

**PENGARUH HANDOVER PADA SISTEM KOMUNIKASI HSDPA
(HIGH SPEED DOWNLINK PACKET ACCESS) KE EDGE
(ENHANCED DATA RATE FOR GSM EVOLUTION) UNTUK
LAYANAN MOBILE TV**

SKRIPSI
KONSENTRASI TEKNIK TELEKOMUNIKASI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun Oleh :

ANNISA TAUFIIKA FIRDAUSI
NIM: 0610630016 - 63

KEMENTERIAN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2010

LEMBAR PERSETUJUAN

PENGARUH HANDOVER PADA SISTEM KOMUNIKASI HSDPA (HIGH SPEED DOWNLINK PACKET ACCESS) KE EDGE (ENHANCED DATA RATE FOR GSM EVOLUTION) UNTUK LAYANAN MOBILE TV

SKRIPSI KONSENTRASI TEKNIK TELEKOMUNIKASI

Diajukan untuk memenuhi persyaratan
memperoleh gelar sarjana teknik



Disusun Oleh :

ANNISA TAUFIIKA FIRDAUSI
NIM: 0610630016 - 63

Telah diperiksa dan disetujui oleh

Dosen Pembimbing

M. Fauzan Edy P, ST., MT.
NIP 19710609 200002 1 001

Dwi Fadilla K., ST., MT.
NIP 19720630 200003 1 002

LEMBAR PENGESAHAN

**PENGARUH HANDOVER PADA SISTEM KOMUNIKASI HSDPA
(HIGH SPEED DOWNLINK PACKET ACCESS) KE EDGE
(ENHANCED DATA RATE FOR GSM EVOLUTION) UNTUK
LAYANAN MOBILE TV**

Disusun Oleh :

ANNISA TAUFIKA FIRDAUSI

NIM: 0610630016 - 63

Skripsi ini telah diuji dan dinyatakan lulus
pada tanggal **21 Desember 2010**

Majelis Penguji :

Gaguk Asmungi, ST., MT.
NIP 19670627 199802 1 001

Ir. Endah Budi P., MT.
NIP 19621116 198903 2 002

Ali Mustofa, ST., MT.
NIP 19710601 200003 1 001

Mengetahui :

Ketua Jurusan Teknik Elektro

Rudy Yuwono, ST., MSc.
NIP. 19710615 199802 1 003



PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih dan Penyayang. Segala puji bagi Allah SWT karena atas rahmat dan hidayahNya-lah penulis dapat menyelesaikan Tugas Akhir yang berjudul "**Pengaruh Handover pada Sistem Komunikasi HSDPA (High Speed Downlink Packet Access) ke EDGE (Enhanced Data Rate for GSM Evolution)** untuk Layanan Mobile TV". Tugas Akhir ini disusun untuk memenuhi persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Program Studi Teknik Telekomunikasi Fakultas Teknik Universitas Brawijaya Malang.

Melalui kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan baik lahir maupun batin selama penggerjaan Tugas Akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih kepada :

1. Ibundaku tercinta, almarhum Papa, Bapak, dan Adikku, serta seluruh keluarga yang senantiasa memberikan doa dan dukungan demi terselesaiannya tugas akhir ini.
2. Bapak M. Fauzan Edy P, ST., MT. dan Bapak Dwi Fadilla Kurniawan, ST., MT. Selaku dosen Pembimbing tugas akhir penulis yang telah membimbing penulis dalam menyelesaikan tugas akhir ini.
3. Bapak Rudy Yuwono,ST.,M.Sc. dan Bapak Muhammad Aziz Muslim, ST., MT.,PhD. Selaku Ketua dan Sekretaris Jurusan Teknik Elektro, serta segenap Bapak/Ibu Dosen, Staff Administrasi dan perpustakaan Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang.
4. EkaDhe, Ratih, Asih, sahabat seperjuangan yang paling baik, yang selalu menemani dalam suka dan duka dengan tawa dan canda.
5. Teman-teman angkatan 2006 (Ge-Force) serta teman-teman Paket C 2006, terimakasih untuk semuanya.
6. Mbak Fatimah, Tupil, Dini, Mami, Riris, Henry, Ranto, Rizcky, Danny, Nanang, Heru, Nuke, atas bantuannya dalam proses penggerjaan tugas akhir ini.
7. Seluruh rekan asisten Laboratorium Dasar Elektrik dan Pengukuran, terimakasih untuk bantuan, kebersamaan, serta kekompakannya.



8. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung selama proses penyelesaian tugas akhir ini.

Semoga Allah SWT memberikan Rahmat dan hidayah-Nya kepada semua yang telah membantu penulis dalam penyelesaian tugas akhir ini. Penulis menyadari bahwa masih ada kekurangan pada tugas akhir ini. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga tugas akhir ini memberikan manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 29 November 2010

Penulis



UNIVERSITAS BRAWIJAYA



ABSTRAK

Annisa Taufika Firdausi, 2010 : Pengaruh Handover pada Sistem Komunikasi HSDPA (High Speed Downlink Packet Access) ke EDGE (*Enhanced Data Rate for GSM Evolution*) untuk Layanan Mobile TV. Skripsi Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. Dosen Pembimbing : M. Fauzan Edy P, ST., MT. dan Dwi Fadilla Kurniawan, ST., MT.

Semakin tingginya tingkat mobilitas masyarakat saat ini, mendorong jasa layanan operator seluler untuk meningkatkan pelayanan pada *real time* multimedia, salah satunya adalah *mobile TV*. Teknologi komunikasi bergerak digital yang berkembang saat ini adalah HSDPA (*High Speed Downlink Packet Access*). Terbatasnya jangkauan HSDPA yang ada, membuat pengguna tidak bisa bergerak bebas saat menjauhi jangkauan area HSDPA, karena layanan *mobile TV* dapat terganggu. Oleh karena itu perlu adanya proses pemindahalihan penanganan akses layanan ini ketika pengguna bergerak menjauhi jangkauan area HSDPA menuju jangkauan area EDGE.

Penyusunan tugas akhir ini bertujuan untuk mengetahui pengaruh yang terjadi pada *throughput* jaringan EDGE untuk layanan *mobile TV* selama proses *handover* dari jaringan HSDPA ke EDGE berdasarkan prosedur *handover* pada dua jaringan tersebut.

Hasil analisis perhitungan untuk *handover* dari jaringan HSDPA ke EDGE dengan aplikasi *mobile TV* menggunakan prosedur *handover*, menunjukkan *delay handover* terbesar yang masih dapat ditolelir adalah 0,2629 s, yaitu pada penggunaan MCS-6 (*Modulation and Coding Scheme-6*) dengan faktor utilisasi 0,9. Selain itu, terjadi penurunan nilai *throughput* pada jaringan EDGE dari nilai yang semestinya, karena adanya pengaruh *delay handover*. Dari hasil analisis, penggunaan layanan *mobile TV* saat *handover* dari HSDPA ke EDGE dapat dilakukan pada MCS-6 sampai MCS-9, karena *delay* yang terjadi masih dapat ditolelir.

Kata kunci : Handover, HSDPA (*High Speed Downlink Packet Access*), EDGE (*Enhanced Data Rate for GSM Evolution*), mobile TV, throughput



DAFTAR ISI

PENGANTAR	i
ABSTRAK	iii
DAFTAR ISI	iv
DAFTAR TABEL	vii
DAFTAR GAMBAR.....	ix
DAFTAR SINGKATAN.....	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Ruang Lingkup.....	2
1.4 Tujuan	2
1.5 Sistematika Penulisan.....	2
BAB II TINJAUAN PUSTAKA	4
2.1 EDGE (<i>Enhanced Data rate for GSM Evolution</i>)	4
2.1.1 Arsitektur Jaringan EDGE	5
2.1.1.1 BSS (Base Station Subsystem).....	6
2.1.1.2 NSS (Network Switching Subsystem)	7
2.1.2 Air Interface	8
2.1.2.1 Radio Source Management	8
2.1.2.2 Channel Coding	8
2.1.2.3 Protokol Stack pada Jaringan EDGE.....	9
2.1.2.4 Format Protokol pada Jaringan EDGE.....	10
2.2 HSDPA (<i>High Speed Downlink Pcket Access</i>)	12
2.2.1 Karakteristik HSDPA	12
2.2.1.1 Shared channel transmission	12
2.2.1.2 AMC (<i>Adaptive Modulation Coding</i>)	15
2.2.1.3 TTI (<i>Short Transmission Time Interval</i>)	16
2.2.1.4 Fast Link Adaptation.....	16
2.2.1.5 Fast scheduling	16
2.2.1.6 H-ARQ (<i>Fast Hybrid Automatic Repeat Request</i>)	16



