisia Tsunami-siirtoja 896 Mbit/s kokonaisnopeudella voidaan suorittaa ilman ongelmia 1 Gbit/s kuidun yli, ja siirtoja voidaan harkita jopa tuotantokäyttöön. Testin päätarkoituksena oli kuitenkin osoittaa, että Metsähovin PC-EVN-järjestelmällä on myös käytännössä mahdollista ottaa lähes koko kuidun kapasiteetti hyötykäyttöön. Samalla Metsähoville tulisi taas uusi maailmanennätys VLBI-siirtokokeissa. Siirtotestin kaikki nämä tavoitteet onnistuivat yli odotusten.

#### 4.4 Tuotantokäyttö geodesiassa

Vuoden 2007 helmikuusta alkaen useat asemat ovat siirtyneet käyttämään Metsähovin PC-EVN-järjestelmää tiedonkeruuseen geodeettisessa VLBI:ssä. Month7-demonstraation jälkeen Bonnin Max-Planck -instituutissa sijaitseva geo-VLBI -korrelaattorikeskus sai 1 Gbit/s verkkoyhteyden Deutsches Forschungsnetz eV (DFN) ylläpitämään 10 Gbit/s testiverkkoon (VIOLA). Bonn korreloi EVN:n geodeettiset havainnot ja on kehittämässä nopeampaa tulospalvelua. Tavoitteena on saada IVS Rapid -sarjan sekä viikoittaisten IVS Intensive VLBI-havainnointisarjojen tiedot siirrettyä verkon yli Bonniin, jotta asemien tietojen korrelointi voidaan aloittaa myöhäisintään pari tuntia VLBI-havainnoinnin

päädyttyä. Näin korrelointitulokset tulevat geodeettien käyttöön vielä samana päivänä.

Tällä hetkellä e-VLBI -siirrot suoritetaan joitakin lyhyitä testiajoja lukuun ottamatta tavallisella Tsunamilla, off-line VLBI -moodissa. Skannit nauhoitetaan PC-EVN -koneen RAID-levylle VSIB-apuohjelmien ja automatisointiskriptien avulla, ja siirretään kokeen jälkeen Tsunamilla. Bonnin kapasiteetti ei tällä hetkellä riitä usean aseman reaaliaikasiirroille, mutta yhden aseman reaaliaikaiset Tsunami-testiajot ovat tiedonsiirron puolesta olleet menestyksekkäitä.

Ensimmäinen Metsähovin jälkeen PC-EVN e-VLBI -havainnointiin siirtynyt asema oli Onsala Ruotsissa. Varmuuden vuoksi Onsala käytti ensimmäiset kuukaudet PC-EVN:n kanssa rinnakkain Mark5-järjestelmää ja lähetti levypakat Bonniin, mutta siirtyi toukokuussa täysin PC-EVN:n tuotantokäyttöön ja e-VLBI -siirtoihin. Saksan Wettzell-asema otti elokuussa käyttöön kaksi PC-EVN-konetta, joilla tallentaa ja siirtää nykyisin joka maanantain Intensive-sessiot Bonniin. Samoihin Intensive-sessioihin osallistuvat Ny-Ålesund Norjassa ja Tsukuba Japanissa. Näiden Mark5- ja tavallisilta RAID PC -koneilta skannit siirretään Tsunamilla Bonniin.

Bonnin korrelaattoriryhmä ylläpitää omaa e-VLBI -sivustoa, jolle julkaistaan säännöllisesti raportteja ja lokitiedostoja Tsunami-siirroista (MPI07). Useat muut EVN:n asemat ovat vuoden 2007 aikana saaneet 1 tai 10 Gbit/s kuituyhteyden ja ovat osoittaneet kiinnostusta PC-EVN-järjestelmän tai Tsunami-tiedonsiirron käyttöä kohti, joten oletettavasti tämä diplomityön kehitystulosten käyttäjämäärä tulee tulevaisuudessa vielä kasvamaan.



## Luku 5

### Yhteenveto ja tulevaisuudennäkymä

Tässä diplomityössä kuvattiin VLBI:ssä käytettyjen tiedonkeruujärjestelmien yleisiä ominaisuuksia, ja esitettiin Metsähovissa vuonna 2003 kehitetyn, mutta sen jälkeen EVN:ssä hetken hiljaiseloa viettäneen PC-EVN -järjestelmän perusominaisuudet. Esiteltiin kansainvälisen VLBI:n tekninen kehityssuunta ja teknisten tavoitteiden tieteelliset perusteet, sekä syyt käyttää tietoverkkoja VLBI-asemien tietojen siirtoon. VLBI:n tietoliikennepuolta edistää eurooppalainen EXPReS-projekti.

EXPReS-projektiin liittyviin kehitystöihin sisältyi VSIB-kortin ajurin ja apuohjelmien päivittäminen uudenaikaiseen Linux-järjestelmään uudella 2.6-sarjan ytimellä. Metsähovin aiemmin kokeilema ja kansainvälisesti demonstroima Indiana University Tsunami-tiedonsiirtoprotokolla ja Tsunami-ohjelmisto vaativat suuren määrän virheiden korjauksia ja yleiskäytettävyyden parannuksia, joiden jälkeen tuotantokäyttöön VLBI:ssä tuli mahdolliseksi. Reaaliaikainen e-VLBI vaati Tsunamin reaaliaikatoteutukseen sekä korjauksia että useita uusia ominaisuuksia ja toimintoja. Observatoriot toivoivat myös helppoa tapaa VLBI-kokeiden tiedonkeruun automatisointiin. Vanhat käsin laadittavasta ja päivitettävästä kovalevytallennusskriptit päivitin uudella sarjalla generointiskriptejä, jotka muuntavat VLBI-kokeen aikataulutiedoston kovalevytallennus- tai reaaliaikasiirtoskripteiksi. Näiden avulla VLBI-kokeen tiedonkeruu PC-EVN -järjestelmällä toimii pitkälti automatisoituna ja vaatii käyttäjältä huomattavasti vähemmän perehtymisaikaa ja tiedonkeruun esivalmistelutöitä.

Vaikka uuden PC-EVN -järjestelmän kokoaminen ja tarvittavien ajureiden sekä ohjelmien asentaminen on periaatteessa helppoa, useat observatoriot tarvitsivat asennusapua ja käyttäjätukea. Tähän tarpeeseen loin muun muassa Debian 3.1 PC-EVN Linux

-referenssijakelun, joka sisältää vaaditut asetukset ja kaikki tarvittavat ohjelmat sekä ajurit valmiiksi asennettuna. Kirjoitin myös järjestelmän käyttöönottoon tarpeellisen dokumentaation ja laadin omat asennusohjeet VSIB-kortin päivitetylle ajurille ja apuohjelmille.

