



UNIVERSITAS INDONESIA

**ANALISA PERILAKU TRAFIK TIDAK BERKELAS PADA
PROTOKOL *NEXT STEP IN SIGNALING***

THESIS

**KAMAL DJUNAEDI
0606151330**

$\frac{T}{25126}$

**FAKULTAS TEKNIK
PROGRAM PASCA SARJANA BIDANG ILMU TEKNIK
DEPOK
DESEMBER 2008**



UNIVERSITAS INDONESIA

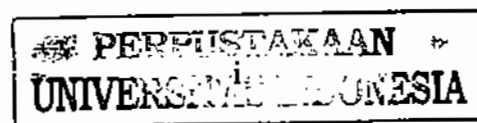
**ANALISA PERILAKU TRAFIK TIDAK BERKELAS PADA
PROTOKOL *NEXT STEP IN SIGNALING***

THESIS

Diajukan sebagai salah satu syarat untuk memperoleh gelar Magister Teknik


**KAMAL DJUNAEDI
0606151330**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
KEKHUSUSAN JARINGAN INFORMASI DAN MULTIMEDIA
DEPOK
DESEMBER 2008**



HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Kamal Djunaedi
NPM : 0606151330
Tanda Tangan : 

Tanggal : 31 Desember 2008

HALAMAN PENGESAHAN

Thesis ini diajukan oleh :
Nama : Kamal Djunaedi
NPM : 0606151330
Program Studi : Teknik Elektro
Judul Tesis : ANALISA PERILAKU TRAFIK TIDAK
BERKELAS PADA PROTOKOL *NEXT STEP IN SIGNALING*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Prof. Dr. Ir. Bagio Budiardjo, MSc (.....)
Penguji : Dr. Ir. Kalamullah Ramli, M.Eng (.....)
Penguji : Dr. Ir. Riri Fitri Sari, MSc, MM (.....)
Penguji : Muhammad Salman, ST., MIT (.....)

Ditetapkan di : Depok
Tanggal : 21 DESEMBER 2008

KATA PENGANTAR/UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan thesis ini. Penulisan thesis ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Magister Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan thesis ini, sangatlah sulit bagi saya untuk menyelesaikan thesis ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Prof. Dr. Ir. Bagio Budiarjo, MSc, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2) Bpk H.Romdhoni (Alm), Ibu Hj. Suharsikah, Ibu Dasinah, Bpk Pendi selaku orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; dan
- (3) sahabat (Bogi, Kely, Bayu, Tezar, Jaswar, Dewi, Amry, Dede, Fauzan, Mba Nur, Rendy, Pa Husni, dll) yang telah banyak membantu saya dalam menyelesaikan thesis ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga thesis ini membawa manfaat bagi pengembangan ilmu.

Depok, 31 Desember 2008

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS
AKHIR UNTUK KEPENTINGAN AKADEMIS**

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

Nama : Kamal Djunaedi
NPM : 0606151330
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik Elektro
Jenis karya : Thesis

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 :

**ANALISA PERILAKU TRAFIK TIDAK BERKELAS PADA PROTOKOL
*NEXT STEP IN SIGNALING***

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 memublikasikan 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 : 31 Desember 2008

Yang menyatakan



(Kamal Djunaedi)

ABSTRAK

Nama : Kamal Djunaedi
Program Studi : Teknik Elektro
Judul : Analisa Perilaku Trafik Tidak Berkelas pada Protokol *Next Step in Signaling*

Pada tahun 2001 IETF membentuk suatu kelompok kerja baru *Next Step in Signaling* (NSIS) untuk menyelidiki arsitektur dan protokol untuk umum dan aplikasi pensinyalan spesifik. NSIS diharapkan dapat mendukung berbagai aplikasi *signaling* yang dapat menginstal dan memanipulasi suatu status atau keadaan dalam suatu jaringan. Pada Thesis ini dilakukan simulasi jaringan, dimana beberapa trafik dijadikan trafik yang tidak berkelas pada protokol NSIS.

Dari hasil simulasi terlihat perilaku dari *video streaming*, ketika dijalankan secara bersamaan dengan HTTP dan FTP, menunjukkan dominasi dan mempengaruhi dari perilaku kedua jenis trafik tersebut. Penggunaan *Hierarchical Token Bucket* (HTB) sebagai *traffic control* pada protokol NSIS memberikan kepastian akan *bandwidth* terhadap kebocoran *bandwidth*.

Kata Kunci: QoS, NSIS, *throughput*, trafik, FTP, *video streaming*, HTTP, dan HTB.

ABSTRACT

Name : Kamal Djunaedi
Study Program : Electrical Engineering
Title : Analisis Behavior Unclass Traffics in Next Step in Signaling Protocol

In the year 2001 IETF Working Group Next Step in Signaling (NSIS) to investigate the architecture protocol generically and specific application signaling. NSIS expected can support various application signaling able to installation and manipulation state in network. These Research writer plans to simulation the network, where some traffic are made unclassified in NSIS protocol.

The result of simulation is present behavior of video streaming, when run concurrently with HTTP and FTP, it's the domination and influence from behavior both types of the traffic. The Hierarchical Token Bucket (HTB) as traffic control at protocol NSIS give certainty would bandwidth to leakage

Keywords: QoS, NSIS, throughput, traffic, FTP, video streaming, HTTP, and HTB

DAFTAR ISI

	Halaman
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR/UCAPAN TERIMA KASIH.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	v
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN	xii
DAFTAR SINGKATAN.....	xiii
BAB I PENDAHULUAN.....	1
1.1 LATAR BELAKANG.....	1
1.2 TUJUAN PENELITIAN.....	2
1.3 PEMBATASAN MASALAH.....	2
1.4 METODOLOGI PENELITIAN.....	2
1.5 SISTEMATIKA PEMBAHASAN.....	3
BAB II DASAR TEORI.....	4
2.1 <i>QUALITY OF SERVICE</i> (QoS).....	4
2.2 <i>RESOURCE RESERVATION PROTOCOL</i> (RSVP).....	4
2.2.1 <i>Integrated Services</i> (IntServ).....	4
2.2.2 <i>Differentiated Services</i> (DiffServ).....	5
2.3 <i>NEXT STEP IN SIGNALING</i> (NSIS).....	5
2.3.1 <i>General Internet Signaling Transport Protocol</i> (GIST).....	6
2.3.2 Aplikasi <i>Signaling QoS</i> pada NSIS	9
2.3.2.1 <i>Model dari Operasi QoS NSLP</i>	9
2.3.2.3 <i>Message Protocol dari QoS NSLP</i>	11
2.3.3 Reservasi pada QoS NSLP.....	12
2.3.4 QSPEC.....	14
2.3.5 QoS Model <i>Controlled-Load Service</i> (QoSM CLS).....	15
2.4 PERBANDINGAN ANTARA RSVP DAN NSIS (QoS) <i>SIGNALING</i>	16
BAB III PEMODELAN SIMULASI JARINGAN.....	19
3.1 ARSITEKTUR APLIKASI.....	19

3.1.1 <i>Traffic Control Interface</i>	19
3.1.2 <i>Resource Management Function (RMF)</i>	20
3.2 IMPLEMENTASI KOMPONEN DAN TOPOLOGI JARINGAN DALAM SIMULASI.....	21
3.2.1 Implementasi Komponen.....	21
3.2.2 Topologi yang digunakan dalam Simulasi.....	22
3.3 PEMBANGUNAN SISTEM.....	22
3.3.1 Instalasi Aplikasi OpenNSIS	22
3.4 PEMBANGUNAN SIMULASI.....	23
3.4.1 Skenario Simulasi	23
3.4 PARAMETER KINERJA.....	26
BAB IV PEMBAHASAN.....	27
4.1 TRAFIK TANPA RESERVASI PADA PROTOKOL NSIS.....	27
4.2 TRAFIK DENGAN RESERVASI PADA PROTOKOL NSIS.....	32
4.3 HAMBATAN PADA PENELITIAN.....	38
BAB V KESIMPULAN.....	39
DAFTAR ACUAN.....	40

DAFTAR GAMBAR

	Halaman
Gambar 2.1. <i>Framework</i> Protokol NSIS.....	6
Gambar 2.2. Mekanisme GIST.....	8
Gambar 2.3. Model QoS NSLP.....	10
Gambar 2.4. <i>Sender-initiated</i> dan <i>receiver-initiated reservation</i>	13
Gambar 2.5. <i>Aggregate reservation</i>	14
Gambar 3.1. Topologi jaringan.....	22
Gambar 4.1. Simulasi 1 pada skenario 1.....	28
Gambar 4.2. Simulasi 2 pada skenario 1.....	29
Gambar 4.3. Simulasi 1 pada skenario 2.....	30
Gambar 4.4. Simulasi 2 pada skenario 2.....	30
Gambar 4.5. Simulasi 1 pada skenario 3.....	31
Gambar 4.6. Simulasi 2 pada skenario 3.....	32
Gambar 4.7. Simulasi 1 pada skenario 1 dengan reservasi.....	33
Gambar 4.8. Simulasi 2 pada skenario 1 dengan reservasi.....	34
Gambar 4.9. Simulasi 1 pada skenario 2 dengan reservasi.....	35
Gambar 4.10. Simulasi 2 pada skenario 2 dengan reservasi.....	36
Gambar 4.11. Simulasi 1 pada skenario 3 dengan reservasi.....	37
Gambar 4.12. Simulasi 2 pada skenario 3 dengan reservasi.....	37

DAFTAR TABEL

	Halaman
Tabel 2.1. Parameter CLS antara RSVP dengan QoS NSLP.....	16
Tabel 2.2. Perbandingan antara RSVP dan NSIS (QoS) Signaling.....	17
Tabel 3.1. Parameter <i>file video</i> untuk trafik <i>video streaming</i>	23
Tabel 3.2. Skenario simulasi tanpa reservasi.....	24
Tabel 3.3. Skenario simulasi dengan reservasi.....	25
Tabel 4.1. Parameter pengujian tanpa reservasi.....	27
Tabel 4.2. Parameter <i>file video</i> untuk trafik <i>video streaming</i>	27
Tabel 4.3. Parameter pengujian dengan reservasi.....	32
Tabel 4.4. Parameter <i>file video</i> untuk trafik <i>video streaming</i>	33

DAFTAR LAMPIRAN

	Halaman
Lampiran 1: Spesifikasi <i>File Video</i> 1.....	43
Lampiran 2: Spesifikasi <i>File Video</i> 2.....	45
Lampiran 3: Data Simulasi 1 Tanpa Reservasi Skenario 1.....	46
Lampiran 4: Data Simulasi 2 Tanpa Reservasi Skenario 1.....	48
Lampiran 5: Data Simulasi 1 Tanpa Reservasi Skenario 2.....	50
Lampiran 6: Data Simulasi 2 Tanpa Reservasi Skenario 2.....	52
Lampiran 7: Data Simulasi 1 Tanpa Reservasi Skenario 3.....	54
Lampiran 8: Data Simulasi 2 Tanpa Reservasi Skenario 3.....	56
Lampiran 9: Data Simulasi 1 Dengan Reservasi Skenario 1.....	58
Lampiran 10: Data Simulasi 2 Dengan Reservasi Skenario 1.....	60
Lampiran 11: Data Simulasi 1 Dengan Reservasi Skenario 2.....	62
Lampiran 12: Data Simulasi 2 Dengan Reservasi Skenario 2.....	64
Lampiran 13: Data Simulasi 1 Dengan Reservasi Skenario 3.....	66
Lampiran 14: Data Simulasi 2 Dengan Reservasi Skenario 3.....	68
Lampiran 15: File Konfigurasi NSIS.....	70

DAFTAR SINGKATAN

C-Mode	Connection Mode
CBQ	Class Based Queueing
CLS	Controlled-Load Service
D-Mode	Datagram Mode
DCCP	Datagram Congestion Control Protocol
EWMA	Ekspontential Weighted Moving Average
FTP	File Transfer Protocol
GIST	General Internet Signaling Transport
HTB	Hierarchichal Token Bucket
HTTP	Hyper Text Transport Protocol
ID	Identifier
IP	Internet Protocol
MA	Message Association
MRS	Message Routing State
NAT	Network Address Translation
NI	NSIS Initiator
NF	NSIS Forwader
NR	NSIS Responder
NSIS	Next Step in Signaling
NSLP	NSIS Signaling Layer Protocol
NTLP	NSIS Transport Layer Protocol
QNE	QoS NSLP Element
QNF	QoS NSLP Forwarder
QNI	QoS NSLP Initiator
QNR	QoS NSLP Receiver
QoS	Quality of Service
QSPEC	QoS Specification
RII	Request Identification Information
RMF	Resource Management Function

RSVP	Resource ReServation Protocol
SCTP	Stream Control Transmission Protocol
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TMOD	Traffic Model
TLS	Transmission Layer Security
UDP	User Datagram Protocol
YESSIR	YEt another Sender Session Internet Reservations

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan aplikasi dan peningkatan pengguna jaringan *internet* pada saat ini, memicu permasalahan yang mendasar, dimana pada awal *internet* didesain hanya bersifat *best-effort* [1], mempunyai paket sederhana yang meneruskan ke suatu *node*, dan memiliki sistem akhir yang kompleks. Bagaimanapun, (dari tahun ke tahun) awal dari ini memiliki banyak tantangan dan kelemahan oleh persyaratan perkembangan aplikasi baru dan peningkatan pengguna pada infrastruktur [2].

Aplikasi-aplikasi yang bermunculan pada saat ini banyak digunakan sebagai bagian dari kehidupan sehari-hari [1]. Diantaranya sebagai fasilitas pendidikan, kedokteran, hiburan, perkantoran. Masing-masing aplikasi memiliki kebutuhan persyaratan QoS tertentu. Salah satu contohnya adalah aplikasi multimedia, *video streaming* yang membutuhkan jaringan yang memberikan jaminan akan kualitas yang baik, agar gambar yang diterima dapat dinikmati dengan baik oleh pengguna.

Berbagai riset beberapa tahun belakangan ini banyak dilakukan oleh pihak akademis dan perusahaan jaringan untuk mengatasi permasalahan pada internet. Dimulai dari ST-II dikembangkan sebagai protokol reservasi untuk komunikasi *point-to-multipoint* [3], lalu RSVP kemudian merancang suatu desain memberikan dukungan untuk reservasi *multipoint-to-multipoint*, kompleksitas dan skalabilitasnya masih menuai kritik, YESSIR, Boomerang [3] yang dikembangkan setelah RSVP. Protokol-protokol tersebut belum mampu mengatasi permasalahan yang terjadi, didalam desainnya masih memiliki banyak kelemahan dan kendala.

Pada tahun 2001 IETF membentuk suatu kelompok kerja baru –*Next Step in Signaling* (NSIS) untuk menyelidiki arsitektur dan protokol untuk umum dan aplikasi *Signaling* spesifik. Lindell dan Branden sebagai pelopor dan mencoba membagi protokol RSVP menjadi dua lapisan arsitektur [2], yaitu NSIS *signaling layer protocol* dan NSIS *transport layer protocol*. NSIS diharapkan dapat

mendukung berbagai aplikasi *signaling* yang dapat menginstall dan memanipulasi suatu status atau keadaan dalam suatu jaringan. Dalam penelitian ini akan dilakukan suatu simulasi dengan melewatkan beberapa jenis trafik kedalam trafik yang tidak berkelas pada jaringan yang menerapkan protokol NSIS.

1.2 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah

1. Mengimplementasikan dan menguji perilaku Protokol *Next Step in Signaling* (NSIS).
2. Menganalisa perilaku dari beberapa jenis trafik sebagai trafik yang tidak berkelas pada protokol *Next Step in Signaling* (NSIS).
3. Menganalisa keterkaitan antara *Hierarchical Token Bucket* (HTB) pada *operating system* linux dengan Protokol *Next Step in Signaling* (NSIS).

1.3 Pembatasan Masalah

Penelitian dibatasi pada pemanfaatan salah satu *tool* yang telah ada dan dikembangkan oleh Universitas Göttingen yaitu OpenNSIS atau saat ini bernama FreeNSIS. Jenis trafik pada simulasi ini adalah *File Transfer Protocol* (FTP), *video streaming*, dan HTTP. Adapun parameter yang diukur adalah *throughput* dari masing-masing jenis trafik dan *aggregate* dari keseluruhan jenis trafik diwaktu bersamaan.

1.4 Metodologi Penelitian

Pembuatan Simulasi terhadap trafik-trafik yang tidak berkelas pada protokol NSIS dengan menggunakan metodologi sebagai berikut:

1. Studi literatur

Melakukan studi literatur yang didapat dari internet, buku-buku, dan media-media lain sebagai landasan teori yang digunakan.

2. Pemilihan jenis trafik yang akan disimulasikan

Menentukan jenis-jenis trafik yang akan dilewatkan pada jaringan dan dijadikan trafik tidak berkelas pada Protokol NSIS.

3. Pemodelan Sistem

Membuat topologi jaringan yang akan digunakan untuk mensimulasikan jenis trafik yang telah ditentukan.

4. Mensimulasikan jenis trafik pada protokol NSIS

Setelah melakukan pemilihan trafik dan pembuatan topologi jaringan, maka dilakukan simulasi dengan menggunakan OpenNSIS sebagai aplikasi dari protokol NSIS dan dijalankan diatas *operating system linux* Mandriva 2008 berbasis *kernel 2.6.23*.

1.5 Sistematika Pembahasan

Thesis ini terdiri dari 5 bab, dimana masing-masing bab mempunyai kaitan satu sama lain, yaitu:

1. **Bab I Pendahuluan**, menjelaskan secara singkat latar belakang, tujuan penelitian, pembatasan masalah, dan metodologi.
2. **Bab II Teori Pendukung**, menjelaskan teori dasar untuk penyelesaian thesis ini. Teori dasar yang diberikan meliputi: *quality of service (QoS)*, *Resource ReReservation Protocol (RSVP)* dengan *Integrated Serviced (IntServ)*, *Differentiated Services (DiffServ)*, *Next Step in Signaling (NSIS)*.
3. **Bab III Pemodelan Simulasi Jaringan**, dalam bab ini akan dibahas mengenai tahap-tahap pembuatan dan proses kerja sistem yang akan dibuat dan perencanaan perangkat lunak.
4. **Bab IV Pembahasan**, pada bab ini berisikan tentang pembahasan dan analisa dari hasil simulasi.
5. **Bab V Kesimpulan**, bab ini berisi tentang hasil dari simulasi yang telah dilakukan.