2.2.2	Arsitektur Jaringan HSDPA.....	17
2.2.2.1	UE (<i>User Equipment</i>)/Terminal	17
2.2.2.2	Node B	17
2.2.2.3	Iub Interface	18
2.2.2.4	RNC (<i>Radio Network Controller</i>).....	18
2.2.2.5	<i>Core Network</i>	18
2.2.3	Protokol Stack HSDPA.....	19
2.2.3.1	<i>Physical Layer</i>	19
2.2.3.2	RLC (<i>Radio Link Control</i>)	20
2.3	MobileTV	20
2.3.1	RTP (<i>Real Time Protocol</i>)	21
2.3.2	UDP (<i>User Datagram Protocol</i>).....	22
2.3.3	IP (<i>Internet Protocol</i>).....	22
2.3.4	Codec (<i>Coder/Decoder</i>) Mobile TV	23
2.3.5	H.264/AVC	24
2.3.6	AMR-WB (<i>Adaptive Multi Rate – WideBand</i>)	24
2.4	Handover	25
2.4.1	Jenis <i>Handover</i> pada Sistem Komunikasi Bergerak	25
2.4.1.1	Berdasarkan Teknologi yang Digunakan:	25
2.4.1.2	Berdasarkan Domain Administratif:	26
2.4.1.3	Berdasarkan Konektivitas:	26
2.4.1.4	Berdasarkan Frekuensi yang digunakan:	26
2.4.2	<i>Handover</i> Jaringan HSDPA ke EDGE	26
2.5	Parameter Performansi <i>Handover</i> dari Jaringan HSDPA ke EDGE.....	29
2.5.1	<i>Delay</i>	29
2.5.1.1	<i>Delay end-to-end</i>	29
2.5.1.2	<i>Delay</i> Proses	30
2.5.1.3	<i>Delay</i> Transmisi	45
2.5.1.4	<i>Delay</i> Propagasi	49
2.5.1.5	<i>Delay</i> Antrian	50
2.5.2	<i>Delay Handover</i>	52
2.5.3	<i>Throughput</i>	52
	BAB III METODOLOGI.....	57

3.1	Studi Literatur	57
3.2	Metode Pengambilan Data	57
3.2.1	Pada Jaringan EDGE	57
3.2.2	Pada Jaringan HSDPA	59
3.2.3	<i>Handover</i> antara jaringan HSDPA dan EDGE	61
3.2.4	Pada mobile TV	61
3.3	Metode Analisis dan Pembahasan.....	62
3.3.1	Menentukan Arsitektur Jaringan EDGE dan HSDPA	62
3.3.2	Menentukan Standar Aplikasi <i>Mobile TV</i>	62
3.3.3	Menentukan Pemodelan Arsitektur Jaringan HSDPA dan EDGE	62
3.3.4	Menganalisis Performansi Jaringan HSDPA dan EDGE	63
3.3.5	Menentukan Prosedur <i>Handover</i> Antara Jaringan HSDPA dan EDGE	69
3.3.6	Menganalisis <i>Handover</i> dari Jaringan HSDPA ke EDGE	69
3.4	Pengambilan Kesimpulan dan Saran.....	70
BAB IV HASIL DAN PEMBAHASAN		71
4.1	Umum.....	71
4.2	Paket Data Aplikasi <i>Mobile TV</i>	71
4.3	Performansi Jaringan EDGE/HSDPA.....	72
4.3.1	Analisis <i>Delay</i>	72
4.3.1.1	<i>Delay</i> Proses	72
4.3.1.2	<i>Delay</i> Transmisi	85
4.3.1.3	<i>Delay</i> Propagasi	89
4.3.1.4	<i>Delay</i> Antrian	94
4.3.1.5	<i>Delay</i> Total	100
4.3.2	Perhitungan <i>Throughput</i>	102
4.4	Analisis <i>Delay Handover</i> , dan Pengaruhnya	110
BAB V PENUTUP		149
5.1	Kesimpulan	149
5.2	Saran.....	150
DAFTAR PUSTAKA.....		151
LAMPIRAN		

DAFTAR TABEL

Tabel 2.1 Modulasi dan <i>Coding Scheme</i> pada EDGE.....	9
Tabel 2.2 Kategori UE pada HSDPA.....	18
Tabel 2.3 Audio dan Video <i>Codec</i> pada Aplikasi <i>Mobile TV</i>	23
Tabel 2.4 Ukuran RLC pada EDGE.....	35
Tabel 2.5 Jarak tiap <i>node</i> pada HSDPA.....	50
Tabel 2.6. Link Budget Air Interface HSDPA	53
Tabel 3.1 Modulasi dan <i>Coding Scheme</i> pada EDGE.....	58
Tabel 3.2 Ukuran RLC Data Unit pada EDGE	58
Tabel 3.3 Kecepatan Tiap <i>Node</i> pada Jaringan EDGE	59
Tabel 3.4 Jarak Antar <i>Node</i> pada Jaringan EDGE	59
Tabel 3.5 Kecepatan tiap <i>Interface</i> pada Jaringan EDGE	59
Tabel 3.6 Kategori UE pada HSDPA.....	60
Tabel 3.7 Kecepatan Tiap <i>Node</i> pada Jaringan HSDPA.....	60
Tabel 3.8 Jarak Antar <i>Node</i> pada Jaringan HSDPA	60
Tabel 3.9 Kecepatan tiap <i>Interface</i> pada Jaringan HSDPA	60
Tabel 3.10 Link Budget Air Interface HSDPA	61
Tabel 3.11 Audio dan Video <i>Codec</i> pada Aplikasi <i>Mobile TV</i>	62
Tabel 4.1 Hasil Perhitungan <i>Delay</i> Proses pada PCU.....	77
Tabel 4.2 Hasil Perhitungan <i>Delay</i> Dekapsulasi pada UE di EDGE	78
Tabel 4.3 Hasil Perhitungan <i>Delay</i> Proses pada Jaringan EDGE	84
Tabel 4.4 Hasil Perhitungan <i>Delay</i> Transmisi pada Abis <i>Interface</i>	86
Tabel 4.5 Hasil Perhitungan <i>Delay</i> Transmisi pada Um <i>Interface</i>	87
Tabel 4.6 Hasil Perhitungan <i>Delay</i> Transmisi Total pada EDGE.....	89
Tabel 4.7 Hasil Perhitungan <i>Delay</i> Propagasi BSC ke BTS	91
Tabel 4.8 Hasil Perhitungan <i>Delay</i> Propagasi dari BTS ke UE	92
Tabel 4.9 Hasil Perhitungan <i>Delay</i> Propagasi Total pada Jaringan EGDE	93
Tabel 4.10 Hasil Perhitungan <i>Delay</i> Antrian pada GGSN.....	95
Tabel 4.11 Hasil Perhitungan <i>Delay</i> Antrian pada 2.75G SGSN	96
Tabel 4.12 Hasil Perhitungan <i>Delay</i> Antrian pada 3.5G SGSN	97
Tabel 4.13 Hasil Perhitungan <i>Delay</i> Antrian Total pada Jaringan EDGE	98
Tabel 4.14 Hasil Perhitungan <i>Delay</i> Antrian Total pada Jaringan HSDPA	98
Tabel 4.15 Hasil Perhitungan <i>Delay</i> Total pada Jaringan EDGE	100
Tabel 4.16 Hasil Perhitungan <i>Delay</i> Total pada Jaringan HSDPA.....	101

Tabel 4.17 Hasil Perhitungan <i>Throughput</i> pada Jaringan EDGE	105
Tabel 4.18. Link Budget Air Interface HSDPA	106
Tabel 4.19 Hasil Perhitungan <i>Throughput</i> pada Jaringan HSDPA	108
Tabel 4.20 Hasil Perhitungan <i>Delay Enkapsulasi</i> pada UE di EDGE	111
Tabel 4.21 Hasil Perhitungan <i>Delay Handover</i> untuk MCS-6 dengan Faktor Utilisasi 0,1	138
Tabel 4.22 Hasil Perhitungan <i>Delay Handover</i>	139
Tabel 4.23 Hasil Perhitungan <i>Delay Proses</i> dengan Pengaruh <i>Handover</i>	141
Tabel 4.24 Hasil Perhitungan <i>Delay Transmisi</i> dengan Pengaruh <i>Handover</i>	141
Tabel 4.25 Hasil Perhitungan <i>Delay Propagasi Total</i> dengan Pengaruh <i>Handover</i>	141
Tabel 4.26 Hasil Perhitungan <i>Delay Antrian Total</i> dengan Pengaruh <i>Handover</i>	142
Tabel 4.27 Perubahan nilai <i>Throughput</i> karena Pengaruh <i>Handover</i>	144



