



UNIVERSITAS INDONESIA

**PENGUKURAN DAN ANALISIS PARAMETER QOS PADA
JARINGAN MOBILE IPV6 UNTUK APLIKASI GAME
ONLINE**

SKRIPSI

**VALDO MARCELINO
0806339383**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK KOMPUTER
DEPOK
JANUARI 2012**



UNIVERSITAS INDONESIA

**PENGUKURAN DAN ANALISIS PARAMETER QOS PADA
JARINGAN MOBILE IPV6 UNTUK APLIKASI GAME
ONLINE**

SKRIPSI

Skripsi ini diajukan untuk melengkapi sebagian persyaratan menjadi
Sarjana Teknik

VALDO MARCELINO
0806339383

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK KOMPUTER
DEPOK
JANUARI 2012

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Valdo Marcelino

NPM : 0806339383

Tanda Tangan :



Tanggal : 17 Januari 2012

HALAMAN PENGESAHAN

SKRIPSI INI DIAJUKAN OLEH :

NAMA : Valdo Marcelino
NPM : 0806339383
PROGRAM STUDI : Teknik Komputer
JUDUL SEMINAR : PENGUKURAN DAN ANALISIS
PARAMETER QOS PADA JARINGAN MOBILE
IPV6 UNTUK APLIKASI GAME ONLINE

**TELAH DIPRESENTASIKAN DAN DITERIMA SEBAGAI BAGIAN
PERSYARATAN YANG DIPERLUKAN UNTUK MEMPEROLEH
GELAR SARJANA TEKNIK PADA PERANGKAT LUNAK STUDI
TEKNIK ELEKTRO, FAKULTAS TEKNIK, UNIVERSITAS INDONESIA**

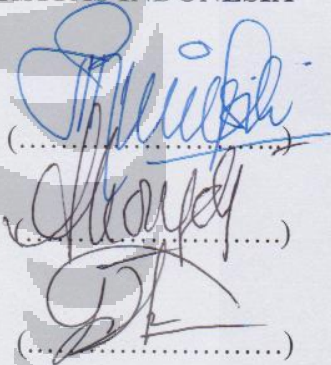
Pembimbing : Ir. A. Endang Sriningsih M.T. Si

Penguji 1 : Prima Dewi Purnamasari ST., MT., MSc.

Penguji 2 : I Gde Dharma ST. MT.

Ditetapkan di : Depok

Tanggal : 17 Januari 2012



KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, proses penulisan skripsi ini dapat terselesaikan. Penulisan skripsi ini dilakukan dalam rangka memenuhi persyaratan dari mata kuliah Skripsi yang terdapat dalam kurikulum program studi Teknik Komputer Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan skripsi, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1). Ir. A. Endang Sriningsih M.T. Si , selaku dosen pembimbing, serta dosen-dosen lainnya, yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2). Orang tua dan keluarga saya yang telah memberikan bantuan berupa dukungan material dan moral;
- (3). Para Sahabat dan teman-teman yang telah banyak membantu saya dalam menyelesaikan penyusunan skripsi ini.

Akhir kata, saya berharap agar Tuhan Yang Maha Esa berkenan membalas segala kebaikan dari semua pihak yang telah membantu. Semoga skripsi ini dapat memberikan manfaat bagi pengembangan ilmu.

Depok, 17 Januari 2012



Valdo Marcelino

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI
UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Valdo Marcelino
NPM : 0806339383
Program Studi : Teknik Komputer
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

**PENGUKURAN DAN ANALISIS PARAMETER QOS PADA JARINGAN
MOBILE IPV6 UNTUK APLIKASI GAME ONLINE**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis / pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 17 Januari 2012

Yang menyatakan,



(Valdo Marcelino)

ABSTRAK

Nama : Valdo Marcelino
Program Studi : Teknik Komputer
Judul : PENGUKURAN DAN ANALISIS PARAMETER QOS
PADA JARINGAN MOBILE IPV6 UNTUK APLIKASI
GAME ONLINE

Mobile IPv6 adalah teknologi jaringan komputer yang mendukung mobilitas *user* untuk berpindah dari satu jaringan ke jaringan lain tanpa harus memutuskan koneksi. Proses perpindahan dalam teknologi *Mobile IPv6* disebut *handover*. Pada skripsi ini telah dibuat sebuah jaringan *Mobile IPv6* untuk mengukur parameter QoS dengan skenario *vertical handover* dan *horizontal handover*. Skripsi ini menggunakan aplikasi *game online* yang akan diukur QoSnya. Pengukuran dilakukan dengan cara memainkan aplikasi antara server dan client. Dari hasil pengukuran QoS, dapat dilihat bahwa proses *horizontal handover* memiliki performa yang lebih baik dibandingkan proses *vertical handover*. Selisih performa skenario *vertical handover* dengan *horizontal handover* untuk parameter *delay*, *packet loss*, dan *transfer rate* adalah 17,5%, 3,504%, dan 2,06%.

Kata kunci:

Quality of Service, QoS, Mobile IPv6, Jaringan komputer, Game online, MIPv6, Handover

ABSTRACT

Name : Valdo Marcelino
Program Study : Computer Engineering
Title : QOS PARAMETERS MEASUREMENT AND
ANALYSIS OF MOBILE IPV6 NETWORK FOR GAME
ONLINE APPLICATION

Mobile IPv6 is computer networking technology which supports user mobility to move between networks without disconnecting current connection. Movement process in Mobile IPv6 is called handover. In this thesis, a Mobile IPv6 network has been made to measure QoS parameters with vertical handover and horizontal handover scenario. This thesis will measure QoS of online game. Measurement will be done by playing the application between server and client. The result shows that horizontal handover has better performance than vertical handover. The difference between vertical handover and horizontal handover scenario performance for *delay*, *packet loss*, and *transfer rate* are 17,5%, 3,504%, and 2,06%.

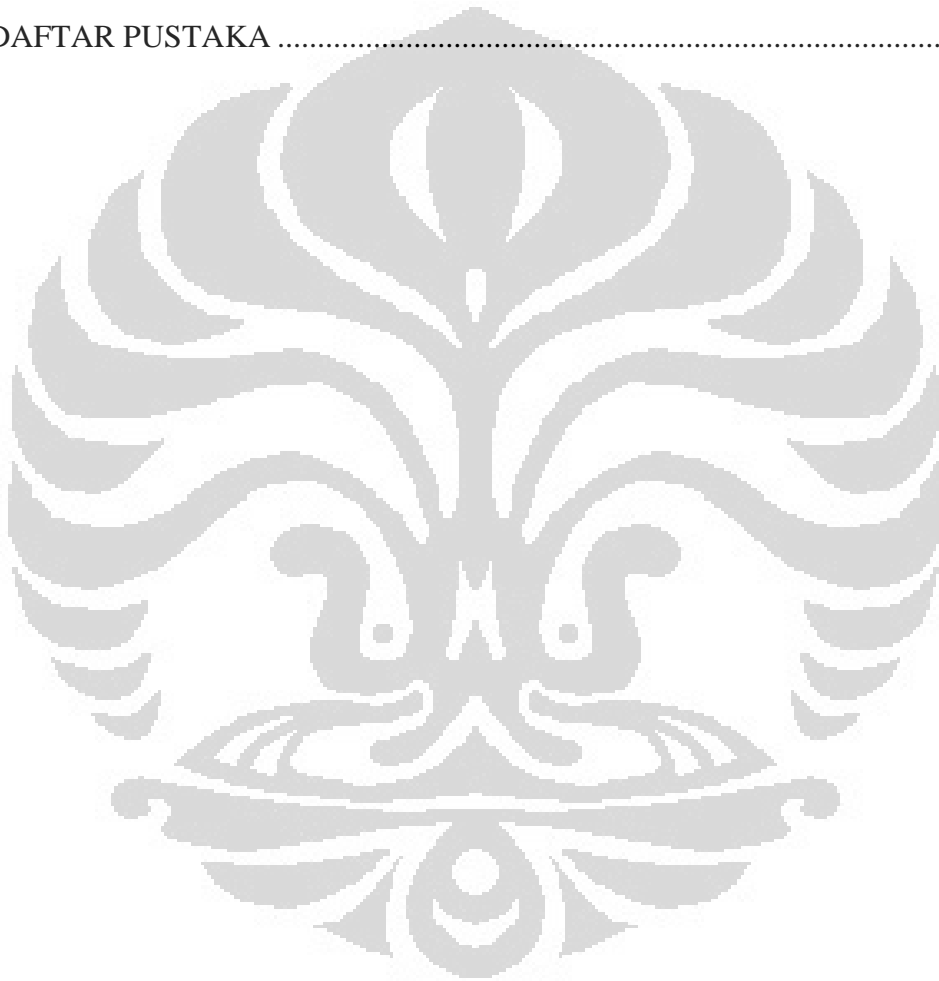
Keyword :

Quality of Service, QoS, Mobile IPv6, Computer network, Online game, MIPv6, Handover

DAFTAR ISI

SKRIPSI.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS	v
ABSTRAK	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x
BAB I.....	1
1.1 Latar Belakang	1
1.2 Tujuan.....	2
1.3 Metodologi Penelitian	2
1.4 Batasan Masalah.....	2
1.5 Sistematika Penulisan.....	3
BAB II.....	4
2.1 IPv6	4
2.2 Mobile IPv6.....	7
2.3 Quality of Service.....	16
2.4 Game Online.....	18
BAB III	19
3.1 Topologi Jaringan.....	19
3.1.1 Horizontal Handover.....	19
3.1.2 Vertical Handover	20
3.2 Spesifikasi Sistem.....	20
3.2.1 Spesifikasi Perangkat Keras	20
3.2.2 Spesifikasi Perangkat Lunak	22
3.4 Cara Kerja Pengujian Jaringan	22
3.5 Pembuatan Sistem	23
3.5.1 Konfigurasi Alamat IPv6	24
3.5.2 Konfigurasi Node	25
3.5.3 Konfigurasi Perangkat Lunak	28
BAB IV	33
4.1 Pengujian Jaringan Mobile IPv6	33
4.2 Pengukuran dan Analisa.....	34
4.2.1 Pengukuran dan Analisa <i>Delay</i>	35

4.2.1.1 Pengukuran <i>Delay</i>	35
4.2.1.2 Analisa <i>Delay</i>	37
4.2.2 Pengukuran dan Analisa <i>Packet loss</i>	41
4.2.2.1 Pengukuran <i>Packet loss</i>	41
4.2.2.2 Analisa <i>Packet loss</i>	41
4.2.3 Pengukuran dan Analisa <i>Transfer rate</i>	43
4.2.3.1 Pengukuran <i>Transfer rate</i>	43
4.2.3.2 Analisa <i>Transfer rate</i>	44
BAB V	47
DAFTAR ACUAN	48
DAFTAR PUSTAKA	49

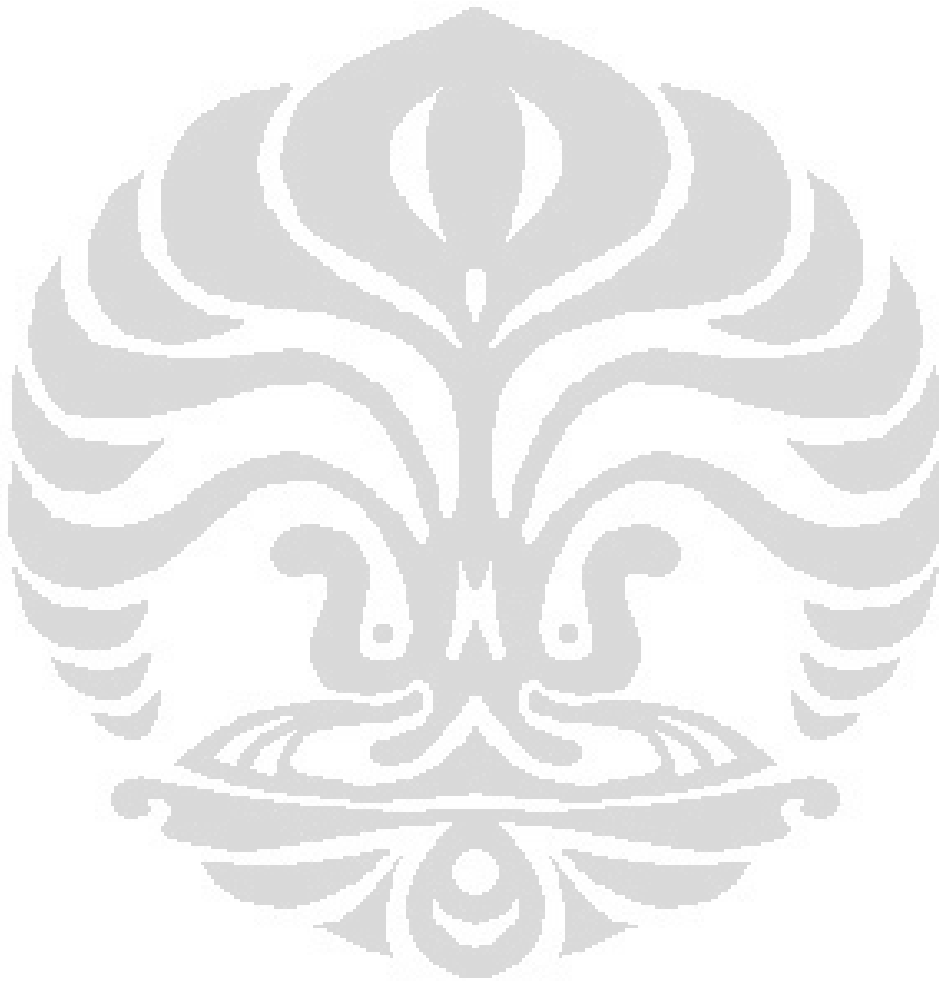


DAFTAR GAMBAR

Gambar 2.1 Contoh jaringan <i>Mobile IPv6</i> [1]	8
Gambar 2.2 <i>Horizontal Handover</i> [2].....	13
Gambar 2.3 <i>Vertical Handover</i> [2]	13
Gambar 2.0.4 Cara kerja jaringan <i>Mobile IPv6</i> [2].....	14
Gambar 3.1 Topologi jaringan untuk skenario <i>Horizontal Handover</i> pada <i>Mobile IPv6</i>	19
Gambar 3.2 Topologi jaringan untuk skenario <i>Vertical Handover</i> pada <i>Mobile IPv6</i>	20
Gambar 3.3 Tampilan Graphical User Interface (GUI) Wireshark[3].....	29
Gambar 3.4 Tampilan GUI Gtetrinet[4]	29
Gambar 4.1 Skenario Access Point Sama / Tanpa Handover	33
Gambar 4.2 Skenario Horizontal Handover.....	34
Gambar 4.3 Skenario Vertical Handover	34
Gambar 4.4 <i>Delay MN – CN</i>	36
Gambar 4.5 <i>Delay CN – MN</i>	36
Gambar 4.6 Grafik Hasil Pengukuran <i>Delay MN - CN</i> terhadap Waktu.....	37
Gambar 4.7 Grafik Hasil Pengukuran <i>Delay MN - CN</i> terhadap Skenario	38
Gambar 4.8 Grafik Hasil Pengukuran <i>Delay CN - MN</i> terhadap Waktu.....	39
Gambar 4.9 Grafik Hasil Pengukuran <i>Delay CN - MN</i> terhadap Skenario	40
Gambar 4.10 Grafik Hasil Pengukuran <i>Packet loss</i> terhadap Waktu	42
Gambar 4.11 Grafik Hasil Pengukuran <i>Packet loss</i> terhadap Skenario.....	42

DAFTAR TABEL

Tabel 2.1 Istilah dalam teknologi <i>Mobile IPv6</i>	10
Tabel 2.2 Rekomendasi ITU-T G.114 untuk <i>delay</i> [6].....	17
Tabel 4.1 Hasil Pengukuran <i>Delay</i>	37
Tabel 4.2 Hasil Pengukuran <i>Packet loss</i>	41
Tabel 4.3 Hasil Pengukuran <i>Transfer rate</i>	44



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, seluruh kegiatan yang terjadi di masyarakat membutuhkan komputer. Tentu saja, tidak mungkin sebuah komputer dapat digunakan untuk membantu kegiatan-kegiatan tersebut jika tidak terhubung dengan komputer lainnya. Untuk itulah dibutuhkan jaringan komputer, dimana komputer dengan jumlah yang sangat besar saling terhubung dan dapat berkomunikasi.

Pada jaringan komputer yang memiliki jumlah *host* yang besar, bagaimana cara membedakan *host* yang satu dengan *host* yang lain? Saat ini sudah ada teknologi yang bernama *Internet Protocol*, yang mendukung pengalamatan jaringan komputer. *Internet Protocol* saat ini merupakan *Internet Protocol* versi 4 (IPv4), dengan jumlah bit sebanyak 32, sehingga jumlah alamat *host* yang dimungkinkan adalah 2^{32} *host*. Jumlah *host* yang sangat banyak ini, seiring dengan perkembangan zaman, hampir seluruhnya terpakai sehingga teknologi IPv4 tidak lagi mampu mendukung pengalamatan jaringan komputer sekarang. Oleh karena itu, sejak tahun 1994 dikembangkanlah teknologi *Internet Protocol* versi 6 (IPv6) yang bertujuan untuk mengatasi masalah habisnya alamat IPv4. IPv6 memiliki jumlah bit sebanyak 128 bit untuk pengalamatannya, sehingga jumlah alamat *host* yang dimilikinya adalah 2^{128} *host*.

Teknologi IPv6 merupakan solusi dari permasalahan habisnya jumlah alamat. Namun masih ada satu lagi hal yang dimiliki oleh teknologi IPv6 agar dapat sepenuhnya menggantikan teknologi IPv4, yaitu mobilitas. Untuk mendukung mobilitas dalam IPv6, dikembangkanlah sebuah teknologi yang bernama *Mobile IPv6*. *Mobile IPv6* memungkinkan sebuah komputer yang terhubung ke jaringan komputer untuk berpindah tempat tanpa mengganti alamat (*IP address*) yang dimilikinya.

Perpindahan yang dimungkinkan oleh *Mobile IPv6* mendukung berbagai aplikasi, salah satunya *game online*. *Game online* saat ini sudah menjadi tren

dimana jumlah pemainnya sangat banyak. *Game online* tidak hanya terdapat di komputer saja, tetapi di *smartphone* dan *gadget-gadget* lain seperti iPad, iPod touch, juga ada *game online* yang sangat variatif. *Smartphone* dan *gadget-gadget* tersebut memiliki mobilitas yang sangat tinggi, hal ini yang menyebabkan *Mobile IPv6* sangat mendukung perkembangan *game online*.

1.2 Tujuan

Tujuan dari Skripsi ini adalah untuk mengukur dan menganalisa parameter QoS pada game online yang dijalankan di jaringan *Mobile IPv6* dengan menggunakan 2 skenario *handover* yang berbeda, yaitu *Vertical Handover* dan *Horizontal Handover*, dengan skenario tanpa *handover* sebagai pembanding

1.3 Metodologi Penelitian

Metode yang digunakan untuk Skripsi ini adalah :

1. Studi literatur, dengan membaca dan menggunakan buku sebagai referensi untuk Skripsi ini
2. Perancangan jaringan, setelah studi literature dan , maka dilakukan perancangan jaringan secara teori untuk mengukur dan menganalisis parameter QoS dalam jaringan *Mobile IPv6* dengan menggunakan 2 skenario *handover* yang berbeda
3. Pengukuran dan analisis, mengukur dan menganalisis parameter QoS yang didapat dengan menggunakan perangkat lunak *Wireshark*

1.4 Batasan Masalah

Batasan masalah dari Skripsi ini adalah menganalisa QoS dengan protokol *Mobile IPv6* menggunakan sistem operasi *Linux* untuk nantinya dapat melakukan hal-hal sebagai berikut:

1. Membangun sebuah jaringan lokal *Mobile IPv6*, yang terdiri dari 2 PC router, 2 laptop yang berperan sebagai *Mobile node*, 1 switch, dan 2 access point
2. Mengukur dan menganalisa parameter-parameter untuk menguji jaringan *Mobile IPv6*, yaitu *Delay*, *Transfer rate* dan *Packet loss* dengan proses *horizontal* dan *vertical handover*

1.5 Sistematika Penulisan

Sistematika penulisan pada Skripsi ini adalah sebagai berikut:

BAB 1 Pendahuluan, terdiri dari Latar Belakang, Tujuan, Metodologi Penelitian, Batasan Masalah, dan Sistematika Penulisan.

BAB 2 Internet Protocol dan *Quality of Service*, berisikan pembahasan akan landasan teori tentang *Internet Protocol* dan *Quality of Service*, yang akan dibahas dalam Skripsi ini.

BAB 3 Perancangan dan Implementasi Jaringan *Mobile IPv6* untuk Aplikasi Game Online, berisikan tentang bagaimana jaringan *Mobile IPv6* dirancang, penjelasan tentang komponen dan alat yang digunakan

BAB 4 Pengukuran dan Analisis Parameter QoS Jaringan *Mobile IPv6*

BAB 5 Kesimpulan, berisikan kesimpulan dari Skripsi ini.

BAB II

INTERNET PROTOCOL DAN QUALITY OF SERVICE

2.1 IPv6

IPv6 adalah standar *Internet Protocol* generasi baru yang mulai dibentuk sejak tahun 1994. IPv6 memiliki kombinasi alamat sebanyak 2^{128} , cukup untuk memberikan setiap orang di dunia ini dengan sebuah alamat IP yang unik.

Sebagai protokol baru, IPv6 memiliki banyak fitur, beberapa di antaranya adalah : [5]

1. Format *header* baru

Header IPv6 memiliki format baru yang didesain untuk meminimalisir pemrosesan *header*. Hal ini dicapai dengan memindahkan bagian yang tidak penting dan *optional* pada tambahan *header* yang diletakkan setelah *header* IPv6. *Header* IPv4 dan IPv6 tidak *interoperable*. Sebuah *host* atau router harus mengimplementasikan IPv4 dan IPv6 agar dapat mengenali dan memproses kedua format *header*. Ukuran *header* IPv6 yang baru 2 kali lebih besar dibandingkan dengan *header* IPv4 yang sekarang, meskipun jumlah bit dalam alamat IPv6 4 kali lebih banyak dibandingkan dengan alamat IPv4.

2. Jumlah alamat yang banyak

IPv6 memiliki 128-bit (16-byte) alamat sumber dan tujuan. Meskipun 128 bit dapat dibentuk menjadi $3,4 \times 10^{38}$ kombinasi, alamat IPv6 yang sangat banyak sudah didesain untuk memungkinkan beberapa level *subnetting* dan alokasi alamat, dari *backbone* Internet hingga *subnet* individual dalam sebuah organisasi.

Meskipun tiap alamat digunakan oleh setiap *host*, masih banyak alamat yang tersedia untuk kegunaan di masa depan. Hal ini menyebabkan teknik untuk menghemat alamat, seperti *Network Address Translation* (NAT), tidak lagi dibutuhkan.

3. Konfigurasi alamat *stateless* dan *stateful*

Untuk menyederhanakan konfigurasi *host*, IPv6 mendukung konfigurasi alamat *stateful* (seperti konfigurasi alamat dengan menggunakan *Dynamic Host Configuration Protocol* (DHCP) pada IPv6) dan konfigurasi alamat *stateless* (seperti konfigurasi alamat tanpa server *DHCPv6*). Dengan konfigurasi alamat *stateless*, *host* pada sebuah *link* secara otomatis mengkonfigurasi dirinya dengan alamat-alamat IPv6 untuk *link* tersebut (disebut *link-local addresses*), dengan alamat transisi, dan dengan alamat yang berasal dari prefiks yang disebar oleh router lokal. Meskipun tidak ada router, *host-host* pada link yang sama dapat mengkonfigurasi dirinya dengan *link-local addresses* dan berkomunikasi tanpa konfigurasi manual. *Link-local addresses* dikonfigurasi secara otomatis dalam hitungan detik, dan komunikasi dengan *node* yang bersebelahan pada link yang sama dapat langsung dilakukan. Sebagai pembandingan, beberapa *host* IPv4 yang menggunakan *DHCP* harus menunggu semenit penuh sebelum membuang konfigurasi *DHCP* dan mengkonfigurasi diri dengan alamat IPv4.