BAB II DASAR TEORI

2.1 *Quality of Service (QoS)*

Istilah *quality of service* sangat populer dan sering dilihat dari perspektif yang berbeda dalam komunitas jaringan dan pembuat aplikasi [1]. Pada jaringan “*Quality of Service*” mengacu pada kemampuan untuk menyediakan perlakuan yang berbeda untuk kelas yang berlainan dari trafik [1]. Tujuan utamanya adalah untuk meningkatkan kegunaan secara keseluruhan dari jaringan dengan mengijinkan ke nilai yang tertinggi atau yang memiliki tingkat sensitifitas yang tinggi pada *flow*. Kata “Prioritas” memiliki makna kemungkinan dibuang yang lebih rendah atau perlakuan istimewa *queuing* pada *congested interfaces*. QoS digunakan untuk mengangkat prioritas dari *flow* tertentu diatas tingkatan yang diberikan kepada kelas *best-effort service*, memerlukan *admission control* dan *policing* untuk menjaga ketertiban *flow* untuk mencegah pencurian dari layanan.

Berbeda sekali, pandangan dari QoS dari para pemakai aplikasi dan komunitas pengembang aplikasi, lebih subjektif menilainya. QoS dipandang sebagai sesuatu yang dapat meningkatkan pencapaian [1]. Hasilnya adalah sangat berhubungan antara kegunaan aplikasi tertentu dan kemampuan dari jaringan, istilah kegunaan adalah suatu mutu yang dirasakan, menyenangkan atau tidak menyenangkan dari suatu presentasi bagi para pengguna aplikasi (misalkan mutu dari Gambar dari suatu jalannya *video* yang ditunjukkan) [1].

2.2 *Resource ReReservation Protocol (RSVP)*

2.2.1 *Integrated Services (IntServ)*

IntServ [4] telah diperkenalkan pada jaringan IP dengan tujuan untuk menyediakan *guaranteed* dan *controlled service* sebagai tambahan terhadap layanan *best-effort* yang tersedia. Masing-masing trafik *flow* dapat diklasifikasikan dalam 3 kelas layanan:

1. *Guaranteed-service class* [5]: kelas ini menyediakan persetujuan untuk layanan *delay-bound* seperti *voice* dan aplikasi *real-time* lainnya, yang membutuhkan *delay constraints*.
2. *Controlled-load service class* [6]: kelas ini menyediakan suatu format dari persetujuan layanan *statistical delay*, sebagai contoh, dengan nominal *mean delay*.
3. *Best-effort service*: kelas ini telah termasuk pada layanan IP saat ini untuk layanan interaktif (misalnya *web*), *interactive bulk traffic* (misalnya FTP), dan *background* atau *asynchronous traffic* (misalnya *email*).

Resource reservation protocol (RSVP) [7] digunakan untuk *guaranteed* dan *controlled-load services*, dengan berbasis pada kebutuhan layanan secara kuantitatif, memerlukan *signaling*, dan *admission control* pada *node* jaringan, RSVP merupakan protokol yang digunakan untuk reservasi *resource* pada *router*, pada basis *hop-by-hop*, dengan memperhatikan kebutuhan aplikasi untuk *flow* IP yang telah ditentukan.

2.2.2 Differentiated Services (DiffServ)

Arsitektur *Differentiated Services* didasarkan pada suatu model yang sederhana dimana suatu trafik yang memasuki suatu jaringan digolongkan dan dimungkinkan pengkondisian batasan-batasan pada jaringan, dan ditetapkan pada perbedaan perilaku *aggregates* [8].

2.3 Next Step in Signaling (NSIS)

NSIS merupakan sistem *signaling* pada IP, yang diharapkan dapat mengatasi kelemahan pengganti RSVP. NSIS diharapkan bersifat generik dan extensible. Untuk itu, protokol NSIS memisahkan fungsionalitas seperti reliabilitas, fragmentasi, kontrol kongesti, dan integritas; dengan aplikasi *signaling* [9]. Maka dalam NSIS terdapat dua *layer* protokol [9]:

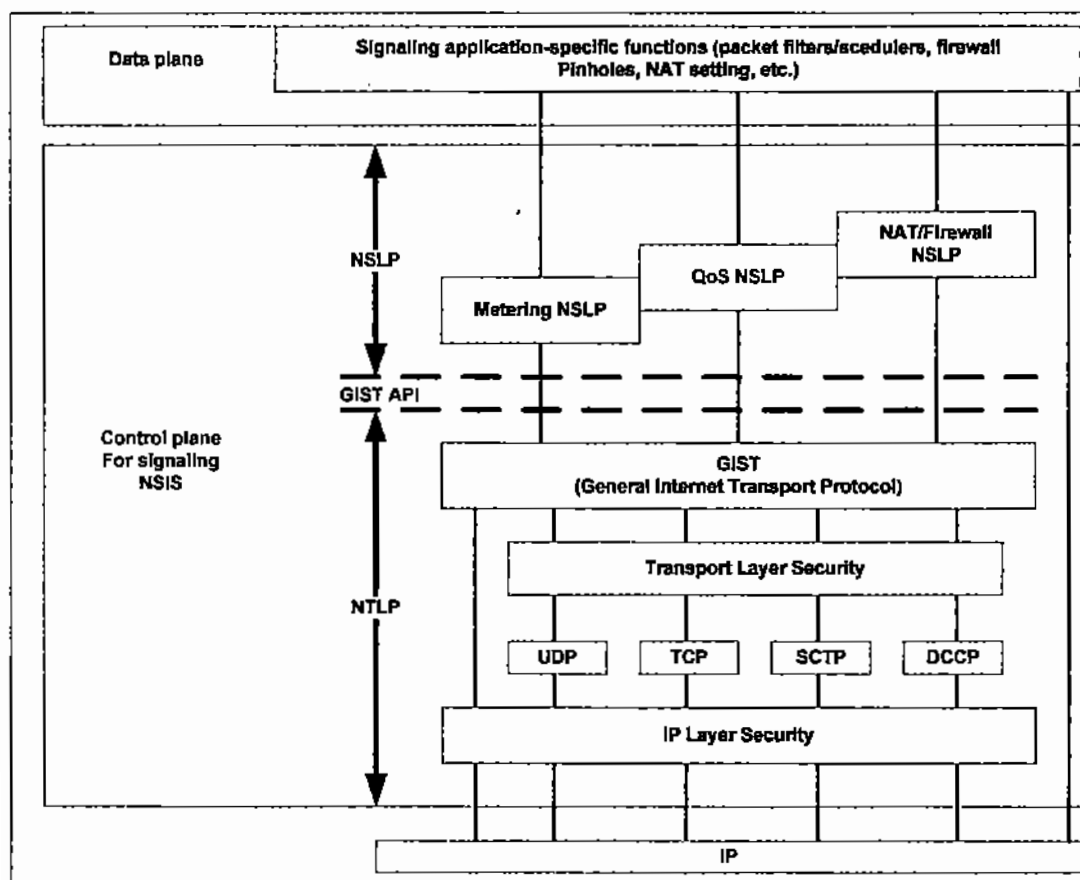
1. NSIS *Transport Layer Protocol* (NTLP), digunakan untuk *layer messaging*, yang disebut GIST, untuk mentransportasikan pesan *layer* aplikasi *signaling*. *Layer* GIST dijalankan di atas protokol *transport* dan *security* standar, seperti *User Datagram Protocol* (UDP), *Transmission*

Control Protocol (TCP), Stream Control Transmission Protocol (SCTP), dan Datagram Congestion Control Protocol (DCCP).

2. NSIS Signaling Layer Protocols (NSLPs), masing-masing menjalankan fungsionalitas persinyalan yang spesifik menurut aplikasi, termasuk menyampaikan format dan aturan pengolahan antar NSLP sendiri.

Contoh NSLPs adalah

- QoS NSLP untuk reservasi,
- NAT/Firewall NSLP untuk konfigurasi *middlebox*,
- dan NSLP untuk konfigurasi pemetaran.



Gambar 2.1. Framework Protokol NSIS [10]

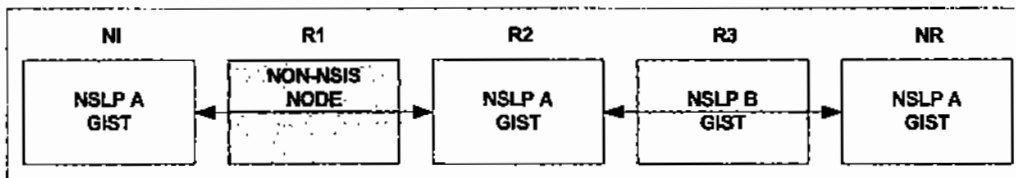
2.3.1 General Internet Signaling Transport Protocol (GIST)

Protokol GIST merupakan bagian pokok dari deretan protokol yang membentuk NSIS. Tugas utama dari GIST adalah untuk menyampaikan pesan

signaling dari berbagai NSLP antara GIST yang berdekatan dengan yang mendukung NSLP yang sama. NSLP sendiri yang bertanggung jawab untuk mendorong pesan *signaling* dari NSIS *Initiator* (NI) kearah NSIS *Responder* (NR), secara khusus sumber *flow* dan tujuan masing-masing, GIST hanya menyediakan cara untuk mengantarkan pesan dari satu *node* ke *node* selanjutnya didalam *path*. NI dan NR bagaimanapun juga dapat diwakili oleh *proxy*, misalnya untuk mendukung *end system* yang tidak memiliki kemampuan NSIS.

Daripada membangun protokol *transport* yang baru, GIST menggunakan kembali protokol *transport* dan *security* yang sudah ada untuk mendukung *transport* layanan pesan yang universal. Seperti RSVP, GIST merupakan *soft-state protocol*. GIST membuat dan memelihara dua jenis dari *state* yang dihubungkan dengan *signaling transport*: suatu *session message routing state* (MRS) untuk mengatur proses dari keluarnya pesan, dan suatu *message association* (MA) *state* untuk mengatur *per-peer state* yang dihubungkan dengan model koneksi pesan kepada *peer* tertentu, mencakup *signaling* alamat tujuan, protokol dan nomor *port*, konfigurasi internal protokol, dan informasi status. Sebagai tambahan terhadap GIST yang berdekatan mengamati informasi, GIST juga memelihara pesan informasi *routing* tertentu, seperti *flow identifier* (*flow ID*), jenis NSLP dan *session identifier* (*session ID*), untuk mengidentifikasi secara unik aplikasi *signaling layer session* pada *flow*.

GIST memiliki 2 jenis tipe operasinya; *datagram mode* (*D-mode*), dimana menggunakan suatu mekanisme *datagram transport* yang tidak dapat dipercaya keamanannya, dengan UDP sebagai pilihan awalnya, dan *connection mode* (*C-mode*), yang mana digunakan beberapa *stream* atau *message-oriented transport protocol* (saat ini TCP sebagai pilihan awal) dan dapat juga digunakan IPsec *security* atau *Transport Layer Security* (TLS). Dimungkinkan terjadi penggabungan dari dua tipe dari suatu *node*, tanpa koordinasi atau konfigurasi manual. Itu diijinkan, sebagai contoh, pada *edges network* menggunakan *D-Mode* dan pada *core network* digunakan *C-mode*.



Gambar 2.2. Mekanisme GIST [2]

Pada Gambar 2.2 dapat diperhatikan mekanisme dari GIST, dimana A sebagai QoS NSLP sedangkan B adalah NSLP jenis yang lainnya. Asumsi pesan QoS *RESERVE* merupakan yang dihasilkan dari GIST pada NI, suatu *flow sender*. Gagasan pertama Modul GIST adalah pesan *GIST-query*, yaitu alamat UDP *datagram* kepada suatu *flow* tujuan dan termasuk *IP Router Alert Option*, seperti pada RSVP. *Downstream NSIS peer* mendukung suatu QoS NSLP, R2 dalam kasus ini, mengenali pesan ini dan menjawab pesan tersebut dengan pesan *GIST-Response*, NI membuat *message association (MA)* dengan R2. Setelah itu, semua pesan *GIST-Confirm* atau *GIST-Data* yang berikutnya, yaitu pesan GIST dengan *NSLP payload*, diantara dua *peers* dapat dikirimkan melewati MA ini. Ketika penerimaan seperti pesan GIST pada R2, *NSLP payload* dan *flow ID* diberikan kepada QoS NSLP yang memprosesnya. Dengan catatan ini adalah tanggung jawab dari *NSLP layer* untuk menentukan tindakan atas penerimaan suatu pesan GIST. Jika QoS NSLP di *node* ini menentukan bahwa itu mempunyai cukup *resource* seperti yang diminta oleh QoS NSLP *RESERVE*, itu akan membuat suatu reservasi yang bersifat sementara untuk sesi tersebut. Pesan QoS NSLP *RESERVE* akan dikirimkan ke QoS NSLP *node* berikutnya sepanjang jalur, menurut prosedur yang telah diuraikan di atas. Ketika akhirnya NR QoS NSLP menerima pesan *RESERVE*, itu menjawab sepanjang alur kebalikan ke arah NI dengan suatu pesan *RESPONSE* untuk mengkonfirmasi penetapan akhir dari reservasi itu. Pada contoh ini, QoS NSLP *payload* (misalnya untuk *signaling IntServ* terutama *TSpec* dan *RSpec*) dikirimkan, diuji, dan modifikasi dimungkinkan terjadi diantara *node*, berseberang jalur dari suatu QoS NSLP *aware nodes*.

Suatu pesan GIST terdiri dari suatu *common header* dan suatu urutan dari *type-length-value (TLV) object*. Suatu *common header* menandai jenis pesan seperti *Query/Response* dan lain-lain, seperti halnya *NSLP identifier (NSLP ID)*

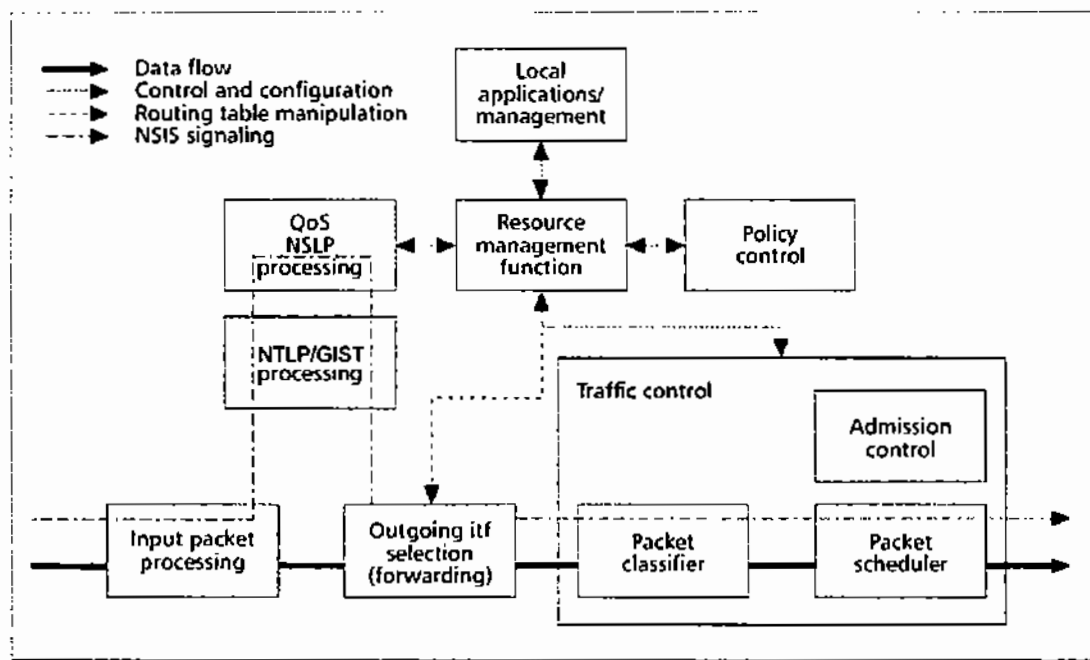
dan lompatan perhitungan untuk menghindari pengulangan pesan. Sebagai tambahan, GIST dapat menggunakan *query-cookies* dan *response-cookies* untuk perlindungan melawan *spoofing* dan *denial of service attacks*.

Pesan GIST *query* dikirimkan kembali dengan *exponential back-off* jika suatu tanggapan yang sesuai tidak diterima tepat waktu. Pesan NSLP lainnya *encapsulated* dalam *datagram-mode* adalah tidak dikirimkan kembali; mereka bersandar pada pesan *query* awal itu akhirnya dikirim kembali. Kapan saja mungkin, penggunaan dari *reliable transport* dan *security protocol* yang saat ini ada dapat direkomendasikan, melalui *connection-mode* pada GIST. *Connection mode* adalah sangat penting untuk objek data yang berukuran besar, ketika *state setup* berjalan cepat di permukaan dari *packet loss* adalah *desirable*, atau dimana keamanan saluran adalah diperlukan. Kesangsian *node* dapat dipilih untuk menyegarkan pesan *routing state* dengan mengirimkan GIST *query*. Bagaimanapun, apakah untuk memelihara *messaging association* adalah ditentukan oleh *local policy*. Sebagai contoh, suatu *node* boleh memilih untuk mempertahankan *association* jika ada *flow* masih di dalam tempat yang mungkin menghasilkan pesan menggunakan itu [11].

2.3.2 Aplikasi *Signaling QoS* pada NSIS

2.3.2.1 Model dari Operasi *QoS NSLP*

Bagian ini menyajikan suatu model operasi dari *QoS NSLP* dan mekanisme dalam satu *node* [11].



Gambar 2.3. Model QoS NSLP [12]

Dilihat dari satu *node*, permintaan dari QoS mungkin dihasilkan dari permintaan aplikasi lokal, diajukan oleh pengguna aplikasi (misalkan aplikasi multimedia) [13] atau manajemen jaringan, misalkan, pengajuan *tunnel* untuk menangani *aggregate*, atau dari proses kedatangan pesan QoS NSLP. Pesan yang datang pada kasus ini memerlukan pesan NSIS untuk ditangkap, selama pengolahan paket yang masuk dan ditangani oleh GIST. Hanya pesan yang berhubungan dengan QoS yang diabaikan ke QoS NSLP yang memproses modul.

Pada masing-masing QoS NSLP *node*, permintaan QoS, terutama QSPEC, ditangani oleh *Resource Management Function* (RMF). Model QoS lokal menggambarkan bagaimana RMF akan menterjemahkan QSPEC, dan bagaimana untuk mengakui dan mengkonfigurasi *resource*. Pengelolaan keputusan melibatkan dua modul lokal tambahan; *policy control* dan *admission control*. *Policy control* menentukan apakah pengguna diberikan hak untuk membuat reservasi [13], *admission control* menentukan jaringan dari *node* mempunyai *resource* yang cukup tersedia untuk menyediakan permintaan QoS [13]. Jika kedua pengecekan sukses, parameter diset pada *packet classifier* dan pada *link layer interface* (misalkan pada *packet scheduler*) untuk memperoleh QoS yang diminta [13]. Akhirnya, QoS NSLP *node* menunjukkan bahwa *resource* yang

diperlukan telah dikonfigurasi. Oleh karena itu, QoS NSLP menghasilkan suatu *acknowledgment message* di satu arah, dan menyebarkan permintaan *resource* sepanjang lebih lanjut ke arah penerima data.