DAFTAR GAMBAR

Gambar 2.1. Arsitektur Jaringan EDGE	6
Gambar 2.2 Arsitektur Protokol EDGE.....	10
Gambar 2.3 <i>Interface</i> antar <i>node</i> pada EDGE	10
Gambar 2.4 Pewaktuan antara HS-SCCH dan HS-DSCH	13
Gambar 2.5. Struktur <i>Frame</i> Data HS-DSCH	14
Gambar 2.6 Struktur <i>Frame</i> HS-DPCCH.....	15
Gambar 2.7 Proses HARQ pada HSDPA.....	17
Gambar 2.8 Arsitektur Jaringan HSDPA	17
Gambar 2.9 Protokol Stack HSDPA.....	19
Gambar 2.10 Proses Pembentukan MAC PDU	20
Gambar 2.11 Skema Sistem MobileTV Berbasis Digital Video Broadcasting-Handheld	20
Gambar 2.12 Format <i>Header</i> Ipv4	22
Gambar 2.13 Format <i>Header</i> Ipv6	23
Gambar 2.14 Paket Data Mobile TV	23
Gambar 2.15 <i>Intersystem Handover</i> HSDPA ke EDGE	27
Gambar 2.16 Prosedur <i>signaling handover</i> HSDPA ke EDGE	27
Gambar 2.17 Arsitektur <i>Node-to-node</i> HSDPA ke EDGE.....	29
Gambar 3.1 Arsitektur <i>Node-to-node</i> HSDPA ke EDGE.....	63
Gambar 3.2 Diagram Alir Perhitungan Parameter <i>Handover</i>	63
Gambar 3.3 Diagram Alir Perhitungan <i>Delay</i> Proses.....	64
Gambar 3.4 Diagram Alir Perhitungan <i>Delay</i> Transmisi	65
Gambar 3.5 Diagram Alir Perhitungan <i>Delay</i> Propagasi	66
Gambar 3.6 Diagram Alir Perhitungan <i>Delay</i> Antrian	67
Gambar 3.7 Diagram Alir Perhitungan <i>Throughput</i>	68
Gambar 3.8 Diagram Alir Perhitungan Parameter <i>Handover</i>	69
Gambar 3.9 Diagram Alir Perhitungan Paket Data Mobile TV	70
Gambar 4.1 Arsitektur <i>Node-to-node</i> HSDPA ke EDGE.....	71
Gambar 4.2 Arsitektur Protokol EDGE.....	72
Gambar 4.3 Arsitektur Protokol HSDPA	73
Gambar 4.4 Arsitektur <i>Node-to-node</i> <i>Delay</i> Transmisi pada Jaringan EDGE	85
Gambar 4.5 Arsitektur <i>Node-to-node</i> <i>Delay</i> Transmisi pada Jaringan HSDPA.....	85
Gambar 4.6 Arsitektur <i>Node-to-node</i> <i>Delay</i> Propagasi pada Jaringan EDGE	89
Gambar 4.7 Arsitektur <i>Node-to-node</i> <i>Delay</i> Propagasi pada Jaringan HSDPA	90
Gambar 4.8 Arsitektur <i>Node-to-node</i> <i>Delay</i> Antrian pada Jaringan EDGE.....	94
Gambar 4.9 Arsitektur <i>Node-to-node</i> <i>Delay</i> Antrian pada Jaringan HSDPA	94
Gambar 4.10 Grafik Hubungan <i>Delay</i> Antrian Total Jaringan EDGE terhadap Faktor Utilisasi	99

Gambar 4.11 Grafik Hubungan <i>Delay</i> Antrian Total Jaringan HSDPA terhadap Faktor Utilisasi	99
Gambar 4.12 Grafik Hubungan <i>Delay</i> Total Jaringan EDGE terhadap Faktor Utilisasi.....	101
Gambar 4.13 Grafik Hubungan <i>Delay</i> Total Jaringan HSDPA terhadap Faktor Utilisasi	102
Gambar 4.14 Grafik Hubungan <i>Throughput</i> Jaringan EDGE terhadap Faktor Utilisasi.....	105
Gambar 4.15 Grafik Hubungan <i>Throughput</i> Jaringan HSDPA terhadap Faktor Utilisasi	109
Gambar 4.16 <i>Delay Handover</i> pada Prosedur Pensinyalan <i>Handover</i> dari HSDPA ke EDGE	140
Gambar 4.17 Grafik Hubungan <i>Throughput Handover</i> Jaringan EDGE terhadap Faktor Utilisasi.....	144
Gambar 4.18 Grafik Perbandingan <i>Throughput</i> Jaringan EDGE MCS-1 dengan Pengaruh <i>Handover</i> dan Tanpa Pengaruh <i>Handover</i> terhadap Faktor Utilisasi.....	145
Gambar 4.19 Grafik Perbandingan <i>Throughput</i> Jaringan EDGE MCS-2 dengan Pengaruh <i>Handover</i> dan Tanpa Pengaruh <i>Handover</i> terhadap Faktor Utilisasi.....	145
Gambar 4.20 Grafik Perbandingan <i>Throughput</i> Jaringan EDGE MCS-3 dengan Pengaruh <i>Handover</i> dan Tanpa Pengaruh <i>Handover</i> terhadap Faktor Utilisasi.....	146
Gambar 4.21 Grafik Perbandingan <i>Throughput</i> Jaringan EDGE MCS-4 dengan Pengaruh <i>Handover</i> dan Tanpa Pengaruh <i>Handover</i> terhadap Faktor Utilisasi.....	146
Gambar 4.22 Grafik Perbandingan <i>Throughput</i> Jaringan EDGE MCS-5 dengan Pengaruh <i>Handover</i> dan Tanpa Pengaruh <i>Handover</i> terhadap Faktor Utilisasi.....	146
Gambar 4.23 Grafik Perbandingan <i>Throughput</i> Jaringan EDGE MCS-6 dengan Pengaruh <i>Handover</i> dan Tanpa Pengaruh <i>Handover</i> terhadap Faktor Utilisasi.....	147
Gambar 4.24 Grafik Perbandingan <i>Throughput</i> Jaringan EDGE MCS-7 dengan Pengaruh <i>Handover</i> dan Tanpa Pengaruh <i>Handover</i> terhadap Faktor Utilisasi.....	147
Gambar 4.25 Grafik Perbandingan <i>Throughput</i> Jaringan EDGE MCS-8 dengan Pengaruh <i>Handover</i> dan Tanpa Pengaruh <i>Handover</i> terhadap Faktor Utilisasi.....	147
Gambar 4.26 Grafik Perbandingan <i>Throughput</i> Jaringan EDGE MCS-9 dengan Pengaruh <i>Handover</i> dan Tanpa Pengaruh <i>Handover</i> terhadap Faktor Utilisasi.....	148

DAFTAR SINGKATAN

3GPP	: Third Generation Partnership Project
AMC	: Adaptive Modulation Coding
AMPS	: Advanced Mobile Phone System
AMR-WB	: Adaptive Multi Rate – WideBand
ARQ	: Automatic Repeat Request
AuC	: Authentication Center
BCS	: Block Check Sequence
BER	: Bit Error Rate
BSC	: Base Station Controller
BSS	: Base Station Subsystem
BSSGP	: BSS Gateway Protocol
BTI	: Block Type Indicator
BTS	: Base Transceiver Station
CS	: Circuit Switch
CPCCH	: Common Packet Channel
CPICH	: Common Pilot Channel
CQI	: Channel Quality Information
CRC	: Cyclic Redundancy Check
DSCH	: Downlink Shared Channel
DVB-H	: Digital Video Broadcasting Handheld
DVB-T	: Digital Video Broadcasting Terrestrial
EDGE	: Enhanced Data rate for GSM Evolution
EIR	: Equipment Identity Register
ETSI	: European Telecommunication Standards Institute
FEC	: Forward Error Correction
GGSN	: Gateway GPRS Support Node
GSM	: Global System for Mobile Communication
GRPS	: General Packet Radio Service
GTP	: GPRS Tunnelling Protocol
HARQ	: Hybrid Automatic Repeat Request
HLR	: Home Location Register
HS-DPCCH	: High Speed-Dedicated Physical Control Channel
HS-DSCH	: High Speed- Downlink Shared Channel
HS-PDSCH	: High-Speed Physical Downlink Shared Channel
HS-SCCH	: High Speed-Shared Control Channel
HSDPA	: High Speed Downlink Packet Access
IMEI	: International Mobile Equipment Identity
IP	: Internet Protocol
LLC	: Logical Link Control
MAC	: Medium Acces Control
MCS	: Modulation Coding Scheme

MM	: Mobility Management
MS	: Mobile Station
MSS	: Maximum Segmen Size
MPE-FEC	: Multiprotocol Encapsulation Forward Error Correction
MPEG-2 TS	: Moving Picture Expert Group-2 Transport Stream
MSC	: Mobile Switching Subsystem
MTU	: Maximum Transfer Unit
NBAP	: Node B Application Part
NS	: Network Service
NSS	: Network Switching Subsystem
OSI	: Open System Interconnection
PC	: Power Control
PCH	: Pagging Channel
PCU	: Packet Control Unit
PDCH	: Packet Data Channel
PDN	: Packet Data Network
PDP	: Packet Data Protocol
PS	: Packet Switch
QAM	: Quadrature Amplitude Modulation
QPSK	: Quadrature Phase Shift Keying
RLC	: Radio Link Control
RNC	: Radio Network Controller
RTP	: Real Time Protocol
SGSN	: Serving GPRS Support Node
SNDCP	: Sub Network Dependent Convergence Protocol
TBS	: Transport Block Size
TRFC	: Transport Format and Resource Combination
TTI	: Transmission Time Interval
UDP	: User Datagram Protocol
UE	: User Equipment
UMTS	: Universal Mobile Telecommunication System
USF	: Uplink State Flag
UTRAN	: UMTS Terrestrial Radio Acces Network
VLR	: Visitor Location Register
WCDMA	: Wideband Code Division Multiple Access

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini kebutuhan masyarakat akan informasi sangat tinggi, baik informasi tersebut berupa suara, gambar, maupun *video*. Dengan tingkat mobilitas masyarakat yang semakin tinggi, maka diperlukan suatu teknologi telekomunikasi yang sangat handal dan dapat menawarkan berbagai layanan multimedia.

