

TEKNILLINEN KORKEAKOULU  
Sähkö- ja tietoliikennetekniikan osasto  
Tietoverkkolaboratorio

Antti Gröhn

## **TESTIVERKON KELLOSYNKRONOINTI**

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi diplomi-insinöörin tutkintoa varten Espoossa 22.10.2004

Työn valvoja

Professori Raimo Kantola

Työn ohjaaja

Tekniikan lisensiaatti Markus Peuhkuri

<b>Tekijä:</b>	Antti Gröhn
<b>Työn aihe:</b>	Testiverkon kellosynkronointi
<b>Päivämäärä:</b>	22.10.2004
<b>Sivumäärä:</b>	113
<b>Osasto:</b>	Sähkö- ja tietoliikennetekniikka
<b>Professuuri:</b>	S-38 Teletekniikka
<b>Työn valvoja:</b>	Professori Raimo Kantola
<b>Työn ohjaaja:</b>	Tkl Markus Peuhkuri
<b>Yhteenveto:</b>	<p>Vaatimukset tietoverkon laitteiden välisen synkronoinnin tarkkuudelle kasvavat suoraan suhteessa verkon siirtonopeuteen. Varsinainen tiedonsiirron ohella myös verkon suorituskyvyn mittaaminen edellyttää tarkkaa synkronointia. Tiedonsiirron kannalta voi olla riittävää, että laitteissa käytettävän kellosignaalin vaihe on identtinen ja tarkasti toistettavissa jokaisessa verkon laitteessa.</p> <p>Verkon suorituskyvyn mittaaminen puolestaan edellyttää usein, että myös laitteissa ylläpidettävä absoluuttinen aika on yhtenevä vähintäänkin mittauksista suorittavien verkkokomponenttien osalta.</p> <p>Tämän työn tavoitteeksi asetettiin tietoverkkolaboratorion tiloissa sijaitsevan testiverkon kellojen synkronointi mikrosekunnin tarkkuudella. Tavoite edellyttää tarvittavan synkronointijärjestelmän suunnittelua ja rakentamista sekä luotettavien mittausmenetelmien kehittämistä saavutetun synkronoinnin tarkkuuden toteamiseksi.</p> <p>Asetettu tavoite osoittautui kuitenkin liian tiukaksi peruskokoonpanon omaaville PC-tietokoneille. Myös koneiden välisen aikapoikkeaman mittaaminen osoittautui haastavaksi ongelmaksi, koska ajan ylläpito suoritetaan periaatteessa kokonaan käyttöjärjestelmän toimenpitein. Poikkeaman yläraja on kuitenkin estimoitavissa tässä työssä esitettyjen menetelmien avulla. Työn lopussa esitetään mahdollisia ratkaisuja saavutetun poikkeaman pienentämiseksi.</p>
<b>Avainsanat:</b>	ajan ylläpito ja jakaminen, tietokoneiden kellojen synkronointi, GPS-järjestelmä, Network Time Protocol

<b>Author:</b>	Antti Gröhn
<b>Subject:</b>	Clock Synchronization of Computer Test Network
<b>Date:</b>	22.10.2004
<b>Number of pages:</b>	113

<b>Department:</b>	Department of Electrical and Communications
<b>Professorship:</b>	S-38 Telecommunications Technology

<b>Supervisor:</b>	Professor Raimo Kantola
<b>Instructor:</b>	Lic. (Tech.) Markus Peuhkuri

**Summary:**

Demands of clock synchronization accuracy increase directly proportional to network capacity. In addition to actual data transmission also precise network performance measurements are conditional on accurate synchronization. From data transmissions point of view it may be sufficient that the phase of clock signal is identical in every network component. Performance measurements may also require synchronization of absolute time in every network element participating in the actual measurement process.

The goal of this thesis is to synchronize clocks of a test network well within one micro second bound. This requires planning and construction of a new synchronization system. Also some methods for measuring the achieved accuracy have to be considered.

During this thesis it became apparent that this goal won't be fully reached using off-the-self PC hardware. Also the measurement of relative time difference between computers appeared to be a very challenging task. This is because the time management is done in the operating system. The upper bound of time difference can be estimated with methods described in this thesis. Also some future work is suggested in order to increase the achieved accuracy.

<b>Keywords:</b>	Time Management and Distribution, Computer Clock Synchronization, GPS - Global Positioning System, Network Time Protocol
------------------	--

## Alkulause

Tämä työ liittyy tietoverkkolaboratorion yhdessä yhteiskumppaniensa kanssa toteuttamaan IROnet-projektiin, joka keskittyy älykkäiden palvelun laatua tukevien reitittimien mallinnukseen ja kehittämiseen.

Kiitän työn valvojaa professori Raimo Kantolaa opastuksesta ja työn eri vaiheissa käydyistä idearikkaista keskusteluista. Työn ohjaajalle Tkl Markus Peuhkurille kuuluu suuri kiitos aktiivisesta otteesta työn ohjauksessa sekä lukuisista ennakkoluulottomista ideoista työn keskeisiä ongelmia ratkaistaessa.

Lämmin kiitos kuuluu Tietoliikennelaboratorion Seppo Saastamoiselle opastuksesta ja ohjauksesta elektroniikkasuunnittelussa sekä tämän työn ulkoasun muodostamisessa. Esitän kiitoksen myös Radiolaboratorion Lauri Laaksolle signaalijakajan mekaaniseen työstöön liittyvien menetelmien opetuksesta sekä työstössä tarvittavien työkalujen lainaamisesta.

Lopuksi haluan kiittää aamujeni aurinkoa Heliä, joka on jaksanut läpi lyhyehkön mutta sitäkin intensiivisemmän opiskelujakson. Suuri kiitos kuuluu myös vanhemmilleni taloudellisesta ja henkisestä tuesta opiskelun aikana.

Espoossa, 22.10.2004

Antti Gröhn

# Sisällysluettelo

<b>ALKULAUSE</b> .....	<b>IV</b>
<b>SISÄLLYSLUETTELO</b> .....	<b>V</b>
<b>LYHENNELUETTELO</b> .....	<b>VII</b>
<b>KÄSITELUETTELO</b> .....	<b>X</b>
<b>1. JOHDANTO</b> .....	<b>1</b>
1.1. TYÖN TAUSTA.....	1
1.2. TYÖN TAVOITE.....	2
1.3. TYÖN RAKENNE.....	3
<b>2. AJAN YLLÄPITO JA SYNKRONOINTI</b> .....	<b>4</b>
2.1. AJAN MÄÄRITTELY.....	4
2.1.1. <i>SI-järjestelmä ja sekunnin määritelmä</i> .....	4
2.1.2. <i>Ajan ylläpito</i> .....	5
2.2. AIKAJÄRJESTELMÄT.....	6
2.2.1. <i>TAI-aika</i> .....	6
2.2.2. <i>UT1-aika</i> .....	7
2.2.3. <i>UTC-aika</i> .....	7
2.2.4. <i>GPS-aika</i> .....	8
2.3. AJAN YLLÄPITO ELEKTRONISESSA JÄRJESTELMÄSSÄ.....	8
2.3.1. <i>Synkronoinnin tarve ja toteutus</i> .....	9
2.4. KELLON VIRHEET JA HYVYYDEN MÄÄRITTELY.....	10
2.4.1. <i>Taajuusvirhe</i> .....	10
2.4.2. <i>Vaihevirhe ja absoluuttinen poikkeama</i> .....	11
2.4.3. <i>Kellosignaalin vääristymät</i> .....	12
2.4.4. <i>Kellon hyvyyden määrittely ja tunnusluvut</i> .....	13
2.4.4.1. <i>Tarkkuus</i> .....	14
2.4.4.2. <i>Stabiilius</i> .....	14
2.4.4.3. <i>Aika-alueen tunnusluvut</i> .....	14
2.4.4.4. <i>Taajuusalueen tunnusluvut</i> .....	18
2.5. AIKA- JA TAAJUUSREFERENSSEJÄ.....	20
2.5.1. <i>Kideoskillaatit</i> .....	20
2.5.2. <i>Atomikellot</i> .....	21
2.5.3. <i>GPS-järjestelmä</i> .....	23
2.5.4. <i>Taajuusstandardiasemat</i> .....	23
<b>3. GPS – MAAILMANLAAJUINEN SATEELLIITTINAVIGOINTIJÄRJESTELMÄ....</b>	<b>25</b>
3.1. GLONASS.....	25
3.2. GPS-JÄRJESTELMÄN TOIMINTA.....	26
3.3. JÄRJESTELMÄN RAKENNE.....	28
3.3.1. <i>Avaruuslohko</i> .....	29
3.3.2. <i>Valvontalohko</i> .....	31
3.3.3. <i>Käyttäjälohko</i> .....	31
3.3.3.1. <i>Normaali navigointijärjestelmä</i> .....	32
3.3.3.2. <i>Tarkkuusnavigointijärjestelmä</i> .....	32
3.4. INFORMAATION VÄLITYS.....	33
3.4.1. <i>Radioyhteys</i> .....	33
3.4.2. <i>Näennäissatunnaiset koodit</i> .....	34
3.4.3. <i>Navigointiviestin rakenne</i> .....	35
3.4.4. <i>Ajan ylläpito ja välitys</i> .....	36
3.5. DIFFERENTIAALINEN GPS-PAIKANNUS.....	36
3.6. VIRHETEKIJÄT JA TARKKUUS.....	37

<b>4.</b>	<b>AJAN HALLINTA JA AJOITUS PC-TIETOKONEESSA.....</b>	<b>40</b>
4.1.	AJAN YLLÄPIDON TOTEUTUS.....	40
4.1.1.	Piiritason yleiskuvaus.....	40
4.1.1.1.	Ajan ja ajastuksen toteutus piiritasolla.....	42
4.1.2.	Käyttöjärjestelmän toimenpiteet ajan ylläpidossa.....	44
4.1.2.1.	Systeemikello.....	46
4.2.	NTP - NETWORK TIME PROTOCOL.....	47
4.2.1.	Synkronoinnin periaate.....	48
4.2.2.	Systeemikellon korjaaminen.....	51
4.2.3.	PPS-signaalin prosessointi.....	52
<b>5.</b>	<b>SYNKRONOINTIJÄRJESTELMÄ.....</b>	<b>54</b>
5.1.	SYNKRONOINTIJÄRJESTELMÄN RAKENNE.....	54
5.2.	PPS-SIGNAALIJAKAJA.....	56
5.2.1.	Signaalijakajan mekaaninen rakenne.....	57
5.2.2.	Emolevyn toimintaperiaate.....	58
5.2.3.	Erotettujen käyttöjännitteiden tuottaminen.....	60
5.2.4.	Linjakortin toimintaperiaate.....	61
5.2.5.	Signaalijakajan ja synkronointiketjun aiheuttama virhe.....	63
5.3.	PPS-SIGNAALIN PROSESSOINNIN EDELLYTYKSET.....	64
5.3.1.	FreeBSD 5.21.....	65
5.3.2.	Linux 2.4.20.....	68
5.4.	TIETOKONEEN SYNKRONOINNIN TARKKUUS.....	70
5.4.1.	Systeemikellon värinä ja satunnainen vaihtelu.....	71
5.4.1.1.	Systeemikutsun suoritusviive.....	71
5.4.1.2.	Synkronoinnin suhteellinen ja absoluuttinen tarkkuus.....	72
5.4.2.	Systeemikellon taajuuspoikkeama.....	76
5.4.2.1.	Lämpötilan vaikutus synkronointitarkkuuteen.....	76
5.4.2.2.	Kokoonpanon vaikutus synkronointitarkkuuteen.....	78
5.5.	TIETOKONEIDEN VÄLISEN SYNKRONOINNIN TARKKUUS.....	79
5.5.1.	Suhteellisen poikkeaman mittausmenetelmä.....	79
5.5.2.	Suhteellisen poikkeaman yläraja.....	81
5.5.3.	Suhteellisen poikkeaman arviointi PING-kyselyn avulla.....	86
<b>6.</b>	<b>JOHTOPÄÄTÖKSET.....</b>	<b>87</b>
6.1.	TULOSTEN ARVIOINTI.....	87
6.2.	MAHDOLLISIA SOVELLUSKOHTEITA.....	88
6.3.	TYÖN JATKOKEHITYS.....	89
	<b>LÄHDELUETTELO.....</b>	<b>92</b>
	<b>LIITTEET.....</b>	<b>94</b>
	Liite A: Emolevyn periaatepiirros.....	94
	Liite B: Emolevyn komponenttien sijoittelu ja tarvittavat johdinvedot.....	97
	Liite C: Linjakortin periaatepiirros.....	98
	Liite D: Linjakortin komponenttien sijoittelu ja tarvittavat johdinvedot.....	99
	Liite E: Rakennettujen laitteiden ulostuloporttien välinen ajastus.....	100
	Liite F: Systeemikutsun vaatima suoritus aika.....	101
	Liite G: Systeemikellon aikapoikkeamaan vaikuttavia tekijöitä.....	103
	Liite H: Lämpötilan vaikutus synkronointitarkkuuteen.....	104
	Liite I: Kokoonpanon vaikutus synkronointitarkkuuteen.....	106
	Liite J: Kaiutusviive ja PPS-signaalin aikaleima.....	107
	Liite K: Kaiutusviive ja PPS-signaalin aikaleima kuormitetussa verkossa.....	109
	Liite L: Poikkeama suhteessa UTC-aikaan.....	110
	Liite M: Suhteellisen poikkeaman yläraja.....	111
	Liite N: Suhteellisen poikkeaman jakauma ja kertymäfunktio.....	112
	Liite O: Suhteellisen poikkeaman estimointi PING-kyselyn avulla.....	113

## Lyhenneluettelo

<b>ADEV</b>	Allan Deviation, Allan-keskihajonta
<b>AGP</b>	Accelerated Graphics Port
<b>API</b>	Application Programming Interface, Sovellusrajapinta
<b>ARP</b>	Address Resolution Protocol
<b>AS</b>	Anti-spoofing, GPS:n P-koodin salaaminen Y-koodiksi
<b>ASCII</b>	American Standard Code for Information Interchange
<b>ATA</b>	Advanced Technology Attachment
<b>AVAR</b>	Allan Variance, Allan-varianssi
<b>BCD</b>	Binary Coded Decimal
<b>BIOS</b>	Basic Input Output System
<b>BIPM</b>	Bureau of Weights and Measures, Kansainvälinen paino- ja mittatoimisto
<b>C/A-koodi</b>	Coarse/Acquisition code, Karkea GPS-koodi
<b>CDMA</b>	Code Division Multiple Access, Koodijakoinen monipääsymenetelmä
<b>CDROM</b>	Compact Disc Read Only Memory
<b>CGPM</b>	Conférence générale des poids et mesures, Yleinen paino- ja mittakonferenssi
<b>CMOS</b>	Complementary Metal-Oxide Semiconductor
<b>CPU</b>	Central Processing Unit, Prosessori
<b>DCD</b>	Data Carrier Detect
<b>DIMM</b>	Dual In-line Memory Module
<b>DoD</b>	Department of Defence, Yhdysvaltain puolustusministeriö
<b>ESD</b>	Electrostatic Discharge, Sähköstaattinen purkaus
<b>FDMA</b>	Frequency Division Multiple Access, Taajuusjakoinen monipääsymenetelmä
<b>FLL</b>	Frequency-Locked Loop, Taajuuslukittu silmukka
<b>FFT</b>	Fast Fourier Transform, Nopea Fourier-muunnos
<b>GM(S)T</b>	Greenwich Mean (Sideral) Time, Nollameridiaanin alkuperäinen keskitähti-aika
<b>GLONASS</b>	Global Navigation Satellite System, Venäläinen satelliittipaikannusjärjestelmä
<b>GPS</b>	Global Positioning System, Maailmanlaajuinen satelliittinavigointijärjestelmä
<b>HF</b>	High Frequency, Korkeat taajuudet
<b>HOW</b>	Handover Word, Navigointiviestin vaihtosana
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IC</b>	Integrated Circuit, Integroitu piiri

<b>IDE</b>	<b>Integrated Drive Electronics</b>
<b>IP</b>	<b>Internet Protocol</b>
<b>IRQ</b>	<b>Interrupt Request, Keskeytyspyyntö</b>
<b>ISA</b>	<b>Industry Standard Architecture</b>
<b>LORAN-C</b>	<b>Long-Range Navigation, Radiopaikannusjärjestelmä</b>
<b>LF</b>	<b>Low Frequency, Matalat taajuudet</b>
<b>MADEV</b>	<b>Modified Allan Deviation, Modifioitu Allan-keskihajonta</b>
<b>MSI</b>	<b>Micro-Star International</b>
<b>MTIE</b>	<b>Maximum Time Interval Error, Maksimaalinen aikavälivirhe</b>
<b>MVAR</b>	<b>Modified Allan Variance, Modifioitu Allan-varianssi</b>
<b>NAVSTAR</b>	<b>Navigation System with Time and Ranging, Navigointijärjestelmä ajan ja etäisyyden suhteen</b>
<b>NIST</b>	<b>National Institute of Standards and Technology, USA:n kansallinen standardien ja tekniikan instituutti</b>
<b>NMEA</b>	<b>National Marine Electronics Association</b>
<b>NNSS</b>	<b>Navy Navigation Satellite System, USA:n laivaston satelliittinavigointijärjestelmä</b>
<b>PC</b>	<b>Personal Computer</b>
<b>PCC</b>	<b>Processor Cycle Counter, Prosessorin kellojaksolaskuri</b>
<b>PCI</b>	<b>Peripheral Component Interconnect</b>
<b>PCM</b>	<b>Pulse Code Modulation, Pulssikoodimodulaatio</b>
<b>P-koodi</b>	<b>Precision code, Tarkka GPS-koodi</b>
<b>PLL</b>	<b>Phase-Locked Loop, Vaihelukittu silmukka</b>
<b>PPS</b>	<b>Precise Positioning System, Tarkkuusnavigointijärjestelmä</b>
<b>PPS-Signal</b>	<b>Pulse per Second Signal, Signaali joka sisältää yhden pulssin sekunnissa</b>
<b>PPM</b>	<b>Parts per Million, Taajuussuhteen miljoonasosa</b>
<b>PRN</b>	<b>Pseudorandom Noise, Näennäissatunnainen koodi</b>
<b>PSD</b>	<b>Power Spectral Density, Tehospektritiheys</b>
<b>PSK</b>	<b>Phase Shift Keying, Digitaalinen vaihemodulaatio</b>
<b>QoS</b>	<b>Quality of Service, Palvelun laatu</b>
<b>QPSK</b>	<b>Quadrature Phase shift Keying, Vaihesiirrettyihin kantaalto-komponentteihin perustuva digitaalinen vaihemodulaatio.</b>
<b>RTC</b>	<b>Real-Time Clock, Reaaliaikakello</b>
<b>RTT</b>	<b>Round Trip Time, Kiertoaikaviive</b>
<b>SA</b>	<b>Selective Availability, Tahallinen heikennys</b>
<b>SDH</b>	<b>Synchronous Digital Hierarchy</b>
<b>SDRAM</b>	<b>Synchronous Dynamic Random Access Memory</b>
<b>SI</b>	<b>Système International d'Unités, International system of units, Kansainvälinen mittayksikköjärjestelmä</b>



<b>SONET</b>	<b>Synchronous Optical NETwork</b>
<b>SPS</b>	<b>Standard Positioning System, Normaali navigointijärjestelmä</b>
<b>SVN</b>	<b>Space Vehicle launch Number, GPS-satelliitin laukaisunumero</b>
<b>TAI</b>	<b>International Atomic Time, Kansainvälinen atomiaika</b>
<b>TCXO</b>	<b>Temperature-Compensated Crystal Oscillator, Lämpötilakompensoitu kideoskillaatori</b>
<b>TDEV</b>	<b>Time Deviation, Aikakeskihajonta</b>
<b>TE</b>	<b>Time Error, Aikavirhe</b>
<b>TIE</b>	<b>Time Interval Error, Aikavälivirhe</b>
<b>TIE<sub>rms</sub></b>	<b>Root Mean Square of the Time Interval Error, Aikavälivirheen neliösumman neliöjuuri (tehollisarvo)</b>
<b>TLM</b>	<b>Telemetry Word, Navigointiviestin telemetriasana</b>
<b>TOW</b>	<b>Time of Week, GPS-viikon alusta lasketut sekunnit</b>
<b>TTL</b>	<b>Transistor-to-Transistor Logic, 0-5 V elektroniikkalogiikka</b>
<b>TVAR</b>	<b>Time Variance, Aikavarianssi</b>
<b>UART</b>	<b>Universal Asynchronous Receiver/Transmitter</b>
<b>UCXO</b>	<b>Oven-Controlled Crystal Oscillator, Uunitettu kideoskillaatori</b>
<b>UDP</b>	<b>Unicast Datagram Protocol</b>
<b>USB</b>	<b>Universal Serial Bus</b>
<b>USNO</b>	<b>U.S. Naval Observatory, USA:n laivaston observatorio</b>
<b>UTC</b>	<b>Universal Coordinated Time, Koordinoitu yleisaika</b>
<b>UT1</b>	<b>Universal Time 1, Yleisaika</b>
<b>VCO</b>	<b>Voltage-Controlled Oscillator, Jänniteohjattu oskillaatori</b>
<b>VCXO</b>	<b>Voltage-Controlled Crystal Oscillator, Jänniteohjattu kideoskillaatori</b>
<b>XO</b>	<b>Crystal Oscillator, Kideoskillaatori</b>
<b>Y-koodi</b>	<b>GPS:n P-koodin salattu versio</b>

## Käsiteluettelo

**Sideerinen vuorokausi** eli tähtivuorokausi on kevättasauspisteen kahden yläkulminaation välinen aika. Tällä tarkoitetaan aikaa, jonka kuluessa tähdet ovat Maasta katsoen taas samassa asennossa. Maan pyörimisliike hidastuu hiljalleen, minkä seurauksena tähtivuorokausi pitenee. Lisäksi pyörimisessä on pieniä epäsäännöllisyyksiä. [1]

**Synodinen vuorokausi** eli aurinkovuorokausi on aika, jonka kuluttua Aurinko näkyy taas samassa suunnassa. Se on hiukan (3 min 56,56 s) pidempi kuin tähtivuorokausi, koska Maa on edennyt radallaan ja sen on pyörähdettävä hiukan yli kierros, jotta Aurinko näkyisi taas samassa suunnassa. Vuoden aikana tähtivuorokausia kuluu yksi enemmän kuin aurinkovuorokausia. [1]

**Trooppinen vuosi** tarkoittaa auringon kiertoaikaa kevättasauspisteestä kevättasauspisteeseen. [1]

**Sideerinen vuosi** on kulunut, kun Aurinko on tehnyt täyden kierroksen tähtien suhteen. [1]

**Absoluuttinen GPS-paikannus** tarkoittaa satelliittien lähettämän signaalin mittaamista yhdellä vastaanottimella.

**Differentiaalinen GPS-paikannus** (DGPS) käyttää hyväkseen satelliittien lähettämän signaalin mittaamista kahdella vastaanottimella samanaikaisesti, joista toinen on paikallaan ja toinen liikkuu sen ympäristössä.

**BCD-koodauksessa** jokainen desimaaliluvun numero koodataan omalla neljän bitin mittaisella binääriluvulla. Tällöin desimaalilukua 57 vastaa BCD-luku 0101 0111.

# 1. Johdanto

## 1.1. Työn tausta

Ajan synkronointi eli yhtenäistäminen on tarpeen kaikilla yhteiskunnan osa-alueilla. Harva meistä tuskin voisi kuvitellakaan liikkuvansa julkisilla liikennevälineillä olematta vähintään minuuttien tarkkuudella varma linja-auton tai raitiovaunun oletetusta saapumisajankohdasta. Voimme siis helposti kuvitella millaisia hankaluuksia kellojen synkronoinnin puuttuminen aiheuttaisi pelkästään ihmisten jokapäiväiseen elämään.

Tiedonsiirtoverkkojen siirtonopeuden kasvun seurauksena ovat vaatimukset synkronoinnin tarkkuudelle kasvaneet sekä varsinaisen tiedonsiirron että tietoverkon suorituskyvyn mittaamisen mahdollistamiseksi. Tietoverkkolaboratoriossa sijaitsevaa testiverkkoa käytetään reititysprotokollien ja palvelunlaatuun (QoS, Quality of Service) liittyvien sovellusten suorituskyvyn mittaamiseen vaikeiden tai äärimmäisten olosuhteiden vallitessa tietoverkossa.

Tämä tarkoittaa mittausten tekemistä tietoverkossa, jossa välitettävän ja käsiteltävän liikenteen määrä ja siirtonopeus on suuri. Testiverkon synkronointi ei kuitenkaan ole riittävän tarkka, jotta saatujen mittaustulosten ja edelleen niiden pohjalta tehtyjen johtopäätösten oikeellisuudesta voitaisiin olla täysin varmoja. Näin ollen on olemassa motivaatio synkronoinnin tarkkuuden parantamiselle riittävälle tasolle, mikä mahdollistaisi edelleen luotettavien mittausten suorittamisen.

## 1.2. Työn tavoite

Tämän diplomityön tavoitteena on synkronoida noin kahdestakymmenestä testireititimestä ja -palvelimesta muodostuva testiverkko. Synkronoinnin tavoitteeksi on asetettu, että verkon sisältämien tietokoneiden kellot saavat poiketa toisistaan korkeintaan yhden mikrosekunnin.

Asetettu tavoite voidaan jakaa kahteen välitavoitteeseen. Ensimmäisessä vaiheessa on pystyttävä synkronoimaan yksittäinen tietokone siten, että sen kellon taajuus ja aika saadaan seuraamaan referenssikellon taajuutta ja aikaa mahdollisimman tarkasti. Asetetun tavoitteen saavuttamiseksi yksittäisen tietokoneen kellon virheen on oltava huomattavasti asetettua yhden mikrosekunnin tavoitetta pienempi.

Tämän jälkeen on pystyttävä synkronoimaan kaikki tietokoneet referenssikelloon siten, että kaikkien kellojen välinen suhteellinen poikkeama ei missään tilanteessa ylitä annettua raja-arvoa. Verkossa suoritettavien mittausten kannalta kellojen absoluuttisella poikkeamalla referenssikellosta ei kuitenkaan ole merkitystä, vaan sillä paljonko kellot poikkeavat suhteessa toisiinsa. Toisaalta kellojen saamisella mahdollisimman lähelle koordinoitua yleisaikaa (UTC, Universal Coordinated Time) olisi varmasti hyödyllistä lisäarvoa.

Absoluuttinen ja suhteellinen poikkeama ovat sidoksissa toisiinsa ja kellojen välisen absoluuttisen poikkeaman erisuuruus aiheuttaa suoraan suhteellista poikkeamaa kellojen välille. Suhteellisen poikkeaman mittaaminen asettaa kuitenkin mielenkiintoisen haasteen. Synkronointiin käytettävien pakettien ja signaalin siirrossa ja käsittelyssä on viivettä, jonka suuruus vaihtelee tietokoneen rakenteesta, käyttöjärjestelmästä ja tietokoneen prosessorin kuormasta riippuen.

Varsinaisessa kellon päivityksessä on siis viivettä, jota ei välttämättä tarkasti voida tietää tai mitata. Vastaavan viiveen voidaan olettaa esiintyvän myös kellonajan kysymisessä tietokoneilta. Lisäksi viive vaihtelee eri koneiden ja kysymisajanhetken perusteella. Kuinka siis voidaan luotettavasti todeta, paljonko koneiden kelloissa on suhteellista poikkeamaa? Vastaus tähän kysymykseen löytyy tästä työstä.

### 1.3. Työn rakenne

Työssä käsitellään asetetun tarkkuusvaatimuksen mahdollistavia menetelmiä sekä näiden olennaisia eroja. Lopulliseen toteutukseen valitun menetelmän rakennetta sekä sen vahvuuksia ja heikkouksia käsitellään niin koko järjestelmän kuin sen eri osa-alueiden kannalta. Koska lopullinen synkronointitarkkuus on summa siihen vaikuttavista osa-alueista, esitetään työssä kuhunkin osa-alueeseen liittyviä menetelmiä synkronoinnin tarkkuuden toteamiseksi.

Luvussa 2 käydään ensin läpi perusteita ajasta, ajan ylläpidosta, määritelmistä ja synkronoinnista. Luvussa 3 käsitellään lopullisen ratkaisun kannalta keskeisen GPS-järjestelmän rakennetta ja ominaisuuksia. Luvussa 4 esitetään kuinka aikaa hallitaan tietokoneessa niin elektroniikan kuin käyttöjärjestelmän avulla. Lisäksi esitetään kuinka synkronointi ja kellon aika-/taajuuspoikkeaman korjaaminen voidaan toteuttaa NTP-protokollan avulla.

Luvussa 5 käydään läpi varsinainen tutkimusongelma sekä sen asettamat haasteet. Lisäksi esitellään toteutettu synkronointijärjestelmä ja perehdytään sen suorituskykyyn ja sen toteamiseksi tarvittaviin mittausmenetelmiin. Lopuksi tuloksia, järjestelmän mahdollisia sovelluskohteita sekä jatkokehityksen tarvetta arvioidaan luvussa 6.

## 2. Ajan ylläpito ja synkronointi

### 2.1. Ajan määrittely

Aika voidaan määrittellä maapallon pyörimisen perusteella (aurinko- ja tähtiaika), kvanttimekaanisten ilmiöiden avulla (nykyinen SI-järjestelmän mukainen sekunti) tai taivaankappaleiden liikkeestä (efemeridi- ja dynaaminen aika). [1]

Tähtiajan perusmitta on tähtivuorokausi eli sideerinen vuorokausi, joka määritellään kevättasauspisteen kahden peräkkäisen yläkulminaation välisenä aikana. Tällä tarkoitetaan aikaa, jonka kuluessa tähdet ovat Maasta katsoen taas samassa asennossa. Tähtiaika on siis yhtä tasaista kuin maapallon pyöriminen. Pyörimisliike hidastuu kuitenkin vähitellen. Lisäksi yhteen kierrokseen kuluessa ajassa esiintyy millisekunnin suuruusluokkaa olevia vaihteluita. [1]

Näennäinen tähtiaika määräytyy todellisen kevättasauspisteen mukaan ja saadaan siten suoraan havainnoista. Kevättasauspisteessä on kuitenkin pientä vaihtelua. Keskimääräisellä tähtiajalla eli keskitähtiajalla tarkoitetaan aikaa, joka saataisiin ilman tätä vaihtelua. [2]

Synodinen vuorokausi eli aurinkovuorokausi on aika, jonka kuluttua Aurinko näkyy taas samassa suunnassa. Se on hiukan (3 min 56,56 s) pidempi kuin tähtivuorokausi, koska Maa on edennyt radallaan ja sen on pyörähdettävä hiukan yli kierros, jotta Aurinko näkyisi taas samassa suunnassa. Vuoden aikana tähtivuorokausia kuluu yksi enemmän kuin aurinkovuorokausia. [1]

#### 2.1.1. *SI-järjestelmä ja sekunnin määritelmä*

Kansainvälistä mittayksikköjärjestelmää (SI, *Système International d'Unités*) hallinnoi yleinen paino- ja mittakonferenssi (CGPM, *Conférence générale des poids et mesures*). SI-järjestelmä perustuu seuraaviin seitsemään perusyksikköön: metri (m), kilogramma (kg), sekunti (s), ampeeri (A), kelvin (K), mooli (mol) ja kandela (cd). [3]

Ajan perusyksikkö sekunti määritellään seuraavasti [3]:

*”Sekunti on 9 192 631 770 kertaa sellaisen jakson aika, joka vastaa cesium 133-atomin siirtymää perustilan ylihienorakenteen kahden energiatason välillä.”*

Vastaava cesium 133-atomin spektriviivan taajuus on siis 9,192 631 770 GHz. Aika realisoidaan cesium-atomikellolla, jonka taajuuden suhteellinen epävarmuus on  $3 \cdot 10^{-13}$  vastaten 1 sekunnin virhettä 100 000 vuodessa. [3]

### 2.1.2 Ajan ylläpito

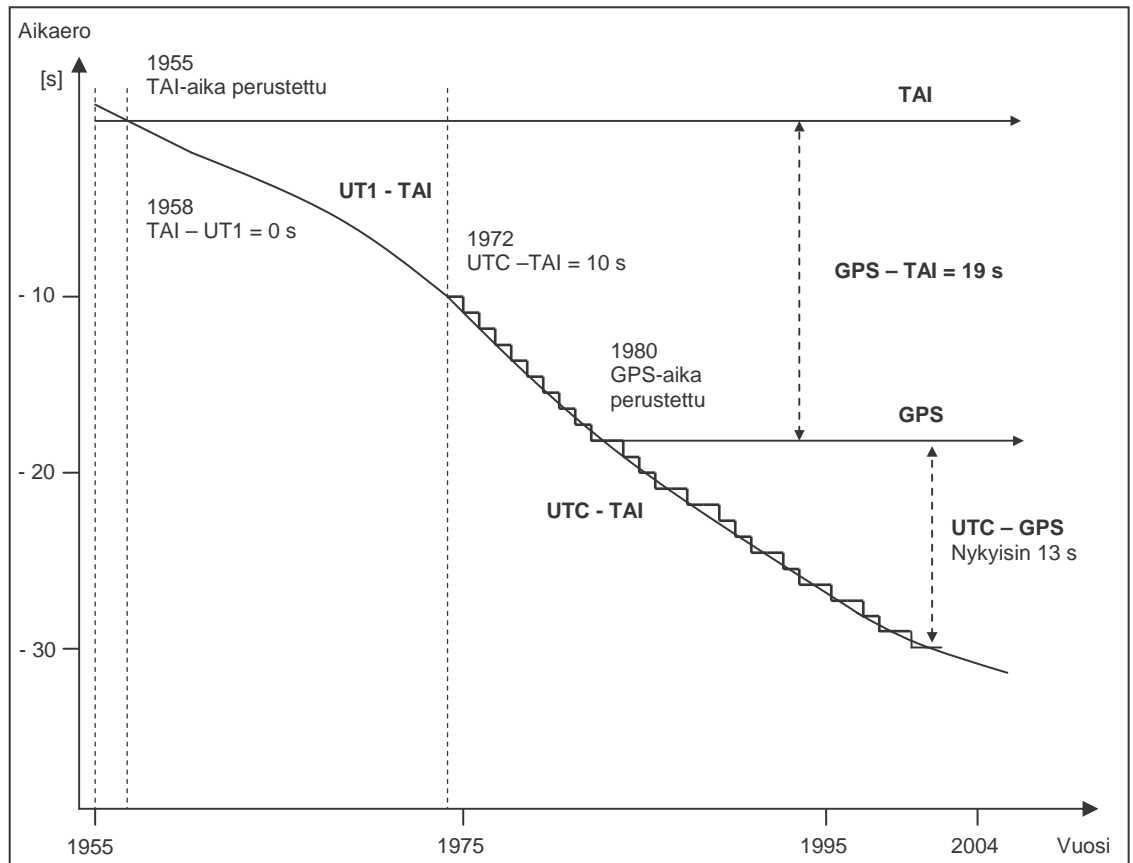
Useat laboratoriot ympäri maailmaa ylläpitävät toisistaan riippumatonta atomiaikaa. Laboratoriot työskentelevät yhteistyössä Kansainvälinen paino- ja mittatoimiston (BIPM, Bureau International des Poids et Mesures) kanssa, joka vastaa eri lähteiden atomiaikojen yhdistämisestä ja kansainvälisen atomiajan (TAI, International Atomic Time) määrittämisestä. [2]

Suomen virallisen ja cesium-taajuusstandardiin perustuvan ajan ylläpidosta vastaa Mittatekniikan keskus, MIKES. Suomen aikaa ja taajuutta verrataan jatkuvasti GPS:n avulla suoraan USNO:n pääkelloon ja LORAN-C avulla BIPM:n sekä Venäjän aikaan. Lisäksi on rakennettu GPS Common View -mittausjärjestelmä BIPM-aikavertailuun. MIKESin ylläpitämän Suomen ajan epävarmuus (UTC) on nykyisin n. 200 ns. Taajuuden lyhytaikainen (10 min) suhteellinen epävarmuus on  $10^{-12}$  ja vuorokaudessa  $10^{-13}$ . [3]

GPS-järjestelmä (Global Positioning System) muodostaa hyvän maailmanlaajuisen tavoitettavuutensa ansionsa eri lähteiden ylläpitämän atomiajan vertailun perustan. Maailmanlaajuisen synkronoinnin tavoitteeksi on asetettu 100 nanosekuntia tai sitä parempi tarkkuus. BIPM laskee tarvittavat korjaukset eri lähteiden ylläpitämille atomiajoille, jotta ne voivat säätää kellonsa määritettyjen rajojen sisälle. [2]

## 2.2. Aikajärjestelmät

Erilaisien aikajärjestelmien määrä on suuri, mutta lähes poikkeuksetta kaikkien järjestelmien käytölle löytyy looginen selitys. Kuvassa 2.1 on esitetty periaate joidenkin aikajärjestelmien keskinäisistä suhteista.



Kuva 2.1. Periaatteellinen tarkastelu aikajärjestelmien välisistä suhteista. [4]

### 2.2.1. TAI-aika

Kansainvälisten laboratorioiden yhdessä ylläpitämä kansainvälinen atomiaika (TAI) muodostaa nykyisin kansainvälisen ajanlaskun perustan. TAI-ajan ylläpito aloitettiin vuonna 1955 ja vuonna 1958 TAI- ja UT1-ajat asetettiin lähelle toisiaan. Järjestelmän mittayksikkönä käytetään SI-järjestelmän mukaista sekuntia merenpinnan tasossa. [4]



### 2.2.2. UTC-aika

Kansainvälinen atomiaika (TAI) ei ole mitenkään sidoksissa maan pyörimisliikkeeseen vaan se on johdettu atomin energiatilojen muuttumisesta. Esimerkiksi satelliittipainannuksen kannalta on kuitenkin tärkeää sitoa aika maan pyörimisliikkeeseen. Tämä on välttämätöntä määritettäessä satelliittien maakeskeistä etäisyyttä, joka riippuu ajasta. [2]

Ongelman ratkaisemiseksi vuonna 1972 otettiin käyttöön UTC-aika eli koordinoitu yleisaika (ns. seinäkelloaika), joka sitoo maan liikkeen yhteen ylläpidetyn ajan kanssa [2]. UTC määritellään siten, että se käy samalla nopeudella kuin atomiaika TAI, mutta poikkeaa siitä kokonaisilla sekunneilla. Poikkeamaa muutetaan tarpeen mukaan siten, että UTC ei koskaan poikkeaa UT1:stä enempää kuin  $\pm 0,9$  sekuntia. [1]

UTC-ajan ja TAI-ajan eroksi asetettiin vuonna 1972 tarkasti 10 sekuntia, mikä oli myös sen hetkinen ero TAI- ja UT1-aikojen välillä. Samalla otettiin käyttöön karkaussekunti (leap second). Karkaussekunti tarkoittaa, että UTC-aikaa siirretään sekunnin verran UT1-aikaan päin aina kun UT1- ja UTC-ajan välinen ero on suurempi kuin  $\pm 0,9$  sekuntia [4].

### 2.2.3. UT1-aika

Yleisaika UT1 (Universal Time 1) on johdettu maapallon pyörimisliikkeestä ja se määritellään GMST-ajan (Greenwich Mean Sideral Time) pohjalta tietyin korjauksin [1]. GMT on nollameridiaanin alkuperäinen keskitähtiaika, jonka tieteellinen määrittely on kuitenkin melko väljä [4].

Vuodesta 1958 eteenpäin UT1-aika on systemaattisesti jäänyt jälkeen TAI-ajasta, koska pyörimisliikkeen perusteella määritetty vuorokauden pituus ei ole tasan 86400 sekuntia vaan hieman enemmän. Lisäksi vuorokausi pitenee keskimäärin 1,4 millisekuntia vuosisadassa maapallon pyörimisnopeuden hidastuessa. Maan pyörimisnopeudessa on myös pientä kausittaista vaihtelua. [5]

#### 2.2.4. GPS-aika

GPS-aika on GPS-satelliittinavigointijärjestelmän perusaika, joka käy samaa tahtia kuin TAI-aika [4]. GPS-ajan nolлахetki sijoittuu keskiyöhön (00:00 UTC) tammikuun 6. päivä 1980. Sitä ei kuitenkaan säädetä UT1-ajan mukana, vaan se on säännöllisesti 19 sekuntia TAI-aikaa jäljessä [5]. Tämän seurauksena myös GPS- ja UTC-ajan välinen ero muuttuu karkaussekuntien aikana. Tällä hetkellä GPS-aika on UTC-aikaa 13 sekuntia edellä.

Järjestelmän satelliittien kellot voivat poiketa hieman GPS-ajasta (ns. satelliitti-aika). Tarvittavat korjaukset satelliitin ajan muuttamisesta GPS-ajaksi välitetään vastaanottimeen satelliitin lähettämässä navigointiviestissä. [4]

#### 2.3. Ajan ylläpito elektronisessa järjestelmässä

Elektroniikassa kello voidaan mieltää signaaliksi, jossa esiintyy säännöllisesti pulsseja. Pulssit ilmaisevat ajan kulumisen ja niiden esiintymistiheyden avulla voidaan päätellä kellon oikea taajuus. Tietyissä sovelluksissa riittää, että kellot käyvät samaa tahtia, mutta yleensä myös kellojen vaiheen tulee olla tarkasti yhtenevä. Tällöin kaikissa järjestelmän komponenteissa on mahdollista suorittaa esimerkiksi informaation lähettäminen samanaikaisesti siirtotielle laitteen sijainnista riippumatta.