2.3.2.3 Message Protocol dari QoS NSLP

QoS NSLP merupakan *soft state protocol*. Memiliki 4 tipe *message* [13];

1. *RESERVE message* adalah pesan satu-satunya yang memanipulasi status reservasi QoS NSLP. Itu digunakan untuk *create*, *refresh*, *modify* dan *remove*.
2. *QUERY message* meminta informasi tentang alur data tanpa membuat reservasi. Kemampuan ini digunakan untuk “pemeriksaan” jaringan untuk karakteristik alur, untuk reservasi *receiver-initiated* atau untuk dukungan dari model QoS tertentu.
3. *RESPONSE message* menyediakan informasi tentang hasil dari pesan QoS NSLP sebelumnya.
4. *NOTIFY message* dapat digunakan untuk menyampaikan informasi kepada *node* QoS NSLP. Ini berbeda dengan suatu *RESPONSE message* oleh karena itu dikirim secara tidak serempak dan tidak perlu mengacu pada status tertentu atau pesan yang diterima sebelumnya. Informasi yang disampaikan oleh *NOTIFY message* merupakan secara khusus berhubungan dengan kondisi-kondisi kesalahan.

QoS NSLP *messages* memuat 3 tipe objek [13]:

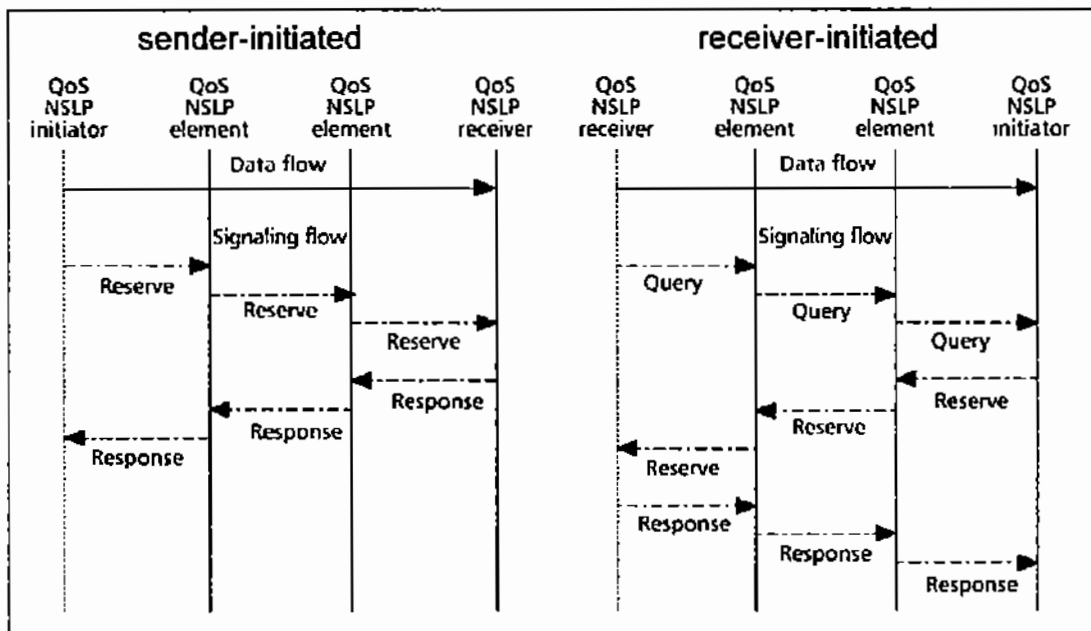
1. *Control Information*: membawa informasi umum untuk QoS NSLP processing, seperti *sequence numbers* atau suatu *response* diperlukan.
2. *QoS specifications (QSPECs)*: menguraikan rincian *resource* yang diperlukan dan tergantung pada model QoS yang digunakan. Disamping hal tersebut QSPEC juga berisikan *control information* yang digunakan dalam *RMF's processing*.
3. *Policy objects*: berisi data yang digunakan untuk memberikan hak reservasi dari *resources*.

QoS-NSLP dapat digunakan pada skenario *signaling* yang kompleks, yang mendukung mekanisme tingkat lanjut [14]:

1. *Summary refreshes* memungkinkan format yang disingkat dari penyegaran *RESERVE message*.
2. *Message scoping: local policy* memungkinkan pengguna untuk menentukan apakah meneruskan pesan atau tidak;
3. *Session binding* menyelenggarakan suatu hubungan antar sesi yang berbeda. Informasi ini dapat digunakan untuk optimasi lokal dalam kasus reservasi *bi-direction* atau *aggregate*;
QoS NSLP mendukung fasilitas *aggregation* serupa dengan halnya skenario aplikasi yang lain seperti *signaling* untuk parameter QoS lokal dan model.
4. *Route change detection*: QoS NSLP dapat mendeteksi perubahan *route* dan secara otomatis menyesuaikan *route* yang baru.
5. *Reduced state*: QoS NSLP tidak memberikan tugas pada masing-masing QoS NSLP *node* untuk menyimpan status reservasi QoS. Itu mendukung operasi pengurangan status, dimana status reservasi dengan *granularity* yang kasar (misalkan per *class*) digunakan, atau *stateless operation* dimana tidak ada status QoS NSLP diciptakan. Sebagai contoh untuk operasi ini adalah *Resource Management* pada *DiffServ* (RMD), yang dapat digunakan untuk *signal DiffServ* pada *core routers*.

2.3.3 Reservasi pada QoS NSLP

Pesan QoS NSLP dikirim NSIS *peer-to-NSIS* per. Berbeda dengan RSVP, QoS NSLP mendukung kedua reservasi *sender-initiated* dan *receiver-initiated* [14]. Gambar 2.4 menunjukkan mekanisme dari kedua jenis reservasi yang dapat dilakukan oleh QoS NSLP.



Gambar 2.4. *Sender-initiated* dan *receiver-initiated reservation* [12]

1. *Sender-initiated reservation*

Untuk membuat reservasi, QNI membuat “*Reserve*” message yang mengandung *QSPEC object*. Message akan dikirimkan melalui QNE sampai ke QNR, dan QNR akan memberikan konfirmasi dari state, dengan mengirimkan suatu “*Response*” message.

2. *Receiver-initiated reservation*

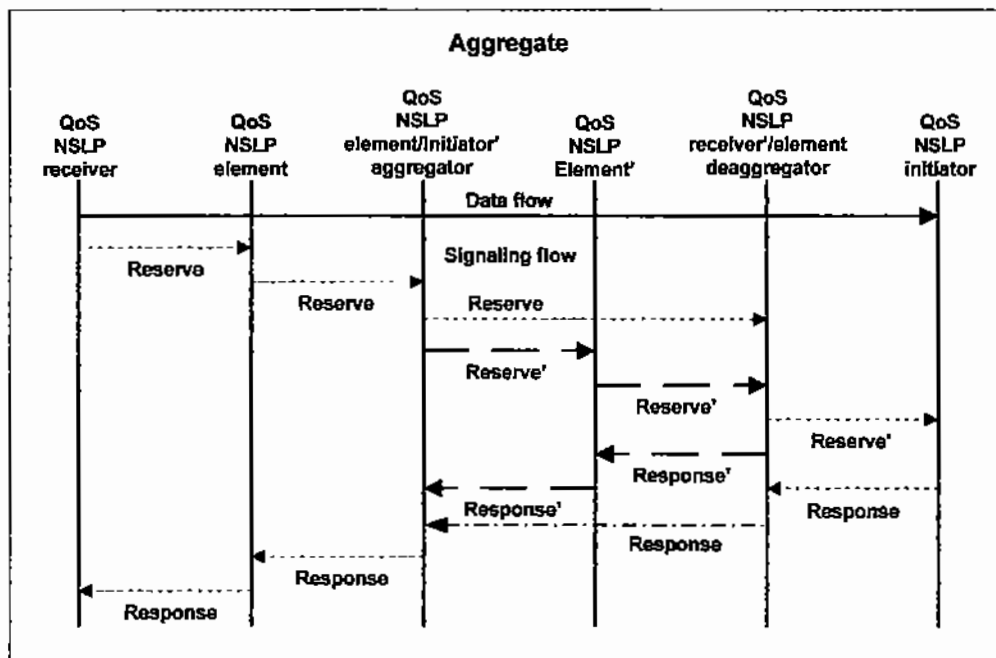
Pada reservasi *receiver-initiated*, QNR (*sender*) membuat “*Query*” message dengan “*Reserve-Init flag*” dan termasuk *QSPEC object*. Karena “*Query*” message tidak harus trigger suatu “*Response*” message, maka tidak mengandung “*RII object*”. Disaat berapa pada QNI (*receiver*), “*Query*” message menyimpan informasi *bandwidth* yang tersedia pada jaringan kedalam *QSPEC object*.

QNI mendeteksi “*Query*” message yang membawa “*Reserve-Init flag*” dan dengan menggunakan informasi yang terdapat pada *QSPEC*, QNI membuat “*Reserve*” message lalu meneruskan *peer-to-peer* dengan *GIST reserve path state*. Setiap node terdapat didalamnya *RII* pada “*Reserve*” message. “*Response*” message dikirimkan untuk mengkonfirmasi *resource* telah siap.

3. Aggregate reservation

untuk mengurangi *signaling* dan manajemen *per-flow state* pada jaringan, reservasi *end-to-end per flow* dimulai dengan *RESERVE message*. Pada aggregator reservasi untuk *aggregated flow* dimulai dengan *RESERVE' message*. Aggregator melakukan *marking* dengan memodifikasi QoS NSLP nilai NSLP-ID dengan menentukan ulang nilai NSLP-ID.

Deaggregator bertindak sebagai QNR untuk reservasi *aggregate*. Informasi *session binding* yang dibawa pada *RESERVE message* membuka peluang *deaggregator* untuk berhubungan *end-to-end* dan reservasi *aggregate* dengan satu sama lainnya (menggunakan *BOUND_SESSION_ID*).



Gambar 2.5. Aggregate reservation

2.3.4 QSPEC

QSPEC merupakan suatu objek dari QoS NSLP yang mengandung seluruh informasi secara spesifik. QoS NSLP merupakan protokol QoS *signaling* umum yang dapat digunakan untuk bermacam-macam QoS *Model* (QoS Ms) [15].

Suatu QoS M menetapkan suatu parameter QSPEC yang menggambarkan tentang QoS *desired* dan bagaimana *resource* akan dikelola oleh RMF. RMF

menerapkan fungsi sebagai manajemen *resource* dan mengelola parameter QSPEC. QoSM merupakan suatu metode pencapaian dari QoS pada *traffic flow*, misalnya [15];

1. *IntServ controlled load* [CL-QOSM],
2. *resource management with DiffServ* [RMD-QOSM],
3. dan QoS *signaling* for Y.1541 QoS *classes* [Y.1541-QOSM].

QSPEC terdiri dari QSPEC *objects*, yaitu QoS *Desired*, QoS *Available*, dan *Minimum* QoS. QSPEC memiliki kemiripan parameter seperti hanya pada RSVP, misalnya: TSpec, RSpec dan AdSpec.

1. QoS *Desired*: berisi parameter yang menggambarkan QoS yang diinginkan oleh suatu QNI.
2. QoS *Available*: berisi parameter yang menggambarkan *resource* yang tersedia.
3. QoS *Reserved*: menggambarkan QoS aktual yang diminta dalam reservasi.
4. *Minimum* QoS: parameter tambahan yang dimasukkan oleh QNI untuk mengindikasikan *range* dari QoS yang dapat diterima.

Selain memiliki *object* QSPEC memiliki parameter, parameter tersebut meliputi:

1. TMOD parameter, trafik
2. *Constraints* parameter (misalnya *path latency*)
3. *Traffic handling directives* (misalnya *excess treatment*)
4. *Traffic classifier* (misalnya PHB *class*)

2.3.5 QoS Model *Controlled-Load Service* (QoS M CLS)

Pada [6], *controlled-load service* ditetapkan pada *IntServ* mendukung aplikasi yang memiliki tingkat sensitifitasnya tinggi terhadap kondisi yang melampaui batas dari kemampuan pada jaringan, seperti aplikasi *real-time*. *Controlled-load service* QoS M menggunakan parameter TMOD1, dengan *token bucket* sebagai parameternya. Parameter tersebut meliputi [15]:

1. *Bucket rate* r
2. *Bucket depth* b
3. *Peak rate* p

4. *Minimum policed unit m*

Controlled-load service QoSM bermaksud tetap konsisten terhadap spesifikasi pada *RSVP/Controlled Load Service* [6][15].

Tabel 2.1. Parameter CLS antara RSVP dengan QoS NSLP [15]

	Message	Object (s)	Parameter (s)
RSVP	Path	Sender Tspec	token bucket
		ADSpec	available bw and MTU
QoS NSLP	QUERY	<QoS Desired>	<TMOD1>
		<QoS Available>	<TMOD1> (bandwidth)
RSVP	Resv	FlowSpec	TMOD1
QoS NSLP	RESERVE	<QoS Desired>	<TMOD1>
		<QoS Available>	<TMOD1> (bandwidth)
RSVP	ResvConfirm		
QoS NSLP	RESPONSE	<QoS Reserved>	<TMOD1>

Untuk *signaling* CLS IntServ pada QoS NSLP, *SenderTspec* dan *ADSpec* diterjemahkan menjadi *QoS Desired* dan *QoS Available* dengan *QUERY* pada jaringan untuk mengetahui ketersediaan *resource*. Untuk *RESERVE message*, *FlowSpec* diterjemahkan menjadi *QoS Desired* dan *QoS Available*. Parameter *QoS Available* digunakan untuk mengetahui kondisi pada jaringan berdasarkan *update* yang dilakukan oleh seluruh QNE pada jaringan. Sedangkan untuk *RESPONSE message* membawa informasi *QoS Desired object* kepada *initiator* tentang *resource* yang telah dipesan [12].

2.4 Perbandingan antara RSVP dan NSIS (QoS) Signaling

Perbedaan yang mendasar antara RSVP dan NSIS terutama berkenaan dengan QoS *signaling* yang diringkas pada tabel II.1 dibawah ini.

Tabel 2.2. Perbandingan antara RSVP dan NSIS (QoS) Signaling [14]

	RSVP	NSIS
Protocol structure	single layer	two layers
Transport	IP or UDP	reliable (TCP, STCP) /datagram (UDP, DCCP)
Reservation initiator	Receiver	sender or receiver
States	soft + explicit release	soft + explicit release
QoS models	IntServ/DiffServ	IntServ/DiffServ/other
Scope of signaling	end2end	end2end/host2edge/edge2edge
Multicast	yes	no
Mobility	no	yes
Bi-directional	no	yes
Aggregation	yes	yes
Summary refresh	yes	yes
Priority/preemption	yes	yes

1. *Transport of signaling messages*: pesan RSVP ditranspor *unreliably* oleh UDP atau secara langsung diatas IP. Pada NSIS, pesan mekanisme *transport* (NTLP) terpisah dari aplikasi *signaling* (NSLP). Dengan mempertimbangkan aplikasi *signaling* yang berbeda-beda. Perbedaan aplikasi dan *session* dapat berbagi yang sama *message associations* sedemikian adalah tidak perlu menciptakan *message associations* untuk masing-masing *session*. NTLP menggunakan protokol *transport* yang ada saat ini, meliputi TCP dan UDP.
2. *Reservation model*: model reservasi RSVP adalah *receiver-initiated*, dan *signaling* meluas dari *flow* pengirim kepada *flow* penerima. NSIS QoS NSLP mendukung kedua reservasi *sender* dan *receiver*. Penggunaan *proxy* didukung, yaitu pesan NSIS dapat diaktifkan dan diakhiri dalam *node* selain pada sumber atau akhir dari *data flow*, dengan demikian, ruang lingkup dari NSIS *signaling* dapat *end-to-end*, *edge-to-edge*, *host-to-edge* atau *edge-to-host*.
3. *Multicast*: tidak seperti RSVP, NSIS tidak mendukung *multicast*, mengurangi kompleksitas untuk mayoritas dari aplikasi pengguna adalah *unicast*. Bagaimanapun, model dasar protokol NSIS adalah dimungkinkan dapat diperluas ke IP *multicast*.

4. *Bidirectional reservation*: QoS NSLP mendukung reservasi *bidirectional* dengan bungkus *session* pada kedua arah. Tidak ada dukungan seperti ini pada RSVP.
5. *QoS Model*: NSIS QoS NSLP mengizinkan kepada pensinyalan model QoS apapun.
6. *Mobility Support*: berdasarkan identifikasi *signaling session* dengan *random session identifier*, dibandingkan oleh *flow identifier* mencakup *IP address*, NSIS dapat mendukung mobilitas lebih mudah.
7. *Security*: ketika keamanan ditambahkan pada RSVP setelah rancangan awal, NSIS telah dirancang dengan keamanan sejak awal pemikiran, mengintegrasikan protokol keamanan yang baku, seperti TLS atau IPsec//IKEv2. Protokol ini menawarkan keunggulan seperti metode pengesahan yang fleksibel, negosiasi dari algoritma *crypto*, dan secara ekstensif protokol yang terbukti, dan perlindungan *Denial of service*.

BAB III

PEMODELAN SIMULASI JARINGAN

3.1 Arsitektur Aplikasi

Simulasi yang dilakukan dalam penelitian ini menggunakan salah satu aplikasi protokol NSIS yang saat ini dikembangkan oleh Georg-August University of Goettingen, yang bernama OpenNSIS (FreeNSIS) yang dibuat dalam bahasa pemrograman C++, dan dijalankan pada *operating system linux*. Implementasi yang telah dikembangkan untuk mendukung protokol NSIS, diantaranya implementasi dari GIST, QoS NSLP, NAT/FW NSLP dan *Diagnostics* NSLP digunakan untuk pengujian. Aplikasi ini dapat diambil pada [16].

Berdasarkan pada kemampuan pengangkutan yang ditawarkan oleh GIST, NSLP menerapkan logika pensinyalan secara riil. Sebagai contoh, *Quality of Service* (QoS) NSLP dapat melakukan signaling reservasi QoS (misalkan *bandwidth*, *delay*, dan lain-lain), sedangkan NAT/*Firewall* (NAT/FW) dapat melakukan konfigurasi pada *Middleboxes* seperti *firewall* atau *router* NAT.

3.1.1 Traffic Control Interface

Hierarchical Token Bucket (HTB) merupakan teknik penjadwalan paket yang baru-baru ini diperkenalkan bagi *router* berbasis Linux, dikembangkan oleh Martin Devera pertama kali pada akhir 2001 untuk diproyeksikan sebagai pilihan atau pengganti dari mekanisme penjadwalan yang masih banyak dipakai pada saat ini yaitu *Class Based Queueing* (CBQ) [17]. HTB diklaim menawarkan kemudahan pemakaian dengan teknik peminjaman dan implementasi pembagian trafik yang lebih akurat. Dasar kerja HTB hampir sama dengan disiplin antrian CBQ bahkan diagram blok sistem CBQ dengan HTB tidak ada perbedaan, hanya saja pada *General Scheduler* HTB menggunakan mekanisme *Deficit Round Robin* (DRR) dan pada blok umpan balik, Estimator, HTB tidak menggunakan *Exponential Weighted Moving Average* (EWMA) melainkan *Token Bucket Filter* (TBF) [18].