4. Mendukung pengiriman dengan prioritas

Bagian baru pada *header* IPv6 menjelaskan bagaimana *traffic* ditangani dan dikenali. *Traffic* diprioritaskan menggunakan *Traffic Class Field*, sama seperti IPv4. Bagian *Flow Field* pada *header* IPv6 memungkinkan router-router untuk mengidentifikasi dan menyediakan penanganan spesial untuk paket-paket yang dimiliki oleh sebuah *flow* (serangkaian paket antara sebuah sumber dan tujuan).

Pengalamatan IPv6

Tipe alamat yang terdapat pada IPv6 :

1. Alamat *Unicast*, sebuah alamat *Unicast* bertindak sebagai pengenal untuk sebuah antarmuka tunggal. Sebuah paket IPv6 yang dikirimkan ke sebuah alamat *Unicast* disampaikan ke antarmuka yang diidentifikasi oleh alamat tersebut.

2. Alamat *Multicast*, sebuah alamat *Multicast* bertindak sebagai pengenalan untuk sebuah grup atau set antarmuka yang mungkin dimiliki oleh *node-node* yang berbeda. Sebuah paket IPv6 yang dikirimkan ke sebuah alamat *Multicast* disampaikan ke beberapa antarmuka.
3. Alamat *Anycast*, alamat *Anycast* bertindak sebagai pengenalan untuk sebuah set antarmuka yang mungkin dimiliki oleh *node-node* yang berbeda. Sebuah paket IPv6 yang dikirimkan ke sebuah alamat *Anycast* disampaikan ke salah satu dari antarmuka yang diidentifikasi oleh alamat tersebut.

Dengan jumlah bit yang berbeda, tentu saja IPv6 memiliki cara pengalamatan yang berbeda dengan IPv4. Ukuran alamat IPv6 adalah 128 bit, yang direpresentasikan dengan 8 bagian, masing-masing bagian terdiri dari 16 bit biner yang dijadikan 4 bilangan hexadesimal. Contoh :

2001:cdba:0000:0000:0000:0000:3257:9652

Jika ada 4 buah angka 0 seperti contoh di atas, maka dapat diperpendek menjadi 1 buah angka 0 saja, sehingga menjadi 2001:cdba:0:0:0:0:3257:9652. Apabila ada banyak angka 0, dapat disatukan menjadi 2 buah tanda titik dua, contoh : 2001:cdba:0:0:0:0:3257:9652 menjadi 2001:cdba::3257:9652. URL untuk contoh alamat di atas adalah :

[http://\[2001:cdba:0000:0000:0000:0000:3257:9652\]/](http://[2001:cdba:0000:0000:0000:0000:3257:9652]/)

Pengalamatan jaringan pasti memiliki notasi *network*, yang pada IPv6 menggunakan notasi *Classless InterDomain Routing (CIDR)*. Sebuah jaringan atau *subnet* yang menggunakan protokol IPv6 dikenali sebagai grup alamat IPv6 yang berdekatan yang ukurannya harus merupakan perpangkatan 2. Bit awal pada alamat IPv6 membentuk prefiks *network*. Ukuran bit pada prefiks *network* dipisahkan dengan sebuah /. Sebagai contoh :

2001:cdba:9abc:5678::/64 menandakan *network address* 2001:cdba:9abc:5678.

Jaringan ini terdiri dari alamat yang dimulai dari 2001:cdba:9abc:5678:: sampai 2001:cdba:9abc:5678:ffff:ffff:ffff:ffff. Dengan cara ini, 1 *host* dapat dijadikan sebagai sebuah jaringan dengan prefiks 128-bit. Hal ini memungkinkan sebuah jaringan untuk memiliki hanya satu *host* atau lebih.

Alamat spesial pada IPv6

1. $::/96$, prefiks 0 menandakan alamat-alamat yang cocok dengan protokol IPv4
2. $::/128$, sebuah alamat IPv6 dengan angka 0 semua menandakan sebuah alamat yang digunakan dalam sebuah perangkat lunak
3. $::1/128$, alamat ini disebut alamat *loopback* dan digunakan untuk menunjukkan *localhost*. Aplikasi yang mengirimkan paket ke alamat ini akan mendapatkan paket tersebut setelah dikembalikan oleh IPv6 *stack*.
4. $2001:db8::/32$, ini merupakan prefiks dokumentasi pada IPv6. Semua contoh alamat pada IPv6 seharusnya secara ideal menggunakan prefiks ini untuk menandakan bahwa alamat ini adalah sebuah contoh.
5. $Fec0::/10$, ini merupakan prefiks *site-local* pada IPv6. Prefiks alamat ini menunjukkan bahwa alamat ini sah hanya dalam organisasi lokal.
6. $Fc00::/7$, disebut *Unique Local Address* (ULA). Alamat-alamat ini digunakan dalam IPv6 untuk menggantikan alamat *site-local*. Alamat ini juga menyediakan 40-bit angka *pseudorandom* yang mengurangi resiko konflik alamat.
7. $Ff00::/8$, prefiks ini disediakan oleh IPv6 untuk menandakan alamat-alamat *multicast*. Alamat manapun yang mengandung prefiks ini secara otomatis dibaca sebagai alamat *multicast*.
8. $Fe80::/10$, prefiks *link-local* pada IPv6. Alamat ini menunjukkan bahwa alamat ini hanya sah dalam *physical link* lokal.

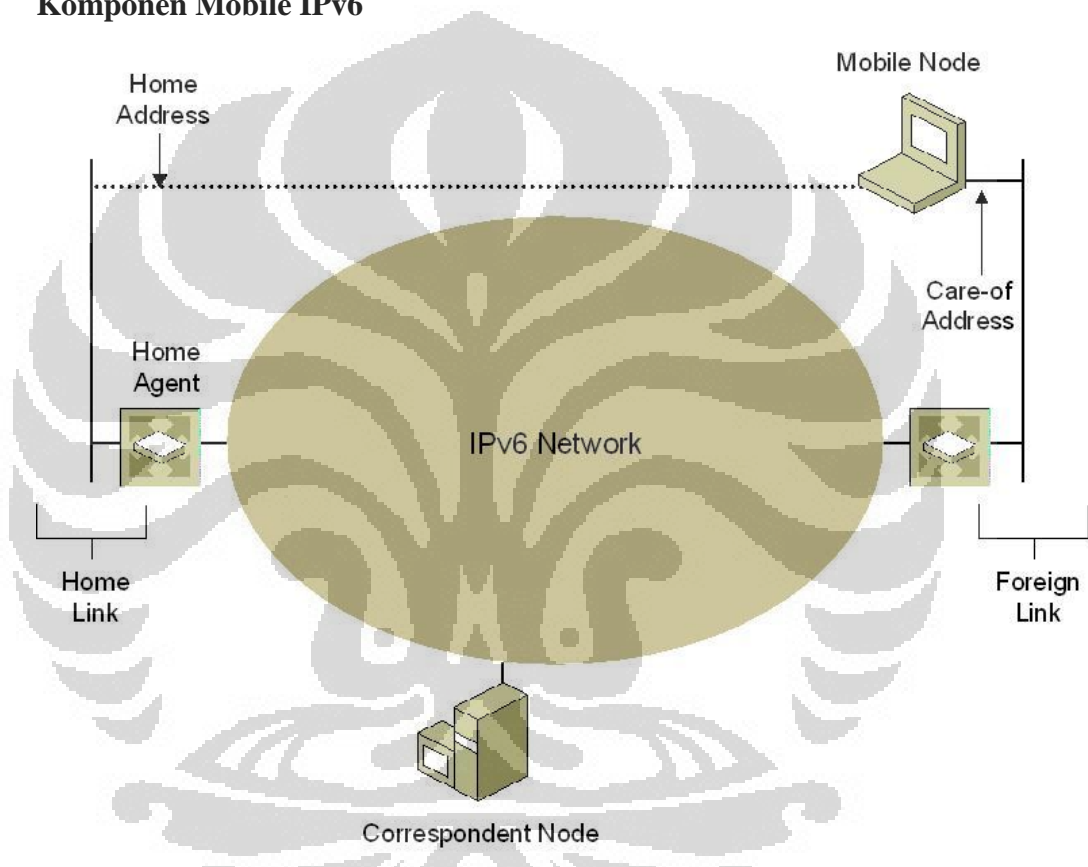
2.2 Mobile IPv6

Mobile IPv6 memungkinkan *node* IPv6 menjadi *mobile* – berpindah lokasi dalam sebuah jaringan IPv6 – dan masih mempertahankan alamatnya. Ketika sebuah *node* IPv6 berpindah lokasi, *link* nya juga mungkin berubah. Ketika sebuah *node* IPv6 mengubah linknya, alamat IPv6 nya juga mungkin berubah untuk mempertahankan konektivitasnya. Ada beberapa mekanisme yang memungkinkan pergantian alamat ketika berpindah ke *link* yang berbeda, seperti konfigurasi alamat otomatis *stateful* dan *stateless* untuk IPv6. Ketika alamat berubah, koneksi *mobile node* yang menggunakan alamat didapat dari

link yang terhubung sebelumnya tidak bisa dipertahankan dan terpaksa dihentikan.

Manfaat utama dari *Mobile IPv6* adalah meskipun *mobile node* mengubah lokasi dan alamat, koneksi yang ada tetap dipertahankan. Untuk mencapai ini, koneksi *mobile node* dibuat dengan alamat spesifik yang ditetapkan untuk *mobile node*, dan melalui alamat inilah *mobile node* selalu terhubung.

Komponen Mobile IPv6



Gambar 2.1 Contoh jaringan *Mobile IPv6*[1]

Komponen-komponen *Mobile IPv6* dapat dilihat pada Gambar 2.1

1. *Home Link*

Link yang mengandung prefiks *home subnet*, darimana *mobile node* mendapatkan *home address*nya. *Home agent* berada pada *home link*.

2. *Home Address*

Sebuah alamat yang diberikan kepada *mobile node* ketika terhubung dengan *home link* yang membuat *mobile node* selalu dapat dijangkau,

tanpa memperhatikan lokasi jaringan IPv6. Jika *mobile node* terhubung dengan *home link*, proses *Mobile IPv6* tidak digunakan dan komunikasi terjadi secara normal. Jika *mobile node* tidak terhubung dengan *home link*, paket yang ditujukan ke *home address* dari *mobile node* diambil oleh *home agent* dan ditujukan ke jaringan IPv6 tempat *mobile node* berada. Karena *mobile node* selalu menggunakan *home address*, secara logika *mobile node* selalu terhubung dengan *home link*.

3. *Home Agent*

Sebuah router pada *home link* yang menjaga pendaftaran *mobile node* yang terpisah dari *home* dan alamat yang sedang digunakannya. Jika *mobile node* sedang terpisah dari *home*, *home agent* mendaftarkan alamatnya, yang akan mengirimkan data yang ditujukan ke *home address mobile node* ke alamat sekarang dalam jaringan IPv6 dan meneruskan data yang dikirimkan oleh *mobile node*. *Home agent* tidak harus berupa router, bisa berupa *node* yang tidak melakukan apa-apa saat *mobile node* berada di *home*.

4. *Mobile node*

Sebuah *node* IPv6 yang dapat berubah *link* dan alamat, dan selalu dapat diakses menggunakan *home address*. Sebuah *mobile node* memiliki *home address* dan *global address* untuk *link* yang sedang terhubung dengannya (disebut juga *care-of-address*), dan memberitahukan *home address/care-of-addressnya* kepada *home agent* dan *node Mobile IPv6* yang sedang berkomunikasi dengannya.

5. *Foreign Link*

Link selain *home link* milik *mobile node*.

6. *Care-of-Address*

Sebuah alamat yang digunakan oleh *mobile node* ketika terhubung dengan sebuah *foreign link*. Untuk konfigurasi alamat *stateless*, *care-of-address*

adalah kombinasi dari prefiks *foreign subnet* dan *ID* antarmuka yang ditentukan oleh *mobile node*. Sebuah *mobile node* dapat memiliki beberapa *care-of-address*, tetapi hanya satu *care-of-address* yang dapat didaftarkan sebagai *care-of-address* utama kepada *home agent*. Hubungan antara *home address* dan *care-of-address* untuk sebuah *mobile node* disebut *binding*. *Correspondent node* dan *home agent* menyimpan informasi *binding* di dalam *binding cache*.

7. *Correspondent node*

Sebuah *node IPv6* yang berkomunikasi dengan *mobile node*. Sebuah *correspondent node* tidak harus mendukung *Mobile IPv6*. Jika *correspondent node* mendukung *Mobile IPv6*, bisa jadi merupakan *mobile node* yang sedang terpisah dari *home*.

Selain beberapa komponen yang sudah disebutkan di atas, ada beberapa istilah penting yang terdapat dalam teknologi *Mobile IPv6*, yang dapat dilihat pada Tabel 2.1

Tabel 2.0.1 Istilah dalam teknologi *Mobile IPv6*

Istilah	Penjelasan
<i>Binding</i>	Ikatan atau hubungan antara <i>home agent</i> dan <i>mobile node</i>
<i>Foreign subnet prefix</i>	Setiap <i>prefix subnet IP</i> selain <i>mobile node prefix</i>
<i>Home Registration</i>	Registrasi dari <i>mobile node</i> dengan <i>primary care of address</i>
<i>Home Subnet Prefix</i>	<i>Subnet IP prefix</i> yang menunjuk kepada <i>Home Address</i> dari <i>mobile node</i>
<i>Movement</i>	Perubahan <i>point of attachment</i> dari <i>mobile node</i> ke internet sepanjang tidak terkoneksi secara langsung dengan <i>home addressnya</i> , jika tidak berada pada <i>home linknya</i> dapat dikatakan <i>mobile node</i> berada pada keadaan “ <i>away from home</i> ”

Mobile IPv6 sangat membutuhkan adanya pertukaran informasi antar *link* (*home link* dan *foreign link*). Semua pesan yang digunakan dalam *Mobile IPv6* didefinisikan sebagai *IPv6 Destination Option*.

Dibawah ini adalah beberapa opsi yang digunakan dalam *Mobile IPv6* untuk membawa informasi tambahan yang nantinya dibutuhkan untuk diperiksa hanya oleh sebuah *node* tujuan dari *node* :

1. *Binding update*

Opsi ini digunakan oleh *mobile node* untuk memberikan informasi tentang *care of address*nya kepada *home agent* atau *correspondent node* yang lain. Semua paket yang termasuk dalam *binding update* harus mengandung *Authentication Header (AH)* atau *Encapsulating Security Payload (ESP) header*.

2. *Binding Acknowledgement*

Opsi ini digunakan untuk memberitahukan adanya *binding update* yang diterima dari *mobile node*, opsi ini juga harus mempunyai *header AH* atau *ESP*.

3. *Binding Request*

Opsi ini digunakan oleh tiap *node* untuk meminta *mobile node* mengirimkan sebuah *binding update*.

Struktur data Mobile IPv6

Mobile IPv6 mempunyai konsep struktur data seperti dibawah ini :

1. *Binding Cache*

Setiap *IPv6 node* mempunyai sebuah *binding cache* yang berfungsi untuk menyimpan *binding* dari tiap *node*. Jika sebuah *node* menerima sebuah *binding update*, *node* tersebut akan menambahkan *binding update* tersebut ke *binding cache*. Setiap mengirim paket, *binding cache* akan

memeriksa tiap *entry* yang ada. Jika ada *entry* pada *binding cache*, maka paket akan dikirimkan ke *care of address* ke *correspondent node* melalui *routing header*.

2. *Binding Update list*

Setiap *mobile node* mempunyai *binding update list* yang berfungsi untuk mempunyai informasi tentang tiap-tiap *binding update* yang dikirimkan oleh *mobile node* yang *lifetime* nya belum habis. *Binding update list* berisi semua *binding update* yang dikirimkan ke semua *corespondent node* (*mobile* atau statis) dan *home agent*nya

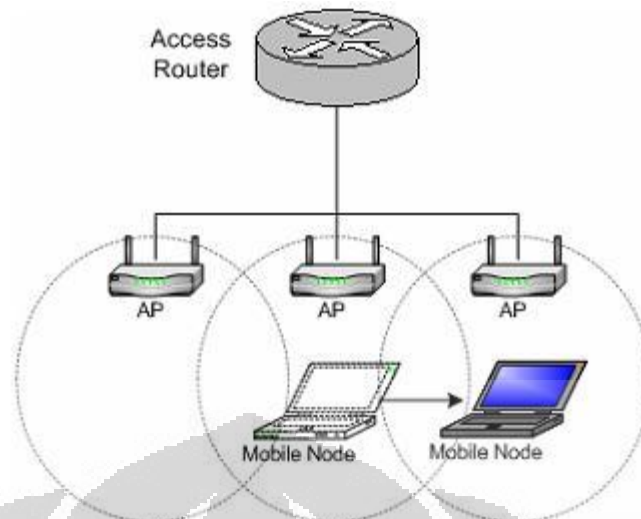
3. *Home Agent list*

Untuk tiap *home link* sebuah *node* dikatakan sebagai *home agent* jika *node* tersebut menghasilkan sebuah daftar yang mengandung informasi tentang semua *home agent* dalam *link* yang ada. Informasi yang berada dalam data didapatkan dari *unsolicitated multicast router advertisement*, yang dikirimkan oleh semua *home agent* dan semua *node* yang melakukan *setting* bit-nya sebagai *home agent*. Informasi tentang semua *home agent* yang lain didapatkan dari mekanisme *home agent discovery*.

Dalam jaringan *Mobile IPv6*, ada istilah yang disebut *handover*. *Handover* adalah perpindahan *mobile node* dari satu titik jaringan ke titik yang lain. Secara umum ada 2 jenis proses *handover*, yaitu : [2]

1. *Horizontal Handover*

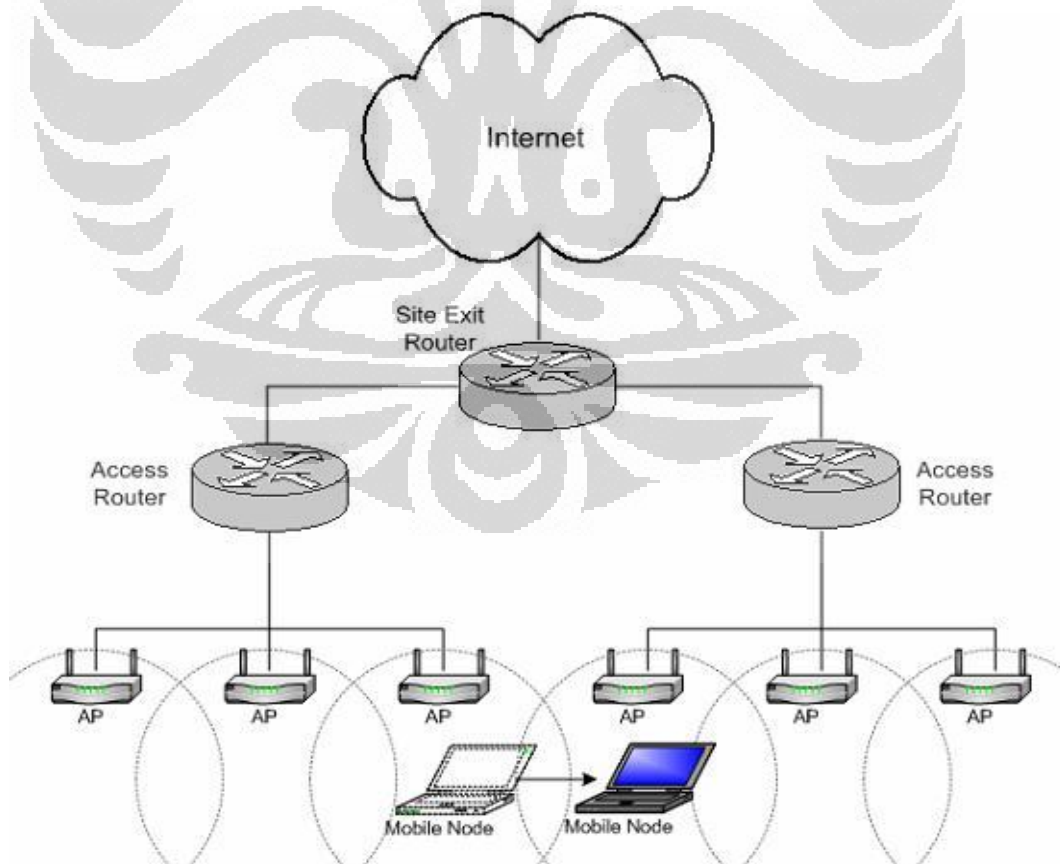
Horizontal handover adalah proses perpindahan *mobile node* yang hanya memengaruhi *link layer*, tanpa mengubah IP *mobile node* tersebut. Gambar *horizontal handover* dapat dilihat pada Gambar 2.2



Gambar 2.2 *Horizontal Handover* [2]

2. *Vertical Handover*

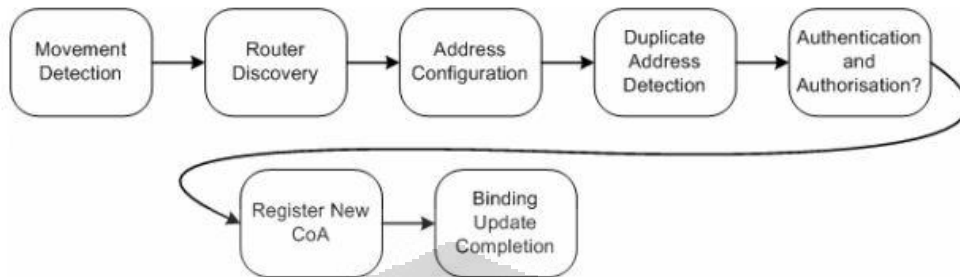
Vertical handover adalah proses perpindahan *mobile node* dimana *mobile node* berpindah dari jaringan *access router* yang satu ke jaringan *access router* yang lain. Gambar *vertical handover* dapat dilihat pada Gambar 2.3



Gambar 2.3 *Vertical Handover* [2]

Cara Kerja Jaringan

Jaringan *Mobile IPv6* memiliki cara kerja seperti Gambar 2.4



Gambar 2.0.4 Cara kerja jaringan *Mobile IPv6* [2]

1. *Movement Detection*

Dalam *Mobile IPv6*, ada pendeteksian jika *Mobile node* sudah bergerak atau berpindah jaringan. Ada 2 cara untuk mengetahui jika *Mobile node* sudah bergerak atau belum :

- *Access Router* yang sekarang tidak dapat mencapai *Mobile node*
- Sebuah *access router* yang baru dan berbeda sudah tersedia

Agar *access router* mengetahui masih terhubung atau tidak dengan *Mobile node*, maka dilakukan *Neighbour Unreachability Detection* (NUD). NUD akan menentukan *Mobile node* masih terhubung dengan mengirimkan paket dan jika ada konfirmasi bahwa paket tersebut sudah diterima.

2. *Router Discovery*

Router discovery terjadi melalui penerimaan *router advertisement* yang dikirimkan oleh *access router* yang baru. Setelah *mobile node* menentukan bahwa *access router* yang sekarang tidak lagi bisa dihubungi, maka *mobile node* akan mengirimkan *router solicitation*, lalu akan menerima *router advertisement* dari *access router* yang baru.

3. *Address Configuration*

Mobile node harus mengkonfigurasi dirinya dengan alamat *IPv6* untuk digunakan pada jaringan yang baru, yang disebut *Mobile node's New Care-of-Address* (NCoA). Konfigurasi alamat bisa terjadi dalam kondisi

stateful atau *stateless*. Jika tidak ingin mengkonfigurasi alamat secara otomatis, tersedia pilihan untuk mengkonfigurasi alamat secara manual.

4. *Duplicate Address Detection*

Saat berpindah jaringan, *mobile node* harus memastikan bahwa alamatnya tidak dipakai oleh *host/mobile node* lain. Bila diperlukan adanya pengecekan, yang disebut *Duplicate Address Detection* (DAD). DAD memungkinkan *mobile node* untuk mengetahui alamatnya sudah dipakai dengan cara memberikan *advertisement* dari *node* di sebelahnya. Hal ini sangat kecil kemungkinannya mengingat jumlah alamat yang disediakan oleh *IPv6*.