Sistem komunikasi bergerak merupakan alternatif yang sangat tepat untuk mengakomodasi hal tersebut. Salah satu bentuk teknologi komunikasi bergerak digital adalah *High Speed Downlink Packet Access* (HSDPA). Teknologi ini merupakan generasi 3.5 (3.5G) sebagai peningkatan dari teknologi 3G, sehingga dapat melayani peningkatan kapasitas trafik serta lebih meningkatkan kemampuan multimedia. Salah satu kemampuan multimedia yang dimiliki oleh HSDPA adalah *Mobile TV*. *Mobile TV* memberikan layanan multimedia yang mampu menayangkan siaran televisi digital pada seluler pemakai (*user*).

Saat ini operator seluler yang menyediakan teknologi HSDPA baru pada kota-kota besar saja, sehingga bila operator seluler hanya mengimplementasikan *Mobile TV* pada jaringan HSDPA saja, *Mobile TV* hanya bisa dinikmati pengguna seluler 3.5G yang berada di wilayah jangkauan BTS yang menggunakan teknologi HSDPA saja. Akibat yang lain adalah pengguna seluler 3.5G tidak bisa bergerak secara bebas. Tentunya hal ini tidak sesuai dengan harapan masyarakat dengan mobilitas hidup yang tinggi.

Teknologi *Global System for Mobile Communication* (GSM) yang telah berkembang dengan teknologi *Enhanced Data rate for GSM Evolution* (EDGE) mempunyai kemampuan multimedia dan masih banyak dipakai oleh operator seluler. Bila operator seluler dapat memanfaatkan EDGE untuk aplikasi *Mobile TV*, tentunya pengguna seluler dapat menikmati aplikasi *Mobile TV* di seluruh area *coverage* operator.

Penerapan aplikasi *Mobile TV* pada EGDE tentunya akan timbul beberapa masalah karena *bandwidth*, *delay*, *throughput*, dan *Bit Error Rate* (BER) EDGE berbeda dengan HSDPA. Selain itu, jaringan EDGE dan HSDPA juga berbeda, sehingga penerapan *Mobile TV* pada EGDE harus diperhitungkan secara cermat agar kualitas layanannya aplikasi *Mobile TV* masih cukup baik. Untuk melakukan *handover*

antara sistem HSDPA dan EDGE pun terdapat beberapa masalah, seperti apakah *Mobile TV* ini terputus atau kualitasnya berkurang, serta bagaimana cara melakukan *handover* dengan parameter pada aplikasi *Mobile TV*.

Dengan melihat aspek di atas, ada kemungkinan operator untuk dapat menerapkan fasilitas *Mobile TV* pada EDGE, di daerah yang tidak terjangkau HSDPA.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka pembahasan tugas akhir ini ditekankan pada rumusan masalah sebagai berikut :

1. Konsep dasar HSDPA dan EDGE beserta arsitektur jaringannya.
2. Bagaimana *handover* dari HSDPA ke EDGE.
3. Bagaimana paket yang dipakai untuk penerapan *Mobile TV* pada HSDPA dan EDGE.
4. Bagaimana *handover* pada *Mobile TV* pada syarat parameter kualitas layanan pada HSDPA dan EDGE yang meliputi kebutuhan *delay handover* dan *throughput*.
5. Bagaimana pengaruh *delay handover* terhadap perubahan *throughput* dalam jaringan HSDPA dan EDGE pada layanan *Mobile TV* selama proses *handover*.

1.3 Ruang Lingkup

Berdasarkan rumusan masalah di atas, maka pembahasan tugas akhir ini memiliki batasan – batasan sebagai berikut :

1. Pembahasan difokuskan pada pengaruh *handover* dengan prosedur yang digunakan pada *delay time* dan *throughput* pada layanan *Mobile TV*.
2. Analisis *handover* difokuskan pada prosedur *signalling handover* dari jaringan HSDPA ke EDGE.
3. Sistem komunikasi bergerak yang digunakan adalah sistem telepon seluler yang berbasis HSDPA dan EDGE.
4. Menggunakan *codec* H.264 untuk *codec* video dan AMR-WB untuk *codec* audio.
5. Media fisik dan model sinyal tidak dibahas

1.4 Tujuan

Tujuan dari penulisan tugas akhir ini adalah menganalisis pengaruh *handover* dari jaringan HSDPA ke EDGE terhadap *throughput* di jaringan EDGE dengan aplikasi *Mobile TV*.

1.5 Sistematika Penulisan

Sistematika penulisan dan gambaran untuk setiap bab pada skripsi ini adalah sebagai berikut :



BAB I PENDAHULUAN

Memuat tentang latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Meliputi uraian mengenai teori dasar teknologi EDGE dan HSDPA, membahas tentang konsep dasar *Mobile TV* yang mencakup layer DVB-H, format *frame* dan format *header* DVB-H, serta membahas tentang konsep dasar metode *handover*.

BAB III METODOLOGI

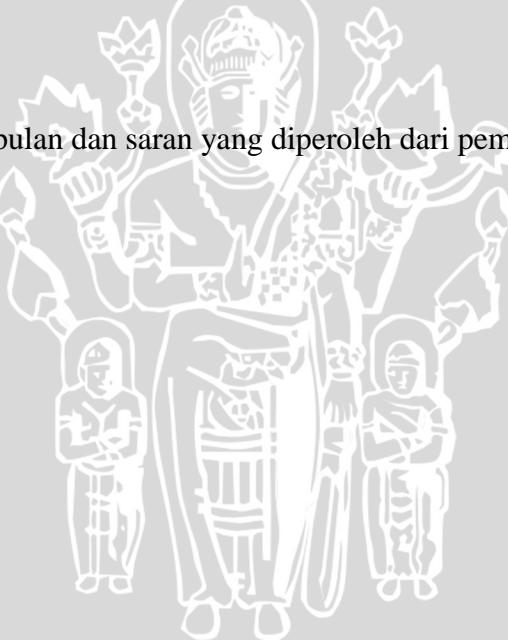
Memuat langkah-langkah yang diambil untuk menyelesaikan permasalahan dalam tugas akhir ini.

BAB IV ANALISIS HANDOVER PADA JARINGAN HSDPA DAN EDGE

Memuat pembahasan mengenai analisis *handover* pada sistem komunikasi HSDPA dan EDGE terhadap layanan *mobile TV* yang meliputi *delay* dan *throughput*.

BAB V PENUTUP

Berisi tentang kesimpulan dan saran yang diperoleh dari pembahasan.



BAB II

TINJAUAN PUSTAKA

Perkembangan telekomunikasi di dunia internasional, termasuk Indonesia, berjalan sangat pesat. Perkembangan ini dimulai dari teknologi AMPS (*Advanced Mobile Phone System*) yang hanya mengandalkan layanan *voice* dengan menggunakan teknologi analog, dimana tidak boleh ada dua pengguna yang menggunakan kanal yang sama baik dalam satu sel maupun sel tetangganya. Teknologi AMPS yang memiliki banyak kelemahan digantikan oleh teknologi lain yaitu GSM (*Global System for Mobile Communication*) yang mampu menyalurkan komunikasi suara dan data berkecepatan rendah, yang kemudian berkembang menjadi GPRS (*General Packet Radio Service*) dengan layanan *Packet Switched*. Pada fase berikutnya, diperkenalkan EDGE (*Enhanced Data rates for GSM Evolution*) dengan layanan *Packet Switched* dan kemampuan telekomunikasi yang lebih fleksibel, *real time* dan bahkan memberikan layanan yang lebih mengeksplorasi kemampuan multimedia seperti *Mobile TV*. Pada pengembangan selanjutnya, EDGE berkembang menjadi teknologi WCDMA (*Wideband Code Division Multiple Access*) yang memberikan kecepatan akses data yang tinggi.