Yleisen digitaalisen järjestelmän ja reaaliaikakellojen synkronoinnissa on merkittävä ero. Digitaalisen järjestelmän kannalta on yleensä riittävää, että kellosignaalin taajuus ja vaihe on tarkasti toistettavissa jokaisessa järjestelmän komponentissa. Suorituskyvyn kannalta ei ole välttämättä merkitystä sillä, mihin kellosignaalin peräkkäisistä pulsseista järjestelmän eri osat ovat synkronoituneet.

Vaikka synkronoitavan testiverkon tapauksessa kaikkien tietokoneiden kellot kävisivät täysin samalla taajuudella ja sekunti vaihtuisi kaikkialla samanaikaisesti, ei voida olla varmoja, ovatko kellot täysin samassa ajassa vai onko kellojen suhteellinen poikkeama kokonaisia sekunteja.

Testiverkon tapauksessa on kyse juuri reaaliaikakellojen synkronoinnista. Verkon synkronoimiseksi riittävällä tarkkuudella on pystyttävä siirtämään ja toistamaan absoluuttinen ajastusinformaatio jokaisessa verkon laitteessa. Kellon taajuuden ja vaiheen synkronoimiseksi tarvittava informaatio voidaan välittää kellosignaalin avulla. Synkronointi absoluuttisen ajan suhteen on saavutettavissa kellosignaalin rinnalla käytettävän aikaleiman avulla.

Sähköinen kellosignaali tuotetaan tavallisimmin oskillaattorilla, joista tyypillisin on kvartsikideoskillaattori. Suurempaa tarkkuutta tarvittaessa voidaan käyttää rubidium- tai cesium-pohjaisia atomikelloja. Tieteellisessä tutkimuksessa voidaan käyttää vielä paremman stabiiliuden omaavia vety-MASER:eita tai optisia taajuusstandardeja. Tyypillisesti nämä referenssikellot tuottavat kiinteän 1 MHz, 5 MHz tai 10 MHz ulostulosignaalin. [6]

### *2.3.1. Synkronoinnin tarve ja toteutus*

Tietoliikenneverkoissa synkronoinnin tehtävänä on varmistaa luotettava ja nopea tiedonsiirto verkon eri komponenttien välillä riippumatta niiden välisestä etäisyydestä. Digitaalisen informaation siirtäminen kahden verkon komponentin välillä on mahdollista vain, jos yhteyden molemmat päät ovat synkronoituja ja yhteysvälillä siirrettävät bitit voidaan tulkita yksiselitteisesti. [6]

Tiedonsiirto voidaan jakaa verkon synkronointiasteen perusteella synkroniseen ja asynkroniseen tiedonsiirtoon. Synkronoidussa järjestelmässä kaikkien verkon laitteiden kellot on sidottu tarkasti toisiinsa. Asynkronisessa järjestelmän kellot on synkronoitu nimellisesti samaan referenssiin, mutta niiden välillä voi olla suhteellista poikkeamaa järjestelmän suorituskyvyn heikentymättä. Testiverkon liikennöinti on luonteeltaan asynkronista. [6]

Yhteyden päätepisteissä synkronointia tarvitaan alkuperäisen kellosignaalin muodostamiseksi. Kellosignaalin siirtäminen heikentää kuitenkin ajastuksen tarkkuutta. Siirrettäessä kellosignaali altistuu kohinalle, signaalitason muutoksille ja dispersiolle. Vastaanottimen tulee poistaa signaaliin muodostuneet virheet ja muodostaa alkuperäisen kaltainen kellosignaali. [6]

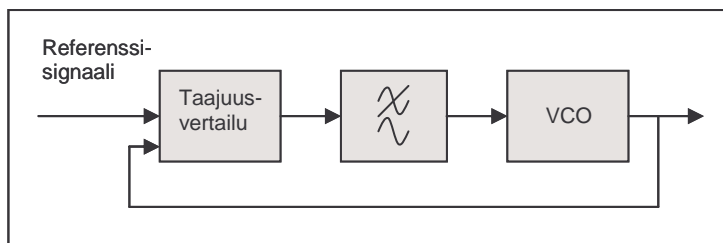
Synkronoinnin toteuttamiseksi tarvitaan ilmaisim, jolla voidaan havaita ero referenssikellon ja paikallisen kellon välillä halutun suureen osalta. Saatua erotussignaalia suodatetaan häiriöistä johtuvien nopeiden muutoksien poistamiseksi ennen sen käyttämistä paikallisen kellon virheen korjaamiseen. [6]

Tähän tarkoitukseen käytetään esimerkiksi kuvan 2.3. mukaista vaihelukittua silmukkaa (PLL, Phase Locked Loop). Vaikka pohjimmiltaan on kyseessä synkronointi taajuuden suhteen, kellot synkronoidaan vaiheen avulla. Näin saavutetaan huomattavasti parempi erottelukyky ja herkkyys kuin taajuuden avulla tapahtuvassa synkronoinnissa [6].

## 2.4. Kellon virheet ja hyvyyden määrittely

### 2.4.1. Taajuusvirhe

Taajuusvirheellä tarkoitetaan paikallisen kellon taajuuden poikkeamaa referenssikellon taajuudesta. Taajuuslukitussa silmukassa voidaan vertaamalla referenssisignaalia oskillaattorilta takaisinkytkettyyn signaaliin määrittää referenssikellon ja paikallisen kellon välinen taajuusero. Saadusta erotuksesta poistetaan nopeat transientit alipäästösuodattimen avulla ja sitä käytetään edelleen säädettävän oskillaattorin (VCO, Voltage Controlled Oscillator) ohjaamiseen. [6]

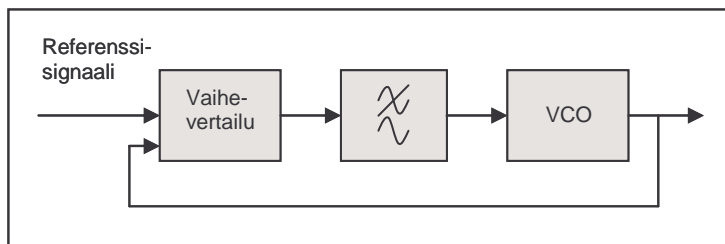


Kuva 2.2. Taajuuslukitun silmukan periaate. [6]

Johtuen taajuuden määritelmästä tulee taajuuseron mittaaminen suorittaa vähintäänkin kokonaisen signaalijakson yli. Käytännössä tarvitaan vieläkin pidempi mittaus, jotta tulokseen saadaan riittävä resoluutio. Pitkän mittausajan seurauksena tulokseksi saadaan kuitenkin vain keskimääräisiä arvoja ja taajuuteen jää edelleen epävarmuutta. Taajuuseron systemaattinen mittausvirhe tai säätöketjun itsensä aiheuttama virhe johtaa näin ollen helposti oskillaattorin virheelliseen ulostulotaajuuteen. [6]

### 2.4.2. Vaihevirhe ja absoluuttinen poikkeama

Vaihevirheellä tarkoitetaan kellosignaalin vaiheen poikkeamista referenssikellon vaiheesta. Suorakaideaaltoa sisältävien kello- ja referenssisignaalien kohdalla tämä voidaan havaita signaalien nousevien reunojen välisenä poikkeamana. Vaihe on kuitenkin täysin sidoksissa taajuuteen, mikä mahdollistaa taajuuden synkronoinnin vaihetta apuna käyttäen [6].



Kuva 2.3. Vaihelukitun silmukan periaate. [6]

Kuvan 2.3. mukaisessa vaihelukitus silmukassa vaihevertailun tuloksena saatu hetkellinen vaihe-ero sisältää joka tapauksessa hieman vaihevirhettä säätöketjun epäideaalisuuksien seurauksena. Jos vaihevirhe saadaan kuitenkin pidettyä lähes vakiona, saadaan ulostulosignaalin taajuus synkronoitua tarkasti referenssisignaalin taajuuteen vaihevirheen suuruudesta riippumatta. [6]

Taajuus- tai vaihelukittua silmukkaa voidaan käyttää taajuuden muuttamiseen lisäämällä takaisinkytkentään VCO:n ja vertailupiirin väliin komponentti, jolla näytesignaalin taajuutta voidaan muuttaa. Näytetaajuuden jakaminen kahdella nostaa VCO:n ulostulotaajuuden kaksinkertaiseksi. Vastaavasti näytetaajuuden kaksinkertaistaminen puolittaa VCO:n ulostulotaajuuden.

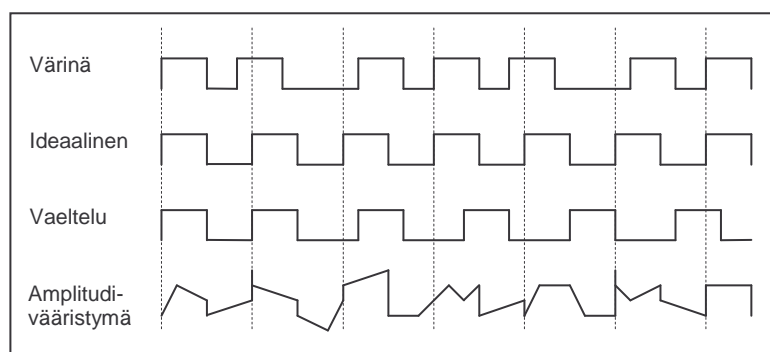
Reaaliaikakelloissa voi olla edellä mainittujen virheiden lisäksi absoluuttista poikkeamaa. Tällöin paikallinen kello poikkeaa referenssikellosta kellosignaalin jakson kokonaisia monikertoja. Reaaliaikasynkronoinnissa on välttämätöntä kyetä yksiselitteisesti numeroimaan kellosignaalin pulssit ja ilmaisemaan, paljonko kellonaika kunkin pulssin kohdalla on.

Tähän tarkoitukseen käytettävä ns. aikaleima voidaan sijoittaa pulssin aloittavaan eli nousevaan reunaan tai päättävään eli laskevaan reunaan. Näistä nousevan reunan käyttö on suositeltavin, koska elektroniikkakomponenttien ominaisuuksista johtuen nouseva reuna pystytään yleensä muodostamaan tarkemmin kuin laskeva reuna.

### 2.4.3. Kellosignaalin vääristymät

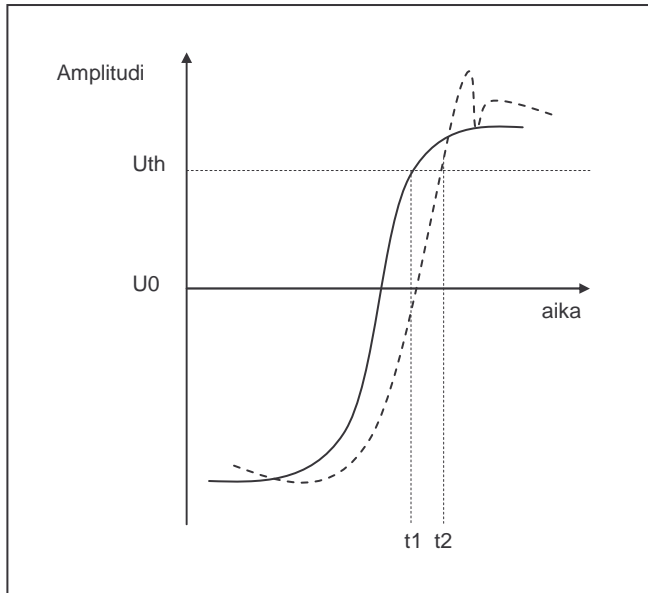
Häiriötekijät voivat vääristää sekä signaalin vaihetta että amplitudia. Signaalin vaiheessa voi ilmetä hidasta vaeltamista (wander) tai nopeaa värinää (jitter). Jälkimmäistä nimitystä käytetään, mikäli vaiheen muutosnopeus on suurempi kuin 10 Hz [7].

Kuvassa 2.4. on esitetty kellosignaalin erilaisia vääristymiä. Värinän seurauksena kello-signaalin vaiheessa on nopeita hetkellisiä poikkeamia ideaaliseen tilanteeseen verrattuna. Vaeltelu taas poikkeuttaa signaalin vaihetta vähitellen. Signaalin amplitudin vääristyminen muuttaa signaalin muotoa ja voi edellä mainittujen poikkeamien ohella aiheuttaa pulssin virheellisen tulkinnan. [7]



Kuva 2.4. Kellosignaalin vääristymiä.

Digitaalinen järjestelmä havaitsee pulssin nousevan reunan kuvan 2.5. mukaisesti, kun signaalin jännite ylittää määritellyn raja-arvon  $U_{th}$ . Mikäli pulssin reuna siirtyy tai vääristyy virheiden seurauksena, liipaisu tapahtuu ideaaliseen tilanteeseen verrattuna liian aikaisin tai myöhään. Synkronoinnissa tämä aiheuttaa virhettä vastaanottimen saamaan ajastusinformaation, mikäli kellosignaalia ei pystytä korjaamaan tai virheellisiä pulsseja jätetä huomiotta.



Kuva 2.5. Pulssin vääristymän vaikutus liipaisuhetkeen.

Kuvan 2.5. mukaisen pulssin (yhtenäinen viiva) liipaisu tapahtuu ajanhetkellä  $t_1$  signaalin tason ylittäessä asetetun raja-arvon  $U_{th}$ . Elektroniikkakomponenttien muodostamalla pulssilla on aina äärellinen nousuaika (Rise Time) eli jännite ei nouse hetkessä huippuarvoonsa vaikka kyseessä olisikin suorakaideaalto. Tämä näkyy kuvassa 2.5. pyöristyneenä pulssin reunana.

Katkoviivalla esitetyn pulssin amplitudi ja vaihe on vääristynyt. Vääristymisen seurauksena liipaisuehto täyttyy vasta ajanhetkellä  $t_2$ , mikä tarkoittaa ajastukseen  $t_2 - t_1$  suurista virhettä. Myös liipaisuehdossa on epämääräisyyttä eli liipaisu tapahtuu signaalitason ollessa mahdollisesti hieman raja-arvon  $U_{th}$  ylä- tai alapuolella. Saavutettava tarkkuus määräytyy elektroniikkakomponenttien nopeudesta ja erottelukyvystä.

#### 2.4.4. Kellon hyvyyden määrittely ja tunnusluvut

Jos kellolla on suuri taajuusvirhe nimelliseen taajuuteen verrattuna, sen taajuuden tarkkuus on huono. Jos taajuusvirhe on kuitenkin vakio, kellon taajuuden stabiilius on hyvä riippumatta taajuuden huonosta tarkkuudesta. [7]

#### 2.4.4.1. Tarkkuus

Ajan tarkkuus (accuracy) ilmoittaa kuinka tarkasti kellonaika vastaa UTC-aikaa tietyllä tarkasteluvälillä. Taajuuden kohdalla tarkkuus ilmaisee kuinka hyvin taajuus vastaa nimellistä taajuutta. Kellon tarkkuus puolestaan kertoo suurimman aika- tai taajuusvirheen, joka voidaan mitata käytännössä kellon koko elinaikana. [7]

Taajuuden tarkkuuden ilmoittamiseen käytetään taajuuspoikkeaman ja nimellisen taajuuden suhdetta, joka ilmoitetaan yksiköissä [ $\mu\text{Hz}/\text{Hz}$ ]. Tästä käytetään tekniikassa merkintää 1 ppm (Parts per Million), mikä tarkoittaa taajuudessa  $1 \cdot 10^{-6}$  Hz virhettä.

#### 2.4.4.2. Stabiilius

Kellon stabiiliudella (stability) tarkoitetaan kellon kykyä muodostaa tasaisia aikavälejä eli taajuutta. Stabiilius kuvaa satunnaisten ja säännöllisten vaihteluiden suuruutta tietyllä aikavälillä kellon hetkellisessä taajuudessa nimelliseen arvoon verrattuna. Tarkastelussa käytettävän aikavälin pituuden perusteella voidaan määritellä lyhyt- ja pitkäaikainen stabiilius. Määritelmien väliraja on kuitenkin tulkinnanvarainen ja vaihtelee käyttökohteesta riippuen. [7]

#### 2.4.4.3. Aika-alueen tunnusluvut

Aika-alueessa kellon stabiiliutta voidaan mitata ottamalla näytteitä  $y_i$  kellosignaalin taajuudesta  $y$ . Näytteiden perusteella kellon stabiiliutta voidaan kuvata erilaisilla tilastollisilla suureilla. ITU-T (International Telecommunication Union – Telecommunication Standardization Sector) ja ETSI (European Telecommunications Standards Institute) ovat määritelleet viisi tunnuslukua ajan ja taajuuden stabiiliuden kuvaamiseen. [7]

Allan-keskihajonta (ADEV, Allan Deviation) määritellään Allan-varianssin (AVAR) neliöjuurena. Modifioitu Allan-keskihajonta (MADEV, Modified Allan Deviation) on modifioidun Allan-varianssin (MVAR) neliöjuuri. Aikakeskihajonta (TDEV, Time Deviation) määritellään aikavarianssin (TVAR) neliöjuurena. Määritelmään kuuluvat myös aikavälivirheen neliösumman neliöjuuri ( $\text{TIE}_{\text{rms}}$ ) ja maksimaalinen aikavälivirhe (MTIE). [7]



Aika-alueen stabiiliuden tunnusluvut kuvaavat periaatteessa sitä, paljonko ajan ja taajuuden odotettava hajonta on tietyllä tarkasteluvälillä. Lisäksi ne kuvaavat taajuusalueen tunnuslukuja tehokkaammin järjestelmän pitkäaikaista suorituskykyä. Tämän takia aika-alueen mittauksen palvelevat ennen kaikkea tietoliikenteen sovelluksia. [7]

Todellinen (klassinen) varianssi (True Variance) on teoreettinen tunnusluku, joka lasketaan äärettömän näytemäärän yli. Todellinen varianssi ilmaisee tilastollisen suureen epästabiiliutta. Määritelmän mukaan todellinen varianssi [8]:

$$I^2(\tau) = \langle \bar{y}_k^2 \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=N}^N y_k^2, \quad (2.1)$$

jossa kulmasulut merkitsevät keskiarvoa äärettömän näytemäärän yli. Näytevarienssi (Sample Variance) on käytännöllisempi estimaatti aika-alueen stabiiliudesta ja perustuu äärelliseen näytemäärään  $N$ . Näytevarienssin määritelmäksi saadaan [8]:

$$\sigma^2(N, T, \tau) = \frac{1}{N-1} \sum_{i=1}^N \left( \bar{y}_i - \frac{1}{N} \sum_{j=1}^N \bar{y}_j \right)^2, \quad (2.2)$$

jossa välein  $T$  toistuvia  $\tau$ :n mittaisia näytteitä otetaan huomioon  $N$  kappaletta. Allan-varienssi (AVAR) määritellään edellisen pohjalta siten, että näytemäärä  $N = 2$  ja  $T = \tau$ . Laskenta perustuu kahden peräkkäisen näytteen aikakeskiarvon laskentaan. Allan-varienssi saa seuraavan lausekkeen [8]:

$$\sigma_y^2(\tau) = \frac{1}{2} \langle (\bar{y}_2 - \bar{y}_1)^2 \rangle. \quad (2.3)$$

Myös Allan-varienssi on teoreettinen tunnusluku, joka määritellään äärettömälle näytemäärälle. Äärellisen näytemäärän perusteella voidaan kuitenkin määrittellä vain estimaatteja Allan-varienssille. Yleisesti käytetty äärellisen näytemäärän ja mittausajan perusteella saatava estimaatti on [8]:

$$\sigma_y^2(\tau, m) = \frac{1}{2(m-1)} \sum_{i=1}^{m-1} (\bar{y}_{i-1} - \bar{y}_i)^2, \quad (2.4)$$

jossa  $\tau$  on mittausaika.

Allan-varianssin muunnelma ns. modifioitu Allan-varianssi (MVAR) määritellään äärelliselle näytejoukolle [8]:

$$Mod\sigma_y^2(n\tau_0) = \frac{1}{2} \left\langle \left[ \frac{1}{n} \sum_{i=1}^n (\bar{y}_{i+n} - \bar{y}_i) \right]^2 \right\rangle, \quad (2.5)$$

jossa  $\tau_0$  on näyteväli ja  $n\tau_0$  mittauksen kokonaisaika.

Allan-varianssi ja modifioitu Allan-varianssi sopivat taajuusreferenssien epästabiiliuden kuvaamiseen. Tietyissä tilanteissa on kuitenkin havainnollisempaa mitata stabiiliutta ajan avulla taajuuden sijasta. Tähän tarkoitukseen soveltuu aikavarianssi (TVAR, Time Variance) [9]:

$$\sigma_x^2(\tau) = \frac{\tau^2}{3} Mod\sigma_y^2(\tau). \quad (2.6)$$

AVAR ja MVAR ovat taajuuden stabiiliutta kuvaavia tunnuslukuja. AVAR soveltuu erityisesti kellojen ja oskillaattorien pitkäaikaisen stabiiliuden kuvaamiseen. MVAR soveltuu paremmin sähköisen kohinan ja lyhytaikaisen taajuuden stabiiliuden kuvaamiseen. [9]

TVAR kuvaa ajan stabiiliutta ja soveltuu myös sähköisen kohinan kuvaamiseen. Sitä voidaan soveltaa ajan jakamiseen, vertailuun ja mittaamiseen käytettävien järjestelmien stabiiliuden kuvaamiseen. Lisäksi se soveltuu erinomaisesti televerkkojen synkronoinnin stabiiliuden tunnusluvuksi. [9]

Kellon poikkeamaa ideaalisesta referenssinajasta voidaan kuvata aikavälivirheellä (TIE, Time Interval Error). Aikavälivirhe määritetään kellon aikavirheen (TE, Time error) avulla. Aikavirhe tarkoittaa kellon hetkellisen poikkeaman suuruutta suhteessa referenssikelloon. [7]

Aikavälivirhe [10]:

$$TIE(t) = TE(t_0 + t) - TE(t_0), \quad (2.7)$$

jossa tarkastelu alkaa hetkellä  $t_0$  ja päättyy hetkellä  $t_0 + t$ .

Aikavälivirheen neliökeskiarvon neliöjuuri ( $TIE_{rms}$ , Root Mean Square of the Time Interval Error) [7]:

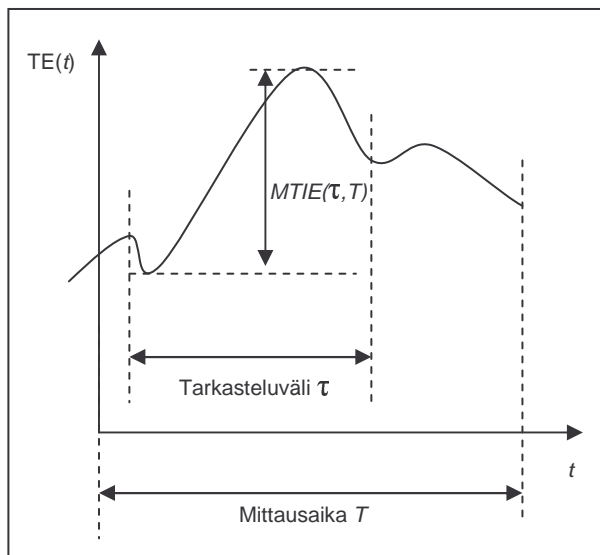
$$TIE_{rms}(t; \tau) = \sqrt{E\{(TE(t + \tau) - TE(t))^2\}} . \quad (2.8)$$

Äärellisen näytämäärän  $N$  perusteella sille saadaan seuraavaa estimaatti [7]:

$$TIE_{rms}(n\tau_0) = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-n} (TE_{i+n} - TE_i)^2} .$$

Maksimaalinen aikavälivirhe ( $MTIE$ , Maximum Time Interval Error) tarkoittaa aikavälivirheen suurinta muutosta (peak-to-peak) mittausajan  $T$  tarkasteluvälillä  $\tau$  kuvan 2.6 mukaisesti. Sen matemaattinen määrittely on seuraava [7]:

$$MTIE(\tau, T) = \max_{0 \leq t \leq T-\tau} \left\{ \max_{t \leq t+\tau} [TE(t)] - \min_{t \leq t+\tau} [TE(t)] \right\} . \quad (2.9)$$



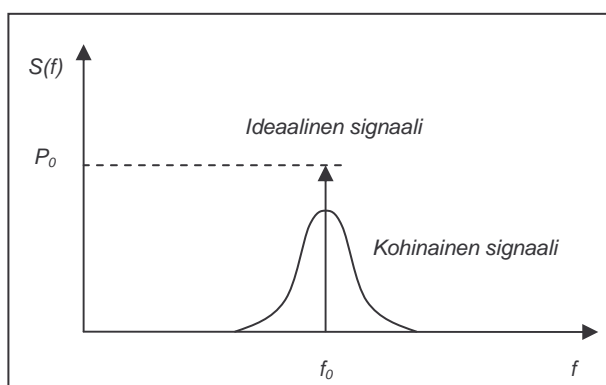
Kuva 2.6. Maksimaalisen aikavälivirheen määrittely. [7]

#### 2.4.4.4. Taajuusalueen tunnusluvut

Taajuusalueen mittausten avulla voidaan tutkia tehokkaasti referenssinkellojen kuten oskillaattorien ulostulotaajuuden ominaisuuksia. Suoraviivaisin tapa tutkia kellosignaalin tarkkuutta ja stabiiliutta on tarkastella suoraan sen tehospektritiheyttä (PSD, Power Spectral Density). [7]

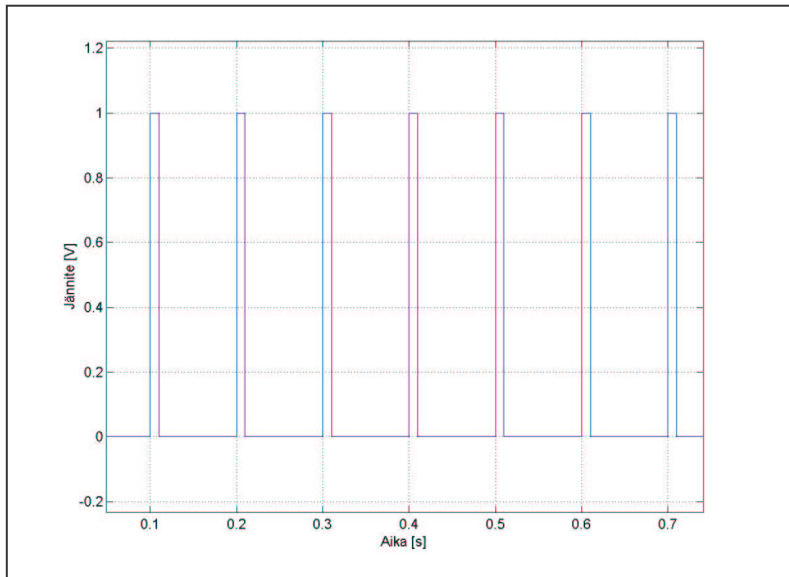
Tehospektritiheys  $S(f)$  on jatkuva funktio. Se kuvaa tehoa taajuuden funktiona, jonka kellosignaali siirtää sovitettuun kuormaan. Ideaalisessa tapauksessa yhdestä sinimuotoisesta taajuuskomponentista muodostuvan kellosignaalin tehospektritiheys on impulssi  $P_s\delta(f-f_0)$ , jossa  $P_s$  on signaalin teho ja  $f_0$  kellosignaalin perustaajuus kuvan 2.7. mukaisesti. [7]

Epäideaalisessa tapauksessa kellosignaali altistuu vaihekohinalle ja amplitudivääristymille, minkä seurauksena tehospektritiheys sisältää myös perustaajuudesta  $f_0$  poikkeavia taajuuskomponentteja [7]. Tämä aiheuttaa impulssin levenemisen kuvassa 2.7. esitetyn periaatteen mukaisesti.



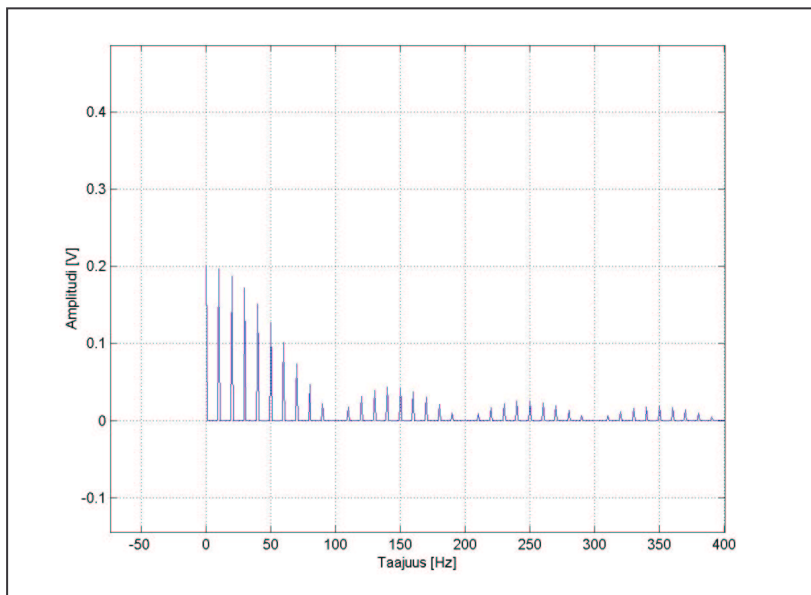
Kuva 2.7. Ideaalisen ja kohinaisen sinisignaalin tehospektritiheys. [7]

Käytännön kellosignaali on yleensä suorakaideaaltoa. Kuvassa 2.8. on esitetty 10 Hz suorakaideaalto, jonka pulssisuhde (varsinaisen pulssin suhde signaalin jaksonaikaan) on 0,1. Sen amplitudispektri saadaan laskettua edellä mainitusta aika-alueen esitysmuodosta FFT-muunnoksen (Fast Fourier Transform) avulla.



Kuva 2.8. 10 Hz suorakaideaalto Matlab-ohjelmassa.

Kuvan 2.9. perusteella suorakaideaallosta muodostuvan kellosignaalin amplitudispektri ei enää olekaan yksittäinen impulssi vaan se muodostuu perustaajuuden  $f_0$  välein esiintyvistä taajuuskomponenteista (perustaajuuden harmonisista), joiden muodostama verhoikäyrä noudattaa sinc-funktion<sup>1</sup> itseisarvoa.



Kuva 2.9. 10 Hz suorakaideaallon FFT-muunnos (amplitudispektri).

[1] Sinc-funktion matemaattinen määrittely on  $\text{sinc}(x) = \frac{\sin(\pi x)}{(\pi x)}$ .

Kuvan 2.8. mukaisen suorakaidepulssin leveys  $d = 10$  ms. Tällöin verhokäyrän ensimmäinen nollakohta (kuva 2.9.) sijoittuu taajuudelle  $1/d = 100$  Hz. Seuraavat nollakohdat toistuvat säännöllisesti tämän monikertoina taajuuksilla  $2/d$ ,  $3/d$  jne.

Amplitudispektri kuvaa taajuuskomponenttien amplitudia, eikä siitä käy ilmi mikä on tietyn taajuuskomponentin vaihe. Jos kuvan 2.9. muunnoksessa otettaisiin huomioon myös taajuuskomponenttien vaihe, saisi spektri negatiivisia arvoja välillä 100 Hz–200 Hz, 300 Hz–400 Hz jne.

## 2.5. Aika- ja taajuusreferenssejä

### 2.5.1. Kideoskillaattorit

Kvartsikideoskillaattorien toiminta perustuu pietsosähköiseen ilmiöön. Halpoja kideoskillaattoreita käytetään yleisesti elektronisissa laitteissa kellosignaalin muodostamiseen. Korkealuokkaisia kideoskillaattoreita voidaan käyttää myös toissijaisina taajuusstandardeina tai viimeisenä, atomin resonanssin avulla ohjattavana asteena atomikelloissa. [7]

Kideoskillaattori (XO, Crystal Oscillator) on käytännössä sähköinen oskillaattori, jonka kvartsikide tuottaa resonanssitaajuutensa mukaisen signaalin. Oskillaattorin taajuus on tyypillisesti 5 MHz–10 MHz. Takaisinkytkentäsilmuksaan sijoitetun kideoskillaattorin avulla voidaan saavuttaa erittäin hyvä lyhytaikainen ( $< 1$  s) stabiilius. [7]

Kvartsikideoskillaattorien tarkkuuteen ja stabiiliuteen vaikuttaa eniten lämpötila, käytetty signaalitaso, mekaaninen rasitus, materiaalien absorptio ja emissio, kiihtyvyyden tärinä sekä ympäröivä sähkö- ja magneettikenttä [11].

Kideoskillaattorin resonanssitaajuus määräytyy lähes yksinomaan kiteen fyysisistä dimensioista. Lämpötilan muutoksen vaikuttavat lämpölaajenemisen seurauksena kiteen dimensioihin, mikä vaikuttaa edelleen oskillaattorin ulostulotaajuuteen. Taajuuden muutoksen suuruus riippuu kiteen leikkaustavasta sekä muista oskillaattorin valmistukseen käytetyistä menetelmistä. [11]

Jännitteellä ohjattava kideoskillaattori (VCXO, Voltage-Controlled Crystal Oscillator) on oskillaattori, jonka taajuutta voidaan säätää ulkoisen jännitteen avulla. Sitä käytetään yleisesti vaihelukitun silmukan toteuttamiseen. Oskillaattorin taajuuden säätäminen perustuu takaisinkytkentä silmukassa olevaan säädettävään kapasitanssiin. Se muodostuu tavallisesti kiinteästä kondensaattorista ja varaktorista eli kapasitanssidiodista, jonka kapasitanssi pienenee estosuuntaisen jännitteen noustessa. [7]

Kideoskillaattorin taajuuden lämpötilariippuvuutta voidaan pienentää lämpötilakompensoinnin (TCXO, Temperature-Compensated Crystal Oscillator) tai kiteen uunittamisen (OCXO, Oven-Controlled Crystal Oscillator) avulla. Lämpötilakompensointi tapahtuu esimerkiksi liittämällä kiteen yhteyteen komponentti tai säätöpiiri, jonka lämpötilariippuvuus kompensoi kiteen resonanssitaajuuden muutoksen. [7]

Lämpötilakompensoinnin avulla voidaan saavuttaa jopa  $10^{-8}$ -suuruusluokkaa oleva taajuuden stabiilius 0 - 70 °C lämpötilavälillä. Kaksinkertaisen uunittamisen avulla saavutettava stabiilius on päivää kohti parhaimmillaan  $10^{-11}$ , mikä vastaa käytännössä rubidium-atomikellojen suorituskykyä. [7]

### *2.5.2. Atomikellot*

Atomikellojen toiminta perustuu atomin energiatilojen muutokseen. Atomin viritystilan purkautuessa atomi emittoi energiakvantin, fotonin, jonka sisältämä energia vastaa kahden energiatilan erotusta. Nykyiset atomipohjaiset taajuusstandardit perustuvat vedyn, rubidiumin tai cesiumin ominaisuuksiin. [7]

Vety-MASER (Microwave Amplification by Stimulated Emission of Radiation) perustuu sähkömagneettisen säteily stimuloituun emissioon taajuudella, joka vastaa vetyatomin energiatilojen välistä muutosta. Lähteestä peräisin olevien vetyatomien joukosta valitaan voimakkaan magneettikentän avulla virittyneessä tilassa olevat atomit. [7]

Atomit ohjataan kammioon, joka on suojattu ulkoisilta häiriöiltä magneettikentällä. Kammiota ympäröi lisäksi resonaattori jonka resonanssitaajuus on viritetty vedyn ominaistajuudelle (1,42040575177 GHz). Kammiossa atomit kimpoilevat ja niiden viritystila purkautuu. [7]

Viritystilän purkautuminen emittoi säteilyä vakiotaajuudella. Atomeista purkautuva säteily stimuloi edelleen uusien atomien emissiota. Saatua taajuutta käytetään atomikellon viimeisenä asteena olevan VCXO-oskillaattorin taajuuden säätämiseen. [7]

Vety-MASER:lla on tämän hetkisistä atomikelloista paras lyhyen ja keskipitkän aikavälin stabiilius. Kellon pitkäaikaiseen stabiiliuteen vaikuttaa kuitenkin mekaaninen värinä ja lämpötilan muutokset, jotka voivat muuttaa resonaattorin resonanssitaajuutta. [7]

Rubidium-taajuusstandardin toiminta perustuu  $^{87}\text{Rb}$ -isotoopin energiatilojen muutokseen.  $^{87}\text{Rb}$ -atomeista saatavaa säteilyä suodatetaan  $^{85}\text{Rb}$ -atomeista koostuvan kaasun avulla ennen säteilyn ohjaamista  $^{87}\text{Rb}$ -atomeilla täytettyyn absorptiokennoon. [7]

Suodatuksen avulla valitaan vain sellainen säteilyn taajuuskomponentti, joka kykenee virittämään absorptiokennossa sijaitsevat atomit. Viritystilän purkautumisesta vapautuvan säteilyn taajuudella ohjataan edelleen atomikellon viimeisenä asteena olevaa VCXO-oskillaattoria. Viritystilojen purkautumista stimuloidaan radioaallolla, jonka taajuus on 6,3846826128 GHz. Kalibroinnin jälkeen rubidium-atomikellolla ylläpidettävän taajuuden tarkkuus on  $10^{-11}$ -suuruusluokkaa. [7]

Cesium-taajuusstandardin toiminta perustuu  $^{133}\text{Cs}$ -isotoopin energiatilojen muutokseen. Uunista lähtevien atomien energiatilat ovat tasaisesti jakautuneet 16 energiatason kesken. Atomien joukosta valitaan magneettikentän avulla toisella viritystasolla olevat atomit, jotka ohjataan edelleen Ramsey-resonaattoriin. Resonaattorissa atomit altistetaan säädettävälle sähkömagneettiselle kentälle. Atomien viritystilän purkautumisehdon täytyessä sähkömagneettisen kentän taajuus on 9,192631770 GHz. [7]

Resonaattorin jälkeen atomien joukosta valitaan magneettikentän avulla ensimmäisellä energiatasolla olevat atomit, jotka ohjataan edelleen tunnistimelle. Tunnistimella virta on verrannollisen resonaattoriin saapuvien atomien määrään ja siten myös siirtymätodennäköisyyteen energiatilojen 1 ja 2 välillä. [7]

Tunnistin muodostaa yhdessä VCXO:n kanssa takaisinkytkentä silmukan, jonka avulla generoidaan atomikellon lähtösignaali ja säädetään resonaattorissa vaikuttavan sähkömagneettisen kentän taajuutta. Näin ollen kvartsioskillaattorin lyhytaikainen stabiilius yhdistyy sitä ohjaavan atomiresonaattorin pitkäaikaiseen stabiiliuteen. [7]



### 2.5.3. GPS-järjestelmä

GPS-järjestelmä (Global Positioning System) on monimutkainen satelliittinavigointijärjestelmä, joka mahdollistaa maailmanlaajuisen ja kolmiulotteisen paikannuksen reaaliajassa. Järjestelmä mahdollistaa myös nopeuden ja ajan määrittämisen maalla, merellä ja ilmassa missä päin maailmaa tahansa. [7]

Navigoinnin ja paikan määrittämisen lisäksi järjestelmää voidaan käyttää aika- ja taajuusreferenssinä, joka tarjoaa maailmanlaajuisesti tarkkaa aikaa, aikavälejä sekä taajuutta. GPS-järjestelmän rakennetta ja ominaisuuksia on käsitelty laajemmin luvussa 3.

### 2.5.4. Taajuusstandardiasemat

Ajan synkronointiin on käytettävissä maanpäällisiä radioasemia, jotka lähettävät ajastusinformaatiota tavallisesti LF- ja HF-radiotaajuuksilla (LF, Low Frequency, matalat taajuudet: 20 kHz – 80 kHz) (HF, High Frequency, korkeat taajuudet: 3 MHz – 30 MHz). Signaalin vastaanottaminen voidaan toteuttaa yksinkertaisilla ja halvoilla vastaanottimilla. [12]

Asemien käyttämät taajuudet ovat: 77,5 kHz (DCF77, Mainflingen, Saksa), 60 kHz (MSF, Rygby, Englanti), 60 kHz (WWVB, Fort Collins, Colorado), 2,5 MHz, 5 MHz, 10 MHz, 15 MHz sekä 20 MHz (WWV, Fort Collins, Colorado ja WWVH, Hawaii). Tämän lisäksi radioasemia löytyy Japanista, Ranskasta, Kanadasta ja Sveitsistä. [13] [14] [12]

USA:ssa sijaitsevien asemien (WWV/B/H) lähettämä signaali on käytettävissä lähes koko USA:n alueella ja osittain myös Etelä-Amerikassa [13]. Englannissa sijaitseva radioasema (MSF) palvelee osittain myös Pohjois- ja Länsi-Eurooppaa [14]. Saksassa sijaitsevan radioaseman (DCF77) signaali voidaan vastaanottaa pääsääntöisesti myös Etelä-Suomessa [12].