5. *Register New Care of Address (CoA)*

Setelah pindah ke jaringan baru dan mendapat CoA baru, maka *mobile node* akan memberitahu *home agent* dengan mengirimkan *binding update*. *Home agent* akan membalasnya dengan *binding acknowledgement*, barulah *mobile node* bisa mengirimkan paket lagi. Selama proses *binding*, semua paket yang ditujukan ke *mobile node* didrop (dibuang), dan *mobile node* tidak dapat mengirimkan paket ke *correspondent node* manapun.

6. *Binding Update Completion*

Tahap ini merupakan tahap *handover* terakhir, dengan *mobile node* mengirimkan *binding update* ke *correspondent node* dan dibalas dengan test dari *correspondent node* agar dijamin bahwa *binding update* ini berasal dari *mobile node*, bukan dari pihak yang tidak diketahui. *Correspondent node* mengirimkan test ini melalui *home agent* yang akan meneruskannya ke *mobile node*. Jika *mobile node* menjawab test ini dengan benar, maka proses *handover* pun selesai.

2.3 Quality of Service

Quality of Service (QoS) merupakan pengukuran kemampuan jaringan untuk menyediakan servis pada suatu *traffic* tertentu. Salah satu contoh jaringan yang dimaksud adalah jaringan IP.

QoS memiliki beberapa parameter yang dapat diukur, di antaranya :

1. *Transfer rate*

Transfer rate adalah banyaknya data yang dapat dikirimkan dalam satu detik, dihitung dengan satuan paket per detik (*packet/second*)

2. *Packet loss*

Packet loss adalah perbandingan seluruh paket IP yang hilang dengan seluruh paket IP yang dikirimkan antara pada sumber dan tujuan. Salah satu penyebab *packet loss* adalah antrian yang melebihi kapasitas *buffer* pada setiap *node*. Beberapa penyebab terjadinya *packet loss* yaitu:

- *Congestion*, disebabkan terjadinya antrian yang berlebihan dalam jaringan
- *Node* yang bekerja melebihi kapasitas *buffer*
- *Memory* yang terbatas pada node
- *Policing* atau kontrol terhadap jaringan untuk memastikan bahwa jumlah *traffic* yang mengalir sesuai dengan besarnya bandwidth. Jika besarnya *traffic* yang mengalir didalam jaringan melebihi dari kapasitas *bandwidth* yang ada maka *policing control* akan membuang kelebihan *traffic* yang ada.

3. *Delay*

Delay adalah total waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya. *Delay* dalam jaringan dapat digolongkan sebagai berikut :

- Paketisasi *delay*

Delay yang disebabkan oleh waktu yang diperlukan untuk proses pembentukan paket IP dari user, yang harus melalui proses enkapsulasi 7 lapisan OSI atau 4 lapisan TCP/IP. *Delay* ini hanya terjadi 1 kali di sumber informasi.

- *Queueing delay*

Delay ini disebabkan oleh waktu proses yang diperlukan oleh router di dalam menangani transmisi paket di sepanjang jaringan. Umumnya *delay* ini sangat kecil, kurang lebih sekitar 100ms.

- *Delay propagasi*

Proses perjalanan informasi selama media transmisi, misalnya *coaxial* atau tembaga, menyebabkan *delay* yang disebut *delay propagasi*.

Menurut ITU G.114, ada 3 *band* standar *delay* yang direkomendasikan, dapat dilihat pada Gambar 2.5

Tabel 2.2 Rekomendasi ITU-T G.114 untuk *delay*[6]

Range in Milisecon	Description
0 – 150 msec	Acceptable for most user application
150 – 400 msec	Acceptable provided that administrators are aware of the transmission time and it's impact on transmission quality of user application
> 400 msec	Unacceptable for general network planning purpose, it is recognized that in some exceptional cases this limit will be exceeded.

2.4 Game Online

Game online merupakan permainan yang membutuhkan koneksi internet untuk dimainkan. Umumnya permainan ini dimainkan oleh lebih dari 1 orang secara bersamaan sehingga *traffic* pada jaringan *game online* umumnya cukup padat, meskipun tetap bergantung kepada banyaknya jumlah pemain.

Di seluruh dunia, jumlah pemain *game online* sangat banyak, yang menimbulkan persaingan antar pemain. Persaingan itu merupakan salah satu faktor mengapa *game online* sangat digemari.

Game online yang ada saat ini sangat banyak, salah satunya adalah tetris. Tetris adalah permainan teka-teki bangunan yang pertama kalinya didesain oleh Alexey Pajitnov pada bulan Juni 1985, saat itu, ia masih bekerja di Pusat Komputer Dorodnicyn di Akademi Sains Uni Soviet di Moskow, nama tetris sendiri diambil dari bahasa numerik yunani, yaitu tetra yang artinya empat. Empat sendiri adalah jumlah susunan balok yang bisa diubah-ubah ke dalam berbagai bentuk untuk selanjutnya disusun oleh pemain untuk dapat saling terkunci.

Permainan tetris ini sukses setelah dirilis oleh gameboy pada tahun 1989, setelah itu, permainan ini mulai dirilis ulang oleh berbagai perusahaan game dunia seperti nintendo, playstation, office game, dan perusahaan-perusahaan game ternama lainnya. walaupun sudah berusia sangat lama, tapi permainan ini tetap menjadi primadona di mata gamer-gamer dunia, menurut mereka, game ini adalah game yang banyak menguras otak dan mengasah cara pandang ruang otak seseorang. [7]

BAB III

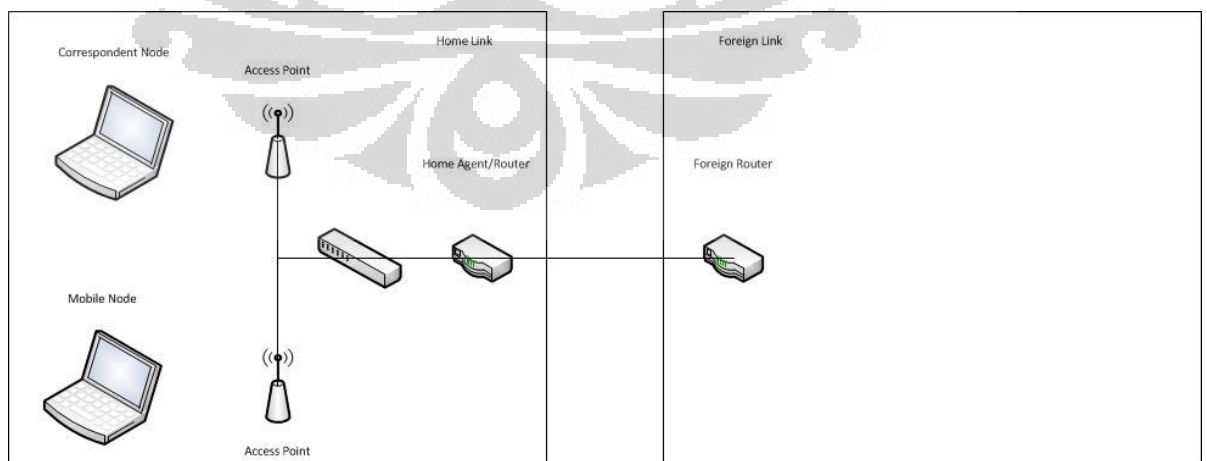
PERANCANGAN DAN IMPLEMENTASI JARINGAN MOBILE IPV6 UNTUK APLIKASI GAME ONLINE

3.1 Topologi Jaringan

Pada bab ini, akan dibahas perancangan jaringan untuk game online pada *Mobile IPv6*. Perancangan jaringan dilakukan dengan menggunakan sistem operasi yang mendukung *Mobile IPv6*. Untuk pengukuran parameter Qos, terdapat 2 skenario, yaitu *Horizontal Handover* dan *Vertical Handover*.

3.1.1 Horizontal Handover

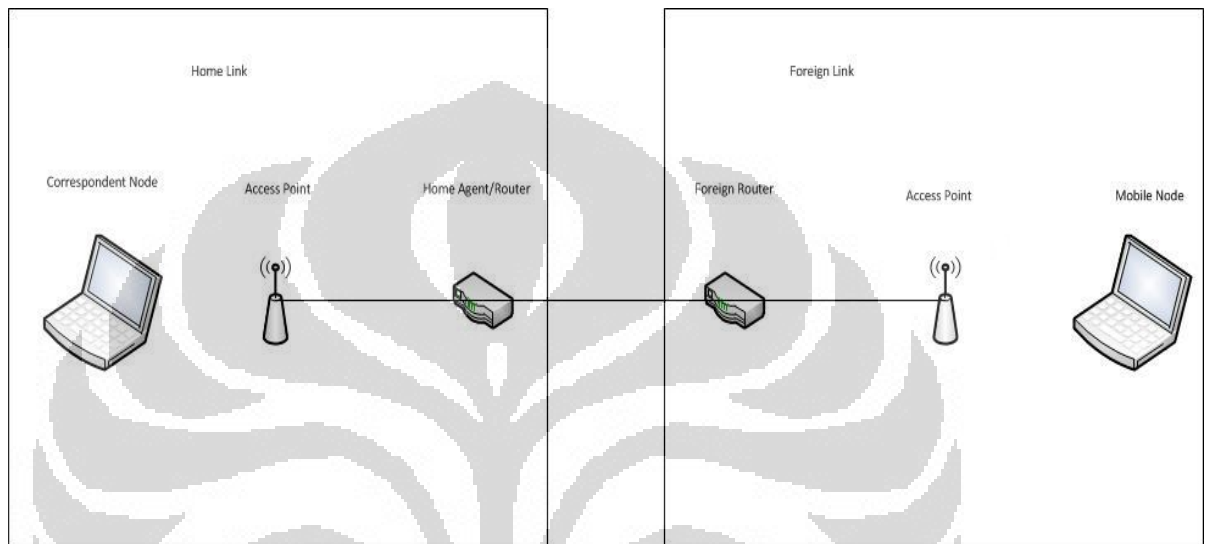
Rancangan jaringan horizontal yang akan diimplementasikan pada *Mobile IPv6* terdiri dari 1 switch, 2 laptop, 2 *access point*, dan 2 PC router. Setiap perangkat memiliki fungsi dan konfigurasi yang berbeda. 1 laptop berfungsi sebagai *mobile node*, 1 laptop sebagai *correspondent node*, 1 PC router sebagai *home agent*, 1 PC router sebagai *foreign router*, 1 switch sebagai penghubung 2 *access point* untuk *home link*, dan 2 *access point* untuk menghubungkan laptop dengan PC router. Sistem operasi yang akan digunakan adalah *Linux Ubuntu 8.04*. Pada jaringan horizontal, kedua *mobile node* berada pada jaringan yang sama, namun berbeda *access point*. Topologi jaringan yang akan dirancang dapat dilihat pada Gambar 3.1



Gambar 3.1 Topologi jaringan untuk skenario *Horizontal Handover* pada *Mobile IPv6*

3.1.2 Vertical Handover

Jaringan vertikal memakai perangkat yang sama, hanya tidak memakai switch karena tiap PC router hanya terhubung dengan 1 *access point* dengan jaringan horizontal, tetapi posisi *mobile node* berada pada jaringan yang berbeda (*foreign link*). Topologi jaringan vertikal dapat dilihat pada Gambar 3.2



Gambar 3.2 Topologi jaringan untuk skenario *Vertical Handover* pada *Mobile IPv6*

3.2 Spesifikasi Sistem

3.2.1 Spesifikasi Perangkat Keras

Perangkat keras yang akan digunakan dalam perancangan jaringan *Mobile IPv6* ini adalah :

1. *Home Router*

Untuk *Home Router*, akan digunakan sebuah PC Router yang berfungsi ganda sebagai *Home Agent* dan *Home Router*

Prosesor : Intel(R) Pentium(R) dual cpu

Memori : 1024 MB

Harddisk : 20 GB

2. *Foreign Router*

Untuk *Foreign Router*, akan digunakan sebuah PC Router yang mendukung *Mobile IPv6*

Prosesor : Intel(R) Pentium(R) 4 CPU

Memori : 512 MB

Harddisk : 20 GB

3. *Mobile node*

Untuk *mobile node*, akan digunakan laptop yang akan berpindah *access point* dalam *home link* (untuk topologi horizontal) dan berpindah jaringan ke *foreign link* (untuk topologi vertikal)

Prosesor : Intel(R) Core(TM)2 Duo CPU

Memori : 4GB

Harddisk : 4GB

4. *Correspondent node*

Untuk *Correspondent node*, akan digunakan sebuah laptop yang berhubungan dengan *mobile node*

Prosesor : AMD E350

Memori : 6 GB

Harddisk : 320 GB

5. *Access Point*

Access point akan digunakan sebagai penghubung router dengan *mobile node* dan *correspondent node*.

Tipe : TP Link-WA-601G dan Linksys WAP-54G

6. Switch

Switch digunakan untuk menghubungkan *home agent* dengan 2 buah *access point*.

Tipe : TL-SF1008D

3.2.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang akan digunakan untuk perancangan jaringan *Mobile IPv6* ini adalah :

1. *Traffic Generator*

Traffic generator adalah sebuah perangkat lunak yang berfungsi untuk memberikan *traffic* pada jaringan sehingga lebih menyerupai jaringan internet yang sebenarnya. Perangkat lunak ini akan diinstal pada sebuah node dalam jaringan, yang nantinya akan terus mengirimkan *traffic*. Nama perangkat lunak yang digunakan adalah *Multi-Generator* (MGEN).

2. Sistem operasi *Linux Ubuntu 8.04*

Dalam perancangan jaringan *Mobile IPv6* ini, akan digunakan sistem operasi *Linux Ubuntu 8.04*, karena mendukung *Mobile IPv6* dan mudah digunakan.

3. *Gtetrinet* dan *Tetrinetx*

Gtetrinet adalah sebuah perangkat lunak permainan tetris yang dapat dijalankan pada sistem operasi *Linux* dan mendukung IPv6. *Tetrinetx* adalah perangkat lunak yang menyediakan sebuah server untuk perangkat lunak *Gtetrinet*.

4. Wireshark

Wireshark adalah sebuah perangkat lunak *freeware* untuk memantau dan menganalisa protokol jaringan.

3.4 Cara Kerja Pengujian Jaringan

Rancangan jaringan yang dibahas pada Bab III ini akan diuji dengan cara berikut :

1. *Vertical Handover*

Pada skenario *vertical handover*, *correspondent node* dan *mobile node* terhubung dengan *access point* pada *home link* dan memainkan *Gtetrinet*. Server *Gtetrinet* terletak pada *correspondent node*. Lalu saat bermain, *mobile node* digerakkan sehingga keluar dari jangkauan *access point* pada *home link* dan memasuki jangkauan *access point* pada *foreign link*. Pada *mobile node* digunakan perangkat lunak *Wireshark* untuk mengukur parameter QoS. Parameter QoS yang diukur pada percobaan ini adalah *transfer rate*, *delay*, dan *packet loss*. *Mobile node* dan *correspondent node* akan bermain selama 1 menit sebanyak 10 kali dan diambil data QoSnya.

2. *Horizontal Handover*

Pada skenario *horizontal handover*, proses pengukuran dilakukan sama seperti skenario *vertical handover*, *correspondent node* dan *mobile node* terhubung dengan *access point* yang sama pada *home link* dan memainkan *Gtetrinet*. Server *Gtetrinet* terletak pada *correspondent node*. Lalu saat bermain, *mobile node* digerakkan sehingga keluar dari jangkauan *access point* 1 dan memasuki jangkauan *access point* 2 pada *home link*. Pada *mobile node* digunakan perangkat lunak *Wireshark* untuk mengukur parameter QoS.

3.5 Pembuatan Sistem

Tahap-tahap pembuatan sistem yang mendukung *Mobile IPv6* adalah : [8]

1. Membuka file *sources.list* pada direktori */etc/apt/* dengan perintah :

```
#sudo gedit /etc/apt/sources.list
```

2. Mengedit file *sources.list* untuk menambahkan sumber perangkat lunak dengan mengetik :

```
deb http://software.nautilus6.org/packages/ubuntu hardy/
```

```
deb-src http://software.nautilus6.org/packages/ubuntu hardy/
```

3. Mengupdate packages dengan perintah :

```
#sudo apt-get update
```

4. Menginstall perangkat lunak dari nautilus :

```
#sudo apt-get install nautilus6-keyring
```
5. Klik System > Administration > Synaptic Package Manager
6. Search mipv6
7. Klik kanan pada mipv6-daemon-umip lalu klik Mark for Installation
8. Klik Apply, setelah selesai diinstall maka restart PC atau laptop
9. Saat booting dan GRUB loading, tekan esc lalu pilih Ubuntu 8.04 LTS,
kernel 2.6.22-14-mip6

Setelah selesai menginstall kernel mip6, maka perlu diinstall radvd untuk *home router* dan *foreign router*, yang berguna untuk mentransmisikan *router advertisement* secara periodik. *Router advertisement* berisi prefix alamat IPv6 sehingga *host* secara otomatis akan mendapatkan alamat jika terhubung ke jaringan *Mobile IPv6*.

Cara install radvd :

1. Klik System > Administration > Synaptic Package Manager
2. Search radvd
3. Klik kanan pada radvd lalu klik Mark for Installation
4. Klik Apply

3.5.1 Konfigurasi Alamat IPv6

Setelah semua *node* jaringan terinstall sistem operasi yang mendukung *Mobile IPv6*, setiap *node* harus disetting dengan sebuah alamat IPv6.

Alamat IPv6 dapat ditambahkan pada terminal dengan perintah :

```
#sudo ifconfig (interface) inet6 add (alamat)
```

Untuk menghapus alamat yang sudah diberikan, gunakan perintah :

```
#sudo ifconfig (interface) inet6 del (alamat)
```

3.5.2 Konfigurasi Node

- Konfigurasi *Home Router*

Router ini memiliki 2 *interface*. *Interface* untuk *home link* memiliki alamat 2001:db8:3c4d:5a::2/64 dan *interface* ke *foreign router* memiliki alamat 2001:db8:3c4d:5b::2/64. Konfigurasi *node Mobile IPv6* disimpan dalam file *mip6d.conf* pada direktori */usr/local/etc*. Sebelum mengedit file konfigurasi, pada terminal *home router* harus diketik perintah berikut :

1. Konfigurasi alamat eth1

```
#ifconfig eth1 inet6 add 2001:db8:3c4d:5a::2/64
```

2. Konfigurasi alamat eth2

```
# ifconfig eth2 inet6 add 2001:db8:3c4d:5b::2/64
```

3. Enable beberapa fungsi *Home Router*

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

```
# echo 0 > /proc/sys/net/ipv6/conf/all/autoconf
```

```
# echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra
```

```
# echo 0 > /proc/sys/net/ipv6/conf/all/accept_redirects
```

4. Menambahkan *static routing*

```
# ip route add 2001:db8:3c4d:5c::/64 via 2001:db8:3c4d:5b::3
```

Selanjutnya mengedit file *mip6d.conf*, isinya adalah sebagai berikut :

```
NodeConfig HA;
```

```
DebugLevel 10;
```

```
Interface "eth1";
```

```
UseMnHaIPsec disabled;
```

Lalu untuk *radvd*, dibuat file konfigurasi dengan nama *radvd.conf* pada direktori */etc*. Isinya adalah sebagai berikut :

```

interface eth1
{ AdvSendAdvert on;
MinRtrAdvInterval 3;
MaxRtrAdvInterval 7;
AdvIntervalOpt off;
AdvHomeAgentFlag on;
HomeAgentLifetime 10000;
HomeAgentPreference 20;
AdvHomeAgentInfo on;
prefix 2001:db8:3c4d:5a::2/64
{ AdvRouterAddr on;
AdvOnLink on;
AdvAutonomous on;
AdvPreferredLifetime 10000;
AdvValidLifetime 12000;
};
};

```

- Konfigurasi *Foreign Router*

Router ini memiliki 2 *interface*, yaitu *interface* yang terhubung dengan *foreign link* dengan alamat 2001:db8:3c4d:5c::2/64 dan *interface* ke *home router* dengan alamat 2001:db8:3c4d:5b::3/64. *Foreign router* tidak memerlukan file konfigurasi *mip6d.conf* karena *foreign router* tidak menjaga koneksi dengan *mobile node*. Pada terminal *foreign router*, perlu diketikkan perintah sebagai berikut :

1. Konfigurasi alamat eth2

```
# ifconfig eth2 inet6 add 2001:db8:3c4d:5b::3/64
```

2. Konfigurasi alamat eth3

```
# ifconfig eth3 inet6 add 2001:db8:3c4d:5c::2/64
```

3. Enable beberapa fungsi *foreign router*

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

```
# echo 0 > /proc/sys/net/ipv6/conf/all/autoconf
```

```
# echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra
```

```
# echo 0 > /proc/sys/net/ipv6/conf/all/accept_redirects
```

4. Menambahkan *static routing*

```
# ip route add 2001:db8:3c4d:5a::/64 via 2001:db8:3c4d:5a::2
```

Foreign router hanya memerlukan file `radvd.conf` yang isinya adalah sebagai berikut :

```
interface eth3
{
  AdvSendAdvert on;
  AdvIntervalOpt off;
  MinRtrAdvInterval 3;
  MaxRtrAdvInterval 7;
  AdvHomeAgentFlag off;
  prefix 2001:db8:3c4d:5c::/64
  {
    AdvOnLink on;
    AdvAutonomous on;
    AdvRouterAddr on;
  };
}
```

- Konfigurasi *Mobile node*

Mobile node hanya memiliki 1 *interface*, yaitu *wireless LAN (wlan)*. *Mobile node* memiliki file konfigurasi `mip6d.conf`, tetapi sebelum itu ada beberapa perintah yang harus diketikkan pada terminal :

1. Konfigurasi alamat *wlan0*

```
# ifconfig wlan0 inet6 add 2001:db8:3c4d:5a::3/64
```

2. *Enable* beberapa fungsi *mobile node*

```
# echo 0 > /proc/sys/net/ipv6/conf/wlan0/forwarding
```

```
# echo 1 > /proc/sys/net/ipv6/conf/wlan0/autoconf
```

```
# echo 1 > /proc/sys/net/ipv6/conf/wlan0/accept_ra
```

```
# echo 1 > /proc/sys/net/ipv6/conf/wlan0/accept_redirects
```

File konfigurasi `mip6d.conf` untuk *mobile node* berisi :

```
NodeConfig MN;
DebugLevel 10;
MnDiscardHaParamProb enabled;
Interface "wlan0";
MnHomeLink "wlan0" {
    HomeAgentAddress 2001:db8:3c4d:5a::2;
    HomeAddress 2001:db8:3c4d:5a::3/64;
}
UseMnHaIPsec disabled;
```

- Konfigurasi *Correspondent node*
Correspondent node hanya memiliki 1 *interface*, sama seperti *mobile node*. Ada perintah yang harus diketik pada terminal *correspondent node* :

1. Konfigurasi alamat `wlan0`

```
# ifconfig wlan0 inet6 add 2001:db8:3c4d:5a::4/64
```

Untuk file konfigurasi `mip6d.conf` :

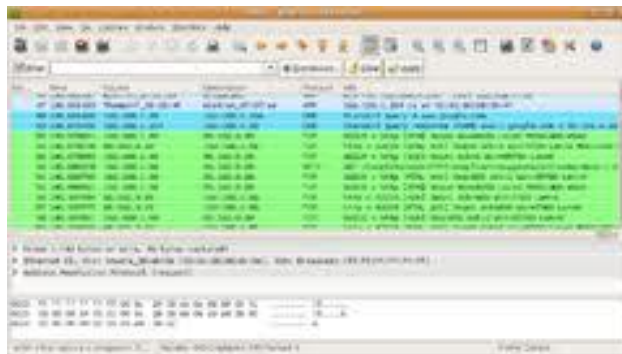
```
NodeConfig CN;
DebugLevel 10;
DoRouteOptimizationCN enabled;
```

Setelah semua *node* dikonfigurasi, saatnya menjalankan file konfigurasi yang telah dibuat tadi. Pada terminal tiap *node*, ketikkan nama file yang sudah dibuat, yaitu `mip6d` dan `radvd` pada tiap *node* yang memiliki file konfigurasi yang bersangkutan.