HSDPA (*High Speed Downlink Packet Access*) merupakan sebuah konsep yang termasuk dalam spesifikasi WCDMA 3GPP (*Third Generation Partnership Project*) *Release 5*. Penerapan HSDPA pada jaringan WCDMA merupakan sebuah evolusi teknologi yaitu dari teknologi 3G menjadi 3.5G, yang bertujuan untuk meningkatkan kemampuan performansi *downlink* jaringan WCDMA.

2.1 EDGE (*Enhanced Data rate for GSM Evolution*)

Sepanjang era 1980-an, sistem telepon seluler analog mengalami pertumbuhan pesat di Eropa. Setiap negara mengembangkan sistem telepon seluler analog ini sesuai dengan keinginan mereka. Sehingga jika dibandingkan sistem yang satu dengan yang lainnya tidak ditemukan kesamaan, baik dari sisi cara kerja sistem maupun perangkat yang digunakan. Di samping itu sistem tersebut mempunyai daya jelajah yang sangat terbatas di negara yang bersangkutan. Negara-negara Eropa segera menyadari hal tersebut sehingga pada tahun 1982, CEPT (*Conference of European Posts and Telegraphs*) membentuk *Group Special Mobile* (singkatan awal dari GSM) untuk



membangun sistem selular di Eropa dan mengganti berbagai macam sistem yang telah ada yang dinilai tidak sesuai antara satu dengan lainnya tersebut.

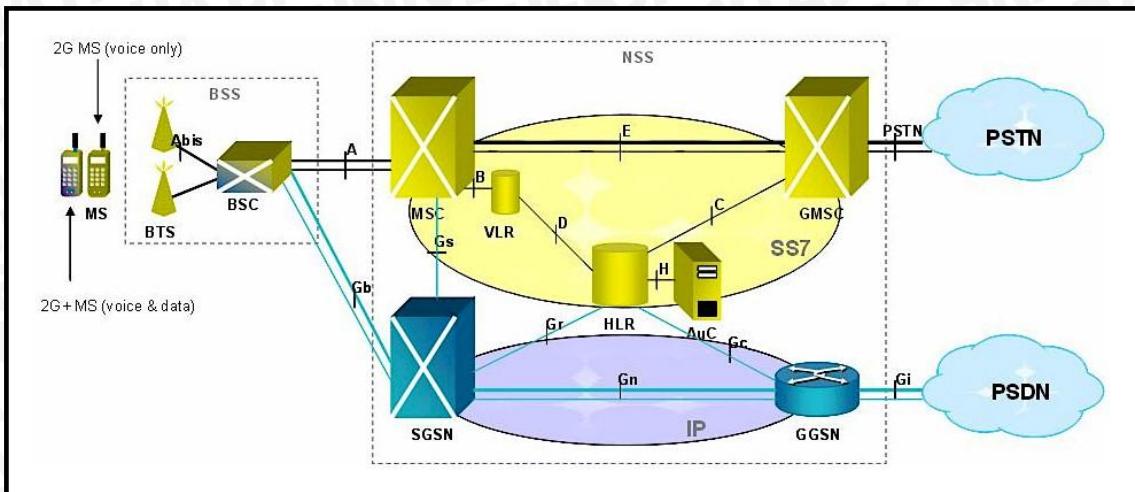
GSM kemudian diambil alih oleh ETSI (*European Telecommunication Standards Institute*) dari CEPT untuk kemudian menyelesaikan suatu standar GSM pada tahun 1990. Seiring dengan hal tersebut singkatan GSM kemudian diubah menjadi *Global System for Mobile Communications*. Dalam perkembangannya, GSM yang mampu menyalurkan komunikasi suara dan data berkecepatan rendah (9.6 - 14.4 kbps), kemudian berkembang menjadi GPRS (*General Packet Radio Service*) yang mampu menyalurkan suara dan juga data dengan kecepatan yang lebih baik, yaitu 115 kbps.

Seiring dengan makin tingginya tingkat kebutuhan bagi masyarakat modern yang menginginkan adanya aplikasi yang mendukung layanan dengan *bitrate* tinggi dan paket data yang tersambung pada jaringan internet, intranet, dan extranet yang memungkinkan adanya paket video dan layanan interaktif lainnya maka teknologi GPRS dikembangkan lebih lanjut dengan menambahkan infrastruktur-infrastruktur yang mendukung untuk aplikasi tersebut. *Enhanced Data rate for GSM Evolution* (EDGE) merupakan teknologi yang dikembangkan dengan dasar teknologi GSM dan GPRS. EDGE dirancang untuk meningkatkan kinerja GPRS sehingga mampu mempunyai laju bit yang lebih besar dari sebelumnya. EDGE mampu menyalurkan data dengan kecepatan hingga 3 kali kecepatan GPRS, yaitu 384 kbps. EDGE mendukung protokol komunikasi data yang mendukung adanya akses ke jaringan internet, *Mobile TV*, *Video Streaming*, dan layanan interaktif lainnya.

2.1.1 Arsitektur Jaringan EDGE

EDGE merupakan pengembangan pada antarmuka udara dari GPRS yang memungkinkan penggunaan laju bit lebih tinggi dan kapasitas sistem yang lebih besar. Sebuah sistem EDGE dikembangkan dengan tetap menggunakan perangkat yang terdapat pada jaringan GSM/GPRS. Untuk mengimplementasikan EDGE pada jaringan *existing* GPRS diperlukan penambahan perangkat keras maupun perangkat lunaknya. Pada jaringan radio, BTS (*Base Station Transceiver*) harus berkemampuan EDGE, oleh karena itu perangkat keras yang ditambahkan adalah perangkat EDGE Transceiver Unit (TRU) pada setiap BTS yang dirancang menjadi jaringannya. Kebutuhan perangkat lunak pada *Base Station* juga perlu diperbarui. *Upgrading software* dilakukan secara serentak ke beberapa BTS dari pusat kontrolnya, yaitu BSC (*Base Station Controller*). Sedangkan pada sisi *core network*, EDGE menggunakan perangkat dan protokol yang

sama dengan pada jaringan GPRS. Gambar 2.1 menunjukkan arsitektur jaringan EDGE yang dibagi menjadi beberapa bagian.



Gambar 2.1. Arsitektur Jaringan EDGE
Sumber : www.blogspot.com, 2008

2.1.1.1 BSS (Base Station Subsystem)

Sebuah BSS terdiri dari BSC (*Base Station Controller*) dan BTS (*Base Transceiver Station*). Fungsi utama dari BSS adalah untuk menghubungkan MS (*mobile station*) dengan NSS (*Network Switching Subsystem*) di *core network*. BSS diupgrade dengan protokol-protokol GPRS pada *Gb interface* (antara BSS dengan SGSN) serta penambahan protocol RLC/MAC pada *air interface*.

a) BTS (*Base Transceiver Station*)

BTS merupakan fasilitas pentransmisian dan penerimaan, termasuk antena dan semua pensinyalan yang berkaitan dengan *radio interface*. MS (*Mobile Station*) dihubungkan ke BTS melalui *radio interface*. MS dan BTS secara konstan mengirimkan sinyal informasi tentang lokasi dan kondisi level sinyal. Pada saat sinyal radio diterima, BTS memisahkan CS (*circuit switch*) dengan PS (*packet switch*), kemudian meneruskan kedua sinyal tersebut menuju BSC.

b) BSC (*Base Station Controller*)

BSC berfungsi untuk *men-set up*, mengawasi dan memutuskan (*disconnect*) hubungan CS dan PS. Pada sisi radio, hubungan-hubungan ini menuju ke BTS dan berasal dari BTS, sedangkan pada sistem di sisi jaringan inti (*core network*) hubungan tersebut menuju ke dan berasal dari satu MSC dan satu SGSN. Dalam BSC terdapat *switch* dengan kapasitas tinggi yang menyediakan fungsi seperti data konfigurasi sel

handover dan penempatan kanal. Sebuah BSC menginformasikan ke MSC tentang lokasi dari MS.

Untuk mengatur hubungan-hubungan tersebut, dalam BSC terdapat *switch* dengan kapasitas tinggi yang menyediakan fungsi seperti data konfigurasi sel *handover* dan penempatan kanal. BSC menganalisis ketersediaan sumber radio pada sel yang berbeda dan membuat keputusan mengenai *handover* dari sebuah MS ke sebuah sel dengan level sinyal yang lebih baik atau penggunaan level sinyal lebih rendah. BSC akan memerintahkan ke MSC untuk melakukan *switch* kontrol dari MS ke BSC lain jika dibutuhkan BSC *handover*.

BSC bertugas sebagai pengkonsentrasi rangkaian (*circuit concentrator*), membawa hubungan/koneksi dari beberapa BTS. Dalam hal *hardware*, BSC membutuhkan PCU (*Packet Control Unit*) untuk menangani paket-paket GPRS. PCU bertanggung jawab untuk lapisan RLC (*Radio Link Control*) dan MAC (*Medium Acces Control*) pada *interface* udara dan juga mengatur transfer paket data pengguna dari UE ke SGSN.