Radioasemat välittävät tyypillisesti oman maansa virallista aikaa. DCF77:n lähettämä aika (Saksan virallinen aika) on kesä- ja talviajasta riippuen yhden tai kaksi tuntia UTC-aikaa edellä. Ajan ylläpitoon käytetään PTB:n (Physikalisch-Technische Bundesanstalt) atomikelloja. Lähetykseen käytettävän 77,5 kHz kanta-aallon vaihe on synkronoitu atomikelloihin siten, että siinä ei koskaan esiinny 0,3 µs suurempia vaihteluita. [12]

Aika välitetään 59 bitin mittaisessa kehyksessä, joka sisältää oman osionsa sekunneille, minuuteille, tunneille, viikonpäivälle, kuukaudelle ja vuodelle. Tämän lisäksi kehyksessä välitetään tietoja käytettävästä aikavyöhykkeestä ja antennista sekä informaation pariteetista virhetarkistusta varten. [12]

Kehys sidotaan aikaan pienentämällä kantoaallon amplitudi neljännekseen aina sekunnin vaihtuessa joko 100 ms tai 200 ms ajaksi. Minuutin vaihtuminen ilmaistaan jättämällä jokaisen kehyksen viimeinen sekuntimerkki lähettämättä. Tiedot välitetään BCD-koodattuna (Binary Coded Decimal) käyttämällä pulssinleveysmodulaatiota nopeudella 60 bit/s. Tällöin sekuntimerkit lähetetään joko 100 ms (looginen nolla) tai 200 ms (looginen yksi) mittaisina. [12]

Signaalin etenemisestä aiheutuva muutaman millisekunnin suuruinen viive tulee korjata vastaanotossa manuaalisesti. Käytetyn moduloinnin takia signaali on herkkä häiriöille (säätötilan ja sähköisten häiriöiden vaikutus). Järjestelmän avulla suoritettavan synkronoinnin tyypillinen tarkkuus on 3 ms suuruusluokkaa. Suurempaa taajuutta ja erilaista modulaatiota käyttävillä järjestelmillä voidaan saavuttaa huomattavasti parempi suorituskyky. [12]

### 3. GPS – Maailmanlaajuinen satelliittinavigointijärjestelmä

GPS on maailmanlaajuinen satelliittinavigointijärjestelmä, jota voidaan käyttää paikannukseen, navigointiin, kohteen sijainnin selvittämiseen, kartoitukseen ja ajastukseen liittyvissä sovelluksissa. GPS:n tarjoama paikannus- ja navigointipalvelu palvelee sekä siviili- että sotilaskäyttäjää. GPS:n avulla voidaan myös päästä erittäin hyvään ajastuksen tarkkuuteen esimerkiksi tietoliikenteeseen liittyvissä sovelluksissa.

GPS-järjestelmän rakentamispäätös tehtiin vuonna 1973 Yhdysvaltain puolustusministeriössä DoD (Department of Defence). Tuolloin päätettiin laivaston ja ilmavoimien omien Timation ja Program 612B -suunnitelmien sulauttamisesta uudeksi maailmanlaajuiseksi järjestelmäksi nimeltään Navstar / GPS (NAVigation System with Time and Ranging / Global Positioning System). Laivaston aiemmasta Transit (NNSS) – satelliittijärjestelmästä saadut kokemukset otettiin huomioon uutta järjestelmää suunniteltaessa. [15]

Transit-järjestelmän satelliittien kiertorata oli 1100 km korkeudella maan pinnasta. GPS-järjestelmän noin 20200 km sijaan, joten maan gravitaatiokentän vaihtelun vaikutus satelliitteihin oli huomattavasti suurempi. Radioyhteydellä käytetyt matalat taajuudet, 150 ja 400 MHz, olivat alttiina ionosfäärin aiheuttamille viiveille ja häiriöille. Alhaisen kiertoradan seurauksena kukin satelliitti oli näkyvissä ainoastaan 20 minuuttia kerrallaan. [2]

#### 3.1. GLONASS

GLONASS (Global Navigation Satellite System) on venäläinen vastine GPS-järjestelmälle. Tällä hetkellä toiminnassa on yhteensä yhdeksän satelliittia kahdella eri ratatasolla. Satelliitit on laukaistu 13.10.2000 – 10.12.2003 välisenä aikana kolmen kappaleen erissä. [16]

GLONASS-järjestelmän perusrakenne on hyvin samankaltainen GPS-järjestelmän kanssa. Järjestelmien keskeisiä eroja ja yhtäläisyyksiä on vertailtu taulukossa 3.1. Suurimmat erot järjestelmien välillä ovat satelliittien jakautumisessa ratatasoille sekä ratatasojen määrässä ja radioyhteydessä. [2]

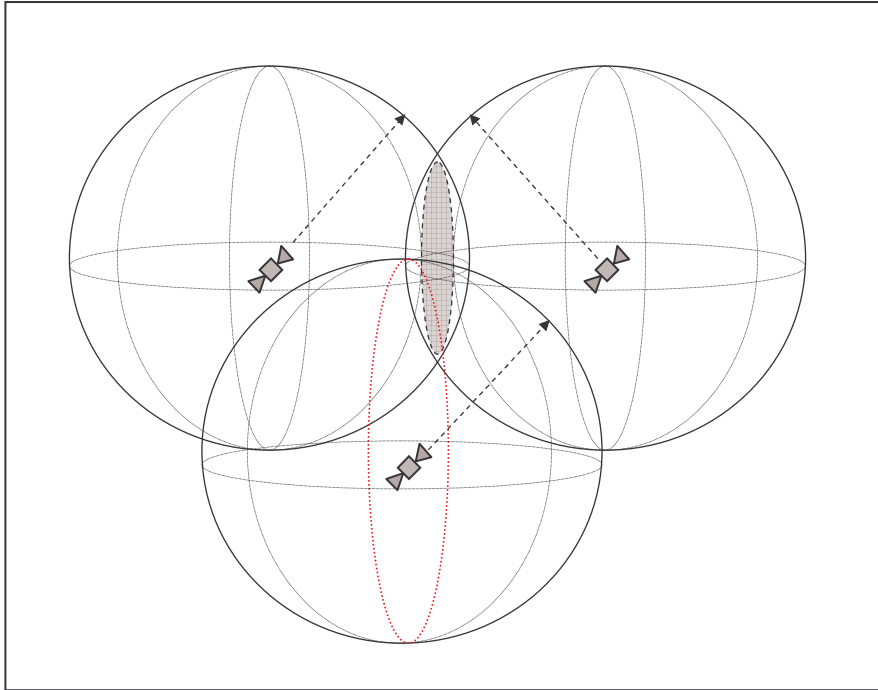
Taulukko 3.1. GPS- ja GLONASS-järjestelmien vertailu. [2]

Satelliitit	GPS	GLONASS
Satelliitteja	24	24
Ratatasoja	6	3
Satelliitteja per ratataso	4	8
Ratatason inkliinaatio	55 astetta	64,8 astetta
Kiertoradan säde	26560 km	25510 km
Kiertoaika	n. 11 h 58 min	n. 11 h 15 min
Signaalit		
Radioyhteyden kanavointi	CDMA	FDMA
Kantoaallot	L1: 1575,42 MHz L2: 1227,60 MHz	L1: 1602,5625 - 1615,5 MHz L2: 1246,4375 - 1256,5 MHz
C/A-koodin symbolinopeus (L1)	1,023 MHz	0,511 MHz
P-koodin symbolinopeus (L1, L2)	10,23 MHz	5,11 MHz
Navigointiviesti		
Kesto (minuuttia)	12,5	2,5
Koko (bittinä)	37500	7500
Yleistä		
Referenssiaika	UTC(USNO)	UTC(SU)
Tahallinen heikennys	kyllä	ei suunniteltu
P-koodin salaus	kyllä	Mahdollinen

### 3.2. GPS-järjestelmän toiminta

Yksinkertaisuudessaan GPS-paikannus perustuu etäisyyden mittaamiseen tarkasti tiedetyissä koordinaateissa sijaitsevien satelliittien ja maanpäällisen vastaanottimen välillä. Etäisyyden mittaaminen toteutetaan mittaamalla signaalin kulku-aika satelliittien ja vastaanottimen välillä. Kun tiedetään, että signaali etenee valonnopeudella, voidaan kulkuajan perusteella laskea, kuinka pitkän matkan signaali on kulkenut ennen saapumistaan vastaanottimeen. [15]

Kuvassa 3.1 on havainnollistettu paikanmäärityksen geometriaa. Etäisyyden mittaaminen yhdestä satelliitista rajoittaa sijainnin pallopinnalle. Mittaamalla samanaikaisesti etäisyys toisesta satelliitista, tarkentuu sijainti mittaustuloksena saatujen kahden pallopinnan leikkausuralle. Lopulta kolmas mittaus rajoittaa sijainnin kahteen pisteeseen kolmen pallopinnan leikkauspisteissä. Käytännössä etäisyys pitää kuitenkin mitata neljästä satelliitista, kun kolmiulotteinen paikannus halutaan tehdä yksiselitteisesti. [15]



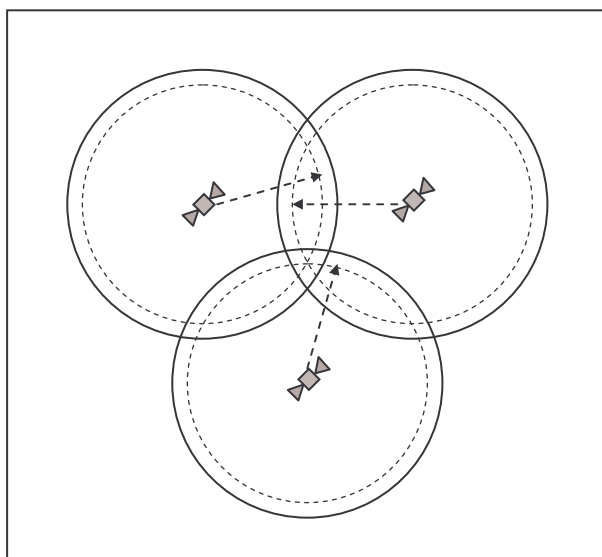
Kuva 3.1. Paikannäärityksen geometria.

Kulkuajan määrittämisessä ajanhetki, jolloin signaali lähti satelliitista, määritetään siten, että kaikki satelliitit ja vastaanottimet muodostavat samanlaisen koodin tarkalleen samanaikaisesti. Vastaanottimen tarvitsee vain vastaanottaa koodi ja mitata kuinka kauan sitten se itse muodosti vastaavan koodin. [15]

Käytännössä tämä toteutetaan siten, että jokainen satelliitti lähettää omaa näennäis-satunnaista koodiaan (PRN, Pseudorandom Noise), joka on monimutkainen sarja bittejä. Käynnistysvaiheessa GPS-vastaanotin saa satelliitilta navigointiviestin, joka sisältää mm. tiedot satelliitin radasta ja käytetystä PRN-koodista. [15]

Useimmissa GPS vastaanottimissa voidaan valita toimintatavaksi kaksi- tai kolmi-ulotteinen paikannus. Jos sijainnilla korkeussuunnassa ei ole merkitystä tai se on tiedossa (esim. merellä), voidaan käyttää kaksiulotteista paikannusta. Tällöin sijainti voidaan määrittää tarkasti käyttämällä vain kolmea mittausetta kolmiulotteisessa paikannuksessa tarvittavan neljän mittauksen sijaan. Lisämittauksen käytöllä on kuitenkin huomattava hyöty, sillä voidaan poistaa systemaattinen ajoitusvirhe satelliitin ja vastaanottimen välillä.

Mittausvirheen eliminointia on havainnollistettu kuvassa 3.2. Satelliitin lähettämä signaali etenee noin 300 000 km sekunnissa. Tällöin 10 mikrosekunnin virhe ajoituksessa tarkoittaa etäisyyksmittauksessa kolmea kilometriä. Kahdesta satelliitista mitatut etäisyydet (yhtenäiset ympyrät) leikkaavat vaikka etäisyydet olisi mitattu väärin.



Kuva 3.2. Kulkuajamittauksen virheen vaikutus.

Kun etäisyys mitataan kolmesta satelliitista, paljastuu ajoituksen virhe, sillä yhden yksiselitteisen leikkauspisteen sijaan saadaan tuloksena kuusi pistettä, joissa mitatut etäisyydet leikkaavat. Saaduista leikkauspisteistä voidaan kuitenkin varsin yksinkertaisesti päätellä, millä ajoituksen korjauksella mitatut etäisyydet (katkoviivoista muodostuvat ympyrät) saadaan leikkaamaan yhdessä ainoassa pisteessä.

Ajastuksen toimintaperiaatetta on käsitelty luvussa 3.4.4.

### 3.3. Järjestelmän rakenne

Nykyinen GPS-järjestelmä koostuu kolmesta lohkoista, jotka ovat avaruuslohko, valvontalohko ja käyttäjälohko. Avaruuslohko muodostuu kiertoradoilla olevista satelliiteista, jotka lähettävät paikannukseen ja ajastukseen tarvittavia tietoja maahan. Valvontalohko koostuu maanpäällisistä valvonta-asemista, jotka vastaanottavat satelliittien lähettämää signaalia ja tarvittaessa päivittävät uusia tietoja tai korjauksia satelliitteihin. Käyttäjälohkon muodostavat vastaanottimet, jotka vastaanottavat signaalia ja hyödyntävät GPS-järjestelmän tarjoamia palveluja.

### 3.3.1. Avaruuslohko

Alustavassa suunnitelmassa avaruuslohko koostui 24 satelliitista, jotka kiertäisivät maapalloa kolmella eri ratatasolla. Myöhemmin satelliittien määrää pienennettiin taloudellisista syistä kahdeksaantoista. Tässä suunnitelmassa satelliitit jakaantuivat kuudelle eri ratatasolle. Suunnitelma jouduttiin kuitenkin hylkäämään, koska se ei tarjonnut maailmanlaajuista 24 tunnin kattavuutta. [17]

Vuonna 1986 suunniteltu satelliittien kokonaismäärä nostettiin 21:een yhteensä kuudella eri ratatasolla. Suunnitelman mukaan kullakin ratatasolla oli 3 aktiivista satelliittia ja loput kolme satelliittia toimivat varasatelliitteina. [17]

Viimeisimmän suunnitelman mukaan avaruuslohko koostuu 24 satelliitista, jotka kiertävät maapalloa lähes pyöreillä kiertoradoilla kuudella eri ratatasolla noin 20200 km korkeudessa. Satelliittien kiertoaika on noin 11 tuntia 58 minuuttia ja niitä on neljä kappaletta ratatasoa kohti. Satelliiteista 21 on varsinaisessa käytössä ja kolme muuta ovat aktiivisia varasatelliitteja mahdollisten vikatilanteiden varalle. [18]

Järjestelmä mahdollistaa säästä riippumattoman ja jatkuvan maailmanlaajuisen saavutettavuuden samanaikaisesti neljästä kahdeksaan satelliittiin, jotka sijaitsevat vähintään 15 asteen elevaatiokulmassa<sup>2</sup> maanpäällisen tarkastelupisteen suhteen. [17]

Satelliitit jaetaan kolmeen luokkaan: Block I, Block II / IIA ja Block IIR. Yksitoista Block I -luokan satelliittia (SVN-numerot 1 - 11) laukaistiin vuosien 1978 ja 1985 välisenä aikana. Kaikki laukaisut onnistuivat lukuun ottamatta vuonna 1981 tapahtunutta laukaisuvikaa [2]. [5]

Maaliskuussa 1992 viisi näistä satelliiteista oli edelleen toimintakunnossa, vaikka Block I -luokan satelliittien suunniteltu elinikä oli vain 5 vuotta. [17] Kolme vuotta myöhemmin (1995) toimintakuntoisia satelliitteja oli enää kolme kappaletta [2]. Tällä hetkellä kaikki Block I -luokan satelliitit on poistettu käytöstä [5].

[2] Elevaatiokulmalla tarkoitetaan maanpinnalle tarkastelupisteeseen kuvitellun tangenttitason ja satelliitin näkösuoran välistä pienempää kulmaa.

Block I -luokan satelliitit laukaistiin siten, että niiden ratatasot tarjosivat parhaan tavoitettavuuden Yhdysvaltoihin. Block I -luokan satelliitit sisälsivät yhden cesium-atomikellon ja kaksi rubidium-atomikelloa [5]. Satelliitit pystyivät toimimaan 3 - 4 päivää ilman yhteyttä valvonta-asemaan. Satelliittien lähettämän informaation saatavuutta ei voitu mitenkään rajoittaa. [2]

Block II -luokan satelliitit (SNV-numerot 13 - 21) laukaistiin 1989 ja 1990 välisenä aikana [5]. Tämän luokan satelliittien suunniteltu elinikä on 7,3 vuotta. Jokaisessa Block II/IIA -luokan satelliitissa on kaksi kappaletta sekä rubidium- että cesium-atomikelloja [5]. Satelliiteissa on täysi valmius tahalliseen heikennykseen (SA, Selective Availability). Block II -luokan satelliitti kykenee toimimaan 14 vuorokautta ilman yhteyttä valvonta-asemaan tarkkuuden heikentymättä. [2]

Ensimmäinen Block IIA -luokan satelliitti (SVN-numerot 22 -40) laukaistiin 1990 ja viimeinen 1997. Block IIA -luokan satelliitit on modifioitu siten, että ne kykenevät periaatteessa toimimaan 180 vuorokautta ilman yhteyttä valvonta-asemaan. Normaaliosuhteissa valvontalohko kuitenkin päivittää satelliitin tietoja päivittäin. Mikäli yhteys valvontalohkoon katoaa, satelliitit siirtyvät poikkeus-tilaan, jolloin niiden avulla suoritettavan paikannuksen tarkkuus heikkenee vähitellen. [2]

Block IIR -luokan satelliitit (SVN-numerot 41 - 60) kykenevät toimimaan lähes autonomisesti. Ne pystyvät määrittämään oman kiertoratansa automaattisesti ja laskemaan tarvittavat arvot maahan välitettäviin navigointiviesteihin. Nämä satelliitit pystyvät toimimaan puoli vuotta ilman yhteyttä valvontalohkoon menettämättä tarkkuuttaan. Satelliittien suunniteltu elinikä on hieman alle kahdeksan vuotta [5]. [2]

Tällä hetkellä GPS-konstellatio koostuu 28 Block II/IIA/IIR -luokan satelliitista. Eri luokkien satelliittien määrä on yllä mainitussa järjestyksessä: 2, 16 ja 10 kappaletta. Tahallinen heikennys on ollut viimeksi aktivoituna Persianlahden sodan aikana vuonna 1991. Vuodesta 2000 eteenpäin tahallinen heikennys on ollut täysin poissa käytöstä. UTC-ajan viimeisin karkaussekunti ilmeni vuonna 1999, jonka seurauksena GPS- ja UTC-ajan välinen ero on nykyinen 13 sekuntia. [5]



### 3.3.2. Valvontalohko

Valvontalohko vastaa GPS-järjestelmän toiminnasta ja sen muodostavat maanpäälliset valvonta-asetat, jotka vastaanottavat satelliittien lähettämää signaalia ja valvovat satelliitteja mahdollisten vikatilanteiden varalta. Valvonta-asetat sijaitsevat ympäri maapalloa seuraavissa paikoissa: Colorado Springs, Hawaiji, Florida, Ascension Islands (eteläinen Atlantti), Diego Garcia (Intian valtameri) ja Kwajalein (pohjoinen Tyyni valtameri). [2]

Jokainen asema on varustettu cesium-pohjaisella aikastandardilla ja ne mittaavat satelliittien lähettämää signaalia. Mittauksissa otetaan huomioon ionosfäärin ja sään tila. [17] Näillä muuttujilla korjatut mittaustulokset lähetetään keskusvalvonta-asetalle Colorado Springs:iin, jossa lasketaan ajan tasalla olevat navigointiviestit ladattaviksi edelleen takaisin satelliitteihin. [2]

### 3.3.3. Käyttäjälohko

Käyttäjälohko muodostuu GPS-vastaanottimista, jotka käyttävät järjestelmän tarjoamia palveluja. Vastaanottimien markkinoilla oleva tarjonta on jo nykyisellään varsin laaja. Vastaanottimien ominaisuuksissa on eroja suunnitellusta käyttötarkoituksesta tai sovel-luskohteesta riippuen. Vastaanottimien ominaisuudet on yleensä suunniteltu tiettyä käyttötarkoitusta, esimerkiksi navigointia, paikannusta tai ajastusta silmällä pitäen, mutta tarjolla on myös monipuolisia yhdistelmälaitteita.

Keskitasen vastaanottimessa on tavallisesti yksi tai useampi liityntä, joiden välityksellä paikka- ja aikatietoja voidaan lukea suoraan tietokoneen sarjaporttiin. Aikasynkro-nointiin soveltuvissa vastaanotinmalleissa on lisänä myös PPS-signaalin (PPS, Pulse per Second) ulosotto, jota voidaan käyttää ajastus- ja synkronointisovelluksissa tarkkuuden parantamiseen. PPS-signaali on tavallisesti TTL (Transistor-to-Transistor Logic) -tasoinen ja sidottu aikaan siten, että signaalitason muutos liipaistaan UTC-sekunnin vaihtuessa.

### 3.3.3.1. *Normaali navigointijärjestelmä*

GPS-järjestelmä on suunniteltu Yhdysvaltojen ja sen liittolaisten sotilaalliseen käyttöön. Tämän takia siviilikäyttäjille tarjottavan paikannuspalvelun tarkkuutta voidaan tarvittaessa huonontaa tahallisella heikennyksellä (SA, Selective Availability). Tahallisella heikennyksellä voidaan heikentää normaalia navigointijärjestelmää (SPS, Standard Positioning System), joka pohjautuu satelliitin lähettämään C/A-koodiin (Coarse / Acquisition). [2]

Tahallinen heikennys toteutetaan sekoittamalla satelliitin kellon taajuutta siten, että käyttäjä ei voi tarkasti tulkita satelliitin lähettämää koodia. Tämä vaikuttaa pääasiassa yhdellä vastaanottimella toteutettuun navigointiin. Tahallinen heikennys voidaan toteuttaa myös lyhentämällä satelliittien lähettämiä navigointiviestejä siten, että satelliitin sijainnin ilmaisevien koordinaattien tarkkuus heikkenee. [17]

Tahallisen heikennyksen vaikutusta voidaan minimoida käyttämällä differentiaalista GPS-vastaanottoa (*luku 3.5.*) tai tarkkailemalla varsinaisen välitettävän informaation lisäksi myös kantaallon vaihetta. Molempien yhdistelmää kutsutaan interferometriseksi GPS-paikannukseksi. Tällä hetkellä tahallista heikennystä ei käytetä, mutta se voidaan tarvittaessa ottaa käyttöön. Normaali navigointijärjestelmä mahdollistaa vaakasuunnassa 100 metrin ja pystysuunnassa 156 metrin paikannustarkkuuden 95 % ajasta. Järjestelmä välittämä aika on 340 nano-sekunnin tarkkuudella UTC-ajassa (95 % ajasta). [5]

### 3.3.3.2. *Tarkkuusnavigointijärjestelmä*

Tarkkuusnavigointijärjestelmä (PPS, Precise Positioning System) pohjautuu satelliittien lähettämään P-koodiin (Precision) tai sen salattuun versioon eli Y-koodiin, jota siviilikäyttäjät eivät pysty hyödyntämään [17]. P-koodi on salaamattomana täysin siviilikäyttäjien hyödynnettävissä.

P-koodin salaamisesta käytetään nimitystä AS (Anti-Spoofing). Tällä rajoitetaan P-koodin käyttöä ja estetään mahdollinen P-koodin tahallinen häiritseminen, koska oletettu häiriölähde ei kykene muodostamaan oikeankaltaista salattua Y-koodia. [2]

Tarkkuusnavigointijärjestelmä mahdollistaa vaakasuunnassa 22 metrin ja pystysuunnassa 27,7 metrin paikannustarkkuuden (95 % ajasta). Välitettävä aika on 200 nanosekunnin tarkkuudella UTC-ajassa (95 % ajasta). [5]

### 3.4. Informaation välitys

Molempiin satelliittien lähettämiin kantaaltoihiin on PRN-koodien lisäksi moduloitu navigointiviesti, joka sisältää mm. tiedot GPS-ajasta, satelliitin kellokorjauksista, rata-tiedoista ja tilasta. [18]

#### 3.4.1. Radioyhteys

Satelliittien ja vastaanottimen välisessä radioyhteydessä käytetään kahta kantaaltoaajuutta. Käytetyt taajuudet on johdettu 10,23 MHz perustaajuudesta  $f_0$ . Kantaaltoaajuus  $L_1$  1575,42 MHz saadaan kertomalla perustaajuutta 154:llä. Kantaaltoaajuus  $L_2$  1227,60 MHz saadaan vastaavasti kertomalla perustaajuutta 120:llä. [2]

Radioyhteydellä käytetyt kantaaltoaajuudet ovat jokaisessa satelliitissa täysin samat. Satelliittien lähettämä signaali kanavoidaan koodijakoisella monipääsymenetelmällä (CDMA, Code Division Multiple Access), jossa jokaisella satelliitilla on toisista satelliiteista olennaisesti poikkeava, ortogonaalinen PRN-koodi, jolla se moduloi kantaaltoa. [2]

Navigointiviesti yhdistetään edellä mainittuun näennäissatunnaiseen PRN-koodiin käyttämällä modulo-2-yhteenlaskua. Tällöin navigointiviestin sisältämä 0-bitti ei aiheuta muutosta PRN-koodin bitteihin, mutta 1-bitti kääntää kaikki PRN-koodin bitit vastakkaisiksi. Tällöin navigointiviestin vaatima kaistanleveys on paljon suurempi kuin varsinaisen kantataajuuden informaation. [2]

Tästä menetelmästä käytetään nimitystä hajaspektritekniikka. Tekniikan ansiosta signaalin teho jakaantuu laajemmalle taajuuskaistalle kuin alkuperäinen informaatio. Käytännössä hajaspektrisignaali peittyy lähes kokonaan taustakohinan alle. Vastaanotin generoi satelliitin koodin kanssa identtisen koodin ja laskee korrelaatiota vastaanottamansa ”kohinan” ja generoimansa koodin välillä. [2]

Kun koodit aikasiirron seurauksena asettuvat aika-alueessa kohdakkain, saa korrelaatio suurimman arvonsa. Tämä johtuu siitä, että taustakohinan keskiarvo on pitkällä aikavälillä nolla ja korrelaatio on hyvin lähellä nollaa muodostetun koodin ja taustakohinan välillä. Menetelmän ansiosta järjestelmä sietää hyvin häiriöitä ja häirintää. Lisäksi voidaan käyttää alhaista lähetystehoa, koska signaalin tehotason jääminen taustakohinaa pienemmäksi ei vaikuta järjestelmän suorituskykyyn. [2]

### 3.4.2. Näennäissatunnaiset koodit

C/A-koodin pääasiallinen käyttötarkoitus on mahdollistaa varsinaisessa kulkuajamittauksessa käytettävän P-koodin saatavuus [5]. C/A-koodin modulointinopeus on 1,023 Mbit/s ja sen jakso toistuu millisekunnin välein koostuen 1023 bitistä. Käytetyillä C/A-koodeilla on matala ristikorrelaatio, mikä tekee eri satelliittien lähettämien koodien erottamisen helpoksi. P(Y)-koodin modulointinopeus vastaa perustaajuutta  $f_0$  ja yhden satelliitin osuus koodista on 7 vuorokauden mittainen [5]. [2]

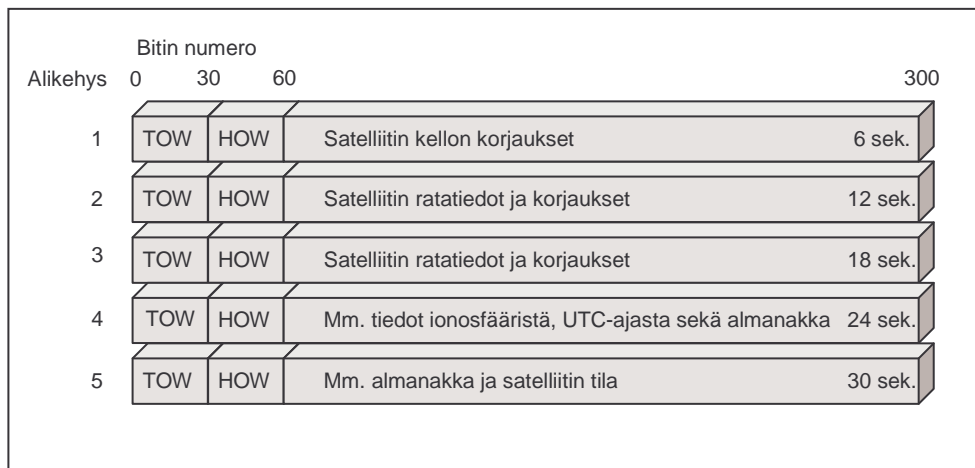
Täysimittainen P-koodi ei toista itseään noin 267 päivän aikana, jolloin satelliiteille on mahdollista määrätä oma viikoittainen osuus käytettävästä P-koodista siten, että eri satelliittien lähettämät koodit voidaan erottaa toisistaan. Kaikki koodit alustetaan keskiyöllä kerran GPS-viikon aikana, minkä seurauksena viikko on GPS-järjestelmän perusaikayksikkö. [2]

Navigointiviesti ja P(Y)-koodi moduloidaan kanta-aaltoon  $L_2$  PSK-modulaatiolla (Phase Shift Keying), jossa informaation bittien muuttuminen nollostä ykköseksi tai päinvastoin kääntää kanta-aallon vaihetta 180 astetta. Kanta-aaltoa  $L_1$  moduloidaan kuitenkin kaikilla kolmella koodilla (navigointiviesti, C/A-koodi, P(Y)-koodi). Modulointi suoritetaan QPSK-menetelmällä (Quadrature Phase Shift Keying), jossa kanta-aalto jaetaan kahteen komponenttiin, joista toista viivästetään 90 astetta. Tällöin kaikki tarvittava informaation saadaan välitetyksi moduloimalla kahden komponentin vaihetta. Y-koodi välitetään signaalin Q-haarassa. [2]

Viikoittaisen näennäissatunnaisen koodin järjestysnumero on eräs tapa numeroida satelliitit. Esimerkiksi PRN 12 viittaa satelliittiin, joka lähettää näennäissatunnaisen koodin kahdettatoista osaa. Toinen tapa on numeroida satelliitit laukaisunumeron perusteella. Tällöin PRN 12 vastaan arvoa SVN 10 (Space Vehicle launch Number). [2]

### 3.4.3. Navigointiviestin rakenne

Navigointiviesti moduloidaan molempiin kanta-aaltoihin nopeudella 50 bit/s. Täydellinen navigointiviesti koostuu 25 kehyksestä. Yhden kehyksen pituus on 1500 bittiä, joten koko navigointiviestin siirtoon kuluu aikaa 12,5 minuuttia. Kukin 25 kehyksestä on edelleen jaettu viiteen 300 bitin mittaiseen alikehykseen kuvan 3.3. mukaisesti. [2]



Kuva 3.3. Navigointiviestin rakenne. [2]

Jokainen alikehys alkaa 30 bitin mittaisella TLM-sanalla (Telemetry Word), joka sisältää kohdistusosan (preamble) sekä joitakin tietoja järjestelmänvalvojalle. TLM-sanaa seuraa samanmittainen HOW-sana (Handover Word), joka muodostuu viikon alusta laskettujen sekuntien kokonaismäärän lyhennetystä esitysmuodosta (TOW, Time of Week) GPS-aikaan sidottuna. [2]

Ensimmäisen alikehyksen loppuosa sisältää GPS-viikon numeron sekä tiedot satelliitin kellon korjaustermeistä. Alikehykset 2 ja 3 sisältävät lähettävän satelliitin rataan liittyviä tietoja ja korjauksia. Alikehys 4 sisältää tiedot ionosfäärin vaikutuksen korjaustermeistä, kertoimista GPS-ajan muuttamisesta UTC-ajaksi sekä almanakan satelliiteille, joiden SVN-numero on suurempi kuin 24. Vastaavasti alikehys 5 sisältää almanakan alkupään satelliiteille sekä lähettävän satelliitin tilatiedot. [2]

Edellä mainittu almanakka sisältää tiedot, joita tarvitaan muiden kuin tietoja lähettävän satelliitin sijainnin laskemiseen. Almanakan sisältö on käytännössä lyhennelmä aliehyksen 1-3 sisältämästä informaatiosta. Almanakka sisältää mm. tiedot satelliitin sijainnin määrittämiseksi ja kertoimet, joilla satelliitin ilmoittama aika muunnetaan GPS-ajaksi. Sisältönä on lisäksi muita satelliitin kiertorataan liittyviä parametreja ja korjauksia. [2]

#### 3.4.4. Ajan ylläpito ja välitys

Satelliitit lähettävät PRN-koodeja sekä navigointiviestiä omaan aikaansa sitoen. Navigointiviestin sisältämä data on kuitenkin sidottu suoraan GPS-aikaan valvontalohkon toimesta. GPS-aika on puolestaan sidottu UTC (USNO) –aikaan mikrosekunnin tarkkuudella. GPS-järjestelmässä suurin yksikkö ajan ylläpitoon on viikko, joka koostuu 604 800 sekunnista. Absoluuttinen aika välitetään navigointiviestin Week Number – ja Time of Week –arvojen avulla. [2]

Satelliittien kelloissa tulee lisäksi huomioida satelliitin nopeudesta johtuva ajan nopeutuminen. Tämän takia satelliittien perustajuudeksi asetetaan ennen laukaisua 10,22999999543 MHz, mikä vastaa kellon hidastamista 38,3  $\mu$ s vuorokaudessa. Lisäksi kelloissa tulee huomioida kiertoradan maakeskeisyys. Koska satelliitit eivät sijaitse täysin pyöreillä kiertoradoilla, voi virhettä pahimmillaan aiheutua 45 ns, mikä vastaa 14 metrin virhettä paikannuksessa. [2]

Lukittuessaan C/A-koodiin vastaanotin selvittää ensimmäisenä navigointiviestin aliehyksen HOW-sanan. Sen perusteella voidaan yksiselitteisesti päätellä mitä osaa P-koodista käytetään, koska HOW-sanan sisältämä arvo vastaa P-koodin generoinnissa käytettävän rekisterin arvoa. Rekisterin arvo vuotaa yli 1,5 minuutin välein, mikä vastaa navigointiviestin yhden aliehyksen kestoa. Näin ollen vastaanotin voi heti seuraavan aliehyksen alkaessa lukittua P-koodiin. [2]

### 3.5. Differentiaalinen GPS-paikannus

Yhdellä vastaanottimella suoritettavaa paikannusta kutsutaan absoluuttiseksi GPS-paikannukseksi. Differentiaalisessa GPS-paikannuksessa (DGPS) satelliittien lähettämää signaalia mitataan kahdella vastaanottimella samanaikaisesti, joista toinen on paikallaan ja toinen liikkuu sen ympäristössä. [16]

Menetelmän avulla saavutettava tarkkuus on parhaimmillaan alle metrin suuruusluokkaa. Vielä parempaan tarkkuuteen päästään interferometrisella GPS-paikannuksella, jossa käytetään hyväksi myös kantoaaltojen vaihetta.[18]

Koska GPS-satelliitit ovat kaukana avaruudessa, ei vastaanottimien välinen muutamien kymmenien tai satojen kilometrien etäisyys oleellisesti muuta satelliitista vastaanotetun signaalin kulkemaa matkaa. Tällöin voidaan hyvällä tarkkuudella olettaa, että molempien vastaanottimien signaaleissa on sama virhe. [16]

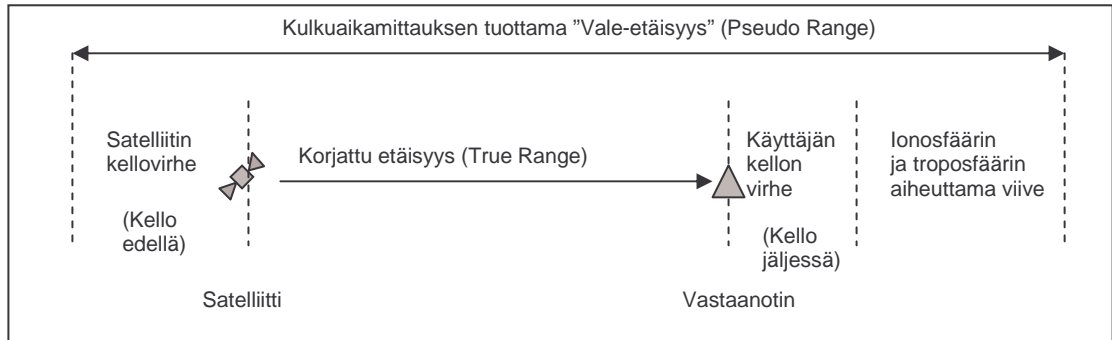
Tunnetuissa koordinaateissa sijaitseva vastaanotin ei määritä ajastuksen perusteella omaa sijaintiaan vaan käyttää omaa sijaintiaan laskeakseen ajastuksen. Referenssivastaanotin voi päätellä, paljonko signaalin viive pitäisi olla ja laskee paljonko vastaanotetussa signaalissa on virhettä. [16]

Referenssivastaanotin lähettää saadun korjauksen lähistöllä liikkuvalla vastaanottimelle, joka voi näin ollen korjata omaa sijaintiaan saadulla korjauksella. Koska liikkuva vastaanotin ei voi tietää mitä satelliittia referenssivastaanotin parhaillaan ajanhetkellä käyttää, referenssivastaanottimen on määritettävä korjaukset kaikille saatavilla oleville satelliiteille. [16]

### 3.6. Virhetekijät ja tarkkuus

Virheiden vaikutusta kulkuajamittauksessa on havainnollistettu kuvassa 3.4. Satelliitin kello on GPS-ajasta hieman jäljessä ja käyttäjän kello vastaavasti GPS-aikaa edellä. Kulkuajamittauksen tulosta voi lisäksi vääristyä myös ionosfäärin aiheuttaman signaalin hidastumisen seurauksena. Ajastuksen virhe voidaan kuitenkin päätellä matemaattisesti. [15]

Suurimman virheen GPS-järjestelmän paikannustarkkuuteen aiheuttaa tahallinen heikennys (SA). Sen aiheuttama paikannusvirhe on noin 15-35 metriä tahallisen heikennyksen ollessa aktivoituna. Tahallisen heikennyksen vaikutusta on mahdollista kompensoida käyttämällä differentiaalista GPS-paikannusta, mutta korjaus pätee tällöin vain rajoitetulla alueella. [18]



Kuva 3.4. Virheiden vaikutus kulkuajamittaukseen. [15]

Tahallinen heikennys poisluettuna merkittävimmän virheen aiheuttaa maan ionosfääri, joka sijaitsee noin 80-400 km korkeudella maan pinnasta ja koostuu sähköisesti varatuista hiukkasista. Ionosfääri vaikuttaa satelliittien lähettämän signaalin kulku-aikaan. Radiokantoaallolla kulkevan PRN-koodin hidastuminen aiheuttaa virhettä kulkuajamittaukseen ja edelleen etäisyyden määrittämiseen satelliitin ja vastaanottimen välillä. [15]

Ionosfäärin vaikutusta on mahdollista arvioida kantoaaltojen  $L_1$  ja  $L_2$  avulla, koska signaalin hidastuminen on kääntäen verrannollinen signaalin taajuuteen. Tämä on kuitenkin mahdollista vain molemmille kantoaaltoille moduloitua P-koodia vastaanottavissa sovelluksissa. [15]

Ionosfäärin jälkeen signaali läpäisee troposfäärin, jossa varsinaiset sääilmiöt tapahtuvat. Ilmakehän sisältämän veden seurauksena signaalin tehoa absorboituu vesipisaroihin. Etenevän signaalin voimakkuus pienenee, koska vesipisarot sirottavat absorboimansa tehon koko avaruuskulmaan. Troposfäärin vaikutus on huomattava, mikäli satelliitti on matalassa korkeuskulmassa eli signaali kulkee pitkän matkan troposfäärissä. [15]

Satelliitin kello- ja ratavirheet vaikuttavat osaltaan järjestelmän tarkkuuteen. Toisaalta vastaanotinpuolella sähköinen kohina voi aiheuttaa PRN-koodien virheellisen korrelaation ja vääristää kulkuajamittauksia. Vastaanotin saattaa myös tehdä jonkin matemaattisen virheen, esimerkiksi pyöristettäessä laskutoimituksesta saatua tulosta. [15]



Ongelmallista on myös monitie-eteneminen, jonka seurauksena signaali heijastuu jostakin kohteesta ennen kuin se saavuttaa vastaanottimen. Tämän seurauksena mitattu kulku-aika on todellista pidempi. Monitie-etenemisestä aiheutuvaa virhettä voidaan minimoida erikoisantennien tai erityisten ohjelmistojen avulla. [15]

Satelliittien keskinäinen sijainti eli satelliittigeometria vaikuttaa järjestelmän paikannustarkkuuteen, koska satelliittien välisen kulman pienentyessä satelliiteista mitattujen etäisyyksien leikkauskuvio suurenee ja näin ollen alue, jolla vastaanotin voi sijaita, kasvaa suuremmaksi. Satelliittigeometriaa kuvataan DOP-luvuilla, joista käytetyin on PDOP (Position Dilution of Precision). Mitä pienempi DOP-luku sitä kauempana satelliitit ovat toisistaan ja sitä parempi on paikannustarkkuus. [18]

**Taulukko 3.2. Virhelähteitä sekä niiden aiheuttaman virheen tyypillisiä arvoja. [18]**

Virhetekijä	Etäisyysvirhe	Aikavirhe
Satelliitin kellovirheet	1 m	3,34 ns
Satelliitin ratavirheet	1 m	3,34 ns
Vastaanottimen kohina	2 m	6,67 ns
Ilmakehän aiheuttama viive	4 m	13,34 ns
Tahallinen heikennys	10 - 30 m	33,36 - 100,07 ns
Etäisyyden kokonaisvirhe	15 - 35 m	50,03 - 116,75 ns

Taulukossa 3.2 on lueteltu GPS-järjestelmän virhelähteitä sekä niiden aiheuttaman virheen tyypillisiä arvoja. Signaalin tahallinen heikennys on ylivoimaisesti suurin virhelähde. Yllämainitut virhelähteet vaikuttavat erikseen kuhunkin satelliittiin mitattavan kulkuajan ja etäisyyden tarkkuuteen. Satelliittigeometria (PDOP) huomioon otettuna (tyypillinen arvo on 2-6) saadaan lopulliseksi paikannustarkkuudeksi 30-150 metriä kertomalla taulukon 3.2. mukainen kokonaisvirhe PDOP-arvolla [18]. Virallisen määritelmän mukaan tarkkuus vaakasuunnassa on alle 100 metriä 95 % ajasta [2].