Pada HTB terdapat parameter *ceil* sehingga kelas akan selalu mendapatkan *bandwidth* di antara *base link* dan nilai *ceil link*. Parameter ini dapat dianggap sebagai Estimator kedua, sehingga setiap kelas dapat meminjam *bandwidth* selama *bandwidth* total yang diperoleh memiliki nilai di bawah nilai *ceil*. Hal ini mudah diimplementasikan dengan cara tidak mengizinkan proses peminjaman *bandwidth* pada saat kelas telah melampaui *link* ini (keduanya *leaves* dan *interior* dapat memiliki *ceil*). Sebagai catatan, apabila nilai *ceil* sama dengan nilai *base link*, maka akan memiliki fungsi yang sama seperti parameter *bounded* pada CBQ, di mana kelas-kelas tidak diizinkan untuk meminjam *bandwidth*. Sedangkan jika nilai *ceil* diset tak terbatas atau dengan nilai yang lebih tinggi seperti kecepatan *link* yang dimiliki, maka akan didapat fungsi yang sama seperti kelas *non-bounded* [18].

Implementasi QoS NSLP yang dikembangkan [12] menggunakan HTB sebagai *traffic control* pada sistemnya. HTB pada dasarnya berbasis pada mekanisme *token bucket*. Suatu *bucket* dapat menentukan tiap-tiap kelas dari *outgoing traffic*. Suatu *bucket* dapat diisi pada waktu tertentu dengan *token*. Jika ada *token* pada *bucket*, data dapat dilakukan *dequeued* dari kelas. Jika tidak ada data dalam kelas untuk dikirimkan *token* dapat disimpan untuk digunakan dikemudian. Jumlah maksimum dari suatu *token* adalah setara dengan maksimum *burst size*. HTB mampu menangani berbagai macam kelas *token bucket* dalam suatu *tree*. *Token* yang tidak digunakan dalam satu kelas dapat dipinjamkan oleh kelas yang lain. Perbedaan tingkatan prioritas dapat menentukan kelas pada *tree*. Prioritas yang lebih tinggi memiliki *queueing delay* yang pendek [12].

3.1.2 Resource Management Function (RMF)

IntServ Controlled Load (CLS) QoS Model diterapkan pada RMF (*resource management function*). CLS memungkinkan *flow* dari suatu trafik dapat dipersamakan kepada *unloaded network* yang menawarkan *best effort service* untuk *flow* tersebut. Karakteristik dari trafik adalah *signalled* sebagai *token bucket* dan reservasi aktual menggunakan *linux traffic control interface* [19].

RMF yang telah diimplementasikan pada OpenNSIS saat ini, diantaranya [16]:

1. *NullRMF*, hanya menguji pengoperasian kemampuan QoS NSLP tanpa interaksi RMF.
2. *SimpleRMF*, menggunakan *Traffic Control (Linux)* untuk memesan/mencadangkan QoS pada antrian.
3. *ClassRMF*, implementasi model QoS untuk *controlled-load service* dari *IntServ*. Dimana informasi *signaling* adalah *bandwidth* dalam bentuk dari *token bucket*.

3.2 Implementasi Komponen dan Topologi Jaringan dalam Simulasi

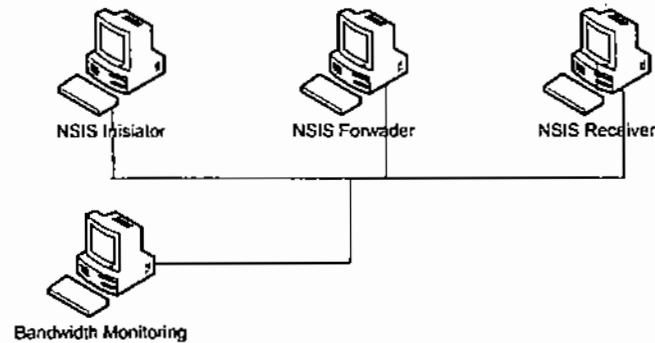
3.2.1 Implementasi Komponen

Parameter komponen yang akan dikonfigurasi dalam simulasi ini meliputi:

- Jenis RMF (*resource management function*)
 Pada simulasi ini akan digunakan salah satu kelas RMF untuk mengetahui interaksi yang terjadi antara trafik yang dilewatkan pada protokol NSIS yaitu *ClassRMF*, dikarenakan kelas ini sudah mendukung interaksi antara aplikasi OpenNSIS dengan kernel *operating system linux* secara langsung.
- *Maximum overall bandwidth*
 Standar maksimum dari *bandwidth* secara keseluruhan telah diatur dalam aplikasi OpenNSIS adalah 100Mbit/s (*default*).
- *Minimum bandwidth unclassified traffic*
 Standar minimum dari *bandwidth* untuk trafik tidak berkelas telah diatur dalam aplikasi OpenNSIS adalah 100 kbit/s (*default*).
- *Maximum bandwidth unclassified traffic*
 Standar maksimum dari *bandwidth* untuk kelas tidak berkelas telah diatur dalam aplikasi OpenNSIS adalah 100Mbit/s (*default*).
- Ukuran paket yang akan dikirim
 Pada simulasi ini akan digunakan pada ketiga trafik ini sebuah file *video*.
- Jenis trafik yang akan digunakan sebagai trafik yang tidak berkelas
 Pada penelitian ini, dilakukan pemilihan beberapa jenis trafik yaitu FTP, *video streaming*, dan HTTP. Ketiga aplikasi tersebut akan dijadikan trafik yang tidak berkelas pada protokol NSIS.

3.2.2 Topologi yang digunakan dalam Simulasi

Secara umum bentuk topologi yang digunakan dalam penelitian ini adalah



Gambar 3.1. Topologi jaringan

Dari Gambar diatas NI adalah NSIS *Initiator*, NF sebagai NSIS *Forwarder*, dan NR sebagai NSIS *Receiver*, dimana pada NR berfungsi sebagai 3 buah *server*, diantaranya *server video streaming (VLC)*, *server FTP (ProFTP)*, dan *server HTTP (Apache)*. *Bandwidth monitoring* sebagai *server monitoring*.

Perangkat keras yang dibutuhkan dalam simulasi adalah

- Intel Pentium 4 2GHz
- 4 kartu Ethernet
- 256 MB DDRAM PC3200
- 20 GB HDD

Dimana nantinya ketiga *node* akan diimplementasikan dengan diatas aplikasi VMWare, aplikasi ini digunakan untuk memudahkan dalam melakukan penerapan dari protokol NSIS, aplikasi ini merupakan *virtual machine* dimana memiliki fungsi yang sama dengan sebuah *personal computer* aslinya, pada aplikasi tersebut akan diinstal *operating system linux* pada masing-masing *node*.

3.3 Pembangunan Sistem

3.3.1 Instalasi Aplikasi OpenNSIS

Pembangunan simulasi ini dibuat diatas *operating system linux* Mandriva 2008 yang berbasis kernel 2.6.23. Aplikasi OpenNsis yang digunakan pada simulasi ini merupakan versi 0.5.0. berikut ini cara dalam proses instalasi aplikasi:

```
# tar xvzf nsis-0.5.0.tar.gz
```

```
# cd nsis-0.5.0
# ./configure
# make
```

Dimana masing-masing *node* (NI, NF, NR) pada simulasi ini harus sudah terinstal aplikasi OpenNSIS.

3.4 Pembangunan Simulasi

3.4.1 Skenario Simulasi

Pada penelitian ini akan dilakukan suatu simulasi, dimana akan digunakan 3 buah *node*, dimana masing-masing *node* akan menjalankan aplikasi OpenNSIS (*Server QoS NSLP*) dan 1 *node* sebagai *bandwidth monitoring*. Model QoS akan dijalankan dalam simulasi ini adalah *IntServ Controlled Load Service* (CLS) pada ClassRMF. Beberapa parameter yang dikonfigurasi pada masing-masing *node* pada file nsis.conf, diantaranya

- *Maximum overall bandwidth* = 4 Mbit
- *Minimum bandwidth unclassified traffic* = 100 kbit
- *Maximum bandwidth unclassified traffic* = 2 Mbit

Dimana nantinya parameter tersebut digunakan sebagai parameter standar pada setiap skenario yang disimulasikan, nilai rendah tersebut dimaksudkan untuk melihat perilaku tiap trafik ketika nantinya dijalankan sebagai trafik yang tidak berkelas, atau dikatakan memanfaatkan *bandwidth* yang tersisa pada jaringan *internet* yang menerapkan protokol NSIS.

Data yang digunakan dalam simulasi ini (yang digunakan untuk trafik HTTP, FTP, dan *video streaming*) adalah sebuah file *video*. Dapat dilihat pada lampiran 1 dan lampiran 2 kriteria secara detail. Pada Tabel 3.1 terdapat parameter *file video* yang digunakan dalam simulasi untuk trafik *video streaming*.

Tabel 3.1. Parameter *file video* untuk trafik *video streaming*

	Video 1	Video 2
Movie name	streetking	trimas
Format	AVI	Windows Media
File size	697 MB	52.0 MB
Bit rate	926 Kbps	124 Kbps
Width	608 pixels	320 pixels
Height	256 pixels	240 pixels

Dalam penelitian ini akan dijalankan 6 skenario pengujian, diantaranya terdapat 3 skenario tanpa dilakukan reservasi dan 3 skenario dengan dilakukan proses reservasi. Dimana pada tiap skenario akan dijalankan 2 simulasi yang berbeda, perbedaan terletak pada jenis trafik *video streaming* yang dilewatkan, pada simulasi pertama trafik *video streaming* menggunakan *video* versi 1 dengan *format* AVI dan *bit rate* sebesar 926 Kbps, sedangkan pada simulasi kedua menggunakan *video* versi 2 dengan *format* *Windows Media* dan *bit rate* 124 Kbps. Kedua simulasi tersebut digunakan untuk setiap skenario yang dijalankan. Untuk trafik FTP dan trafik HTTP menggunakan *video* versi 1 sebagai data yang digunakan pada simulasi ini.

Dibawah ini skenario pengujian tanpa melakukan reservasi suatu *bandwidth*:

Tabel 3.2. Skenario simulasi tanpa reservasi

Parameter	Bandwidth
maximum overall	4 Mbit/s
maximum unclass traffic	2 Mbit/s
minimum unclass traffic	100 kbit/s

1. Skenario pertama

Pada skenario ini masing-masing trafik dijalankan secara bersamaan dengan selang waktu yang telah ditentukan, dimana selang waktu tersebut adalah 10 menit untuk tiap trafik yang dijalankan. Dimulai menit awal dijalankan trafik HTTP, disaat menit ke-10 dijalankan trafik FTP, sehingga pada periode tersebut terdapat dua trafik yang berjalan bersamaan, lalu pada menit ke-20 dilewatkan trafik *video streaming*, sehingga pada menit ke-20 ketiga trafik tersebut berjalan bersamaan, simulasi ini dilakukan selama 30 menit.

2. Skenario kedua

Pada skenario ini akan dilakukan simulasi seperti halnya pada simulasi pertama, hanya urutan jenis trafik yang berbeda dari simulasi pertama. Pada simulasi ini akan dijalankan trafik FTP, lalu trafik *video streaming*, dan terakhir HTTP.

3. Skenario ketiga

Pada simulasi ketiga akan dilakukan simulasi seperti pada simulasi pertama, yang berbeda hanya pada urutan jenis trafiknya (*video streaming*, HTTP, dan FTP).

Dibawah ini skenario pengujian dengan melakukan reservasi suatu *bandwidth*:

Tabel 3.3. Skenario simulasi dengan reservasi

Parameter	Bandwidth
maximum overall	4 Mbit/s
maximum unclass traffic	2 Mbit/s
minimum unclass traffic	100 kbit/s
reservation	3.2 Mbit/s

1. Skenario pertama

Pada skenario pertama ini akan dilakukan reservasi *bandwidth*, dimana reservasi tersebut meminta *bandwidth* sebesar 3.2 Mbit, dengan *bandwidth* yang tersisa hanya 0.8 Mbit. Dengan skenario yang sama seperti pada simulasi pertama, trafik-trafik dijalankan dengan selang waktu 10 menit secara bersamaan, dengan urutan trafik HTTP, FTP, dan *video streaming*.

2. Skenario kedua

Pada simulasi yang kedua, akan dilakukan reservasi *bandwidth* seperti pada skenario pertama, yang membedakan hanya pada urutan trafik yang dijalankan yaitu pada skenario kelima adalah FTP, *video streaming*, dan HTTP.

3. Skenario ketiga

Pada simulasi yang ketiga, akan dilakukan reservasi *bandwidth* seperti pada skenario pertama, dengan urutan trafik *video streaming*, HTTP, dan FTP.

Pada *node* NF (bertindak sebagai *router*) akan dilakukan *capture bandwidth* yang digunakan oleh trafik yang tidak berkelas (FTP, *video streaming*, dan HTTP). *Capture* dari *bandwidth* dilakukan dengan menggunakan *tool* PRTG (*Paessler Router Traffic Grapher*) sebagai *bandwidth* monitoring pada *router*, dan script *traffic control* [20] sebagai monitoring pada *iptables* untuk melihat proses reservasi yang telah dilakukan oleh OpenNSIS.

3.4 Parameter Kinerja

Parameter kinerja yang diukur pada simulasi ini antara lain:

1. *Throughput* dari masing-masing jenis trafik
2. *Aggregate* dari keseluruhan jenis trafik diwaktu bersamaan.

BAB IV PEMBAHASAN

Penelitian ini digunakan untuk menganalisa perilaku trafik tidak berkelas pada protokol NSIS. Simulasi ini dilakukan dengan melibatkan beberapa jenis trafik, yang dikelompokkan dalam trafik yang tidak berkelas. Perilaku trafik akan diamati, baik pada trafik secara individu maupun secara *aggregate*.

4.1 Trafik Tanpa Reservasi pada Protokol NSIS

Pada simulasi ini akan dilakukan pengujian yang melibatkan 3 jenis trafik, diantaranya HTTP, FTP, dan *video streaming*. Ketiga jenis trafik akan dijalankan secara berurutan dengan masing-masing selang antar trafik adalah 10 menit tanpa dilakukan proses reservasi terlebih dahulu. Pada masing-masing *node* dijalankan protokol NSIS, dimana tiap *node* diset maksimum *bandwidth* untuk trafik tidak berkelas sebesar 2 Mbit. Parameter pengujian dapat dilihat pada Tabel 4.1 dibawah ini untuk konfigurasi pada masing-masing *node*.

Tabel 4.1. Parameter pengujian tanpa reservasi

Parameter	Bandwidth
maximum overall	4 Mbit/s
maximum unclass traffic	2 Mbit/s
minimum unclass traffic	100 kbit/s

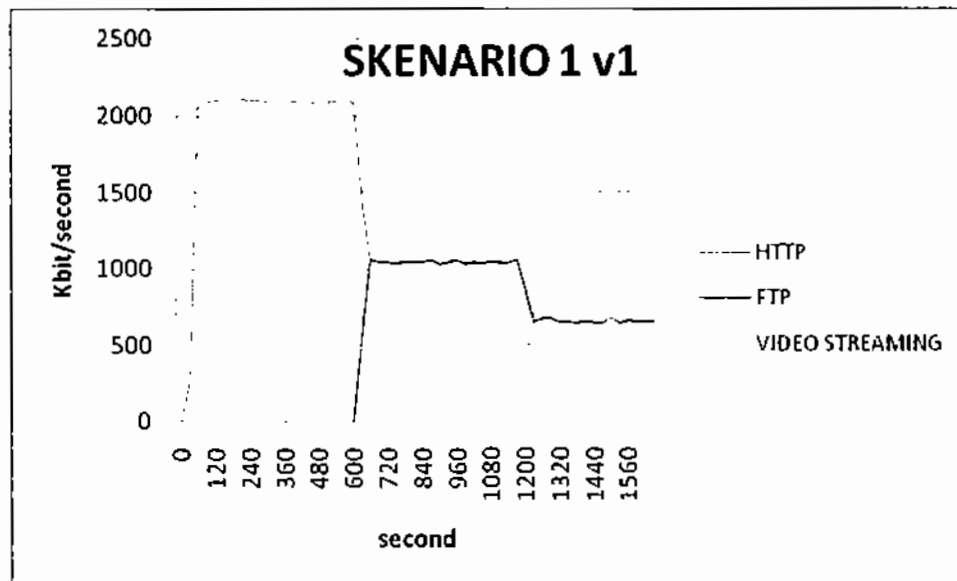
Sedangkan parameter untuk pengujian trafik yang digunakan dapat dilihat pada Tabel 4.2 untuk simulasi pertama dan kedua untuk tiap skenario yang akan dijalankan.

Tabel 4.2. Parameter *file video* untuk trafik *video streaming*

	Simulasi 1	Simulasi 2
Data	Video 1	Video 2
Format	AVI	Windows Media
Bit rate	926 Kbps	124 Kbps

Skenario pertama dilakukan dengan menjalankan trafik HTTP dari menit ke-0 sampai menit ke-30, pada menit ke-10 dilewatkan trafik FTP, sehingga kedua trafik berjalan bersamaan pada kurun waktu tersebut, pada menit ke-20 dijalankan

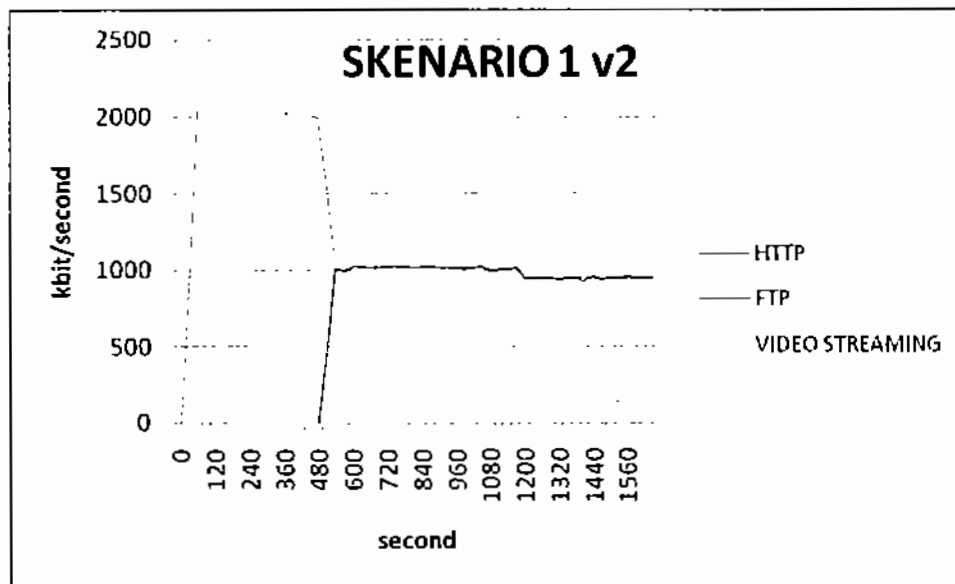
trafik *video streaming*, sehingga ketiga trafik berjalan bersamaan. *Video 1* digunakan sebagai data pada trafik HTTP, FTP, dan *video streaming*.