3.5.3 Konfigurasi Perangkat Lunak

1. Wireshark

Setelah diinstall, wireshark memiliki tampilan seperti Gambar 3.3



Gambar 3.3 Tampilan Graphical User Interface (GUI) Wireshark[3]

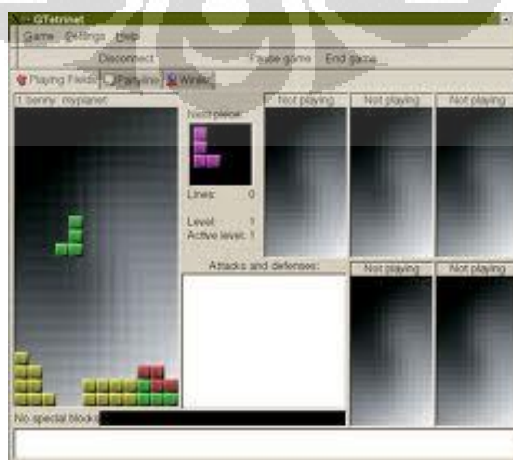
Wireshark dapat dengan mudah diinstall pada sistem operasi *Linux Ubuntu* lewat Synaptic Package Manager. Setelah membuka Synaptic Package Manager, search wireshark, klik kanan pada wireshark, lalu klik Mark for Installation, klik Apply, maka wireshark akan diinstall secara otomatis. Setelah diinstall, maka wireshark dapat dijalankan dari terminal dengan perintah :

`#sudo wireshark`

Setelah wireshark dijalankan, ada bagian filter yang dapat digunakan untuk memilah paket yang kita ingin lihat, misalnya paket dengan protokol TCP, UDP, dan sebagainya.

2. *Gtetrinet* dan *Tetrinetx*

Gtetrinet setelah diinstall akan memiliki tampilan seperti Gambar 3.4



Gambar 3.4 Tampilan GUI Gtetrinet[4]

Gtetrinet dan Tetrinetx bisa dengan mudah didapatkan sama seperti wireshark, yaitu pada Synaptic Package Manager. Keduanya bisa diinstall dengan cara yang sama dengan wireshark. Gtetrinet adalah perangkat lunak client dari permainan tetris ini, sedangkan Tetrinetx adalah perangkat lunak servernya. Tetrinetx akan dijalankan pada *correspondent node*, sedangkan Gtetrinet akan dijalankan pada *correspondent* dan *mobile node*. Kedua perangkat lunak ini dapat dijalankan dengan perintah :

```
#sudo tetrinetx
```

```
#sudo gtetrinet
```

Setelah server dijalankan, maka client dapat menekan tombol connect di bagian atas kiri dan memasukkan alamat server, lalu tekan Connect. Client yang sudah terkoneksi dengan server dapat melakukan chat sebelum game dimulai. Client yang pertama memasuki server disebut moderator. Moderator dapat menentukan kapan game dimulai, dipause, atau dihentikan.

3. MGEN

MGEN adalah perangkat lunak open source yang dapat membuat pola *traffic* sehingga jaringan dapat diberi beban dengan cara yang bervariasi. MGEN bisa didownload dari <http://cs.itd.nrl.navy.mil/work/mgen/>. *Traffic* yang dibuat juga dapat dicatat dalam sebuah log untuk kemudian dianalisa. Data log dapat digunakan untuk menghitung statistik performa jaringan, diantaranya *throughput*, *packet loss*, *delay*, dan sebagainya. MGEN berjalan di atas sistem operasi berbasis Unix seperti *Linux* dan juga platform WIN32.

Untuk menggunakan MGEN sebagai pembangkit *traffic*, diperlukan *host* yang berperan sebagai *server* dan *client*. *Server* mengirimkan paket-paket yang dibangkitkan, sedangkan *client* yang

mendengarkan permintaan koneksi dan meresponnya serta menerima paket-paket tersebut.

MGEN memiliki antarmuka command line. Untuk menjalankannya, format perintah yang digunakan adalah

```
#mgen [ipv6][input <scriptFile>][output <logFile>]
```

dan perintah yang dimasukkan pada command line adalah:

```
#mgen ipv6 input /bin/script.mgn
```

[ipv6] menunjukkan bahwa protokol yang digunakan adalah IPv6
[input <scriptFile>] menunjukkan script file yang berisi perintah-perintah membangkitkan *traffic* yang harus dijalankan oleh MGEN. <scriptFile> diganti dengan direktori tempat menyimpan script file yang diinginkan.

[output <logFile>] menunjukkan MGEN harus mencatat perintah-perintah yang dijalankan dalam sebuah log file. <logFile> diganti dengan direktori penyimpanan dan nama log file.

Script file adalah file berisi perintah-perintah yang harus dijalankan MGEN yang berperan dalam membangkitkan *traffic* pada jaringan.

Pada *client*, isi script file adalah sebagai berikut :

```
0.0 LISTEN UDP 5001,5002
```

```
0.0 LISTEN TCP 6000-6002
```

Dengan script file ini, *client* MGEN akan mendengarkan paket-paket protokol UDP yang dikirimkan *server* pada port 5001 dan 5002. *Client* MGEN juga akan mendengarkan permintaan sambungan dan kiriman paket TCP pada port 6000 sampai 6002.

Pada *server*, isi script file adalah sebagai berikut :

```
0.0 ON 1 UDP SRC 5001 DST 2001:db8:3c4d:5b::2/5001  
PERIODIC [5.0 2048]
```

```
0.0 ON 2 UDP SRC 5002 DST 2001:db8:3c4d:5b::2/5002 BURST
[RANDOM 3.0 PERIODIC [15.0 1024] EXP 2.0]
```

```
0.0 ON 3 UDP SRC 5002 DST 2001:db8:3c4d:5a::2/5002 BURST
[RANDOM 3.0 PERIODIC [15.0 1024] EXP 2.0]
```

```
0.0 ON 4 TCP DST 2001:db8:3c4d:5a::2/6000 PERIODIC [1
1048576] COUNT -1
```

Baris pertama berisi perintah untuk membangkitkan paket UDP dari port sumber (SRC) 5001 dengan alamat tujuan (DST) 2001:db8:3c4d:5b::2 pada port 5001. Paket yang dibangkitkan berukuran 2048 Byte dan dibangkitkan secara periodik sebanyak 5 paket per detik.

Baris kedua juga membangkitkan paket UDP dengan port asal 5002 dan port tujuan 5002 pada alamat yang sama, yaitu 2001:db8:3c4d:5b::2. Pola kemunculan paket adalah BURST yaitu berupa rentetan paket yang dikirimkan pada interval acak dengan rata-rata interval dari awal mulainya sebuah rentetan sampai rentetan berikutnya adalah 3 detik. Paket dikirimkan secara periodik sebanyak 15 paket per detik berukuran 1024 Byte. Durasi setiap rentetan berupa statistik eksponensial dengan rata-rata durasi rentetan adalah 2 detik. Baris ketiga sama dengan baris kedua tapi dengan alamat tujuan yang berbeda, yaitu 2001:db8:3c4d:5a::2.

Baris terakhir membangkitkan paket TCP dengan alamat tujuan 2001:db8:3c4d:5a::2 pada port 6000 secara periodik sebanyak 1 paket per detik berukuran 1024 Byte. COUNT menunjukkan jumlah paket yang dikirimkan tidak terbatas sampai proses MGEN dihentikan. Untuk pengiriman paket TCP, sebuah *client* harus terlebih dahulu mendengarkan untuk permintaan sambungan TCP agar *server* dapat mengirimkan paket.

MGEN *server* dijalankan pada *correspondent node* yang berada pada *home link* dan *client* pada *foreign router*.

BAB IV

PENGUJIAN DAN ANALISIS PARAMETER QOS JARINGAN MOBILE IPV6

4.1 Pengujian Jaringan Mobile IPv6

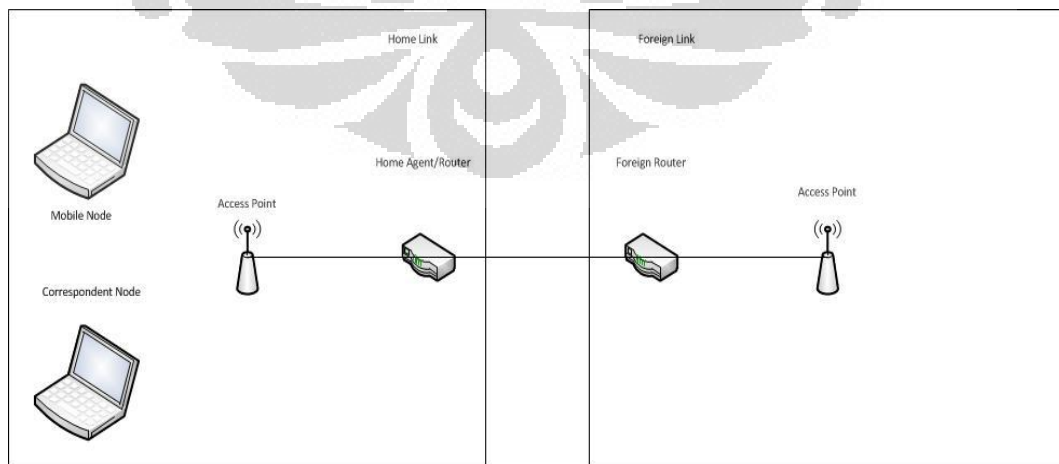
Jaringan *Mobile IPv6* pada skripsi ini terdiri dari *Correspondent node*, *Mobile node*, *Foreign Router*, dan *Home Router*. Jenis pengalamatan yang digunakan pada jaringan uji coba ini adalah *global address IPv6*.

Jaringan yang digunakan untuk uji coba dalam skripsi ini memiliki keterbatasan, yaitu tidak terkoneksi dengan jaringan luar sehingga tidak ada *traffic* yang persis sama seperti jaringan internet aslinya. Dengan mempertimbangkan hal ini, maka digunakan *traffic generator* sehingga jaringan uji coba bisa lebih menyerupai jaringan internet.

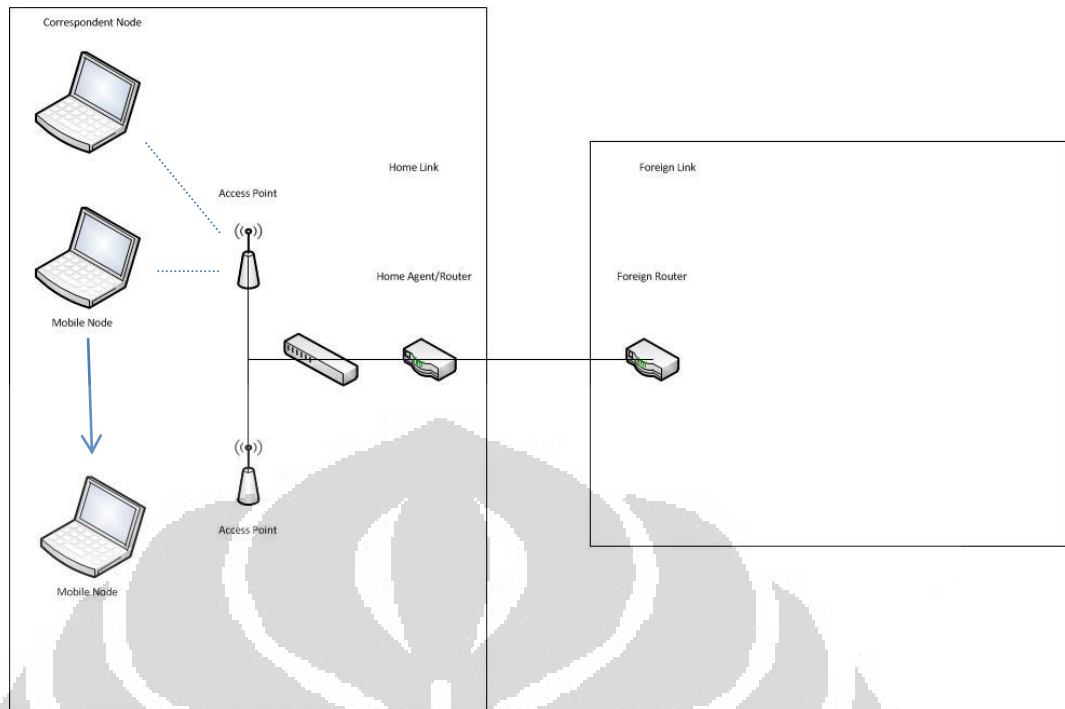
Dalam pengujian jaringan dilakukan beberapa skenario, tiap skenario akan dianalisa parameter QoSnya dengan menggunakan perangkat lunak *wireshark*. *Wireshark* akan menampilkan data yang lewat pada jaringan yang kemudian diolah sehingga didapatkan nilai *delay*, *transfer rate*, dan *packet loss* dari jaringan uji coba. Data yang akan dianalisa yaitu paket TCP.

Correspondent node akan menjalankan *Tetrinetx* dan *Gtetrinet*, lalu *mobile node* akan menjalankan *Gtetrinet* dan dihubungkan ke server. Untuk setiap jenis *handover*, akan dilakukan 10 kali pengambilan data.

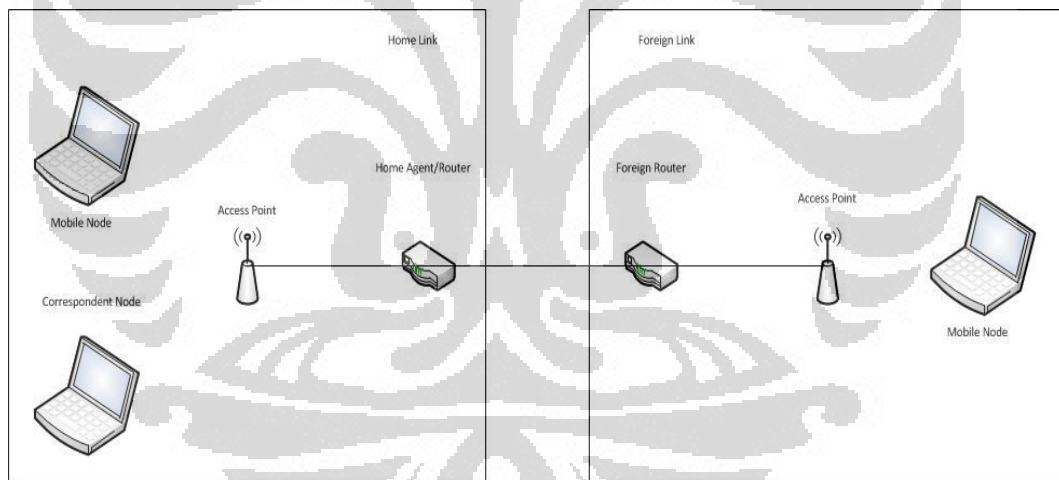
Gambar tiap skenario dapat dilihat pada Gambar 4.1, 4.2, dan 4.3



Gambar 4.1 Skenario Access Point Sama / Tanpa Handover



Gambar 4.2 Skenario Horizontal Handover



Gambar 4.3 Skenario Vertical Handover

4.2 Pengukuran dan Analisa

Pada bagian ini akan dilakukan analisa terhadap performansi jaringan *Mobile IPv6* untuk aplikasi *Gtetrinet*. Analisa ini dilakukan dengan memainkan aplikasi antara *mobile node* dan *correspondent node*. Analisa dilakukan saat *mobile node* tidak bergerak, yaitu saat berada pada *access point* yang sama dengan *correspondent node*, bergerak dengan *horizontal handover*, dan bergerak dengan *vertical handover*. Pergerakan dalam tiap skenario *handover*

dilakukan dengan berjalan kaki dari area jangkauan *access point* pertama ke *access point* kedua dengan kecepatan sekitar 10 km/jam. Proses analisa dilakukan terhadap data-data parameter QoS yaitu *delay*, *transfer rate*, dan *packet loss* yang diperoleh dari tiap skenario *handover*.

Analisa *delay* dan *packet loss* dilakukan berdasarkan data yang sampelnya terdapat pada lampiran 1, 3, dan 5. Sedangkan analisa *transfer rate* dilakukan berdasarkan data yang sampelnya terdapat pada lampiran 2, 4, dan 6.

4.2.1 Pengukuran dan Analisa Delay

4.2.1.1 Pengukuran Delay

Delay adalah waktu yang dibutuhkan oleh paket dari sisi pengirim untuk mencapai sisi penerima. *Delay* merupakan parameter yang sangat penting dalam game online, karena game online merupakan permainan real time.

Pada uji coba pengukuran *delay*, akan dimainkan perangkat lunak *Gtetrinet* antara *correspondent node* dan *mobile node*. Pengukuran parameter *delay* dilakukan sebanyak 10 kali untuk tiap skenario, dan dilakukan pengukuran tambahan yaitu saat kedua *node* bermain saat berada pada *access point* yang sama (tanpa *handover*). Nilai yang didapat saat berada pada *access point* yang sama hanya digunakan sebagai nilai pembanding untuk melihat seberapa kecil nilai *delay* minimum yang bisa didapatkan.

Pengukuran *delay* dibagi menjadi 2 bagian, yaitu *delay* saat mengupdate data dari *mobile node* → *correspondent node* (MN → CN), dan *delay* pengupdatean data *mobile node* (CN → MN).

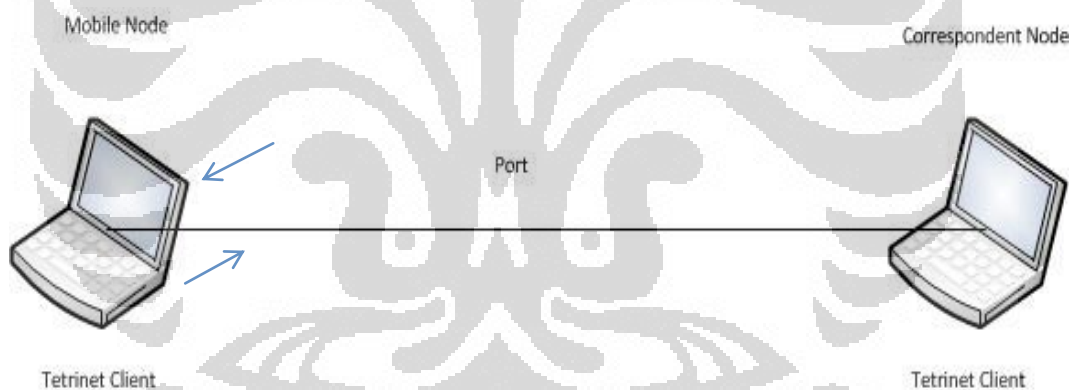
Gambar *delay* MN → CN dapat dilihat pada Gambar 4.4



Gambar 4.4 Delay MN – CN

Delay MN → CN adalah *delay* yang pengiriman data dari *tetrinet client* pada *mobile node* ke *port*, *port* ke *tetrinet client* pada *correspondent node*, lalu sampai kembali pada *tetrinet client* pada *mobile node*

Gambar *delay CN → MN* dapat dilihat pada Gambar 4.5 :



Gambar 4.5 Delay CN – MN

Delay CN → MN adalah *delay* dari *port* ke *tetrinet client* pada *mobile node*, lalu dari *tetrinet client* kembali ke *port*, *delay* ini sangat kecil karena hanya merupakan waktu update pada *mobile node*.

Hasil pengukuran *delay* dapat dilihat pada Tabel 4.1

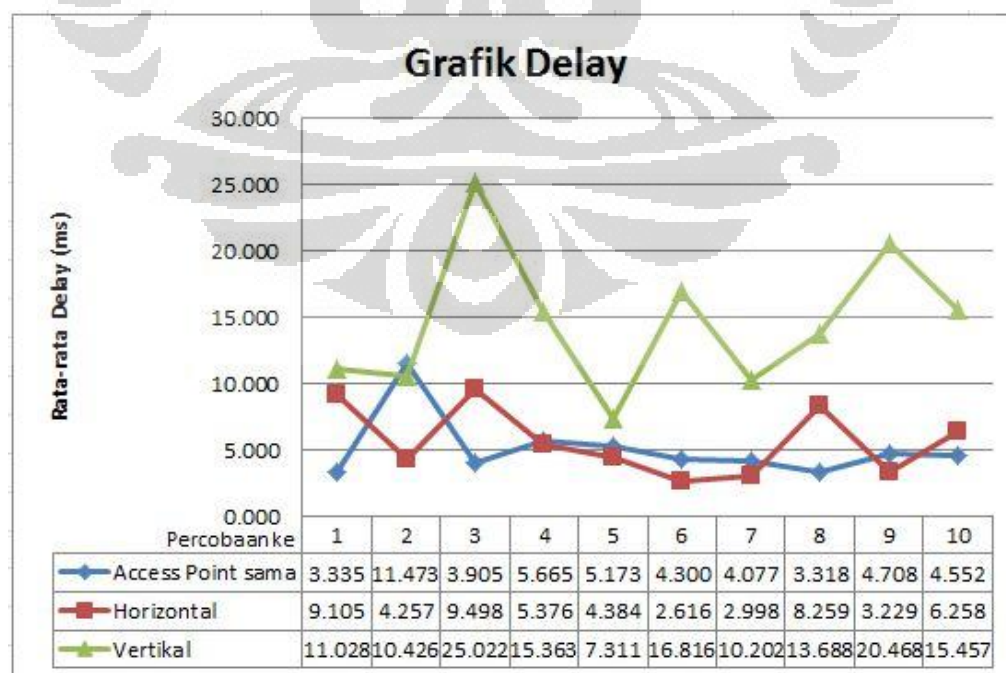
Tabel 4.1 Hasil Pengukuran *Delay*

Percobaan ke	<i>Access Point</i> sama		<i>Horizontal Handover</i>		<i>Vertical Handover</i>	
	MN - CN	CN - MN	MN - CN	CN - MN	MN - CN	CN - MN
	<i>Delay (ms)</i>					
1	3.335	0.054	9.105	0.074	11.028	0.090
2	11.473	0.046	4.257	0.071	10.426	0.087
3	3.905	0.050	9.498	0.078	22.022	0.090
4	5.665	0.045	5.376	0.066	15.363	0.087
5	5.173	0.053	4.384	0.078	7.311	0.089
6	4.300	0.046	2.616	0.078	16.816	0.105
7	4.077	0.039	2.998	0.074	10.202	0.079
8	3.318	0.043	8.259	0.079	13.688	0.084
9	4.708	0.046	3.229	0.081	20.468	0.086
10	4.552	0.046	6.258	0.082	15.457	0.097
Rata-rata	5.051	0.047	5.525	0.075	14.147	0.089

4.2.1.2 Analisa *Delay*

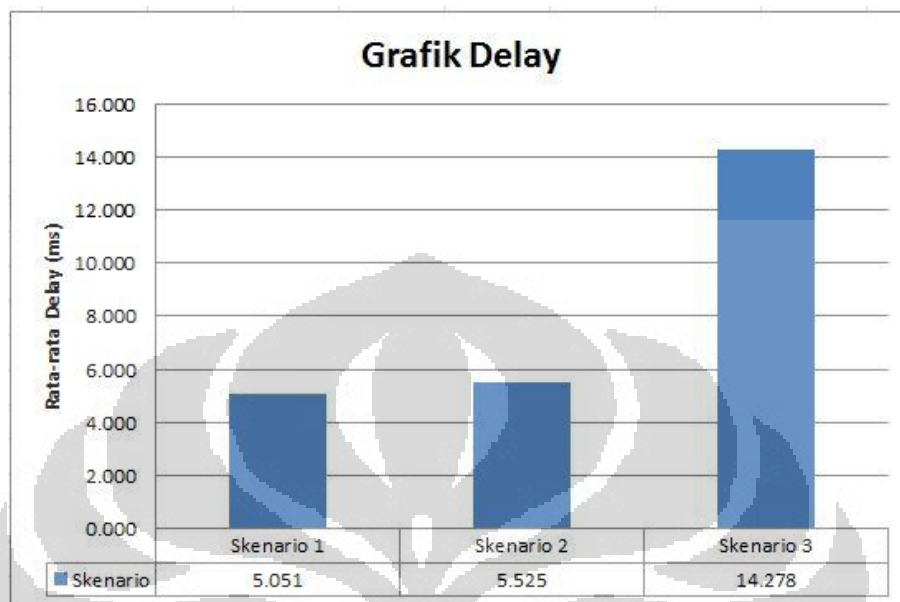
- a. Analisa pengukuran *delay* MN → CN

Grafik *delay* MN → CN terhadap waktu dapat dilihat pada Gambar 4.6



Gambar 4.6 Grafik Hasil Pengukuran *Delay* MN - CN terhadap Waktu

Grafik *delay* MN → CN terhadap skenario dapat dilihat pada Gambar 4.7



Gambar 4.7 Grafik Hasil Pengukuran *Delay* MN - CN terhadap Skenario

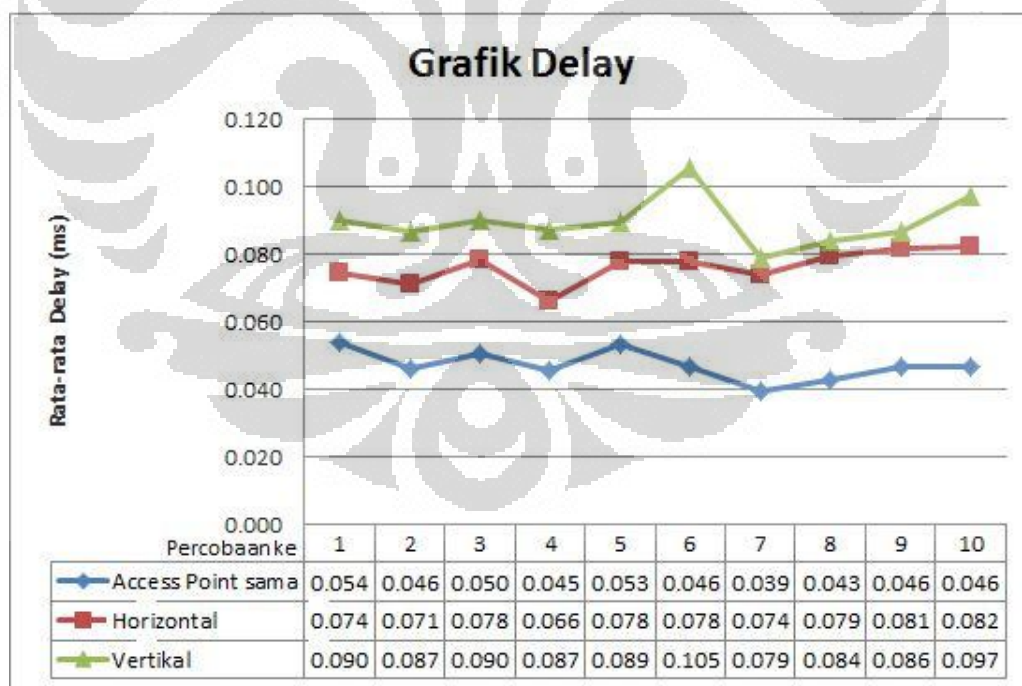
Skenario 1 adalah skenario tanpa *handover*, skenario 2 adalah skenario *horizontal handover*, dan skenario 3 adalah skenario *vertical handover*.

