

**Riku Kuismanen**

# **IP verkkojen suorituskyvyn mittaaminen**

Tietotekniikan  
kandidaatintutkielma  
7. elokuuta 2006

**Jyväskylän yliopisto**

**Tietotekniikan laitos**

**Jyväskylä**

**Tekijä:** Riku Kuismanen

**Yhteystiedot:** rtkuisma@cc.jyu.fi

**Työn nimi:** IP verkkojen suorituskyvyn mittaaminen

**Title in English:** Measuring performance of IP networks

**Työ:** Tietotekniikan kandidaatintutkielma

**Sivumäärä:** 33

**Tiivistelmä:** IPPM<sup>1</sup> on IETF<sup>2</sup>:n työryhmä, joka on luonut joukon dokumentteja koskien IP<sup>3</sup>-verkkojen suorituskyvyn määrittämistä. Tässä tutkielmassa keskitytään näiden dokumenttien sisältöön ja siihen, miten IP-verkon suorituskykyä voidaan mitata, ja mitä mittausparametreja mittaamiseen tarvitaan.

**English abstract:** IPPM is a workgroup of IETF, which has created a group of documents concerning measuring IP-networks performance. In this thesis the focus is in the content of these documents and to create a general view how the performance of IP-network can be measured and what kind of metrics are needed for measurements.

**Avainsanat:** suorituskykyparametri, IP-verkko

**Keywords:** performance metric, IP-network

---

<sup>1</sup>IP Performance Metrics

<sup>2</sup>The Internet Engineering Task Force

<sup>3</sup>Internet Protocol

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
1.1	Täsmälähetys . . . . .	1
1.2	Ryhmälähetys . . . . .	2
<b>2</b>	<b>Suorituskykyparametrit</b>	<b>3</b>
2.1	Saatavuus . . . . .	3
2.1.1	Hetkellinen yhdensuuntainen saatavuus . . . . .	3
2.1.2	Hetkellinen kaksisuuntainen saatavuus . . . . .	4
2.1.3	Yhdensuuntainen saatavuus . . . . .	4
2.1.4	Kaksisuuntainen saatavuus . . . . .	5
2.1.5	Kaksisuuntainen väliaikainen saatavuus . . . . .	5
2.1.6	Mittaaminen . . . . .	6
2.2	Yhdensuuntainen viive . . . . .	7
2.2.1	Yleistä ajasta . . . . .	7
2.2.2	Yhdensuuntainen viive . . . . .	8
2.2.3	Mittaustavoista . . . . .	8
2.2.4	Virheanalyysistä . . . . .	9
2.2.5	Mittaamisesta . . . . .	10
2.2.6	Spatiaalinen viiveen mittaaminen . . . . .	11
2.3	Yhdensuuntainen hävikki . . . . .	12
2.3.1	Parametri . . . . .	13
2.3.2	Mittaustavoista . . . . .	13
2.3.3	Virheanalyysistä . . . . .	14
2.3.4	Spatiaalinen yksisuuntaisen hävikin mittaaminen . . . . .	14
2.4	Edestakainen viive . . . . .	15
2.4.1	Parametri . . . . .	16
2.4.2	Mittaustavoista . . . . .	16
2.4.3	Virheanalyysistä . . . . .	17
2.5	Viiveen vaihtelu . . . . .	17
2.5.1	Parametri . . . . .	18
2.5.2	Mittaustavoista . . . . .	19
2.5.3	Virheanalyysistä . . . . .	19

2.6	Hävikkimallit . . . . .	20
2.6.1	Parametrit . . . . .	20
2.6.2	Mittaustavoista . . . . .	21
2.6.3	Tilastoja . . . . .	22
2.7	Kaistan kapasiteetti . . . . .	22
2.7.1	Määritykset . . . . .	22
2.8	Hyötykuormakapasiteetti . . . . .	24
2.8.1	Muut mitattavat parametrit . . . . .	24
2.9	Pakettien uudelleenjärjestely . . . . .	25
2.9.1	Parametri . . . . .	26
2.9.2	Muut parametrit . . . . .	26
<b>3</b>	<b>Yhteenveto</b>	<b>28</b>
	<b>Viitteet</b>	<b>29</b>

# 1 Johdanto

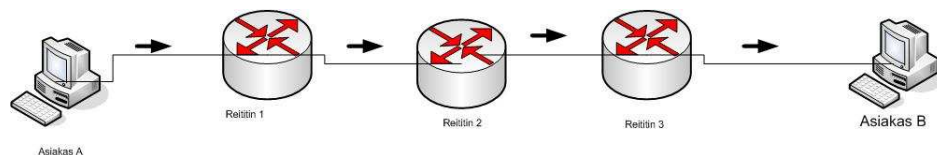
Tutkielman tarkoitus on tarkastella IP-verkon suorituskyvyn mittaamista ja suorituskykyparametreja (engl. *performance metrics*). Tutkielmassa tutustutaan IPPM:n määrittämiin suorituskykyparametreihin ja niiden mittaamiseen täsmälähetyksessä (engl. *unicast*) ja tarkastellaan, miten mittaaminen eroaa mitattaessa verkon suorituskykyä ryhmälähetyksessä (engl. *multicast*).

Verkon suorituskyvyn mittaaminen on tärkeää palvelunlaadun (engl. Quality of Service) eli QoS:n kannalta. Yhä useammat sovellukset ovat nykyään riippuvaisia verkon suorituskyvystä toimiakseen. Esimerkiksi VoIP<sup>1</sup>-puhelut ja IPTV<sup>2</sup>-lähetykset vaativat verkolta tiettyä suorituskykyä, jotta ne voivat toimia ilman että käyttäjä havaitsee palveluissa häiriöitä. Suorituskyvyn mittaaminen on siis tärkeää näiden palveluiden yleistyessä tulevaisuudessa.

Tutkielmassa tarkastellaan myös mittaamiseen liittyviä virhelähteitä ja niiden huomioimista mittaamisessa. Virheanalyysi on tärkeää, kun osa mitattavista parametreista saa melko pieniä arvoja. Tällöin pienikin virhe voi vääristää tulosta huomattavasti.

## 1.1 Täsmälähetyks

Täsmälähetyksellä tarkoitetaan lähetyksmuotoa, jossa yksi lähde lähettää tietoa yhdelle kohteelle IP-verkon yli. Lähde lähettää IP-paketit kohteen IP-osoitteeseen, jolloin verkon aktiivilaitteet osaavat toimittaa paketit oikealle kohteelle.



Kuva 1.1: Kuva täsmälähetyksestä

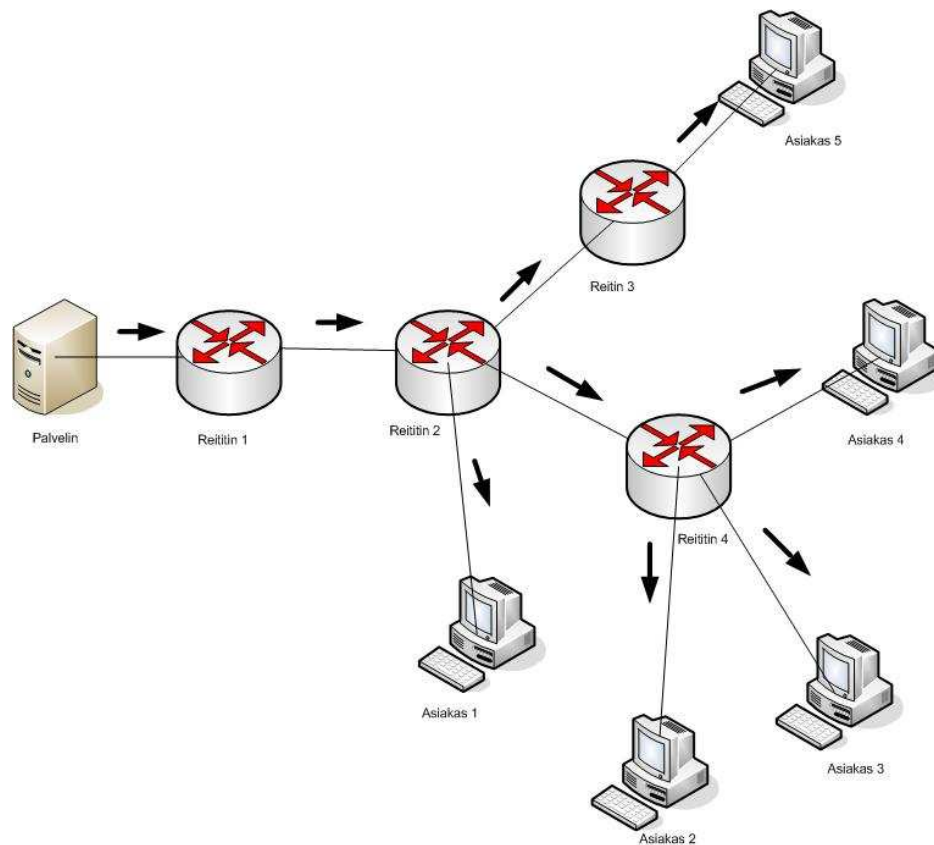
---

<sup>1</sup>Voice over IP

<sup>2</sup>Internet Protocol Television

## 1.2 Ryhmälähetys

Ryhmälähetyksellä tarkoitetaan lähetyksmuotoa, jossa paketti lähetetään usealle kohteelle yhtäaikaaisesti. Tällaista lähetyksmuotoa varten täytyy tiedon vastaanottajien liittyä ryhmälähetystä varten luotuun ryhmään. Paketti lähetetään tälle ryhmälle annettuun ryhmälähetysosoitteeseen. Kukaan paketti lähetetään vain kerran ja reititimet monistavat paketin ryhmän kaikille jäsenille. Paketin monistaminen tehdään myöhäisimmässä mahdollisessa haarautumiskohdassa, jolloin ei muodosteta turhaa liikennettä verkkoon eikä ruuhkauteta verkkoa turhaan. Ryhmälähetys on huomattavasti tehokkaampi ja järkevämpi lähetyksstapa kuin monta täsmälähetystä esimerkiksi IPTV-lähetyksissä, joissa samaa videokuvaa lähetetään usealle vastaanottajalle. Monta täsmälähetystä muodostaisivat verkkoon huomattavasti enemmän liikennettä.