Gambar 4.1. Simulasi 1 pada skenario 1

Hasil dari simulasi pertama dengan menggunakan *video* versi 1 pada Gambar 4.1 menunjukkan bahwa perilaku trafik HTTP dan FTP dapat dipengaruhi oleh perilaku trafik *video streaming*. Trafik *video streaming* yang menggunakan data *video* versi 1 dengan *bit rate* 926 kbps memiliki rata-rata nilai *throughput* mendekati *bit rate* yang dimilikinya.

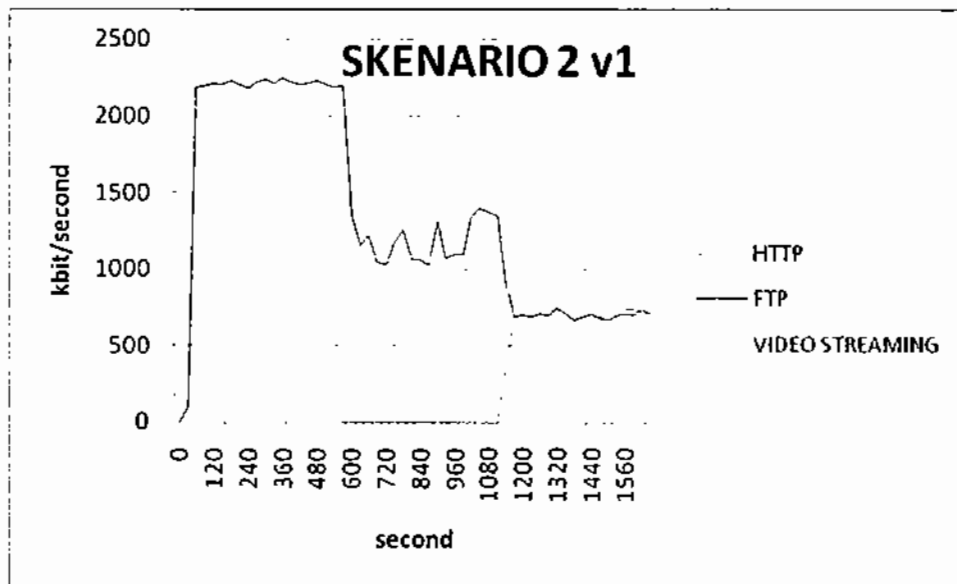
Simulasi kedua pada skenario pertama dijalankan seperti halnya pada simulasi pertama dengan jenis *file video streaming-video* versi 2 dengan *bit rate* lebih rendah sebesar 124 kbps dibandingkan pada simulasi pertama yang menggunakan *bit rate* 926 kbps.



Gambar 4.2. Simulasi 2 pada skenario 1

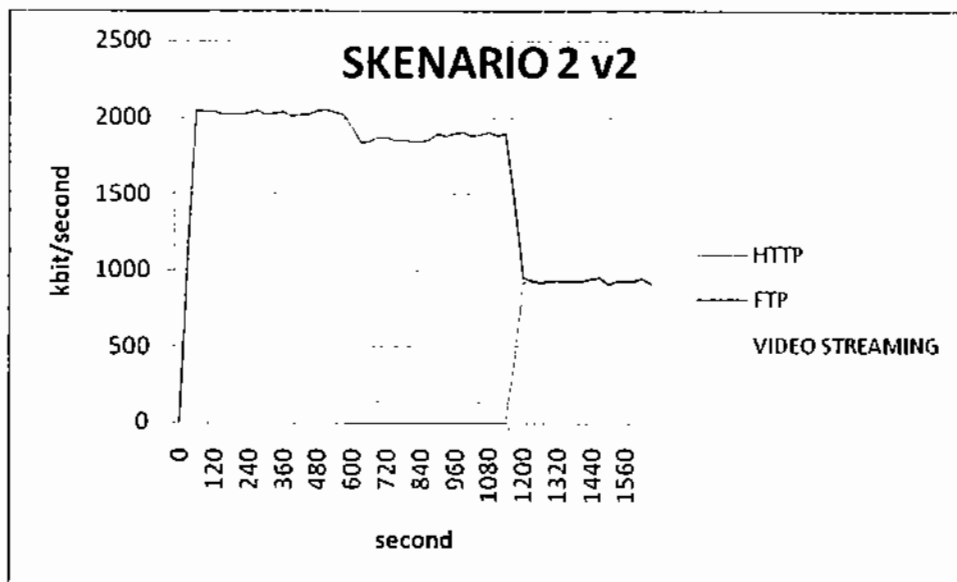
Pada Gambar 4.2 hasil simulasi kedua pada skenario pertama menunjukkan perilaku dari trafik *video streaming* yang tidak terlalu dominan dalam mempengaruhi dari perilaku trafik HTTP dan FTP, nilai rata-rata *throughput* dari trafik *video streaming* mendekati *bit rate* yang dimilikinya sebesar 124 kbps, jika dibandingkan hasil simulasi antara Gambar 4.1 dan Gambar 4.2 menunjukkan perilaku dari *video streaming* dipengaruhi oleh besarnya *bit rate* dari data yang dilewatkan, semakin tinggi nilai suatu *bit rate* suatu data pada trafik *video streaming* yang dilewatkan maka akan semakin kuat mempengaruhi perilaku dari trafik yang lainnya.

Pada skenario kedua untuk simulasi pertama dan kedua, akan dijalankan trafik FTP pada menit ke-0, menit ke-10 dilewatkan trafik *video streaming*, sehingga kedua trafik berjalan bersamaan pada kurun waktu tersebut, pada menit ke-20 dijalankan trafik HTTP. Dengan parameter kedua simulasi seperti pada Tabel 4.2.



Gambar 4.3. Simulasi 1 pada skenario 2

Pada Gambar 4.3 memperlihatkan perilaku dari trafik HTTP, FTP, dan *video streaming*, dimana terjadi perubahan pada pola perilaku trafik FTP ketika dilewatkan trafik *video streaming* secara bersamaan, yang pada awalnya terlihat konstan menjadi tidak beraturan mengikuti pola dari trafik *video streaming* dan ketika trafik HTTP dilewatkan pada saat kedua trafik berjalan bersamaan terjadi perubahan perilaku menjadi lebih konstan dibandingkan perilaku sebelumnya.

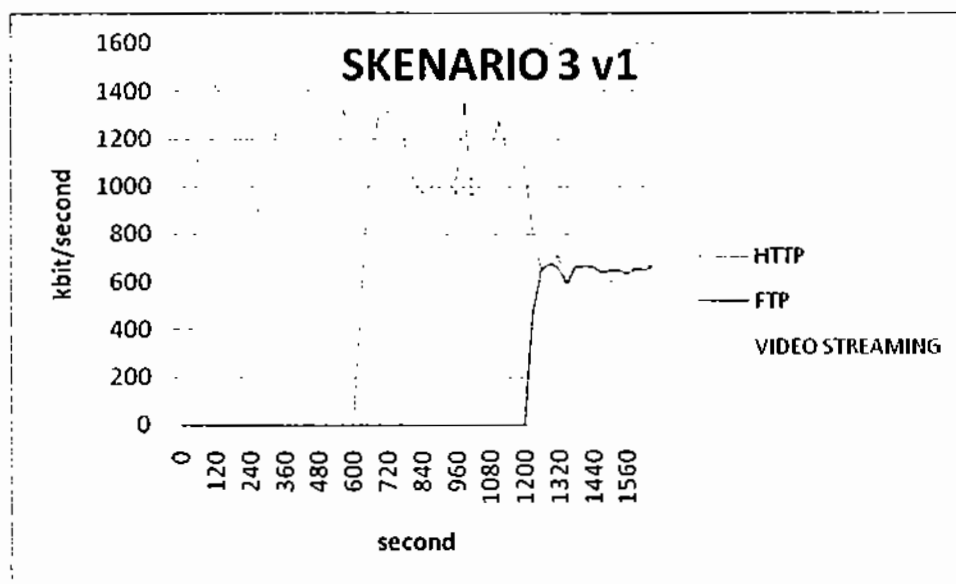


Gambar 4.4. Simulasi 2 pada skenario 2

Pada Gambar 4.4 memperlihatkan hasil yang berbeda dari perilaku trafik HTTP, FTP, dan *video streaming* yang ditunjukkan pada Gambar 4.3, dimana

hasil simulasi pada Gambar 4.3 trafik *video streaming* memiliki dominasi yang kuat dalam mempengaruhi dari trafik HTTP dan FTP. Pada Gambar 4.4 perilaku dari trafik *video streaming* yang memiliki *bit rate* 124 kbps memiliki nilai rata-rata *throughput* yang konstan. Jika dibandingkan hasil yang diperoleh dari simulasi pada skenario 2 pada Gambar 4.3 dan Gambar 4.4 menunjukkan perilaku trafik *video streaming* antara *bit rate* dengan nilai 926 kbps dengan *bit rate* 124 kbps, *bit rate* dengan nilai 124 kbps memiliki nilai rata-rata *throughput* yang lebih konstan dibandingkan dengan trafik *video streaming* yang memiliki *bit rate* 926 kbps.

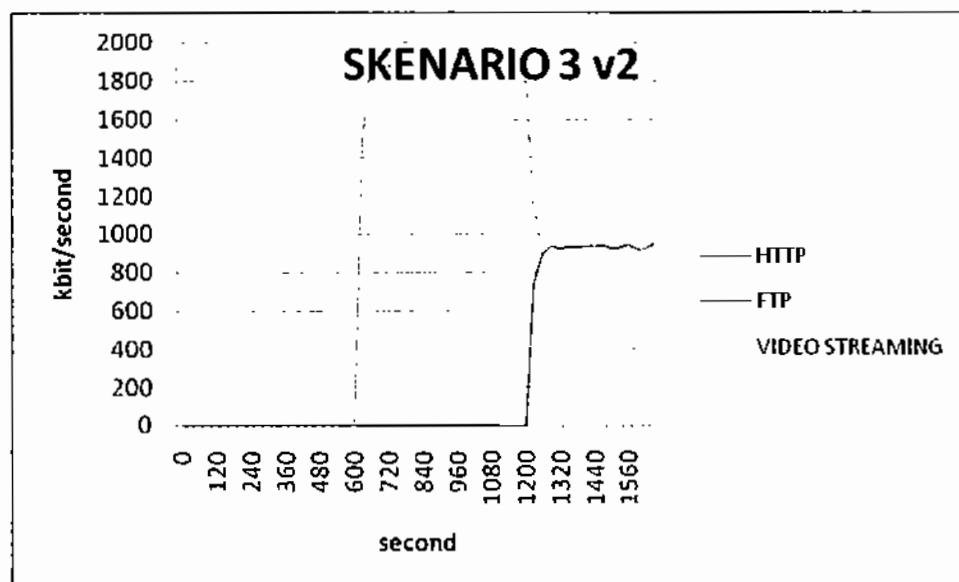
Pada skenario ketiga akan dilakukan 2 simulasi dengan menjalankan trafik *video streaming* dari menit ke-0, pada menit ke-10 dilewatkan trafik HTTP, sehingga kedua trafik berjalan bersamaan pada kurun waktu tersebut, pada menit ke-20 dijalankan trafik FTP.



Gambar 4.5. Simulasi 1 pada skenario 3

Hasil simulasi 1 pada skenario 3 pada Gambar 4.5 dapat memperlihatkan pengaruh dari trafik *video streaming*, dimana trafik HTTP pada simulasi-simulasi sebelumnya terlihat konstan, akan tetapi pada simulasi pertama dengan skenario ketiga, trafik *video streaming* ketika dijalankan pada menit awal simulasi terlihat mendominasi jalannya simulasi, dimana ketika trafik HTTP dijalankan pada menit ke-10, perilaku trafik tersebut mengikuti pola perilaku dari trafik *video streaming*

dan pada akhirnya perilaku dari trafik keduanya menjadi lebih konstan ketika pada menit ke-20 trafik FTP dilewatkan.



Gambar 4.6. Simulasi 2 pada skenario 3

Hasil simulasi kedua pada skenario ke 3 pada Gambar 4.6, memperlihatkan trafik *video streaming* dengan *bit rate* 124 kbps dari mulai menit 0 sampai menit ke-30 berjalan konstan, dimana trafik HTTP dan FTP tidak dapat mempengaruhi perilaku dari trafik *video streaming*. Berbeda hasil yang didapat ketika trafik *video streaming* dengan *bit rate* 926 kbps perilakunya tidak konstan tetapi tidak beraturan.

4.2 Trafik dengan Reservasi pada Protokol NSIS

Pada simulasi ini akan dilakukan reservasi *bandwidth* oleh NSIS *Initiator* (NI) sebesar 3.2 Mbit kepada NSIS *Receiver* (NR), sehingga *bandwidth* tersisa hanya sebesar 0.8 Mbit, Parameter keseluruhan dapat dilihat pada tabel 4.3 dibawah ini.

Tabel 4.3. Parameter pengujian dengan reservasi

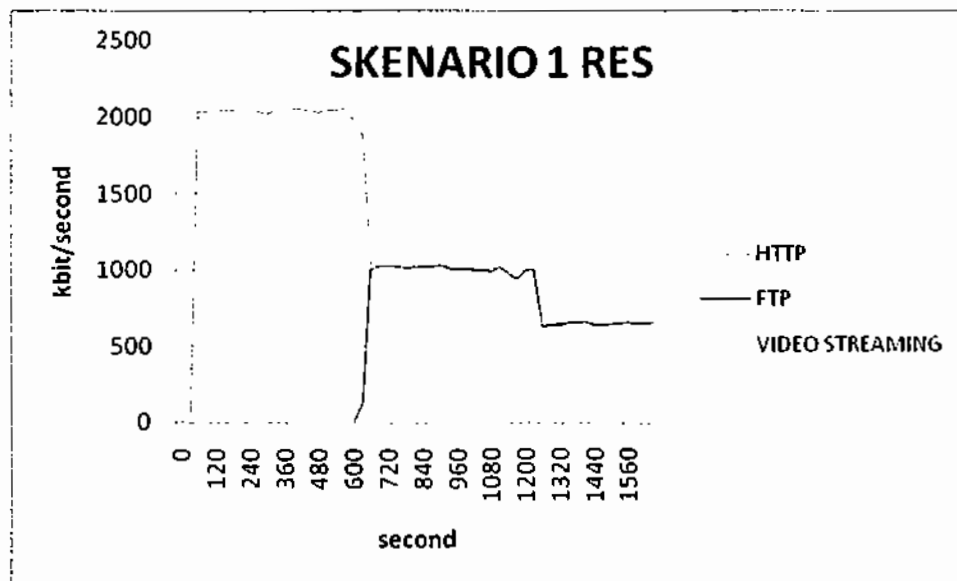
Parameter	Bandwidth
maximum overall	4 Mbit/s
maximum unclass traffic	2 Mbit/s
minimum unclass traffic	100 kbit/s
reservation	3.2 Mbit/s

Sedangkan parameter untuk pengujian trafik yang digunakan dapat dilihat pada Tabel 4.4 untuk simulasi pertama dan kedua untuk tiap skenario yang akan dijalankan.

Tabel 4.4. Parameter *file video* untuk trafik *video streaming*

	Simulasi 1	Simulasi 2
Data	Video 1	Video 2
Format	AVI	Windows Media
Bit rate	926 Kbps	124 Kbps

Pada skenario pertama akan dilakukan 2 simulasi, dimana simulasi akan dilakukan dengan trafik HTTP dilewatkan pada jaringan sejak menit ke-0, setelah menit ke-10 dilewatkan kemudian trafik FTP, dan pada menit ke-20 dilewatkan trafik selanjutnya *video streaming*, yang membedakan antara simulasi dan kedua adalah jenis *video* yang akan dilewatkan pada trafik *video streaming*. Simulasi pertama menggunakan *file video* dengan bit rate 926 kbps dan simulasi kedua menggunakan *file video* dengan bit rate 124 kbps.

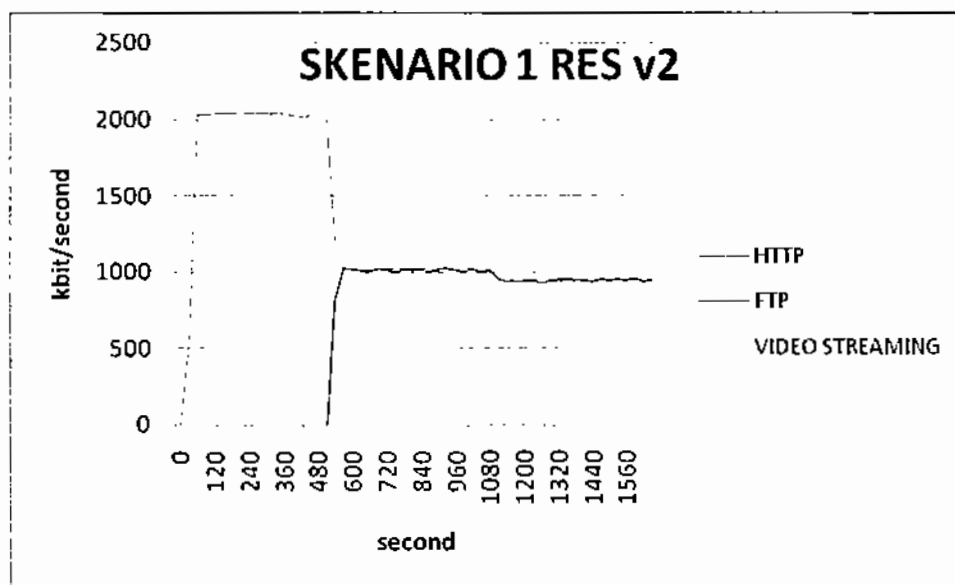


Gambar 4.7. Simulasi 1 pada skenario 1 dengan reservasi

Hasil simulasi pertama pada Gambar 4.7 menunjukkan bahwa reservasi yang dilakukan sebesar 3.2 Mbit tidak memiliki pengaruhnya ketika kelas yang telah

di-reservasi tidak dilewatkan suatu trafik. Hal ini disebabkan oleh penggunaan *Hierarchical Token Bucket* (HTB), sebagai *traffic control*. Perilaku dari trafik *video streaming* menunjukkan konstan ketika trafik HTTP dan trafik FTP dijalankan terlebih dahulu.