Berdasarkan Tabel 4.1 dan grafik pada Gambar 4.6 dan Gambar 4.7 di atas, hasil pengukuran menunjukkan *delay* yang tidak stabil untuk ketiga skenario, hal ini disebabkan oleh *traffic generator* yang memberi *traffic* dengan jeda waktu tertentu. *Delay* terbesar terjadi ketika uji coba skenario *vertical handover* yaitu dengan rata-rata 14,278 ms. *Delay* menurun pada uji coba skenario *horizontal handover*, dengan rata-rata 5,525 ms. Sebagai nilai pembandingan, skenario *access point* sama memiliki *delay* rata-rata 5,106 ms. *Delay* pada skenario *vertical handover* mengalami kenaikan 160,4% dibandingkan skenario *horizontal handover*. Hal ini disebabkan karena proses *horizontal handover* hanya berpindah *access point* dalam jaringan yang sama, sedangkan *vertical handover* berpindah *access point* pada jaringan yang berbeda. Proses *vertical handover*

menyebabkan *mobile node* semakin jauh dari *correspondent node* sehingga *delay*nya semakin besar. Perpindahan jaringan menyebabkan hop yang harus ditempuh untuk mencapai *correspondent node* meningkat, dan komunikasi yang terputus sementara menyebabkan keterlambatan data. *Mobile node* akan kembali terhubung dengan *correspondent node* setelah terhubung dengan jaringan barunya. Sebagai nilai pembandingan, skenario *handover* memiliki kenaikan *delay* yang signifikan jika dibandingkan dengan skenario tanpa *handover*. Skenario *horizontal handover* mengalami kenaikan sebesar 10,8%, sedangkan skenario *vertical handover* mengalami kenaikan sebesar 188,6%. Dari hasil ini dapat dilihat bahwa proses *handover* menyebabkan penurunan performa jaringan.

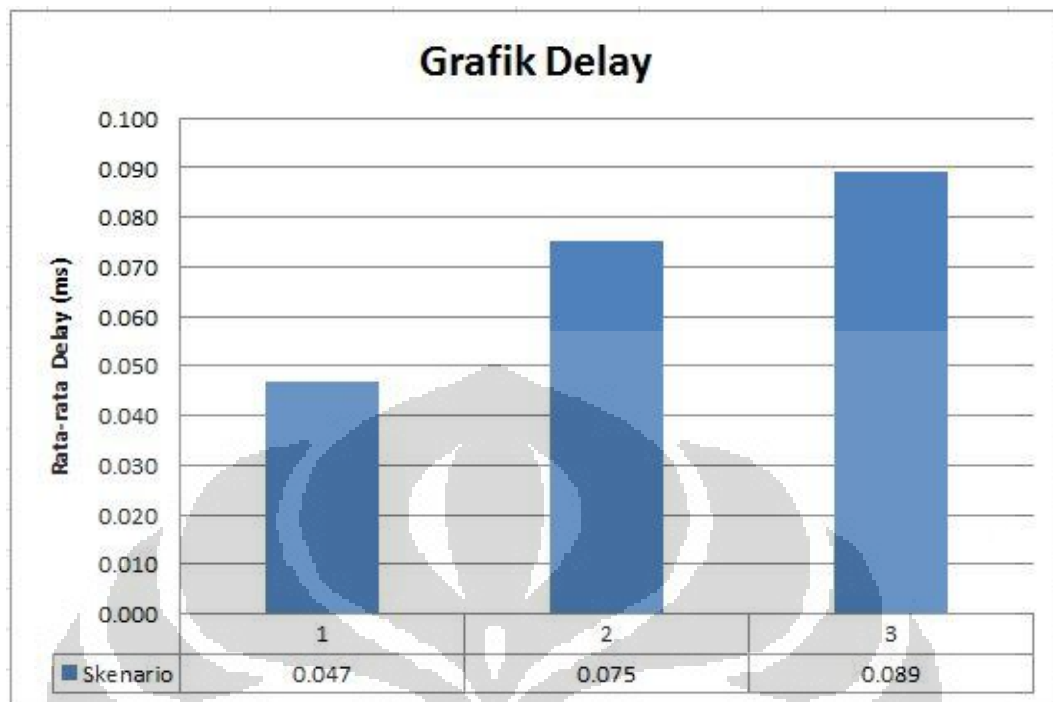
b. Analisa pengukuran *delay* CN → MN

Grafik *delay* CN → MN terhadap waktu dapat dilihat pada Gambar 4.8



Gambar 4.8 Grafik Hasil Pengukuran *Delay* CN - MN terhadap Waktu

Grafik *delay* CN → MN terhadap skenario dapat dilihat pada Gambar 4.9



Gambar 4.9 Grafik Hasil Pengukuran Delay CN - MN terhadap Skenario

Grafik CN → MN menunjukkan *delay* ketika *correspondent node* mengirimkan paket ke *mobile node*. Delay pada grafik ini sangat kecil, tidak mencapai 1 ms, dikarenakan *delay* ini adalah *delay* data dari port ke *mobile node*, lalu dikembalikan ke port. Jarak ini hanya waktu update gambar lawan pada *mobile node*.

Berdasarkan Gambar 4.8 dan Gambar 4.9, skenario *horizontal handover* memiliki *delay* yang sedikit lebih kecil dibandingkan dengan skenario *vertical handover*. Hal ini disebabkan oleh proses *handover* pada skenario *vertical handover* memakan waktu lebih lama, jika dibandingkan dengan skenario *horizontal handover*.

Skenario *vertical handover* memiliki rata-rata *delay* sebesar 0,089 ms, skenario *horizontal handover* memiliki rata-rata *delay* sebesar 0,075 ms, sedangkan skenario tanpa *handover* memiliki rata-rata *delay* sebesar 0,047 ms. Besarnya kenaikan *delay* yang terjadi pada skenario *vertical handover* dibandingkan dengan skenario *horizontal handover* adalah 17,5%, tetapi besarnya kenaikan *delay* skenario *vertical*

handover dan *horizontal handover* dibandingkan dengan skenario tanpa *handover* adalah 90,8% dan 62,3%.

4.2.2 Pengukuran dan Analisa Packet loss

4.2.2.1 Pengukuran Packet loss

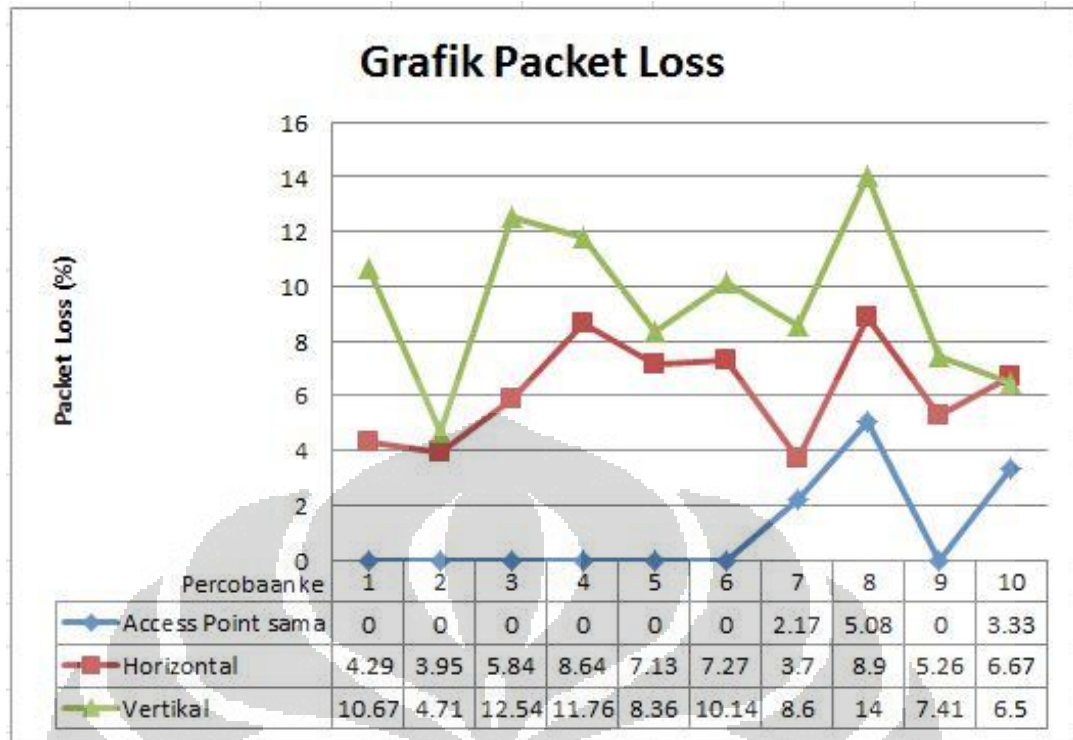
Packet loss merupakan besarnya paket yang hilang per seluruh paket yang dikirimkan pada saat transfer data berlangsung dari *source address* ke *destination address*. Semakin besar nilai *packet loss* suatu jaringan maka semakin menurun kualitas jaringan tersebut. Untuk game *Gtetrinet*, *packet loss* menyebabkan retransmisi data sehingga menyebabkan meningkatnya *delay* pada jaringan. Hasil pengukuran *packet loss* pada *mobile node* dapat dilihat pada Tabel 4.2

Tabel 4.2 Hasil Pengukuran *Packet loss*

Data ke	<i>Access Point</i> sama	<i>Horizontal Handover</i>	<i>Vertical Handover</i>
	Packet Loss (%)		
1	0	4.29	10.67
2	0	3.95	4.71
3	0	5.84	12.54
4	0	8.64	11.76
5	0	7.13	8.36
6	0	7.27	10.14
7	2.17	3.7	8.6
8	5.08	8.9	14
9	0	5.26	7.41
10	3.33	6.67	6.5
Rata-rata	1.058	6.165	9.469

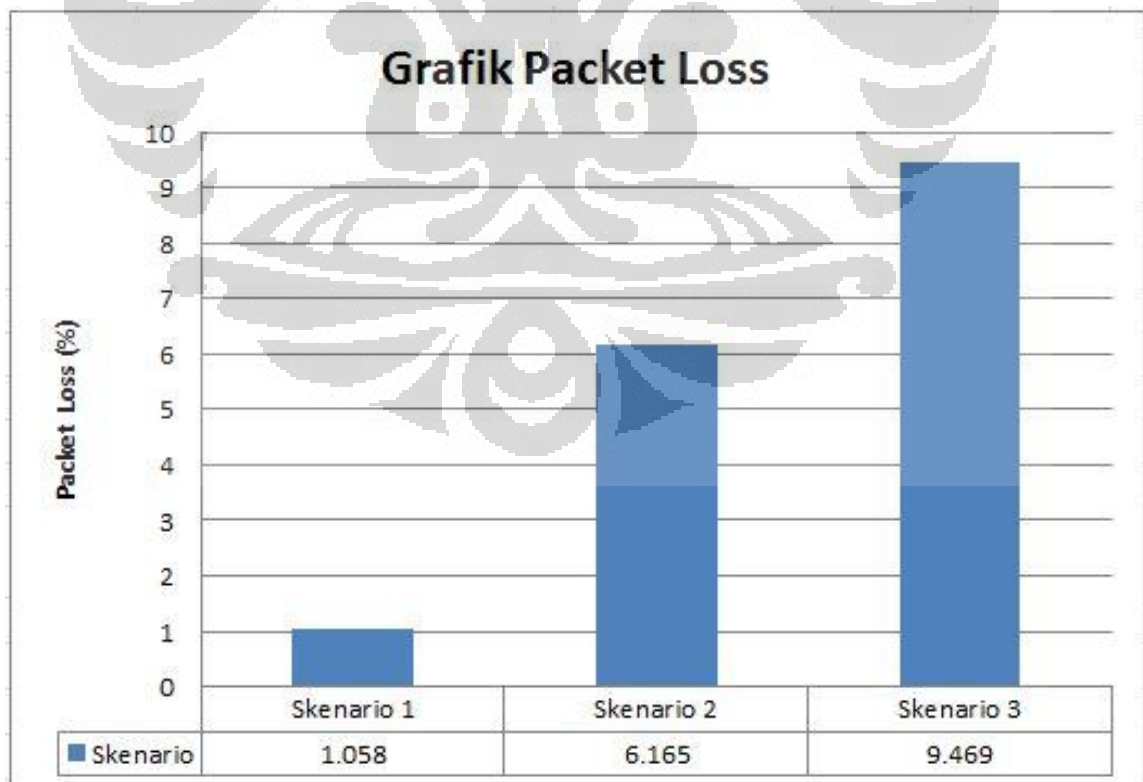
4.2.2.2 Analisa Packet loss

Grafik *packet loss* terhadap waktu dapat dilihat pada Gambar 4.10



Gambar 4.10 Grafik Hasil Pengukuran *Packet loss* terhadap Waktu

Grafik *packet loss* terhadap skenario dapat dilihat pada Gambar 4.11



Gambar 4.11 Grafik Hasil Pengukuran *Packet loss* terhadap Skenario

Berdasarkan Tabel 4.2 dan grafik pada Gambar 4.10 dan 4.11 di atas, maka dapat dilihat bahwa besarnya *packet loss* semakin bertambah dari skenario 1 sampai skenario 3. Skenario 1 memiliki nilai *packet loss* terkecil karena tidak terjadi proses *handover* sehingga komunikasi tidak terputus sama sekali. Sedangkan untuk skenario 2 dan 3, yaitu skenario *horizontal handover* dan *vertical handover*, komunikasi terputus selama beberapa saat karena *mobile node* berpindah tempat, sesuai dengan proses *handover* yang terjadi. Nilai *packet loss* pada uji coba ini berkisar dari 0% sampai 14%. *Packet loss* yang besar biasanya terjadi karena *traffic* jaringan yang besar, serta terputusnya komunikasi.

Dari Tabel 4.2 di atas dapat dilihat bahwa skenario *vertical handover* memiliki waktu *handover* yang lebih lama dibandingkan skenario *horizontal handover*, sehingga menghasilkan nilai *packet loss* terbesar. Selisih rata-rata *packet loss* antara skenario 2 dan 3 adalah sebesar 3,304%, angka yang masih tergolong kecil untuk aplikasi game online. *Packet loss* pada *Gtetrinet* bukanlah paket yang hilang lalu dibiarkan begitu saja, melainkan paket yang memiliki *delay* terlalu besar sehingga didrop dan diretransmisikan. Berdasarkan percobaan yang telah dilakukan, paket dengan *delay* di atas 150ms didrop dan diretransmisikan, paket-paket inilah yang diolah menjadi *packet loss*.

4.2.3 Pengukuran dan Analisa Transfer rate

4.2.3.1 Pengukuran Transfer rate

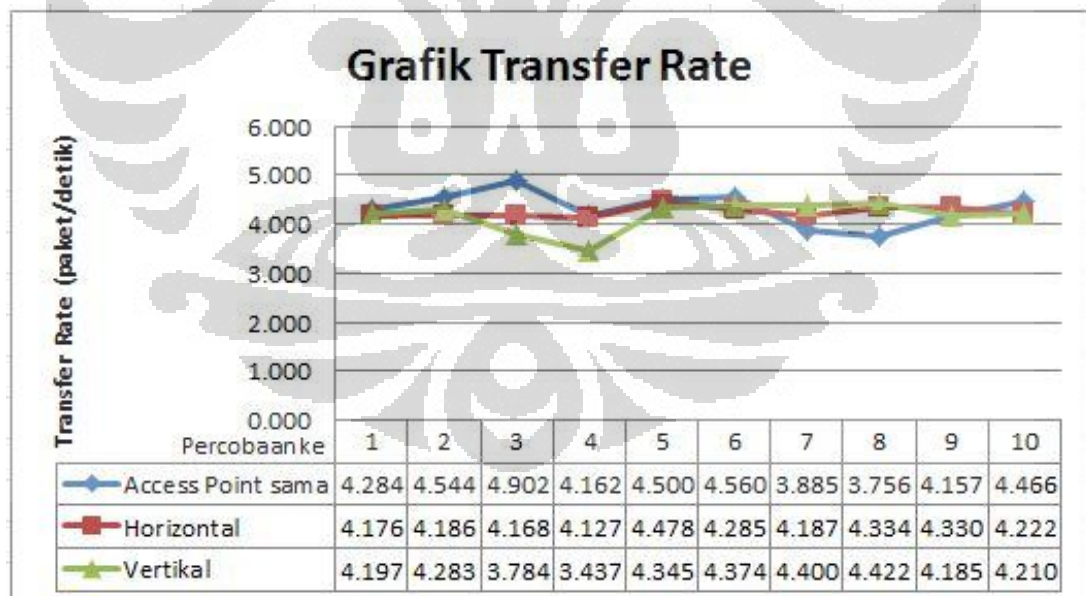
Transfer rate menunjukkan kemampuan sebuah jaringan untuk mengirimkan paket dalam satuan waktu. Pada pengukuran kali ini, satuan yang digunakan untuk merepresentasikan besarnya *transfer rate* adalah paket/detik. Pengukuran dilakukan dengan memberikan input sebanyak mungkin dalam satu detik. Hasil pengukuran *transfer rate* dapat dilihat pada Tabel 4.3

Tabel 4.3 Hasil Pengukuran *Transfer rate*

Data ke	<i>Access Point sama</i>	<i>Horizontal Handover</i>	<i>Vertical Handover</i>
	Transfer Rate (paket/detik)		
1	4.284128438	4.176462245	4.196509886
2	4.543735155	4.185996608	4.282681632
3	4.901648996	4.168021274	3.784240756
4	4.162062798	4.12745916	3.436780424
5	4.499718256	4.478075242	4.34525903
6	4.559711652	4.285377678	4.373768423
7	3.885209272	4.18685539	4.399919399
8	3.75625908	4.33416217	4.422123995
9	4.157111373	4.33024102	4.185038246
10	4.466074682	4.22241833	4.210383705
Rata-rata	4.32156597	4.249506912	4.16367055

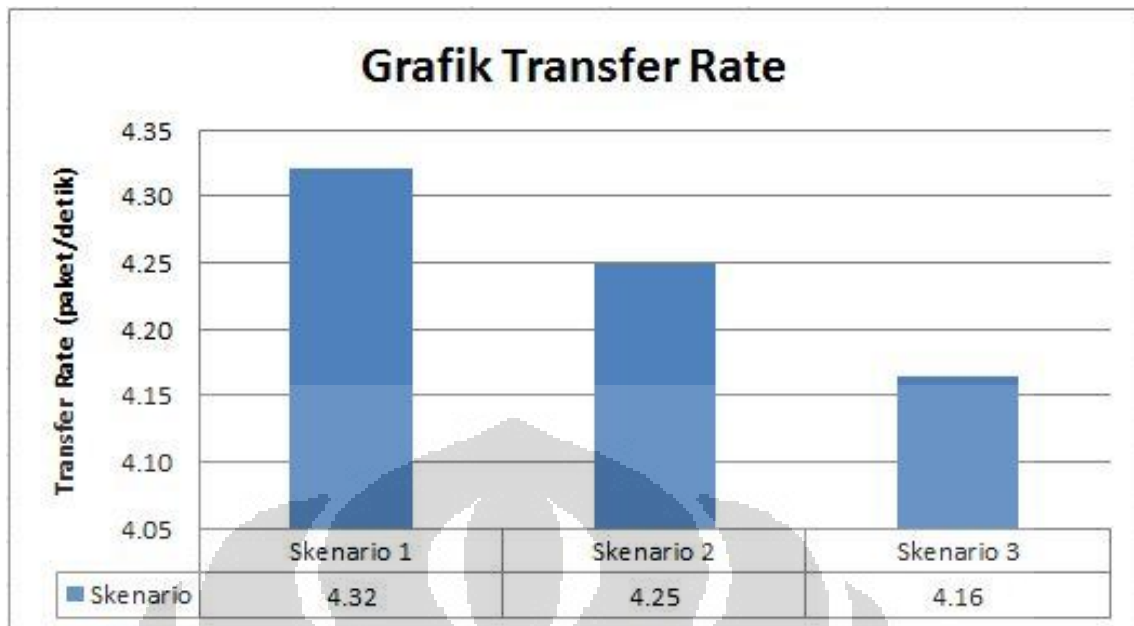
4.2.3.2 Analisa *Transfer rate*

Grafik *transfer rate* terhadap waktu dapat dilihat pada Gambar 4.12



Gambar 4.12 Grafik Hasil Pengukuran *Transfer rate* terhadap Waktu

Grafik *transfer rate* terhadap skenario dapat dilihat pada Gambar 4.13



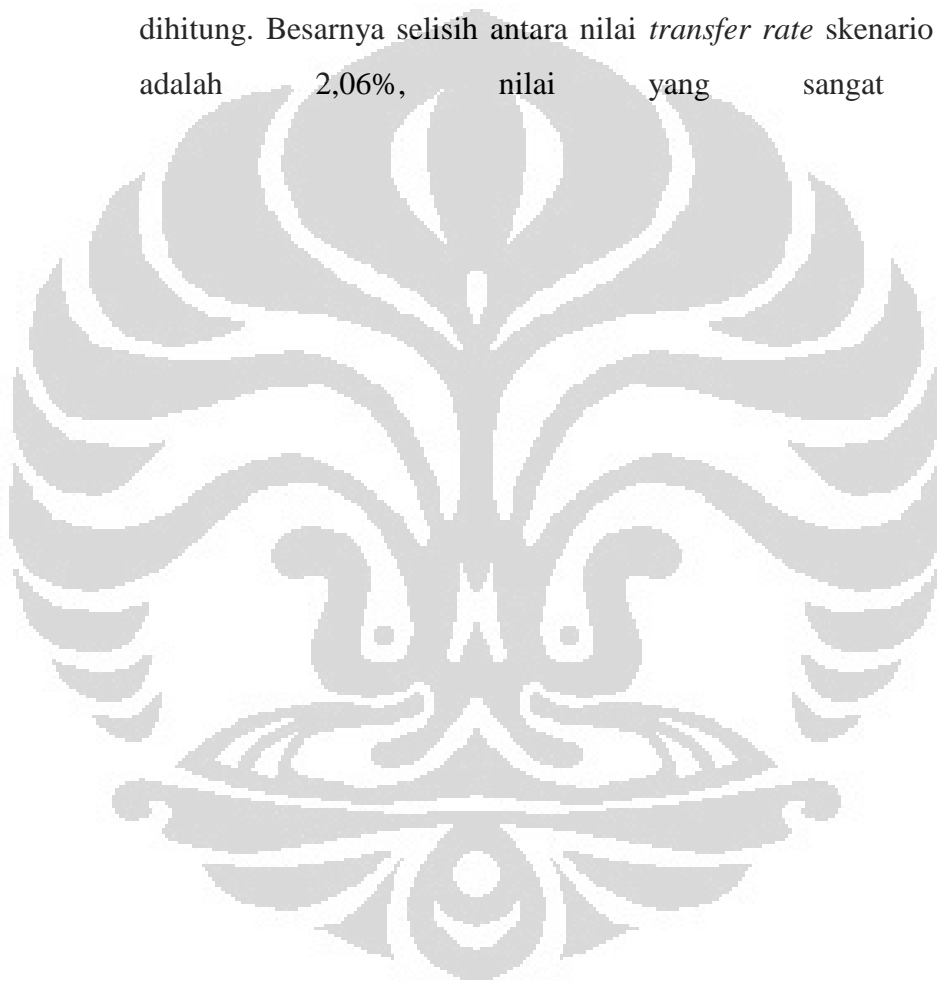
Gambar 4.13 Grafik Hasil Pengukuran *Transfer rate* terhadap Skenario

Dari grafik pada Gambar 4.12 dan 4.13, dapat dilihat bahwa *transfer rate* dari ketiga skenario tidak berbeda jauh, berkisar pada angka 4, tetapi skenario pertama tetap memberikan hasil yang paling baik. Hal ini disebabkan karena skenario 1 memiliki *delay* yang paling kecil di antara ketiga skenario tersebut. Besarnya nilai *transfer rate* sangat dipengaruhi oleh *delay*, karena jika *delay* suatu jaringan besar maka sangat sedikit paket yang bisa dikirimkan per detiknya.