Kuva 1.2: Kuva ryhmälähetyksestä

## 2 Suorituskykyparametrit

Tässä luvussa käsitellään IPPM:n määrittelemiä suorituskykyparametreja ja niiden mittaamista.

### 2.1 Saatavuus

Saatavuus tarkoittaa sitä, että kaksi asiakaslaitetta voivat tavoittaa toisensa verkon yli ja näin ollen kommunikoida keskenään. Saatavuus onkin yksi internetin peruspiilareista. IPPM-työryhmä määrittelee viisi eri suorituskykyparametria saatavuudelle (engl. *connectivity*). J. Mahdavi ja V. Paxson eittelevät nämä parametrit dokumentissaan 'IPPM Metrics for Measuring Connectivity' [1]. Näiden parametrien kuvauksessa käytetään lähettäjistä lyhennettä Src ja vastaanottajasta Dst. V. Paxson, G. Almes, J. Mahdavi ja M. Mathis määrittelevät dokumentissaan 'Framework for IP Performace Metrics' [2] suorituskykyparametrien mittaukseen käytettävää pakettia type-P paketiksi, joka tarkoittaa että mittaustulos on riippuvainen käytetystä mitauspaketista. Käytettävää pakettia ei lyödä lukkoon vaan se jätetään määriteltäväksi suunniteltaessa mittaussovellusta tai suoritettaessa itse mittauksia.

#### 2.1.1 Hetkellinen yhdensuuntainen saatavuus

Hetkellinen yhdensuuntainen saatavuus (eng. *instantaneous one-way connectivity*) -parametrin attribuutit:

**Parametrin nimi:** *Type – P – Instantaneous – Unidirectional – Connectivity*

- Src = Lähettäjän IP-osoite
- Dst = Vastaanottajan IP-osoite
- T = Aika

Parametri saa arvokseen tosi tai epätosi.

Src:llä on hetkellinen yhdensuuntainen saatavuus Dst:hen ajanhetkellä T, eli `Type-P-Instantaneous-Unidirectional-Connectivity` saa arvon Tosi, jos paketti `type-P`, joka on lähetetty ajanhetkellä T saapuu Dst:lle.

Yhdensuuntainen saatavuus ei ole kovin yleisesti tarpeellinen parametri käytännön sovelluksia silmälläpitäen. Monet käytännön sovellukset vaativat kaksisuuntaista saatavuutta ja usein jopa saatavuutta yli jonkun aikaintervallin. Tällä parametrilla voidaan kuitenkin testata verkkolaitteiden tietoturvallisuutta; esimerkiksi suodattaako palomuri *Ping of Death* -hyökkäykset pois oikein.

### 2.1.2 Hetkellinen kaksisuuntainen saatavuus

Hetkellinen kaksisuuntainen saatavuus (engl. *instantaneous two-way connectivity*) -parametrin attribuutit:

**Parametrin nimi:** `Type-P-Instantaneous-Bidirectional-Connectivity`

- A1 = Ensimmäisen asiakkaan IP-osoite
- A2 = Toisen asiakkaan IP-osoite
- T = Aika

Parametri saa arvokseen tosi tai epätosi.

Osoitteiden A1 ja A2 välillä on hetkellinen kaksisuuntainen saatavuus ajanhetkellä T, eli `Type-P-Instantaneous-Bidirectional-Connectivity` saa arvon tosi, jos A1:llä on hetkellinen yhdensuuntainen saatavuus A2:een ja A2:lla on hetkellinen yhdensuuntainen saatavuus A1:een.

### 2.1.3 Yhdensuuntainen saatavuus

Yhdensuuntainen saatavuus (engl. *one-way connectivity*) -parametrin attribuutit:

**Parametrin nimi:** `Type-P-Interval-Unidirectional-Connectivity`

- Src = Lähettäjän IP-osoite
- Dst = Vastaanottajan IP-osoite
- T = Aika
- dT = Kesto



Parametri saa arvokseen tosi tai epätosi.

Src:llä on ajanjakson  $[T, T + dT]$  aikana yhdensuuntainen saatavuus Dst:hen, eli `Type-P-Interval-Unidirectional-Connectivity` saa arvon tosi, jos jollain arvolla  $T'$ , kun  $T'$  kuuluu väliin  $[T, T + dT]$ , Src:llä on hetkellinen yhdensuuntainen saatavuus Dst:hen.

#### 2.1.4 Kaksisuuntainen saatavuus

Kaksisuuntainen saatavuus (engl. *two-way connectivity*) -parametrin attribuutit:

**Parametrin nimi:** `Type-P-Interval-Bidirectional-Connectivity`

- $A1$  = Ensimmäisen asiakkaan IP-osoite
- $A2$  = Toisen asiakkaan IP-osoite
- $T$  = Aika
- $dT$  = Kesto

Parametri saa arvokseen tosi tai epätosi.

Osoitteiden  $A1$  ja  $A2$  on ajanjakson  $[T, T + dT]$  aikana kaksisuuntainen saatavuus, eli `Type-P-Interval-Bidirectional-Connectivity` saa arvon tosi, jos  $A1$ :llä on yhdensuuntainen saatavuus  $A2$ :een ja  $A2$ :lla on yhdensuuntainen saatavuus  $A1$ :een ajanjakson  $[T, T + dT]$  aikana.

#### 2.1.5 Kaksisuuntainen väliaikainen saatavuus

Kaksisuuntainen väliaikainen saatavuus (engl. *two-way temporal connectivity*) -parametrin attribuutit:

**Parametrin nimi:** `Type-P1-P2-Interval-Temporal-Connectivity`

- $Src$  = Lähettäjän IP-osoite
- $Dst$  = Vastaanottajan IP-osoite
- $T$  = Aika
- $dT$  = Kesto

Parametri saa arvokseen tosi tai epätosi.

Src:llä on kaksisuuntainen väliaikainen saatavuus Dst:hen ajanjakson  $[T, T + dT]$  aikana, eli `Type-P1-P2-Interval-Temporal-Connectivity` saa arvon tosi, jos on olemassa ajat  $T_1$  ja  $T_2$ , sekä intervallit  $dT_1$  ja  $dT_2$  siten että:

- $T_1, T_1 + dT_1, T_2, T_2 + dT_2$  kuuluvat kaikki välille  $[T, T + dT]$
- $T_1 + dT_1 \leq T_2$
- Src:llä on ajanhetkellä  $T_1$  hetkellinen saatavuus `Type-P1` Dst:hen
- Dst:llä on ajanhetkellä  $T_2$  hetkellinen saatavuus `Type-P2` Src:hen
- $dT_1$  on aika, joka paketilta kuluu matkalla Dst:hen, kun Src lähettää sen ajanhetkellä  $T_1$
- $dT_2$  on aika, joka paketilta kuluu matkalla Src:hen, kun Dst lähettää sen ajanhetkellä  $T_2$

Tämä parametri määrittelee yleisesti hyödyllisen saatavuuden. Dst vastaa Src:n lähettämään pakettiin. Paketit voivat olla eri tyyppiä kuten monissa käytännön sovelluksissakin.

### 2.1.6 Mittaaminen

IPPM-työryhmä määrittelee rungon mittaustavalle. Mittaustapaa ei voida määrittää yksikäsitteisesti, koska yksityiskohdat riippuvat käytetyistä paketeista. Työryhmä määrittelee syötteen ja algoritmin, jota käytetään.

Syötteen:

- Pakettityypit  $P_1$  ja  $P_2$ , osoitteet  $A_1$  ja  $A_2$ , sekä intervalli  $[T, T + dT]$ .
- Lähetettävien pakettien määrä  $N$ .
- Odotusaika  $W$ , joka määrää kuinka kauan odotetaan vastausta lähetetylle paketille. Vaaditaan:  $W \leq 255 \text{ s}$ ,  $dT > W$ .
- Suositellut arvot:  $dT = 60 \text{ s}$ ,  $W = 10 \text{ s}$  ja  $N = 20$

**Algoritmi:** Lasketaan  $N$  lähetyssaikaa väliltä  $[T, T + \Delta T - W]$ , jotka on satunnaisesti hajautettu tuolle välille. Jokaisena lähetyssaikana lähetetään hyvinmuodostettu tyyppin  $P_1$  paketti  $A_1$ :ltä  $A_2$ :lle. Tutkitaan saapuvaa liikennettä  $A_1$ :llä. Jos saadaan sallittu vastauspaketti ( $P_1$  tai  $P_2$ ), mittauksen tulokseksi asetetaan tosi, ja mittaus voidaan lopettaa. Jos sallittua vastausta ei saada ajanhetkeen  $T + \Delta T$  mennessä, mittauksen tulokseksi asetetaan epätosi.

Algoritmi ei ole täysin pitävä, sillä mittaus ei todista saatavuutta jokaisen ajanjakson  $[T, T + \Delta T]$  hetkenä. Sopivan  $N$ :n arvon valinta on hankalaa; suurempi  $N$ :n arvo antaa tarkemman tuloksen, mutta generoi enemmän liikennettä verkkoon. Arvolle ei ole määritelty eksaktia arvoa.

## 2.2 Yhdensuuntainen viive

Tässä luvussa käsitellään yhdensuuntaista viivettä ja arvioidaan parametrin mittaukseen liittyviä virhetekijöitä.

### 2.2.1 Yleistä ajasta

Ennen kuin voidaan tarkastella itse viivettä, on tiedettävä muutamia asioita koskien aikaa. IPPM-työryhmä määrittelee neljä kelloihin ja aikaan liittyvää epätarkkuustekijää:

**Tahdistus** (engl. *Synchronization*) tarkoittaa, kahden eri laitteen kellojen olemista samassa ajassa. Esimerkiksi laitteen A kello voi olla kaksi millisekuntia laitteen B kelloa edellä.

**Tarkkuus** (engl. *Accuracy*) tarkoittaa, laitteen kellon samassa ajassa olemista koor-dinoituun yleisaikaan (engl. *coordinated universal time = UTC*) verrattuna.