2.1.1.2 NSS (Network Switching Subsystem)

a) MSC (*Mobile Switching Subsystem*)

MSC merupakan elemen penting dalam NSS. MSC berfungsi untuk *switching* suatu panggilan telepon dari jaringan internal atau dari jaringan lain (eksternal), *call routing* untuk pelanggan yang melakukan roaming, menyimpan informasi *billing*, membantu proses *handover*, serta data base lain yang berisi informasi *subscriber ID*.

b) HLR (*Home Location Register*) dan VLR (*Visitor Location Register*)

HLR dan VLR bersama dengan MSC menyediakan *call-routing* dan fungsi *roaming* dari EDGE. HLR berisi informasi administrasi setiap pelanggan yang tersambung pada jaringan EDGE. VLR berisi informasi administrasi teripilih dari HLR, yang penting untuk control panggilan (*call control*) dan provisi dari layanan pelanggan, serta kontrol posisi setiap ponsel pada area geografis.

c) EIR (*Equipment Identity Register*)

EIR merupakan database yang berisi suatu daftar valid *mobile equipment* pada jaringan. Setiap *mobile station* diidentifikasi dengan IMEI (*International Mobile Equipment Identity*).

d) AuC (*Authentication Center*)

AuC merupakan *database* proteksi yang menyimpan salinan dari kunci rahasia (*secret key*) yang terdapat pada setiap kartu SIM pelanggan. Proteksi ini digunakan untuk autentifikasi dan enkripsi pada kanal radio.

e) GGSN (*Gateway GPRS Support Node*)

GGSN menyediakan *interface* sebagai penghubung dengan eksternal *network* seperti internet dan X.25. GGSN merupakan suatu *router* untuk alamat-alamat IP semua pelanggan yang dilayani oleh jaringan EDGE.

f) SGSN (*Serving GPRS Support Node*)

SGSN menangani masalah *mobility management*, *authentication*, serta *register*. SGSN mengkonversi protokol dari protokol IP yang digunakan di dalam *backbone network* menjadi SNDCP (*Sub Network Dependent Convergence Protocol*) dan LLC (*Logical Link Control*) protokol yang digunakan antara SGSN dan MS. Selain itu, menangani autentifikasi dari EDGE *mobile*, dan apabila autentifikasi sukses maka SGSN akan menangani registrasi UE pada jaringan EDGE serta manajemen mobilitasnya.

g) PCU (*Pakcet Control Unit*)

PCU yang berfungsi sebagai pengatur segmentasi paket. PCU akan memisahkan paket data dari *frame-frame* untuk ke SGSN ketika BSC menerima sinyal dari *mobile station*. Selain itu, PCU berfungsi untuk menyusun kembali paket data ke *frame-frame* ketika BSC mengirim sinyal ke MS.

2.1.2 *Air Interface*

2.1.2.1 *Radio Source Management*

Pada EDGE, kanal hanya akan ditempati ketika paket data dikirim dan diterima, sehingga sangat efisien untuk trafik yang padat. Dengan prinsip ini, beberapa *user* dapat berbagi satu *physical channel*. *Physical channel* ini disebut *packet data channel* (PDCH). PDCH merupakan kanal-kanal yang tersedia di dalam suatu sel sehingga *radio resources* dalam suatu sel dapat dibagi oleh UE EDGE maupun non-EDGE yang berada pada area cakupan sel tersebut.

2.1.2.2 *Channel Coding*

Channel coding berfungsi sebagai proteksi pada paket-paket data yang ditransmisikan dari kesalahan yang terjadi. Skema pengkodean memungkinkan tingkat perlindungan kesalahan dan kecepatan data yang berbeda yang juga dapat menyebabkan perubahan dalam data *throughput*. Sistem ini mendekripsi jumlah kesalahan bit dan

menyesuaikan skema pengkodean yang sesuai. Pada EDGE dikenal 9 macam skema pengkodean, yaitu MCS (*Modulation Coding Scheme*) 1 sampai dengan MCS 9. Tabel 2.1 menunjukkan modulasi dan *coding scheme* pada EDGE.

Tabel 2.1 Modulasi dan *Coding Scheme* pada EDGE

MCS	USF	Precoded USF	Header	HCS	Header Code Rate	Data	BCS	Data Code Rate	Data Rate in Kbps
MCS-1 DL	3	12	28	8	=1/2	178	12	=1/2	8.9
MCS-1 UL			31	8	=1/2	178	12	=1/2	8.9
MCS-2 DL	3	12	28	8	=1/2	226	12	0.64	11.3
MCS-2 UL			31	8	=1/2	226	12	0.64	11.3
MCS-3 DL	3	12	28	8	=1/2	298	12	0.83	14.9
MCS-3 UL			31	8	=1/2	298	12	0.83	14.9
MCS-4 DL	3	12	28	8	=1/2	354	12	=1	17.7
MCS-4 UL			31	8	=1/2	354	12	=1	17.7
MCS-5 DL	3	36	25	8	=1/3	450	12	=1/3	22.5
MCS-5 UL			37	8	=1/3	450	12	=1/3	22.5
MCS-6 DL	3	36	25	8	=1/3	594	12	=1/2	29.7
MCS-6 UL			37	8	=1/3	594	12	=1/2	29.7
MCS-7 DL	3	36	37	8	=1/3	2*450	12	=3/4	45
MCS-7 UL			46	8	=1/3	2*450	12	=3/4	45
MCS-8 DL	3	36	37	8	=1/3	2*546	12	0.90	54.6
MCS-8 UL			46	8	=1/3	2*546	12	0.90	54.6
MCS-9 DL	3	36	37	8	=1/3	2*594	12	=1	59.4
MCS-9 UL			46	8	=1/3	2*594	12	=1	59.4

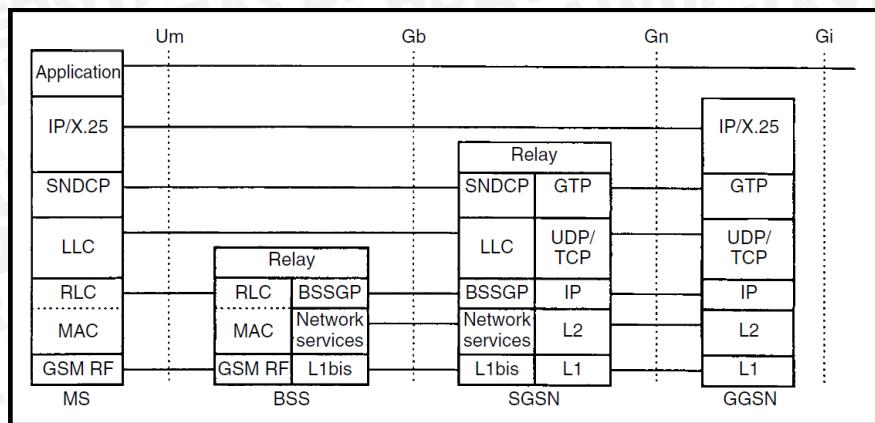
Sumber : Emmanuel Seurre, 2003

Coding scheme digunakan blok radio untuk membawa blok data RLC. Blok radio terdiri dari MAC header, blok data RLC atau RLC/MAC control block serta BCS (*block check sequence*). MAC header terdiri dari USF (*uplink state flag*), BTI (*block type indicator*) dan PC (*power control*). RLC data block terdiri dari RLC header dan RLC data. Sedangkan RLC/MAC control block terdiri dari elemen informasi *signalling* RLC dan MAC.

2.1.2.3 Protokol Stack pada Jaringan EDGE

Jaringan EDGE menggunakan protokol TCP/IP dalam pengiriman paket data mobile TV dari *broadcast content provider* ke UE. Gambar 2.2 menunjukkan protokol stack pada jaringan EDGE.



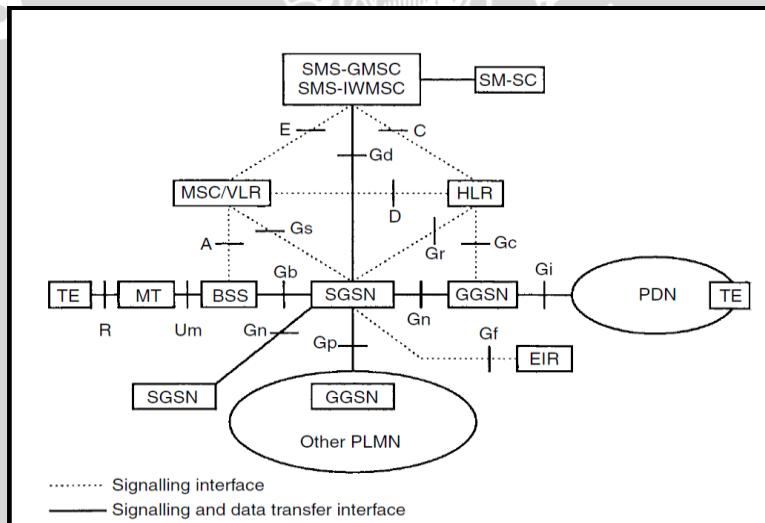


Gambar 2.2 Arsitektur Protokol EDGE

Sumber : Timo Halonen, 2003

2.1.2.4 Format Protokol pada Jaringan EDGE

Format protokol yang mendukung layanan GPRS dan eksternal berbasis IP antara lain Protokol Um *Interface*, Protokol Gb *Interface*, Protokol Gn *Interface*, dan Protokol Gi *Interface*. Masing-masing protokol ini akan dijelaskan secara singkat.