Nilai *transfer rate* yang didapat pada uji coba ini juga dipengaruhi oleh aplikasi yang digunakan. Berdasarkan percobaan dan pengambilan data yang telah dilakukan, aplikasi *Gtetrinet* membatasi jumlah paket yang dapat dikirimkan, sesuai dengan jumlah pemain. Tiap pemain hanya dapat mengirimkan paket, dalam hal ini menjatuhkan balok tetris yang dimainkan, dengan interval tertentu, sekitar 0,9 detik. Saat pemain menjatuhkan balok tetris dalam permainan, dikirimkan 2 paket, yaitu paket dari client tetrinet ke port, lalu dari port ke tetrinet. Dengan bertambahnya jumlah pemain, maka jumlah paket yang dikirimkan dalam sebuah jaringan akan semakin banyak. Dengan mengasumsikan seorang pemain dapat mengirimkan

2 paket per detik, maka *transfer rate* jaringan akan berkisar pada angka = jumlah pemain x 2, yang dalam uji coba ini terdapat 2 pemain sehingga *transfer rate* berkisar pada angka 4.

Nilai *transfer rate* terkecil terjadi pada skenario 3, yaitu skenario *vertical handover*, tetapi tidak berbeda jauh dengan 2 skenario lainnya. Hal ini disebabkan karena nilai *transfer rate* yang dihitung tidak memerdulikan *packet loss*, sehingga semua *packet loss* ikut dihitung. Besarnya selisih antara nilai *transfer rate* skenario 2 dan 3 adalah 2,06%, nilai yang sangat kecil.



BAB V

KESIMPULAN

Setelah melakukan pengujian dan analisis jaringan *Mobile IPv6*, dapat disimpulkan beberapa poin sebagai berikut :

1. Proses *vertical handover* memiliki performa yang lebih rendah dibandingkan proses *horizontal handover*
2. Hasil pengukuran menunjukkan bahwa nilai *delay* untuk proses *vertical handover* mengalami penurunan sebesar 17,5% untuk *delay Mobile Node* → *Correspondent Node* dan 160,4% untuk *delay Correspondent Node* → *Mobile Node* jika dibandingkan dengan proses *horizontal handover*. Hal ini disebabkan karena proses *vertical handover* memerlukan waktu yang lebih lama dibandingkan *horizontal handover*.
3. Hasil pengukuran *packet loss* pada skenario *vertical handover* dapat dilihat dari *packet loss* sebesar 9,401%. Sedangkan hasil pengukuran skenario *horizontal handover* menunjukkan *packet loss* sebesar 6,165%.
4. Hasil pengukuran *transfer rate* menunjukkan bahwa besarnya nilai *transfer rate* dipengaruhi oleh 3 hal, yaitu : *delay*, *packet loss*, dan aplikasi yang bersangkutan, terlihat dari hasil pengukuran yang tidak jauh berbeda untuk tiap skenario. Skenario 1 memiliki *transfer rate* sebesar 4,32 paket/detik, skenario 2 sebesar 4,25 paket/detik, sedangkan skenario 3 sebesar 4,16 paket/detik. *Transfer rate* pada skenario 3 hanya 2,06% lebih rendah dibandingkan dengan skenario 2.
5. Berdasarkan pengamatan, skenario *vertical handover* memerlukan waktu yang lebih lama untuk proses *handover* dibandingkan dengan skenario *horizontal handover*. Hal ini berpengaruh terhadap pengukuran parameter QoS seperti *delay* dan *packet loss*.
6. Secara umum, proses *handover* mengurangi performa jaringan dibandingkan dengan tidak adanya proses *handover*, tetapi proses *handover* sangat dibutuhkan untuk mendukung mobilitas *user* di masa yang akan datang.

DAFTAR ACUAN

- [1] Understanding Mobile IPv6 (2007, January). Microsoft Corporation
- [2] Martin Dunmore, & Theo Pagtzis Mobile (2005, June). Mobile IPv6 Handovers: Performance Analysis and Evaluation. 6net.
- [3] <http://liessundari.blogspot.com/2010/06/wireshark.html> diakses pada tanggal 3-1-2012 jam 2:30 AM
- [4] <http://www.home.unix-ag.org/bmeurer/NetBSD/packages.html> diakses pada tanggal 3-1-2012 jam 2:45 AM
- [5] Understanding IPv6 (2008). Microsoft Corporation
- [6] www.scribd.com/doc/61283919/QoS
- [7] <http://sekedar-tahu.blogspot.com/2010/06/sejarah-dan-asal-usul-game-tetris.html> diakses pada tanggal 9-1-2012 jam 10:52 AM
- [8] Supriyatna, Dedi. “Analisa Performansi Aplikasi Video Streaming pada Jaringan Mobile IPv6”, Universitas Indonesia, Desember 2010

DAFTAR REFERENSI

- [1] <http://ipv6.com/articles/general/IPv6-Addressing.htm> diakses pada tanggal 22-11-2011 jam 3:28 AM
- [2] <http://technet.microsoft.com/en-us/library/bb726997.aspx#ECAA> diakses pada tanggal 24-11-2011 jam 3:41 AM
- [3] www.ipv6-es.com/02/docs/patrick_grossetete_2.pdf diakses pada tanggal 24-11-2011 jam 3:53 AM
- [4] <http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ipv6-mobile.html> diakses pada tanggal 24-11-2011 jam 4:10 AM
- [5] <http://www.fmipv6.org/index.php/Documentation/Testbed#toc6> diakses pada tanggal 25-11-2011 jam 2:40 PM
- [6] http://ipv6tf.org.eg/documents/MobilityIPv6/Mobile_IPv6.pdf diakses pada tanggal 25-11-2011 jam 5:00 PM
- [7] Charles Sellers (2008, April). Introduction to Mobile IPv6
- [8] <http://toko.baliwae.com/download/buku-pintar/buku-jarkom2.pdf> diakses pada tanggal 25-11-2011 jam 7:00 PM
- [9] <http://software.nautilus6.org/packages/ubuntu/hardy> diakses pada tanggal 25-11-2011 jam 10:00 PM

LAMPIRAN

1. Sampel Data *Vertical Handover Delay* dan *Packet loss*

No.	Time	Protocol	Length
13	2.289149	TCP	100
15	2.352767	TCP	86
21	3.88945	TCP	100
22	3.889505	TCP	86
46	5.300792	TCP	100
48	5.318982	TCP	86
67	6.799859	TCP	100
68	6.799953	TCP	86
71	8.175893	TCP	100
76	8.396374	TCP	86
80	9.322442	TCP	100
81	9.3225	TCP	86
84	10.288505	TCP	100
86	10.337225	TCP	86
89	11.87513	TCP	100
90	11.875219	TCP	86
91	12.608741	TCP	100
92	12.627634	TCP	86
97	13.925192	TCP	100
98	13.925278	TCP	86
101	15.016123	TCP	100
104	15.31504	TCP	100
105	15.317336	TCP	86
106	15.321663	TCP	98
114	16.489006	TCP	100
115	16.489062	TCP	86
118	17.291021	TCP	100
121	17.77504	TCP	100
122	17.780453	TCP	86
130	18.53306	TCP	100
131	18.533109	TCP	86
134	19.556774	TCP	100
135	19.561941	TCP	86
139	21.090292	TCP	100
140	21.090385	TCP	86
142	21.673903	TCP	100
143	21.682679	TCP	86
148	22.629589	TCP	100
149	22.629648	TCP	86
151	23.538328	TCP	100
153	23.556281	TCP	86
171	24.756425	TCP	100
172	24.756512	TCP	86
184	25.401129	TCP	100
185	25.407155	TCP	86
214	26.86245	TCP	100
215	26.862541	TCP	86
224	27.306616	TCP	100
225	27.313212	TCP	86
260	34.030863	TCP	94
261	34.391039	TCP	94
265	35.110915	TCP	94
270	35.532643	TCP	100
271	35.532717	TCP	86
274	36.040985	TCP	100

275	36.041038	TCP	98
276	36.554857	TCP	94
287	39.443035	TCP	94
303	45.211036	TCP	94
305	45.365734	TCP	86

No. Info

13 tetrinet > 36138 [PSH, ACK] Seq=1 Ack=1 Win=120 Len=14 TSval=165797 TSecr=1063289

15 36138 > tetrinet [ACK] Seq=1 Ack=15 Win=71 Len=0 TSval=1064031 TSecr=165797

21 36138 > tetrinet [PSH, ACK] Seq=1 Ack=15 Win=71 Len=14 TSval=1064389 TSecr=165797

22 tetrinet > 36138 [ACK] Seq=15 Ack=15 Win=120 Len=0 TSval=166197 TSecr=1064389

46 tetrinet > 36138 [PSH, ACK] Seq=15 Ack=15 Win=120 Len=14 TSval=166550 TSecr=1064389

48 36138 > tetrinet [ACK] Seq=15 Ack=29 Win=71 Len=0 TSval=1064786 TSecr=166550

67 36138 > tetrinet [PSH, ACK] Seq=15 Ack=29 Win=71 Len=14 TSval=1065050 TSecr=166550

68 tetrinet > 36138 [ACK] Seq=29 Ack=29 Win=120 Len=0 TSval=166925 TSecr=1065050

71 tetrinet > 36138 [PSH, ACK] Seq=29 Ack=29 Win=120 Len=14 TSval=167269 TSecr=1065050

76 36138 > tetrinet [ACK] Seq=29 Ack=43 Win=71 Len=0 TSval=1065503 TSecr=167269

80 36138 > tetrinet [PSH, ACK] Seq=29 Ack=43 Win=71 Len=14 TSval=1065730 TSecr=167269

81 tetrinet > 36138 [ACK] Seq=43 Ack=43 Win=120 Len=0 TSval=167555 TSecr=1065730

84 tetrinet > 36138 [PSH, ACK] Seq=43 Ack=43 Win=120 Len=14 TSval=167797 TSecr=1065730

86 36138 > tetrinet [ACK] Seq=43 Ack=57 Win=71 Len=0 TSval=1066033 TSecr=167797

89 36138 > tetrinet [PSH, ACK] Seq=43 Ack=57 Win=71 Len=14 TSval=1066327 TSecr=167797

90 tetrinet > 36138 [ACK] Seq=57 Ack=57 Win=120 Len=0 TSval=168194 TSecr=1066327

91 tetrinet > 36138 [PSH, ACK] Seq=57 Ack=57 Win=120 Len=14 TSval=168377 TSecr=1066327

92 36138 > tetrinet [ACK] Seq=57 Ack=71 Win=71 Len=0 TSval=1066611 TSecr=168377

97 36138 > tetrinet [PSH, ACK] Seq=57 Ack=71 Win=71 Len=14 TSval=1066889 TSecr=168377

98 tetrinet > 36138 [ACK] Seq=71 Ack=71 Win=120 Len=0 TSval=168706 TSecr=1066889

101 tetrinet > 36138 [PSH, ACK] Seq=71 Ack=71 Win=120 Len=14 TSval=168979 TSecr=1066889

104 [TCP Retransmission] tetrinet > 36138 [PSH, ACK] Seq=71 Ack=71 Win=120 Len=14 TSval=169054 TSecr=1066889

105 36138 > tetrinet [ACK] Seq=71 Ack=85 Win=71 Len=0 TSval=1067212 TSecr=168979

106 [TCP Dup ACK 105#1] 36138 > tetrinet [ACK] Seq=71 Ack=85 Win=71 Len=0 TSval=1067287 TSecr=169054 SLE=71 SRE=85

114 36138 > tetrinet [PSH, ACK] Seq=71 Ack=85 Win=71 Len=14 TSval=1067465 TSecr=169054

115 tetrinet > 36138 [ACK] Seq=85 Ack=85 Win=120 Len=0 TSval=169347 TSecr=1067465

118 tetrinet > 36138 [PSH, ACK] Seq=85 Ack=85 Win=120 Len=14 TSval=169548 TSecr=1067465

121 [TCP Retransmission] tetrinet > 36138 [PSH, ACK] Seq=85 Ack=85 Win=120 Len=14 TSval=169669 TSecr=1067465

122 36138 > tetrinet [ACK] Seq=85 Ack=99 Win=71 Len=0 TSval=1067902 TSecr=169669

130 36138 > tetrinet [PSH, ACK] Seq=85 Ack=99 Win=71 Len=14 TSval=1068034 TSecr=169669

131 tetrinet > 36138 [ACK] Seq=99 Ack=99 Win=120 Len=0 TSval=169858 TSecr=1068034

134 tetrinet > 36138 [PSH, ACK] Seq=99 Ack=99 Win=120 Len=14 TSval=170114 TSecr=1068034

135 36138 > tetrinet [ACK] Seq=99 Ack=113 Win=71 Len=0 TSval=1068348 TSecr=170114

139 36138 > tetrinet [PSH, ACK] Seq=99 Ack=113 Win=71 Len=14 TSval=1068650 TSecr=170114

140 tetrinet > 36138 [ACK] Seq=113 Ack=113 Win=120 Len=0 TSval=170497 TSecr=1068650

142 tetrinet > 36138 [PSH, ACK] Seq=113 Ack=113 Win=120 Len=14 TSval=170643 TSecr=1068650

143 36138 > tetrinet [ACK] Seq=113 Ack=127 Win=71 Len=0 TSval=1068878 TSecr=170643

148 36138 > tetrinet [PSH, ACK] Seq=113 Ack=127 Win=71 Len=14 TSval=1069078 TSecr=170643

149 tetrinet > 36138 [ACK] Seq=127 Ack=127 Win=120 Len=0 TSval=170882 TSecr=1069078

151 tetrinet > 36138 [PSH, ACK] Seq=127 Ack=127 Win=120 Len=14 TSval=171109 TSecr=1069078

153 36138 > tetrinet [ACK] Seq=127 Ack=141 Win=71 Len=0 TSval=1069346 TSecr=171109

171 36138 > tetrinet [PSH, ACK] Seq=127 Ack=141 Win=71 Len=14 TSval=1069647 TSecr=171109

172 tetrinet > 36138 [ACK] Seq=141 Ack=141 Win=120 Len=0 TSval=171414 TSecr=1069647

184 tetrinet > 36138 [PSH, ACK] Seq=141 Ack=141 Win=120 Len=14 TSval=171575 TSecr=1069647

185 36138 > tetrinet [ACK] Seq=141 Ack=155 Win=71 Len=0 TSval=1069809 TSecr=171575

214 36138 > tetrinet [PSH, ACK] Seq=141 Ack=155 Win=71 Len=14 TSval=1070173 TSecr=171575

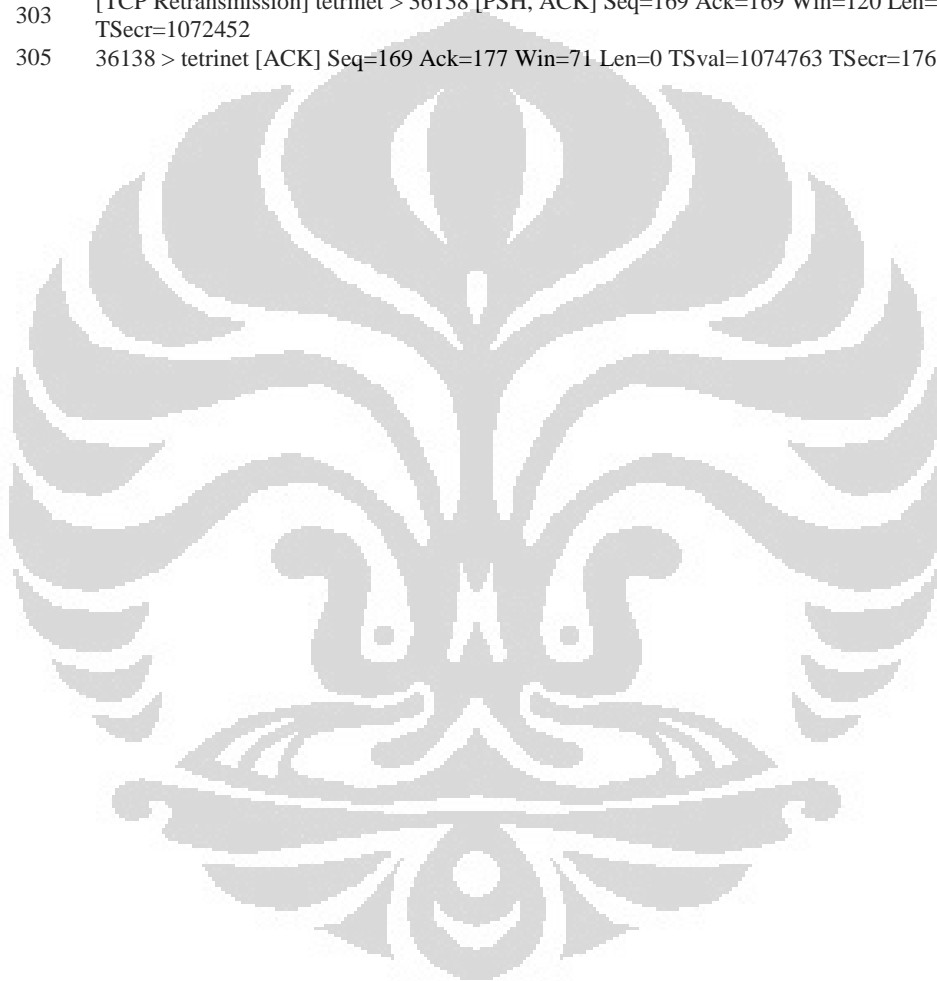
215 tetrinet > 36138 [ACK] Seq=155 Ack=155 Win=120 Len=0 TSval=171940 TSecr=1070173

224 tetrinet > 36138 [PSH, ACK] Seq=155 Ack=155 Win=120 Len=14 TSval=172051 TSecr=1070173

225 36138 > tetrinet [ACK] Seq=155 Ack=169 Win=71 Len=0 TSval=1070285 TSecr=172051

260 tetrinet > 36138 [PSH, ACK] Seq=169 Ack=155 Win=120 Len=8 TSval=173733 TSecr=1070285

261 [TCP Retransmission] tetrinet > 36138 [PSH, ACK] Seq=169 Ack=155 Win=120 Len=8 TSval=173823
TSecr=1070285
265 [TCP Retransmission] tetrinet > 36138 [PSH, ACK] Seq=169 Ack=155 Win=120 Len=8 TSval=174003
TSecr=1070285
270 36138 > tetrinet [PSH, ACK] Seq=155 Ack=169 Win=71 Len=14 TSval=1072244 TSecr=172051
271 tetrinet > 36138 [ACK] Seq=177 Ack=169 Win=120 Len=0 TSval=174108 TSecr=1072244
274 [TCP Retransmission] 36138 > tetrinet [PSH, ACK] Seq=155 Ack=169 Win=71 Len=14 TSval=1072452
TSecr=172051
275 [TCP Dup ACK 271#1] tetrinet > 36138 [ACK] Seq=177 Ack=169 Win=120 Len=0 TSval=174235 TSecr=1072452
SLE=155 SRE=169
276 [TCP Retransmission] tetrinet > 36138 [PSH, ACK] Seq=169 Ack=169 Win=120 Len=8 TSval=174364
TSecr=1072452
287 [TCP Retransmission] tetrinet > 36138 [PSH, ACK] Seq=169 Ack=169 Win=120 Len=8 TSval=175086
TSecr=1072452
303 [TCP Retransmission] tetrinet > 36138 [PSH, ACK] Seq=169 Ack=169 Win=120 Len=8 TSval=176528
TSecr=1072452
305 36138 > tetrinet [ACK] Seq=169 Ack=177 Win=71 Len=0 TSval=1074763 TSecr=176528



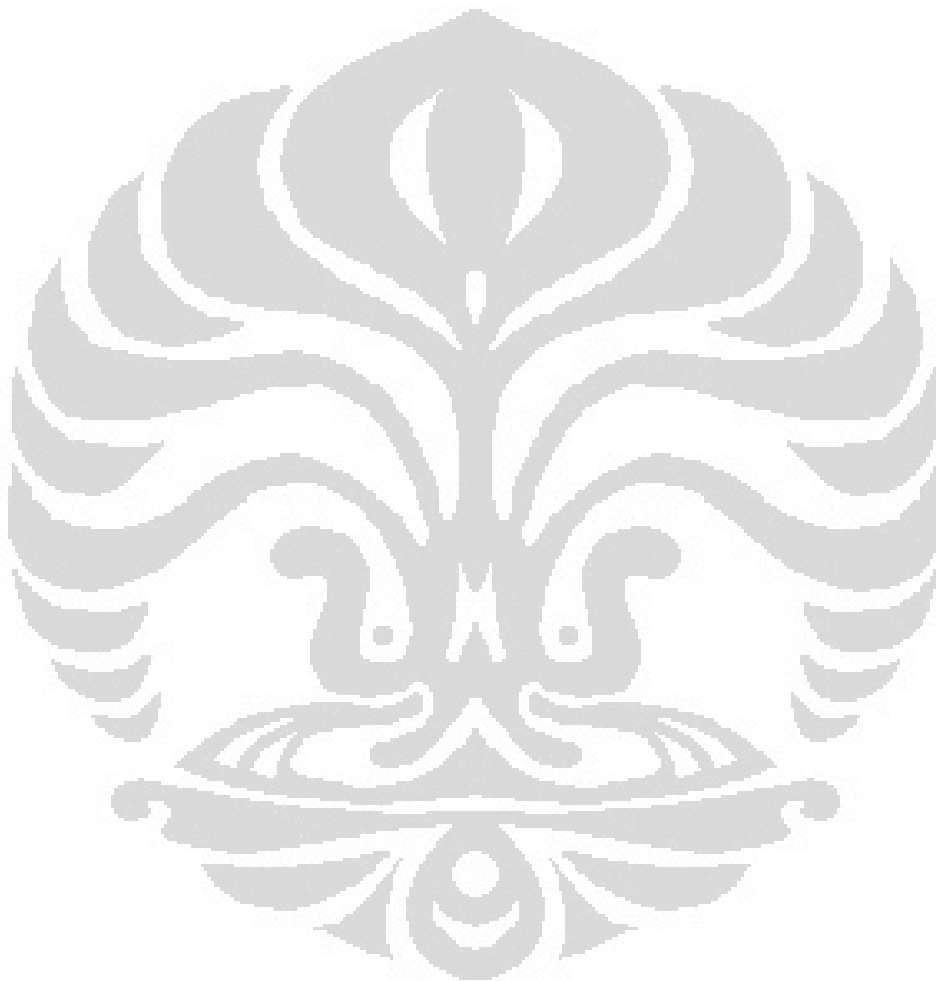
2. Sampel Data *Vertical Handover Transfer rate*