Tsunamiin, VSIB-ajuriin ja automatisointiskripteihin liittyneet kehitystyöt mahdollistivat PC-EVN -tiedonkeruujärjestelmän täyden menestymisen EXPReS-projektin Month7-demonstraatiossa. Demonstraatiossa aitoja VLBI-tietoja siirrettiin Tsunamilla eri nopeuksilla sekä off-line- että reaaliaikamoodeissa. Tämä osoitti EVN:n VLBI-yhteisölle, että Metsähovin järjestelmä on enemmän kuin käyttökelpoinen off-line -tiedonkeruuseen ja reaaliaikaiseen e-VLBI -tiedonsiirtoon. Demonstraatiossa jopa ylitettiin EXPReS-projektissa sille asetetut tavoitteet.

Month7-demonstraation jälkeen lisäsin Tsunamiin uusia ominaisuuksia, joista tässä työssä esiteltiin tarkemmin 1 Gbit/s kuidun koko kapasiteetin hyödyntävä 896 Mbit/s siirtomoodi. Jodrell Bankin kanssa tehdyssä kokeessa uusi siirtomoodi osoittautui toimivan erittäin hyvin. Koe oli myös maailman ensimmäinen VLBI-siirto, jonka kokonaisnopeus ei ollut kahden potenssi. Siirtomoodin toimintaperiaatetta voidaan soveltaa sekä vanhoilla 1 Gbit/s että uusilla 10 Gbit/s kuituyhteyksillä.

Yllättävää koko kehitystyön tulosten tämänhetkisessä käyttötilanteessa on, että oikeastaan Euroopan tähtitieteellistä VLBI:n yhteisöä varten laadittua ja myös EU EXPReS -projektissa Metsähovilta vaadittua tiedonkeruujärjestelmää ei olla vielä todellisesti otettu käyttöön kovin monessa EXPReS-projektiin osallistuvassa observatoriossa, ei edes Month7-demonstraation jälkeen. EVN:n astronomisen korrelaattorin JIVE:n verkkokapasiteetti ja tietokonejärjestelmät eivät ole olleet tähän riittäviä. Myös EVN:n poliittinen tahto on puuttunut. Sen sijaan EXPReS-projektiin vain epäsuorasti liittyvä geodeettinen VLBI-yhteisö, johon kuuluu myös joitakin EXPReS-projektiin osallistuvaa asemaa, on huomannut Metsähovin järjestelmän ja tekemäni kehitystyön. Bonnin Max-Planck-instituutin geodeettisen korrelaattoriryhmän myötävaikutuksella useissa näistä observatorioista on aloitettu PC-EVN-järjestelmän ja Tsunami-siirtojen testaus ja osin myös tuotantokäyttö.

PC-EVN-järjestelmän suosio on geodeettien puolella yhä kasvussa. Vaikka siis tämän diplomityön puitteissa tehdyt EXPReS-projektin vaatimat kehitystyöt eivät vielä ole löytäneet tietään EXPReS-projektin astronomiseen VLBI-käyttöön, työt ovat onneksi osoit-

tautuneet varsin hyödyllisiksi muualla VLBI:ssä.

Tässä diplomityössä kuvattu kehitystyö loi osaltaan perustan toimivalle ja helppokäyttöiselle e-VLBI -tiedonsiirrolle ja tiedonkeruulle, joka käyttää tämän hetken PC-EVN -järjestelmää. Näin siis e-VLBI:n päätavoite, tiedon siirtäminen postin sijaan verkon yli korrelaattoriasemalle, joko reaaliajassa tai off-line moodissa, on saavutettu.

Diplomityön aikana kehittämiäni Tsunami-protokolla löytää toivon mukaan uusia sovellutusalueita myös VLBI:n ulkopuolella. Tavalliseen TCP-tiedonsiirtoon verrattuna Tsunami on ylivertainen sekä siirtonopeudeltaan että siirron käyttäytymisen ja aikarajojen tarkan hallittavuuden takia. Esimerkiksi hiukkasfyysikoiden tiedonsiirrot, Grid-laskentaverkot ja reaaliaikaiset Internet-televisiolähetykset saattavat hyötyä Tsunami-protokollasta.

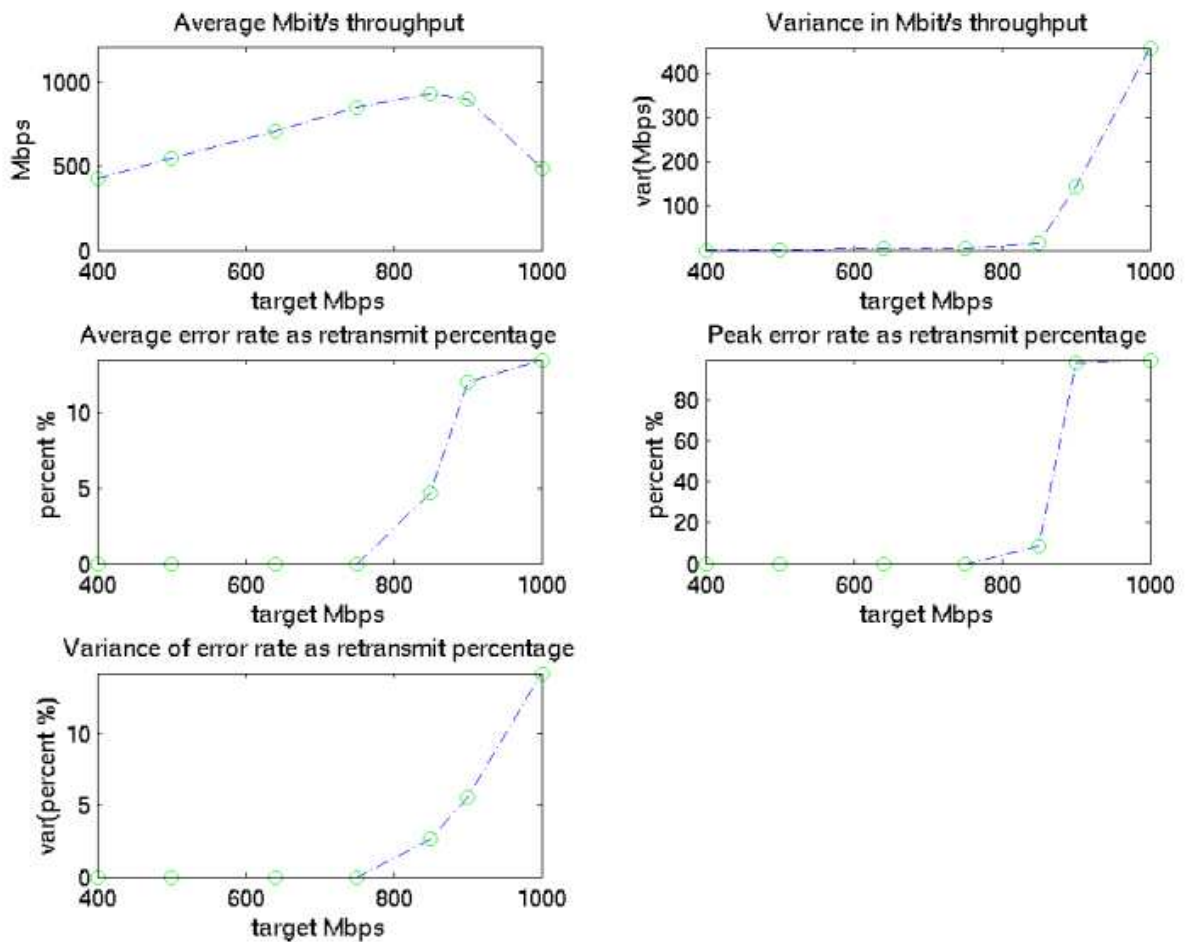
VLBI-tiedonkeruujärjestelmissä seuraava askel tulee olemaan nykyisten antennisignaalia näytteistävien järjestelmien päivittäminen siten, että leveämmät RF-taajuuskaistat, parempi VLBI-järjestelmien herkkyys ja tarkemmat tähtitieteelliset kuvat todella tulevat mahdollisiksi. Tätä myöten kasvavat tietonopeudet tulevat myös lähestymään Tsunamilla jo nyt mahdollisia korkeita tiedonsiirtonopeuksia.