Gambar 2.3 Interface antar node pada EDGE

Sumber: Timo Halonen, 2003

a) Protokol Um Interface

Um *interface* menghubungkan antara UE dan BSS. Protokol yang berperan pada Um *interface* adalah RLC dan MAC.

- Fungsi utama dari RLC adalah segmentasi, yaitu proses pengambilan satu atau lebih frame LLC dan membaginya menjadi blok RLC yang lebih kecil.
- MAC berfungsi mengontrol prosedur akses pensinyalan untuk kanal radio, termasuk manajemen pembagian kanal transmisi (pengaturan blok radio untuk *multiple user* pada *timeslot* yang sama). Sebuah MAC header dan sebuah BCS ditambahkan pada blok data RLC untuk membentuk sebuah blok radio (*radio*



block). MAC *header* berisi *field* kontrol untuk transmisi *uplink* dan *downlink* yang panjangnya tetap, yaitu 8 bit. BCS digunakan untuk deteksi *error*. Proses *convolutional coding* diaplikasikan pada blok radio ini dengan menambahkan beberapa bit penutup (*tail bit*) kemudian membentuk blok radio yang telah dikodekan (*coded radio block*) dengan panjang tetap 456 bit. Jumlah bit informasi dalam unit transmisi bervariasi (181 sampai 428) tergantung pada *coding scheme* yang digunakan. Blok RLC yang diterima oleh UE dalam kondisi terjadi kesalahan akan ditransmisikan ulang dengan protokol ARQ (*Automatic Repeat Request*).

b) Protokol Gb Interface

Gb interface menghubungkan SGSN dengan PCU yang terletak pada BSS. Protokol-protokol pada *Gb interface* antara lain:

- SNDCP (*Sub Network Dependent Convergence Protocol*)

SNDCP digunakan untuk mentransfer paket data antara SGSN dan MS. *Header* SNDCP memiliki ukuran sebesar 4 *byte*.

- LLC (*Logical Link Control*)

Fungsi LLC meliputi *sequence control*, *flow control*, pendekripsi *error* transmisi, dan retransmisi (ARQ). *Header* LLC memiliki ukuran 4 *byte* dan digunakan sebagai kontrol informasi.

- BSSGP (*Base Station Subsystem GPRS Protocol*)

BSSGP berfungsi mengirimkan informasi *routing* antara BSS dan SGSN. Fungsi *relay* pada BSSGP mentransfer *frame* LLC antara *layer* RLC/MAC dan *layer* BSSGP. BSSGP memiliki kontrol informasi (*header*) sebesar 48 *byte*.

- NS (*Network Service*)

Layer network service merupakan koneksi *point-to-point* antara SGSN dan BSS. *Layer* NS berfungsi mengindikasi jalur *routing* antara SGSN dan BSS. *Layer* NS memiliki kapasitas transfer maksimum (*maximum transfer unit-MTU*) sebesar 1600 *byte* dan 4 *byte* *header* NS.

c) Protokol Gn Interface

Gn interface menghubungkan SGSN dan GGSN. Protokol-protokol pada *Gn interface* antara lain:

- GTP (*GPRS Tunnelling Protocol*)

GTP merupakan protokol pensinyalan yang menghubungkan paket data pengguna dan informasi pensinyalan antar GSN. GTP menerima IP datagram dari jaringan

eksternal dan menghubungkannya ke GGSN dan akan menambahkan *header* sebesar 20 byte.

- UDP (*User Datagram Protocol*)

UDP merupakan protokol *transport layer* yang berfungsi untuk mengirimkan paket-paket data yang disebut datagram dari satu *host* ke *host* yang lain melalui Gn *interface*.

- IP (*Internet Protocol*)

IP digunakan untuk *routing* data *user* dan informasi pensinyalan pada Gn *interface*.

IP datagram memiliki ukuran maksimum 65535 byte dan *header* sebesar 20 byte.

d) Protokol Gi Interface

Gi *interface* berfungsi menghubungkan GGSN dengan jaringan eksternal seperti jaringan internet berbasis IP. Protokol yang berperan dalam Gi *interface* ini adalah Ipsec (IP security) yang berfungsi sebagai *secure protocol* untuk menjamin keamanan dan kerahasiaan data yang dikirimkan ke jaringan privat.

2.2 HSDPA (*High Speed Downlink Pcket Access*)

HSDPA adalah suatu teknologi dalam sistem telekomunikasi bergerak yang termasuk dalam spesifikasi WCDMA (*Wideband Code Division Multiple Access*) 3GPP (*Third Generation Partnership Project*) *Release 5*. Penerapan HSDPA pada jaringan WCDMA merupakan sebuah evolusi teknologi yaitu dari teknologi 3G menjadi 3.5G, yang bertujuan untuk meningkatkan kemampuan performansi *downlink* jaringan WCDMA. Tujuan utama HSDPA adalah untuk meningkatkan *data rate* dan QoS, serta secara umum meningkatkan efisiensi spektral untuk layanan paket data *downlink*.

2.2.1 Karakteristik HSDPA

HSDPA merupakan pengembangan dari *air interface* pada jaringan WCDMA, yang bertujuan meningkatkan performansi WCDMA untuk mengatasi peningkatan *user* dan perkembangan komunikasi data yang mengarah pada komunikasi multimedia. Konsep HSDPA didasarkan pada beberapa karakteristiknya, yaitu: (Harri Holma, 2004:311)

2.2.1.1 *Shared channel transmission*

Beberapa kanal baru telah diperkenalkan pada operasi HSDPA, untuk transmisi HS-DSCH. Kanal-kanal tersebut yaitu:

a) HS-SCCH (*High Speed-Shared Control Channel*)

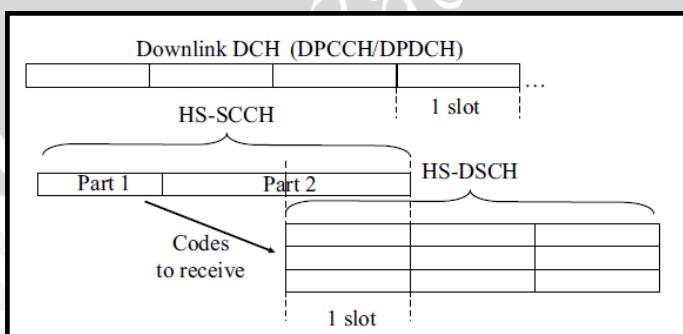
HS-SCCH adalah kanal pensinyalan *downlink* yang membawa informasi untuk UE (*User Equipment*). HS-SCCH membawa informasi *timing* dan *coding* diantaranya *channel code set*, skema modulasi, TBS (*Transport Block Size*) dan UE *identity* serta informasi lain seperti *redundancy* dan jumlah proses H-ARQ.

HS-SCCH memiliki dua slot untuk mengimbangi HS-DSCH, seperti yang ditampilkan dalam Gambar 2.4. HS-SCCH memiliki *spreading factor* sebesar 128 sehingga terdapat 40 bit per slot yang dibawa (dengan modulasi QPSK). Tidak terdapatnya bit pilot maupun kontrol daya membuat referensi fasa HS-SCCH selalu sama dengan HS-DSCH. HS-SCCH dibagi menjadi dua bagian yaitu:

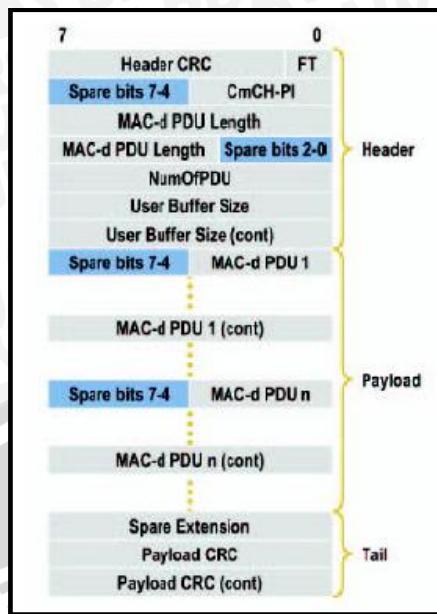
- Bagian pertama membawa informasi yang dibutuhkan untuk proses *despreading* kode yang benar dan juga membawa informasi modulasi.
- Bagian kedua terdiri dari informasi yang tidak terlalu mendesak, seperti proses ARQ mana yang tengah ditransmisikan. Selain itu bagian kedua juga mengindikasi apakah suatu transmisi paket benar-benar baru atau terkait dengan transmisi paket sebelumnya.

b) HS-DSCH (*High Speed- Downlink Shared Channel*)

HS-DSCH merupakan kanal transpor yang membawa data pengguna pada HSDPA. Pada *layer* fisik HS-DSCH dipetakan pada HS-PDSCH (*High-Speed Physical Downlink Shared Channel*). Karakteristik penting HS-DSCH adalah sifat dinamis yang memungkinkan *resource sharing* dengan alokasi periode 2 ms. Saat terdapat data untuk pengguna yang dialokasikan pada HS-PDSCH, data-data tersebut akan dikirimkan terus menerus selama periode TTI 2 ms. Struktur *frame* data HS-DSCH ditunjukkan dalam Gambar 2.5.