No.	Time	Protocol	Length
12	0.563466	TCP	100
13	0.563556	TCP	86
22	0.988958	TCP	100
23	0.991306	TCP	86
36	1.723293	TCP	100
37	1.723382	TCP	86
43	2.021307	TCP	100
51	2.417539	TCP	100
52	2.420257	TCP	86
59	2.767807	TCP	100
60	2.767885	TCP	86
65	3.028629	TCP	100
66	3.033046	TCP	86
84	3.903264	TCP	100
85	3.903356	TCP	86
89	4.087152	TCP	100
92	4.147413	TCP	86
109	5.043322	TCP	100
110	5.043409	TCP	86
111	5.110773	TCP	100
114	5.165602	TCP	86
133	6.163245	TCP	100
134	6.229258	TCP	86
140	6.363406	TCP	100
141	6.363484	TCP	86
156	7.20798	TCP	100
160	7.281069	TCP	86
166	7.644013	TCP	100
167	7.644101	TCP	86
178	8.260629	TCP	100
181	8.322664	TCP	86
189	8.836078	TCP	100
190	8.836162	TCP	86
198	9.312578	TCP	100
201	9.360247	TCP	86
210	9.988532	TCP	100
211	9.988623	TCP	86
221	10.369427	TCP	100
222	10.375175	TCP	86
239	11.249306	TCP	100
240	11.249395	TCP	86
246	11.459938	TCP	100
253	11.72154	TCP	100
254	11.725669	TCP	86
265	12.31959	TCP	100
266	12.319681	TCP	86
268	12.463686	TCP	355
269	12.467096	TCP	86
270	12.467156	TCP	435
276	12.71337	TCP	435
277	12.716422	TCP	86
265	35.110915	TCP	94
270	35.532643	TCP	100
271	35.532717	TCP	86
274	36.040985	TCP	100
275	36.041038	TCP	98
276	36.554857	TCP	94

287	39.443035	TCP	94
303	45.211036	TCP	94
305	45.365734	TCP	86

No.	Info
12	51264 > tetrinet [PSH, ACK] Seq=1 Ack=1 Win=129 Len=14 TSval=1506622 TSecr=607829
13	tetrinet > 51264 [ACK] Seq=1 Ack=15 Win=137 Len=0 TSval=608369 TSecr=1506622
22	tetrinet > 51264 [PSH, ACK] Seq=1 Ack=15 Win=137 Len=14 TSval=608475 TSecr=1506622
23	51264 > tetrinet [ACK] Seq=15 Ack=15 Win=129 Len=0 TSval=1506729 TSecr=608475
36	51264 > tetrinet [PSH, ACK] Seq=15 Ack=15 Win=129 Len=14 TSval=1506912 TSecr=608475
37	tetrinet > 51264 [ACK] Seq=15 Ack=29 Win=137 Len=0 TSval=608659 TSecr=1506912
43	tetrinet > 51264 [PSH, ACK] Seq=15 Ack=29 Win=137 Len=14 TSval=608733 TSecr=1506912 [TCP Retransmission] tetrinet > 51264 [PSH, ACK] Seq=15 Ack=29 Win=137 Len=14 TSval=608833
51	TSecr=1506912
52	51264 > tetrinet [ACK] Seq=29 Ack=29 Win=129 Len=0 TSval=1507086 TSecr=608833
59	51264 > tetrinet [PSH, ACK] Seq=29 Ack=29 Win=129 Len=14 TSval=1507171 TSecr=608833
60	tetrinet > 51264 [ACK] Seq=29 Ack=43 Win=137 Len=0 TSval=608920 TSecr=1507171
65	tetrinet > 51264 [PSH, ACK] Seq=29 Ack=43 Win=137 Len=14 TSval=608985 TSecr=1507171
66	51264 > tetrinet [ACK] Seq=43 Ack=43 Win=129 Len=0 TSval=1507239 TSecr=608985
84	51264 > tetrinet [PSH, ACK] Seq=43 Ack=43 Win=129 Len=14 TSval=1507457 TSecr=608985
85	tetrinet > 51264 [ACK] Seq=43 Ack=57 Win=137 Len=0 TSval=609204 TSecr=1507457
89	tetrinet > 51264 [PSH, ACK] Seq=43 Ack=57 Win=137 Len=14 TSval=609250 TSecr=1507457
92	51264 > tetrinet [ACK] Seq=57 Ack=57 Win=129 Len=0 TSval=1507503 TSecr=609250
109	51264 > tetrinet [PSH, ACK] Seq=57 Ack=57 Win=129 Len=14 TSval=1507742 TSecr=609250
110	tetrinet > 51264 [ACK] Seq=57 Ack=71 Win=137 Len=0 TSval=609489 TSecr=1507742
111	tetrinet > 51264 [PSH, ACK] Seq=57 Ack=71 Win=137 Len=14 TSval=609506 TSecr=1507742
114	51264 > tetrinet [ACK] Seq=71 Ack=71 Win=129 Len=0 TSval=1507759 TSecr=609506
133	tetrinet > 51264 [PSH, ACK] Seq=71 Ack=71 Win=137 Len=14 TSval=609769 TSecr=1507759
134	51264 > tetrinet [ACK] Seq=71 Ack=85 Win=129 Len=0 TSval=1508022 TSecr=609769
140	51264 > tetrinet [PSH, ACK] Seq=71 Ack=85 Win=129 Len=14 TSval=1508071 TSecr=609769
141	tetrinet > 51264 [ACK] Seq=85 Ack=85 Win=137 Len=0 TSval=609819 TSecr=1508071
156	tetrinet > 51264 [PSH, ACK] Seq=85 Ack=85 Win=137 Len=14 TSval=610030 TSecr=1508071
160	51264 > tetrinet [ACK] Seq=85 Ack=99 Win=129 Len=0 TSval=1508284 TSecr=610030
166	51264 > tetrinet [PSH, ACK] Seq=85 Ack=99 Win=129 Len=14 TSval=1508391 TSecr=610030
167	tetrinet > 51264 [ACK] Seq=99 Ack=99 Win=137 Len=0 TSval=610139 TSecr=1508391
178	tetrinet > 51264 [PSH, ACK] Seq=99 Ack=99 Win=137 Len=14 TSval=610293 TSecr=1508391
181	51264 > tetrinet [ACK] Seq=99 Ack=113 Win=129 Len=0 TSval=1508547 TSecr=610293
189	51264 > tetrinet [PSH, ACK] Seq=99 Ack=113 Win=129 Len=14 TSval=1508689 TSecr=610293
190	tetrinet > 51264 [ACK] Seq=113 Ack=113 Win=137 Len=0 TSval=610437 TSecr=1508689
198	tetrinet > 51264 [PSH, ACK] Seq=113 Ack=113 Win=137 Len=14 TSval=610556 TSecr=1508689
201	51264 > tetrinet [ACK] Seq=113 Ack=127 Win=129 Len=0 TSval=1508810 TSecr=610556
210	51264 > tetrinet [PSH, ACK] Seq=113 Ack=127 Win=129 Len=14 TSval=1508977 TSecr=610556
211	tetrinet > 51264 [ACK] Seq=127 Ack=127 Win=137 Len=0 TSval=610725 TSecr=1508977
221	tetrinet > 51264 [PSH, ACK] Seq=127 Ack=127 Win=137 Len=14 TSval=610821 TSecr=1508977
222	51264 > tetrinet [ACK] Seq=127 Ack=141 Win=129 Len=0 TSval=1509074 TSecr=610821
239	51264 > tetrinet [PSH, ACK] Seq=127 Ack=141 Win=129 Len=14 TSval=1509271 TSecr=610821
240	tetrinet > 51264 [ACK] Seq=141 Ack=141 Win=137 Len=0 TSval=611040 TSecr=1509271
246	tetrinet > 51264 [PSH, ACK] Seq=141 Ack=141 Win=137 Len=14 TSval=611093 TSecr=1509271 [TCP Retransmission] tetrinet > 51264 [PSH, ACK] Seq=141 Ack=141 Win=137 Len=14 TSval=611159
253	TSecr=1509271
254	51264 > tetrinet [ACK] Seq=141 Ack=155 Win=129 Len=0 TSval=1509412 TSecr=611159
265	51264 > tetrinet [PSH, ACK] Seq=141 Ack=155 Win=129 Len=14 TSval=1509550 TSecr=611159
266	tetrinet > 51264 [ACK] Seq=155 Ack=155 Win=137 Len=0 TSval=611308 TSecr=1509550
268	tetrinet > 51264 [PSH, ACK] Seq=155 Ack=155 Win=137 Len=269 TSval=611344 TSecr=1509550
269	51264 > tetrinet [ACK] Seq=155 Ack=424 Win=138 Len=0 TSval=1509598 TSecr=611344
270	tetrinet > 51264 [PSH, ACK] Seq=424 Ack=155 Win=137 Len=349 TSval=611345 TSecr=1509598 [TCP Retransmission] tetrinet > 51264 [PSH, ACK] Seq=424 Ack=155 Win=137 Len=349 TSval=611407
276	TSecr=1509598
277	51264 > tetrinet [ACK] Seq=155 Ack=773 Win=146 Len=0 TSval=1509660 TSecr=611407
265	[TCP Retransmission] tetrinet > 36138 [PSH, ACK] Seq=169 Ack=155 Win=120 Len=8 TSval=174003

TSecr=1070285
270 36138 > tetrinet [PSH, ACK] Seq=155 Ack=169 Win=71 Len=14 TSval=1072244 TSecr=172051
271 tetrinet > 36138 [ACK] Seq=177 Ack=169 Win=120 Len=0 TSval=174108 TSecr=1072244
274 [TCP Retransmission] 36138 > tetrinet [PSH, ACK] Seq=155 Ack=169 Win=71 Len=14 TSval=1072452
TSecr=172051
275 [TCP Dup ACK 271#1] tetrinet > 36138 [ACK] Seq=177 Ack=169 Win=120 Len=0 TSval=174235 TSecr=1072452
SLE=155 SRE=169
276 [TCP Retransmission] tetrinet > 36138 [PSH, ACK] Seq=169 Ack=169 Win=120 Len=8 TSval=174364
TSecr=1072452
287 [TCP Retransmission] tetrinet > 36138 [PSH, ACK] Seq=169 Ack=169 Win=120 Len=8 TSval=175086
TSecr=1072452
303 [TCP Retransmission] tetrinet > 36138 [PSH, ACK] Seq=169 Ack=169 Win=120 Len=8 TSval=176528
TSecr=1072452
305 36138 > tetrinet [ACK] Seq=169 Ack=177 Win=71 Len=0 TSval=1074763 TSecr=176528



3. Sampel Data *Horizontal Handover Delay* dan *Packet loss*

No.	Time	Protocol	Length
394	1.650716	TCP	100
395	1.650745	TCP	86
498	2.056934	TCP	100
499	2.059377	TCP	86
797	3.207454	TCP	100
798	3.207525	TCP	86
903	3.604808	TCP	100
904	3.606512	TCP	86
1311	5.189239	TCP	100
1312	5.189305	TCP	86
1464	5.776325	TCP	100
1465	5.777954	TCP	86
1882	7.354667	TCP	100
1883	7.354734	TCP	86
2015	7.858796	TCP	100
2018	7.869525	TCP	86
2272	8.791909	TCP	100
2273	8.791936	TCP	86
2444	9.433529	TCP	100
2446	9.435612	TCP	86
2650	10.233132	TCP	100
2651	10.233177	TCP	86
2910	11.219085	TCP	100
2913	11.222834	TCP	86
3061	11.765027	TCP	100
3062	11.76508	TCP	86
3338	12.78362	TCP	100
3345	12.818617	TCP	86
3462	13.259925	TCP	100
3463	13.259987	TCP	86
3789	14.543294	TCP	134
3790	14.545021	TCP	86
3919	15.039196	TCP	100
3920	15.039263	TCP	86
4241	16.288376	TCP	100
4243	16.290048	TCP	86
4378	16.806812	TCP	100
4379	16.806862	TCP	86
4920	18.802629	TCP	100
4921	18.802666	TCP	86
4930	18.830961	TCP	100
4932	18.834214	TCP	86
5473	20.830266	TCP	100
5475	20.832487	TCP	86
5478	20.843788	TCP	100
5488	20.881105	TCP	86
6014	22.919417	TCP	100
6015	22.919461	TCP	86
6107	23.262476	TCP	100
6117	23.304223	TCP	86
6520	24.807166	TCP	100
6521	24.808866	TCP	86
6593	25.07687	TCP	100
6594	25.076933	TCP	86
7097	26.942734	TCP	100
7098	26.942791	TCP	86
7116	26.992832	TCP	100

7117	26.994549	TCP	86
7549	28.695647	TCP	100
7550	28.695712	TCP	86
7663	29.133535	TCP	100
7664	29.135277	TCP	86
8159	31.114826	TCP	94
8160	31.119179	TCP	86

No. Info

54886 > tetrinet [PSH, ACK] Seq=1 Ack=1 Win=339 Len=14 TSval=389672

394 TSecr=1529961

395 tetrinet > 54886 [ACK] Seq=1 Ack=15 Win=1831 Len=0 TSval=1530603 TSecr=389672

tetrinet > 54886 [PSH, ACK] Seq=1 Ack=15 Win=1831 Len=14 TSval=1530704

498 TSecr=389672

499 54886 > tetrinet [ACK] Seq=15 Ack=15 Win=339 Len=0 TSval=389775 TSecr=1530704

54886 > tetrinet [PSH, ACK] Seq=15 Ack=15 Win=339 Len=14 TSval=390062

797 TSecr=1530704

798 tetrinet > 54886 [ACK] Seq=15 Ack=29 Win=1831 Len=0 TSval=1530992 TSecr=390062

tetrinet > 54886 [PSH, ACK] Seq=15 Ack=29 Win=1831 Len=14 TSval=1531091

903 TSecr=390062

904 54886 > tetrinet [ACK] Seq=29 Ack=29 Win=339 Len=0 TSval=390161 TSecr=1531091

54886 > tetrinet [PSH, ACK] Seq=29 Ack=29 Win=339 Len=14 TSval=390557

1311 TSecr=1531091

1312 tetrinet > 54886 [ACK] Seq=29 Ack=43 Win=1831 Len=0 TSval=1531488 TSecr=390557

tetrinet > 54886 [PSH, ACK] Seq=29 Ack=43 Win=1831 Len=14 TSval=1531634

1464 TSecr=390557

1465 54886 > tetrinet [ACK] Seq=43 Ack=43 Win=339 Len=0 TSval=390704 TSecr=1531634

54886 > tetrinet [PSH, ACK] Seq=43 Ack=43 Win=339 Len=14 TSval=391098

1882 TSecr=1531634

1883 tetrinet > 54886 [ACK] Seq=43 Ack=57 Win=1831 Len=0 TSval=1532029 TSecr=391098

tetrinet > 54886 [PSH, ACK] Seq=43 Ack=57 Win=1831 Len=14 TSval=1532155

2015 TSecr=391098

2018 54886 > tetrinet [ACK] Seq=57 Ack=57 Win=339 Len=0 TSval=391225 TSecr=1532155

54886 > tetrinet [PSH, ACK] Seq=57 Ack=57 Win=339 Len=14 TSval=391456

2272 TSecr=1532155

2273 tetrinet > 54886 [ACK] Seq=57 Ack=71 Win=1831 Len=0 TSval=1532388 TSecr=391456

tetrinet > 54886 [PSH, ACK] Seq=57 Ack=71 Win=1831 Len=14 TSval=1532549

2444 TSecr=391456

2446 54886 > tetrinet [ACK] Seq=71 Ack=71 Win=339 Len=0 TSval=391619 TSecr=1532549

54886 > tetrinet [PSH, ACK] Seq=71 Ack=71 Win=339 Len=14 TSval=391818

2650 TSecr=1532549

2651 tetrinet > 54886 [ACK] Seq=71 Ack=85 Win=1831 Len=0 TSval=1532749 TSecr=391818

tetrinet > 54886 [PSH, ACK] Seq=71 Ack=85 Win=1831 Len=14 TSval=1532995

2910 TSecr=391818

2913 54886 > tetrinet [ACK] Seq=85 Ack=85 Win=339 Len=0 TSval=392065 TSecr=1532995

54886 > tetrinet [PSH, ACK] Seq=85 Ack=85 Win=339 Len=14 TSval=392201

3061 TSecr=1532995

3062 tetrinet > 54886 [ACK] Seq=85 Ack=99 Win=1831 Len=0 TSval=1533132 TSecr=392201

tetrinet > 54886 [PSH, ACK] Seq=85 Ack=99 Win=1831 Len=14 TSval=1533386

3338 TSecr=392201

3345 54886 > tetrinet [ACK] Seq=99 Ack=99 Win=339 Len=0 TSval=392464 TSecr=1533386

54886 > tetrinet [PSH, ACK] Seq=99 Ack=99 Win=339 Len=14 TSval=392575

3462 TSecr=1533386

3463 tetrinet > 54886 [ACK] Seq=99 Ack=113 Win=1831 Len=0 TSval=1533505 TSecr=392575

tetrinet > 54886 [PSH, ACK] Seq=99 Ack=113 Win=1831 Len=48 TSval=1533826

3789 TSecr=392575

3790 54886 > tetrinet [ACK] Seq=113 Ack=147 Win=339 Len=0 TSval=392896 TSecr=1533826

54886 > tetrinet [PSH, ACK] Seq=113 Ack=147 Win=339 Len=14 TSval=393020

3919 TSecr=1533826

tetrinet > 54886 [ACK] Seq=147 Ack=127 Win=1831 Len=0 TSval=1533950

3920 TSecr=393020

```

tetrinet > 54886 [PSH, ACK] Seq=147 Ack=127 Win=1831 Len=14 TSval=1534262
4241 TSecr=393020
4243 54886 > tetrinet [ACK] Seq=127 Ack=161 Win=339 Len=0 TSval=393332 TSecr=1534262
54886 > tetrinet [PSH, ACK] Seq=127 Ack=161 Win=339 Len=14 TSval=393462
4378 TSecr=1534262
tetrinet > 54886 [ACK] Seq=161 Ack=141 Win=1831 Len=0 TSval=1534392
4379 TSecr=393462
54886 > tetrinet [PSH, ACK] Seq=141 Ack=161 Win=339 Len=14 TSval=393960
4920 TSecr=1534392
tetrinet > 54886 [ACK] Seq=161 Ack=155 Win=1831 Len=0 TSval=1534891
4921 TSecr=393960
tetrinet > 54886 [PSH, ACK] Seq=161 Ack=155 Win=1831 Len=14 TSval=1534898
4930 TSecr=393960
4932 54886 > tetrinet [ACK] Seq=155 Ack=175 Win=339 Len=0 TSval=393968 TSecr=1534898
tetrinet > 54886 [PSH, ACK] Seq=175 Ack=155 Win=1831 Len=14 TSval=1535398
5473 TSecr=393968
5475 54886 > tetrinet [ACK] Seq=155 Ack=189 Win=339 Len=0 TSval=394468 TSecr=1535398
54886 > tetrinet [PSH, ACK] Seq=155 Ack=189 Win=339 Len=14 TSval=394471
5478 TSecr=1535398
tetrinet > 54886 [ACK] Seq=189 Ack=169 Win=1831 Len=0 TSval=1535411
5488 TSecr=394471
54886 > tetrinet [PSH, ACK] Seq=169 Ack=189 Win=339 Len=14 TSval=394990
6014 TSecr=1535411
tetrinet > 54886 [ACK] Seq=189 Ack=183 Win=1831 Len=0 TSval=1535920
6015 TSecr=394990
tetrinet > 54886 [PSH, ACK] Seq=189 Ack=183 Win=1831 Len=14 TSval=1536006
6107 TSecr=394990
6117 54886 > tetrinet [ACK] Seq=183 Ack=203 Win=339 Len=0 TSval=395086 TSecr=1536006
tetrinet > 54886 [PSH, ACK] Seq=203 Ack=183 Win=1831 Len=14 TSval=1536392
6520 TSecr=395086
6521 54886 > tetrinet [ACK] Seq=183 Ack=217 Win=339 Len=0 TSval=395462 TSecr=1536392
54886 > tetrinet [PSH, ACK] Seq=183 Ack=217 Win=339 Len=14 TSval=395529
6593 TSecr=1536392
tetrinet > 54886 [ACK] Seq=217 Ack=197 Win=1831 Len=0 TSval=1536459
6594 TSecr=395529
54886 > tetrinet [PSH, ACK] Seq=197 Ack=217 Win=339 Len=14 TSval=395996
7097 TSecr=1536459
tetrinet > 54886 [ACK] Seq=217 Ack=211 Win=1831 Len=0 TSval=1536926
7098 TSecr=395996
tetrinet > 54886 [PSH, ACK] Seq=217 Ack=211 Win=1831 Len=14 TSval=1536938
7116 TSecr=395996
7117 54886 > tetrinet [ACK] Seq=211 Ack=231 Win=339 Len=0 TSval=396009 TSecr=1536938
54886 > tetrinet [PSH, ACK] Seq=211 Ack=231 Win=339 Len=14 TSval=396434
7549 TSecr=1536938
tetrinet > 54886 [ACK] Seq=231 Ack=225 Win=1831 Len=0 TSval=1537364
7550 TSecr=396434
tetrinet > 54886 [PSH, ACK] Seq=231 Ack=225 Win=1831 Len=14 TSval=1537474
7663 TSecr=396434
7664 54886 > tetrinet [ACK] Seq=225 Ack=245 Win=339 Len=0 TSval=396544 TSecr=1537474
tetrinet > 54886 [PSH, ACK] Seq=245 Ack=225 Win=1831 Len=8 TSval=1537969
8159 TSecr=396544
8160 54886 > tetrinet [ACK] Seq=225 Ack=253 Win=339 Len=0 TSval=397040 TSecr=1537969

```

4. Sampel Data *Horizontal Handover Transfer rate*

No.	Time	Protocol	Length
16	0.833975	TCP	100
18	0.841432	TCP	86
24	1.062633	TCP	100
25	1.062697	TCP	86
38	1.885779	TCP	100
40	1.892393	TCP	86
47	2.181372	TCP	100
48	2.181431	TCP	86
62	2.898636	TCP	100
64	2.903785	TCP	86
72	3.236064	TCP	100
73	3.236114	TCP	86
86	3.946415	TCP	100
88	3.951862	TCP	86
96	4.309922	TCP	100
97	4.309958	TCP	86
110	5.070417	TCP	100
112	5.075173	TCP	86
119	5.415273	TCP	100
120	5.415332	TCP	86
129	6.081362	TCP	100
132	6.132035	TCP	86
138	6.51955	TCP	100
139	6.519624	TCP	86
155	7.463705	TCP	100
156	7.467818	TCP	86
161	7.598352	TCP	100
162	7.598399	TCP	86
176	8.561509	TCP	100
178	8.571233	TCP	86
180	8.602612	TCP	100
181	8.602667	TCP	86
202	9.673981	TCP	100
205	9.714223	TCP	100
206	9.714298	TCP	86
221	10.71887	TCP	100
224	10.765675	TCP	86
226	10.846227	TCP	100
227	10.846289	TCP	86
250	12.087236	TCP	100
251	12.087285	TCP	86
253	12.136076	TCP	100
254	12.140016	TCP	86
272	13.009546	TCP	355
273	13.009593	TCP	86
274	13.012751	TCP	99
275	13.012767	TCP	86
276	13.01527	TCP	94
277	13.034866	TCP	86
278	13.034902	TCP	420
279	13.045267	TCP	86