**Resoluutio** (engl. *Resolution*) tarkoittaa kellon lyöntitiheyttä. Esimerkiksi kello voi lyödä kerran kymmenessä millisekunnissa, jolloin kellon resoluutio on kymmenen millisekuntia.

**Vääristymä** (engl. *Skew*) tarkoittaa tarkkuuden tai tahdistuksen muutosta ajan suhteen. Esimerkiksi kello saattaa saavuttaa UTC:tä kaksi millisekuntia joka tunti. Kello A on aluksi 22 millisekuntia UTC:tä jäljessä, ja tunnin päästä enään 20 millisekuntia. Tällöin sanotaan että kellolla A on kahden millisekunnin vääristymä verrattuna UTC:hen. Vääristymä voi olla myös suhteessa johonkin toiseen kellon.

### 2.2.2 Yhdensuuntainen viive

Viive (engl. *delay*) on verkon ominaisuus, joka on erittäin merkittävässä roolissa monien sovellusten kannalta. Monet sovellukset vaativat toimiakseen, että viiveen on pysyttävä jonkun tietyn rajan alapuolella. Muutoin sovellusta ei voida käyttää tai se toimii huonosti. Tämän takia tietyn verkon läpi kulkevan polun (engl. *path*), tai verkon virekkäiseltä laitteelta toiselle kulkevan linkin (engl. *link*) paketilta kuluvan ajan, eli viiveen, selvittäminen on tärkeää.

G. Almes, S. Kalidindi ja M. Zekauskas käsittelevät yhdensuuntaista viivettä ja sen mittaamista verkossa, ja määrittelevät parametrin Yhdensuuntainen viive (engl. *One-way Delay Metric*) dokumentissaan 'A One-way Delay metric for IPPM' [3].

Yhdensuuntaisen viiveen mittaaminen edestakaisen viiveen sijasta on perusteltua, koska nykyisissä verkoissa reitti lähettäjältä vastaanottajalle voi olla eri kuin reitti takaisin. Edestakainen viive mittaakin itseasiassa kahden eri reitin yhteistä viivettä. Kun mitataan yhdensuuntaisia viiveitä, korostuvat eri reittien suorituskykyerot paremmin. Vaikka reitit olisivat molempiin suuntiin symmetriset, voi viive voidella paljonkin reittien välillä, koska verkon laitteiden jonot voivat olla epäsymmetriset.

Yhdensuuntainen viive -parametrin attribuutit:

**Parametrin nimi:** `Type-P-One-way-Delay`

- `Src` = Lähettäjän IP-osoite
- `Dst` = Vastaanottajan IP-osoite
- `T` = Aika

Parametri saa arvokseen reaalisen määrän sekunteja tai määrittelemätön (ei voitu mitata).

Parametrin arvolla `dT Type-P-One-way-Delay`:lle tarkoitetaan sitä intervalia, kun `Src` on lähettänyt ensimmäisen bitin `Type-P` paketista ajanhetkellä `T` `Dst`:lle ja `Dst` on vastaanottanut paketin viimeisen bitin ajanhetkellä `T + dT`. Jos arvoksi saadaan määrittelemätön, `Src` lähetti paketin, mutta `Dst` ei saanut sitä.

### 2.2.3 Mittaustavoista

IPPM-työryhmä määrittelee algoritmin yhdensuuntaisen viiveen mittaamiselle. Yksityiskohtaisempi määrittely jätetään mittaussovelluksen kehittäjälle, koska yksi-

tyiskohdat algoritmissa riippuvat käytettävästä mittauspaketista.

**Algoritmi:** Huolehditaan että Src ja Dst on tahdistettu keskenään erittäin tarkasti ja, että kellot ovat mahdollisimman oikeassa ajassa. Src-laitteessa muodostetaan `type-P`-testipaketti ja asetetaan Src- ja Dst -kentiin lähettäjän ja vastaanottajan IP-osoitteet. Pakettiin laitetaan täytettä (engl. *padding*), jotta muodostettu paketti on tarpeeksi iso mittauksen oikeellisuuden varmistamiseksi. Liian pinet pakettikoot aiheuttavat pienemmän viiveen, johtuen polulla käytetyistä pakkausmenetelmistä. Dst-laitteessa valmistaudutaan vastaanottamaan mittauspakettia. Src:ssä asetetaan aikaleima pakettiin ja lähetetään se Dst:lle. Jos paketti saapuu järkevän ajan kuluessa Dst:hen, otetaan talteen mahdollisimman nopeasti paketin saapumisaika. Aikaleima ja saapumisaika vähennetään toisistaan ja näin saadaan arvio yhdensuuntaiselle viiveelle. Jos paketti ei saapunut Dst:hen järkevän ajan kuluessa, asetetaan arvoksi määrittelemätön. Järkevän ajan määrittelemisen jätetään mittaussovelluksen kehittäjälle.

#### 2.2.4 Virheanalyysistä

Viivettä mitattaessa täytyy ottaa huomioon monia virhelähteitä. Koska viiveen arvo on ajanjakso, täytyy aikaan liittyviin epätarkkuuksiin kiinnittää erityistä huomiota. Virhe voi liittyä kahteen eri epätarkkuustekijään. Src:n ja Dst:n kelloihin tai niiden väliseen epätarkkuuteen, tai mittausajan (engl. *host time*) ja siirtoajan (engl. *wire time*) väliseen erotukseen.

**Kelloihin liittyvät virhelähteet:** Virheanalyysissä käytetään Src:n kellosta nimitystä lähettäjän kello ja Dst:n kellosta vastaanottajan kello. Paketin lähetysaikaa merkitään  $T_{source}$ :lla ja vastaanottoaikaa  $T_{dest}$ :llä.

Virhe lähettäjän kellon ja vastaanottajan kellon tahdistuksessa aiheuttaa mittaus-tulokseen virheen. Sanotaan, että lähettäjän kellon ja vastaanottajan kellon välillä on tahdistusvirhe (engl. *synchronization error*)  $T_{synch}$ , jos lähettäjän kello on arvon  $T_{synch}$  verran vastaanottajan kelloa edellä. Jos tämä tiedettäisiin tarkasti, voitaisiin virhe korjata lisäämällä  $T_{synch}$  arvoon  $T_{dest} - T_{source}$ .

Kellojen tarkkuudella on merkitystä ainoastaan selvittäessä, milloin viive on mitattu. Tarkkuudella ei ole vaikutusta viiveen arvoon.

Kellojen resoluutio tuo tulokseen epävarmuutta. Jos laitteen kellolla on 10 millisekunnin resoluutio, tuo se 10 millisekunnin epävarmuuden tulokseen. Tässä tutkielmassa kellojen resoluutioita merkitään seuraavasti: Lähettäjän kellon resoluutiota merkitään  $R_{source}$ :lla ja vastaanottajan  $R_{dest}$ :llä.

Vääristymä vaikuttaa kellojen tahdistukseen ja arvoon  $T_{synch}$ . Vääristymän

vaikutusta voidaan approksimoida tahdistuksen lineaarisena funktiona ajan suhteen, lisättynä joillakin korkeamman asteen termeillä. Merkitään tätä  $E_{\text{synch}}$ :llä siten, että  $E_{\text{synch}}$  on tahdistuksen epävarmuuden yläraja ajan funktiona, eli

$$|T_{\text{synch}}(t)| \leq E_{\text{synch}}(t).$$

Kun huomioidaan nämä kellojen virhelähteet, saadaan naivi arvio tuloksen  $T_{\text{dest}} - T_{\text{source}}$  virheelle  $T_{\text{synch}}(t) \pm (R_{\text{source}} + R_{\text{dest}})$ . Käyttämällä edellä mainittua notaatiota saadaan arvio kelloihin liittyvästä kokonaisvirheestä  $E_{\text{synch}}(t) + R_{\text{source}} + R_{\text{dest}}$ .

**Mittausajan ja siirtoajan erotuksen tuomat virheet:** Siirtoajalla tarkoitetaan paketin todellista siirtoaikaa, eli aikaa siitä hetkestä kun ensimmäinen bitti paketista lähetetään Src:stä, siihen hetkeen kun viimeinen bitti paketista on vastaanotettu Dst:ssä. Mittausaika eroaa tästä jonkinverran, sillä mittausaika on se aika kun Src on laittanut aikaleiman pakettiin ja Dst on tämän aikaleiman saanut paketista selvitettyä. Tämä erotus tunnetaan, joten päästään lähemmäksi haluttua siirtoaika-arvoa. Tämä erotus on kuitenkin epävarma, joten se on huomioitava virheanalyysissä. Merkitään  $H_{\text{source}}$ :lla mittausajan ja todellisen siirtoajan erotuksen epävarmuuden ylärajaa lähetyspäässä, ja  $H_{\text{dest}}$ :llä vastaanottopäässä.

Yleisesti voidaan sanoa, että mittautulos on seuraavanlainen: mitattu arvo = todellinen arvo + systemaattinen virhe + satunnainen virhe. Jos systemaattinen virhe voidaan selvittää, voidaan se poistaa ilmoitetusta tuloksesta: Ilmoitettu tulos = mitattu arvo - systemaattinen virhe. Tästä saadaan: Ilmoitettu tulos = todellinen arvo + satunnainen virhe.

Edellisistä kappaleista saadaan nyt kokonaisvirheelle arvio:  $E_{\text{synch}} + R_{\text{source}} + H_{\text{source}} + H_{\text{dest}}$ . Kelloihin liittyvät virheet ovat pieniä nykyään, koska kellot saadaan tahdistettua tarkasti GPS<sup>1</sup>:n avulla. Kellojen tahdistus voidaan hoitaa myös NTP<sup>2</sup>:n avulla. Mittausaikaan liittyvät virheet saadaan rajattua yhdistämällä kaksi laitetta nopealla LAN<sup>3</sup>-yhteydellä. Toistuvat mittaukset mittaavat tällöin samaa viiveen arvoa. Käyttämällä mahdollisimman pieniä testipaketteja viiveen voidaan katsoa olevan nolla, joten toistuvien mittausten keskiarvo on systemaattisen virheen arvo ja hajonta satunnainen virhe.