Metsähovissa tulemme jatkossa keskittymään EXPReS-kehitystyössä 10 Gbit/s tiedonkeruujärjestelmään, jossa täyden 512 MHz kaistanleveyden sallivan A/D-muunninkortin antamat näytteet kerätään FPGA-kortilla ja siirretään suoraan 10 Gbit/s kupari- tai kuituyhteydelle. Laitteistoalusta on jo olemassa ja on Berkeley Wireless Research -tutkimuskeskuksen kehittämä, mutta FPGA-ohjelmaan on Metsähovissa tehtävä useita muutoksia, jotta laite soveltuisi VLBI-käyttöön.

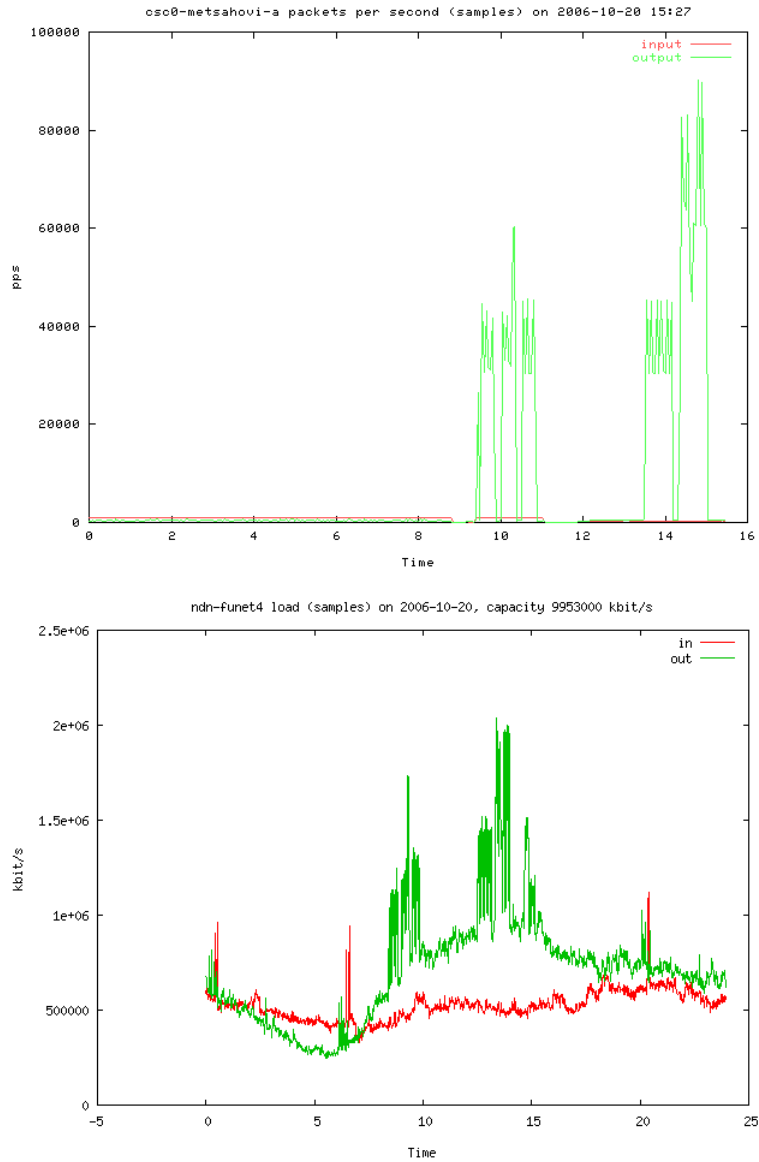
Järjestelmän FPGA-kortille on laadittava muun muassa sopiva tiedonsiirtoprotokolla. Tähän voidaan todennäköisesti soveltaa ainakin osittain nykyisiä Tsunami-protokollan kehitystyön tuloksia ja skriptipohjaista VLBI-kokeen tiedonkeruun automatisointimenetelmää. Järjestelmä tulee myös käyttämään toimivaksi osoitetun 896 Mbit/s siirtomoodin toimintaperiaatetta, eli kokonaislähetysnopeus sovitetaan verkon saatavissa olevaan kapasiteettiin tarpeen mukaan karsimalla lähetettäviä kanavia. Kehitystyö tulee alkamaan 2007 lokakuussa. Tärkein Metsähovissa kehitettävän järjestelmän käyttäjä tulee olemaan Jodrell Bankin observatorio ja sen e-MERLIN -teleskooppiryhmä. Myös muut observatoriot saattavat jatkossa kiinnostua järjestelmästä.