No.	Info
16	tetrinet > 53454 [PSH, ACK] Seq=1 Ack=1 Win=1027 Len=14 TSval=1241016 TSecr=99564
18	53454 > tetrinet [ACK] Seq=1 Ack=15 Win=87 Len=0 TSval=100071 TSecr=1241016
24	53454 > tetrinet [PSH, ACK] Seq=1 Ack=15 Win=87 Len=14 TSval=100126 TSecr=1241016
25	tetrinet > 53454 [ACK] Seq=15 Ack=15 Win=1027 Len=0 TSval=1241073 TSecr=100126

```
38 tetrinet > 53454 [PSH, ACK] Seq=15 Ack=15 Win=1027 Len=14 TSval=1241279 TSecr=100126
40 53454 > tetrinet [ACK] Seq=15 Ack=29 Win=87 Len=0 TSval=100334 TSecr=1241279
47 53454 > tetrinet [PSH, ACK] Seq=15 Ack=29 Win=87 Len=14 TSval=100406 TSecr=1241279
48 tetrinet > 53454 [ACK] Seq=29 Ack=29 Win=1027 Len=0 TSval=1241353 TSecr=100406
62 tetrinet > 53454 [PSH, ACK] Seq=29 Ack=29 Win=1027 Len=14 TSval=1241532 TSecr=100406
64 53454 > tetrinet [ACK] Seq=29 Ack=43 Win=87 Len=0 TSval=100587 TSecr=1241532
72 53454 > tetrinet [PSH, ACK] Seq=29 Ack=43 Win=87 Len=14 TSval=100670 TSecr=1241532
73 tetrinet > 53454 [ACK] Seq=43 Ack=43 Win=1027 Len=0 TSval=1241616 TSecr=100670
86 tetrinet > 53454 [PSH, ACK] Seq=43 Ack=43 Win=1027 Len=14 TSval=1241794 TSecr=100670
88 53454 > tetrinet [ACK] Seq=43 Ack=57 Win=87 Len=0 TSval=100849 TSecr=1241794
96 53454 > tetrinet [PSH, ACK] Seq=43 Ack=57 Win=87 Len=14 TSval=100938 TSecr=1241794
97 tetrinet > 53454 [ACK] Seq=57 Ack=57 Win=1027 Len=0 TSval=1241885 TSecr=100938
110 tetrinet > 53454 [PSH, ACK] Seq=57 Ack=57 Win=1027 Len=14 TSval=1242075 TSecr=100938
112 53454 > tetrinet [ACK] Seq=57 Ack=71 Win=87 Len=0 TSval=101130 TSecr=1242075
119 53454 > tetrinet [PSH, ACK] Seq=57 Ack=71 Win=87 Len=14 TSval=101215 TSecr=1242075
120 tetrinet > 53454 [ACK] Seq=71 Ack=71 Win=1027 Len=0 TSval=1242161 TSecr=101215
129 tetrinet > 53454 [PSH, ACK] Seq=71 Ack=71 Win=1027 Len=14 TSval=1242328 TSecr=101215
132 53454 > tetrinet [ACK] Seq=71 Ack=85 Win=87 Len=0 TSval=101394 TSecr=1242328
138 53454 > tetrinet [PSH, ACK] Seq=71 Ack=85 Win=87 Len=14 TSval=101491 TSecr=1242328
139 tetrinet > 53454 [ACK] Seq=85 Ack=85 Win=1027 Len=0 TSval=1242437 TSecr=101491
155 tetrinet > 53454 [PSH, ACK] Seq=85 Ack=85 Win=1027 Len=14 TSval=1242673 TSecr=101491
156 53454 > tetrinet [ACK] Seq=85 Ack=99 Win=87 Len=0 TSval=101728 TSecr=1242673
161 53454 > tetrinet [PSH, ACK] Seq=85 Ack=99 Win=87 Len=14 TSval=101761 TSecr=1242673
162 tetrinet > 53454 [ACK] Seq=99 Ack=99 Win=1027 Len=0 TSval=1242707 TSecr=101761
176 tetrinet > 53454 [PSH, ACK] Seq=99 Ack=99 Win=1027 Len=14 TSval=1242948 TSecr=101761
178 53454 > tetrinet [ACK] Seq=99 Ack=113 Win=87 Len=0 TSval=102003 TSecr=1242948
180 53454 > tetrinet [PSH, ACK] Seq=99 Ack=113 Win=87 Len=14 TSval=102012 TSecr=1242948
181 tetrinet > 53454 [ACK] Seq=113 Ack=113 Win=1027 Len=0 TSval=1242958 TSecr=102012
202 tetrinet > 53454 [PSH, ACK] Seq=113 Ack=113 Win=1027 Len=14 TSval=1243226 TSecr=102012
205 53454 > tetrinet [PSH, ACK] Seq=113 Ack=127 Win=87 Len=14 TSval=102290 TSecr=1243226
206 tetrinet > 53454 [ACK] Seq=127 Ack=127 Win=1027 Len=0 TSval=1243236 TSecr=102290
221 tetrinet > 53454 [PSH, ACK] Seq=127 Ack=127 Win=1027 Len=14 TSval=1243487 TSecr=102290
224 53454 > tetrinet [ACK] Seq=127 Ack=141 Win=87 Len=0 TSval=102552 TSecr=1243487
226 53454 > tetrinet [PSH, ACK] Seq=127 Ack=141 Win=87 Len=14 TSval=102573 TSecr=1243487
227 tetrinet > 53454 [ACK] Seq=141 Ack=141 Win=1027 Len=0 TSval=1243519 TSecr=102573
250 53454 > tetrinet [PSH, ACK] Seq=141 Ack=141 Win=87 Len=14 TSval=102858 TSecr=1243519
251 tetrinet > 53454 [ACK] Seq=141 Ack=155 Win=1027 Len=0 TSval=1243829 TSecr=102858
253 tetrinet > 53454 [PSH, ACK] Seq=141 Ack=155 Win=1027 Len=14 TSval=1243841 TSecr=102858
254 53454 > tetrinet [ACK] Seq=155 Ack=155 Win=87 Len=0 TSval=102896 TSecr=1243841
272 53454 > tetrinet [PSH, ACK] Seq=155 Ack=155 Win=87 Len=269 TSval=103108 TSecr=1243841
273 tetrinet > 53454 [ACK] Seq=155 Ack=424 Win=1094 Len=0 TSval=1244060 TSecr=103108
274 53454 > tetrinet [PSH, ACK] Seq=424 Ack=155 Win=87 Len=13 TSval=103114 TSecr=1244060
275 tetrinet > 53454 [ACK] Seq=155 Ack=437 Win=1094 Len=0 TSval=1244061 TSecr=103114
276 tetrinet > 53454 [PSH, ACK] Seq=155 Ack=437 Win=1094 Len=8 TSval=1244061 TSecr=103114
277 53454 > tetrinet [ACK] Seq=437 Ack=163 Win=87 Len=0 TSval=103118 TSecr=1244061
278 tetrinet > 53454 [PSH, ACK] Seq=163 Ack=437 Win=1094 Len=334 TSval=1244066 TSecr=103118
279 53454 > tetrinet [ACK] Seq=437 Ack=497 Win=96 Len=0 TSval=103121 TSecr=1244066
```

5. Sampel Data *Access Point* sama *Delay* dan *Packet loss*

No.	Time	Protocol	Length
27	1.543747	TCP	100
28	1.580263	TCP	86
32	2.61699	TCP	100
33	2.631707	TCP	86
47	3.180813	TCP	100
48	3.180893	TCP	86
80	4.723488	TCP	100
81	4.727915	TCP	86
97	5.451135	TCP	100
98	5.451227	TCP	86
119	6.53607	TCP	100
120	6.540062	TCP	86
132	7.253795	TCP	100
133	7.25389	TCP	86
152	8.055363	TCP	100
153	8.057231	TCP	86
169	8.820337	TCP	100
170	8.820428	TCP	86
190	9.739288	TCP	100
191	9.744798	TCP	86
200	10.308671	TCP	100
201	10.308758	TCP	86
225	11.345757	TCP	100
226	11.347747	TCP	86
238	11.850503	TCP	100
239	11.850598	TCP	86
270	13.008446	TCP	100
272	13.010919	TCP	86
281	13.43635	TCP	100
282	13.436437	TCP	86
309	14.546064	TCP	100
310	14.549903	TCP	86
318	15.114971	TCP	100
319	15.115062	TCP	86
346	16.146249	TCP	100
347	16.148827	TCP	86
224	10.765675	TCP	86
226	10.846227	TCP	100
227	10.846289	TCP	86
250	12.087236	TCP	100
251	12.087285	TCP	86
253	12.136076	TCP	100
254	12.140016	TCP	86
272	13.009546	TCP	355
273	13.009593	TCP	86
274	13.012751	TCP	99
275	13.012767	TCP	86
276	13.01527	TCP	94
277	13.034866	TCP	86
278	13.034902	TCP	420
279	13.045267	TCP	86

No.	Info
27	46380 > tetrinet [PSH, ACK] Seq=1 Ack=1 Win=390 Len=14 TSval=865680 TSecr=987000
28	tetrinet > 46380 [ACK] Seq=1 Ack=15 Win=280 Len=0 TSval=987774 TSecr=865680
32	tetrinet > 46380 [PSH, ACK] Seq=1 Ack=15 Win=280 Len=14 TSval=988033 TSecr=865680
33	46380 > tetrinet [ACK] Seq=15 Ack=15 Win=390 Len=0 TSval=865960 TSecr=988033

47 46380 > tetrinet [PSH, ACK] Seq=15 Ack=15 Win=390 Len=14 TSval=866098 TSecr=988033
48 tetrinet > 46380 [ACK] Seq=15 Ack=29 Win=280 Len=0 TSval=988174 TSecr=866098
80 tetrinet > 46380 [PSH, ACK] Seq=15 Ack=29 Win=280 Len=14 TSval=988559 TSecr=866098
81 46380 > tetrinet [ACK] Seq=29 Ack=29 Win=390 Len=0 TSval=866485 TSecr=988559
97 46380 > tetrinet [PSH, ACK] Seq=29 Ack=29 Win=390 Len=14 TSval=866666 TSecr=988559
98 tetrinet > 46380 [ACK] Seq=29 Ack=43 Win=280 Len=0 TSval=988741 TSecr=866666
119 tetrinet > 46380 [PSH, ACK] Seq=29 Ack=43 Win=280 Len=14 TSval=989012 TSecr=866666
120 46380 > tetrinet [ACK] Seq=43 Ack=43 Win=390 Len=0 TSval=866938 TSecr=989012
132 46380 > tetrinet [PSH, ACK] Seq=43 Ack=43 Win=390 Len=14 TSval=867117 TSecr=989012
133 tetrinet > 46380 [ACK] Seq=43 Ack=57 Win=280 Len=0 TSval=989192 TSecr=867117
152 tetrinet > 46380 [PSH, ACK] Seq=43 Ack=57 Win=280 Len=14 TSval=989392 TSecr=867117
153 46380 > tetrinet [ACK] Seq=57 Ack=57 Win=390 Len=0 TSval=867318 TSecr=989392
169 46380 > tetrinet [PSH, ACK] Seq=57 Ack=57 Win=390 Len=14 TSval=867509 TSecr=989392
170 tetrinet > 46380 [ACK] Seq=57 Ack=71 Win=280 Len=0 TSval=989584 TSecr=867509
190 tetrinet > 46380 [PSH, ACK] Seq=57 Ack=71 Win=280 Len=14 TSval=989813 TSecr=867509
191 46380 > tetrinet [ACK] Seq=71 Ack=71 Win=390 Len=0 TSval=867740 TSecr=989813
200 46380 > tetrinet [PSH, ACK] Seq=71 Ack=71 Win=390 Len=14 TSval=867881 TSecr=989813
201 tetrinet > 46380 [ACK] Seq=71 Ack=85 Win=280 Len=0 TSval=989956 TSecr=867881
225 tetrinet > 46380 [PSH, ACK] Seq=71 Ack=85 Win=280 Len=14 TSval=990215 TSecr=867881
226 46380 > tetrinet [ACK] Seq=85 Ack=85 Win=390 Len=0 TSval=868140 TSecr=990215
238 46380 > tetrinet [PSH, ACK] Seq=85 Ack=85 Win=390 Len=14 TSval=868265 TSecr=990215
239 tetrinet > 46380 [ACK] Seq=85 Ack=99 Win=280 Len=0 TSval=990341 TSecr=868265
270 tetrinet > 46380 [PSH, ACK] Seq=85 Ack=99 Win=280 Len=14 TSval=990631 TSecr=868265
272 46380 > tetrinet [ACK] Seq=99 Ack=99 Win=390 Len=0 TSval=868557 TSecr=990631
281 46380 > tetrinet [PSH, ACK] Seq=99 Ack=99 Win=390 Len=14 TSval=868663 TSecr=990631
282 tetrinet > 46380 [ACK] Seq=99 Ack=113 Win=280 Len=0 TSval=990738 TSecr=868663
309 tetrinet > 46380 [PSH, ACK] Seq=99 Ack=113 Win=280 Len=14 TSval=991015 TSecr=868663
310 46380 > tetrinet [ACK] Seq=113 Ack=113 Win=390 Len=0 TSval=868941 TSecr=991015
318 46380 > tetrinet [PSH, ACK] Seq=113 Ack=113 Win=390 Len=14 TSval=869082 TSecr=991015
319 tetrinet > 46380 [ACK] Seq=113 Ack=127 Win=280 Len=0 TSval=991157 TSecr=869082
346 tetrinet > 46380 [PSH, ACK] Seq=113 Ack=127 Win=280 Len=14 TSval=991415 TSecr=869082
347 46380 > tetrinet [ACK] Seq=127 Ack=127 Win=390 Len=0 TSval=869341 TSecr=991415
224 53454 > tetrinet [ACK] Seq=127 Ack=141 Win=87 Len=0 TSval=102552 TSecr=1243487
226 53454 > tetrinet [PSH, ACK] Seq=127 Ack=141 Win=87 Len=14 TSval=102573 TSecr=1243487
227 tetrinet > 53454 [ACK] Seq=141 Ack=141 Win=1027 Len=0 TSval=1243519 TSecr=102573
250 53454 > tetrinet [PSH, ACK] Seq=141 Ack=141 Win=87 Len=14 TSval=102858 TSecr=1243519
251 tetrinet > 53454 [ACK] Seq=141 Ack=155 Win=1027 Len=0 TSval=1243829 TSecr=102858
253 tetrinet > 53454 [PSH, ACK] Seq=141 Ack=155 Win=1027 Len=14 TSval=1243841 TSecr=102858
254 53454 > tetrinet [ACK] Seq=155 Ack=155 Win=87 Len=0 TSval=102896 TSecr=1243841
272 53454 > tetrinet [PSH, ACK] Seq=155 Ack=155 Win=87 Len=269 TSval=103108 TSecr=1243841
273 tetrinet > 53454 [ACK] Seq=155 Ack=424 Win=1094 Len=0 TSval=1244060 TSecr=103108
274 53454 > tetrinet [PSH, ACK] Seq=424 Ack=155 Win=87 Len=13 TSval=103114 TSecr=1244060
275 tetrinet > 53454 [ACK] Seq=155 Ack=437 Win=1094 Len=0 TSval=1244061 TSecr=103114
276 tetrinet > 53454 [PSH, ACK] Seq=155 Ack=437 Win=1094 Len=8 TSval=1244061 TSecr=103114
277 53454 > tetrinet [ACK] Seq=437 Ack=163 Win=87 Len=0 TSval=103118 TSecr=1244061
278 tetrinet > 53454 [PSH, ACK] Seq=163 Ack=437 Win=1094 Len=334 TSval=1244066 TSecr=103118
279 53454 > tetrinet [ACK] Seq=437 Ack=497 Win=96 Len=0 TSval=103121 TSecr=1244066

6. Sampel Data *Access Point* sama *Transfer rate*

No.	Time	Protocol	Length
5	0.096311	TCP	86
42	2.346417	TCP	100
43	2.384258	TCP	86
44	2.83799	TCP	100
47	2.84869	TCP	86
53	3.193502	ICMPv6	134
58	3.383775	TCP	100
59	3.383862	TCP	86
60	3.424218	TCP	257
61	3.424322	TCP	98
69	3.936437	TCP	100
70	4.148463	TCP	100
71	4.454657	TCP	100
72	4.454752	TCP	86
80	5.060973	TCP	100
81	5.068351	TCP	86
91	5.570703	TCP	100
92	5.570779	TCP	86
104	6.153526	TCP	100
105	6.164475	TCP	86
114	6.695077	TCP	100
115	6.695167	TCP	86
128	7.155085	TCP	100
129	7.166122	TCP	86
140	7.805537	TCP	100
141	7.805629	TCP	86
152	8.257725	TCP	100
153	8.259586	TCP	86
165	8.834456	TCP	100
166	8.834545	TCP	86
178	9.390456	TCP	100
184	9.688463	TCP	100
185	9.695678	TCP	98
192	9.955194	TCP	100
193	9.955271	TCP	86
203	10.396074	TCP	100
204	10.403093	TCP	86
217	10.990279	TCP	100
218	10.990368	TCP	86
229	11.422658	TCP	100
230	11.433479	TCP	86
243	11.992366	TCP	355
244	11.992454	TCP	86
245	11.994352	TCP	99
246	11.994434	TCP	86
247	11.99488	TCP	94
248	11.99673	TCP	86
249	11.996756	TCP	367
250	11.998806	TCP	86
251	11.998883	TCP	144
252	12.000716	TCP	86

No.	Info
5	tetrinet > 50185 [FIN, ACK] Seq=1 Ack=1 Win=129 Len=0 TSval=857952 TSecr=727818
42	46380 > tetrinet [PSH, ACK] Seq=1 Ack=1 Win=87 Len=14 TSval=736433 TSecr=857675
43	tetrinet > 46380 [ACK] Seq=1 Ack=15 Win=129 Len=0 TSval=858524 TSecr=736433
44	tetrinet > 46380 [PSH, ACK] Seq=1 Ack=15 Win=129 Len=14 TSval=858637 TSecr=736433

```

47 46380 > tetrinet [ACK] Seq=15 Ack=15 Win=87 Len=0 TSval=736559 TSecr=858637
53 Destination Unreachable (Address unreachable)
58 46380 > tetrinet [PSH, ACK] Seq=15 Ack=15 Win=87 Len=14 TSval=736693 TSecr=858637
59 tetrinet > 46380 [ACK] Seq=15 Ack=29 Win=129 Len=0 TSval=858773 TSecr=736693
60 50185 > tetrinet [FIN, PSH, ACK] Seq=4294967125 Ack=1 Win=81 Len=171 TSval=736702 TSecr=820014
[TCP Dup ACK 5#1] tetrinet > 50185 [ACK] Seq=2 Ack=1 Win=129 Len=0 TSval=858784 TSecr=736702
61 SLE=4294967125 SRE=1
69 tetrinet > 46380 [PSH, ACK] Seq=15 Ack=29 Win=129 Len=14 TSval=858912 TSecr=736693
70 [TCP Retransmission] tetrinet > 46380 [PSH, ACK] Seq=15 Ack=29 Win=129 Len=14 TSval=858965 TSecr=736693
71 46380 > tetrinet [PSH, ACK] Seq=29 Ack=29 Win=87 Len=14 TSval=736959 TSecr=858965
72 tetrinet > 46380 [ACK] Seq=29 Ack=43 Win=129 Len=0 TSval=859041 TSecr=736959
80 tetrinet > 46380 [PSH, ACK] Seq=29 Ack=43 Win=129 Len=14 TSval=859193 TSecr=736959
81 46380 > tetrinet [ACK] Seq=43 Ack=43 Win=87 Len=0 TSval=737113 TSecr=859193
91 46380 > tetrinet [PSH, ACK] Seq=43 Ack=43 Win=87 Len=14 TSval=737238 TSecr=859193
92 tetrinet > 46380 [ACK] Seq=43 Ack=57 Win=129 Len=0 TSval=859320 TSecr=737238
104 tetrinet > 46380 [PSH, ACK] Seq=43 Ack=57 Win=129 Len=14 TSval=859466 TSecr=737238
105 46380 > tetrinet [ACK] Seq=57 Ack=57 Win=87 Len=0 TSval=737386 TSecr=859466
114 46380 > tetrinet [PSH, ACK] Seq=57 Ack=57 Win=87 Len=14 TSval=737519 TSecr=859466
115 tetrinet > 46380 [ACK] Seq=57 Ack=71 Win=129 Len=0 TSval=859601 TSecr=737519
128 tetrinet > 46380 [PSH, ACK] Seq=57 Ack=71 Win=129 Len=14 TSval=859716 TSecr=737519
129 46380 > tetrinet [ACK] Seq=71 Ack=71 Win=87 Len=0 TSval=737636 TSecr=859716
140 46380 > tetrinet [PSH, ACK] Seq=71 Ack=71 Win=87 Len=14 TSval=737797 TSecr=859716
141 tetrinet > 46380 [ACK] Seq=71 Ack=85 Win=129 Len=0 TSval=859879 TSecr=737797
152 tetrinet > 46380 [PSH, ACK] Seq=71 Ack=85 Win=129 Len=14 TSval=859992 TSecr=737797
153 46380 > tetrinet [ACK] Seq=85 Ack=85 Win=87 Len=0 TSval=737912 TSecr=859992
165 46380 > tetrinet [PSH, ACK] Seq=85 Ack=85 Win=87 Len=14 TSval=738055 TSecr=859992
166 tetrinet > 46380 [ACK] Seq=85 Ack=99 Win=129 Len=0 TSval=860136 TSecr=738055
178 tetrinet > 46380 [PSH, ACK] Seq=85 Ack=99 Win=129 Len=14 TSval=860275 TSecr=738055
184 [TCP Retransmission] tetrinet > 46380 [PSH, ACK] Seq=85 Ack=99 Win=129 Len=14 TSval=860350 TSecr=738055
185 46380 > tetrinet [ACK] Seq=99 Ack=99 Win=87 Len=0 TSval=738270 TSecr=860350 SLE=85 SRE=99
192 46380 > tetrinet [PSH, ACK] Seq=99 Ack=99 Win=87 Len=14 TSval=738335 TSecr=860350
193 tetrinet > 46380 [ACK] Seq=99 Ack=113 Win=129 Len=0 TSval=860416 TSecr=738335
203 tetrinet > 46380 [PSH, ACK] Seq=99 Ack=113 Win=129 Len=14 TSval=860526 TSecr=738335
204 46380 > tetrinet [ACK] Seq=113 Ack=113 Win=87 Len=0 TSval=738447 TSecr=860526
217 46380 > tetrinet [PSH, ACK] Seq=113 Ack=113 Win=87 Len=14 TSval=738594 TSecr=860526
218 tetrinet > 46380 [ACK] Seq=113 Ack=127 Win=129 Len=0 TSval=860675 TSecr=738594
229 tetrinet > 46380 [PSH, ACK] Seq=113 Ack=127 Win=129 Len=14 TSval=860783 TSecr=738594
230 46380 > tetrinet [ACK] Seq=127 Ack=127 Win=87 Len=0 TSval=738703 TSecr=860783
243 46380 > tetrinet [PSH, ACK] Seq=127 Ack=127 Win=87 Len=269 TSval=738845 TSecr=860783
244 tetrinet > 46380 [ACK] Seq=127 Ack=396 Win=137 Len=0 TSval=860926 TSecr=738845
245 46380 > tetrinet [PSH, ACK] Seq=396 Ack=127 Win=87 Len=13 TSval=738846 TSecr=860926
246 tetrinet > 46380 [ACK] Seq=127 Ack=409 Win=137 Len=0 TSval=860926 TSecr=738846
247 tetrinet > 46380 [PSH, ACK] Seq=127 Ack=409 Win=137 Len=8 TSval=860926 TSecr=738846
248 46380 > tetrinet [ACK] Seq=409 Ack=135 Win=87 Len=0 TSval=738846 TSecr=860926
249 tetrinet > 46380 [PSH, ACK] Seq=135 Ack=409 Win=137 Len=281 TSval=860927 TSecr=738846
250 46380 > tetrinet [ACK] Seq=409 Ack=416 Win=96 Len=0 TSval=738847 TSecr=860927
251 tetrinet > 46380 [PSH, ACK] Seq=416 Ack=409 Win=137 Len=58 TSval=860927 TSecr=738847
252 46380 > tetrinet [ACK] Seq=409 Ack=474 Win=96 Len=0 TSval=738847 TSecr=860927

```