## 2.2.5 Mittaamisesta

Mittaukset suoritetaan yleensä usemman mittauksen sarjoissa. Mittaukset hajautetaan jollekin tietylle aikavälille mahdollisimman satunnaisesti. Hajautukseen käytetään

---

<sup>1</sup>Global Positioning System

<sup>2</sup>Network Time protocol

<sup>3</sup>Local Area Network

yleensä Poisson-prosessia, koska se jäljittelee verkossa todellisuudessakin esiintyvää tilannetta todella hyvin.

### 2.2.6 Spatiaalinen viiveen mittaaminen

E. Stephan, L. Liang ja A. Morton määrittelevät dokumentissaan 'IP Performance Metrics (IPPM) for spatial and multicast' [4] spatiaaliset suorituskykyparametrit viiveelle, hävikille ja viiveen vaihtelulle. Suorituskykyparametrin sanotaan olevan spatiaalinen, jos sen mittaamiseen osallistuu muitakin osapuolia kuin lähettäjä ja vastaanottaja.

Spatiaalisessa mittauksessa mittaustuloksia kerätään koko polun matkalta lähettäjältä vastaanottajalle. Mittaukseen osallistuvat siis muutkin verkon aktiivilaitteet, kuin lähettävän ja vastaanottavan asiakkaan laitteet.

Spatiaalisen yhdensuuntaisen viiveen mittaaminen perustuu yhdensuuntaisen viiveen mittaamiseen, mutta mittaus suoritetaan kaikilla Src:n ja Dst:n välisen polun laitteilla. Parametrin attribuutit ovat seuraavanlaiset:

**Parametrin nimi:** `Type-P-Spatial-One-way-Delay-Stream`

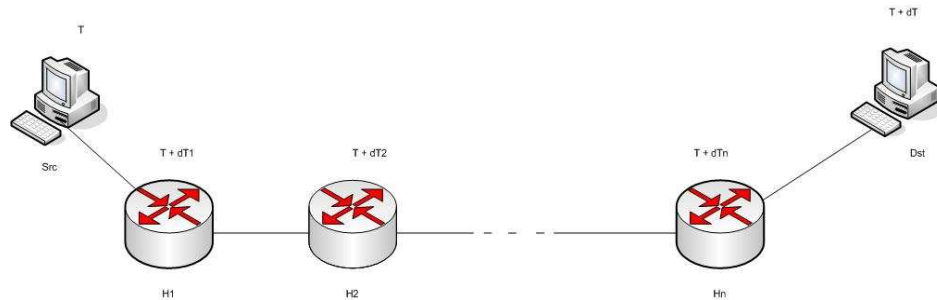
- `Src` = Lähettäjän IP-osoite
- `Dst` = Vastaanottajan IP-osoite
- `i` = Polun laitteiden määrä
- `Hi` = Polun laite
- `T` = Aika
- `dT` = Yhdensuuntainen viive
- `dT1, . . . , dTn` = Yhdensuuntaisten viiveiden lista
- `P` = Pakettityyppi
- `<Src, H1, . . . , Hn, Dst>` = Kooste polusta

Parametri saa arvokseen sarjan aikoja.

Src lähettää `Type-P` -paketin Dst:lle ajanhetkellä `T` polkua `<H1, H2, . . . , Hn>` pitkin. Saadan sarja aikoja `<T + dT1, T + dT2, . . . , T + dTn, T + dT>`, joista

$dT$  on yhdensuuntainen viive Src:stä Dst:hen ja jokaiselle polun laitteelle  $H_i$  vastaavat arvot  $T + dT_i$ , jotka ovat ajanhetket jolloin paketti ohittaa laitteen  $H_i$ .

Parametrin `Type-P-Spatial-One-way-Delay-Stream` arvo polulle  $\langle Src, H_1, \dots, H_n, Dst \rangle$  on siis  $T, dT_1, \dots, dT_n, dT$ .



Kuva 2.1: Kuva spatiaalisen viiveen mittaamisesta

Tällä mittaustavalla voidaan selvittää myös yksittäisten linkkien ja alipolkujen (engl. *subpath*) viiveitä. Kun halutaan mitata jonkun polulla olevan välin viivettä, vähennetään viiveen arvot toisistaan. Esimerkiksi viive välin  $[H_a, H_b]$  välillä saadaan kun lasketaan  $dT_b - dT_a$ . Tässä  $H_b$ :n on oltava polulla myöhemmin, kuin  $H_a$ . Dokumentissa [4] ei oteta kantaa tapaan, jolla verkon laitteet tämän hoitavat.

## 2.3 Yhdensuuntainen hävikki

Viiveen lisäksi myös pakettien hävikki (engl. *packet loss*) vaikuttaa suuresti etenkin reaaliaikaisten sovellusten toimintaan. Joidenkin sovellusten toiminta on vaikeaa tai jopa mahdotonta, jos pakettien hävikki on liian suurta. Suuri hävikki vaikuttaa myös kuljetuskerroksen protokollien toimintaan.

G. Almes, S. Kalidindi ja M. Zekauskas määrittelevät dokumentissaan 'A One-way Packet Loss Metric for IPPM' [5] parametrin yhdensuuntainen hävikki (engl. *one-way packet loss*) ja sen, kuinka parametri voidaan mitata.

Yhdensuuntainen hävikki on monesti mielenkiintoisempi parametri kuin edestakaisin mitattu hävikki. Tähän pätevät pitkälti samat perusteet kuin aikaisemmin yhdensuuntaisen viiveen kanssa. Monesti toiseen suuntaan tapahtuva hävikki on paljon mielenkiintoisempaa sovelluksen toiminnan kannalta kuin toiseen. Esimerkiksi tiedonsiirrossa käyttäen TCP<sup>4</sup>:tä on mielenkiintoisempaa tietää hävikin määrä tiedonsiirtosuuntaan, kuin suuntaan johon kuittausviestit (engl. *acknowledgement*) kulkevat.

<sup>4</sup>Transmission Control Protocol



### 2.3.1 Parametri

Yhdensuuntainen hävikki -parametrin attribuutit:

**Parametrin nimi:** `Type-P-One-way-Packet-Loss`

- `Src` = Lähettäjän IP-osoite
- `Dst` = Vastaanottajan IP-osoite
- `T` = Aika

Parametri saa arvokseen nollan tai ykkösen.

`Type-P-One-way-Packet-Loss` saa arvon nolla, jos `Src`:n ajanhetkellä `T` lähetetty paketti `Type-P` saapuu `Dst`:lle. `Type-P-One-way-Packet-Loss` saa arvon yksi, jos `Src`:n lähettää ajanhetkellä `T` paketin `Type-P` `Dst`:lle, mutta `Dst` ei saa kyseistä pakettia.

Parametrin arvot nolla ja yksi eivät ole aina välttämättä yksikäsitteisiä. Jos paketti saapuu pitkän ajan jälkeen perille, voidaan paketti olla tulkittu jo hävinneeksi. Tällaisia tilanteita varten onkin mittausalgoritmiin sisällytettävä jonkinlainen yläraja saapumisajalle, jonka jälkeen saapuneet paketit ovat hävinneitä. Myös korruptoituneet paketit lasketaan hävinneiksi. Jos paketti on monistunut reitillä ja se saapuu useana ehjänä kopiona, lasketaan se saapuneeksi perille. Jos paketti on pilkottu, eikä sen uudelleenrakennus onnistu, lasketaan paketti hävinneeksi.

### 2.3.2 Mittaustavoista

Tässäkin kuten viiveessä algoritmin tarkempi määrittely riippuu käytetystä pakettityypistä, mutta algoritmin pääpiirteet ovat seuraavanlaiset:

Järjestetään että `Src`:n ja `Dst`:n kellot on tahdistettu. Tarkkuus riippuu käytetystä raja-arvosta jolla paketti määritellään hävinneeksi. `Src`:ssä määritellään lähettäjän ja vastaanottajan IP-osoitteet ja muodostetaan `Type-P`-paketti. `Dst`:ssä valmistaudutaan vastaanottamaan testipakettia. `Src`:ssä asetetaan pakettiin aikaleima ja lähetetään se `Dst`:lle. Jos paketti saapuu järkevän ajan kuluessa, on `one-way-packet-loss`:n arvo nolla. Jos paketti ei saavu perille järkevän ajan kuluessa, on `one-way-packet-loss`:n arvo yksi. Järkevä odotusaika on tässä algoritmin attribuutti.

### 2.3.3 Virheanalyysistä

Yhdensuuntaisen hävikin mittaamiseen liittyy kolme eri virhelähdettä. Src:n ja Dst:n kellojen tahdistus, raja-arvo paketin katoamiselle ja resurssien rajoitukset Dst:n verkko-rajapinnassa tai ohjelmistossa. Näistä kaksi ensimmäistä liittyvät toisiinsa. Jos kellojen tahdistus on pielessä, saattaa paketti ,joka saapuu järkevässä ajassa, tulla lasketuksi hävinneeksi. Tai vastaavasti paketti, joka saapuu liian myöhään, saattaa tulla hyväksytyä. Jos taas hyväksymisen raja-arvo asetetaan liian pieneksi, saattaa turhia paketteja tulla lasketuksi hävinneiksi. Tahdistukseen liittyvät virheet on käsitelty tarkemmin yhdensuuntaista viivettä käsiteltäessä.

Resurssien rajoitukset aiheuttavat virhettä mittaustuloksiin, jos paketti hylätään mitaavalla laitteella laitteen ollessa ruuhkainen. Verkko siis toimitti paketin perille asianmukaisesti, mutta laite hylkäsi paketin ja se katsottiin hävinneeksi. Tällaisen tapahtuman todennäköisyys on mitaavalla laitteella järjestettävä mahdollisimman pieneksi.