## Liite A

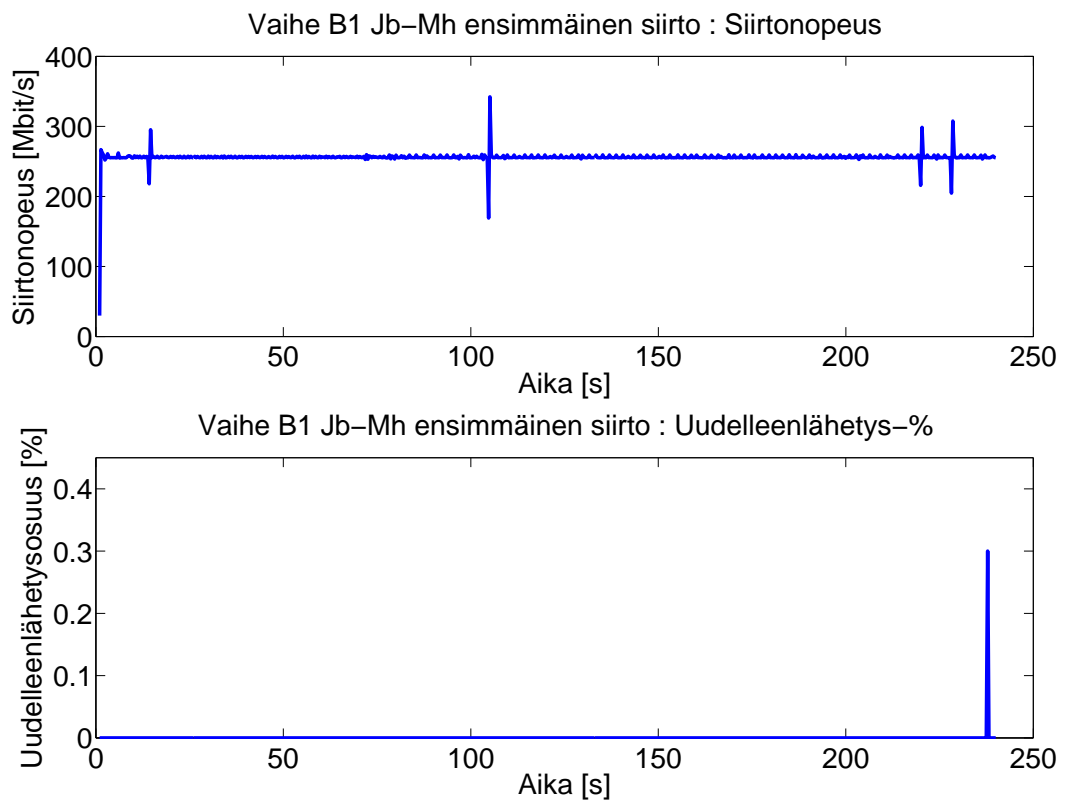
## Liitteet



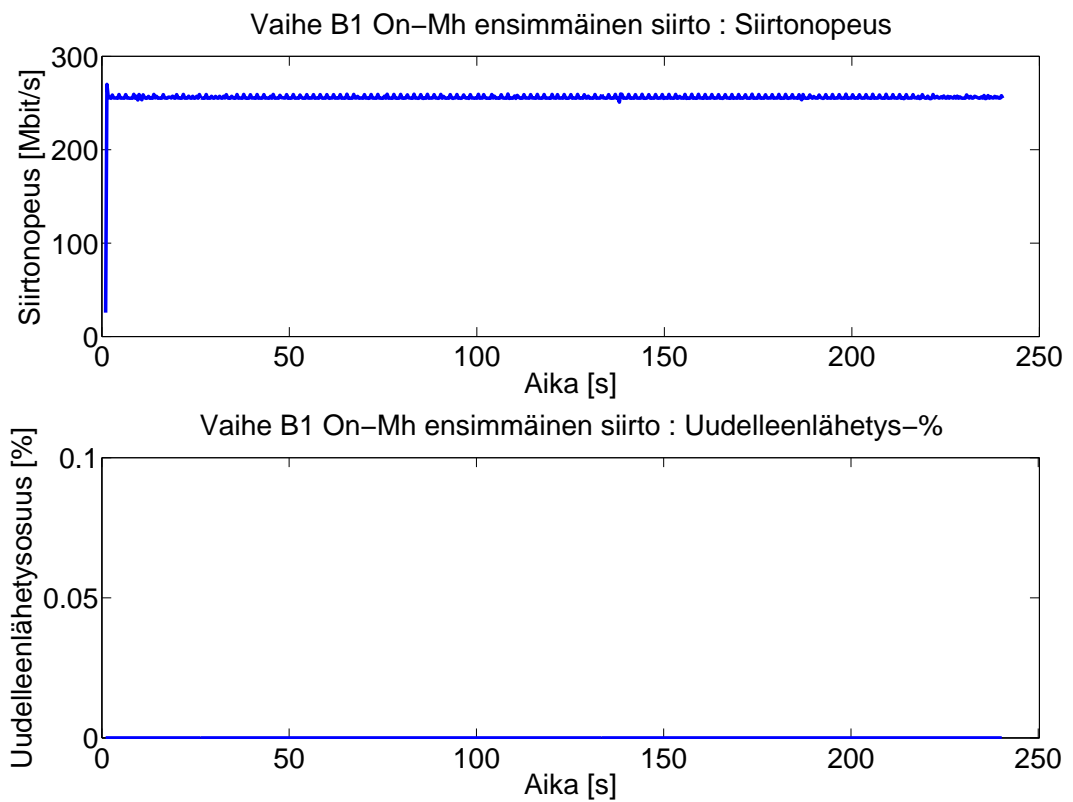
Kuva A.1. Tsunami-siirtojen tilastot siirroille JIVE:stä Jodrell Bankiin 1 Gbit/s kuidun yli. 6.4 tai 25.6GB siirrettiin muistista muistiin eri tavoitenopeusasetuksilla. Yhteyden laatu huononee yli 850 Mbit/s nopeuksilla (MoW06).



Kuva A.2. Ylimmässä kuvassa on Metsähovin sisääntuleva liikenne (paketteja sekunnissa, skaala 0–100000) koepäivänä 20/10/2006. Liikenteessä näkyvät aamun kolme siirtoesikoetta ja varsinaiset 256 Mbit/s ja 512 Mbit/s siirtokokeet. Aika on UT+3. Alimassa kuvassa on FUNET:in Suomeen saapuva liikenne (ylempi käyrä piikeillä) sekä lähtevä liikenne, 0–2.5 Gbit/s skaalalla. Ajat ovat UT+2. Kuvat on saatu CSC:ltä (CSC; CWK06).