Käytännössä absoluuttisen paikannusmenetelmän tarkkuus on kuitenkin huomattavasti tätä parempi. Differentiaalisen GPS-paikannuksen avulla saavutettava tarkkuus on parhaimmillaan alle metrin suuruusluokkaa. Interferometrisella GPS-paikannuksella voidaan saavuttaa vielä suurempia tarkkuuksia. [18]

## 4. Ajan hallinta ja ajoitus PC-tietokoneessa

### 4.1. Ajan ylläpidon toteutus

Ajan ja ajoituksen piiritason toteutus tietokoneissa on vahvasti arkkitehtuuri- ja valmistajakohtainen, joten yleispätevän kuvauksen antaminen on tämän työn laajuuden ulkopuolella. Kun siirrytään tarkastelemaan tilannetta piiritasolta käyttöjärjestelmätasolle, on ajan hallinta lisäksi vahvasti käyttöjärjestelmäriippuvainen.

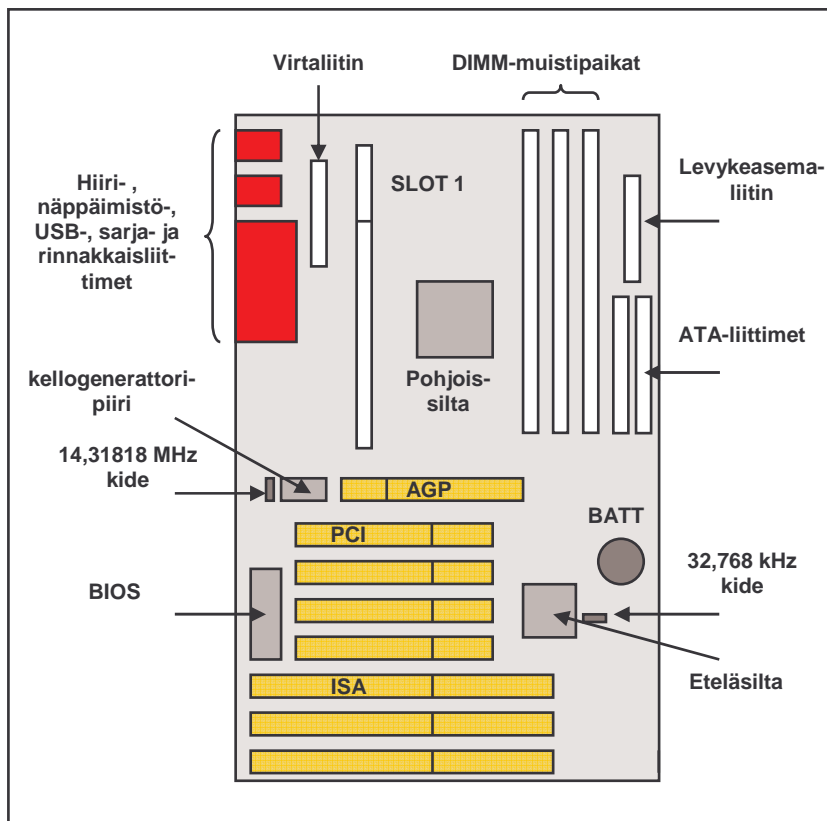
On kuitenkin hyvin havainnollista käydä läpi periaatteellisella tasolla aikaan ja ajastukseen liittyvät toiminnot edes yhdessä näistä lukuisista vaihtoehdoista. Tähän työhön liittyvien ohjelmien ja laitteiden testausvaiheessa käytettiin pääsääntöisesti Intel Celeron –prosessorilla (300 MHz) varustettua tietokonetta. Prosessorityyppi edustaa varsin hyvin myös varsinaisen testiverkon laitteiden keskitasoa, joten on mielekästä valita tarkastelun kohteeksi tämän tyyppinen arkkitehtuuri.

#### 4.1.1. Piiritason yleiskuvaus

Kuvassa 4.1. on esitetty Intel Celeron –yhteensopivan ja Micro-Star Internationalin (MSI) valmistaman MS-6119 –emolevyn (ATX) piiritason asettelu. ATX-tyypin emolevy sallii virrankatkaisun ohjelmallisesti käyttöjärjestelmän avulla, kun taas vanhemmissa emolevyissä virran katkaisu suoritettiin manuaalisella kytkimellä [19].

Emolevyille on sijoitettu omat liittimensä hiirelle, näppäimistölle sekä USB- (Universal Serial Bus), sarja- ja rinnakkaisliittimille. Laajennusväylät AGP (Accelerated Graphics Port), PCI (Peripheral Component Interconnect) ja ISA (Industry Standard Architecture) mahdollistavat esimerkiksi näytönohjaimen ja verkko- tai äänikortin liittämisen tietokoneeseen.

DIMM (Dual In-line Memory Module) – muistipaikat on tarkoitettu keskusmuistia (SDRAM, Synchronous dynamic random access memory) varten. Muistipaikkojen vieressä on liitin levykeasemalle sekä ATA (Advanced Technology Attachment) -liitin kovalevyille ja esimerkiksi CDROM-asemalle (Compact Disc Read Only Memory). ATA-liitynnästä käytetään myös IDE (Integrated Drive Electronics) -nimitystä.

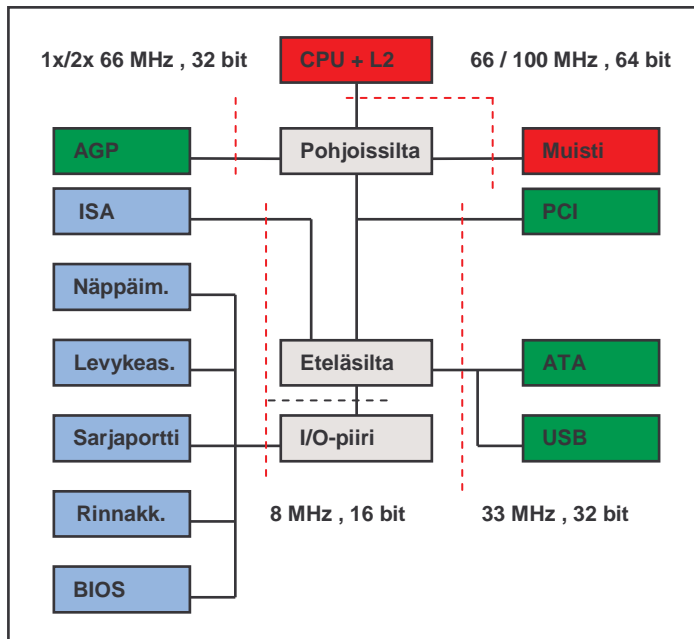


Kuva 4.1. MSI MS-6119 –emolevyn piiritason asettelu. [20]

Slot 1 on 66 MHz tai 100 MHz kellotaajuudella toimiva väylä, joka on tarkoitettu prosessorin (CPU, Central Processing Unit) liittämiseksi emolevyyn. Sen oikealla puolella sijaitsee emolevyn piirisarjan pohjoissilta. Piirisarjaryhmään kuuluu tässä tapauksessa kaksi piiriä, pohjois- ja eteläsilta. [20]

Pohjoissilta huolehtii tietoliikenteestä prosessorin, muistin sekä PCI- ja AGP-väylien välillä. Eteläsilta vastaa PCI- ja ISA-väylien sekä lukuisten muiden liitännöiden välisestä tiedonsiirrosta kuvan 4.2. mukaisesti. Samalle piirille on integroitu myös CMOS (Complementary Metal-Oxide Semiconductor) –muisti sekä reaaliaikakello (RTC, Real-Time Clock, ts. Hardware Clock), jotka on varmennettu 3 V paristolla. [19]

Paristo varmentaa CMOS-muistin tietojen ja reaaliaikakellon ajan säilymisen, jos tietokoneen virtalähde on irrotettu kokonaan sähköverkosta. Muussa tapauksessa ATX-virtalähde syöttää ylläpitoon tarvittavan jännitteen emolevylle virtaliittimen johtimen 9 (+ 5V StandBy, violetti) avulla, vaikka tietokone olisikin sammutettuna. [19]



Kuva 4.2. Pentium II –yhteensopivan prosessorin piirisarjaesimerkki. [19]

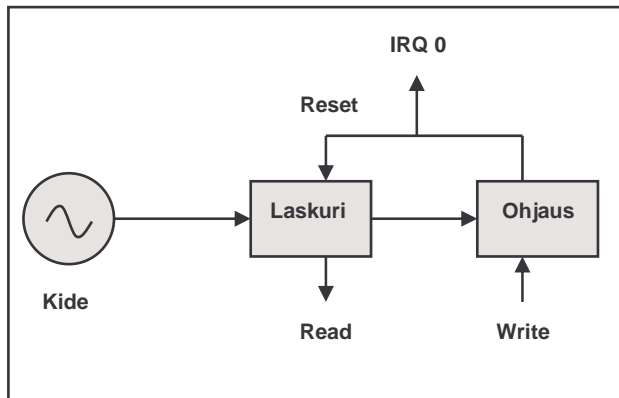
CMOS-muistilla säilytetään BIOS:n (Basic Input Output System) avulla muokattavia tietokoneen perusasetuksia. BIOS on toteutettu yhdellä piirillä, joka huolehtii myös tietokoneen käynnistyksestä ja perusrutiinien suorittamisesta käytön aikana. [19]

#### 4.1.1.1. Ajan ja ajastuksen toteutus piiritasolla

Emolevyllä on kaksi komponenttia, jotka mahdollistavat ajan ylläpidon, reaaliaikakello ja ajastin. Näiden komponenttien ominaisuudet ja implementointi vaihtelevat laite- ja valmistajakohtaisesti, mutta niiden pääpiirteinen käyttötarkoitus on hyvin samankaltainen.

Reaaliaikakello perustuu 32,768 kHz ulkoiseen kiteeseen, jonka perusteella se ylläpitää sekunnin tarkkuudella tietoa ajasta ja päivämäärästä viikonpäivä mukaan luetuna [21]. Reaaliaikakello vastaa ajan ylläpidosta erityisesti tietokoneen ollessa sammutettuna, mutta suorittaa samoja toimenpiteitä myös tietokoneen käytön aikana [19]. Reaaliaikakellon asetuksia voi muuttaa BIOS:sta käsin. Jotkut käyttöjärjestelmät mahdollistavat reaaliaikakellon tietojen lukemisen ja kirjoittamisen yksinkertaisten komentojen avulla.

Eteläsilta sisältää myös integroidun ajastimen (System Timer). Se on ohjelmoitava ajastin, jonka avulla voidaan generoida esimerkiksi käyttöjärjestelmän ylläpitämän systeemikellon (System Clock) päivittämiseen tarvittavat säännölliset keskeytykset osoitteeseen IRQ 0. Ajastin tarvitsee toimiakseen 14,31818 MHz signaalin, joka saadaan kellogeneraattoriin vieressä emolevyllä sijaitsevasta kiteestä. [21] [22]



Kuva 4.3. Ajastimen toimintaperiaate. [25]

Kuvan 4.3. mukainen ajastin rakentuu laskurista, joka laskee kiteen muodostaman signaalin jaksoja. Ohjausyksikkö nolaa laskurin ja aiheuttaa keskeytyksen, kun laskuri saavuttaa ohjausyksikköön ohjelmoidun arvon. [22]

IRQ (Interrupt Request) on yksi tietokoneessa käytettävistä keskeytysmenetelmistä. Keskeytys toimii siten, että kun laite haluaa emolevyn palvelevan sitä, se ilmoittaa tästä keskeytyssignaalin. Keskeytysten prioriteettijärjestys tarkoittaa sitä, missä järjestyksessä samaan aikaan saapuneita keskeytyksiä palvellaan. Näistä korkeimmalla prioriteetilla on kellokeskeytys IRQ 0 ja alhaisimmalla rinnakkaisportti (LPT) IRQ 7. Muita keskeytyksiä ovat esimerkiksi IRQ 1 (näppäimistö), IRQ 4 (COM 1 tai COM 4 -sarjaportti) sekä IRQ 6 (levykeasemaohjain). Keskeytysnumerointi ei ole yhtenevä prioriteettijärjestyksen kanssa. [19]

Pentium-yhteensopivissa emolevyissä emolevy määrää prosessorin sisäisen ja ulkoisen kellotaajuuden. Celeron- ja Pentium II – prosessoreissa sisäisen kellotaajuuden kerroin on kiinteä eikä sitä voi muuttaa emolevyn määritteillä. Prosessorin ulkoinen kellotaajuus asetetaan BIOS:n Setup-ohjelmalla tai emolevyn valitsimilla. [19]

Kaikki emolevyn toimintaan liittyvät kellosignaalit generoidaan 14,31818 MHz taajuudesta kellogeneraattoriin avulla. Muunnoksessa käytetään hyväksi vaihelukitun silmukan kykyä toimia taajuuskertojana tai –jakajana (kts. 2.4.2). Kellogeneraattorin avulla muodostetaan väylien ja muistien toimintaan tarvittavat kellotaajuudet. Useimmat laajennuspaikkoihin liitettävät kortit sisältävät myös omat kiteensä, joiden avulla muodostetaan korttien sisäisiin toimintoihin tarvittavat kellosignaalit.

#### *4.1.2. Käyttöjärjestelmän toimenpiteet ajan ylläpidossa*

Tässä luvussa tarkastellaan ajan ja ajoituksen hallintaa Linux-käyttöjärjestelmässä (kernel-versiot 2.4.20 ja 2.4.20-NANO). Kernel-version NANO-liite tarkoittaa, että kernel (käyttöjärjestelmän toiminnallinen ydin) on päivitetty (patch) PPSkit-paketilla, joka muun muassa parantaa systeemikellon ylläpitotarkkuuden mikrosekunneista nanosekunneihin ja mahdollistaa sarjaporttiin kytkettävän PPS-signaalin käyttämisen synkronoinnissa. Lisäksi se tarjoaa uusia systeemikutsuja ajan hallintaan ja korjaamiseen. Jäljempänä kuvatut toimenpiteet ovat pääpiirteittäin yhteneviä muiden Unix-tyyppisten käyttöjärjestelmien kanssa, mutta kuvausta ei voida suoraan yleistää mille tahansa käyttöjärjestelmälle.

Käyttöjärjestelmän toiminnallisen ytimen kannalta voidaan erottaa suhteellisen ja absoluuttisen ajan mittaaminen ja määrittäminen [23]. Suhteellista aikaa tarvitaan esimerkiksi prosessien keston määrittämiseksi. Prosessille voidaan varata prosessoriaikaa esimerkiksi 5 ms tästä hetkestä eteenpäin. Kellonajan ylläpito tai prosessin käynnistäminen kauempana tulevaisuudessa vaatii kerneliltä kyvyn mitata absoluuttista aikaa ja ajan kulumista [23].

Säännöllisesti esiintyviä tapahtumia ohjataan ajastimen (System Timer) avulla. Sen aiheuttamia ja piiritasolta saapuvia keskeytyssignaaleja prosessori kernelissä keskeytyskäsittelijä (Timer Interrupt). Tämän lisäksi kernel voi asettaa dynaamisia ajastimia yhden kerran esiintyvien tapahtumien vuorotteluun (scheduling). [23]

Ajastimen aiheuttamien keskeytysten esiintymistiheyttä nimitetään tahdiksi (Tick Rate). Se ohjelmoidaan emolevyn ajastimeen käyttöjärjestelmän käynnistyksen yhteydessä. Koska kernel tietää ajastimeen ohjelmoidun tahdin, se tietää edelleen peräkkäisten kellokeskeytysten välisen ajan, jota nimitetään lyönniksi (Tick). [23]

Käytetty tahti määritellään *include/asm/param.h*-otsikkotiedostossa:

```
#define HZ 100
```

Määrittelyn mukaisesti ajastin aiheuttaa keskeytyksen 100 kertaa sekunnissa eli 10 ms välein. Toinen arkkitehtuureissa suosittu ajastimen tahti on 1000 Hz. Saapuneiden keskeytysten kokonaismäärä käynnistyksestä lähtien saadaan selville esimerkiksi */proc/interrupts*-tiedostosta:

```
linux:/ # cat /proc/interrupts
CPU0
 0:    6853661          XT-PIC  timer
 1:         13          XT-PIC  keyboard
 2:          0          XT-PIC  cascade
 4:   14526057          XT-PIC  serial
10:    313073          XT-PIC  eth1
12:   1716435          XT-PIC  usb-uhci, eth0
14:   258459          XT-PIC  ide0
NMI:          0
LOC:   6853624
ERR:          0
MIS:          0
linux:/ #
```

Toistuva tarkastelu osoittaa, että kellokeskeytysten (timer) lukumäärä kasvaa sadalla sekuntia kohti.

Ajastimen tahdin kasvattaminen parantaa ajastimen avulla suoritettavien toimenpiteiden resoluutiota ja tarkkuutta. Näistä toimenpiteistä merkittävin on prosessien vuorottelu. Sadan hertzin tahtia käytettäessä ajastimen avulla mitattava pienin aikaväli on 10 ms ja tämän seurauksena aikavälin pituudessa on  $\pm 5$  ms virhettä johtuen mittauksen aloitushetken satunnaisuudesta. [23]

Tahdin kasvattaminen arvoon 1000 Hz pienentää edellä mainitut arvot kymmenesosaan, mutta kuormittaa enemmän prosessoria. Mitä suurempi on ajastimen tahti, sitä enemmän prosessoriaikaa kuluu kellokeskeytysten käsittelyyn. [23]

Prossessorin kellojaksolaskuri on tarkoitettu tarkkaan ajan mittaamiseen. Kellojaksolaskurin resoluutio vastaa prosessorin kellojakson pituutta. Sen päivittäminen hoidetaan piiratasolla ja arvoa ylläpidetään 64-bitin tarkkuudella. Näistä 32 vähitenmerkittävää bittiä on luettavissa *get\_cycles()*-systeemikutsun avulla, joka määritellään *include/asm/timex.h*-otsikkotiedostossa.

#### 4.1.2.1. *Systeemikello*

Systeemikello muodostuu kahdesta muuttujasta, jotka sisältyvät *timeval*- tai *timespec*-tyyppiseen *xtime*-tietueeseen. Tietueen *tv\_sec*-muuttuja sisältää sekuntien kokonaismäärän 01.01.1970 lähtien. Tietueen *tv\_usec*- tai *tv\_nsec*-muuttuja sisältää kuluvan sekunnin murto-osan joko mikro- tai nanosekunneissa riippuen käytetystä kernel-versiosta.

Jokaisella keskeytyksellä sekunnin murto-osia ylläpitävän muuttujan arvoa kasvatetaan määrällä, jonka suuruus on ylläpidettävä resoluutio jaettuna keskeytystahdilla. Nanosekuntiresoluution tapauksessa yksittäinen korotus on siis  $1\ 000\ 000\ 000\ \text{ns} / 100\ \text{Hz}$  eli 10 ms. Mikäli muuttujassa tapahtuu ylivuoto, inkrementoidaan myös kokonaisia sekunteja yhdellä.

Käynnistyshetkellä kernel lukee reaaliaikakellon ajan ja varastoi sen *xtime*-tietueeseen. Tämän jälkeen käyttöjärjestelmä ei enää lue reaaliaikakelloa vaan päivittää tietueen arvoa kellokeskeytyksen avulla [23]. Synkronoitaessa systeemikello johonkin ulkoiseen referenssiin on hyödyllistä asettaa reaaliaikakellon arvoksi systeemikellon aika säännöllisin väliajoin, jotta myös reaaliaikakello pysyy oikeassa ajassa.

Systeemikellon ylläpitoon käytetty *xtime*-muuttuja määritellään *kernel/timer.c*-tiedostossa:

```
struct timeval xtime;
```

Käytetty tietue-tyyppi *timeval* määritellään *include/linux/time.h*-otsikkotiedostossa:

```
struct timeval {
    time_t          tv_sec;          /* seconds */
    suseconds_t    tv_usec;        /* microseconds */
};
```

Kernel-versio 2.4.20-NANO määrittelee *xtime*-muuttujan *kernel/time.c*-tiedostossa:

```
volatile struct timespec xtime;
```

*Volatile*-avainsana määrittää, että kyseisen muuttujan arvoa voidaan muuttaa muualta, joten kääntäjä ei voi optimoida sen käsittelyä.



Käytetty tietue-tyyppi *timespec* määritellään *include/linux/time.h*-otsikkotiedostossa:

```
struct timespec {
    time_t    tv_sec;           /* seconds */
    long     tv_nsec;         /* nanoseconds */
};
```

Kernel ylläpitää saapuneiden kellokeskeytysten lukumäärää *jiffies*-muuttujassa, joka määritellään *kernel/timer.c*-tiedostossa:

```
unsigned long volatile jiffies;
```

Kellokeskeytysten lukumäärää ylläpidetään 32-bittisessä kokonaislukutyypissä muuttujassa. Tällöin suurin keskeytysten määrä, joka voidaan esittää, on  $2^{32}-1$  eli 4 294 967 295. 100 Hz taajuudella ylivuoto tapahtuu noin 497 päivän välein. Ylivuodon hallitsemiseksi kernel tarjoaa neljä makroa, joilla voidaan havaita ylivuodon tapahtuminen [23].

Kun sovellusohjelma kysyy systeemikellon aikaa, on mahdollista estimoida edellisestä keskeytyksestä kulunut aika ekstrapoloimalla. Intelin 386 –arkkitehtuurissa ajastimen rekisterin arvo voidaan lukea suoraan emolevyltä, jolloin voidaan päästä alle mikrosekunnin resoluutioon. Myös Pentium-prosessorin kellojaksolaskuria voidaan käyttää ajan ekstrapoloinnissa. Sen resoluutio on huomattavasti parempi ja lukeminen voidaan suorittaa nopeammin kuin ajastinpiiriltä. [24]

## 4.2. NTP – Network Time Protocol

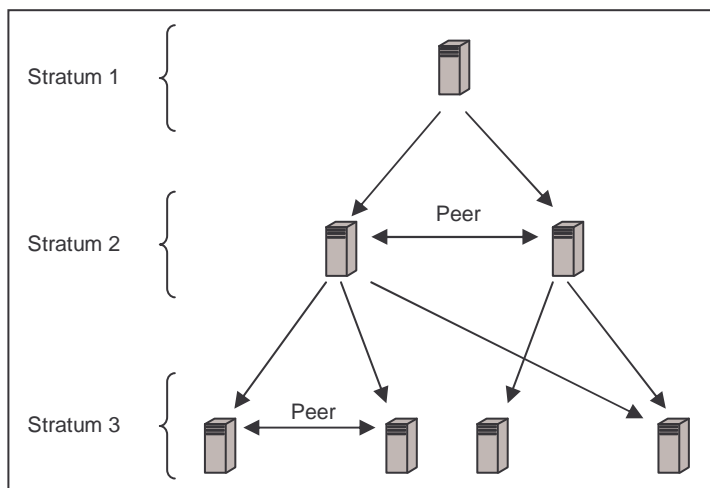
NTP (Network Time Protocol) on tarkoitettu kellojen synkronointiin tietoverkon välityksellä. Synkronoinnissa käytettävä hierarkia (kuva 4.4.) muodostuu ensisijaisista ja toissijaisista palvelimista sekä client-koneista. Hierarkiassa parhaan tarkkuuden omaavat ensisijaiset palvelimet (stratum 1) käyttävät aikareferenssinä taajuusstandardiasemaa, atomikelloa tai satelliitivastaanotinta. Toissijaiset palvelimet (stratum 2 - 15) synkronoituvat ensisijaisiin palvelimiin joko suoraan tai välillisesti toissijaisten palvelinten välityksellä. [25]

Palvelimissa toimiva NTP-prosessi vaihtaa aikaleimoja yhden tai useamman palvelimen kanssa UDP (Unicast Datagram protocol)- ja IP (Internet Protocol) -protokollien avulla määrittääkseen kellojen välisen poikkeaman. Saadut poikkeamat prosessoidaan useilla algoritmeilla ja niistä muodostetaan yhdistetty poikkeama, jota käytetään paikallisen kellon korjaamiseen. [26]

Synkronointiverkon kokonaistarkkuus suhteessa UTC-aikaan riippuu pääasiassa ensisijaisen synkronointilähteen (radio- tai satelliittivastaanotin) sekä ensisijaisen palvelimen paikallisen kellokiteen tarkkuudesta ja stabiiliudesta. Tyypillisen kellokiteen huono stabiilius on tarkkuutta rajoittava tekijä riippumatta siitä, onko ulkoinen referenssikello käytettävissä. [26]

On kuitenkin menetelmiä, joilla systeemikellon tarkkuutta ja stabiiliutta rajoittavat vain ulkoisen referenssikellon ominaisuudet sekä käyttöjärjestelmän aiheuttama värinä (jitter). NTP-versio 3:ssa referenssikellon tuottaman epätarkemman ASCII (American Standard Code for Information Interchange) -aikaleiman rinnalla voidaan käyttää PPS-signaalia tarkkuuden parantamiseksi mikrosekuntien suuruusluokkaan. [26]

#### 4.2.1. Synkronoinnin periaate



Kuva 4.4. Synkronointiverkon rakenne. [27]

NTP tuottaa aikaleimojen vaihdon tuloksena kolme parametria, jotka ovat kellopoikkeama, kiertoaikaviive (RTT, Round Trip Time) ja dispersio. Kellopoikkeama edustaa tarvittavaa korjausta, jolla paikallisen kellon ylläpitämä aika on yhtenevä referenssikellon ylläpitämän ajan kanssa. Dispersio kuvaa suurinta paikallisen kellon virhettä suhteessa referenssikelloon. [25]

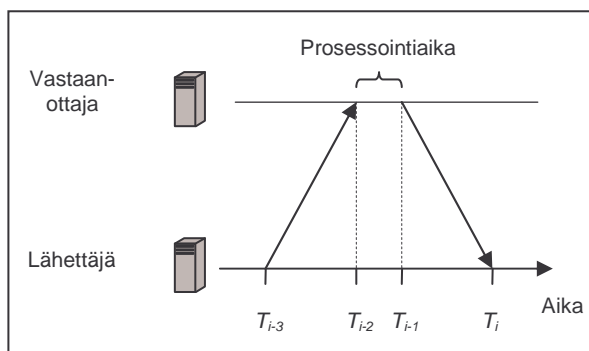
Palvelimien on mahdollista varmentaa yhteyksiä muodostamalla riippuvuuksia myös saman tason palvelinten kanssa kuvan 4.4. mukaisesti. Siirryttäessä hierarkiassa alaspäin riippuu saavutettavan synkronoinnin tarkkuus käytetyn yhteysvälin ja seuraavan hierarkiatason kellon hyväydestä. [25]

Synkronointiverkko ei koskaan saa muodostaa silmukkaa, koska tietyltä palvelimelta johdetun ajan käyttäminen uudelleen aiheuttaa virheiden kasautumisen. Silmukoiden ehkäisy varmistetaan valinta-algoritmin avulla. Käytetty algoritmi on muunneltu Bellman-Ford-algoritmista ja sen avulla lasketaan synkronointiverkolle pienimmän painon virityspuu. Laskennassa käytetään linkkipainona skaalatun tarkkuustason ja synkronointietäisyyden yhdistelmää. Synkronointietäisyys on puolestaan yhdistelmä dispersiosta ja absoluuttisesta viiveestä. [25]

Aikavertailun perustan muodostava aikaleima koostuu yhteensä 64 bitistä. Eniten merkitsevät 32 bittiä ilmaisevat sekunteja ja jälkimmäinen puolikas sekunnin osia suhteessa UTC-aikaan. Näin ollen aikaleiman mahdollistama ajan esitystarkkuus on 232 ps. [27]

Kun UTC-aika otettiin käyttöön 00:00 01.01.1970, NTP:n aikaleiman arvoksi asetettiin 2 272 060 800.0, mikä vastaa täysiä sekunteja 01.01.1900 alkaen ilman aiempia karkaussekunteja. Aikaleima kykenee yksikäsitteisesti esittämään 136 vuoden jakson, minkä seurauksena sen arvo ylivuotaa vuonna 2036. [27]

Aikaleima muodostetaan kopioimalla tietokoneen systeemikellon arvo aikaleimamuuttujaan. NTP-paketin avulla kyselyn lähettäjä voi arvioida käytetyn yhteysvälin kiertoaikaviiveen sekä kellojen välisen poikkeaman. NTP-paketti sisältää kolme aikaleimaa  $T_{i-1}$ ,  $T_{i-2}$  ja  $T_{i-3}$ . Neljäs aikaleima  $T_i$  määritetään paketin saapuessa takaisin vastaanottajalle kuvan 4.5. mukaisesti. [25]



Kuva 4.5. Kiertoaikaviiveen ja kellojen välisen poikkeaman estimointi. [27]

Kyselyn lähettäjä tallentaa lähetyshetken aikaleiman  $T_{i-3}$  NTP-paketin (kuva 4.6) *Originate Timestamp* -kenttään. Vastaanottaja tallentaa kyselyn vastaanottohetken *Receive Timestamp* -kenttään ja paketin takaisinlähetyshetken *Transmit Timestamp* -kenttään. [25]

Kun merkitään  $a = T_{i-2} - T_{i-3}$  ja  $b = T_{i-1} - T_i$ , yhteysvälin kiertoaikaviive saa muodon [27]:

$$\delta_i = a - b \quad (4.1)$$

ja kellojen välisen poikkeaman estimaatti on

$$\theta_i = \frac{a + b}{2} . \quad (4.2)$$

Kaavan 4.2. antama estimaatti on lähellä todellista poikkeamaa, jos käytetty yhteysväli on symmetrinen eli viive kumpaankin suuntaa on yhtä suuri. Mitä suurempi on yhteyden epäsymmetria, sitä enemmän estimaatti eroaa poikkeaman todellisesta arvosta. Kun otetaan huomioon yhteyden viiveen epäsymmetrisyydestä johtuvat virheet, voidaan todeta, että kellojen todellinen poikkeama  $\theta$  on varmuudella välillä [27]:

$$\theta_i - \frac{\delta_i}{2} \leq \theta \leq \theta_i + \frac{\delta_i}{2} . \quad (4.3)$$

NTP-paketin rakenne on esitetty kuvassa 4.6. Edellä mainittujen aikaleimojen ohella se sisältää seuraavia tietoja:

LI	VN	Mode	Stratum	Poll	Precision
Root Delay 32 bit					
Root Dispersion 32 bit					
Reference Clock Identifier 32 bit					
Reference Timestamp 64 bit					
Originate Timestamp 64 bit					
Receive Timestamp 64 bit					
Transmit Timestamp 64 bit					
Authenticator (optional) 96 bit					

Kuva 4.6. NTP-paketin (versio 3) rakenne. [25]

Leap Indicator (LI, 2 bit) ilmoittaa lähestyvistä karkaussekunnista. Version Number (VN, 3 bit) ilmaisee lähettäjän käytettävän NTP-protokollan version. Mode (3 bit) ilmaisee palvelinten välisen assosiaation. Stratum (8 bit) määrittää palvelimen tarkkuustason. Poll Interval (Poll, 8 bit) määrittää pienimmän aikavälin sekunneissa pyöristettynä lähimpään kahden potenssiin, joka voi olla kahden peräkkäisen kyselyn välillä. Precision (8 bit) ilmaisee kellon resoluution sekunneissa pyöristettynä lähimpään kahden potenssiin. [25]

Root Delay (32 bit) kertoo kokonaiskiertoaikaviiveen juuripalvelimeen saakka. Root Dispersion (32 bit) ilmaisee suurimman mahdollisen virheen suhteessa juuripalvelimeen. Reference Clock Identifier (32 bit) identifioi referenssikellon yksikäsitteisesti. Reference Timestamp on viimeisin aika, jolloin paikallista kelloa on korjattu. Authenticator-kenttä mahdollistaa NTP:n tunnistamismekanismien käyttöönoton. [25]

#### 4.2.2. *Systeemikellon korjaaminen*

Saapunut kellokeskeytys aiheuttaa systeemikellon ajan korottamisen ennalta määritellyllä arvolla. Systeemikellon poikkeama voidaan korjata tekemällä ylimääräisiä korjauksia sen ylläpitoon käytetyn tietueen arvoon. Taajuusvirheen korjaaminen suoritetaan muuttamalla yhdellä kellokeskeytyksellä tietueeseen lisättävän korotuksen suuruutta ja näin ollen voidaan vaikuttaa kellon näennäiseen taajuuteen. Tämän lisäksi NTP asettaa 11 minuutin välein reaaliaikakellon ajaksi systeemikellon ajan, jotta myös reaaliaikakello pysyy riittävän tarkasti oikeassa ajassa. [25]

Käyttöjärjestelmän *gettimeofday()*-, *settimeofday()*- ja *adjtime()*-systeemikutsujen lisäksi NTP tarjoaa ajan hallintaan uudet *ntp\_gettime()*- ja *ntp\_adjtime()*-systeemikutsut. Yksinkertaisin tapa on asettaa systeemikelloon uusi aika *settimeofday()*-kutsun avulla. Tämä voi kuitenkin tietyissä tilanteissa aiheuttaa systeemikellon siirtämisen taaksepäin. [26]

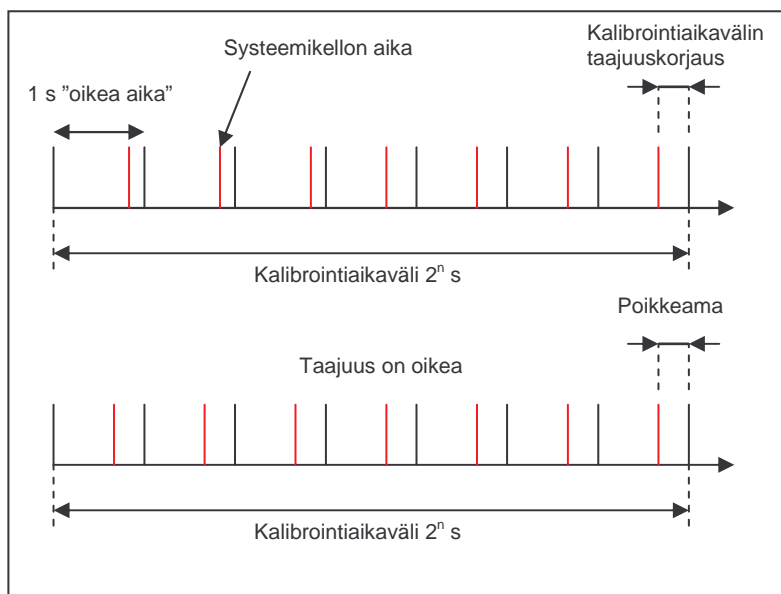
*Adjtime()*-kutsun avulla systeemikelloa voidaan korjata vähitellen muuttamalla sen näennäistä taajuutta ilman, että kelloa siirretään taaksepäin. Kun saavutettu tarkkuus on parempi kuin  $\pm 128$  ms, säännöllinen korjaaminen toteutetaan *ntp\_adjtime()*-kutsun avulla. [26]

### 4.2.3. PPS-signaalin prosessointi

NTP-versio 3 sisältää sovellusrajapinnan (PPS API, Pulse-Per-Second Application Programming Interface), jonka avulla voidaan prosessoida referenssikellon tuottama PPS-signaali. Linux-käyttöjärjestelmässä sovellusrajapinnan käyttöönotto vaatii kernelin päivittämisen PPSkit-paketilla. Moduuli muodostaa aikaleiman PPS-signaalin transitiosta ja mahdollistaa PPS-signaalin käyttämisen synkronointilähteen muodostaman ja absoluuttista aikaa ilmaisevan ASCII-aikaleiman ohella synkronoinnissa. [26]

NTP-prosessissa PPS-signaali tulee assosoida yhteen referenssikelloon, jonka avulla voidaan suorittaa synkronointi absoluuttisen ajan suhteen ASCII-aikaleimaa käyttäen. Tavallisimmin PPS-signaali assosioidaan sen tuottavaan referenssikelloon kuten GPS-vastaanottimeen. Kun PPS-signaalin perusteella mitattu synkronoinnin tarkkuus suhteessa assosioituun referenssikelloon on parempi kuin 128 ms, kernel ei enää huomioi NTP-prosessin ASCII-aikaleimojen pohjalta muodostamia korjauksia, vaan ottaa käyttöön kernelin sisäisen tuen PPS-signaalille ja määrittää tarvittavat korjaukset pelkästään PPS-signaalin perusteella. [28]

PPS-signaalin prosessoivaa *hardpps()*-rutiinia kutsutaan kerran sekunnissa. *Hardpps* vertaa aikaleimaa paikallisesta kellosta otettuun näytteeseen ja muodostaa estimaatin taajuuspoikkeamalle. Tätä poikkeamaa käytetään paikallisen kellon taajuuden korjaamiseen. Näin saavutettava tarkkuus on joitakin osia  $10^8$ :ssa, mikä on kaksi kertaluokkaa parempi kuin korjaamattomalla oskillaattorilla saavutettava tarkkuus. [26]



Kuva 4.7. Aika- ja taajuusvirheen määrittäminen. [29]

*Hardpps()*-rutiinia kutsutaan kahdella parametrilla, joita ovat pulssin absoluuttinen saapumishetki (aikaleima) sekä suhteellinen aika edelliseen pulssiin. Ensimmäistä parametria käytetään systeemikellon sekunnin alun ja PPS-pulssin reunan välisen eron minimointiin. Jälkimmäisen parametrin avulla minimoidaan ero kellojen välisessä taajuudessa. Systeemikellon aika- ja taajuusvirheen määrittämistä on havainnollistettu kuvassa 4.7. [29]

Normaalitilanteessa palvelimet vaihtavat aikaleimoja noin 20 minuutin välein laskeakseen tarvittavat korjaukset paikallisen kellon taajuuteen ja poikkeamaan. Käytettäessä PPS-signaalia vastaava toimenpide suoritetaan joka sekunti ja näin ollen systeemikello sidotaan tiukemmin ulkoiseen referenssiin. [29]

PPS-signaali voidaan luontevasti syöttää tietokoneelle sarjaportin DCD-nastan kautta. Synkronointilähteen muodostama aikaleima voidaan tavallisesti siirtää saman sarjaportin datasiirtoon käytettävissä johtimissa. Tietokone suorittaa aikaleimauksen niin pian kuin se on mahdollista havaitessaan transition sarjaportin DCD-nastassa [30].

Käytettävä sovellusrajapinta voi tukea kaiutustoimintoa, jonka avulla havaittu pulssi kaiutetaan takaisin aikaleimauksen jälkeen. Tämä ominaisuus mahdollistaa pulssin prosessointiviiveen ylärajan estimoinnin [30].

## 5. Synkronointijärjestelmä

Ennen synkronointijärjestelmän toteuttamista testiverkon kellot synkronoitiin NTP-protokollan avulla. Tällöin aikareferenssinä käytettiin ensisijaisen palvelimen paikallista kelloa, johon verkon muut tietokoneet synkronoituivat verkkorajapintojen välityksellä vaihdettavien aikaleimojen avulla.

Tällä tavoin toteutettuna saavutettava synkronoinnin tarkkuus heikkenee nopeasti siirryttäessä hierarkiassa alaspäin. Ensisijaisen palvelimen kellokiteen käyttäminen aikareferenssinä muodostaa merkittävän virhelähteen verkon synkronoinnin kokonais-tarkkuuteen. Lisäksi tietokoneiden kellot sidotaan varsin väljästi referenssikelloon, minkä seurauksena saavutettava synkronoinnin tarkkuus on parhaimmillaankin vain millisekuntien suuruusluokkaa.

Tarkkuus huononee verkon ja tietokoneiden kuormituksen kasvaessa koska kierto-aikaviiveen ja aikaleimojen perusteella johdettu kellojen välisen poikkeaman estimaatti eroaa suuremmalla todennäköisyydellä poikkeaman todellisesta arvosta. Pahimmillaan tämä aiheuttaa sen, että synkronoinnista huolimatta tietokoneiden kellojen poikkeama on niin suuri, että verkossa suoritettavan aikaleimauksen perusteella osa välitettävistä IP-paketeista näyttää saapuneen perille ennen kuin ne on edes lähetetty.

Tämän seurauksena tulevalle synkronointijärjestelmälle haettiin rakenne, jossa tietokoneiden kellot sidotaan tiukemmin referenssikelloon eikä verkon ruuhkautuminen täysin romuta saavutettua synkronointia. Käytännössä tämä on ainakin osittain mahdollista saavuttaa hieman perinteisestä NTP-hierarkiasta poikkeavalla rakenteella, jossa GPS-vastaanottimen tuottama PPS-tarkkuussignaali syötetään ensisijaisen NTP-palvelimen sijasta kaikille synkronointiverkon palvelimille.