### 2.3.4 Spatiaalinen yksisuuntaisen hävikin mittaaminen

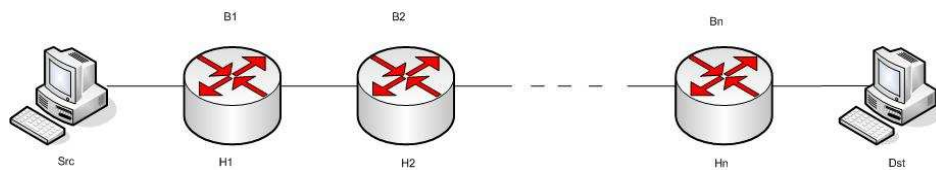
IPPM määrittelee myös yhdensuuntaiselle hävikille spatiaalisen mittaustavan. Spatiaalinen mittaus perustuu yhdensuuntaiseen hävikin mittaamiseen, mutta mittaukseen osallistuvat polun kaikki laitteet. Parametrin attribuutit:

**Parametrin nimi:** `Type-P-Spatial-One-way-Packet-Loss-Stream`

- `Src` = Lähettäjän IP-osoite
- `Dst` = Vastaanottajan IP-osoite
- `i` = Polun laitteiden määrä
- `Hi` = Polun laite
- `T` = Aika
- `dT1, . . . , dTn` = Yhdensuuntaisten viiveiden lista
- `P` = Pakettityyppi
- `<Src, H1, . . . , Hn, Dst>` = Kooste polusta
- `B1, B2, . . . , Bn` = lista boolean-arvoista (tosi/epätosi)

Parametri saa arvokseen sarjan boolean-arvoja.

Src lähettää paketin ajanhetkellä  $T$  Dst:lle polkua  $\langle H_1, \dots, H_n \rangle$  pitkin. Saadaan sarja aikoja  $T+dT_1, \dots, T+dT_n, T+dT$ , kun paketti kulkee laitteiden  $\langle H_1, \dots, H_n, Dst \rangle$  kautta. `Type-P-Spatial-One-way-Packet-Loss-Stream`:lle saadaan arvoksi sarja arvoja  $B_1, B_2, \dots, B_n$ , joissa jokaiselle laitteelle  $H_i$   $B_i$ :n arvo 0 tarkoittaa, että  $dT_i$ :n arvo on määrällinen ja  $B_i$ :n arvo 1 tarkoittaa, että  $dT_i$ :n arvo on määrittelemätön. Dokumentti [4] ei sinänsä ota kantaa tapaan, jolla verkon laitteet tämän hoitavat.



Kuva 2.2: Kuva spatiaalisen hävikin mittaamisesta

## 2.4 Edestakainen viive

Edestakainen viive (engl. *Round-trip delay*) tarkoittaa sitä aikaintervalla, kun paketti lähtee lähettäjältä vastaanottajalle ja saapuu sitten takaisin lähettäjälle. G. Almes, S. Kalidindi ja M. Zekauskas määrittelevät dokumentissaan 'A Round-trip Delay Metric for IPPM' [6] parametrin edestakaiselle viiveelle ja esittelevät tavan sen mittaamiselle.

Aiemmin todettiin, että yhdensuuntainen viive on monessa suhteessa järkevämpi parametri mitattaessa IP-verkon viivettä. Vaikkakin polku ei ole molempiin suuntiin aina sama, on edestakaisella viiveellä etunsa yhdensuuntaiseen viiveeseen nähden. Edestakaisen viiveen mittaaminen on monesti helpompi järjestää, kuin yhdensuuntaisen viiveen mittaaminen. Vastaanottopään laitteeseen ei välttämättä tarvitse asentaa mitään mittaussovellusta. Voidaan käyttää esim ICMP<sup>5</sup>:n Echoa. Tulkinnan helppous on myös edestakaisen viiveen etu. Joskus edestakainen viive on juuri se mistä ollaan kiinnostuneita. Tällöin kahden yhdensuuntaisen viiveen mittaaminen ja yhdistäminen ei anna oikeaa kuvaa, koska vastaanottopään pakettien käsittelyaika ei ole sama, kuin se olisi edestakaista viivettä mitattaessa.

<sup>5</sup>Internet Control Message Protocol

### 2.4.1 Parametri

Edestakainen viive -parametrin attribuutit:

**Parametrin nimi:** `Type-P-Round-trip-Delay`

- `Src` = Lähettäjän IP-osoite
- `Dst` = Vastaanottajan IP-osoite
- `T` = Aika

Parametri saa arvokseen reaalisen määrän sekunteja tai määrittelemätön (ei voitu mitata).

Edestakainen viive `Type-P-Round-trip-Delay` saa arvokseen aikaintervallin  $dT$ , kun `Src` lähettää ajanhetkellä `T` paketin `Type-P` `Dst`:lle ja `Dst` lähettää välittömästi `Type-P` paketin `Src`:lle takaisin, ja `Src` vastaanottaa tämän paketin. Jos  $dT$  saa arvon määrittelemätön, `Dst` ei ole saanut `Type-P` -pakettia, tai ei ole vastannut siihen, tai `Src` ei ole saanut `Type-P` -pakettia takaisin.

### 2.4.2 Mittaustavoista

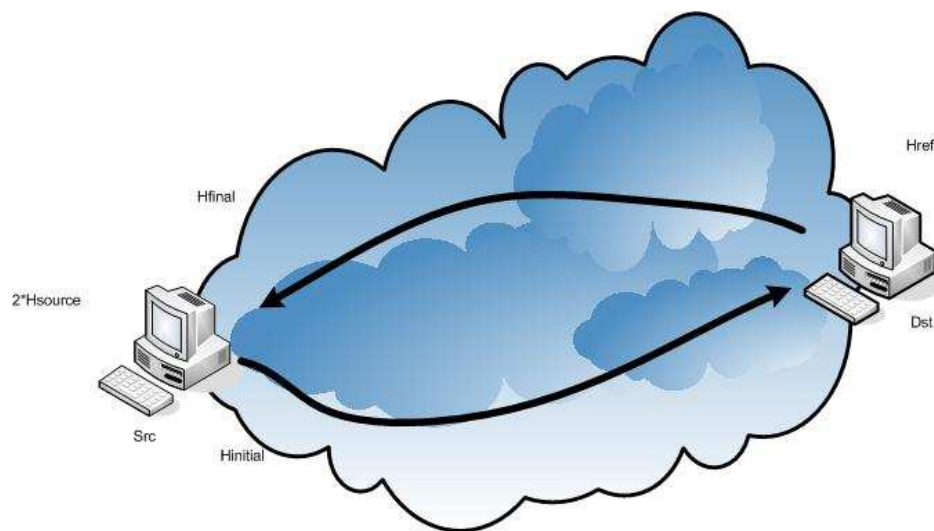
Kuten muissakin suorituskykyparametreissa, tässäkin tapauksessa algoritmin tarkka määrittely riippuu käytettävästä testipaketista. Algoritmin perusperiaate on kuitenkin seuraavanlainen:

`Src` määrittelee IP-osoitteet ja muodostaa `Type-P` -testipaketin, joka muodostetaan halutun kokoiseksi satunnaisella täytteellä. Paketissa täytyy olla jokin tunniste, jotta voidaan tunnistaa vastauspaketti, jonka `Dst` lähettää. `Dst`:ssä valmistaudutaan vastaanottamaan paketti ja vastaamaan siihen mahdollisimman nopeasti. `Src`:ssä valmistaudutaan vastaanottamaan vastauspaketti. `Src`:ssä asetetaan pakettiin aikaleima ja lähetetään se `Dst`:lle. jos paketti saapuu `Dst`:lle, lähetetään vastauspaketti `Src`:lle mahdollisimman nopeasti. Jos vastauspaketti saapuu järkevän ajan kuluessa `Src`:lle, otetaan mahdollisimman nopeasti talteen paketin saapumisaika. Vähentämällä nämä aikaleimat keskenään saadaan edestakainen viive. Jos paketti ei saavu järkevän ajan kuluessa, saa edestakainen viive arvon määrittelemätön.

### 2.4.3 Virheanalyysistä

Virheanalyysi menee kutakuinkin samalla lailla kuin yhdensuuntaisessa viiveessä. Tahdistuksen kanssa ei tule samanlaisia ongelmia kuin yhdensuuntaisessa viiveessä, koska mittauksessa lähettäjä ja vastaanottaja ovat sama laite. Resoluutiosta koituva virhe voidaan naivisti laskea olevan  $2 * R_{source}$ . Mittausajan ja siirtoajan välinen erotus tuo virhettä tulokseen kuten yhdensuuntaisessakin viiveessä. Virhettä tulee Src:n lähettäessä ja vastaanottaessa pakettia. Merkitään  $H_{initial}$ :lla ylärajaa virheelle lähetettäessä ja  $H_{final}$ :lla virheen ylärajaa vastaanottaessa pakettia. Kokonaisvirhe, joka muodostuu mittausajan ja siirtoajan erotuksesta on siis:  $H_{initial} + H_{final}$ . Virhettä tulokseen tuo myös Dst:n päässä tapahtuvat paketin käsittelyt. Merkitään paketin käsittelystä tulevaa virhettä  $H_{refl}$ :llä. Tästä saadaan kokonaisvirheelle arvioksi

$$2 * R_{source} + H_{initial} + H_{final} + H_{refl}.$$



Kuva 2.3: Edestakainen viive ja virheet.

## 2.5 Viiveen vaihtelu

Viiveen vaihtelu (engl. Delay Variation), eli ipdv, tarkoittaa nimensä mukaisesti sitä, kuinka paljon pakettien yhdensuuntainen viive vaihtelee lähettäjän ja vastaanottajan välisellä polulla. Viiveen vaihtelu vaikuttaa monien sovellusten toimintaan. Sovellukset, joille on tärkeää että paketit saapuvat tietyllä tahdilla, voivat asettaa vastaanottopuskurinsa koon sopivaksi, jos viiveen vaihtelu tiedetään.

C. Demichelis ja P. Chimento määrittelevät dokumentissaan

'IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)' [8] suorituskykyparametrin viiveen vaihtelulle ja määrittävät mittaustavan tälle parametrille. Viiveen vaihtelusta käytetään joissain yhteyksissä myös termiä *Jitter*. Tässä dokumentissa sitä ei käytetä, koska se saattaa aiheuttaa sekaannusta eri yhteyksissä saamansa merkityksen vuoksi.