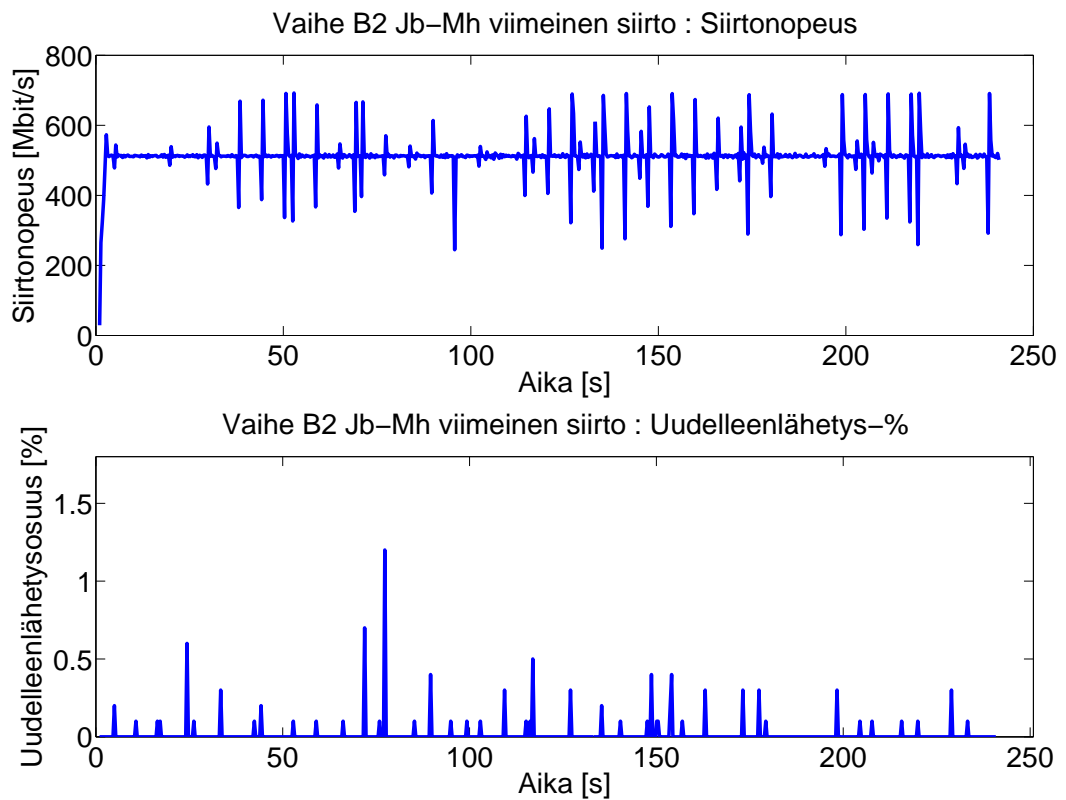


Kuva A.3. Siirtokokeen B1-osan ensimmäisen 256 Mbit/s skannin siirtonopeus- ja uudelleenlähetyks kuvaajat Jodrell Bankin siirrolle Metsähoviin. Kuvaajat on luotu Tsunami-asiakaslokien tilastoista. Neljäminuuttisen skannin reaaliaikainen siirto sujuu lähes täysin tasaisesti.

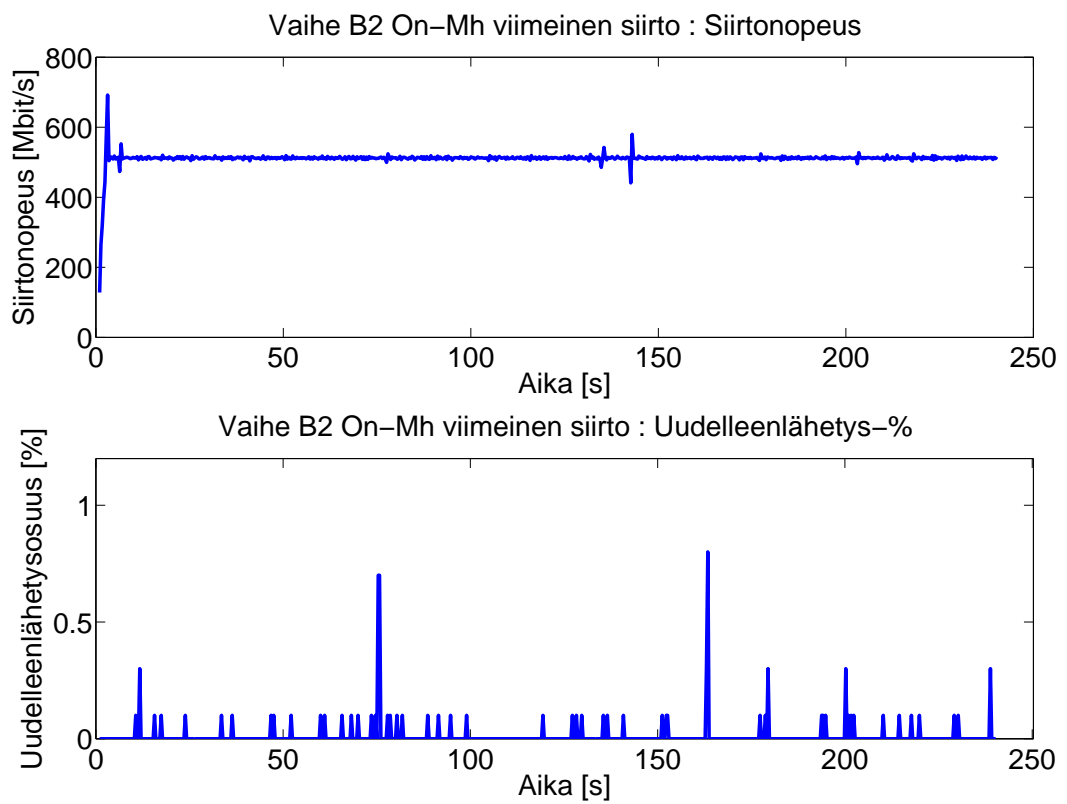


Kuva A.4. Siirtokokeen B1-osan ensimmäisen 256 Mbit/s skannin siirtonopeus- ja uudelleenlähetyksokuvaajat Onsalan reaaliaikasiirroille Metsähoviin. Kuvaajat on luotu Tsunami-asiakaslokien tilastoista. Neljäminuuttisen skannin reaaliaikainen siirto sujuu lähes täysin tasaisesti.