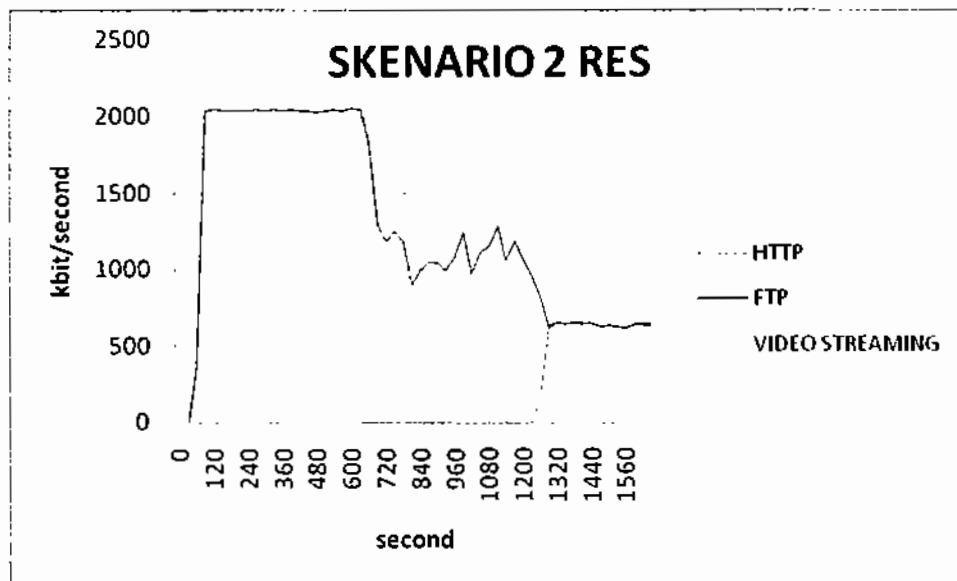
Konsep dari *Hierarchical Token Bucket* (HTB), dimana HTB tersebut memiliki parameter *ceil* sehingga kelas selalu mendapatkan *bandwidth* diantara *bandwidth base link* dan nilai *ceil*-nya. Setiap kelas dapat meminjamkan *bandwidth* selama *bandwidth* total yang diperoleh dibawah dari nilai *ceil*-nya [18].



Gambar 4.8. Simulasi 2 pada skenario 1 dengan reservasi

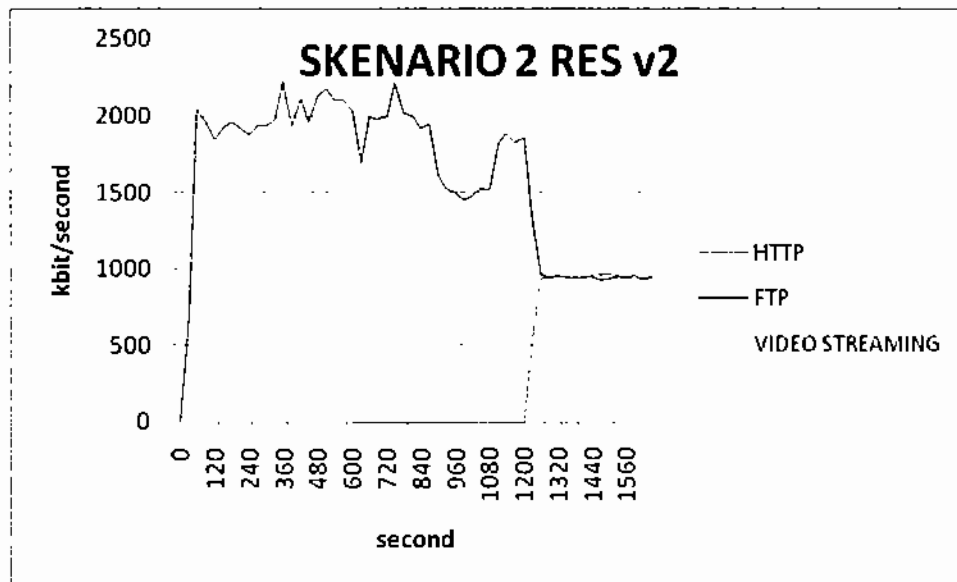
Dari hasil simulasi 1 dan simulasi 2 pada skenario 1 pada Gambar 4.7 dan Gambar 4.8, perilaku dari trafik HTTP ketika dijalankan pada awal simulasi menunjukkan pemanfaatan dari *bandwidth* yang tersedia penuh dari *bandwidth* yang tersedia. Pengaruh dari perbedaan *bit rate* pada simulasi pertama dan kedua terlihat dimana pada simulasi pertama trafik *video streaming* menggunakan *bit rate* 926 kbps lebih mendominasi dibandingkan dengan simulasi kedua yang menggunakan trafik *video streaming* dengan *bit rate* 124 kbps.

Pada skenario kedua, akan dilakukan 2 simulasi dengan perbedaan *bit rate* pada trafik *video streaming* (dapat dilihat pada Tabel 4.4), trafik FTP dilewatkan sejak menit ke-0, setelah menit ke-10 dilewatkan trafik *video streaming*, dan pada menit ke-20 dilewatkan trafik HTTP.



Gambar 4.9. Simulasi 1 pada skenario 2 dengan reservasi

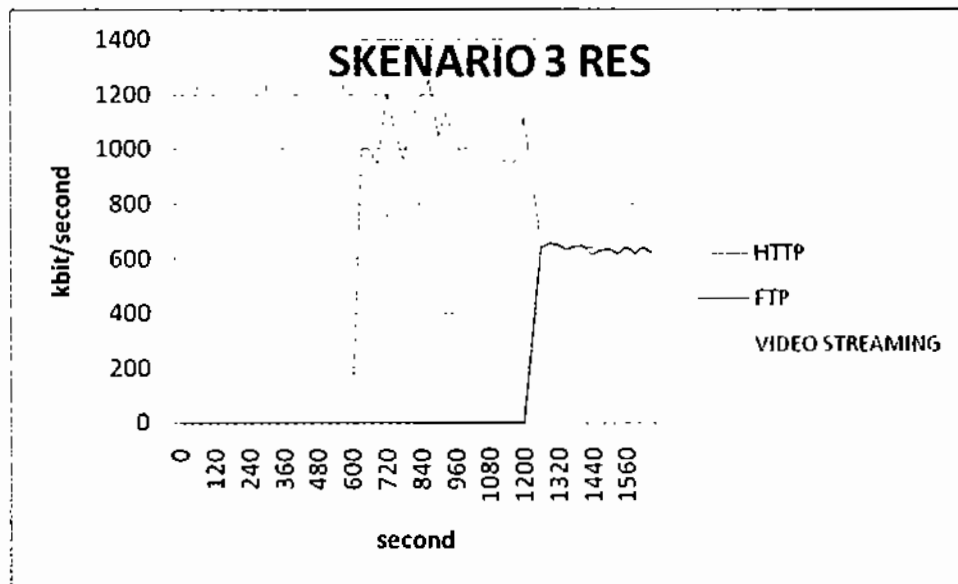
Hasil simulasi pertama pada Gambar 4.9 proses reservasi yang dilakukan menunjukkan hasil yang sama dengan yang didapat pada kedua simulasi pada skenario pertama pada Gambar 4.7 dan Gambar 4.8, dimana HTB sebagai *traffic control* bekerja dalam sistem *bandwidth sharing*. Pada Gambar 4.9 perilaku dari trafik FTP dapat dipengaruhi oleh pola trafik dari *video streaming*, dan ketika trafik HTTP dilewatkan, perilaku dari trafik FTP dan *video streaming* menjadi lebih konstan.



Gambar 4.10. Simulasi 2 pada skenario 2 dengan reservasi

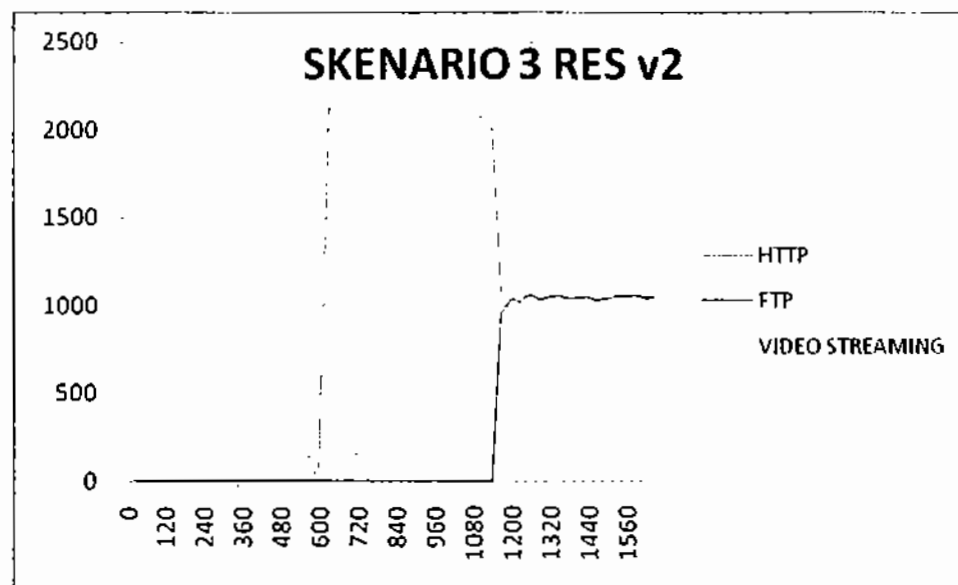
Perbandingan antara hasil dari kedua simulasi pada skenario dua pada Gambar 4.9 dan Gambar 4.10, menunjukkan bahwa perilaku dari trafik FTP dapat dipengaruhi oleh trafik *video streaming* dengan nilai *bit rate* yang tinggi, semakin tinggi nilai suatu *bit rate* dari trafik *video streaming*, maka semakin dominan trafik *video streaming* dapat mempengaruhi trafik HTTP dan trafik FTP.

Pada skenario ketiga, akan dilakukan 2 simulasi dengan parameter untuk simulasi satu dan simulasi kedua pada Tabel IV.4, dimana trafik *video streaming* dilewatkan dari menit awal, setelah menit ke-10 dilewatkan trafik HTTP, dan pada menit ke-20 dilewatkan kembali trafik FTP.



Gambar 4.11. Simulasi 1 pada skenario 3 dengan reservasi

Hasil simulasi pertama pada skenario 3 dengan reservasi pada Gambar 4.11 menunjukkan bahwa pola trafik *video streaming* memiliki dominasi yang tinggi untuk mempengaruhi pola trafik HTTP, sebelum trafik FTP dilewatkan.



Gambar 4.12. Simulasi 2 pada skenario 3 dengan reservasi

Hasil simulasi kedua pada skenario 3 pada Gambar 4.12 menunjukkan bahwa pola trafik *video streaming* tidak dapat mempengaruhi pola trafik HTTP dan FTP, hal ini dikarenakan *bit rate* yang digunakan pada trafik *video streaming* simulasi

kedua ini hanya sebesar 124 kbps, yang pada umumnya *bit rate* sebesar ini digunakan untuk suatu jaringan yang memiliki *bandwidth* jaringan yang terbatas.

Perilaku trafik dari 12 simulasi yang dilakukan dengan kombinasi dari jenis trafik, maka dapat diamati perilaku dari trafik-trafik tidak berkelas pada protokol NSIS, trafik HTTP dan trafik FTP ketika dijalankan pertama kali pada beberapa simulasi memperlihatkan penggunaan keseluruhan dari *bandwidth* yang tersedia secara konstan, berbeda dengan perilaku dari trafik *video streaming* yang tidak menggunakan *bandwidth* secara keseluruhan secara konstan ketika dilewatkan pada awal simulasi. Perilaku dari trafik *video streaming* sangat dipengaruhi oleh nilai dari suatu *bit rate* data yang digunakan, semakin besar *bit rate* suatu trafik *video streaming*, semakin besar dapat mempengaruhi jenis trafik HTTP dan FTP, hal ini dapat memberikan gambaran bahwa trafik *video streaming* yang menggunakan UDP sebagai *transport protocol* dapat mempengaruhi perilaku dari trafik HTTP dan FTP yang menggunakan *transport protocol* TCP dengan syarat nilai suatu *bit rate video* yang dilewatkan.

Pemanfaatan *Hierarchical Token Bucket* (HTB) sebagai *traffic control* pada *ClassRMF* (*QoS Model* (QOSM) *IntServ Controlled Load Service* (CLS) pada *OpenNSIS*) memperlihatkan bagaimana protokol NSIS ingin memberikan suatu kepastian dari suatu reservasi yang dibuatnya mendapatkan jaminan akan ketersediaan *bandwidth* yang diminta oleh pengguna, dimana HTB sendiri memiliki keunggulan akan jaminan terhadap kebocoran dari *bandwidth* [18].

4.3 Hambatan pada penelitian

Hambatan pada penelitian ini adalah aplikasi yang digunakan *OpenNSIS* versi 0.5.0 hanya dapat menjalankan trafik tidak berkelas, dimana untuk proses reservasi hanya mampu melakukan reservasi *bandwidth* kedalam *kernel linux*, tetapi tidak dapat dilewatkan trafik.

BAB V

KESIMPULAN

Hasil dari penelitian perilaku dari trafik HTTP, FTP, dan *video streaming* sebagai trafik tidak berkelas pada protokol *Next Step in Signaling* (NSIS) yang dilakukan yaitu

1. Kombinasi antara trafik HTTP-*video streaming*-FTP dan kombinasi FTP-*video streaming*-HTTP, menunjukkan perilaku trafik *video streaming* yang dapat mempengaruhi trafik HTTP dan FTP.
2. Perilaku dari trafik *video streaming* yang menggunakan UDP sebagai *transport protocol* bersifat dominan dibandingkan perilaku dari Trafik HTTP dan FTP yang menggunakan *transport protocol* TCP, hal ini dipengaruhi oleh besaran dari nilai suatu *bit rate* dari trafik *video streaming*. Semakin besar suatu nilai *bit rate* dari trafik *video streaming* semakin besar dapat mempengaruhi trafik dari HTTP dan FTP.
3. Perilaku trafik HTTP dan FTP pada saat dijalankan pada setiap awal memperlihatkan penggunaan keseluruhan dari *bandwidth* yang tersedia secara konstan, berbeda dengan perilaku dari trafik *video streaming* yang tidak menggunakan *bandwidth* secara keseluruhan secara konstan ketika dilewatkan pada saat awal simulasi.
4. Perbandingan antara hasil simulasi tanpa reservasi *bandwidth* dengan simulasi melakukan reservasi *bandwidth* tidak memperlihatkan perbedaan pada protokol *Next Step in Signaling* (NSIS). Hal ini dikarenakan *bandwidth* yang telah direservasi belum digunakan, sehingga alokasi *bandwidth* tersebut dapat dimanfaatkan oleh kelas, sesuai dengan konsep dari *Hierarchical Token Bucket* (HTB). Dari hasil penelitian tersebut terlihat keterkaitan antara *Hierarchical Token Bucket* (HTB) pada *operating system linux* dengan protokol *Next Step in Signaling* (NSIS).

DAFTAR ACUAN

- [1] Dimitrios Miras (2002). *Network QoS Needs of Advanced Internet Applications*. Diakses 30 September 2008, dari Internet2 QoS Working Group.
<http://qos.internet2.edu/wg/apps/fellowship/Docs/Internet2AppsQoSNeeds.pdf>
- [2] X. Fu, H. Schulzrinne, H. Tschofenig, C. Dickmann, D. Hogrefe, "Overhead and Performance Study of the General Internet Signaling Transport (GIST) Protocol," *To be published in ACM/IEEE Transactions on Networking*, April 2009 (revised version of the INFOCOM 2006 paper). Diakses 30 April 2008, dari Georg-August-University of Goettingen.
http://user.informatik.uni-goettingen.de/~fu/GIST_Perf_ToN_final.pdf
- [3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services," Internet Engineering Task Force, RFC 2475, Desember 1998.
- [4] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview," Internet Engineering Task Force, RFC 1633, Juni 1994.
- [5] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service," Internet Engineering Task Force, RFC 2212, September 1997.
- [6] J. Wroclawski, "Specification of the Controlled-Load Network Element Service," Internet Engineering Task Force, RFC 2211, September 1997.

- [7] L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP)-Version 1 Functional Specification," Internet Engineering Task Force, RFC 2205, September 1997.
- [8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services," Internet Engineering Task Force, RFC 2475, Desember 1998.
- [9] R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch, "Next Steps in Signaling: Framework," Internet Engineering Task Force, RFC 4080, Juni 2005.
- [10] Christian Dickmann. "*An Implementation and Evaluation of the General Internet Signaling Transport (GIST) Protocol.*" Thesis, Program Magister Fakultas Informatika, Georg-August University of Goettingen, German, September 2005.
- [11] N. Steinleitner, X. Fu, D. Hogrefe, H. Tschofenig, T. Schreck, "An NSIS-based Approach for Firewall Traversal in Mobile IPv6 Networks," *ICST International Wireless Internet Conference (WICON) 2007*, ACM Press, Oct 2007.
- [12] Bernd Schlör. "*Implementation and Performance Evaluation of the IETF QoS NSLP Protocol.*" Thesis, Program Magister Fakultas Informatika, Georg-August University of Goettingen, German, Oktober 2007.
- [13] J. Manner, G. Karagiannis, A. McDonald, "NSLP for Quality-of-Service Signaling," Internet draft (draft-ietf-nsis-qos-nslp-16), work in progress, February 7, 2008.