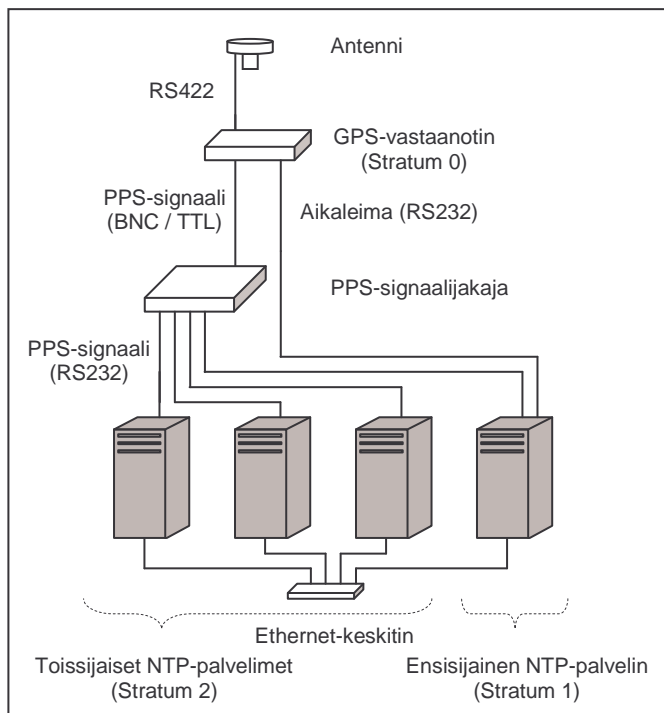
### 5.1. Synkronointijärjestelmän rakenne

Testiverkon synkronoimiseksi suunniteltu synkronointijärjestelmä on esitetty kuvassa 5.1. Järjestelmä käyttää aikareferenssinä Trimble Acutime –GPS-vastaanotinta, joka NTP-hierarkiassa edustaa parasta tarkkuustasoa (stratum 0). Valmistajan ilmoittama PPS-signaalin tarkkuus suhteessa UTC-aikaan on  $\pm 50$  ns (keskihajonta). GPS-vastaanotin kytkeytyy ensisijaiseen NTP-palvelimeen (stratum 1) sarjakaapelilla ja mahdollistaa absoluuttisen ajastusinformaation siirtämisen ASCII-aikaleiman avulla.



PPS-signaalin syöttämiseksi kaikille palvelimille rakennettiin tähän diplomityöhön liittyen PPS-signaalinjakaja, jonka rakenteellisia vaatimuksia ja toteutusta on käsitelty luvussa 5.2. PPS-signaalinjakaja muuntaa GPS-vastaanottimelta tulevan TTL-tasoisin signaalin sarjaportteissa käytetyiksi jännitetasoiksi.

Toissijaiset palvelimet (stratum 2) saavat absoluuttisen ajastusinformaation Ethernet-segmentin yli vaihdettavien aikaleimojen avulla, mikä mahdollistaa edelleen PPS-signaalinjakajalta saatavan tarkkuussignaalin käyttämisen synkronoinnissa. Käytetty hierarkia poikkeaa NTP-määrittelyn mukaisesta synkronointihierarkiasta alimman tarkkuustason palvelinten osalta, joille syötetään tässä toteutuksessa GPS-vastaanottimelta saatava PPS-signaali. NTP-määrittelyn mukaan vain ensisijaiset palvelimet ovat suoraan yhteydessä aika- ja taajuusreferensseihin.



Kuva 5.1. Synkronointijärjestelmän rakenne.

Kuvan 5.1. mukainen järjestelmä on yhtenevä testausvaiheessa käytetyn järjestelmän kanssa. Varsinaisessa testiverkossa ei kuitenkaan ole yksinomaan NTP-synkronointiin tarkoitettuja palvelimia, vaan NTP-prosessit suoritetaan palvelimissa, jotka vastaavat myös liikenteen generoinnista, pakettien välittämisestä ja aikaleimauksesta.

Testiverkon Ethernet-segmentin fyysinen rakenne voi vaihdella suoritettavan mittauksen perusteella, mutta sen tulee kuitenkin mahdollistaa aikaleimojen vaihtaminen ensisijaisen ja toissijaisten palvelinten välillä.

Testiverkossa suoritettavat mittaukset suoritetaan yleensä maksimikuormalla, minkä seurauksena olosuhteet verkon yli suoritettavalle synkronoinnille ovat pahimmat mahdolliset. Kun palvelimet ovat synkronoituneet pelkästään PPS-signaalin, ei verkon kuormituksen kasvamisen tulisi vaikuttaa synkronoinnin tarkkuuteen muutoin kuin prosessorin kuormituksen kasvun välityksellä.

## 5.2. PPS-signaalijakaja

PPS-signaalijakajan suunnittelun lähtökohdaksi otettiin kaksi järjestelmän luotettavuuteen ja tarkkuuteen vaikuttavaa tekijää. Signaalijakajan lähtöliittimistä saatavan PPS-signaalin vaiheessa tulee olla mahdollisimman vähän hajontaa eri lähtöjen välillä. Lisäksi lähtöliittimien välillä tulee olla täydellinen maaerotus.

Ensimmäinen vaatimuksista on itsestään selvä. Jotta palvelinten välinen synkronointi olisi mahdollisimman tarkka, on synkronointiin käytettävän PPS-signaalin oltava identtinen jokaisen palvelimen kannalta.

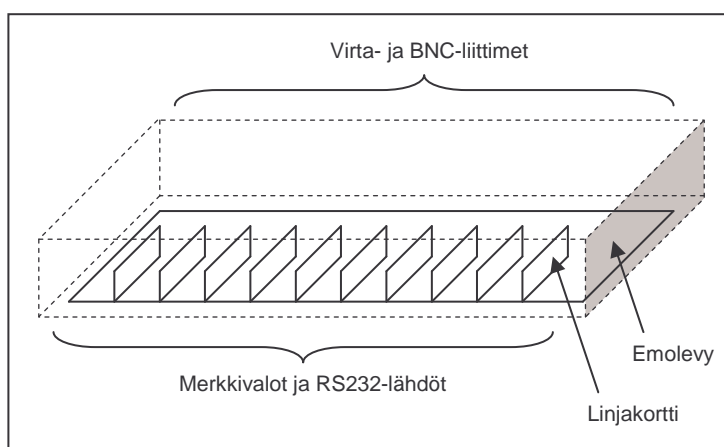
Lähtöjen välinen maaerotus on pakollinen, jotta järjestelmän sähkömagneettisten häiriöiden sietokyky olisi edes välttävä. Tämä johtuu siitä, että tietokoneen sarjaportin maajohdin on emolevyn ja edelleen virtalähteen maatasen kautta yhteydessä suoraan verkkomaahan. Näin ollen kahden tietokoneen sarjaporttien maajohtimien yhdistäminen signaalijakajassa johtaa pahimmillaan useiden neliömetrien suuruisen maasilmukan syntymiseen.

Jos maasilmukka muodostuu, se toimii vastaanottoantennina, johon silmukan läpi kulkeva magneettivuo indusoi häiriövirtoja. Häiriövirrat voivat aiheuttaa elektroniikkalaitteiden maatasen potentiaalin kasvamisen ja edelleen laitteen poikkeuksellisen toiminnan tai pahimmassa tapauksessa täydellisen rikkoontumisen.

### 5.2.1. Signaalijakajan mekaaninen rakenne

Signaalinjakaja rakentuu kuvan 5.2. mukaisesti emolevystä ja 20 linjakortista, joille tarvittavat komponentit on sijoitettu. Laitetekeloksi valittiin 1U (1,75 ” = 4,45 cm) korkea rakkikotelo. Laitetekelön vapaa sisämitta riippuu käytetystä materiaalista ja valmistusmenetelmästä. Valitun laitekotelon pienin vapaa sisäpuolen korkeus on 37,6 mm etu- ja takaseinän kiinnityslaippojen kohdalta, mutta on 40,8 mm muualla kotelon alueella.

Laitetekelön takaseinässä sijaitsevat virtaliitin ja sulakkeet sekä BNC-liittin PPS-pulssin kytkemiseksi laitteeseen. Tämän lisäksi takaseinässä on viisi BNC-liitintä, joista jaettu PPS-pulssi voidaan kytkeä esimerkiksi erilaisille mittalaitteille. Linjakorttien sijoittaminen laitteen etureunaan ja emolevvyyn nähden pystyyn mahdollistaa laitekotelon etuseinän tehokkaan käytön merkkivalojen ja RS232-lähtöjen sijoittelussa.

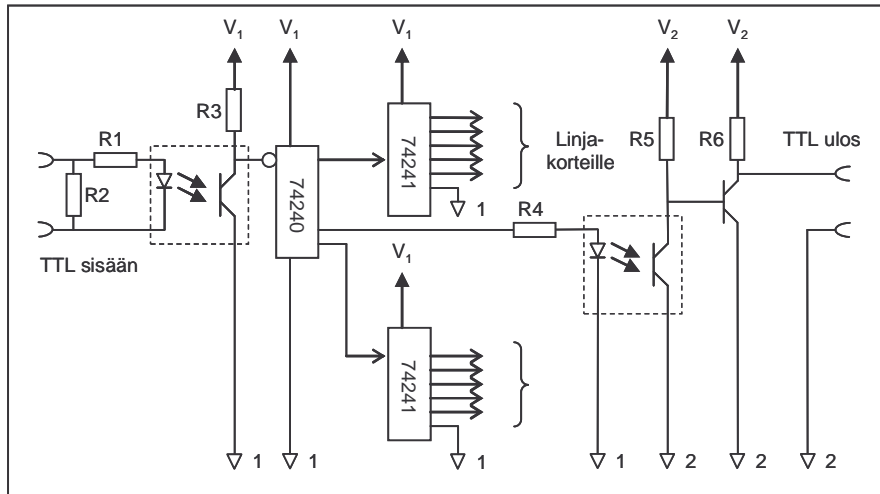


Kuva 5.2. Signaalijakajan mekaaninen rakenne.

Pulssin kytkemiseksi tietokoneen sarjaporttiin tarvitaan vain yksi johdinpari. Lisäksi 9-nastainen sarjaliitin on liian suurikokoinen sijoitettavaksi yhdessä merkkivalojen kanssa 1U-korkean laitekotelon etureunaan. Näin ollen sarjaliittimet korvattiin 3,5 mm monoplugeilla, jotka mahtuvat huomattavasti pienempään tilaan ja tarjoavat silti riittävän määrän johtimia. Kaikissa PPS-pulssin siirtämiseen käytettävissä liittimissä tulee olla täydellinen sähköinen erotus laiterungosta. Muutoin laitteen avulla saavutetun maakerotuksen pilaa laitteen runko, joka kytkee kaikkien lähtöliittimen maajohtimet yhteen.

### 5.2.2. Emolevyn toimintaperiaate

Emolevyllä sijaitsevat tarvittavat komponentit pulssin jakamiseksi, käyttöjännitteiden muodostamiseksi ja BNC-liittimien kautta tapahtuvaa siirtoa varten. Varsinainen maaerotus ja signaalitasojen muuttaminen sarjaporttiin yhteensopivaksi suoritetaan linjakorteilla.



Kuva 5.3. Emolevyn toimintaperiaate.

Kuvassa 5.3. on esitetty emolevyn toiminnan periaate. GPS-vastaanottimelta tuleva PPS-pulssi kytketään vastusten R1 ja R2 muodostaman sovituspierin kautta optoerottiin, jonka avulla GPS-vastaanotin jää sähköisesti täysin irrallalleen signaalijakajasta. Käytetyn optoerottimen erotuskyky on 1,5 kV.

Vastuksen  $R1 = 680 \Omega$  ja  $R2 = 56 \Omega$  muodostavat rinnankytkennän, jonka kokonaisresistanssi on  $51,7 \Omega$ . Näin ollen GPS-vastaanottimelta tuleva koaksiaalikaapeli (ominaisimpedanssi  $50 \Omega$ ) sovituu riittävän hyvin, eivätkä impedanssimuutoksen aiheuttamat heijastukset vääristä pulsseja. Vastus R1 rajoittaa lisäksi optoerottimen valodiodin läpi kulkevan virran 7,4 mA:iin ja estää valodiodin vaurioitumisen.

Optoerotin kääntää pulssin polariteetin, koska pulssin saapuessa valodiodille transistorin kanta johtaa ja näin ollen vastuksen R3 ja transistorin kannan välinen piste maadoituu. Vastuksen  $R3 = 470 \Omega$  avulla transistorin kannan läpi kulkeva virta rajoitetaan 10,6 mA:iin.

Optoerotin jälkeinen invertoiva puskuripiiri (74240) kääntää pulssin polariteetin takaisin alkuperäisen kaltaiseksi. Puskurin tehtävänä on jakaa pulssi kahdelle puskuripiirille (74241), jotka edelleen jakavat pulssin edelleen linjakorteille. Puskuripiirit varmistavat, että signaalin jakamisesta huolimatta jännite ei putoa liian alas. Ilman puskuripiirejä olisi vaarana, että optoerotin läpi kulkeva virta ei ole riittävän suuri ja pulssin aikana jännite putoaa liian paljon nimellisestä 5 V arvostaan.

Invertoiva puskuripiiri (74240) syöttää pulssin lisäksi viidelle TTL-tasoiselle mittauslähdölle, joista on kuvassa 5.3. esitetty ainoastaan yksi. Mittauslähdet erotetaan optoerotin avulla toisistaan ja RS232-lähdöistä. Pulssi syötetään vastuksen  $R4 = 680 \Omega$  läpi optoerottimeen, joka kääntää pulssin polariteetin.

Vastuksen  $R5 = 1 \text{ k}\Omega$  avulla rajoitetaan optoerotin transistorin kannan läpi kulkeva sekä seuraavan transistorin kantaa ohjaava virta 5 mA:iin. Ketjun viimein transistori kääntää pulssin polariteetin takaisin alkuperäisen kaltaiseksi. Vastuksen  $R6 = 100 \Omega$  avulla rajoitetaan transistorin kannan läpi kulkeva virta 50 mA:iin ja saadaan aikaan riittävä sovitus, jotta  $50 \Omega$  ominaisimpedanssinen koaksiaalikaapeli voidaan kytkeä BNC-liittimellä lähtöporttiin. Paras sovitus olisi saatu aikaan käyttämällä  $R6$  vastuksena  $50 \Omega$ , mutta tällöin transistorin läpi maadoittuvan virran suuruus olisi ollut 100 mA, mikä on liian suuri valitulle transistorityypille.

Emolevyn keskeisillä komponenteilla on oma käyttöjännite ja erillinen maataso. Lisäksi mittauslähdöillä on kullakin oma käyttöjännitteensä ja maatasonsa. Myös RS232-lähdöillä on kullakin oma käyttöjännite ja maataso. Kun pulssi syötetään kuhunkin lähtöön optoerotin kautta, vallitsee kaikkien lähtöjen ja myös GPS-vastaanottimen välillä täydellinen maaerotus.

Käytännön nyrkkisääntönä voidaan pitää, että ilman läpilyöntikestävyys on  $1 \text{ kV} / \text{mm}$  ja piirilevyllä vastaava arvo on  $1 \text{ kV} / \text{cm}$ . Komponenttien sijoittelusta ja johdinvedoista johtuen erotuskyky jää muuntajien ja optoerotinien mahdollistamaa  $1 \text{ kV}$  erotuskykyä pienemmäksi, koska  $1 \text{ cm}$  sääntö ei toteudu kaikkialla piirilevyllä ja pahimmillaan erotettuja ja erottamattomia johtimia vedetään lähes toistensa yli  $1,5 \text{ mm}$  paksuisen piirilevyn vastakkaisilla puolilla.

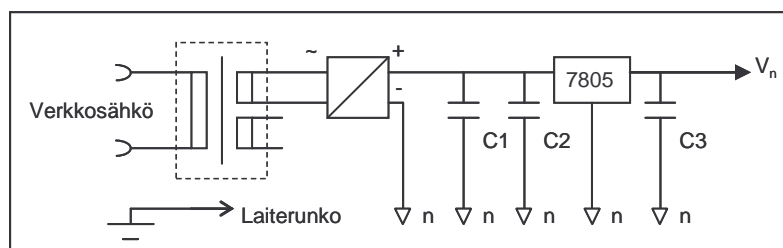
Käytännössä erotuskyky on kuitenkin riittävä sähköverkon normaalitilassa aiheuttamille häiriöille. Näin ollen esimerkiksi verkkojännitteen vuotaminen tietokoneen sarjaporttiin ei vaurioita muita tietokoneita. Ukkosen sähköverkolle tai GPS-vastaanottimen antennille aiheuttamat häiriöt muodostavat kuitenkin oman uhkakuvansa, jota vastaan suojauminen on hyvin vaikeaa.

Emolevyn täydellinen periaatepiirros virtalähteineen on esitetty liitteessä A. Emolevyn komponenttien sijoittelu ja tarvittavat johdinvedot on esitetty liitteessä B. Piirilevyn pinnassa (komponenttipuoli) kulkevat johtimet on esitetty punaisella värillä ja emolevyn pohjassa kulkevat johtimet sinisellä värillä. Kaikki piirilevylle liitettävät komponentit ovat läpijuotettavia. Tietyt komponentit tulee juottaa molemmin puolin, koska läpivienti voidaan toteuttaa kätevästi komponentin jalan avulla komponentin rakenteen niin sallies-  
sa. Emolevyn mitat ovat 250 mm x 437 mm.

### 5.2.3. Erotettujen käyttöjännitteiden tuottaminen

Signaalijakassa tarvitaan yhteensä 26 toisistaan erotettua 5 V käyttöjännitettä. Halvin tapa tarvittavien käyttöjännitteiden muodostamiseen on sijoittaa emolevylle 13 kappaletta kahdella toisiokäämillä varustettuja pienenmuuntajia, jotka muuntavat 230 V verkkojännitteen 9 V vaihtojännitteeksi. Käytetyssä muuntajassa on 1,5 kV erotuskyky ensiö- ja toisiokäämien sekä 1 kV erotuskyky toisiokäämien välillä. Muuntajan toisiokäämeistä voidaan ottaa 9 V jännitteellä virtaa maksimissaan 83 mA.

Laiterunko on kytketty verkkomaahan. Näin ollen varmistetaan laiterungon jännitteet-  
tömyys myös mahdollisissa vikatilanteissa. Vaihtojännitteen muuntamiseksi 5 V tasa-  
jännitteeksi tarvittava oheispiiri on esitetty kuvassa 5.4. Muuntajan jälkeisen tasasuun-  
taussillan avulla suoritetaan vaihtojännitteen tasasuuntaus. Tasasuunnattu jännite on  
sellaiseen hyvin epäpuhdasta, joten käyttöjännite tulee tasata regulaattorin avulla.



Kuva 5.4. Signaalijakajan käyttöjännitteiden muodostaminen.

Käyttöjännitteen epäpuhtaus voi aiheuttaa elektroniikkakomponenttien epänormaalia tai vajaata toimintaa. Tässä sovelluksessa erittäin haitallista olisi esimerkiksi pulssin reunan liipaisurajan vaihtelu tai muodostettavan pulssin epäsäännöllinen nousuaika komponentin lähtöpuolella vaihtelevan tai liian alhaisen käyttöjännitteen seurauksena.

Regulaattorina käytetään 7805-sarjan piiriä, joka muodostaa maksimissaan 35 V tasajännitteestä ulostulojännitteen, jonka arvo on 5 V. Elektrolyyttikondensaattori C1 = 470  $\mu$ F estää jännitteen putoamisen liian alas tasasuuntausjakson aikana. Tavallisesti regulaattori tarvitsee jatkuvan kolmen voltin jännite-eron tulo- ja lähtöjännitteiden välillä toimiakseen hyvin. Kondensaattorit C2 = C3 = 220 nF estävät regulaattorin lähtöjännitteen mahdollisen värähtelyn.

Käytettyjen muuntajien alhainen hyötysuhde (57 %) aiheutti ongelmia laitteen käyttöönoton jälkeen muuntajien ja koko laitekotelon lämpenemisen seurauksena. Alhaisen hyötysuhteen seurauksena huomattava osa muuntajan läpi kulkevasta tehosta muuttuu lämmöksi. Muuntajien läpi kulkevan virran ja edelleen lämpötilan nouseminen liian suureksi aiheuttaa helposti ensiökäämin katkeamisen, joka on tyypillisesti tällaisissa muuntajissa hyvin ohutta käämilankaa.

Oikosulun muodostuminen ilman asianmukaisia sulakkeita olisi luonnollisesti sellaisenaan ilmeinen tulipalon riski. Muuntajien lämpenemisen pienentämiseksi laitekoteloon asennettiin kaksi 30 mm kotelotuuletinta ja muuntajien päälle asennettiin piitahnan avulla jyrsimällä muotoillut ja madalletut jäähdytysrivit. Lisäksi laitekotelon sisälle rakennettiin muovista seinät, jotka parantavat ilmanvirtausta muuntajien ympärillä.

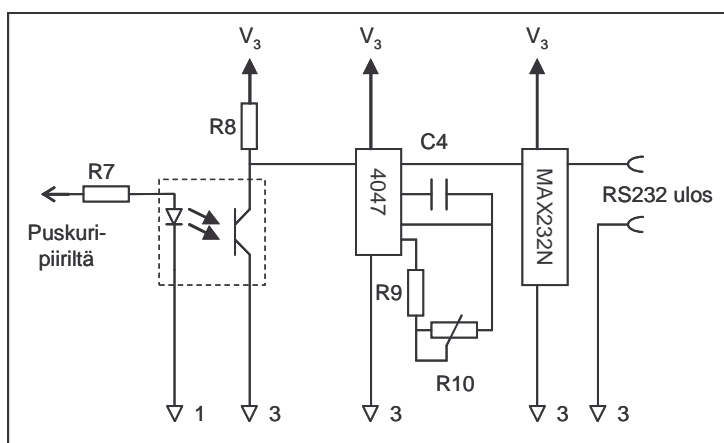
Näillä keinoilla muuntajien ulkolämpötila pienentyi huomattavasti ja muuntajien pinta on pitkänkin käytön jälkeen korkeintaan kädenlämpöinen. Muuntajien muovikuori johtaa kuitenkin huonosti lämpöä, minkä seurauksena muuntajan pinta voi tuntua viileältä vaikka sisällä lämpötila olisikin riittävän korkea vaurioiden syntymiselle.

#### *5.2.4. Linjakortin toimintaperiaate*

Puskuripiireiltä pulssi kytketään linjakorteille, joiden yksinkertaistettu rakenne on esitetty kuvassa 5.5. Optoerottimen ja vastusten R7 ja R8 toiminta on identtinen emolevyn kohdalla esitetyn toiminnan kanssa. Optoerottimen jälkeisen multivibraattoripiirin (4047) tehtävänä on muuttaa tarvittaessa PPS-pulssin leveyttä.

GPS-vastaanottimelta saatavan pulssin leveys on 10  $\mu\text{s}$ . Mikäli kyseessä on ESD-suojattu (Electrostatic Discharge) tietokoneen sarjaportti, on mahdollista, että näin lyhyt pulssi ei pääse sisälle sarjaporttiin, vaan suojaohjelmisto maadoittaa pulssin. Multivibraattoriin kytketyn vastusten  $R9 = 2 \text{ k}\Omega$  ja  $R10 = 100 \text{ k}\Omega$  (lineaarinen säätövastus) sekä kondensaattorin  $C4 = 1 \mu\text{F}$  muodostaman piirin avulla voidaan määrittellä aikavakio väliltä 5  $\mu\text{s}$  - 250  $\mu\text{s}$ , jonka määrittelemän ajan liipaisuehdon täyttävä tila säilytetään riippumatta sisään tulevan pulssin pituudesta.

Koska optoerotin kääntää pulssin polariteetin, multivibraattoriin tulee liipaista pulssin laskevasta reunasta. Muutoin pulssi tulkitaan virheellisesti ja tämä aiheuttaa virhettä mittaus- ja RS232-lähtöjen välillä. Signaalijakajan valmistuksen jälkeen osoitettiin kuitenkin, että pulssin leveyttä voidaan säätää suoraan GPS-vastaanottimen hallintaohjelmiston avulla. Täten pulssin leventämisoptio signaalijakajassa voidaan mieltää tietyssä mielessä hyödyttömäksi.



Kuva 5.5. Linjakortin toimintaperiaate.

Sarjaportin kanssa yhteensopivat jännitetasot muodostetaan lopuksi MAX232N-linjaohjaimen avulla. Looginen nolla ilmaistaan jännitetasolla, joka on välillä +3 V ... +15 V ja looginen yksi -3 V ... -15 V välillä olevalla jännitteellä. Väli -3 V ... +3 V on epämääräistä aluetta, jolla päätöstä ei tehdä puoleen eikä toiseen. [31]

Linjakortin täydellinen periaatepiirros on esitetty liitteessä C. Linjakortin komponenttien sijoittelu ja tarvittavat johdinvedot on esitetty liitteessä D. Tarvittavat johdinvedot on toteutettu yksipuoleisella piirilevyllä, johon liitettävät komponentit ovat läpijuotettavia. Linjakortin mitat ovat 28 mm x 98 mm.



### 5.2.5. Signaalijakajan ja synkronointiketjun aiheuttama virhe

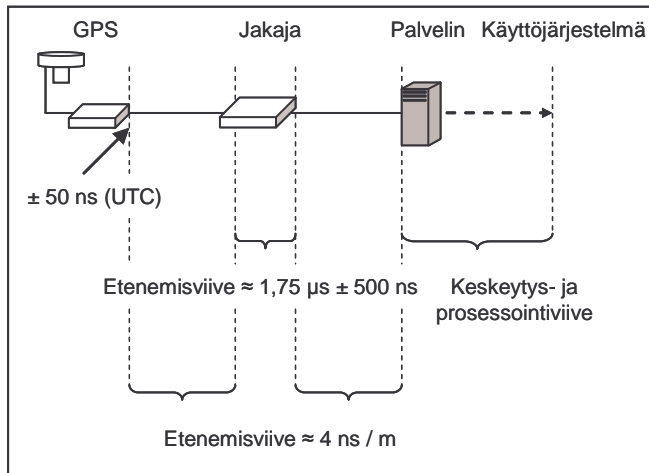
Signaalijakajan lähtöporttien välinen aikaero sekä jakajan PPS-signaaliin aiheuttama etenemisviive mitattiin oskilloskoopin avulla käyttämällä referenssinä suoraan GPS-vastaanottimelta saatavaa PPS-signaalia. Saadut mittaustulokset on havainnollistettu liitteen E kuvassa E1.

Mittaustulosten perusteella signaalijakajan BNC-mittauslähtöjen välinen aikaero on suurimmillaan noin 0,2  $\mu$ s. Mittauslähdöissä on signaalin etenemisviiveestä ja komponenttien nousuajasta johtuen absoluuttista viivettä keskimäärin 800 ns. RS232-lähtöjen välinen aikaero on suurimmillaan noin 0,5  $\mu$ s -8 voltin kohdalta mitattuna. RS232-lähtöjen absoluuttinen viive on keskimäärin 1,75  $\mu$ s -3 voltin kohdalta mitattuna.

Tietokoneiden välisen synkronointitarkkuuden kannalta absoluuttisen viiveen suuruudella ei kuitenkaan ole merkitystä, koska kaikki tietokoneet näkevät lähes identtisen viiveen. Synkronointitarkkuus heikentyy kuitenkin oleellisesti, mikäli sarjaportteissa käytettyjen linjaohjainten liipaisutasoissa on suuria eroja. Jos liipaisutaso vaihtelee -3 voltin ja -8 voltin välillä, on tietokoneiden vastaanottamassa referenssisignaalisissa tällöin aikaeroa noin 1  $\mu$ s.

Kuvassa 5.6. on havainnollistettu synkronointiketjun aiheuttamaa virhettä. Synkronointiketjun aiheuttama keskimääräinen viive voidaan kompensoida aikaistamalla GPS-vastaanottimen muodostamaa PPS-pulssia suhteessa UTC-aikaan, mikäli myös absoluuttinen synkronointi halutaan tehdä mahdollisimman tarkasti.

Tietokoneiden välisen synkronointitarkkuuden kannalta kriittisintä on eri lähdöistä saatavien pulssien välinen aikaero. Eri valmistajien tietokoneissa käytetään varmasti erilaisia sarjaportin linjaohjaimia, joten voidaan olla varmoja, että jännitetasossa, jossa pulssin liipaisu suoritetaan, on eroavuuksia. Voidaan kuitenkin olettaa, että useimmat linjaohjaimet liipaisevat lähes välittömästi -3 V jälkeen linjaohjaimen nopeudesta ja resoluutiosta riippuen.



Kuva 5.6. Synkronointiketjun aiheuttama virhe.

Liitteen E kuvassa E1 on esitetty lähtöporttien välinen hajonta vain tietyllä ajanhetkellä. Lähtöporttien signaaleissa on luonnollisesti pientä ajan funktiona vaihtelevaa värinää, jonka suuruus on korkeintaan joitakin kymmeniä nanosekunteja.

Mittaus ei myöskään paljasta lämpötilan vaihtelun tai verkkojännitteen häiriöiden aiheuttamia poikkeamia. Näiden voidaan kuitenkin olettaa olevan korkeintaan mikrosekuntien suuruusluokkaa, koska signaalijakaja ei sisällä lämpötilan muutokseen herkästi reagoivia komponentteja ja käyttöjännitteet muodostetaan muuntajien ja regulaattorien avulla. Lisäksi mahdollisten poikkeamien voidaan olettaa vaikuttavan symmetrisesti kaikkiin signaalijakajan lähtöportteihin.

Mikäli kuvan 5.6. mukaisen synkronointiketjun aiheuttama viive kompensoidaan GPS-vastaanottimessa, voidaan hyvällä tarkkuudella olettaa, että absoluuttinen ajastusinformaatio saadaan siirretyksi tietokoneiden sarjaportteihin  $\pm 0,6 \mu\text{s}$  tarkkuudella. Synkronoinnin lopullisen tarkkuuden määrää kuitenkin tietokoneen käyttöjärjestelmä, joka vastaa systeemikellon päivittämisestä ja ylläpidosta.

### 5.3. PPS-signaalin prosessoinnin edellytykset

PPS-signaalin käyttäminen hyväksi synkronoinnissa vaatii erilaisia päivityksiä ja asetuksia käyttöjärjestelmästä riippuen. Tässä luvussa perehdytään tarkemmin PPS-signaalin prosessoinnin edellyttämiin muutoksiin Linux 2.4.20 ja FreeBSD 5.21 -käyttöjärjestelmillä varustetuissa tietokoneissa.

### 5.3.1. FreeBSD 5.21

FreeBSD sisältää versiosta 3.0 alkaen ainakin osittaisen tuen PPS-signaalin prosessointiin [32]. Uusimmat versiot tukevat PPS-signaalin prosessointia sekä sarja- että rinnakkaisportin välityksellä. Käyttöjärjestelmä sisältää tarvittavat kirjastot ja funktiot PPS-signaalin prosessoimiseksi, joten prosessoinnin käyttöönotto ei vaadi varsinaisia käyttöjärjestelmään liittyviä päivityksiä. Käyttöönotto vaatii kuitenkin kernelin kääntämisen lisäoptioilla ja NTP-ohjelmiston uudelleenkäntämisen.

Käytettäessä sarjaporttia PPS-signaalin siirtämiseen tulee pulssi kytkeä 9-napaisen liittimen nastaan 1 (DCD, Data Carrier Detect) ja maajohdin nastaan 5. FreeBSD ei peruskokoonpanossa tue RFC-2783 mukaista pulssin takaisinkaiutusmahdollisuutta sarjaportissa, mutta mahdollistaa kyseisen toiminnon rinnakkaisportissa. Käytettäessä rinnakkaisporttia tulee PPS-signaali kytkeä 25-napaisen liittimen nastaan 10 ja maajohdin nastaan 7. Kaiutettava pulssi voidaan vastaanottaa nastan 14 kautta.

FreeBSD:ssä PPS-signaalin prosessoinnin (RFC-2783 / PPS API –sovellusrajapinta) kannalta keskeisiä tiedostoja ovat sovellusrajapinnan osalta:

```
/usr/src/sys/sys/timepps.h  
/usr/src/sys/kern/kern_clock.c
```

ja laiteajureiden osalta:

```
/usr/src/sys/dev/sio/sio.c  
/usr/src/sys/dev/ppbus/pps.c
```

Tarvittavat lisäoptiot voidaan lisätä kernelin asetustiedostoon, joka sijaitsee seuraavassa hakemistossa:

```
/usr/src/sys/i386/conf/
```

PPS-signaalin prosessointi otetaan käyttöön lisäämällä asetustiedoston alkuosan options-alueeseen:

```
options      PPS_SYNC
```

Mikäli PPS-signaali siirretään rinnakkaisportin välityksellä, tulee asetustiedoston laitemäärittäisiin lisätä rinnakkaisportin määrittysten jälkeen rivi:

```
device      pps
```

Esimerkkitapauksessa lisäys tehdään seuraavan osion perään:

```
# Parallel port
device      ppc
device      ppbus          # Parallel port bus (required)
device      lpt            # Printer
device      plip           # TCP/IP over parallel
device      ppi            # Parallel port interface device
#device     vpo            # Requires scbus and da
```

Kernelin kääntämisen ja tietokoneen uudelleenkäynnistämisen jälkeen olemassa oleva NTP-ohjelmisto tulee kääntää uudelleen tai tarvittaessa asentaa ohjelmiston uusin versio, joka löytyy osoitteesta: <http://www.ntp.org/downloads.html>. Tässä diplomityössä käytettiin NTP-ohjelmiston versiota 4.2.0.

Joissakin tilanteissa crypto-kirjasto aiheuttaa virheilmoituksia käännettäessä, joten mikäli mahdollista voidaan NTP-ohjelmisto kääntää lähdekoodihakemistossa ilman kyseistä kirjastoa seuraavasti:

```
bsd# ./configure --without-crypto
bsd# make
bsd# make install
```

NTP-ohjelmiston asetukset tehdään */etc/ntp.conf*-tiedostoon. Tiedoston sisältö on stratum 2 –tarkkuustason palvelimessa esimerkiksi seuraavanlainen:

```
server 127.127.1.0
server 10.0.0.1 prefer
server 127.127.22.0
fudge 127.127.22.0 flag3 1

logfile /var/log/ntp
logconfig = all

driftfile /ntp/ntp.drift
```

```
statsdir /ntp/  
filegen peerstats file peerstats type day enable  
filegen loopstats file loopstats type day enable  
filegen clockstats file clockstats type day enable
```

Tietokoneen oma kello on hyödyllistä määritellä yhdeksi referenssikelloksi mahdollisten verkon vikatilanteiden varalta. Muutoin synkronointi menetetään kokonaan yhteyden katketessa varsinaisille referenssipalvelimille.

```
server 127.127.1.0
```

*Server*-komennon avulla määritetään synkronoinnissa käytettävät palvelimet. Mikäli mahdollista tulee määritellä useita vaihtoehtoisia palvelimia, koska se parantaa synkronoinnin tarkkuutta ja vikasietoisuutta. PPS-signaali assosioidaan yhteen palvelimeen *prefer*-avainsanalla.

```
server 10.0.0.1 prefer
```

PPS-signaalin prosessointiin käytettävä ajuri määritetään myös *server*-komennolla.

```
server 127.127.22.0
```

Ajuri vaatii toimiakseen, että on olemassa laite */dev/pps0*. Laite voidaan tarvittaessa muodostaa linkittämällä sarjaportti komennolla:

```
bsd# ln -s /dev/cuaa0 /dev/pps0
```

Laitteen tunnistenumeron */dev/pps0* tulee vastata osoitetta *127.127.22.u*. Käytettäessä rinnakkaisporttia */dev/pps0*-laite muodostuu kernelin kääntämisen yhteydessä automaattisesti.

*Fudge*-komennon avulla referenssikelloajureille voidaan määrittää lisäoptioita. Tässä tapauksessa synkronoinnissa käytetään toimintamuotoa, jossa GPS-vastaanotin lähettää kerran sekunnissa ASCII-aikaleiman. Vaihtoehtoisesti voidaan käyttää toimintamuotoa, jossa aikaleima pyydetään GPS-vastaanottimelta aina tarvittaessa.

```
fudge 127.127.22.0 flag3 1
```

Lokitiedoston sijainti määritetään *logfile*-komennon avulla. Nyt lokitiedostoon tallennetaan kaikki tapahtumat:

```
logfile /var/log/ntp
logconfig = all
```

*Driftfile*-komennon avulla määritetään tiedosto, johon määritetty taajuuskorjaus kirjoitetaan kerran tunnissa. Tiedoston käyttäminen nopeuttaa synkronointia käynnistettäessä ohjelmaa uudelleen.

```
driftfile /ntp/ntp.drift
```

Määritetään halutut tilastotiedostot sekä yksittäiseen tiedostoon tallennettava aikaväli. Tilastojen avulla saadaan tietoa esimerkiksi oman kellon käyttäytymisestä ja oletetusta poikkeamasta suhteessa referenssikelloihin.

```
statsdir /ntp/
filegen peerstats file peerstats type day enable
filegen loopstats file loopstats type day enable
filegen clockstats file clockstats type day enable
```

### 5.3.2. Linux 2.4.20

Linux tukee PPS-signaalin prosessointia ja RFC-2783 mukaista pulssin takaisin-kaiutusta vain sarjaportissa. Tuen käyttöönotto vaatii kernelin päivittämisen PPSkit-paketilla ja sekä kernelin että NTP-ohjelmiston kääntämisen uudelleen.

PPS-signaali kytketään sarjaportin 9-napaisen liittimen nastaan 1 (DCD) ja maajohdin nastaan 5. Mikäli kysymyksessä on stratum 1 –tarkkuustason palvelin, voidaan ASCII-aikaleima siirtää saman liittimen datasiirtoon tarkoitetuissa johtimissa käyttämällä NMEA-ajuria (National Marine Electronics Association). Tämä on mahdollista myös FreeBSD-käyttöjärjestelmässä.

Linux:ssa PPS-signaalin prosessoinnin (RFC-2783 / PPS API –sovellusrajapinta) kannalta keskeisiä tiedostoja ovat sovellusrajapinnan osalta

```
/usr/src/linux/include/linux/timepps.h
```

ja laiteajureiden osalta:

```
/usr/src/linux/drivers/char/serial.c
```

Kernelin päivittämiseen tarvittava PPSkit-paketti löytyy osoitteesta: <ftp://ftp.kernel.org/pub/linux/daemons/ntp/pps/>. Linux version 2.4.20 kanssa yhteensopiva on PPSkit-2.1.1, joka sisältää tarvittavan patch-tiedoston kernelin päivittämiseksi.

Kernelin päivittämisen jälkeen asetustiedostoa voidaan muuttaa esimerkiksi menu-config-komennon avulla. *Code maturity level options* –valikosta tulee sallia *Prompt for development and/or incomplete code/drivers*, mikä mahdollistaa kokeellisten tai epätäydellisten ominaisuuksien tai ajureiden kääntämisen kerneliin.

*Processor type and features* –valikosta tulee valita kohdat:

```
NTP kernel support
NTP PPS support
```

Lisäksi *Character devices* –valikosta tulee valita:

```
NTP PPS support on serial port
```

Kernelin kääntämisprosessin valmistuttua tulee NTP-ohjelmiston takia linkittää kaksi otsikkotiedostoa komennoilla:

```
linux# ln -s /usr/src/linux/include/linux/timepps.h
      /usr/include/sys/timepps.h
Linux# cp /usr/include/sys/timex.h
      /usr/include/sys/timex.old
Linux# ln -s /usr/src/linux/include/linux/timex.h
      /usr/include/sys/timex.h
```

NTP-ohjelmiston kääntäminen on identtinen edellisessä luvussa esitetyn menettelyn kanssa. NTP-ohjelmiston asetukset tehdään */etc/ntp.conf*-tiedostoon, jonka sisältö on stratum 1 -tarkkuustason palvelimessa esimerkiksi seuraavanlainen:

```
server 127.127.1.0
server 127.127.20.0 prefer
fudge 127.127.20.0 flag3 1
```

```
logfile /var/log/ntp
logconfig =all

driftfile /etc/ntp.drift

statsdir /etc/
filegen peerstats file peerstats type day enable
filegen loopstats file loopstats type day enable
filegen clockstats file clockstats type day enable
```

Asetustiedoston rakenne selvitettiin edellisessä luvussa. Ainoana erona on se, että tässä tapauksessa GPS-vastaanottimelta saatava ASCII-aikaleima prosessoidaan NMEA-ajurin avulla, joka vastaa myös PPS-signaalin prosessoinnista, jos *fudge flag3 1* -optio on asetustiedostossa. Ajuri olettaa, että on olemassa laite */dev/gps0*, joka voidaan muodostaa linkittämällä. Linux-käyttöjärjestelmässä sarjaportit on tavallisesti kuvattu laitetiedostoon, jonka nimi on tyyppiä */dev/ttyS0*.

#### 5.4. Tietokoneen synkronoinnin tarkkuus

Käyttöjärjestelmän vaikutus poisluettuna tietokoneen synkronoinnin tarkkuutta rajoittaa eniten tietokoneen oman kellon eli kideoskillaattorin epästabiilius ja vahva lämpötilariippuvuus. Jos oletetaan, että kiteen korvaaminen esimerkiksi lämpötilastabiloidulla kiteellä ei tule kysymykseen, voidaan saavutettavaan tarkkuuteen kuitenkin vaikuttaa tietyillä valinnoilla.

Palvelinalustaa valittaessa on pyrittävä tietoisesti valitsemaan kokoonpano, jonka kiteellä on paras stabiilius ja pienin lämpötilariippuvuus. Eri valmistajien ja hintaluokkien tietokoneissa on todennäköisesti suuriakin eroja kiteen ominaisuuksissa. Parhaan palvelinalustan valinta voidaan suorittaa erilaisten mittausten perusteella, joita käsitellään jäljempänä tässä luvussa.

Hyvälaatuisen tietokoneen kellon stabiiliuden ja tarkkuuden voi kuitenkin pilata väärä toimintaympäristön valinta. Mikäli mahdollista, tulee valita koneellisesti ilmastoitu ympäristö, jossa ilman lämpötilan ja kosteuden muutokset ovat vähäisempiä kuin tavallisessa huoneilmassa. Tulee kuitenkin muistaa, että virheellisesti toimiva ilmastointi voi itse asiassa huonontaa saavutettavaa tarkkuutta.