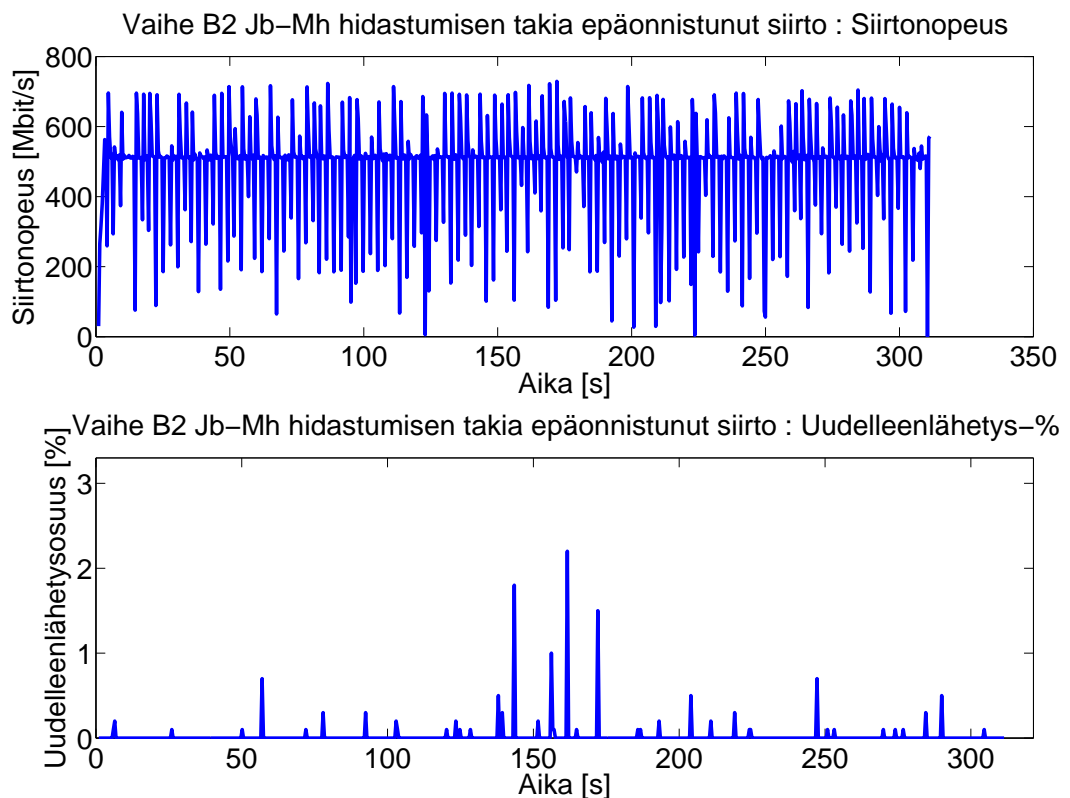




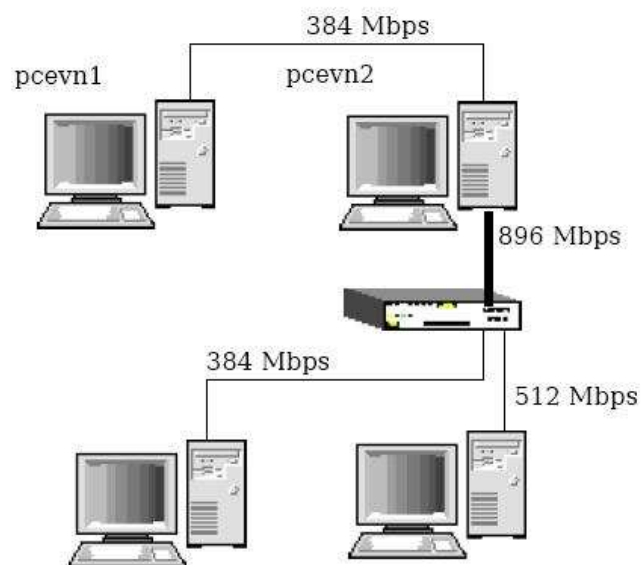
Kuva A.5. Siirtokokeen B2-osan viimeisen 512 Mbit/s skannin siirtonopeus- ja uudelleenlähetyskuvaajat Jodrell Bankin siirrolle Metsähoviin. Kuvaajat on luotu Tsunami-asiakaslokien tilastoista. Neljäminuuttisen skannin reaaliaikasiirto sujuu hyvin, mutta joillakin nopeusvaihteluilla.



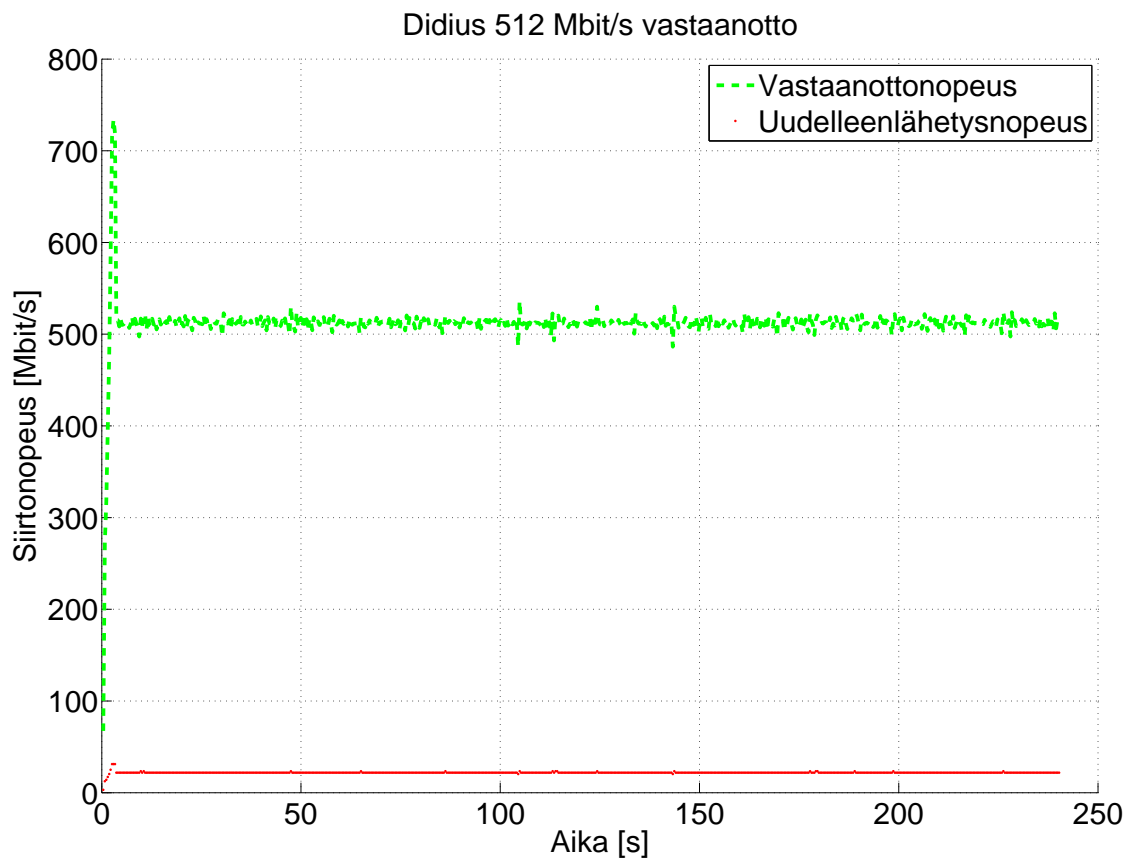
Kuva A.6. Siirtokokeen B2-osan viimeisen 512 Mbit/s skannin siirtonopeus- ja uudelleenlähetytkuvaajat Onsalan siirrolle Metsähoviin. Kuvaajat on luotu Tsunami-asiakaslokien tilastoista. Neljäminuuttisen skannin reaaliaikasiirto sujuu varsin tasaisesti.