### 2.5.1 Parametri

Viiveen vaihtelu -parametrin attribuutit:

**Parametrin nimi:** `Type-P-One-way-ipdv`

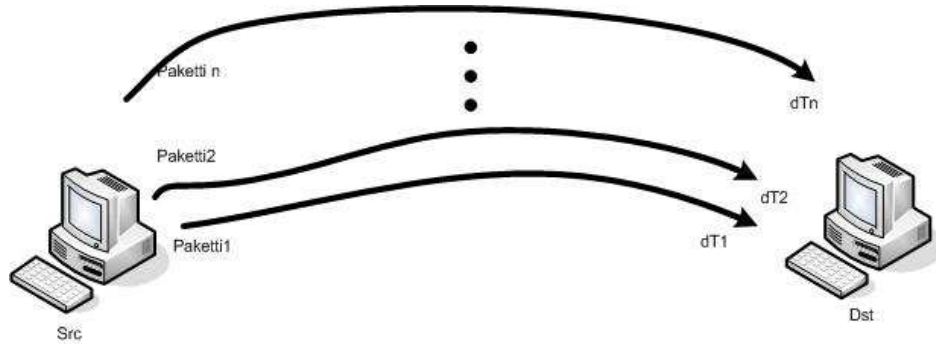
- `Src` = Lähettäjän IP-osoite
- `Dst` = Vastaanottajan IP-osoite
- `T1` = Aika
- `T2` = Aika
- `L` = Mitattavien pakettien pituus bitteinä
- `F` = Valintafunktio, jolla mittauspaketit valitaan yksiselitteisesti tietovirrasta
- `I1, I2` = Mittausaikajakson alku ja loppu
- `P` = Mittauspaketin tyyppi

Parametri saa arvokseen reaalisen määrän sekunteja (positiivinen, negatiivinen tai nolla) tai määrittelemätön (ei voitu mitata).

Tämän parametrin mittaukseen täytyy käyttää tietovirtaa (engl. *stream*). Olkoon meillä `Type-P` -pakettien tietovirta siten, että ensimmäinen mittauspaketti ohittaa mittauspaikan 1 ajanhetken `I1` jälkeen saaden indeksin 0 ja viimeinen paketti ohittaa mittauspaikan ennen ajanhetkeä `I2` saaden korkeimman indeksin arvon. Mittausapaikat ovat verkon laitteet, joiden välistä viivettä mitataan. Mittauspaikat voivat olla `Src` ja `Dst`, tai joku muu ävli `Src:n` ja `Dst:n` väliseltä polulta.

`Type-P-One-way-ipdv` mitataan kahdesta `Src:ltä` `Dst:lle` kulkevasta paketista, jotka valitaan valintafunktiolla `F`. Parametrin arvo on `Type-P-One-way-Delay:n` arvon ajanhetkellä `T2` ja `Type-P-One-way-Delay:n` arvon ajanhetkellä `T1` erotus. `T1` ja `T2` ovat ensimmäisen ja toisen mittauspaketin lähetysajat. Ensimmäinen

paketti vastaanotetaan ajanhetkellä  $T_1 + dT_1$  ja toinen ajanhetkellä  $T_2 + dT_2$ . Näistä saadaan viiveenvaihtelulle arvo  $ddT = dT_2 - dT_1$ . Jos arvoksi saadaan määrittelemätön, Dst ei saanut jompaakumpaa tai kumpaakaan pakettia.



Kuva 2.4: Viiveen vaihtelu.

Jos kuvassa valittaisiin paketit 2 ja n, olisi viiveen vaihtelu  $ddT = dT_n - dT_2$ .

### 2.5.2 Mittaustavoista

Viiveen vaihtelun mittausalgoritmin tarkat yksityiskohdat riippuvat käytetystä mittauspaketista, mutta algoritmin periaate on seuraavanlainen. Tässä tapauksessa algoritmi mittaa viiveen vaihtelua reaaliajassa, mutta yhtä hyvin se voitaisiin määrittää aiemmin mitatuista viiveistä jälkeempään. Ajanhetken  $I_1$  jälkeen valitaan Src:ssä lähettäjän ja vastaanottajan IP-osoitteet ja muodostetaan `Type-P` -paketit. Dst:ssä valmistaudutaan vastaanottamaan paketteja. Src:ssä laitetaan pakettiin aikaleima ja lähetetään se Dst:lle. Jos paketti saapuu järkeväen ajan kuluessa Dst:lle, otetaan aikaleima paketista ja lasketaan arvio yhdensuuntaiselle viiveelle. Jos paketti täyttää valintafunktion  $F$  valintakriteerin, tallennetaan yhdensuuntaisen viiveen arvo. Muutoin jatketaan pakettien generoimista, kunnes kriteeri täyttyy tai ajanhetki  $I_2$  ohitetaan. Src jatkaa pakettien lähettämistä. Jos paketti saapuu perille järkeväen ajan kuluessa, lasketaan viiveelle arvio kuten yllä. Jos paketti täyttää valintakriteerin, tallennetaan sen viive ja lasketaan viiveen vaihtelulle arvo vähentämällä ensimmäinen viive jälkimmäisestä. Muutoin jatketaan kunnes kriteeri täyttyy tai ajanhetki  $I_2$  ohitetaan.

### 2.5.3 Virheanalyysistä

Ensimmäinen oletus voisi olla, että kahden eri viiveen virheet kumoavat toisensa, kun viiveen vaihtelu lasketaan yhteen. Näin ei kuitenkaan ole, sillä osa virheistä

muutuu ajan suhteen. Tahdistusvirheet lasketaan kuten aikaisemminkin molemmille viiveille.

Virheitä jotka muuttuvat ajansuhteen ovat vääristymä ja pitkän aikavälin siirtymä (eng. *drift*). Ajatellaan, että vääristymä on ajan suhteen lineaarinen funktio. Tällöin myös kellojen poikkeama (engl. *offset*) on ajan funktio. Virhe on tällöin  $e(t) = K \cdot t + O$ , jossa  $K$  on vakio ja  $O$  on poikkeama. Kun vähennetään kaksi aikaleimaa toisistaan, saadaan virhe  $e(t_2) - e(t_1) = K \cdot (t_2 - t_1)$ , jossa ( $t_2 > t_1$ ). Koska pitkän aikavälin siirtymää ei voida jättää huomioimatta, mutta oletetaan että se on ajan suhteen lineaarinen funktio, tällöin vääristymä on  $s(t) = M \cdot (t^2) + N \cdot t + s_0$ , jossa  $M$  ja  $N$  ovat vakioita ja  $s_0$  on vääristymä ajanhetkellä 0. Kun otetaan mukaan vääristymä ja pitkän aikavälin siirtymä, saadaan virheelle muoto  $e(t) = O + s(t)$ . Kahden aikaleiman vähennyslaskussa virhe on siis  $e(t_2) - e(t_1) = N \cdot (t_2 - t_1) + M \cdot [(t_2 - t_1)^2]$ .

## 2.6 Hävikkimallit

Hävikkimallit (engl. *Loss patterns*) kuvaavat hävikin jakautumista. Hävikin jakautuminen vaikuttaa monien reaaliaikasovellusten, kuten VoIP-sovellusten suorituskykyyn huomattavasti. Hävikin määrän pysyessä samana, voi suorituskyky muuttua rajustikin hävikin erilaisella jakautumisella.

R. Koodli ja R. Ravikanth määrittelevät dokumentissaan 'One-way Loss Pattern Sample Metrics' [7] kaksi suorituskykyparametria hävikin etäisyyden ja hävikkijakson, sekä mittaustavan näille parametreille. Dokumentissa esitellään myös näiden parametrien avulla saatavia tilastotietoja hävikin jakautumisesta.

### 2.6.1 Parametrit

Parametreja on kaksi, ja niillä on molemmilla samat attribuutit.

Hävikin etäisyys ja hävikkijakso-parametrien attribuutit:

**Parametrien nimet:** `Type-P-One-way-Loss-Distance-Stream` ja `Type-P-One-way-Loss-Period-Stream`

- `Src` = Lähettäjän IP-osoite
- `Dst` = Vastaanottajan IP-osoite
- `T0` = Aika



- $T_f = \text{Aika}$
- $\lambda = \text{Näytteenottotajuus ajan käänteislukuna}$

Parametri hävikin etäisyys saa arvokseen sarjan lukupareja  $\langle \text{loss distance}, \text{loss} \rangle$ , jossa  $\text{loss}$  on ykkönen tai nolla ja  $\text{loss distance}$  joko nolla tai positiivinen kokonaisluku.

Parametri hävikkijakso saa arvokseen sarjan lukupareja  $\langle \text{loss period}, \text{loss} \rangle$ , jossa  $\text{loss}$  on kuten yllä ja  $\text{loss period}$  on kokonaisluku.

Hävikin etäisyys -parametrin arvo kertoo sen, kuinka monta pakettia on kulunut edellisestä hävinneestä paketista. Kun paketti todetaan hävinneeksi paketiksi luvussa 2.3 kerrotulla tavalla, otetaan paketista sen järjestysnumero ja verrataan sitä edellisen hävinneen paketin järjestysnumeroon. Hävikin etäisyys on näiden lukujen erotus. Ensimmäisen hävinneen paketin hävikin etäisyys on nolla.

Hävikkijaksoa mitattaessa aloitetaan laskuri  $n$  arvosta nolla. Kun paketti todetaan luvun yhdensuuntainen hävikki tavalla hävinneeksi,  $\text{loss period}$ :iin laitetaan  $n:n$  arvo. Jos paketti ei hävinnyt,  $\text{loss period}$ :n arvo on nolla. Laskurin  $n$  arvoa lisätään aina yhdellä, kun seuraava ehto täyttyy:

Olkoon  $P_i$  :s paketti.  $f(P_i) = 1$ , jos  $P_i$  on hävinnyt, muutoin 0. Tällöin hävikkijakso alkaa, jos  $f(P_i) = 1$  ja  $f(P_{i-1}) = 0$ . Eli laskuria kasvatetaan, kun hävikkijakso alkaa.  $\text{loss period}$ :n arvo kertookin, mihin hävikkijaksoon hävinnyt paketti kuuluu.

**Esimerkki:** Olkoon meillä seuraavanlainen jono tuloksena yhdensuuntaisen hävikin mitaamisesta:

$\langle T1, 0 \rangle, \langle T2, 1 \rangle, \langle T3, 0 \rangle, \langle T4, 0 \rangle, \langle T5, 1 \rangle$   
 $, \langle T6, 0 \rangle, \langle T7, 1 \rangle, \langle T8, 0 \rangle, \langle T9, 1 \rangle, \langle T10, 1 \rangle$ . Tässä siis paketit, jotka on lähetetty ajanhetkillä  $T2, T5, T7, T9$  ja  $T10$  on hävinneitä. Tästä saadaan jonot hävikkien etäisyyksille ja hävikkijaksoille.