#### 5.4.1. *Systeemikellon värinä ja satunnainen vaihtelu*

Käyttöjärjestelmässä kelloa päivitetään tasaisin väliajoin luvussa 4.1.1.1. esitetyn periaatteen mukaisesti. Vaikka kellonaikaa luettaessa edellisestä päivityksestä kulunut aika voidaan ekstrapoloida kellojaksolaskurin tai ajastinpiirin rekisteriä lukemalla, on kellon lukemisessa silti vaihtelevaa viivettä, joka aiheutuu muista suoritettavista prosesseista.

Kellon lukeminen suoritetaan tietyn systeemikutsun avulla. Aika, joka systeemikutsulta kuluu kellon lukemiseen, määräytyy prosessien vuorottelusta, tarvittavan muistialueen varaamisesta, suoritettavan prosessin vaihtamisesta sekä muilta prosesseilta ja laitteilta saapuvista keskeytyksistä. Nämä tekijät voivat viivästyttää systeemikutsun suorittamista tai aiheuttaa korkeamman prioriteetin seurauksena muiden prosessien suorittamisen ennen systeemikutsun suoritusta.

Vaikka tietokoneen kello olisi absoluuttisen tarkka, ei kelloa silti pystyttäisi lukemaan täysin virheettömästi systeemikutsun suorituksen vaatiman suoritusajan seurauksena. Nopeamman prosessorin käyttäminen tosin pienentää tätä epävarmuutta.

##### 5.4.1.1. *Systeemikutsun suoritusviive*

NTP-distribuutioon kuuluvalla *util/jitter.c*-ohjelmalla voidaan tutkia systeemikellossa esiintyviä jaksollisia ja satunnaisia epäsäännöllisyyksiä. Ohjelma suorittaa for-silmukassa systeemikellon lukemiseen käytettävää systeemikutsua ja laskee kahden peräkkäisen kutsun palauttamien kellonaikojen välisen erotuksen. Näin ollen suorituksessa olevan prosessin vaihtamisen, korkeamman prioriteetin omaavan keskeytyksen palveleminen tai systeemikellon askeltamisen tulisi näkyä keskimääräistä suurempana erotuksena.

Ohjelma palauttaa systeemikellon ajan oletusarvoisesti mikrosekunnin resoluutiolla, mutta korvaamalla käytettävä *gettimeofday()*-systeemikutsu *ntp\_gettime()*-kutsulla voidaan palautettavan kellonajan resoluutio kasvattaa nanosekuntiin. Liitteessä F on esitetty ohjelman avulla saadut tulokset systeemikutsujen välisen aikaeron käyttäytymiselle systeemikellon päivitystaajuuden arvoilla 100 Hz ja 1000 Hz. Tutkitun tietokoneen kokoonpano on Intel Celeron 300 MHz, jonka käyttöjärjestelmänä on Linux-2.4.20-NANO.

Tarkasteltaessa 2 miljoonan näytepisteen joukkoa (kuva F1) voidaan havaita systeemi-kutsujen väliajan pidentyminen pahimmillaan lähes neljään sekuntiin noin 330 000 näytteen välein. Luonteva selitys ilmiölle on PPS-signaalin aiheuttama keskeytys ker-ran sekunnissa. Kun yhteen systeemikutsuun käytetään kuvan F3 perusteella keski-määrin 3  $\mu$ s, tarkoittaa noin 330 000 näyteväliä tällöin 1 s aikaeroa keskeytysten välillä.

Tarkasteltaessa 50 000 näytepisteen joukkoa (kuva F2) voidaan havaita systeemi-kutsujen väliajan pidentyminen noin 20  $\mu$ s:n tasolle kolme kertaa 10 000 näytepisteen aikana. Ilmiön esiintymistiheys on selvästikin identtinen 10 ms välein saapuvan kello-keskeytyksen kanssa.

Vieläkin pienemmän näytejoukon (kuva F3) tarkasteleminen paljastaa systeemi-kutsujen väliajan olevan pääsääntöisesti hieman yli 3  $\mu$ s. Tämän lisäksi väliajassa on tasavälein noin 100 ns hyppäys, joka on todennäköisesti seurausta prosessien vuorot-telusta tai systeemikellon lukemisessa esiintyvistä virheistä.

Systeemikellon päivitystaajuus voidaan muuttaa määrittelemällä uusi taajuuden arvo *asm/param.h*-tiedostossa ja kääntämällä kernel uudelleen. Taajuuden kasvattaminen arvoon 1000 Hz kasvattaa saapuvien kellokeskeytysten määrän kuvan F4 perusteella kymmenkertaiseksi, mutta ei muutoin oleellisesti vaikuta systeemikutsujen väliajan käyttäytymiseen.

#### 5.4.1.2. Synkronoinnin suhteellinen ja absoluuttinen tarkkuus

NTP:n muodostamien loopstats-tilastotietojen perusteella saadaan tietoa systeemi-kellon käyttäytymisestä sekä poikkeamasta suhteessa käytettävään referenssikelloon. *Loopstats*-tilastotietoihin tulostuva informaatio on seuraavanlainen:

```
53250 187.330 -0.000000803 8.501892 0.000010871 0.035841 6
53250 252.333 -0.000000320 8.501892 0.000011567 0.031039 6
53250 315.326 -0.000000819 8.501892 0.000005222 0.026881 6
53250 381.331 0.000000338 8.501892 0.000009828 0.023280 6
53250 447.323 0.000000697 8.518753 0.000012245 0.021852 6
```

Kaksi ensimmäistä saraketta ilmaisevat päivämäärän (modifioitu Juliaaninen päivä) ja UTC-keskiyöstä kuluneiden sekuntien kokonais- ja murto-osat. Seuraavat kentät ilmaisevat systeemikellon aikapoikkeaman [sekuntia], taajuuspoikkeaman [ppm], RMS-väri-  
nän [sekuntia], Allan-keskihajonnan [ppm] ja käytettävän referenssikellon kysely-  
aikavälin (polling interval) kaksikantaisena logaritmina. Jokainen systeemikellon päi-  
vitys aiheuttaa uuden rivin tulostumisen tiedostoon. [33]

Aikapoikkeaman positiivinen arvo ilmaisee, että systeemikello on referenssikelloa jäljessä. Taajuuspoikkeaman positiivinen arvo puolestaan ilmaisee, että systeemikellon nimellinen taajuus on liian pieni verrattuna referenssikelloon. Yllä olevien tilastotietojen tapauksessa aikavertailu suoritetaan 64 sekunnin välein. Tulee kuitenkin muistaa, että kellon korjaamista ei suoriteta yksittäisen mittauksen perusteella, vaan aika- ja taajuus-  
poikkeama määritetään suhteessa käytettävään kalibrointiaikaväliin.

Määritettyä aikapoikkeamaa ylläpidetään muuttujassa, joka ilmaisee kuinka paljon kellonaikaa tulee korjata jatkossa, jotta se vastaisi mahdollisimman tarkasti referenssi-  
kellon aikaa. Määritetty taajuuspoikkeama pyritään korjaamaan määrittelemällä aika,  
joka lisätään systeemikelloon kunkin sekunnin aikana, jotta kellon nimellinen taajuus  
olisi oikea.

Liitteen G kuvassa G1 on esitetty saavutettava synkronointitarkkuus, kun aikarefe-  
renssinä käytetään GPS-vastaanotinta ilman PPS-signaalia. Kuvan perusteella aika-  
poikkeama pääsääntöisesti pienempi kuin  $\pm 10$  ms. Synkronoinnin alhainen tarkkuus  
johtuu tietokoneen systeemikellon väljästä sitomisesta referenssikelloon.

Tämän seurauksena tietokoneen systeemikello ehtii vaeltaa kauemmaksi oikeasta  
ajasta ennen seuraavaa aikavertailua. Kellonaikojen harva vertailuväli johtaa lisäksi  
väistämättä siihen, että NTP ei kykene reagoimaan yhtä nopeasti systeemikellossa  
esiintyviin poikkeamiin kuin PPS-signaalin avulla suoritettussa synkronoinnissa, jossa  
aikavertailu suoritetaan joka sekunti. Lisäksi aikaleiman mahdollistama synkronoin-  
titarkkuus on huonompi kuin PPS-signaalin mahdollistama tarkkuus.

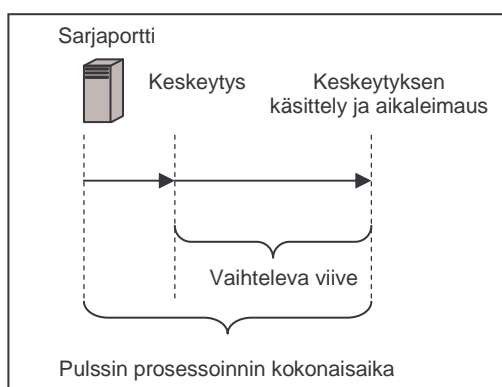
Kuvassa G2 on esitetty edellisen kaltainen tarkastelu, mutta tällä tarkastelujaksolla  
synkronointiin käytettiin myös PPS-signaalia. Aikapoikkeama on keskimäärin  $\pm 1$  us,  
joten synkronointitarkkuuteen on saavutettu lähes 10 000-kertainen parannus.

Aikapoikkeama on lisäksi luonteeltaan kohinan kaltaista eikä siinä esiinny esimerkiksi lämpötilan vaihtelusta johtuvaa jaksollisuutta kuten jäljempänä tullaan havaitsemaan. Saavutettu suorituskyky on käytännössä hyvin lähellä tasoa, johon peruskokopan on omaavalla tietokoneella voidaan parhaimmillaan yltää.

Tilastotietojen sisältämä aikapoikkeama ei kuitenkaan ota huomioon, kuinka paljon PPS-signaalin prosessoinnissa on viivettä. NTP-prosessi kykenee kompensoimaan prosessointiviiveen vaihteluita suodattamalla aikapoikkeamalle saatuja arvoja, mutta prosessointiin kuluva keskimääräinen aika aiheuttaa joka tapauksessa absoluuttista poikkeamaa suhteessa referenssikelloon.

Prosessointiviiveen vaikutus korostuu varsinkin tarkasteltaessa useamman tietokoneen välisen synkronoinnin tarkkuutta. Tietokoneiden väliset erot prosessointiviiveessä ja näin ollen edelleen absoluuttisessa aikapoikkeamassa aiheuttavat tietokoneiden välille suhteellista aikapoikkeamaa. Absoluuttisen poikkeama ja tietokoneiden välinen suhteellinen poikkeama ovat siis vahvasti sidoksissa toisiinsa.

Kuvassa 5.7. on havainnollistettu prosessointiviiveen aiheuttamaa virhettä absoluuttiseen synkronointitarkkuuteen. Pulssin saapuessa tietokoneen sarjaporttiin signaali etenee sarjaportin terminoivan RS232-linjaohjaimen kautta UART-piirille (Universal Asynchronous Receiver/Transmitter). Linjaohjain ja UART-piiri on useimmiten toteutettu samaan fyysiseen komponenttiin.



Kuva 5.7. Prosessointiviiveen aiheuttama virhe.

UART-piiri aiheuttaa keskeytyksen ja ilmoittaa tällä tavoin tarvitsevansa prosessorin huomiota. Pulssin saapumisesta keskeytyspyyntöön kuluva aika voidaan olettaa hyvin lyhyeksi ja säännölliseksi, koska se riippuu ainoastaan UART-piirin ominaisuuksista. Lopulta käyttöjärjestelmä prosessoi saapuneen keskeytyksen ja määrittää keskeytyksen saapumisajan suhteessa systeemikelloon.

Keskeytyspyynnön ja keskeytyksen prosessoinnin välinen aika, riippuu tietokoneen kuormituksesta. Liitteen G kuvassa G3 on havainnollistettu tietokoneen kuormituksen vaikutusta aikapoikkeamaan. Mittausjakson aikana tietokonetta on kuormitettu useilla peräkkäisillä C-kääntäjäprosesseilla, jotka aiheuttavat sekä prosessori- että I/O-kuormitusta.

Ensimmäinen kääntäjäprosessi käynnistyy noin kello 11:45. Prosessorin kuormituksen kasvamisen seurauksena voitaisiin olettaa, että PPS-signaalin prosessointiviive kasvaisi äkillisesti keskimääräistä suuremmaksi ja aikaleimat jäisivät enemmän jälkeen tasasekunnista. Tämä aiheuttaisi edelleen sen, että systeemikello näyttäisi edistävän suhteessa referenssikelloon.

Tilanne on varmasti osittain edellä kuvatun kaltainen, mutta loopstats-tilastotietojen perusteella systeemikellon aikapoikkeama käyttäytyy juuri päinvastoin kuin olettaa saattaisi. NTP-prosessi tulkitsee, että systeemikello on referenssikelloa hieman yli 10  $\mu$ s jäljessä ja alkaa korjata systeemikelloa lähemmäksi referenssikellosta saatua aikaa.

Noin puolen tunnin kuluttua kääntäjäprosessin alkamisesta NTP-prosessi on kyennyt kompensoimaan prosessorin kuormituksen kasvamisen. Kun kääntäjäprosessi päättyy noin kello 12:30, prosessorin kuormitus pienenee takaisin alkuperäiselle tasolle. NTP-prosessi tulkitsee tämän muutoksen siten, että systeemikello on referenssikelloa edellä ja alkaa taas korjata systeemikelloa referenssikellosta saadun ajan suhteen.

Vastaava ilmiö toistuu paljon voimakkaampana seuraavien kääntäjäprosessien aikana. Kun kääntäjäprosessit lopulta päättyvät noin kello 19:00 aikaan, aikapoikkeama palaa vähitellen normaalille tasolle. Edellä havaittuun odottamattomaan aikapoikkeaman käyttäytymiseen palataan luvussa 5.5.2.

Tilastotiedoista saatu aikapoikkeama ei kuitenkaan paljasta kuinka paljon systeemikellon aikaa lopulta korjattiin. Tilastotietoihin tulostuu NTP-prosessin havaitsema poikkeama systeemikellon ja referenssikellon välillä, eikä tiedoista käy ilmi suoritettun korjauksen tai absoluuttisen aikapoikkeaman suuruus.

#### *5.4.2. Systeemikellon taajuuspoikkeama*

Systeemikellon tasainen taajuusvirhe ei ole haitallista saavutettavan synkronoinnin tarkkuuden kannalta, koska NTP havaitsee virheen systeemikellon nimellisessä taajuudessa ja kykenee yleensä kompensoimaan kyseisen virheen. Systeemikellon taajuuden muuttuminen esimerkiksi lämpötilan muuttumisen seurauksena aiheuttaa vaihtelevan taajuusvirheen, jonka täydellinen korjaaminen ei välttämättä ole enää mahdollista.

Systeemikellon värinän ja taajuuspoikkeaman korjaaminen asettavat lisäksi erilaiset vaatimukset käytettävälle mittausaikavälille. Värinän aiheuttaman virheen minimointi edellyttää pitkää mittausaikaa, jotta satunnaiset mittausvirheet keskiarvoistuvat pois. Taajuuspoikkeaman tehokas korjaaminen puolestaan edellyttää, että käytettävä mittausaika on lyhyt, koska tällöin systeemikellon taajuus ei ehdi vaeltaa niin kauas oikeasta arvosta.

##### *5.4.2.1. Lämpötilan vaikutus synkronointitarkkuuteen*

Liitteen H kuvassa H1 on esitetty systeemikellon käyttäytyminen PPS-signaalin kanssa normaalissa (ilmastoimattomassa) toimistohuoneessa Intel Celeron 300 MHz –kokoonpanossa (Linux 2.4.20-NANO), joka käyttää aikareferenssinä GPS-vastaanotinta. UTC-ajanhetkellä 13.00 emolevällä sijaitsevaa 14,31818 MHz kidettä jäähdytettiin äkillisesti elektroniikan vikatestaukseen tarkoitetulla kylmäsuihkeella. Jäähdyttäminen suoritettiin uudelleen noin 20 minuuttia myöhemmin.

Kiteen jäähtymisen seurauksena sen fyysiset dimensiot pienevät ja kiteen muodostaman kellosignaalin taajuus suurenee. Tämän seurauksena systeemikellon aikaa päivitetään liian tiheästi ja systeemikello alkaa edistää suhteessa referenssikelloon. Kiteen lämpeneminen takaisin ympäristön lämpötilaan aiheuttaa lopulta systeemikellon jäämisen jälkeen referenssikellosta.

Toisella jäähdytyskerralla kiteen lämpötila ei muutu yhtä voimakkaasti kuin ensimmäisellä kerralla. Kiteen lämmitessä takaisin ympäristön lämpötilaan ei myöskään nähdä yhtä voimakasta jätättämistä kuin ensimmäisellä kerralla. Tässä tapauksessa NTP kykenee reagoimaan varsin tehokkaasti kiteen lämpenemisen aiheuttamaan taajuusmuutokseen.

Tietokoneen siirtäminen ilmastoituun laboratoriohuoneeseen paljasti, että käytettävä ilmastointilaitte toimii virheellisesti. Liitteen H kuvassa H2 on esitetty tulokset mittausjärjestelystä, jossa huoneen lämpötilaa mitattiin neljässä eri pisteessä lämpötilantureiden avulla.

Tulosten perusteella ilmastointilaitteen puhaltaman ilman lämpötila vaihtelee keskimäärin välillä 0 – 24 °C noin 50 minuutin periodilla. Tämän seurauksena huoneilman lämpötila muuttuu noin 4 °C vastaavalla periodilla huoneen keskeltä mitattuna. Huoneeseen koneellisen ilmanvaihdon syöttämän korvausilman lämpötila on kuitenkin erittäin tasainen eikä näin ollen voi aiheuttaa huoneilman lämpötilavaihtelua.

Huoneilman lämpötilan muuttumisen seurauksena tietokoneen laitekotelon sisälämpötila muuttuu kuvan G3 perusteella noin 2 °C vastaavalla periodilla. Laitekotelon sisälämpötilan muuttuminen muuttaa edelleen systeemikelloa ohjaavan kiteen taajuutta. Kahden asteen lämpötilamuutoksen seurauksena taajuuspoikkeamassa voidaan havaita noin 2,5 ppm muutos. Kiteen taajuuden lämpötilariippuvuus on noin 1,25 ppm/°C, mikä on tyypillinen arvo kvartsikideoskillaattorille.

Kiteen lämpötilavaihtelun seurauksena aika- ja taajuuspoikkeamassa voidaan nähdä lämpötilan kohdalla havaittu jaksollisuus. Huonosti toimiva ilmastointilaitte aiheuttaa lähes kymmenkertaisen heikennyksen ilmastoimattomassa huoneessa saavutettuihin tuloksiin.

Lämpötilan vaikutus tietokoneen kellon käyttäytymiseen voidaan luotettavasti sulkea pois käyttämällä lämpötilastabiloitua kaappia. Testitietokone ja PPS-signaalitasojen muuntamiseen tarvittava elektroniikka sijoitettiin lämpötilastabiloituun kaappiin, joka kykenee säilyttämään asetetun lämpötilan 0,1 °C tarkkuudella.

Saadut mittaustulokset on havainnollistettu liitteen H kuvassa H4. Kolmen tunnin mittaussjakson aikana lämpötilan muutos kaapin sisällä on vain  $\pm 0,1$  °C. Kaapissa saavutettu suorituskyky ei kuitenkaan oleellisesti eroa ilmastoimattomassa huoneessa saavutetuista arvoista ja aikapoikkeama on jälleen kohinan kaltaista.

#### 5.4.2.2. *Kokoonpanon vaikutus synkronointitarkkuuteen*

Tietokonelaitteiston kokoonpanon vaikutusta synkronointitarkkuuteen voidaan tutkia sijoittamalla tietokoneet samaan toimintaympäristöön ja tutkimalla tilastotietojen avulla systeemikellojen käyttäytymistä.

Ilmastoituun huoneeseen sijoitettiin neljä vaihtoehtoista kokoonpanoa: Intel Celeron 300 MHz (Linux), Intel Pentium II 450 MHz (Linux), Intel Pentium II 450 MHz (FreeBSD) ja Intel Celeron 466 MHz (Linux). Näistä kaksi keskimmäistä ovat käyttöjärjestelmä poisluettuna täysin identtisiä.

Liitteen I kuvissa I1 ja I2 on esitetty tilastotietojen perusteella eri kokoonpanojen aika- ja taajuuspoikkeaman käyttäytyminen. Aika- ja taajuuspoikkeamat käyttäytyvät kaikissa kokoonpanoissa hyvin samankaltaisesti, mutta poikkeaman vaihtelun suuruudessa on selviä eroja. Kuvaajissa havaittava aaltoilu johtuu ympäristön lämpötilan jaksollisesta vaihtelusta. Celeron 300 MHz kohdalla epäsäännöllisyyttä aiheuttaa muita tietokoneita suurempi kuormitus.

Pentium II –kokoonpanojen osalta on nähtävissä lähes identtinen käyttäytyminen niin aika- kuin taajuuspoikkeamien lämpötilariippuvuudessa. Taajuuspoikkeaman suuruus on kuitenkin lähes 14 ppm suurempi Linux-käyttöjärjestelmällä varustetussa kokoonpanossa. Taajuusero johtuu todennäköisesti kiteen valmistustoleransseista eikä sillä ole mitään tekemistä käyttöjärjestelmien erilaisuuksien kanssa.

Tulosten perusteella pienin lämpötilariippuvuus on Celeron 466 MHz –kokoonpanossa, jonka aikapoikkeaman vaihteluväli on noin  $\pm 2,5$   $\mu$ s. Muilla kokoonpanoilla vaihteluväli on keskimäärin  $\pm 10$   $\mu$ s. Systeemikellojen stabiiliudessa on näin ollen merkittävä ero. Vastaava ilmiö on luonnollisesta nähtävissä myös taajuuspoikkeamien kohdalla. Tällä perusteella luontevin valinta NTP-palvelinalustaksi olisi pienimmän lämpötilariippuvuuden omaava Celeron 466 MHz.



## 5.5. Tietokoneiden välisen synkronoinnin tarkkuus

Tietokoneiden välisen aikapoikkeaman mittaaminen ei ole niin suoraviivaista kuin voisi olettaa. Tämä johtuu siitä, että ei ole olemassa fyysistä kellosignaalia jota voitaisiin mitata vaan koko systeemikello koostuu kahdesta muuttujasta, joiden arvoa päivitetään kellokeskeytyksen avulla.

Tilastotietojen avulla saadaan toki tietoa kunkin tietokoneen systeemikellon aika- ja taajuuspoikkeamasta, mutta tilastotiedot eivät paljasta systeemikellon ja referenssikellon absoluuttisen aikapoikkeaman suuruutta, joka aiheutuu pääasiassa PPS-signaalin prosessointiviiveestä.

### 5.5.1. Suhteellisen poikkeaman mittausmenetelmä

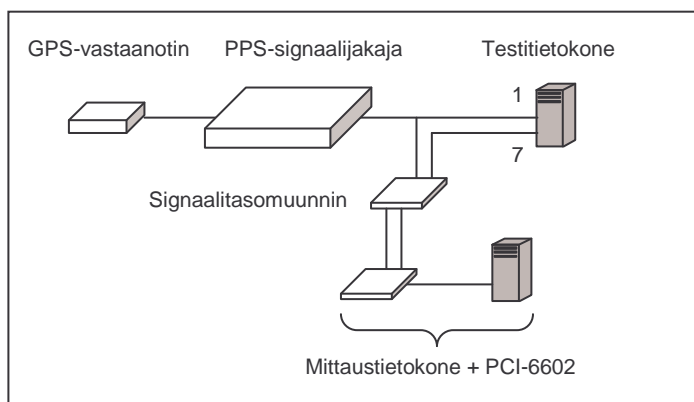
Systeemikellon aikapoikkeaman mittaaminen olisi mahdollista siten, että tietokone generoisi pulssin ulos sarjaportista satunnaisella ajanhetkellä ja tallennettaisiin muistiin systeemikellon ajan pulssin lähtöhetkellä. Aikapoikkeaman mittaamiseksi tarvittaisiin kuitenkin mittalaite, joka kykenee mittaamaan pulssin saapumishetken suhteessa UTC-aikaan. Tämä edellyttäisi, että kyseinen mittalaite synkronoituisi suoraan GPS-vastaanottimeen ja ylläpitäisi näin omaa tarkkaa kelloaan.

Perusominaisuuksilla varustetut mittalaitteet kykenevät kuitenkin mittaamaan vain suhteellista aikaeroa kahden tapahtuman välillä, eikä pulssin saapumisajankohdan määrittäminen näin ollen ole mahdollista. Olisi tietysti mahdollista verrata tietokoneen generoimaa pulssia GPS-vastaanottimen muodostamaan PPS-signaaliin, mutta pulssin generoimisajankohdan luotettava määrittäminen ei välttämättä ole mahdollista.

Vastaava virhemahdollisuus sisältyy mittaukseen, jossa systeemikellon tasasekunnilla generoitaisiin pulssi ulos sarjaportista ja näin ollen voitaisiin suoraan vertailla kahden tietokoneen systeemikellojen poikkeamaa esimerkiksi oskilloskoopin avulla. Määritettyyn pulssin generoimisajanhetkeen sisältyisi kuitenkin paljon epävarmuutta prosessointiviiveen ja prosessien vuorottelun seurauksena.

Näin ollen ainoa luotettava mittausmenetelmä tietokoneiden välisen suhteellisen aikapoikkeaman määrittämiseen on estimoida PPS-signaalin prosessointiviivettä RFC-2783 mukaisella PPS-pulssin takaisin kaiuttamisella. Mittaamalla tietokoneeseen sisään menneen ja sieltä takaisin kaiutetun pulssin välinen aikaero, saadaan yläraja systeemikellon absoluuttiselle aikapoikkeamalle.

Kun absoluuttiseen aikapoikkeamaan lisätään NTP:n tilastotiedoista saatava aikapoikkeama, voidaan estimoida yläraja tietokoneiden väliselle suhteelliselle aikapoikkeamalle. Kaiutusviiveen mittaamiseen käytetty mittausjärjestely on esitetty kuvassa 5.8.



Kuva 5.8. Kaiutusviiveen mittausjärjestely.

GPS-vastaanottimen muodostama PPS-signaali etenee signaalijakajan kautta testitietokoneen sarjaportin 1-nastaan. Havaitessaan pulssin tietokone suorittaa pulssin aikaleimauksen ja kaiuttaa pulssin takaisin nastan 7 välityksellä.

Kaiutusviiveen mittaamiseksi hankittiin National Instruments PCI-6602 -laskuri, joka kykenee mittaamaan pulssien väliaikaa 12,5 ns resoluutiolla. Laskuri koostuu tietokoneen PCI-väylään liitettävästä kortista sekä ulkoisesta kytkentäalustasta, joka on yhteydessä korttiin 64-johtimisella häiriösuojatulla liitoskaapelilla. Kytkentäalusta sisältää ruuvi-liittimet, joihin voidaan kytkeä laskurille syötettävät signaalit.

PCI-6602 sisältää kahdeksan rinnakkaista 32-bittistä laskuria, joilla voidaan samanaikaisesti mitata pulssien väliaikaa. PCI-6602 kykenee kuitenkin käsittelemään vain TTL-tasoisia jännitteitä, joten erilliselle kytkentäalustalle rakennettiin 20-kanavainen signaalitasomuunnin, joka muuntaa sarjaportissa käytettävät jännitteet TTL-tasoisiksi.

Signaalitasomuunnin sisältää 10 kappaletta RS232-linjaohjaimia, joissa kussakin on kaksi vastaanottokanavaa. Signaalitasomuuntimen tarvitsema käyttöjännite muodostetaan laboratoriovirtalähteen avulla. Signaalitasomuuntimen ulostuloporttien välinen ajastus suhteessa PPS-signaalijakajan ulostulosignaaliin mitattiin oskilloskoopin avulla. Saadut mittaustulokset on esitetty liitteen E kuvassa E2.

Tulosten perusteella signaalitasomuuntimen ulostuloporttien välinen hajonta on noin 150 ns. Hajonta johtuu linjaohjainten valmistustoleransseista ja häiriöistä, joita kytkentäalustalle hyppylangoilla toteutetut johdinvedot aiheuttavat. Signaalitasomuuntimen aiheuttama etenemisviive on hyvin pieni eikä viiveen suuruudella ole vaikutusta kaiutusviiveen mittaamiseen, koska sekä suoraan etenevä että kaiutettu pulssi kokevat saman viiveen.

Kaiutusviiveen mittaamisen lisäksi NTP:n lähdekoodia muutettiin siten, että NTP-prosessi tallentaa jokaisen PPS-pulsille määritetyn aikaleiman lokitiedostoon. Tämä mahdollistaa, että voidaan tarkastella miten pulssit todellisuudessa saapuvat tietokoneeseen ilman NTP-prosessin suorittamaa keskiarvoistusta ja suodatusta.

### *5.5.2. Suhteellisen poikkeaman yläraja*

Liitteessä J on havainnollistettu kaiutusviiveen ja PPS-pulsseille määritettyjen aikaleimojen käyttäytymistä, kun tietokoneita yhdistävän Ethernet-segmentin liikenne on vähäistä. Verkossa liikkuu pääasiassa vain NTP-paketteja sekä HTTP (Hypertext Transfer Protocol) -paketteja, jotka aiheutuvat lokitiedostojen siirtämisestä muilta palvelimilta pääpalvelimelle (Intel Celeron 300 MHz) *wget*-ohjelman avulla 10 sekunnin välein.

Kuvassa J1 on esitetty tulokset Intel Celeron 300 MHz –kokoonpanolle ja kuvassa J2 Intel Celeron 466 MHz –kokoonpanolle. Tietokoneiden PPS-pulsseille määrittämien aikaleimojen poikkeamassa systeemikellon tasasekunnista on nähtävissä huomattava ero. Kaiutusviive on Celeron 300 MHz –kokoonpanossa keskimäärin 20  $\mu$ s ja Celeron 466 MHz –kokoonpanossa keskimäärin 15  $\mu$ s. Kaiutusviiveen ero johtuu kokoonpanojen välisestä erosta suorituskyvyssä ja kuormituksessa.

NTP-prosessin havaitsema systeemikellon aikapoikkeama noudattaa PPS-pulssien aikaleimojen keskiarvoa. Kun aikaleimat jäävät tasasekunnista jälkeen NTP-prosessi tulkitsee, että systeemikello on referenssikelloa edellä, mikä näkyy kuvissa aikapoikkeaman negatiivisena arvona.

Aikaleimojen poikkeaman suuret muutokset aiheutuvat systeemikellon aikapoikkeamasta tai systeemikellon lukemisessa esiintyvistä virheistä. Aikaleimojen poikkeamassa havaittava satunnainen kohina aiheutuu PPS-pulssin tai systeemikellon lukemiseen käytettävän systeemikutsun prosessointiviiveen vaihteluista, jotka NTP-prosessi suodattaa tehokkaasti pois määrittäessään systeemikellon aikapoikkeamaa.

Molempien koneiden systeemikellon aikapoikkeama kasvaa UTC-ajanhetkellä 21:00. Aikapoikkeaman kasvaminen johtuu ajoitetuista tehtävistä (cronjob), jotka kuormittavat prosessoria ja suoritetaan 24:00 paikallista aikaa (UTC+3).

Kuvissa J3 ja J4 on esitetty 72 sekunnin mittainen tarkastelu samasta mittauksesta. Intel Celeron 300 MHz –kokoonpanossa aikaleimojen poikkeama on selvästi riippuvainen kaiutusviiveestä. Kaiutusviiveen kasvaessa aikaleima jää enemmän jälkeen systeemikellon tasasekunnista. Stabiilimman systeemikellon omaavassa Intel Celeron 466 MHz –kokoonpanossa riippuvuus ei tosin ole yhtä voimakasta.

Kaiutusviiveessä esiintyy satunnaisesti yli sekunnin piikkejä, jotka aiheutuvat siitä, että tietokone ei jostakin syystä ole kyennyt kaiuttamaan pulssia takaisin vaan laskuri on pysähtynyt vasta sekunnin kuluttua kaiutettavalla pulssilla. On mahdollista, että kaiutusviiveen pitenemisen aiheuttava pulssi on jäänyt tietokoneelta kokonaan huomaamatta tai se on hylätty kokonaan. Parantaakseen systeemikellon stabiiliutta NTP-prosessi hylkää PPS-pulssit, jotka poikkeavat liikaa keskimääräisestä tasosta.

Molempien tietokoneiden kohdalla havaitaan lisäksi odottamaton ilmiö, kun kaiutusviive kasvaa kuorman seurauksena 20 µs suuremmaksi. Kuormitus toistuu 10 sekunnin välein ja aiheutuu edellä mainitusta lokitiedostojen siirrosta palvelimien välillä.

Ilmiö ei selity sillä, että kuorman seurauksena PPS-pulssin aikaleimauksessa olisi enemmän viivettä, koska tällöin aikaleiman poikkeaman tasasekunnista tulisi muuttua päinvastaiseen suuntaan. Tällöin ainoaksi vaihtoehdoksi jää, että prosessorin kuormituksen kasvaminen häiritsee systeemikellon päivitystä.

Kuvissa J3 ja J4 kaiutusviivettä ja aikaleiman poikkeamaa tasasekunnista on mitattu sekunnin välein. Kun systeemikellon päivitystaajuus on 100 Hz, näytepisteiden välissä systeemikellon päivitys tapahtuu 100 kertaa. Kaiutusviiveen kasvaminen siirtää PPS-pulssin aikaleimaa lähes 17  $\mu$ s aikaisemmaksi ensimmäisellä näytteellä, mutta tämän jälkeen aikaleiman poikkeama mukailee jälleen kaiutusviivettä keskimääräistä alhaisemmalla tasolla. Kun prosessorin kuormitus pienenee, palaa aikaleiman poikkeama alkuperäiselle tasolle.

On mahdollista, että prosessorin äkillinen kuormittuminen viivästä systeemikellon päivitystä, jolloin se päivittyy hetkellisesti liian hitaasti ja jää jälkeen referenssikellosta. Kun käyttöjärjestelmä sopeutuu kuormitukseen, systeemikelloa päivitetään taas oikeassa tahdissa ja PPS-pulssien aikaleimat mukailevat kaiutusviivettä. Teoria ei kuitenkaan pysty selittämään, miksi systeemikellon päivitys näyttää hetkellisesti nopeutuvan prosessorin kuormituksen pienentyessä.

Liitteessä K on esitetty mittaustulokset tilanteessa, jossa tietokoneita ja niitä yhdistävää Ethernet-segmenttiä pyrittiin kuormittamaan mahdollisimman paljon iperf-ohjelman avulla. Iperf-ohjelma on tarkoitettu verkon siirtokapasiteetin mittaamiseen, mutta käytettäessä UDP-protokollaa ja pienintä mahdollista UDP-paketin kokoa voidaan tehokkaasti kuormittaa verkon päätelaitteita.

Verkon käyttöaste jää kuitenkin alhaiseksi, koska lähetettävien ja saapuvien pakettien prosessointi kuormittaa prosessoria niin paljon, että keskimääräinen siirtonopeus on ohjelman tulostamien lokitietojen perusteella vain 1 Mbit/s molempiin suuntiin vaikka käytettävissä oleva kapasiteetti olisi 100 Mbit/s.

Voidaan olettaa, että verkossa välitettävän liikenteen määrä on niin vähäinen, että se ei oleellisesti vaikuta verkon läpi vaihdettavien aikaleimojen tarkkuuteen. Näin ollen suurimmat systeemikellojen aikapoikkeamaan aiheutuvat välillisesti prosessorin kuormituksen kasvamisen seurauksena.

Prossessorin kuormituksen lisääntyminen kasvattaa PPS-pulssin prosessointiviiveen vaihtelua ja edelleen aikaleimojen poikkeamassa havaittavaa värinää. Yhä suurempi osa PPS-pulsseista jää havaitsematta tai niitä ei ainakaan kyetä kaiuttamaan takaisin. NTP-prosessi kykenee kuitenkin edelleen tehokkaasti suodattamaan näytteitä määrittäessään systeemikellon aikapoikkeamaa.

Systeemikellon aikapoikkeama on lisääntyneen kuormituksen seurauksena enemmän kohinan kaltaista eikä siinä esiinny enää niin selvää jaksollisuutta ympäristön lämpötilavaihtelun seurauksena. Tämän lisäksi aikapoikkeaman vaihteluväli on lähes kaksinkertainen kuormittamattomaan tilanteeseen verrattuna. Vaihteluvälin lisääntyminen voi olla seurausta systeemikellon epästabiliimmasta käyttäytymisestä tai siitä, että referenssikellon lukemisessa esiintyy epäsäännöllistä viivettä.

NTP-prosessin määrittämää aikapoikkeamaa voidaan pitää systeemikellon absoluuttisen aikapoikkeaman alarajana sillä oletuksella, että PPS-signaalin prosessoinnissa ei esiintyisi lainkaan viivettä. Aikapoikkeaman yläraja saadaan lisäämällä kaiutusviive NTP-prosessin määrittämään aikapoikkeamaan, koska pulssin kaiuttaminen suoritetaan vasta, kun PPS-pulssi on havaittu ja sille on määritetty aikaleima jatkoprosessointia varten. Näin ollen kaiutusviive on yläraja absoluuttiselle poikkeamalle, joka voi olla NTP-prosessin määrittämässä aikaleimoissa.

NTP-prosessin määrittämässä aikapoikkeamassa on näytepisteitä noin 64 sekunnin välein, kun taas kaiutusviive on mitattu joka sekunti. Tämän seurauksena yksittäiseen aikapoikkeaman arvoon lisättiin aikapoikkeaman keskiarvo edeltäviltä 64 näytteeltä, jotta aikapoikkeaman näytepisteen kohdalla oleva kaiutusviiveen arvo ei yksinään määräisi aikapoikkeaman ylärajaa.

Liitteessä L on esitetty edellä mainitun menettelyn avulla saadut ylä- ja alarajat neljän tietokoneen aikapoikkeamalle suhteessa UTC-aikaan. Kaiutusviiveen mittaamiseksi FreeBSD-käyttöjärjestelmällä varustetun tietokoneen PPS-signaali jouduttiin kytkeämään rinnakkaisporttiin, koska kyseinen käyttöjärjestelmä ei tue kaiutustoimintoa sarjaportissa.

Kuormitetussa tilanteessa (Kuva L2) FreeBSD-käyttöjärjestelmällä varustettu tietokone (IP-osoite 10.0.0.3) ei kykene rinnakkaisportin välityksellä vastaanottamaan PPS-signaalia, vaan synkronointi menetetään noin kello 16:40 UTC-aikaa ja tietokone synkronoituu ensisijaiseen NTP-palvelimeen (IP-osoite 10.0.0.1) verkon yli vaihdettavien aikaleimojen avulla.

Verkon kuormitus aloitettiin 12:30 UTC-aikaa, joten PPS-signaali todetaan käytökelvottomaksi noin neljä tuntia kuormituksen aloitushetken jälkeen ja aikapoikkeama kasvaa suurimmillaan yli 30 millisekuntiin. NTP:n lokitietojen perusteella PPS-signaali hylättiin, koska siinä esiintyi liikaa värinää.

Kun vastaava kuormitus toistettiin käytettäessä myös FreeBSD:n kohdalla sarjaporttia PPS-signaalin vastaanottoon, säilyi synkronointi kaikkien tietokoneiden kohdalla käyttöjärjestelmästä riippumatta. Synkronoinnin menettäminen rinnakkaisporttia käytettäessä voi johtua rinnakkaisportin alhaisesta keskeytysprioriteetista tai rinnakkaisportin epä säännöllisestä prosessointiviiveestä FreeBSD-käyttöjärjestelmässä.

Prosessointiviiveen analysointi oli kuitenkin mahdotonta, koska lähes kaikki PPS-pulsit jäivät kaiuttamatta todennäköisesti hylkäämisen seurauksena. Linux-käyttöjärjestelmän osalta vastaavan käyttäytymisen olemassaoloa ei voitu testata, koska kyseinen käyttöjärjestelmä ei oletusarvoisesti tue PPS-signaalin prosessointia rinnakkaisportissa.

Tietokoneiden välisen suhteellisen poikkeaman yläraja saadaan määrittämällä kuvien L1 ja L2 kuvaajille ylä- ja alareunan verhoikäyrä. Ylä- ja alareunan verhoikäyrien välinen erotus ajan funktiona on tietokoneiden välisen suhteellisen poikkeaman yläraja. Edellä mainitut verhoikäyrät ja niiden avulla saatu suhteellisen poikkeaman yläraja on esitetty liitteessä M.

Kuvan M1 perusteella poikkeaman yläraja on 10 tunnin mittausjakson aikana suurimmillaan hieman yli 40  $\mu$ s, kun verkon kuormitus on vähäistä. Kuvan M2 perusteella verkon ja erityisesti tietokoneiden kuormittaminen nostaa poikkeaman ylärajan suurimmillaan lähes 60  $\mu$ s:iin.

Liitteessä N on esitetty Matlab-ohjelman avulla määritetyt jakaumat ja kertymäfunktiot normaalitilassa ja kuormitetussa verkossa. Kuormituksen ollessa vähäistä suhteellisen poikkeaman keskiarvo on 21,2  $\mu$ s ja kuormitetussa tilanteessa 23,6  $\mu$ s. Kertymäfunktioiden perusteella voidaan todeta, että tietokoneiden suhteellinen aikapoikkeama on alle 29  $\mu$ s 90 % ajasta ja alle 32  $\mu$ s 99 % ajasta kuormituksen ollessa vähäistä. Kuormitetussa tilanteessa suhteellinen aikapoikkeama on alle 35  $\mu$ s 90 % ajasta ja alle 46  $\mu$ s 99 % ajasta.