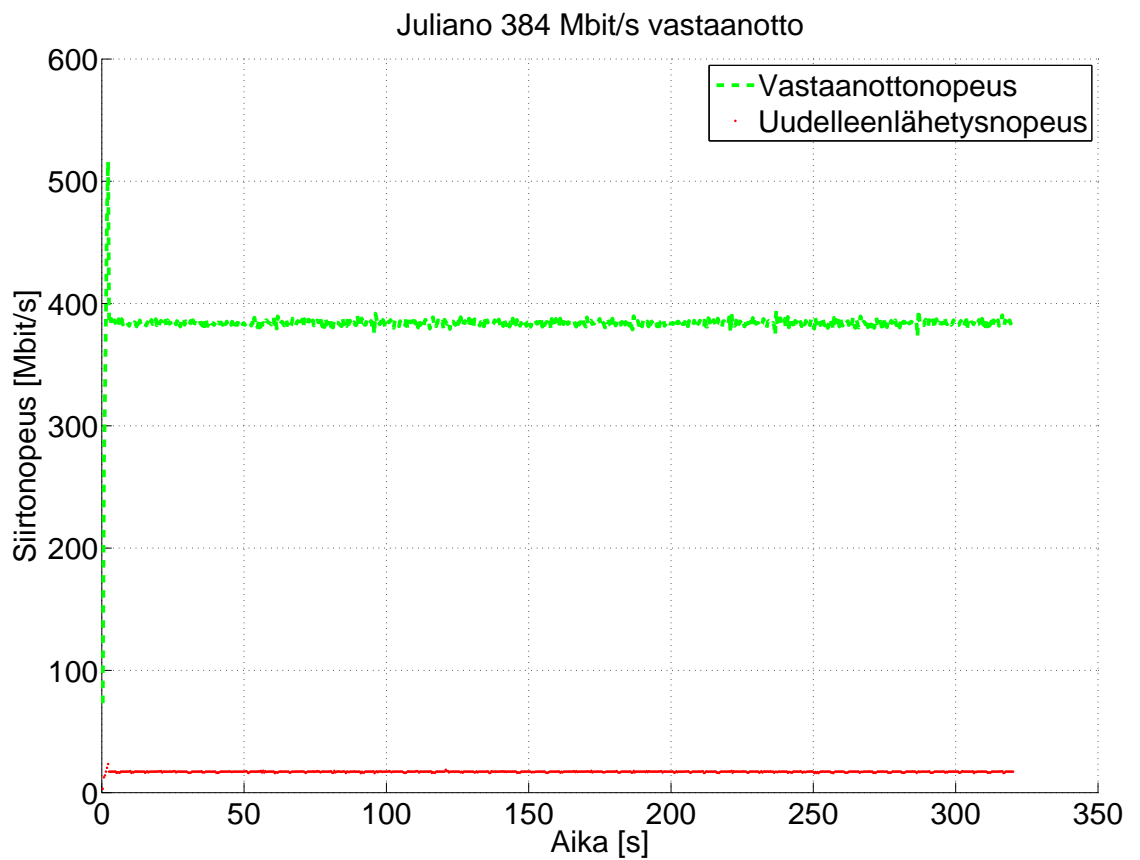
Kuva A.7. Siirtokokeen B2-osan kolmannen skannin 512 Mbit/s skannin siirtonopeus- ja uudelleenlähetyksuvaajat Jodrell Bankin (alla) siirrolle Metsähoviin. Kuvaajat on luotu Tsunami-asiakaslokien tilastoista. Siirtonopeudessa on suurta vaihtelua johtuen lähetyksessä käytetystä virheellisestä VSIB-ajurin puskurikokoasetuksesta. Ongelma voi ilmetä suurilla siirtonopeuksilla.



Kuva A.8. Kahden 384 ja 512 Mbit/s UDP-pakettivirran aggregointi Jodrell Bankin 1 Gbit/s kuidulle väliin sijoitetun palomuurikoneen avulla. Palomuri- tai siltaus-PC (pcevn2) yhdistää oman 512 Mbit/s liikenteen toisesta verkkoliitännästä tulevan 384 Mbit/s liikenteen (pcevn1) kanssa kytkimen yhteen 1 Gbit/s porttiin.



Kuva A.9. Siirtotilastot 896 Mbit/s muistista levyille siirron ensimmäiselle asiakaskoneelle Metsähovissa (Didius, 16GB 512 Mbit/s nopeudella). Kone toimii samalla NAT-palomuurina. Siirron palvelinkone sijaitsee Jodrell Bankissa 1 Gbit/s kuituyhteyden takana.



Kuva A.10. Siirtotilastot 896 Mbit/s muistista levyllä siirron toiselle asiakaskoneelle Metsähovissa (Juliano, 12GB 384 Mbit/s nopeudella). Tämän siirron palvelinkone sijaitsee Jodrell Bankissa saman 1 Gbit/s kuituyhteyden takana kuin 512 Mbit/s siirron palvelin.

## Viiteluettelo

- [1] EU6, "A Production Astronomy e-VLBI Infrastructure, Annex I - Description of Work", EU Sixth Framework Programme, Integrated Infrastructure Initiative, Contract 026642 EXPReS, <http://www.expres-eu.org/DOCS/dowannexi.pdf>, 02/2007
- [bbFTP] G. Farrache, bbFTP tiedostonsiirto-ohjelma, <http://doc.in2p3.fr/bbftp>, 10/2006
- [CSC] Tieteen tietotekniikan keskus CSC, Tietoa Funet-verkon tilasta, <http://www.csc.fi/funet/status/>, 10/2006
- [DTW04] R. Dodson, S. Tingay, C. West, C. Phillips, A. Tzioumis, J. Ritakari, F. Briggs, "The Australian experience with the PC-EVN recorder", Proceedings of the 7th European VLBI Network Symposium, sivut 253-256, 2004
- [EGAE] MIT Haystack Observatory, "Experiment Guided Adaptive Endpoint", <http://evlbi.haystack.mit.edu/twiki/bin/view/EVLBI/EGAE>, 10/2006
- [HMW07] R. Haas (OSO), A. Müskens (MPIfR), J. Wagner (MRO), C. Dulfer (MPIfR), A. Mujunen, J. Ritakari (MRO), A. Bertarini, (MPIfR), "e-VLBI Data Transfer from Onsala and Metsähovi to the Bonn Correlator", Proceedings of the 18th Working Meeting on European VLBI for Geodesy and Astrometry, 2007 (in print)
- [HHK03] Teruyuki Hasegawa, Toru Hasegawa, N. Kawaguchi ja muut, "2.488 Gbps Optical-connected Realtime VLBI Observation Testbed - A Multi-gigabit-rate Distributed Application in Astronomy", Global Telecommunications Conference 2003, Vol. 7, sivut 3752-3757, 2003