Type-P-One-way-Loss-Distance-Stream  
 $= \langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 0, 0 \rangle, \langle 0, 0 \rangle, \langle 3, 1 \rangle, \langle 0, 0 \rangle, \langle 2, 1 \rangle, \langle 0, 0 \rangle, \langle 2, 1 \rangle, \langle 1, 1 \rangle$   
 Type-P-One-way-Loss-Period-Stream  
 $= \langle 0, 0 \rangle, \langle 1, 1 \rangle, \langle 0, 0 \rangle, \langle 0, 0 \rangle, \langle 2, 1 \rangle, \langle 0, 0 \rangle, \langle 3, 1 \rangle, \langle 0, 0 \rangle, \langle 4, 1 \rangle, \langle 4, 1 \rangle$

## 2.6.2 Mittaustavoista

Mittaustapana voidaan käyttää samaa algoritmia, kuin yhdensuuntaisen hävikin mitaamiseen. Virheanalyysi on myös samanlainen kuin yhdensuuntaisen hävikin tapauksessa.

### 2.6.3 Tilastoja

Hävikkimallien parametreista on johdettu myös neljä muuta tilastollista parametria. Nämä parametrit ovat: huomattavuus, jaksojen määrä, jaksojen pituudet ja jaksojen välin pituus.

**Huomattavuus** (engl. *noticeable*) tarkoittaa, että jos kadonneen paketin ja edellisen kadonneen paketin väli on pienempi kuin delta, on katoaminen huomattava. Delta on tässä hävikkirajoite (engl. *loss constraint*).

**Jaksojen määrä** tarkoittaa nimensä mukaisesti hävikkijaksojen lukumäärää.

**Jaksojen pituudet** tarkoittavat hävikkijaksojen pituutta.

**Jaksojen välin pituus** tarkoittaa kahden hävikkijakson välistä välimatkaa.

## 2.7 Kaistan kapasiteetti

Kaistan kapasiteetin (engl. *Bandwidth Capacity*) mittaaminen kuulostaa yksinkertaiselta asialta, mutta on todellisuudessa melko monimutkainen asia. Asiaa vaikeuttaa myös yksikäsitteisen termistön puuttuminen. P. Chimento ja J. Ishac käsittelevät kaistan kapasiteettia dokumentissaan 'Defining Network Capacity' [9].

Jotta informaatio voitaisiin kuljettaa fyysisen median yli, täytyy se ensin koodata. Tämän jälkeen informaatio muunnetaan ,mediasta riippuen, jollain systeemillä signaalijonoksi. Joissain medioissa signaalien maksimitaajuuden voidaan katsoa olevan kaistan kapasiteetti, mutta eri medioissa signaalien siirtotaajuus ja informaation kuljetuskapasiteetti on eri asia. Tässä tutkielmassa keskitytään informaation kuljetuskapasiteettiin. Informaation todellista kuljetuskapasiteettia pienentää hieman eri protokollakerroksien asettamat kehykset ja koodaus. Radiotiellä väliaineen koostumus asettaa siirrolle erityisvaatimuksia, jotka myös vähentävät todellista kapasiteettia.

### 2.7.1 Määrytykset

Tässä luvussa käsitellään kapasiteetille asetettuja määrytyksiä.

**Fyysisen linkin nimellinen kapasiteetti** (engl. *Nominal physical link capacity*): Tarkoittaa linkin teoreettista maksimiarvoa kuljetettavaan datan määrälle. Tämä on fyysisen kerroksen parametri eikä IP-kerroksen. Tämä arvo on yleensä vakio ajan suhteen.

Loput määreet on ilmoitettu IP-kerroksen bitteinä (engl. *IP layer bits*), joka on kaikkien vastaanotettujen IP-pakettien oktetien, kehyksen ensimmäisestä kuorman

viimeiseen, määrä kerrottuna kahdeksalla. Lisäksi määritellään polku  $P$ , joka on  $n:n$  pituinen ja koostuu linkeistä  $(L_1, L_2, \dots, L_n)$ . Linkit yhdistävät toisiinsa verkon laitteet  $(N_1, N_2, \dots, N_n, N_{n+1})$ , joista  $Src$  on  $N_1$  ja  $Dst$   $N_{n+1}$ . Laitteiden  $N_2 \dots N_n$  ei välttämättä tarvitse olla reitittämiä. IP-kerroksen bitit tallennetaan  $Dst$ :ssä alkaen ajanhetkellä  $T$  ja päättyen ajanhetkellä  $T+I$ .

**IP-kerroksen linkin kapasiteetti** (engl. *IP layer link capacity*): Määritellään IP-kerroksen linkin kapasiteetti  $C(L, T, I)$  tarkoittamaan maksimimäärää IP-kerroksen bittejä, joka voidaan oikein siirtää  $Src$ :ltä  $Dst$ :lle linkin  $L$  yli aikajakson  $[T, T+I]$  aikana, jaettuna  $I$ :llä.

**IP-kerroksen polun kapasiteetti** (engl. *IP layer path capacity*): Saadaan edellisestä siten, että koko polun kapasiteetti on sama kuin polulla olevan linkin, jolla on pienin kapasiteetti. Eli polun kapasiteetti on  $C(P, T, I) = \min_{1..n} C(L_n, T, I)$ . Edellisten kapasiteettien mittausta varten tulisi mitattavan polun tai linkin resurssien olla vapaina. Käytännöllisempää on mitata ruuhkaisen linkin tai polun kapasiteettia.

**IP-kerroksen linkin käytettävyys** (engl. *IP layer link usage*): Tarkoittaa linkin  $L$  keskimääräistä käytettävyyttä.  $Used(L, T, I)$  on todellinen määrä IP-kerroksen bittejä, jotka on oikein siirretty linkin  $L$  yli aikajakson  $[T, T+I]$  aikana.

**IP-kerroksen linkin keskimääräinen käyttöaste** (engl. *Average IP layer link utilization*): Tarkoittaa sitä, kuinka suuri osa linkin kapasiteetista on käytössä. Käyttöaste saadaan jakamalla käytettävyys kapasiteetilla:  $Util(L, T, I) = (Used(L, T, I) / C(L, T, I))$ . Kertomalla tulos sadalla saadaan käyttöaste prosentteina.

**IP-kerroksen linkin vapaana oleva kapasiteetti** (engl. *IP layer available link capacity*): Kertoo sen, kuinka paljon linkin kapasiteetista on vapaana. Tämä saadaan edellisen tuloksen avulla:  $Availcap(L, T, I) = C(L, T, I) * (1 - Util(L, T, I))$ .

**IP-kerroksen polun vapaana oleva kapasiteetti** (engl. *IP layer available path capacity*): Saadaan samoin, kuin aiemmin polun kapasiteetti. Polun vapaana oleva kapasiteetti on sama, kuin polulla olevan linkin, jolla on pienin vapaana oleva kapasiteetti. Eli  $Availcap(P, T, I) = \min_{1..n} Availcap(L_n, T, I)$ . Vapaana oleva kapasiteetti on herkempi virheille kuin kapasiteetti, joten aika ja aikaintervalli täytyy määrittää tarkasti koska niillä on suuri vaikutus mittaustulokseen.

## 2.8 Hyötykuormakapasiteetti

Hyötykuormakapasiteetti (engl. *Bulk transport capacity*) kertoo verkon hyötykuorman kuljetuskyvyn kulutettuun aikaan nähden. Tähän parametriin ei lasketa mukaan kehyksiin kuluvia bittejä. IPPM-työryhmä ei ole määritellyt tähän suorituskyky-parametria, mutta M. Mathis ja M. Allman selvittävät periaatteen tällaiselle parametrille dokumentissaan 'A Framework for Defining Empirical Bulk Transfer Capacity Metrics' [10].

Hyötykuormakapasiteetti tarkoittaa siis kuljetettua hyötykuormaa aikajaksossa. Kapasiteetin määrittämisestä tekee monimutkaisen se, että eri kuljetusprotokollilla on erilaiset ruuhkanhallintamekanismit. Hyötykuormakapasiteetti määrää joillekin sovelluksille, esim. www- tai FTP-sovelluksille, suoraan sovelluksen käyttöajan ja käyttäjälle näkyvän suorituskyvyn. Hyötykuormakapasiteetin mittauksen yhteydessä on syytä kerätä muutakin informaatiota verkkoliikenteestä. Ideaalinen mittaustapa on siis  $BTC = \text{lähetetty data} / \text{kulunut aika}$ , jossa lähetetty data ei sisällä kehyksiin kuluvia bittejä. Hyötykuormakapasiteetin mittaus menetelmän täytyy tukea ruuhkanhallintamenetelmiä, joita TCP<sup>6</sup>:kin tukee.

### 2.8.1 Muut mitattavat parametrit

Hyötykuormakapasiteettia mitattaessa voidaan mitata myös muita parametreja, jotka antavat lisätietoa verkosta ja käytettävästä ruuhkanhallintamenetelmästä.

**Ruuhkan välttämiskapasiteetti** (engl. *Congestion avoidance capacity*) tarkoittaa kuljetuskapasiteettia, joka ei vielä laukaise uudelleenlähetystä.

**Tahdistuksen menetyksen välttäminen** (engl. *Preservation of self-clock*) tarkoittaa sitä, että kuittausviestin katoaminen voi vaikuttaa hyötykuormakapasiteetin arvoon huomattavasti, koska katoamisesta toipuminen vie aikaa kun uudelleentahdistus suoritetaan. Tämän vuoksi toipumiseen kulunut aika täytyy ottaa huomioon. Seuraavat neljä tilannetta voivat aiheuttaa tämän.

**Menetetty lähetysmahdollisuudet** (engl. *Lost transmission opportunities*) ovat sellaisia tilanteita, kun kuittausviestin saavuttua ajastin laukeaa ennenkuin seuraavaa pakettia ehditään lähettämään. Tämän vuoksi kapasiteetin tilasta on kerättävä tietoa, jotta tulevaisuudessa ruuhkanhallintamenetelmillä tällaisia tapauksia voitaisiin välttää paremmin.