### 5.5.3. Suhteellisen poikkeaman arviointi PING-kyselyn avulla.

Tietokoneiden suhteellista poikkeamaa voidaan arvioida käyttämällä hyväksi Tcpdump-pakettikaappausohjelmaa ja PING-kyselyä. Neljä testitietokonetta liitettiin samaan Ethernet-kytkimeen ja viidennellä tietokoneella lähetettiin PING-kyselyä aliverkon osoitteeseen, jota verkossa ei ole käytössä. Tämän seurauksena kaikille koneille saapuu säännöllisesti ARP (Address Resolution Protocol) –kyselyjä.

Kun saapuvat ARP-kyselyt kaapataan Tcpdump-ohjelmalla, joka määrittää niille myös saapumisajankohdan suhteessa systeemikelloon, voidaan vertailla tietokoneiden määrittämien aikaleimojen keskinäisiä suhteita ja saada käsitys systeemikellojen välisestä aikapoikkeamasta.

Mittaus ei anna täysin luotettavaa kuvaa tietokoneiden välisestä aikaerosta, koska mittaukseen sisältyy useita virhelähteitä, jotka eivät varsinaisesti ole sidoksissa systeemikelloon. Tietokoneiden yhteen liittämiseen käytetty kytkin ei välttämättä ole täysin symmetrinen, vaan sen portit voivat muodostaa puumaisen rakenteen, jonka eri haaroja palvellaan vuorotellen tai muutaman bitin verran viivästetysti.

Tämän lisäksi tietokoneissa käytetyt verkkokortit puskuroivat tietyn määrän vastaanotettua dataa ennen kuin aiheuttavat keskeytyksiä prosessoreille. Aikaleimaukseen käytettävää Tcpdump-ohjelmaa suoritetaan käyttäjätilassa, joten paketin saapumisajan määrittämisessä voi olla vaihtelevaa tai epäsäännöllistä viivettä. Tcpdump tukee myös kernel-tason aikaleimausta, mikäli se on käyttöjärjestelmän ominaisuuksien perusteella mahdollista.

Liitteessä O on esitetty mittausjärjestelyn avulla saadut tulokset 15 000 peräkkäiselle ARP-paketille, joiden saapumisväli on keskimäärin 1,5 sekuntia. Kuvan O1 perusteella aiemmin määritetty suhteellisen aikapoikkeaman yläraja voidaan todentaa myös käytännön mittauksen avulla. Aikaleimojen poikkeama on pääsääntöisesti 20  $\mu$ s tasolla, mutta kasvaa pahimmillaan yli 25 ms:iin, kuten kuvasta O2 voidaan havaita.



## 6. Johtopäätökset

### 6.1. Tulosten arviointi

Tietokoneen kelloa ei koskaan tulla saamaan absoluuttisen tarkaksi niin kauan kuin kellonajan ylläpito perustuu pelkkien kellokeskeytyksien käyttöön. Kellon ylläpidon toteuttaminen komponenttitasolla ilman käyttöjärjestelmää tekisi ylläpidettävästä ajasta todennäköisesti tarkemman ja stabiiliudeltaan paremman, mutta kellon lukemista ei siltikään pystyttäisi toteuttamaan täysi virheettömästi.

Tietokoneen systeemikelloa ohjaavan kiteen huono stabiilius ja vahva lämpötilariippuvuus ovat suurimmat tarkkuutta rajoittavat tekijät käyttöjärjestelmä poisluettuna. Käyttöjärjestelmän keskeinen rooli kellonajan ylläpidossa ja päivittämisessä muodostaa useita virhelähteitä kellonajan päivitykseen ja lukemiseen.

Systeemikellon päivitystä ohjaavan ajastimen aiheuttamaa keskeytystä ei välttämättä pystytä palvelemaan välittömästi vaan kellon päivittäminen viivästyy. Kellon aikaa luettaessa systeemikutsun suorituksen vaatima aika aiheuttaa oman virheensä palautetun kellonajan tarkkuuteen. PPS-signaalin tai aikaleiman prosessointiviive aiheuttaa absoluuttista poikkeamaa suhteessa referenssikelloon.

Suoritettaessa tietoverkon liikenteen analysointia ei paketeille määritettyjen aikaleimojen tarkkuus ole välttämättä lähelläkään systeemikellon tarkkuuden edellyttämää tasoa. Mikäli pakettien aikaleimauksesta huolehtiva prosessi suoritetaan käyttäjätilassa, voi saaduissa aikaleimoissa olla huomattavaa virhettä systeemikellon lukemisviiveen vaihtelun seurauksena tai verkkoajurin viiveiden johdosta.

Joka tapauksessa saavutettua synkronointitarkkuutta (35  $\mu$ s 90 % ajasta 10 tunnin mittausjaksolla kuormitetussa verkossa) voidaan pitää hyvänä saavutuksena, kun huomioidaan kellonajan ylläpidon toteutus ja lukuisat virhettä aiheuttavat tekijät. Yksittäisen tietokoneen kohdalla saavutettu  $\pm 1$   $\mu$ s suhteellinen synkronointitarkkuus lähestyy käytännössä tasoa, johon peruskokoonpanon omaavalla tietokoneella voidaan parhaimmillaan yltää.

Järjestelmän kustannukset ovat GPS-vastaanottimen osalta noin 1000 € ja PPS-signaalijakajan osalta lähes saman verran. Mikäli työvoimakustannukset jätetään huomiotta, on järjestelmän kokonaiskustannus noin 100 € yhtä synkronoitavaa tietokonetta kohti. PPS-signaalijakajan avulla toteutettu synkronointi on siis kymmenen kertaa halvempaa kuin jokaisen tietokoneen varustaminen omalla GPS-vastaanottimellaan. Alle sadan euron hintaisten GPS-vastaanottimien ominaisuudet eivät välttämättä ole enää riittäviä tietokoneiden tarkkaan synkronointiin.

Tietokoneiden välisen synkronoinnin tarkkuus oli lähtötilanteessa korkeintaan muutamien millisekuntien suuruusluokkaa, joten voidaan varmuudella puhua vähintäänkin satakertaisesta parannuksesta saavutetussa synkronointitarkkuudessa. Näin ollen kymmenkertaisen parannuksen saavuttaminen synkronointitarkkuudessa maksoi laitteiden osalta noin 1000 €. Kokonaisuudessaan toteutuneen järjestelmän hinta-laatusuhdetta voidaan pitää vähintäänkin hyvänä.

Tulee muistaa, että tavallista PC-tietokonetta ei ole suunniteltu tarkkaan ajan ylläpitoon tai verkossa välitettävän liikenteen luotettavaan analysointiin, vaan siitä on pyritty tekemään mahdollisimman monikäyttöinen laite joidenkin ominaisuuksien kustannuksella ja varsinkin kohtuullisilla valmistuskustannuksilla.

Markkinoilla on tarjolla erinäinen joukko suorituskykyisiä liikenneanalyysointilaitteita, jotka synkronoituvat suoraan esimerkiksi GPS-vastaanottimeen ja pystyvät suorittamaan pakettien aikaleimausta luotettavasti hyvin suurella tarkkuudella. Kyseiset laitteet on suunniteltu juuri tätä käyttötarkoitusta varten, jonka seurauksena ajan ja ajoituksen hallinta on toteutettu perinteisestä tietokoneesta poikkeavalla tavalla. Kyseisten laitteiden hankintahinta on vähintäänkin useita tuhansia euroja.

## 6.2. Mahdollisia sovelluskohteita

Synkronointijärjestelmän mahdollisia käyttökohteita rajoittaa eniten se, että yksittäisen järjestelmän avulla synkronoitavien tietokoneiden on sijaittava käytännössä samassa huoneessa. PPS-signaalin siirtäminen pitkiä matkoja sarjakaapelissa vaikuttaa saavutettavaan tarkkuuteen. Luonnollisesti voidaan rakentaa useita rinnakkaisia järjestelmiä erillään sijaitsevien tietokoneiden synkronoimiseksi. Toisaalta tietokoneen kellon synkronoiminen GPS-järjestelmän avulla ei sinällään ole mikään uusi keksintö.

Rinnakkaisten järjestelmien rakentaminen mahdollistaisi luotettavan verkkoliikenteen analysoinnin jopa maapallon vastakkaisilla puolilla sijaitsevien kohteiden välillä. GPS-järjestelmän maailmanlaajuisen saavutettavuuden ansiosta järjestelmän tuottaman ajastusinformaation voidaan olettaa olevan yhtä tarkka, suoritettiinpa mittauksia missä päin maailmaa tahansa napa-alueet ja esteiset maastot poisluettuna.

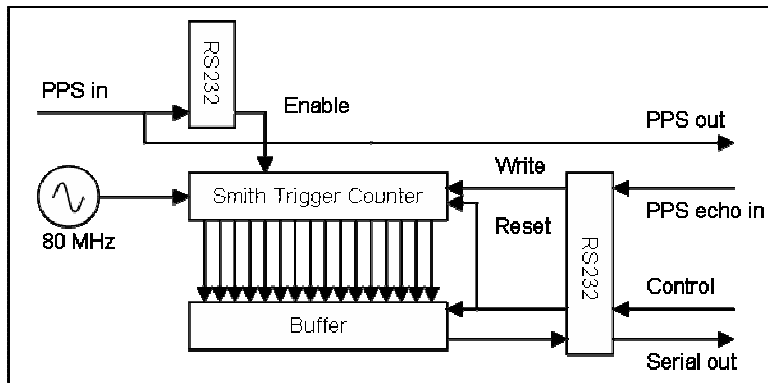
Mittaussovellusten lisäksi järjestelmä palvelee käytännössä mitä tahansa synkronointia tarvitsevaa sovellusta. Kysymykseen voisi tulla esimerkiksi jonkin toimenpiteen suorittaminen samanaikaisesti maantieteellisestä sijainnista riippumatta. Synkronoitavien laitteiden ei välttämättä tarvitse olla perinteisiä tietokoneita vaan riittää, että kyseiset laitteet kykenevät prosessoimaan GPS-vastaanottimelta saatavaa informaatiota.

### 6.3. Työn jatkokehitys

Saavutetun synkronointitarkkuuden parantamiseksi tulisi ensitilassa korvata tietokoneen kideoskillaattori esimerkiksi lämpötilastabiloidulla kiteellä, jonka lämpötilariippuvuus on pienempi. Systeemikellon ajan ja taajuuden korjaaminen käyttöjärjestelmän toimenpitein johtaa kuitenkin väistämättä siihen, että prosessorin kuormittaminen heikentää oleellisesti synkronoinnin tarkkuutta.

Tämän kompensoimiseksi tulisi kehittää kuvan 6.1. mukainen laskuriipiiri, joka mittaa tietokoneeseen sisään menevän ja ulos kaiutetun pulssin välisen aikaeron avulla PPS-pulssin prosessointiviiveen. Kun käyttöjärjestelmä on määrittänyt PPS-pulssin aiheuttamalle keskeytykselle saapumisajan, se lukee laskuriipiirin mittaaman aikaeron ja aikaistaa PPS-pulssille saatua saapumisaikaa mitatun aikaeron verran.

Laskuriipiirin fyysinen rakenne voisi olla PCI-kortin kaltainen, jolloin tarvittava 5 V käyttöjännite saataisiin PCI-väylästä. Tällöin myös datan lukeminen ja kortin logiikan ohjaaminen voitaisiin suorittaa PCI-väylän kautta. Tämä tosin vaatisi myös PCI-väyläohjaimen sijoittamisen laskuriipiirille.



Kuva 6.1. Pulssien välisen aikaeron mittaavan laskuriin periaate.

Laskuriin kuuluu esimerkiksi 80 MHz lämpötilastabiloidusta kiteestä, jonka ulostulosignaalin jaksoja lasketaan laskurin avulla. Käytettäessä 80 MHz kideä laskuriin resoluutio olisi 12,5 ns, joka on vähintäänkin riittävä tarkkuus kaiutusviiveen mittaamiseen. Laskurin arvosta saadaan suoraan pulssien väliaika kertomalla laskurin arvoa kiteen jaksonpituudella.

Tietokoneeseen sisäänmenevä PPS-pulssi kiertää laskuriin kautta ja käynnistää (Enable) laskurin. Tietokoneen havaittua PPS-pulssin se suorittaa aikaleimauksen ja kaiuttaa pulssin takaisin laskuriin (PPS echo in). Kaiutettu pulssi aiheuttaa laskurin arvon kirjoittamisen (Write) puskuriin. Tämän jälkeen tietokone voi asynkronisesti nollata laskurin ja käskä puskuria lähettämään siihen tallennetun arvon Control-linjan avulla. Sekunnin kuluttua saapuva uusi PPS-pulssi käynnistää laskurin jälleen uudelleen.

Tällä tavoin prosessointiviiveen aiheuttama virhe voisi pienentyä huomattavasti. Järjestelmä ottaisi huomioon myös mahdolliset hetkelliset muutokset PPS-pulssin prosessointiviiveessä. Tässä ratkaisumallissa on kuitenkin lukuisia kysymysmerkkejä.

Koska NTP-prosessi suodattaa PPS-pulssien aikaleimoja systeemikellon värinän pienentämiseksi, tulisi koko suodatustoiminto poistaa käytöstä. Nythän kaikkien aikaleimojen voitaisiin olettaa olevan täysin oikeassa ajassa eikä suodatusta enää tarvittaisi. Laskuriin huomioiminen synkronoinnissa edellyttäisi lisäksi tarvittavien hallintarutiinien implementointia NTP:n lähdekoodiin.

On kuitenkin otettava huomioon, että systeemikellon lukemisessa olisi edelleen vaihtelevaa viivettä PPS-pulssin saapumisajankohtaa määritettäessä. Näin ollen suodatuksen poistaminen kokonaan voisi aiheuttaa aivan uuden virhetekijän synkronointitarkkuuteen.

Systeemikellon taajuuden korjaaminen voitaisiin vaihtoehtoisesti suorittaa vaihelukituspiirillä, jolla lämpötilastabiloitu kideoskillaattori lukittaisiin tietyllä jakosuhteella suoraan GPS-vastaanottimeen ilman käyttöjärjestelmän mukanaoloa. Tässä yhteydessä NTP:n lähdekoodia tulisi muokata siten, että se ei missään vaiheessa pyri muuttamaan systeemikellon nimellistä taajuutta vaan suorittaa aikapoikkeaman korjaamisen askeltamalla kelloa eikä muuttamalla yksittäisen korotuksen suuruutta.

Nyhdän voitaisiin olettaa, että kun systeemikello on kertaalleen korjattu oikeaan aikaan ja sen päivitystaajuus on tarkalleen oikea, ei korjauksia jatkossa enää tarvita. Asia ei kuitenkaan ole näin yksinkertainen. Vaikka emolevyn ajastinpiiri tässä tilanteessa aiheuttaakin kellokeskeytyksen juuri oikealla taajuudella, ei silti voida olla varmoja, ennättääkö käyttöjärjestelmä välittömästi huomioimaan saapuneen kellokeskeytyksen ja päivittämään systeemikellon aikaa.

Tämän lisäksi referenssikellon antaman aikaleiman prosessoinnissa tai kellonajan vertailua suoritettavassa systeemikellon lukemisessa voi edelleen esiintyä viiveen seurauksena virhettä. Varsinaisen verkkoliikenteen analysoinnin yhteydessä suoritettava pakettien aikaleimaus tulisi toteuttaa mahdollisimman alhaisella tasolla tarkkuuden parantamiseksi. Tämä tarkoittaisi aikaleimauksen implementoimista esimerkiksi suoraan verkkokortin ajuriin.

Edellä mainituilla järjestelyillä asetettu yhden mikrosekunnin tavoite tietokoneiden välisessä synkronointitarkkuudessa voi hyvinkin olla saavutettavissa. Saavutettava tarkkuus paljastuu kuitenkin vain kokeilemalla vaihtoehtoisia menetelmiä.

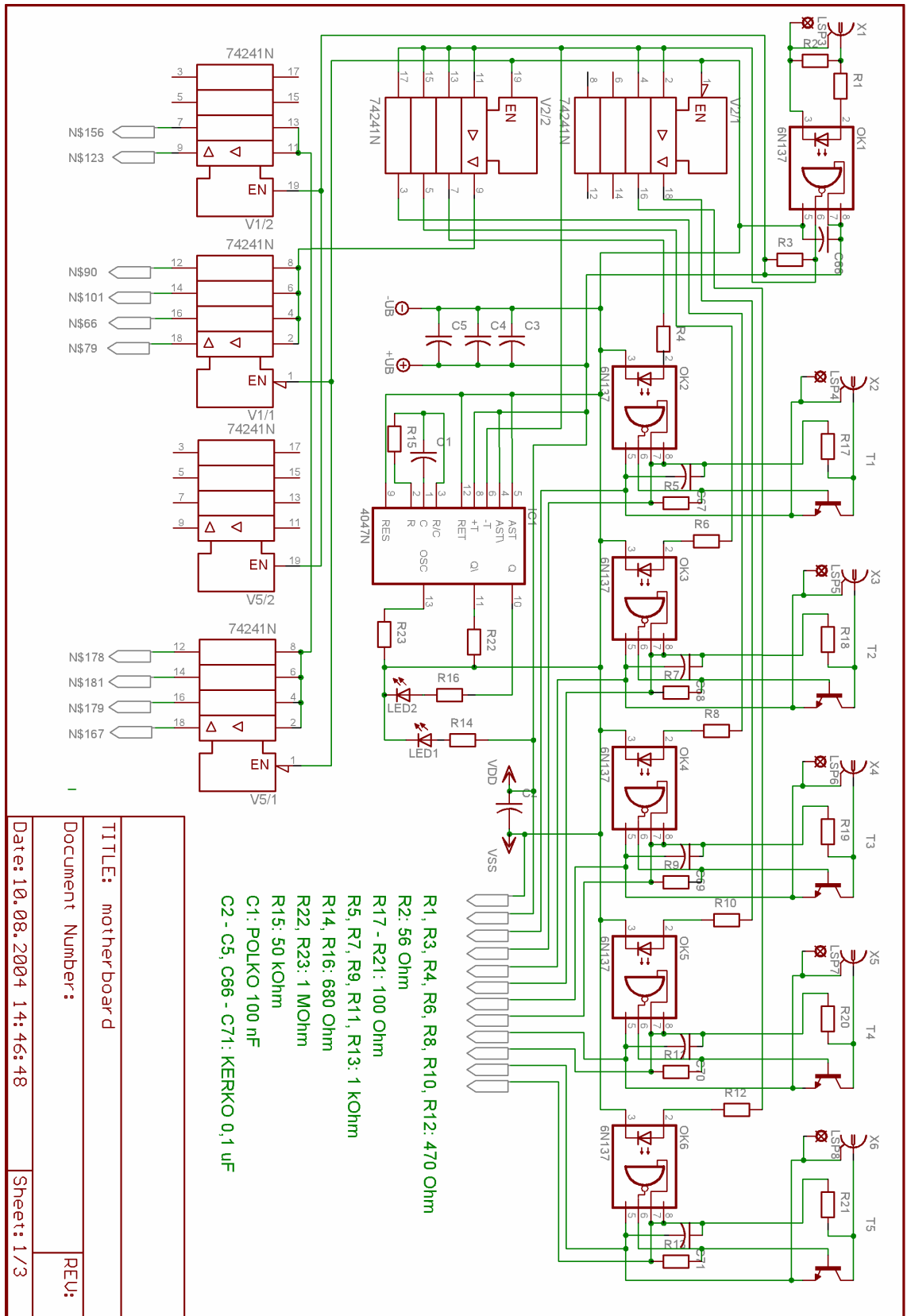
## Lähdeluettelo

1. H. Karttunen, K. J. Donner, P. Kröger, H. Oja, M. Poutanen, *Tähtitieteen perusteet*, neljäs laitos, Helsinki 2003, Tähtitieteellinen yhdistys Ursa ry, Gummerus, s. 63-74.
2. A. Leick, *GPS Satellite Surveying*, 2. painos, 1995, John Wiley & Sons, s. 26-38, 58-92, 543-537.
3. Mittatekniikan keskus, *Metrologia*, <http://www.mikes.fi>, viitattu 11.06.2004.
4. Väinö K. Lehtoranta, *AEL-INSKO-koulutus: GPS - maailmanlaajuinen satelliittinavigointijärjestelmä*, 1995.
5. U.S. Naval Observatory, *USNO GPS Timing Operations*, <http://tycho.usno.navy.mil/gps.html>, viitattu 09.06.2004.
6. M. Kihara, S. Ono, P. Eskelinen, *Digital Clocks for Synchronization and Communications*, Boston 2003, Artech House, Inc., s. 1-10.
7. S. Bregni, *Synchronization of Digital Telecommunications Networks*, West Sussex, Englanti 2002, John Wiley & Sons, Ltd., s. 21-28, 207-209, 229-238, 283-293.
8. J. Rutman, F. L. Walls, *Characterization of frequency stability in precision frequency sources*, Proceedings of the IEEE, vol. 79, no. 9, June 1991.
9. D. W. Allan, M. A. Weiss, J. L. Jespersen, *A frequency-domain view of time-domain characterization of clocks and time and frequency distribution systems*, Proceedings of the 45<sup>th</sup> Annual Symposium of Frequency Control, 1991.
10. IEEE Std. 1139, *Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology*, Revised version approved Oct. 20, 1988.
11. F. L. Walls, J.-J. Gagnepain, *Environmental Sensitivities of Quartz Oscillators*, IEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control, Vol. 39, No. 2, March 1992.
12. Network Time Protocol Project, Information on Time and Frequency Services, <http://www.eecis.udel.edu/~mills/ntp/qth.html>, viitattu 23.07.2004.
13. National Institute of Standards and Technology NIST, Physics Laboratory, Time & Frequency Division, <http://www.boulder.nist.gov/timefreq/>, viitattu 23.07.2004.
14. National Physical Laboratory NPL, <http://www.npl.co.uk/time/>, viitattu 23.07.2004.
15. P. Henttu, V. K. Lehtoranta, *GPS Maailmanlaajuinen satelliittinavigointijärjestelmä*, 1993, Naviprop Oy, s. 15-43, 48-59.
16. Russian Federation Ministry of Defence, Coordination Scientific Information Center, *Global Navigation Satellite System, GLONASS*, <http://www.glonass-center.ru>, viitattu 26.05.2004.
17. B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins, *GPS Theory and Practice*, Wien, Itävalta 1992, Springer-Verlag, s. 13-22.
18. S. Mäenpää, *AEL-INSKO-koulutus: GPS-järjestelmä ja GPS-paikanmäärittelyn perusteet*, Helsinki 1995, Geostar Oy.
19. R. Flyktman, *Inside PC - Laitetekniikka, Volume 1*, Helsinki 1999, IT Press, Gummerus Kirjapaino Oy, Jyväskylä, s. 136-171.

20. Micro-Star MS-6119 -emolevyn käyttöohje, [http://www.msi.com.tw/program/products/mainboard/mbd/pro\\_mbd\\_detail.php?UID=105](http://www.msi.com.tw/program/products/mainboard/mbd/pro_mbd_detail.php?UID=105), viitattu 30.07.2004.
21. Intel 82371 siltapiiri (PIIX4) -datalehti, <ftp://download.intel.com/design/intarch/datashts/29056201.pdf>, viitattu 30.07.2004.
22. Intel 82C54 Interval Timer -datalehti, <ftp://download.intel.com/design/archives/periphrl/docs/23124406.pdf>, viitattu 30.07.2004.
23. R. Love, *Linux Kernel Development*, 8 Sept 2003, SAMS, Developer Library Series s. 141-157.
24. Kernel-2.4.20 dokumentaatio, /usr/src/linux/Docomentation/kernel-time.txt.
25. IEEE RFC 1305, *Network Time Protocol (Version 3) Specification, Implementation and Analysis*, maaliskuu 1992.
26. IEEE RFC 1589, *A Kernel Model for Precision Timekeeping*, maaliskuu 1994.
27. D. L. Mills, *Internet Time Synchronization: The Network Time Protocol*, IEEE Transactions on Communications, vol. 39, No 10, October 1991.
28. NTP Official Documentation, *Mitigation Rules and the prefer Keyword*, <http://www.eecis.udel.edu/~mills/ntp/html/prefer.html>, viitattu 26.08.2004.
29. NTP-FAQ, How does it work?, <http://www.ntp.org/ntpfaq/NTP-a-faq.htm>, viitattu 23.08.2004.
30. IEEE RFC 2783, *Pulse-Per-Second API for UNIX-like Operating Systems, Version 1.0*, maaliskuu 2000.
31. Dallas Semiconductor, *App Note 083: Fundamentals of RS-232 Serial Communications*, [http://www.maxim-ic.com/appnotes.cfm/appnote\\_number/527](http://www.maxim-ic.com/appnotes.cfm/appnote_number/527), viitattu 07.09.2004.
32. FreeBSD Release Notes, <http://www.freebsd.org/releases/>, viitattu 08.09.2004.
33. NTP Official Documentation, *Monitorin Options*, <http://www.eecis.udel.edu/~mills/ntp/html/monopt.html>, viitattu 21.09.2004.

# Liitteet

## Liite A: Emolevyn periaatepiirros



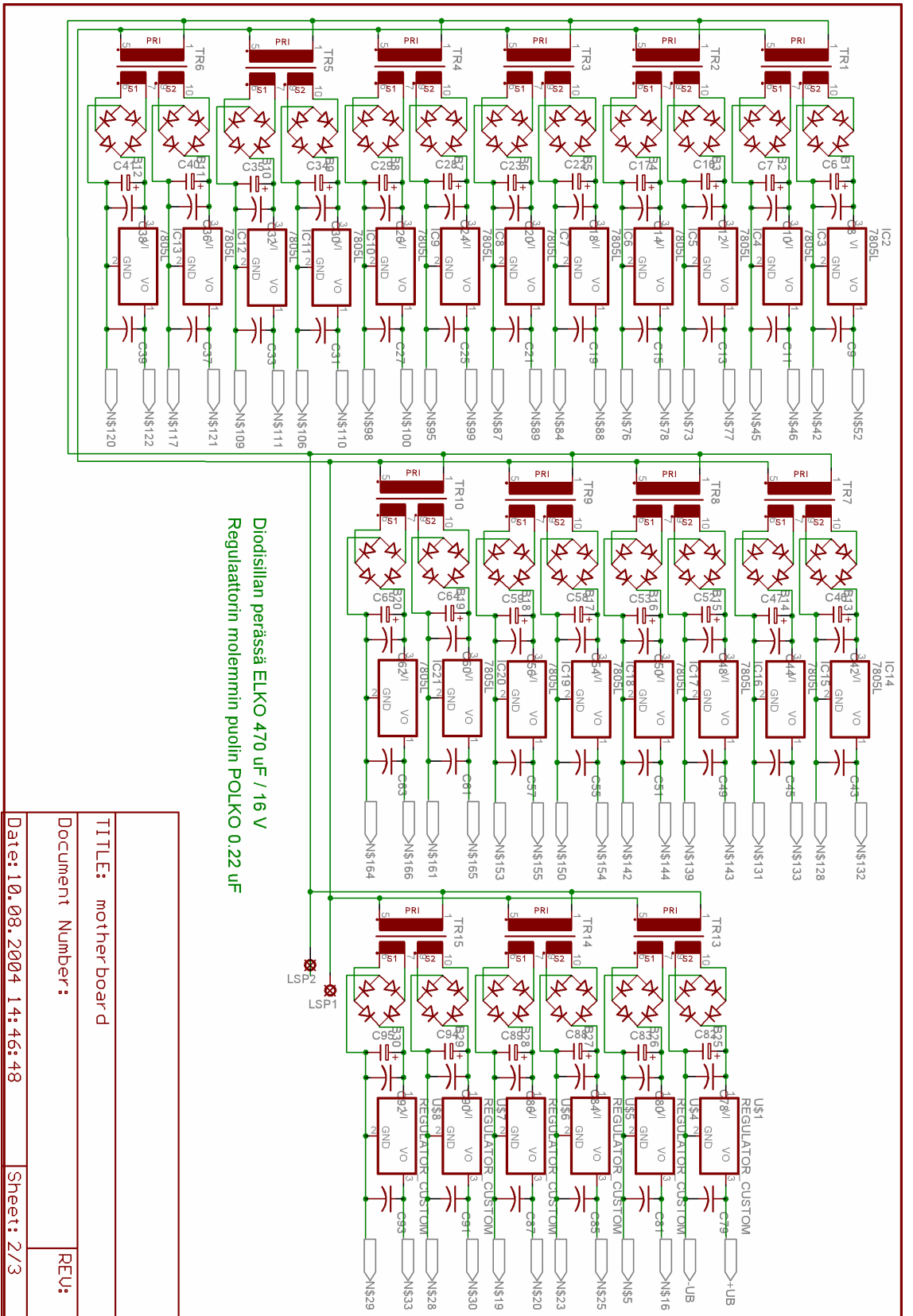
- R1, R3, R4, R6, R8, R10, R12: 470 Ohm
- R2: 56 Ohm
- R17 - R21: 100 Ohm
- R5, R7, R9, R11, R13: 1 Kohm
- R14, R16: 680 Ohm
- R22, R23: 1 Mohm
- R15: 50 Kohm
- C1: POLKO 100 nF
- C2 - C5, C66 - C71: KERKO 0,1 uF

TITLE: motherboard	
Document Number:	
Date: 10.08.2004 14:46:48	Sheet: 1/3
REV:	

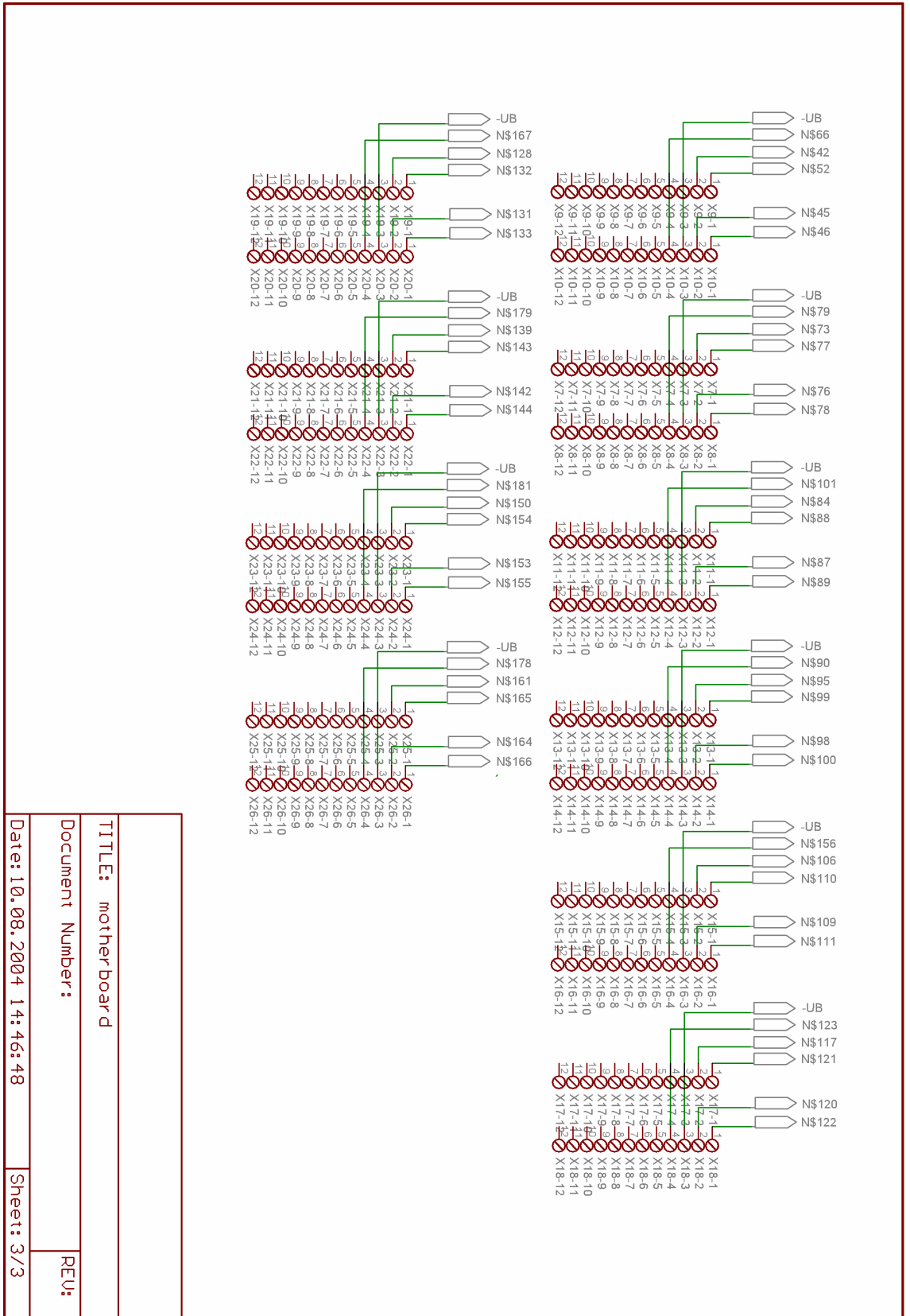
Puskuripiiriin V2 tulee olla piirroksesta poiketen invertoituva (74240).



Liite A: Emolevyn periaatepiirros

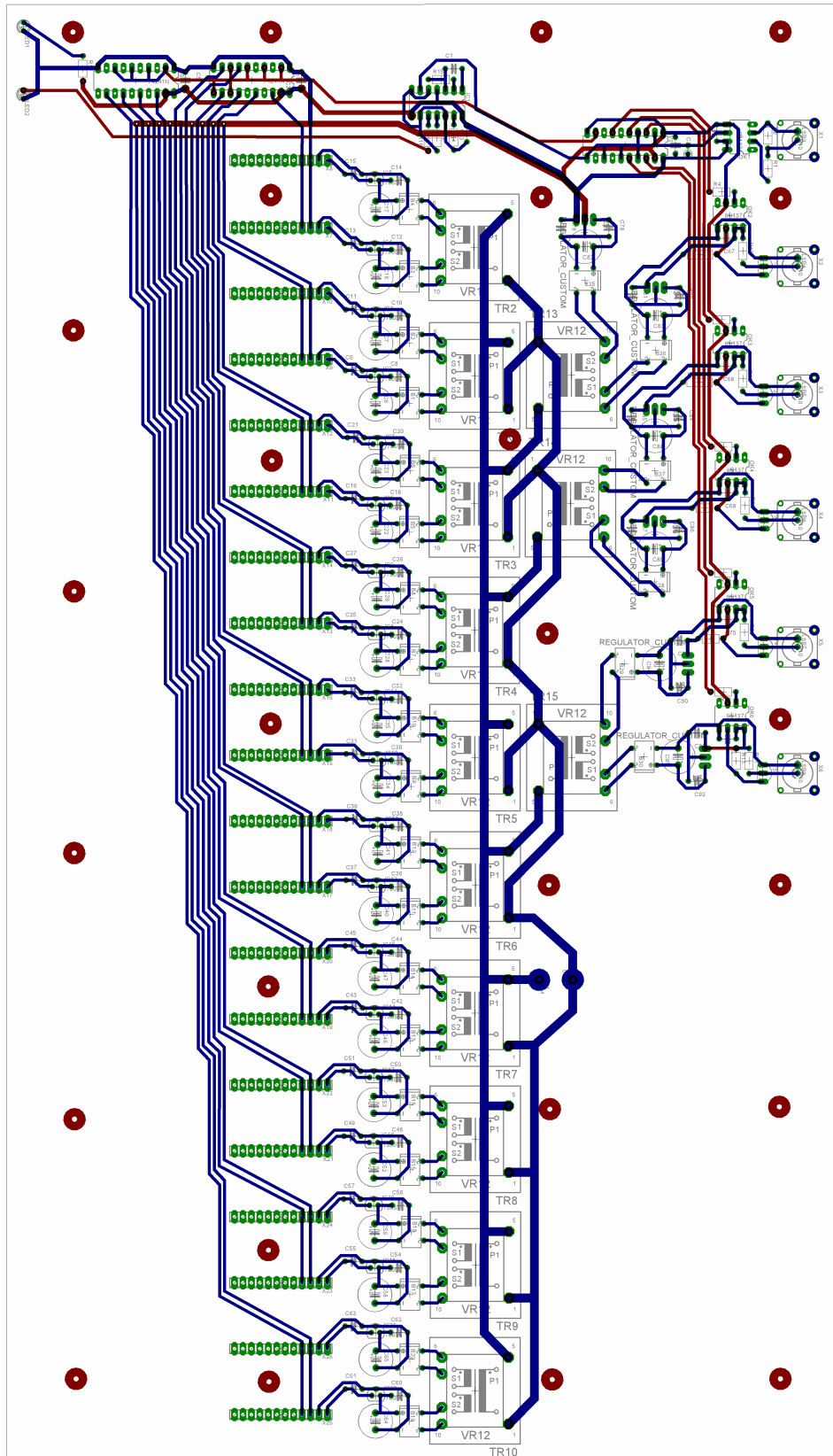


Liite A: Emolevyn periaatepiirros

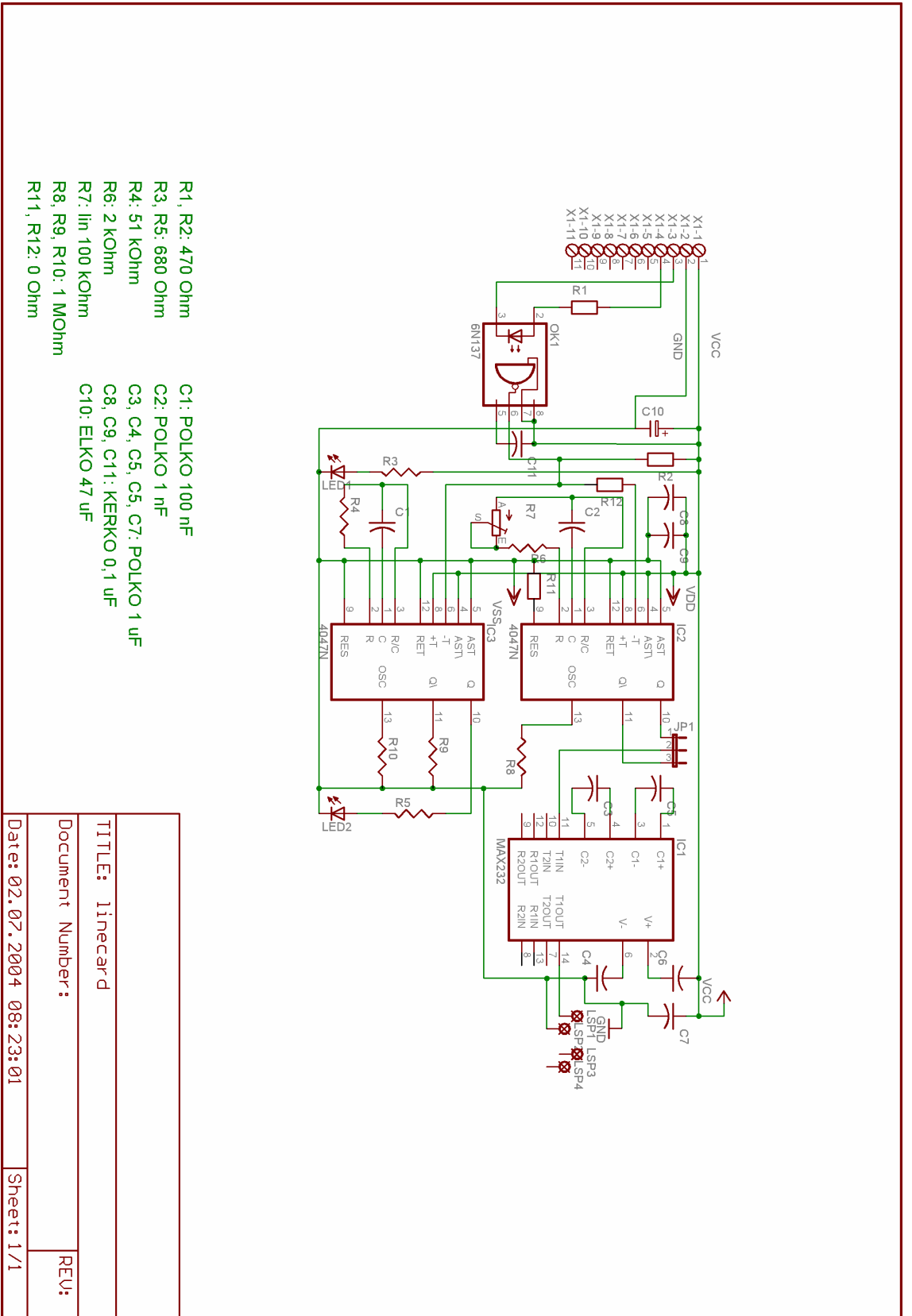


TITLE: motherboard	
Document Number:	REV:
Date: 10.08.2004 14:46:48	Sheet: 3/3

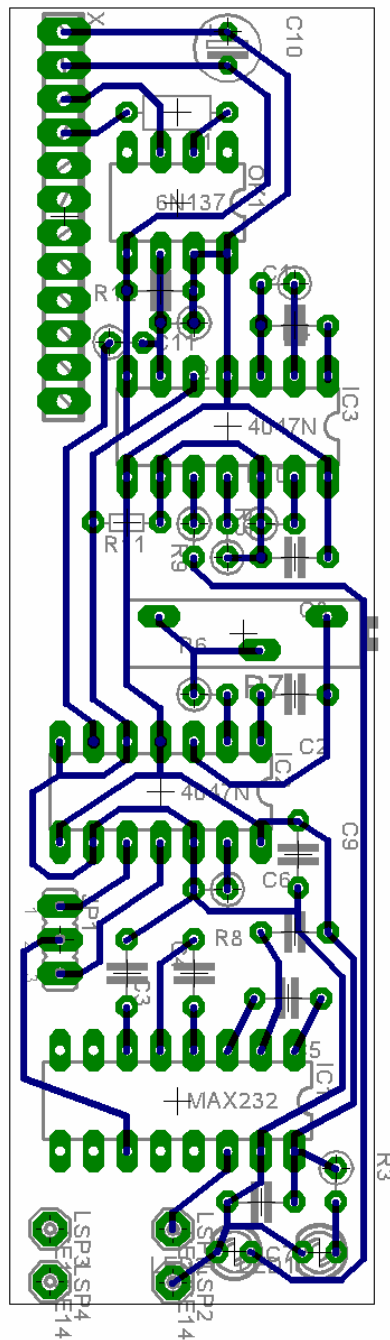
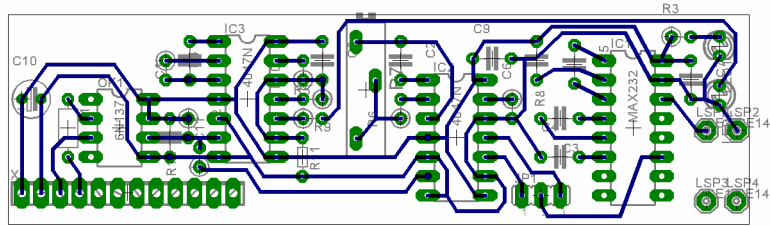
Liite B: Emolevyn komponenttien sijoittelu ja tarvittavat johdinvedot (250 mm x 437 mm)