- [14] X. Fu, H. Schulzrinne, V. Bader, D. Hogrefe, C. Kappler, "NSIS: a new extensible IP signaling protocol suite," *IEEE Communications Magazine*, October 2005 Volume 43, Issue 10, Oct. 2005 Page(s): 133- 141.
- [15] G. Ash, A. Bader, C. Kappler, D. Oran, "QoS NSLP QSPEC Template," draft-ietfnsis-qspec-19, work in progress, August 2008.
- [16] Next Steps In Signaling (NSIS) Implementation by University of Goettingen. Diakses 1 Maret 2008.
<http://user.informatik.uni-goettingen.de/~nsis/download.html>.
- [17] Bert Hubert. *Linux Advanced Routing and Traffic Control*. Diakses 8 Oktober 2008.
<http://lartc.org>
- [18] B. Nyoman, A. Andi, dan P. Eka, "Analisa Perbandingan *Class Based Queuing* (CBQ) dan *Hierarchical Token Bucket* (HTB) Untuk Manajemen *Bandwidth* Pada Jaringan TCP/IP," Diakses 8 Oktober 2008.
http://antix.or.id/pub/efnet/jurnal_cbq_htb.pdf
- [19] M. Arumathurai, X. Fu, B. Schloer, H. Tschofenig, "Performance Study of the NSIS QoS-NSLP Protocol," *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2008), Computer and Communications Network Security Symposium*, New Orleans, LA, USA, December 2008. Diakses 6 November 2008
http://www.net.informatik.uni-goettingen.de/publications/1556/GLOBECOM08_Diameter_WebAuth.pdf
- [20] http://www.docum.org/docum.org/gui/download/monitor_tc_top.pl

File video: streetking.avi

General

Complete name : streetking.avi
Format : AVI
Format/Info : Audio Video Interleave
Format/Family : RIFF
File size : 697 MiB
PlayTime : 1h 44mn
Bit rate : 926 Kbps
StreamSize : 12.8 MiB
Writing library : VirtualDub build 0/release

Video #0

Codec : XviD
Codec/Family : MPEG-4
Codec/Info : XviD project
Codec profile : Streaming Video Profile/Level 1
Codec settings/PacketBitStream : No
Codec settings/BVOP : Yes
Codec settings/QPel : No
Codec settings/GMC : 0
Codec settings/Matrix : Default
PlayTime : 1h 44mn
Bit rate : 790 Kbps
Width : 608 pixels
Height : 256 pixels
Display Aspect ratio : 2.35
Frame rate : 25.000 fps
Resolution : 8 bits
Chroma : 4:2:0
Interlacement : Progressive

(Lanjutan)

Bits/(Pixel*Frame) : 0.201

StreamSize : 590 MiB

Audio #1

Codec : MPEG-1 Audio layer 3

Codec profile : Joint stereo

PlayTime : 1h 44mn

Bit rate : 127 Kbps

Bit rate mode : VBR

Channel(s) : 2 channels

Sampling rate : 48 KHz

Resolution : 16 bits

StreamSize : 94.5 MiB

Writing library : LAME3.96r

Encoding settings : ABR

Lampiran 2: Spesifikasi *File Video* 2

File video: trimas.wmv

General

Complete name: trimas.wmv
Format : Windows Media
File size : 52.0 MiB
PlayTime : 58mn 43s
Bit rate : 124 Kbps
Movie name : Untitled
Application : Windows Movie Maker 6.0.6000.16386
WM/ToolVersion : 6.0.6000.16386
WM/ToolName : Windows Movie Maker
PacketCount : 37722
PacketSize : 1444
Date_Created : UTC 2098-08-31 20:47:49
HeaderSize : 5785
DataSize : 54470618

Video

Codec : WMV3
Codec/Info : Windows Media Video 9
Bit rate : 99 Kbps
Width : 320 pixels
Height : 240 pixels
Display Aspect ratio : 4/3
Resolution : 24 bits

Audio

Codec : WMA2
Codec/Info : Windows Media Audio 2
Bit rate : 20 Kbps
Channel(s) : 1 channel
Sampling rate : 22 KHz

Lampiran 3: Data Simulasi 1 Tanpa Reservasi Skenario 1

	HTTP	FTP	VIDEO STREAMING	TOTAL
0	0	0	0	0
30	282,314	0	0	282,314
60	2.055,44	0	0	2055,435
90	2.086,00	0	0	2086,001
120	2.095,15	0	0	2095,153
150	2.104,33	0	0	2104,33
180	2.071,15	0	0	2071,146
210	2.111,79	0	0	2111,787
240	2.093,18	0	0	2093,177
270	2.092,68	0	0	2092,68
300	2.086,53	0	0	2086,529
330	2.073,09	0	0	2073,09
360	2.102,24	0	0	2102,235
390	2.085,35	0	0	2085,352
420	2.097,44	0	0	2097,437
450	2.089,39	0	0	2089,389
480	2.071,99	0	0	2071,991
510	2.087,47	0	0	2087,467
540	2.088,44	0	0	2088,44
570	2.103,86	0,065	0	2103,927
600	2.083,07	0,294	0	2083,364
630	1.404,02	613,117	0	2017,138
660	1.039,58	1.062,59	0	2102,171
690	1.045,48	1.045,75	0	2091,234
720	1.033,91	1.049,97	0	2083,875
750	1.046,93	1.030,16	0	2077,09
780	1.047,83	1.042,17	0	2089,992
810	1.050,11	1.036,95	0	2087,066
840	1.050,94	1.037,57	0	2088,506
870	1.061,11	1.056,27	0	2117,376
900	1.031,55	1.031,12	0	2062,67
930	1.052,84	1.037,60	0	2090,438
960	1.052,73	1.052,69	0	2105,426
990	1.035,34	1.032,89	0	2068,229
1020	1.048,42	1.040,76	0	2089,183
1050	1.039,81	1.037,64	0	2077,455
1080	1.050,01	1.042,32	0	2092,329
1110	1.051,49	1.036,34	0	2087,827
1140	1.040,90	1.037,60	0	2078,502
1170	1.062,12	1.059,56	0	2121,673
1200	853,617	849,883	367,771	2071,271
1230	663,74	654,382	716,165	2034,287
1260	662,552	677,556	728,961	2069,069
1290	676,876	679,807	709,219	2065,902
1320	649,516	652,223	739,562	2041,301

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1350	657,355	653,566	664,109	1975,03
1380	659,394	643,29	731,52	2034,204
1410	666,094	652,243	737,734	2056,071
1440	649,131	652,819	737,369	2039,319
1470	662,236	643,941	683,995	1990,172
1500	703,986	673,586	647,803	2025,375
1530	654,654	645,493	710,919	2011,066
1560	678,692	657,206	694,596	2030,494
1590	656,938	653,114	715,069	2025,121
1620	666,097	659,265	736,638	2062
1650	668,027	656,361	746,874	2071,262

Lampiran 4: Data Simulasi 2 Tanpa Reservasi Skenario 1

	HTTP	FTP	VIDEO STREAMING	TOTAL
0	0	0	0	0
30	1.005,77	0	0	1005,767
60	2.035,82	0	0	2035,821
90	2.030,21	0	0	2030,209
120	2.012,20	0	0	2012,195
150	2.031,34	0	0	2031,339
180	2.020,07	0	0	2020,067
210	2.027,70	0	0	2027,7
240	2.011,64	0	0	2011,637
270	2.040,40	0	0	2040,404
300	2.038,81	0	0	2038,813
330	2.016,89	0	0	2016,894
360	2.020,83	0	0	2020,833
390	2.020,00	0	0	2019,999
420	2.044,22	0	0	2044,224
450	2.024,00	0,065	0	2024,066
480	1.988,83	1,588	0	1990,414
510	1.518,22	509,399	0	2027,622
540	1.041,53	1.005,69	0	2047,225
570	976,941	1.003,95	0	1980,889
600	1.023,91	1.023,47	0	2047,373
630	1.017,34	1.025,03	0	2042,371
660	1.016,22	1.016,70	0	2032,917
690	1.030,64	1.017,52	0	2048,158
720	1.014,70	1.023,82	0	2038,517
750	1.021,07	1.022,49	0	2043,555
780	1.029,64	1.013,92	0	2043,56
810	1.016,24	1.018,51	0	2034,748
840	1.022,42	1.014,72	0	2037,139
870	1.022,93	1.022,36	0	2045,295
900	1.017,13	1.019,42	0	2036,544
930	1.015,66	1.020,60	0	2036,262
960	1.016,61	1.015,74	0	2032,357
990	1.009,34	1.020,30	0	2029,646
1020	1.020,92	1.012,79	0	2033,707
1050	1.016,67	1.021,79	0	2038,458
1080	1.011,56	992,322	0	2003,885
1110	996,433	1.006,46	0	2002,893
1140	1.019,74	1.004,05	0	2023,785
1170	1.033,28	1.015,85	0	2049,13
1200	936,571	951,698	40,213	1928,482
1230	944,52	948,764	135,629	2028,913
1260	938,814	955,989	135,264	2030,067

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1290	935,044	946,99	135,995	2018,029
1320	948,095	945,005	137,503	2030,603
1350	949,707	950,249	134,167	2034,123
1380	954,481	952,265	134,532	2041,278
1410	954,222	929,827	137,823	2021,872
1440	946,842	962,159	133,07	2042,071
1470	949,946	943,861	134,167	2027,974
1500	944,143	954,925	136,04	2035,108
1530	950,943	951,377	135,264	2037,584
1560	948,393	963,775	133,436	2045,604
1590	944,896	956,614	136,36	2037,87
1620	953,281	947,954	135,629	2036,864
1650	959,371	953,028	133,436	2045,835

Lampiran 5: Data Simulasi 1 Tanpa Reservasi Skenario 2

	HTTP	FTP	VIDEO STREAMING	TOTAL
0	0	0	0	0
30	0	101,872	0	101,872
60	0	2.186,81	0	2186,814
90	0	2.195,29	0	2195,293
120	0	2.213,65	0	2213,654
150	0	2.208,29	0	2208,288
180	0	2.225,39	0	2225,387
210	0	2.203,36	0	2203,364
240	0	2.181,24	0	2181,237
270	0	2.218,98	0	2218,984
300	0	2.238,06	0	2238,06
330	0	2.209,24	0	2209,239
360	0	2.246,32	0	2246,317
390	0	2.223,64	0	2223,637
420	0	2.205,02	0	2205,02
450	0	2.211,29	0	2211,29
480	0	2.227,45	0	2227,446
510	0	2.206,05	0	2206,047
540	0	2.191,23	0	2191,233
570	0	2.192,15	0	2192,151
600	0	1.349,39	768,809	2118,199
630	0	1.158,40	984,865	2143,264
660	0	1.221,95	948,623	2170,576
690	0	1.050,11	1.117,57	2167,676
720	0	1.028,73	1.096,00	2124,725
750	0	1.170,34	1.004,97	2175,307
780	0	1.255,07	931,49	2186,561
810	0	1.067,07	1.107,70	2174,771
840	0	1.063,25	1.075,16	2138,415
870	0	1.032,14	1.117,58	2149,714
900	0	1.318,54	784,528	2103,064
930	0	1.074,98	1.073,33	2148,309
960	0	1.096,53	1.034,22	2130,743
990	0	1.096,24	1.028,00	2124,247
1020	0	1.337,90	802,076	2139,972
1050	0	1.400,09	782,23	2182,315
1080	0	1.371,19	831,322	2202,515
1110	0	1.348,12	838,268	2186,386
1140	315,103	920,801	885,428	2121,332
1170	691,185	699,617	802,076	2192,878
1200	694,656	706,043	782,7	2183,399
1230	698,738	685,783	784,528	2169,049
1260	708,975	712,086	759,191	2180,252

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1290	690,969	703,546	749,067	2143,582
1320	746,99	745,84	659,867	2152,697
1350	715,115	716,566	788,915	2220,596
1380	680,6	664,839	801,71	2147,149
1410	688,683	689,907	781,969	2160,559
1440	706,832	709,126	758,46	2174,418
1470	685,413	678,228	765,884	2129,525
1500	697,017	673,263	770,271	2140,551
1530	717,491	693,837	769,905	2181,233
1560	737,666	715,967	776,486	2230,119
1590	736,995	704,012	743,218	2184,225
1620	717,868	736,618	735,541	2190,027
1650	711,81	716,243	759,191	2187,244

Lampiran 6: Data Simulasi 2 Tanpa Reservasi Skenario 2

	HTTP	FTP	VIDEO STREAMING	TOTAL
0	0	0	0	0
30	0	1.191,51	0	1191,505
60	0	2.047,13	0	2047,128
90	0	2.032,92	0	2032,917
120	0	2.039,47	0	2039,47
150	0	2.019,65	0	2019,649
180	0	2.018,12	0	2018,117
210	0	2.023,13	0	2023,132
240	0	2.031,34	0	2031,337
270	0	2.048,67	0	2048,666
300	0	2.020,81	0	2020,805
330	0	2.029,43	0	2029,429
360	0	2.033,33	0	2033,328
390	0	2.008,00	0	2007,995
420	0	2.019,38	0	2019,38
450	0	2.019,86	0	2019,863
480	0	2.042,97	0	2042,971
510	0	2.053,57	0	2053,57
540	0	2.035,36	0	2035,361
570	0	2.022,93	0	2022,929
600	0	1.937,53	53,374	1990,9
630	0	1.835,20	135,264	1970,46
660	0	1.840,93	136,406	1977,338
690	0	1.865,12	138,188	2003,308
720	0	1.868,15	134,898	2003,051
750	0	1.849,51	135,629	1985,139
780	0	1.849,55	137,091	1986,638
810	0	1.844,82	136,36	1981,181
840	0	1.846,75	135,264	1982,016
870	0	1.855,78	135,674	1991,458
900	0	1.896,65	136,726	2033,375
930	0	1.877,23	135,995	2013,228
960	0	1.895,35	138,188	2033,538
990	0	1.898,80	135,264	2034,067
1020	0	1.877,14	137,823	2014,961
1050	0	1.884,93	135,309	2020,241
1080	0	1.902,13	139,285	2041,412
1110	0	1.873,44	132,339	2005,776
1140	0	1.896,71	135,995	2032,702
1170	388,539	1.448,77	140,747	1978,052
1200	925,488	950,234	137,091	2012,813
1230	922,302	942,404	134,212	1998,918
1260	928,453	917,699	132,339	1978,491

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1290	934,098	923,767	139,285	1997,15
1320	942,112	926,501	133,07	2001,683
1350	928,161	930,489	137,457	1996,107
1380	925,897	934,042	134,532	1994,471
1410	930,963	932,502	136,36	1999,825
1440	944,318	950,448	136,771	2031,537
1470	950,253	956,114	138,188	2044,555
1500	904,652	911,734	133,07	1949,456
1530	939,428	930,007	138,554	2007,989
1560	940,032	930,778	134,532	2005,342
1590	946,995	929,163	137,823	2013,981
1620	955,771	945,608	135,309	2036,688
1650	925,177	915,883	136,36	1977,42

Lampiran 7: Data Simulasi 1 Tanpa Reservasi Skenario 3

	HTTP	FTP	VIDEO STREAMING	TOTAL
0	0	0	0	0
30	0	0	355,825	355,825
60	0	0	1.120,86	1120,859
90	0	0	1.025,81	1025,809
120	0	0	1.440,01	1440,008
150	0	0	1.219,57	1219,565
180	0	0	1.050,67	1050,668
210	0	0	1.412,33	1412,33
240	0	0	1.076,26	1076,259
270	0	0	887,256	887,256
300	0	0	999,853	999,853
330	0	0	1.227,61	1227,608
360	0	0	1.036,05	1036,045
390	0	0	955,618	955,618
420	0	0	1.136,23	1136,227
450	0	0	1.133,29	1133,289
480	0	0	1.141,70	1141,697
510	0	0	1.115,74	1115,741
540	0	0	694,596	694,596
570	0	0	1.337,65	1337,646
600	0	0	1.006,77	1006,769
630	595,234	0	973,166	1568,4
660	1.063,45	0	927,469	1990,922
690	1.285,44	0	737,003	2022,446
720	1.312,59	0	683,995	1996,586
750	1.306,18	0	703,005	2009,188
780	1.197,91	0	813,775	2011,684
810	1.023,03	0	980,073	2003,098
840	975,994	0	1.014,84	1990,836
870	980,659	0	1.020,33	2000,985
900	1.057,43	0	928,566	1985,999
930	1.051,48	0	955,984	2007,464
960	966,064	0	1.036,78	2002,841
990	1.352,13	0	660,452	2012,585
1020	960,907	0	1.035,31	1996,221
1050	1.007,60	0	998,025	2005,624
1080	1.145,81	0	844,849	1990,654
1110	1.272,29	0	745,046	2017,334
1140	1.155,89	0	787,087	1942,973
1170	1.126,88	0	879,14	2006,023
1200	1.089,48	2,153	905,9	1997,536
1230	821,197	454,56	732,251	2008,008
1260	651,454	653,169	727,498	2032,121
1290	660,407	676,407	698,618	2035,432

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1320	711,013	658,356	665,716	2035,085
1350	595,546	592,964	803,904	1992,414
1380	676,192	662,568	724,084	2062,844
1410	656,362	666,491	701,177	2024,03
1440	658,675	664,177	712,875	2035,727
1470	644,238	639,623	722,746	2006,607
1500	647,837	646,255	738,466	2032,558
1530	643,027	648,138	720,187	2011,352
1560	676,55	635,38	712,016	2023,946
1590	643,781	652,471	723,477	2019,729
1620	663,819	651,01	715,434	2030,263
1650	667,034	659,856	691,672	2018,562

Lampiran 8: Data Simulasi 2 Tanpa Reservasi Skenario 3

	HTTP	FTP	VIDEO STREAMING	TOTAL
0	0	0	0	0
30	0	0	112,598	112,598
60	0	0	133,801	133,801
90	0	0	135,629	135,629
120	0	0	135,629	135,629
150	0	0	138,554	138,554
180	0	0	132,383	132,383
210	0	0	134,167	134,167
240	0	0	139,285	139,285
270	0	0	133,436	133,436
300	0	0	133,436	133,436
330	0	0	136,726	136,726
360	0	0	133,846	133,846
390	0	0	136,36	136,36
420	0	0	135,995	135,995
450	0	0	134,532	134,532
480	0	0	135,995	135,995
510	0	0	134,898	134,898
540	0	0	137,503	137,503
570	0	0	133,436	133,436
600	0	0	136,36	136,36
630	1.501,38	0	139,65	1641,031
660	1.873,23	0	136,36	2009,585
690	1.874,51	0	135,629	2010,141
720	1.876,15	0	132,704	2008,852
750	1.877,36	0	138,234	2015,59
780	1.885,37	0	136,726	2022,092
810	1.883,97	0	138,554	2022,52
840	1.870,37	0	134,167	2004,536
870	1.831,61	0	134,167	1965,774
900	1.871,56	0	135,629	2007,191
930	1.877,18	0	137,503	2014,685
960	1.895,24	0	136,726	2031,964
990	1.883,04	0	136,36	2019,402
1020	1.891,99	0	137,457	2029,444
1050	1.851,58	0	137,823	1989,407
1080	1.878,08	0	131,608	2009,683
1110	1.871,51	0	134,167	2005,676
1140	1.876,50	0	138,6	2015,103
1170	1.871,21	0	141,478	2012,691
1200	1.830,03	1,62	134,167	1965,818
1230	1.156,20	749,16	144,037	2049,4
1260	881,601	902,518	125,759	1909,878

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1290	936,383	939,99	136,36	2012,733
1320	920,572	930,552	133,846	1984,97
1350	930,006	932,959	136,726	1999,691
1380	931,497	932,387	135,629	1999,513
1410	921,631	938,415	135,995	1996,041
1440	953,139	931,115	137,823	2022,077
1470	928,243	947,396	136,726	2012,365
1500	921,768	928,937	134,532	1985,237
1530	926,79	932,206	137,137	1996,133
1560	941,228	944,708	135,995	2021,931
1590	934,177	921,162	138,188	1993,527
1620	941,103	924,942	136,726	2002,771
1650	949,176	944,905	134,898	2028,979