**Kokonaisen ikkunan katoaminen** (engl. *Losing entire window*) voidaan havaita, jos viimeinen tapahtuma ennen ajastimen laukeamista oli lähetys. Tällaisessa

---

<sup>6</sup>Transmission Control Protocol

tapauksessa käynnistyy uudelleenlähetyks ja ikkunan koko alustetaan uudelleen. Tämä vaikuttaa merkittävästi hyötykuormakapasiteettiin. Mittausmenetelmien on tämän vuoksi raportoitava kokonaisten ikkunoiden menettämistä.

**Urhea tahdistuksen säilytys** (engl. *Heroic clock preservation*) tarkoittaa joidenkin ruuhkanhallinta-algoritmien tapaa toimia tilanteissa joissa menetetään paljon paketteja. Algoritmit yrittävät saada itsensä uudestaan tahdistettua, kun toiset algoritmit eivät niin tekisi. Tämä vaikuttaa hyötykuormakapasiteetin arvoon, joten mittausmenetelmän täytyy raportoida katoamisten jakaumasta, jotta muiden ruuhkanhallinta-algoritmien suorituskykyä voitaisiin arvioida.

**Ajastimen väärät laukeamiset** (engl. *False timeouts*) ovat sellaisia, joissa uudelleenlähetyksajastin laukeaa ennenkuin kuittausviesti saapuu jollekin aikaisemmin lähetetylle paketille. Tällaiset tilanteet täytyy ottaa huomioon mittausmenetelmässä.

**Tietovirrasta riippuvat polun ominaisuudet** (engl. *Flow based path properties*) vaikuttavat myös hyötykuormakapasiteettiin, joten nämä ominaisuudet on otettava huomioon mittausmenetelmissä.

**Lähetystahdin tarkastus** on tärkeää, jottei käy niin että, mittausalusta on itse polun pullonkaula. Jos lähetyksessä alkaa muodostua merkittävää jonoa lähetykspuskureihin, täytyy lähetyksahtia muuttaa.

**Kaksisuuntainen mittaaminen.** Jos käytetään mittaustekniikkaa, jossa päätelaitteet eivät toimi yhteistyössä, eli voidaan mitata vain edestakainen kapasiteetti, täytyy molempien suuntien kapasiteetit erotella tuloksesta jollain muulla tavalla.

## 2.9 Pakettien uudelleenjärjestely

Pakettien uudelleenjärjestelyllä (engl. *Packet reordering*) tarkoitetaan sitä tilannetta, kun paketit saapuvat perille eri järjestyksessä kuin ne on lähetetty. A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov ja J. Perser käsittelevät pakettien uudelleenjärjestelyä ja määrittelevät sille suorituskykyparametrin dokumentissaan 'Packet Reordering Metric for IPPM' [11]. Dokumentissaan he määrittelevät myös muita parametreja liittyen uudelleenjärjestelyyn.

Pakettien uudelleenjärjestely on joillekin sovelluksille, etenkin reaaliaikaisille verkosovelluksille, suuri haitta. Paketit jotka saapuvat myöhässä voidaan hylätä. Tämä näkyy esimerkiksi reaaliaikaisessa videolähetyksessä häiriönä. Pakettien saapumisjärjestys saattaa muuttua matkalla monesta syystä. Esimerkiksi jos kaksi eri reittiä eroavat vain hieman siirtoajassa, voivat pidempää reittiä kulkevat paketit tulla myöhemmin perille.

### 2.9.1 Parametri

Pakettien uudelleenjärjestely -parametrin attribuutit:

**Parametrin nimi:** `Type-P-Reordered`

- `Src` = Lähettäjän IP-osoite
- `Dst` = Vastaanottajan IP-osoite
- `SrcTime` = Paketin lähtöaika
- `s` = Pakettin vuoronumero
- `NextExp` = Seuraavana odotettava vuoronumero

Parametri saa arvokseen tosi tai epätosi.

Jos ajanhetkellä `SrcTime` lähetetty paketti `s` todetaan uudelleenjärjestellyksi vertaamalla vuoronumeroa `NextExp`-arvoon, on `Type-P-Reordered:n` arvo tosi. Muutoin arvo on epätosi. Uudelleenjärjestely todetaan, jos  $s < \text{NextExp}$ . Tällöin `NextExp:n` arvo ei muutu. Jos arvoksi saadaan epätosi, eli  $s \geq \text{NextExp}$ , tulee `NextExp:n` arvoksi  $s+1$ . Seuraavana odotettavan paketin arvo ei voi pienentyä, koska käytetään kasvavaa vuoronumerointia. Tämä takaa edellisen menetelmän toimivuuden.

### 2.9.2 Muut parametrit

Dokumentissa [11] määritellään myös muita parametreja jotka liittyvät pakettien uudelleenjärjestelyyn. Parametrit kuvaavat muunmuassa uudelleenjärjestelyn laajuutta. Seuraavassa on lueteltuna nämä parametrit.

**Uudelleenjärjestelyn suhde** (engl. *Reordering ratio*) kertoo, kuinka paljon paketteja on uudelleenjärjestelty suhteessa oikeassa järjestyksessä saapuneisiin. **Uudelleenjärjestelyn etäisyys** (engl. *Reorderin extent*) kuvaa sitä kuinka kaukana uudelleenjärjestelty paketti on omalta paikaltaan. Tällä voidaan arvioida esimerkiksi tarvittavien puskurien kokoa, jotta järjestys voidaan palauttaa. **Myöhästymisaika** (engl. *Late time*) kuvaa nimensä mukaan uudelleenjärjestellyn paketin etäisyyttä omalta paikaltaan ajallisesti. **Uudelleenjärjestelyn vastine tavuina** (engl. *Byte offset*) on vaihtoehtoinen tapa määrittää esimerkiksi tarvittava puskurien koko. **Epäjatkuvuusväli** (engl. *Gap between reordering discontinuities*) on parametri, jota mitataan ajallisesti ja paikkaan liittyen. Parametrilla mitataan uudelleenjärjestellyn paketin aiheuttaman

epäjatkuvuuden kesto. **Uudelleenohjausvapaat jaksot** (engl. *Reordering-free runs*) ovat jaksoja, jolloin paketit saapuvat oikeassa järjestyksessä.

### 3 Yhteenveto

Tässä tutkielmassa käsiteltiin siis IP-verkkojen suorituskyvyn mittaamista, ja suorituskykyparametreja. Tutkielman parametrit ja mittaustavat ovat IETF:n IPPM-työryhmän määrittelemiä. Työryhmän työn tarkoitus oli määrittellä standardit suorituskykyparametrit ja mittaustavat IP-verkoille, ja tämän seurauksena parantaa IP-verkkojen palvelunlaatua. Parametrien tarkkuudeksi ei kelvannut hyvä-huono -vertailu.

Ensimmäinen luku johdatteli lukijan aiheeseen. Toisessa luvussa käsiteltiin suorituskykyparametrit ja esiteltiin niiden mittaamiseen käytettävät algoritmit. Parametreista ennaltaan monelle tuttuja ovat varmasti viive, hävikki ja viiveen vaihtelu. Näille parametreille tuo lisäarvoa niiden eri vaihtoehtoiset mittaustavat ja tarkempi parametrien erottelu. Esimerkiksi viivettä voidaan mitata eri tavoin. Hävikkimallit tuovat lisätietoa ja -arvoa hävikin jakautumisesta. Kaikenkaikkiaan tarkemmin määritellyt parametrit ja mittaustavat tarjoavat kattavamman näkökulman IP-verkkojen suorituskyvyn määrittämiseen.

Parametrien ja mittaustapojen määrityksen yhteydessä tarkasteltiin myös mittaamiseen liittyviin virhelähteisiin ja virheanalyysiin. Koska mitattavat arvot ovat usein todella pieniä, on virheanalyysi erittäin tärkeässä roolissa. Virhe voi aiheuttaa mittauksissa sen, että mittaukset antavat erittäin väärän kuvan verkon suorituskyvystä, esimerkiksi kaksinkertaisen arvon viiveelle.

Mittaustavat esitettiin pseudo-koodina tai sanallisesti, johtuen monesta eri asiasta, esimerkiksi käytettävästä mittauspaketista. Mittaustekniikoiden tarkempi ja yksityiskohtaisempi määrittely jätetään mittaussovelluksen suunnittelijoille ja toteuttajille.



## Viitteet

- [1] J. Mahdavi, V. Paxson, *IPPM Metrics for Measuring Connectivity*, RFC2678, Syyskuu 1999,
- [2] V. Paxson, G. Almes, J. Mahdavi, M. Mathis, *Framework for IP Performance Metrics*, RFC2330, Toukokuu 1998,
- [3] G. Almes, S. Kalidindi, M. Zekauskas, *A One-way Delay metric for IPPM*, RFC2679, Syyskuu 1999,
- [4] E. Stephan, L. Liang, A. Morton, *IP Performance Metrics (IPPM) for spatial and multicast*, Internet Draft, 22.10.2005,
- [5] G. Almes, S. Kalidindi, M. Zekauskas, *A One-way Packet Loss Metric for IPPM*, RFC2680, Syyskuu 1999,
- [6] G. Almes, S. Kalidindi, M. Zekauskas, *A Round-trip Delay Metric for IPPM*, RFC2681, Syyskuu 1999,
- [7] C. Demichelis, P. Chimento, *IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)*, RFC3393, Marraskuu 2002,
- [8] R. Koodli, R. Ravikanth, *One-way Loss Pattern Sample Metrics*, RFC3357, Elokuu 2002,
- [9] P. Chimento, J. Ishac, *Defining Network Capacity*, Internet Draft, 17.5.2006,
- [10] R. Koodli, R. Ravikanth, *A Framework for Defining Empirical Bulk Transfer Capacity Metrics*, RFC3148, Heinäkuu 2001,
- [11] A. Morton, L. Ciavattone, G. Ramanchandran, S Shalunov, J. Perser, *Packet Re-ordering Metric for IPPM*, Internet Draft, Huhtikuu 2006,