- [HKR04] E. Himwich, Y. Koyama, C. Reynolds, A. Whitney, "e-VLBI File-Naming Conventions", 2004, [http://haystack.mit.edu/tech/vlbi/evlbi/evlbi\\_memos/filenaming\\_conventions.pdf](http://haystack.mit.edu/tech/vlbi/evlbi/evlbi_memos/filenaming_conventions.pdf), 02/2007
- [Mei02] M. R. Meiss, "Tsunami: A High-Speed Rate-Controlled Protocol for File Transfer", Indiana University, 2002, <http://steinbeck.ucs.indiana.edu/~mmeiss/papers/tsunami.pdf>, 02/2007
- [MPI07] Max-Planck-Institut für Radioastronomie, Bonn, "Geodetic VLBI at the Bonn Correlator", <http://www.mpifr-bonn.mpg.de/div/vlbicor/geodesy/evlbi/index.html>, 06/2007
- [Mol06] G. Molera, "Computer tests in Metsähovi computers", 2006, <http://kurp.hut.fi/~gofrito/vsib-tests/tests.html>, 04/2007
- [MoW06] G. Molera, J. Wagner, "Tsunami data transfer tests", 2006, <http://kurp.hut.fi/~jwagner/tsunamiTests/>, 06/2006
- [RiM06] J. Ritakari, A. Mujunen, "The first 10 Gbps customer link of Funet installed", CSCnews 3, sivut 20-23, 2006
- [MuR06] A. Mujunen, J. Ritakari, Metsähovi Radio Observatory, Helsinki University of Technology TKK, "EXPreS JRA1 FABRIC: Data Acquisition Requirements", EU FABRIC Reports 2006, [http://www.jive.nl/dokuwiki/lib/exe/fetch.php/expres:management:data\\_acquisition\\_design.pdf](http://www.jive.nl/dokuwiki/lib/exe/fetch.php/expres:management:data_acquisition_design.pdf), 12/2006
- [MRW07] G. Molera, A. Mujunen, J. Ritakari, J. Wagner, Metsähovi Radio Observatory, Helsinki University of Technology TKK, B. Anderson, Jodrell Bank Observatory, University of Manchester, "EXPreS JRA1 FABRIC Data Acquisition Design Report", EU FABRIC Reports 2007, [http://www.jive.nl/dokuwiki/lib/exe/fetch.php/expres:management:data\\_acquisition\\_design.pdf](http://www.jive.nl/dokuwiki/lib/exe/fetch.php/expres:management:data_acquisition_design.pdf), 12/2006
- [Phi04] C. Phillips, "eVLBI scientific benefits", LBA eVLBI Memos, ATNF CSIRO, <http://www.atnf.csiro.au/vlbi/evlbi/memos.html>, 12/2006
- [Phi06] C. Phillips, "LBA eVLBI Download Page", ohjelmistojulkaisu, <http://www.>



[atnf.csiro.au/vlbi/evlbi/software/download.html](http://atnf.csiro.au/vlbi/evlbi/software/download.html), 12/2006

- [Pre02] E. Preuss, "The beginnings of VLBI at the 100-m radio telescope", Proceedings of the 6th European VLBI Network Symposium, Bonn, Saksa, kesäkuu 2002
- [RiM04] J. Ritakari, A. Mujunen, "Gbit/s VLBI and eVLBI with Off-The-Shelf Components", International VLBI Symposium 2004 General Meeting Proceedings, sivut 182-185, 2004
- [TKN06] H. Takeuchi, M. Kimura, J.-I. Nakajima ja muut, "Development of a 4 Gbps Multifunctional Very Long Baseline Interferometry Data Acquisition System", Publications of the Astronomical Society of the Pacific, sivut 1739-1748, joulukuun 2006
- [Tho91] A. R. Thompson, "Interferometry and Synthesis in Radioastronomy", ISBN-13: 978-0894648595, Krieger Publishing, 1991
- [CWK06] J. Conway, J. Wagner, P. Burgess, N. Kruithof ja muut, "EXPREs FABRIC Month 7 Demonstration Report", EU FABRIC Reports 2006, [http://www.jive.nl/dokuwiki/lib/exe/fetch.php/fabric:fabric1\\_2\\_3\\_month7jec3.pdf](http://www.jive.nl/dokuwiki/lib/exe/fetch.php/fabric:fabric1_2_3_month7jec3.pdf), 12/2006
- [Wag06-1] J. Wagner, "Tsunami UDP Protocol", SourceForge-ohjelmistoprojekti, <http://tsunami-udp.sourceforge.net>, 10/2006
- [Wag06-2] J. Wagner, "Quick Install Guide And Cheatsheet For Setting Up A VSIB PC", <http://www.metsahovi.fi/vlbi/instr/mro-vsib-pc>, 11/2006
- [Wel96] M. Welch, Big physical area kernel patch, <http://www.feise.com/~jfeise/Downloads/zr36120/>, 10/2006
- [Whi99] A. Whitney, "Concept for an Affordable High-Data-Rate VLBI Recording and Playback System", Mark 5 Memo #1, MIT Haystack Observatory, Mark 5 Memo Series, 1999
- [Whi00] A. R. Whitney ja muut, "VLBI Standard Hardware Interface Specification Rev 1.0", MIT Haystack Observatory, 2000, [http://www.haystack.mit.edu/tech/vlbi/vsi/docs/2000\\_08\\_07\\_vsi-h\\_final\\_rev\\_1.pdf](http://www.haystack.mit.edu/tech/vlbi/vsi/docs/2000_08_07_vsi-h_final_rev_1.pdf), 10/2006

- [Whi02] A. R. Whitney, "Mark 5 Disk-Based Gbps VLBI Data System", MIT Haystack Observatory, 2002, [http://www.haystack.mit.edu/tech/vlbi/mark5/docs/mark5\\_overview.pdf](http://www.haystack.mit.edu/tech/vlbi/mark5/docs/mark5_overview.pdf), 02/2007
- [Whi90] A. R. Whitney ja muut, Mark4 Memo Series, <http://www.haystack.edu/geo/mark4/memos.html>
- [FTP] Wikipedia, "FTP", <http://fi.wikipedia.org/wiki/FTP>, 10/2006