Lampiran 9: Data Simulasi 1 Dengan Reservasi Skenario 1

	HTTP	FTP	VIDEO STREAMING	TOTAL
0	0	0	0	0
30	63,927	0	0	63,927
60	2.036,49	0	0	2036,485
90	2.030,60	0	0	2030,6
120	2.039,25	0	0	2039,246
150	2.045,38	0	0	2045,376
180	2.043,09	0	0	2043,093
210	2.034,45	0	0	2034,449
240	2.049,44	0	0	2049,441
270	2.034,49	0	0	2034,486
300	2.013,30	0	0	2013,301
330	2.042,47	0	0	2042,47
360	2.006,43	0	0	2006,431
390	2.046,33	0	0	2046,327
420	2.047,99	0	0	2047,985
450	2.034,02	0	0	2034,016
480	2.023,90	0	0	2023,899
510	2.038,39	0	0	2038,386
540	2.039,71	0	0	2039,707
570	2.050,13	0	0	2050,129
600	1.969,58	0	0	1969,577
630	1.880,30	133,114	0	2013,409
660	1.041,23	1.005,01	0	2046,232
690	1.018,57	1.019,90	0	2038,466
720	1.025,46	1.025,98	0	2051,443
750	1.023,99	1.023,11	0	2047,097
780	1.038,88	1.016,01	0	2054,89
810	1.027,65	1.013,59	0	2041,237
840	1.023,08	1.020,06	0	2043,135
870	1.020,76	1.026,17	0	2046,938
900	1.022,03	1.031,63	0	2053,657
930	1.027,31	1.018,14	0	2045,451
960	1.024,06	1.007,15	0	2031,213
990	1.030,87	1.004,02	0	2034,889
1020	991,538	1.005,62	0	1997,153
1050	1.002,09	996,555	0	1998,642
1080	1.016,36	993,217	0	2009,579
1110	1.012,86	1.027,28	0	2040,137
1140	989,545	976,038	0	1965,583
1170	949,62	943,063	0	1892,683
1200	1.004,01	1.000,74	0	2004,751
1230	1.007,45	1.005,03	25,59	2038,061
1260	625,061	632,473	712,875	1970,409
1290	639,671	647,904	713,845	2001,42

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1320	634,204	647,829	701,908	1983,941
1350	642,495	650,55	694,596	1987,641
1380	654,773	654,823	688,382	1997,978
1410	656,237	664,659	704,833	2025,729
1440	632,735	644,942	723,111	2000,788
1470	634,825	638,279	716,165	1989,269
1500	646,854	648,027	713,479	2008,36
1530	647,434	654,846	717,262	2019,542
1560	650,197	650,659	700,811	2001,667
1590	644,378	648,509	713,972	2006,859
1620	656,979	648,859	690,941	1996,779
1650	654,295	655,452	704,467	2014,214

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1290	948,49	937,787	134,212	1071,999
1320	955,651	942,19	135,629	1077,819
1350	955,496	941,28	135,629	1076,909
1380	954,679	946,043	136,726	1082,769
1410	950,846	940,199	137,823	1078,022
1440	953,785	936,254	133,436	1069,69
1470	946,559	948,951	133,436	1082,387
1500	946,421	945,218	136,406	1081,624
1530	949,851	953,954	137,457	1091,411
1560	959,911	940,789	133,436	1074,225
1590	949,985	959,972	135,995	1095,967
1620	950,39	934,209	134,898	1069,107
1650	939,157	947,416	135,629	1083,045

Lampiran 11: Data Simulasi 1 Dengan Reservasi Skenario 2

	HTTP	FTP	VIDEO STREAMING	TOTAL
0	0	0	0	0
30	0	0,291	0	0,291
60	0	363,338	0	363,338
90	0	2.034,86	0	2034,86
120	0	2.042,81	0	2042,81
150	0	2.033,12	0	2033,118
180	0	2.031,98	0	2031,976
210	0	2.037,27	0	2037,27
240	0	2.032,44	0	2032,435
270	0	2.045,49	0	2045,486
300	0	2.033,81	0	2033,807
330	0	2.042,07	0	2042,066
360	0	2.034,01	0	2034,012
390	0	2.048,19	0	2048,192
420	0	2.039,90	0	2039,899
450	0	2.037,43	0	2037,433
480	0	2.027,04	0	2027,04
510	0	2.038,71	0	2038,71
540	0	2.043,79	0	2043,792
570	0	2.039,58	0	2039,577
600	0	2.052,75	0	2052,748
630	0	2.041,66	0	2041,664
660	0	1.845,71	161,219	2006,928
690	0	1.287,14	722,746	2009,885
720	0	1.181,41	759,304	1940,712
750	0	1.246,70	754,917	2001,617
780	0	1.180,06	800,149	1980,211
810	0	898,857	1.060,91	1959,762
840	0	984,9	991,445	1976,345
870	0	1.048,41	926,007	1974,421
900	0	1.042,05	951,963	1994,013
930	0	1.000,91	987,789	1988,694
960	0	1.074,18	923,082	1997,261
990	0	1.243,40	724,45	1967,854
1020	0	975,311	1.022,89	1998,196
1050	0	1.104,73	886,159	1990,886
1080	0	1.146,03	836,44	1982,471
1110	0	1.281,06	701,542	1982,6
1140	0	1.057,27	911,018	1968,292
1170	0	1.184,41	795,761	1980,167
1200	0	1.055,17	895,298	1950,47
1230	0	964,879	1.020,69	1985,57
1260	157,7	819,9	935,512	1913,112
1290	631,634	617,599	698,983	1948,216

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1320	655,11	656,76	694,962	2006,832
1350	646,473	645,585	710,682	2002,74
1380	659,296	647,433	689,343	1996,072
1410	654,462	642,278	704,101	2000,841
1440	645,691	654,838	693,134	1993,663
1470	625,801	626,893	726,767	1979,461
1500	625,785	638,8	712,51	1977,095
1530	635,608	627,835	664,985	1928,428
1560	619,317	618,921	699,217	1937,455
1590	640,937	645,915	697,155	1984,007
1620	636,148	645,136	698,983	1980,267
1650	637,209	645,274	687,285	1969,768

Lampiran 12: Data Simulasi 2 Dengan Reservasi Skenario 2

	HTTP	FTP	VIDEO STREAMING	TOTAL
0	0	0,295	0	0,295
30	0	708,162	0	708,162
60	0	2.049,40	0	2049,404
90	0	1.959,71	0	1959,71
120	0	1.845,37	0	1845,372
150	0	1.919,68	0	1919,676
180	0	1.959,85	0	1959,853
210	0	1.920,47	0	1920,466
240	0	1.879,28	0	1879,284
270	0	1.939,38	0	1939,384
300	0	1.939,12	0	1939,115
330	0	1.979,09	0	1979,088
360	0	2.221,74	0	2221,741
390	0	1.933,90	0	1933,896
420	0	2.112,60	0	2112,604
450	0	1.963,75	0	1963,752
480	0	2.125,91	0	2125,905
510	0	2.180,61	0	2180,614
540	0	2.103,92	0	2103,918
570	0	2.101,08	0	2101,075
600	0	2.037,08	0	2037,082
630	0	1.690,39	114,791	1805,185
660	0	1.989,84	144,403	2134,244
690	0	1.977,26	149,887	2127,144
720	0	1.994,97	150,618	2145,592
750	0	2.216,67	170,05	2386,719
780	0	2.024,42	146,596	2171,013
810	0	2.003,62	146,962	2150,578
840	0	1.916,09	138,188	2054,278
870	0	1.942,48	143,672	2086,147
900	0	1.605,79	117,35	1723,136
930	0	1.526,57	107,48	1634,052
960	0	1.498,92	108,613	1607,53
990	0	1.446,88	103,824	1550,706
1020	0	1.474,91	110,039	1584,95
1050	0	1.525,34	109,673	1635,016
1080	0	1.516,74	106,748	1623,491
1110	0	1.823,47	131,973	1955,44
1140	0	1.886,63	142,623	2029,25
1170	0	1.830,12	127,221	1957,345
1200	0,402	1.860,54	137,823	1998,767
1230	522,776	1.347,21	134,898	2004,88
1260	936,192	963,433	135,629	2035,254

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1290	934,991	947,652	134,898	2017,541
1320	928,535	951,921	137,091	2017,547
1350	927,508	950,627	136,771	2014,906
1380	956,828	941,689	136,36	2034,877
1410	939,065	943,809	141,844	2024,718
1440	949,646	948,093	131,973	2029,712
1470	959,48	928,641	135,264	2023,385
1500	962,139	936,244	136,36	2034,743
1530	956,708	948,26	133,48	2038,448
1560	945,477	948,376	137,457	2031,31
1590	928,432	956,757	134,167	2019,356
1620	948,171	934,415	138,919	2021,505
1650	947,761	951,603	132,704	2032,068

Lampiran 13: Data Simulasi 1 Dengan Reservasi Skenario 3

	HTTP	FTP	VIDEO STREAMING	TOTAL
0	0	0	0	0
30	0	0	873,289	873,289
60	0	0	1.247,35	1247,349
90	0	0	1.029,10	1029,099
120	0	0	1.060,54	1060,539
150	0	0	1.029,47	1029,465
180	0	0	1.184,84	1184,835
210	0	0	1.101,85	1101,851
240	0	0	1.179,72	1179,717
270	0	0	757,841	757,841
300	0	0	1.236,38	1236,382
330	0	0	1.061,27	1061,27
360	0	0	997,66	997,66
390	0	0	978,284	978,284
420	0	0	668,498	668,498
450	0	0	692,037	692,037
480	0	0	774,658	774,658
510	0	0	769,54	769,54
540	0	0	1.002,41	1002,412
570	0	0	1.241,50	1241,5
600	40,419	0	1.061,62	1102,043
630	997,698	0	895,298	1892,996
660	998,244	0	940,995	1939,239
690	946,266	0	1.006,43	1952,7
720	1.204,99	0	741,756	1946,746
750	1.069,58	0	898,954	1968,533
780	964,542	0	1.000,22	1964,761
810	1.112,38	0	855,004	1967,383
840	1.184,45	0	795,496	1979,941
870	1.253,55	0	732,982	1986,529
900	1.048,23	0	907,362	1955,596
930	1.135,54	0	840,096	1975,637
960	1.028,71	0	942,092	1970,802
990	971,289	0	1.002,75	1974,036
1020	958,407	0	1.012,65	1971,055
1050	952,974	0	1.017,40	1970,375
1080	925,504	0	1.012,28	1937,787
1110	960,145	0	1.013,01	1973,159
1140	956,51	0	1.010,46	1966,965
1170	955,061	0	1.044,07	1999,132
1200	1.115,57	0	874,46	1990,029
1230	905,249	285,114	751,261	1941,624
1260	637,565	640,452	733,713	2011,73
1290	651,489	658,528	688,747	1998,764

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1320	647,727	656,12	694,962	1998,809
1350	633,261	634,996	713,241	1981,498
1380	647,928	646,196	682,029	1976,153
1410	638,475	646,862	658,039	1943,376
1440	640,246	616,6	700,08	1956,926
1470	630,427	630,637	706,66	1967,724
1500	640,841	635,57	724,574	2000,985
1530	609,802	622,128	751,992	1983,922
1560	636,537	644,959	712,382	1993,878
1590	626,901	614,698	712,144	1953,743
1620	639,012	642,568	707,757	1989,337
1650	621,742	619,585	709,951	1951,278

Lampiran 14: Data Simulasi 2 Dengan Reservasi Skenario 3

	HTTP	FTP	VIDEO STREAMING	TOTAL
0	0	0,397	0	0,397
30	0	0,248	10,602	10,85
60	0	0,132	148,424	148,556
90	0	0,394	148,108	148,502
120	0	0,327	149,521	149,848
150	0	0,132	149,155	149,287
180	0	0,248	150,983	151,231
210	0	0,381	150,983	151,364
240	0	0,132	148,79	148,922
270	0	0,313	145,5	145,813
300	0	0,327	149,205	149,532
330	0	0,132	146,962	147,094
360	0	0,248	136,726	136,974
390	0	0,327	140,382	140,709
420	0	0,132	136,726	136,858
450	0	0,248	135,995	136,243
480	0	0,211	138,6	138,811
510	0	0,169	139,285	139,454
540	0	0,327	134,898	135,225
570	0	0,132	138,554	138,686
600	79,82	0,248	142,209	222,277
630	2.125,37	0,327	153,177	2278,875
660	2.103,02	0,211	149,521	2252,756
690	2.075,27	0,248	149,937	2225,451
720	2.128,96	0,302	151,714	2280,975
750	2.083,09	0,211	150,983	2234,285
780	2.103,02	0,248	150,618	2253,882
810	2.073,32	0,248	149,521	2223,087
840	2.059,96	0,211	145,865	2206,033
870	2.067,83	0,248	146,645	2214,726
900	2.041,94	0,248	146,231	2188,416
930	2.064,10	0,211	150,252	2214,564
960	2.104,50	0,248	149,521	2254,267
990	2.070,49	0,433	149,155	2220,081
1020	2.097,74	0,211	148,424	2246,379
1050	2.129,45	0,327	157,564	2287,342
1080	2.088,82	0,248	147,377	2236,442
1110	2.072,40	0,132	154,639	2227,173
1140	2.031,49	1,93	148,059	2181,475
1170	1.077,67	955,274	149,155	2182,099
1200	1.032,71	1.035,58	146,231	2214,524
1230	1.041,67	1.021,52	150,252	2213,448
1260	1.061,33	1.065,69	152,862	2279,881
1290	1.029,71	1.038,90	149,155	2217,767

(Lanjutan)

	HTTP	FTP	VIDEO STREAMING	TOTAL
1320	1.040,83	1.051,89	152,08	2244,802
1350	1.051,77	1.056,92	151,714	2260,4
1380	1.031,40	1.046,28	148,424	2226,11
1410	1.029,38	1.041,76	146,962	2218,103
1440	1.046,45	1.049,21	151,399	2247,055
1470	1.022,51	1.027,66	149,521	2199,688
1500	1.032,31	1.040,67	148,424	2221,402
1530	1.031,72	1.047,58	149,887	2229,189
1560	1.036,77	1.049,51	157,564	2243,838
1590	1.060,22	1.057,01	148,79	2266,017
1620	1.034,63	1.045,25	145,5	2225,381
1650	1.039,36	1.043,26	149,937	2232,56

Lampiran 15: File Konfigurasi NSIS

```
This is an example configuration file for NSIS
# Please change it to fit your needs (i.e. IP address configuration
# of your host).
#
# For more information about the configuration, check the manual at
# manual/Configuration

# *****
# *****  General Configuration  *****
# *****

# Start Ping and Diagnostics NSLP daemon together with GIST
nslp.startPing = yes
nslp.startQoS = yes
nslp.startNatFw = no
nslp.startDiag = yes

# Accept explicitly routed messages (default is yes)
gist.acceptExplicitMessages = yes

# Accept GIST DATA messages that do not relate to any GIST state (default is
yes)
gist.acceptStatelessGistMessage = yes

# BSD only:
# List of interfaces where NSIS should listen on for GIST Queries
# Interfaces are separated by commas
#bsd.interfaces = ed0, ed1

# *****
# *****  GIST Transport Configuration  *****
# *****

## Policies which transport protocols are offered to peers.
# Offer SCTP as transport to peers?
gist.offerSCTP = no

# Offer TLS over TCP as transport to peers?
gist.offerTLS = no

## Policies
# Prefer SCTP over TCP as transfer protocol?
gist.useSCTP = no
```

(Lanjutan)

```

# *****
# ***** GIST Timer Configuration *****
# *****

## All Timeouts are measured in milliseconds

# How long do we wait for a Response to our initial Query?
# On retransmission, this value is doubled each time. (default: 10000 ms)
gist.timeout.waitForInitialResponse = 10000

# How long do we wait for a Confirm on the Receiver-Side? (default: 10000 ms)
gist.timeout.waitForConfirm = 10000

# How long do we wait between sending refreshing Queries? (default: 30000 ms)
gist.timeout.refreshInterval = 30000

# How long do we wait for a Response to a refreshing Query
# until state is removed? (default: 100000 ms)
gist.timeout.queryingNodeStateExpiration = 100000

# How long do we wait for a refreshing Query
# until state is removed? (default: 100000 ms)
gist.timeout.respondingNodeStateExpiration = 100000

# *****
# ***** IP address/routing configuration *****
# *****

# If readRoutingTable is set to yes, all IP address configuration
# used by NSIS is derived from the local IP routing tables and
# interface information.
# NOTE: If readRoutingTable is set to yes, all remaining IP address
# configuration in this file is NOT used by NSIS.
readRoutingTable = yes

# CAUTION: The address configuration is like a routing table.

# This example IPv4 configuration contains a default route
# as well as special configuration for two network segments
# (i.e 192.168.0.0/24 and 192.168.1.0/24)
#IPv4.entries = 3

# The first entry is meant as a default route. It is used when
# no subsequent entry matches.

```

(Lanjutan)

```

IPv4[0].addr = 10.0.0.1
#IPv4[0].net = 0.0.0.0
#IPv4[0].mask = 0
# Use this address as the external address for NatFW NSLP
#IPv4[0].natfw.useAsExternalAddress = yes
# Default network is public (i.e. the global internet)
#IPv4[0].natfw.isPrivateNet = no

# This entry defines the outgoing local interface card
#IPv4[1].addr = 192.168.0.1
# These entries define the network segment you want to reach
# with the previously defined outgoing interface card
#IPv4[1].net = 192.168.0.0
#IPv4[1].mask = 24
# The network 192.168.0.0/24 is private
IPv4[1].natfw.isPrivateNet = yes

# The second network (192.168.1.0/24) uses the SAME outgoing
# interface again.
#IPv4[2].addr = 192.168.0.1
#IPv4[2].net = 192.168.1.0
#IPv4[2].mask = 24
# The network 192.168.1.0/24 is private
#IPv4[2].natfw.isPrivateNet = yes

# IPv6 configuration with 1 address: (just remove this part completely
# if there is no IPv6 support on your machine)
#IPv6.entries = 1
#IPv6[0].addr = fec0:1::4321:ff:ee12:3355
#IPv6[0].net = fec0:1::4321:ff:ee12:3355
#IPv6[0].mask = 0

# *****
# ***** NatFW NSLP Configuration *****
# *****

# This host runs a NAT, but no firewall
natfw.isNAT = yes
natfw.isFW = no

```

(Lanjutan)

```
# Hosts inside the private network can reserve external addresses/ports.

# As the above configuration shows, 10.0.0.1 is the only external address this
# router has to offer:
natfw.resources.IPv4.entries = 1
natfw.resources.IPv4[0].addr = 10.0.0.1

# *****
# *****  QoS NSLP Configuration  *****
# *****

# Choose the RMF: ClsRMF, SimpleRMF and NullRMF (default) are available.
# See man page on what they do. (case does not matter)
qos.rmf = ClsRMF

## Configuration related to the SimpleRMF
# Use Traffic Control on these addresses if SimpleRMF is used:
qos.simplermf.interfaces = eth0

# Specify maximum overall bandwidth of the interface (default 100mbit)
qos.simplermf.interface[eth0].overall_bandwidth = 4mbit

# Specify minimum bandwidth, that unclassified traffic can use (default 100kbit)
# (Unclassified traffic is such traffic, that is not part of any known flow)
qos.simplermf.interface[eth0].min_unclassified_bandwidth = 100kbit

# Specify maximum bandwidth, that unclassified traffic can use
# (default is maximum overall bandwidth)
qos.simplermf.interface[eth0].max_unclassified_bandwidth = 2mbit
```