Liite C: Linjakortin periaatepiirros

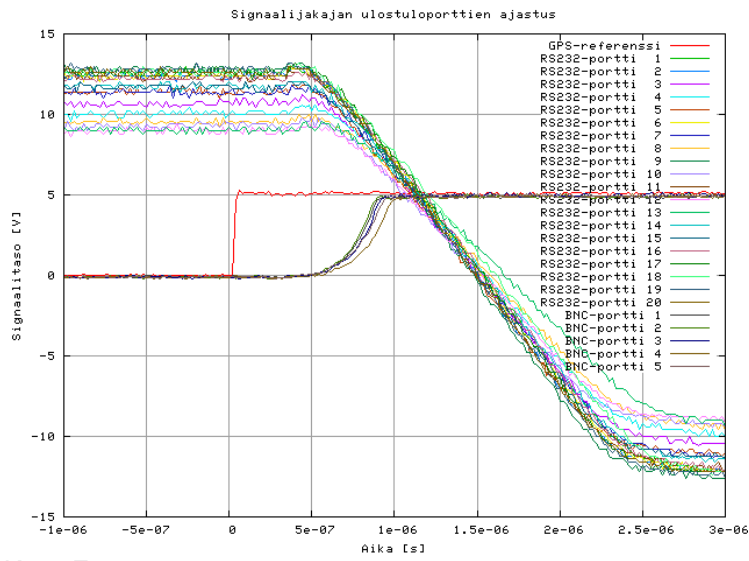


Liite D: Linjakortin komponenttien sijoittelu ja tarvittavat johdinvedot (28 mm x 98 mm)

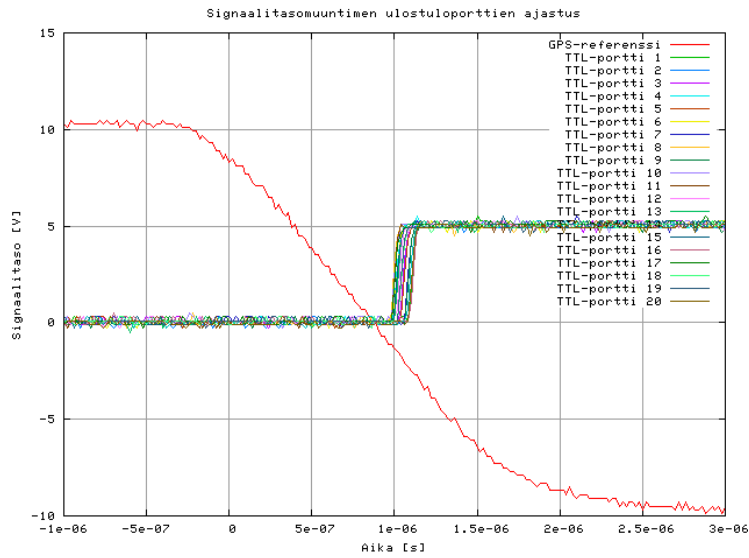


## Liite E: Rakennettujen laitteiden ulostuloporttien välinen ajastus

Kuva E1

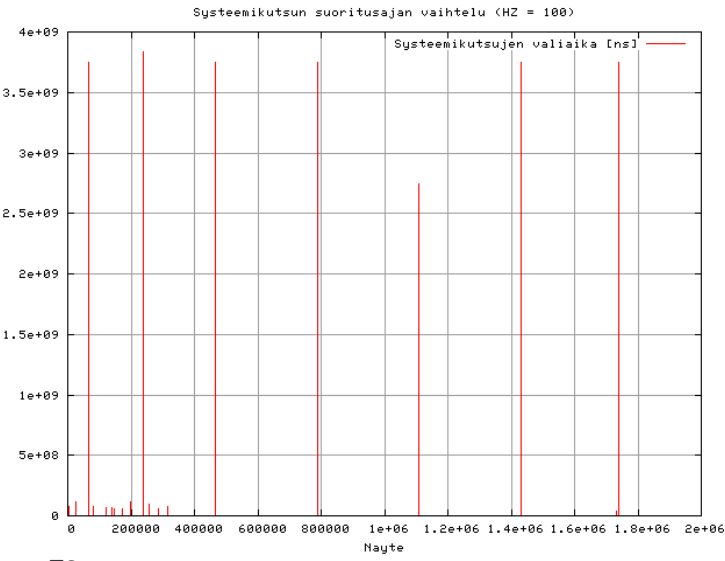


Kuva E2

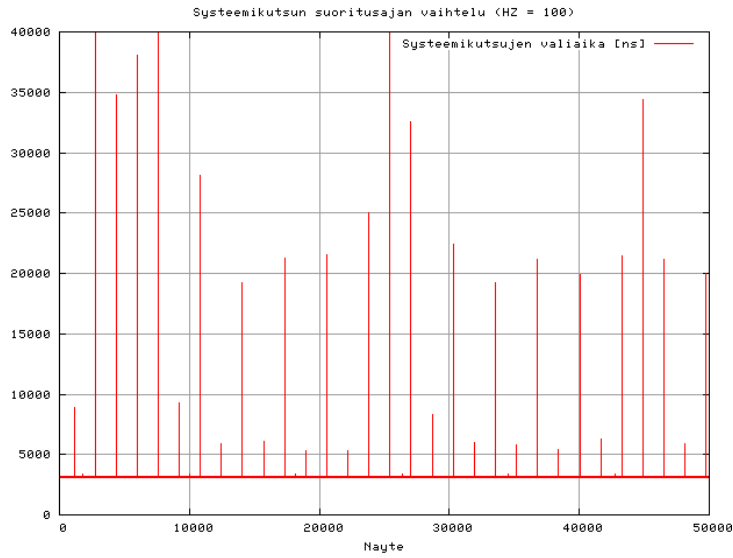


Liite F: Systemikutsun vaatima suoritus aika (Intel Celeron 300 MHz)

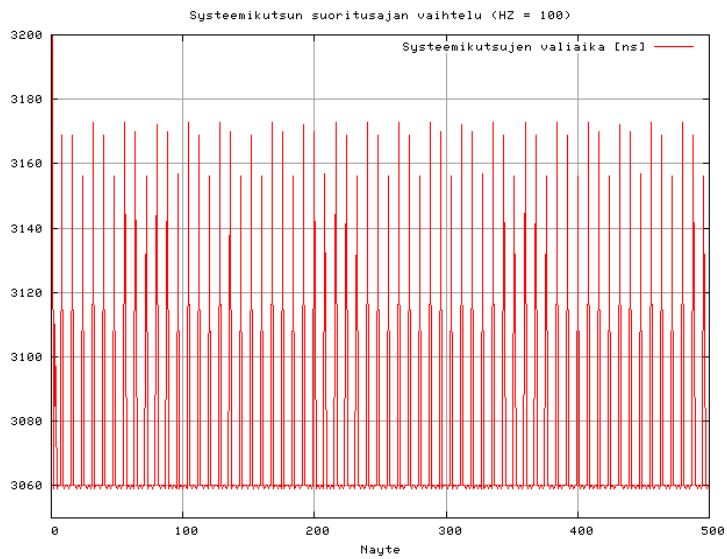
Kuva F1



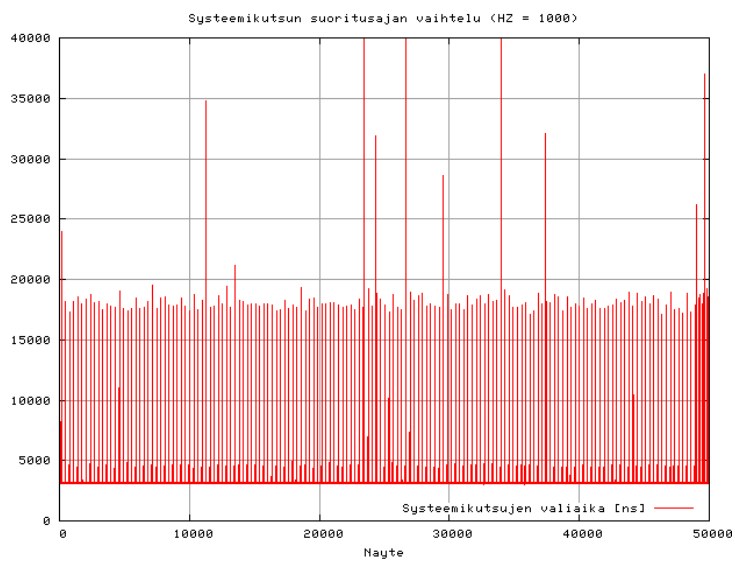
Kuva F2



Kuva F3



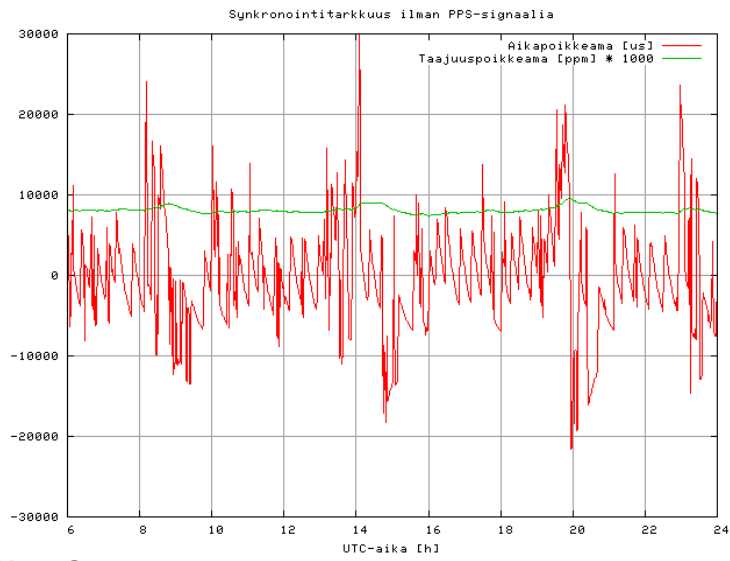
Kuva F4



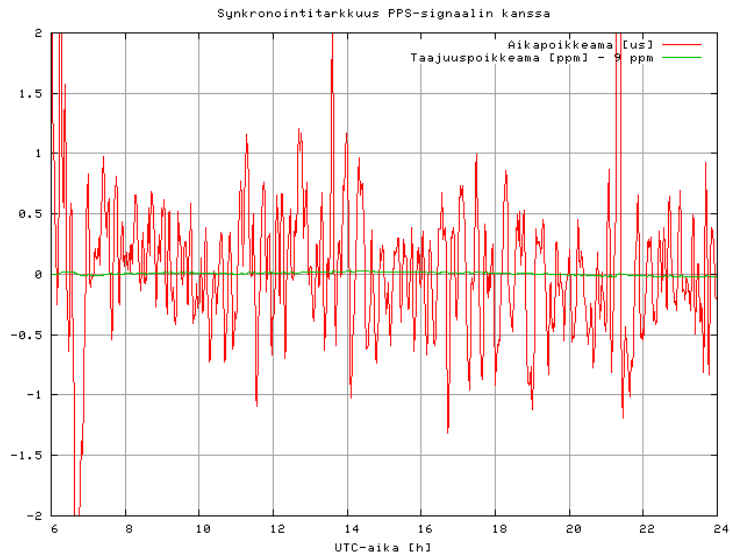


Liite G: Systemikellon aikapoikkeamaan vaikuttavia tekijöitä (Intel Celeron 300 MHz)

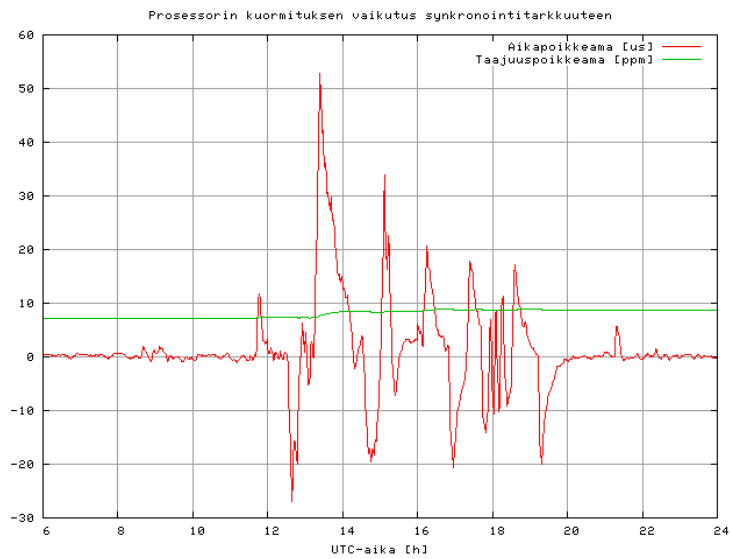
Kuva G1



Kuva G2

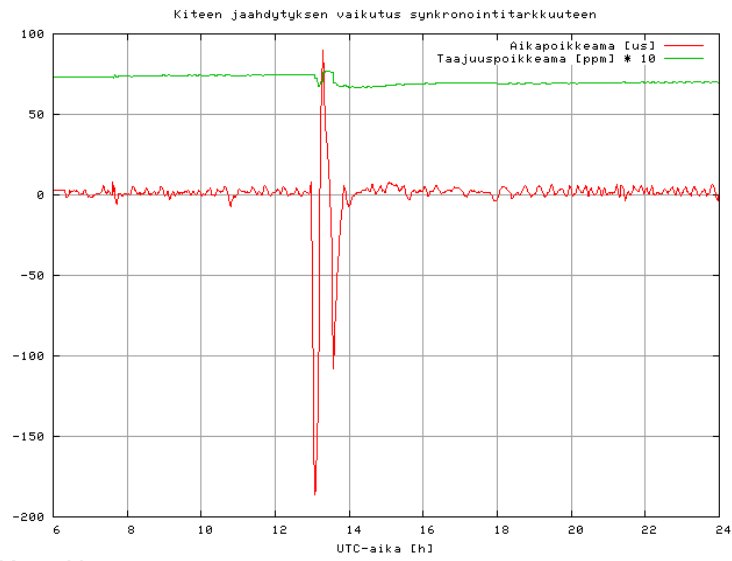


Kuva G3

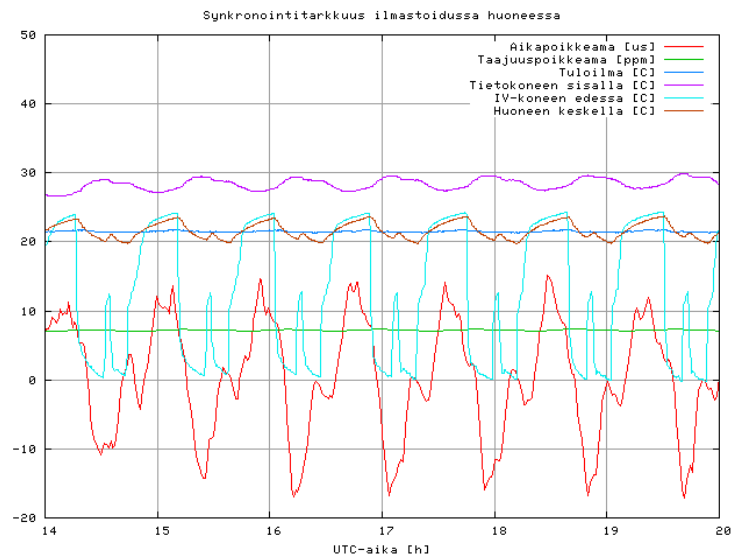


# Liite H: Lämpötilan vaikutus synkronointitarkkuuteen (Intel Celeron 300 MHz)

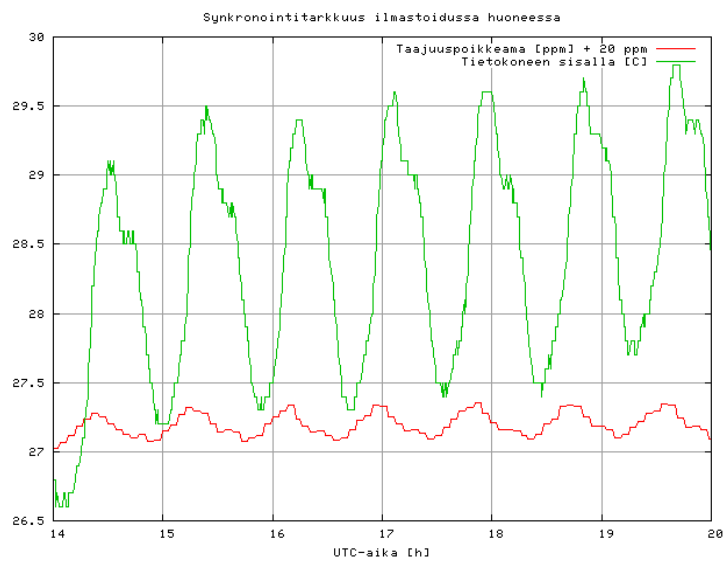
## Kuva H1



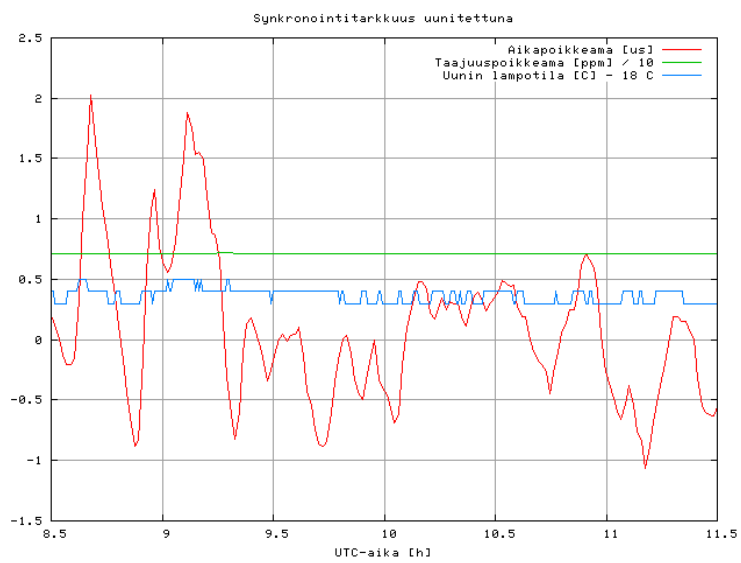
## Kuva H2



## Kuva H3

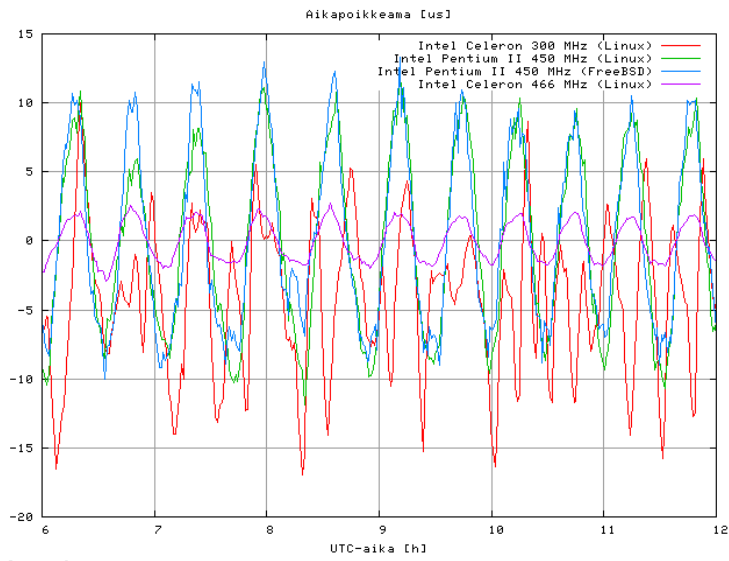


Kuva H4

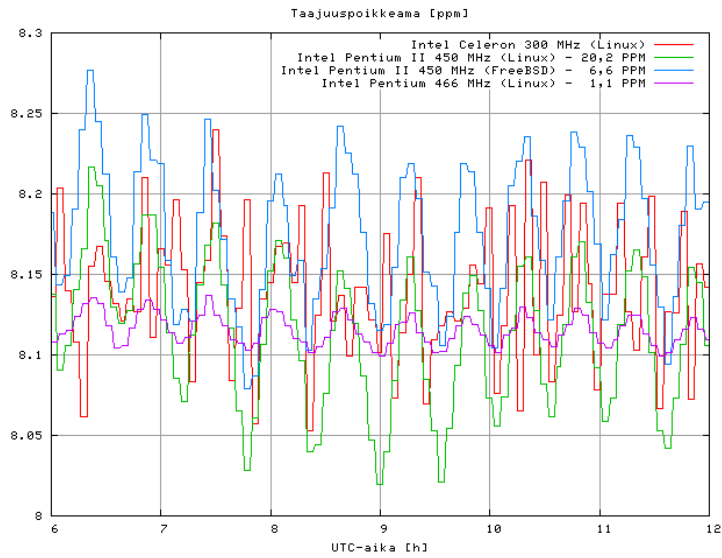


# Liite I: Kokoonpanon vaikutus synkronointitarkkuuteen

## Kuva I1

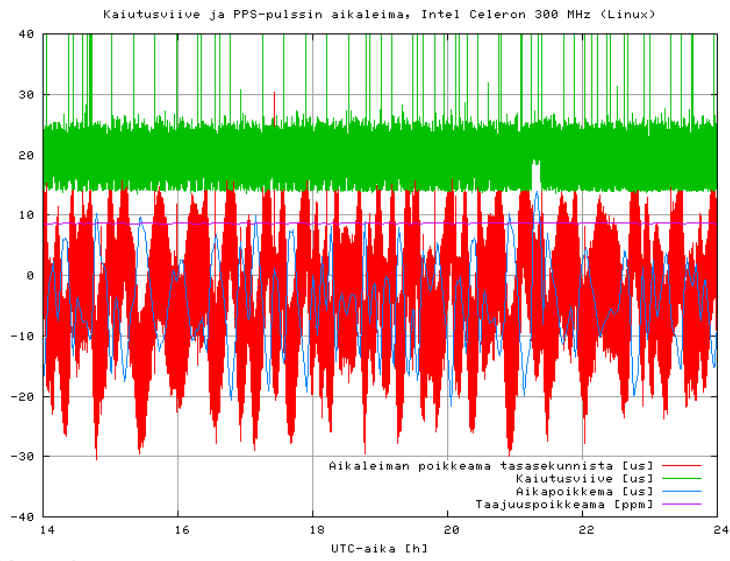


## Kuva I2

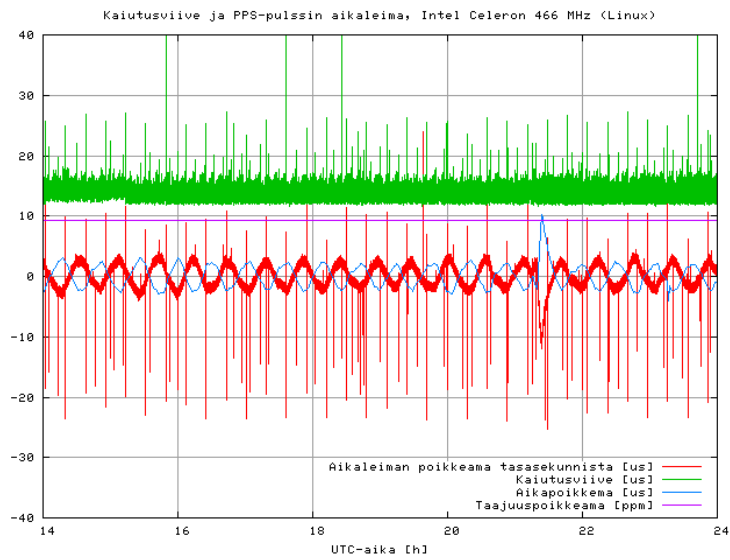


## Liite J: Kaiutusviive ja PPS-pulssin aikaleima

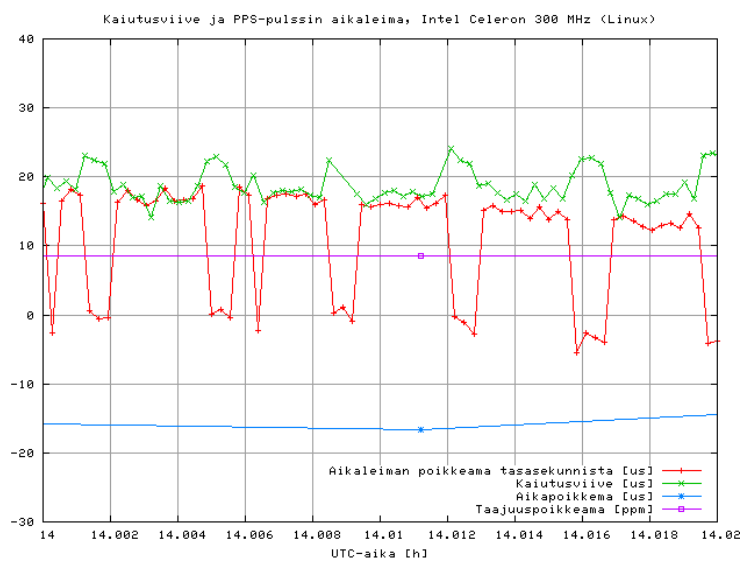
Kuva J1



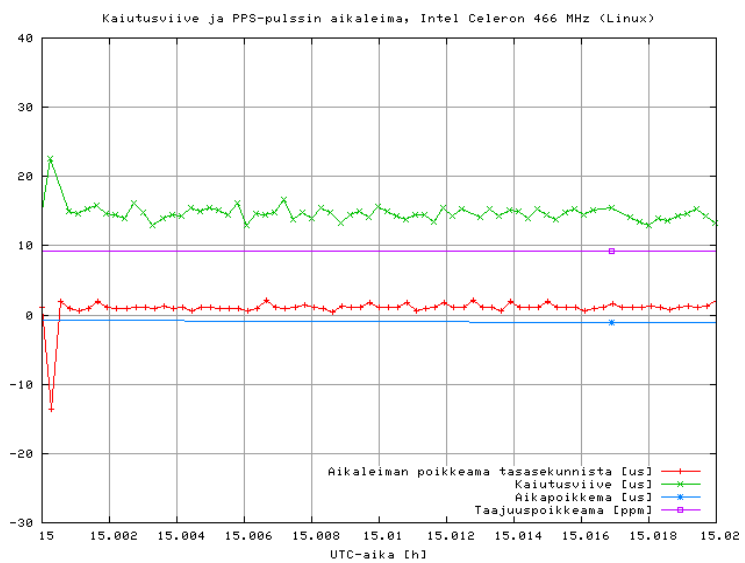
Kuva J2



Kuva J3

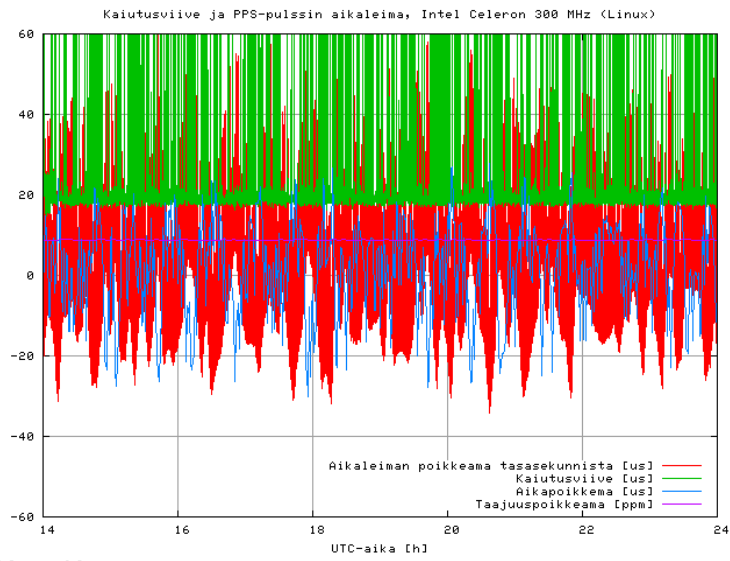


Kuva J4

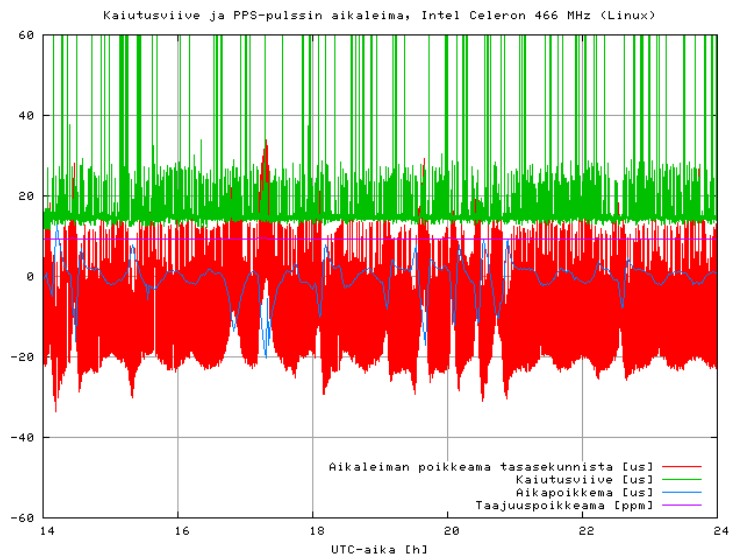


Liite K: Kaiutusviive ja PPS-signaalin aikaleima kuormitetussa verkossa

Kuva K1

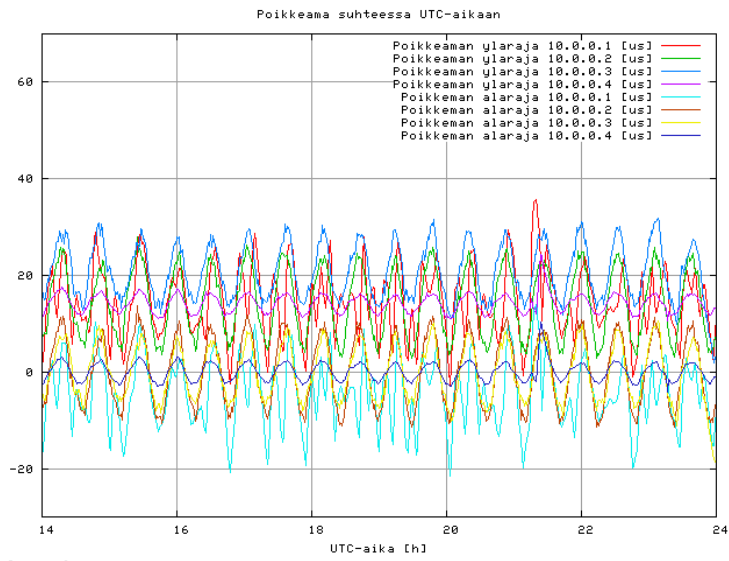


Kuva K2

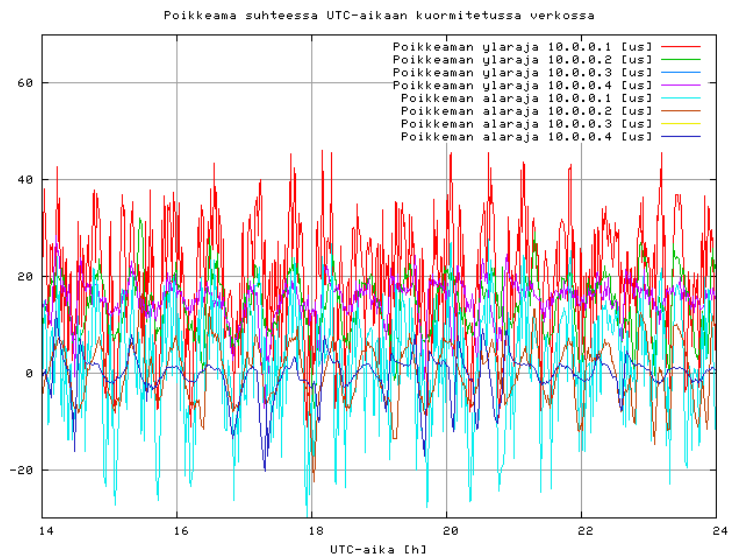


## Liite L: Poikkeama suhteessa UTC-aikaan

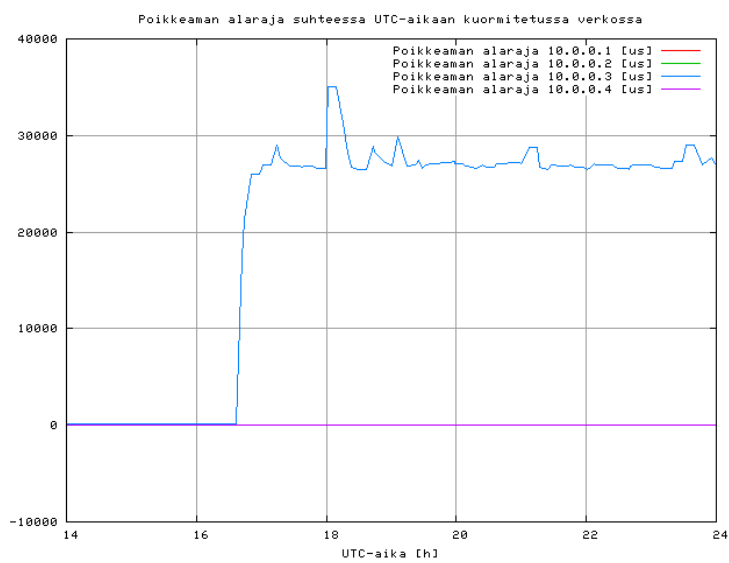
Kuva L1



Kuva L2



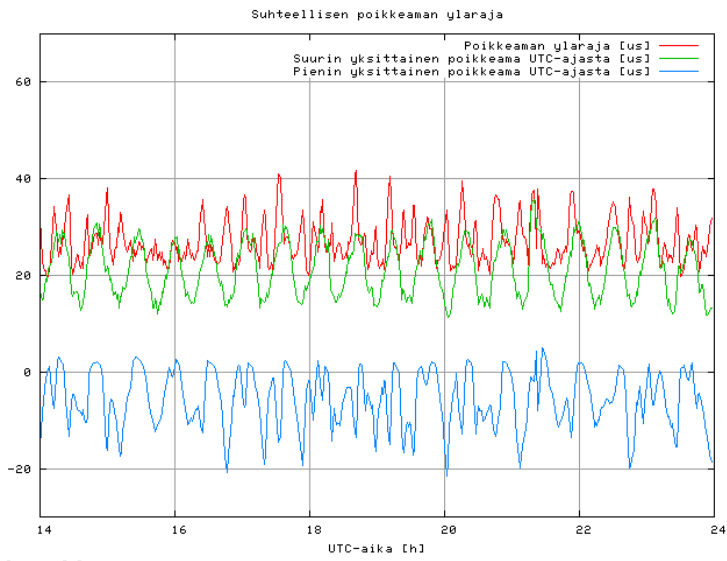
Kuva L3



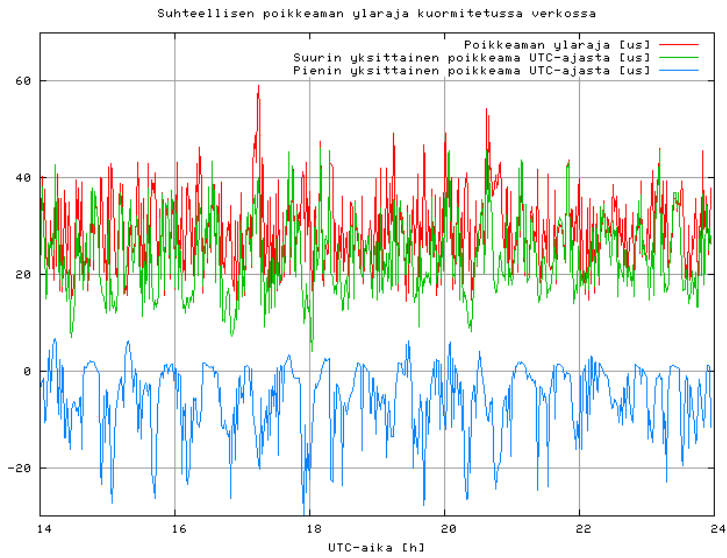


## Liite M: Suhteellisen poikkeaman yläraja

Kuva M1

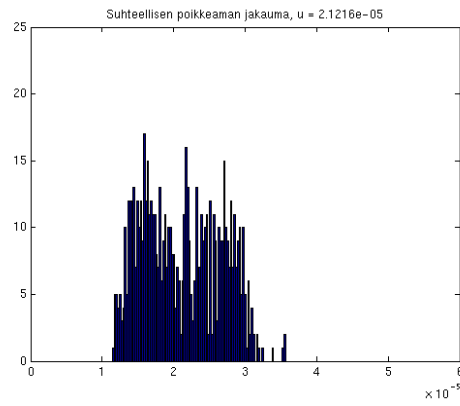


Kuva M2

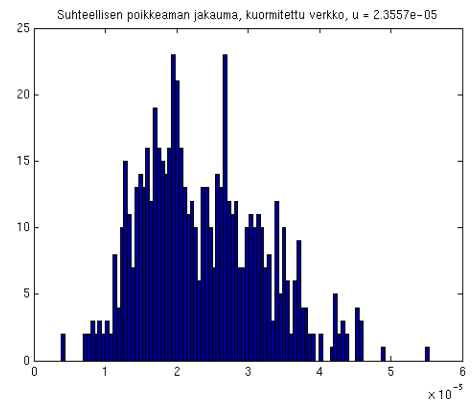


Liite N: Suhteellisen poikkeaman jakauma ja kertymäfunktio (Intel Celeron 466 MHz)

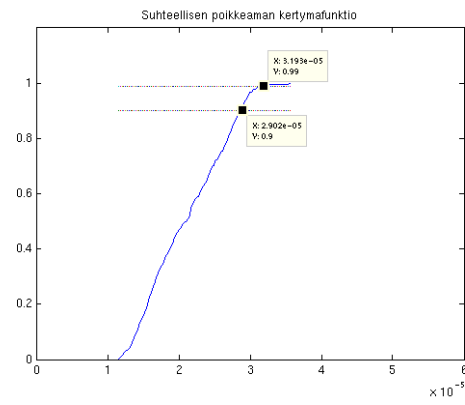
Kuva N1



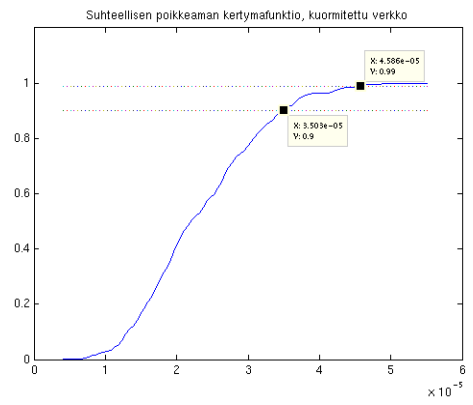
Kuva N2



Kuva N3

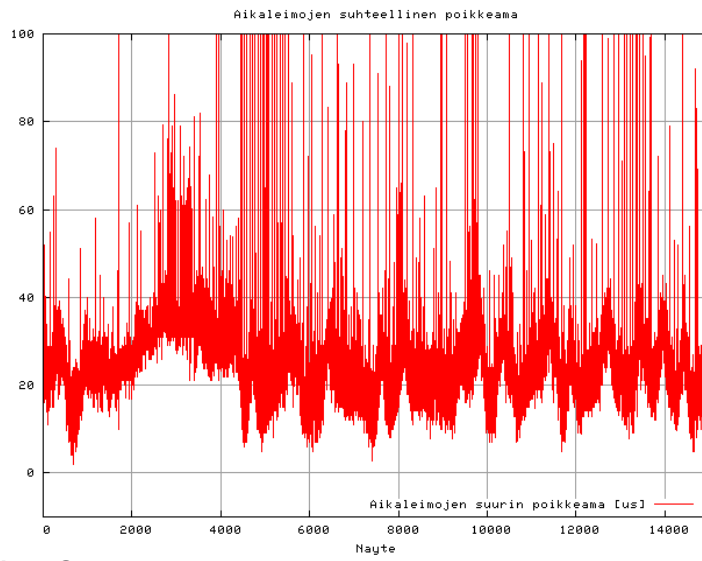


Kuva N4



Liite O: Suhteellisen poikkeaman estimointi PING-kyselyn avulla

Kuva O1



Kuva O2