Gambar 2.4 Pewaktuan antara HS-SCCH dan HS-DSCH
Sumber: Harri Holma, 2006:41



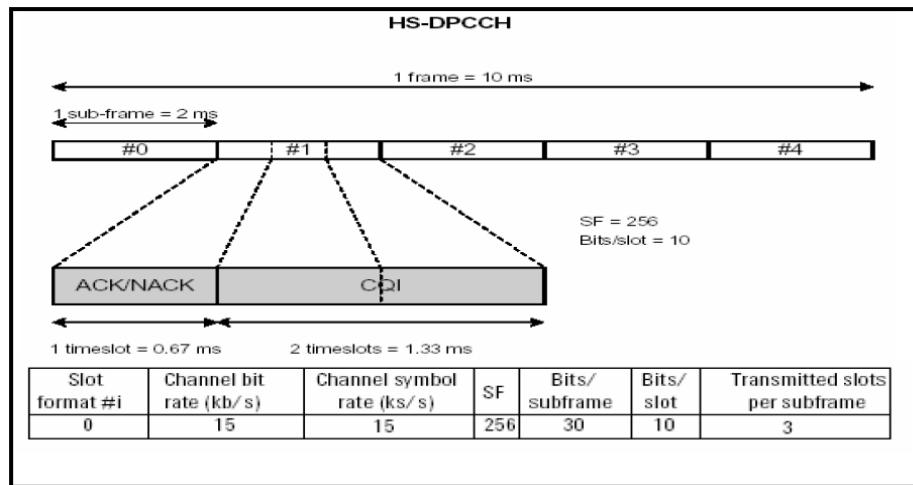
Gambar 2.5. Struktur Frame Data HS-DSCH

Sumber: www.blogspot.com, 2008

c) HS-DPCCH (*High Speed-Dedicated Physical Control Channel*)

Operasi HSDPA memerlukan informasi umpan balik *layer fisik uplink* dari terminal menuju *base station* untuk memungkinkan proses adaptasi *link* dan retransmisi *layer fisik*. Informasi umpan balik *uplink* tersebut dibawa oleh HS-DPCCH. Umpan balik HARQ (*Hybrid Automatic Repeat Request*) menginformasikan kepada *base station* apakah paket telah dikodekan dengan benar atau tidak. Sedangkan CQI (*Channel Quality Information*) memberitahukan *data rate* yang bisa diterima terminal kepada *base station scheduler*.

HS-DPCCH menggunakan *spreading factor* tetap yaitu sebesar 256 dan memiliki panjang *sub-frame* 2 ms yang terdiri dari 3 slot, masing-masing slot terdiri dari 2560 *chip*, seperti yang diilustrasikan dalam Gambar 2.6. Slot pertama dipergunakan untuk informasi HARQ. Sedangkan kedua slot yang tersisa digunakan untuk CQI. Informasi HARQ selalu dikirimkan saat ada HS-SCCH yang telah dikodekan dengan benar diterima pada arah *downlink*. CQI didasarkan pada CPICH (*Common Pilot Channel*) dan digunakan untuk perkiraan ukuran blok transport, tipe modulasi dan jumlah kanal kode yang mendukung keandalan level transmisi *downlink*.



Gambar 2.6 Struktur Frame HS-DPCCH

Sumber: Juha Korhonen, 2004:13

2.2.1.2 AMC (*Adaptive Modulation Coding*)

Pada *Release 5* untuk HS-PSCH, AMC menggantikan posisi kontrol daya. Tujuan utama AMC adalah untuk mengatasi ketidakstabilan kanal radio dengan jalan mengatur parameter transmisi. Hal ini dapat dilakukan dengan menggunakan pengukuran kanal radio yang diekstraksi oleh terminal *mobile*, dalam kasus HSDPA dengan cara memanfaatkan CQI dan prosedur retransmisi. Karena diperlengkap dengan informasi-informasi tersebut serta informasi yang berkaitan dengan trafik, AMC memampukan jaringan untuk memilih metode pengkodean dan modulasi yang paling sesuai. AMC juga mengijinkan seleksi modulasi untuk dikombinasikan dengan proses pengkodean kanal, yang sering disebut sebagai TRFC (*Transport Format and Resource Combination*). Sehingga berdasarkan pengukuran kanal, kombinasi terbaik dari multi kode, *channel rate* dan modulasi dapat dipilih untuk menghasilkan *throughput* maksimal untuk kondisi kanal tertentu.

Sehubungan dengan modulasi, *Release 5* memperbolehkan HSDPA, secara spesifik HS-DSCH, untuk menggunakan baik 16QAM (*16-Quadrature Amplitude Modulation*) maupun QPSK (*Quadrature Phase Shift Keying*). QPSK telah didefinisikan secara spesifik dalam *Release 4*, sedangkan 16QAM didefinisikan dalam *Release 5* untuk operasi HSDPA. Metode modulasi orde tinggi seperti 16QAM menyediakan efisiensi spektral yang lebih tinggi bila dibandingkan dengan QPSK. 16QAM melipatgandakan *data rate* dibandingkan dengan QPSK dengan cara merepresentasikan empat bit informasi per simbol modulasi. Namun aplikasi 16 QAM memiliki kelemahan yaitu hanya dapat diterapkan pada daerah dengan kondisi RF yang

baik karena metode modulasi 16QAM memerlukan informasi magnitudo dan fasa yang akurat untuk menentukan posisi simbol dalam keluaran konstelasi secara tepat.

2.2.1.3 TTI (*Short Transmission Time Interval*)

HSDPA menerapkan TTI sebesar 2 ms untuk sebuah *sub-frame* HS-DSCH yang terdiri dari 3 slot. TTI dapat mengurangi waktu *roundtrip* dan meningkatkan variasi kanal *tracking*. Sebuah UE dapat menggunakan semua kode kanalisasi sepanjang TTI atau beberapa UE menempati TTI yang sama dengan alokai kode kanalisasi yang berbeda.

2.2.1.4 Fast Link Adaptation

Adaptasi *link* pada HSDPA berdasarkan pada layer fisik CQI yang disediakan oleh terminal. Pengguna yang relatif lebih dekat dengan *base station* mentransmisikan level daya yang lebih tinggi daripada yang dibutuhkan. Adaptasi *link* mengatasi ekstra margin tersebut dengan menyeleksi parameter transmisi sehingga energi simbol yang dibutuhkan sesuai dengan daya simbol yang tersedia. Adaptasi *link* dilakukan berdasarkan informasi CQI yang juga mempertimbangkan aspek lain selain kekuatan sinyal atau C/I (*Carrier to Interference ratio*).

2.2.1.5 Fast scheduling

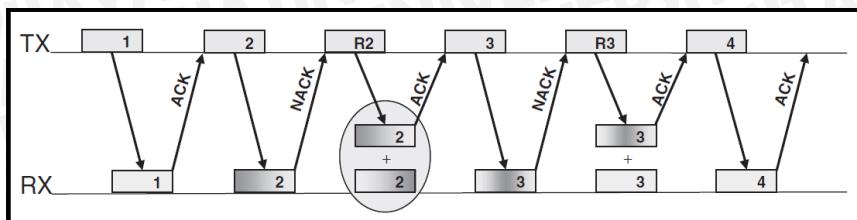
Fitur *fast-scheduling* menentukan UE tertentu dalam transmisi kanal bersama yang harus diarahkan pada waktu tertentu. Tujuannya adalah untuk mentransmisikan sinyal pada pengguna dengan kondisi radio yang terbaik.

2.2.1.6 H-ARQ (*Fast Hybrid Automatic Repeat Request*)

H-ARQ (*Hybrid-Automatic Repeat Request*) adalah sebuah teknik yang menggabungkan metode FEC (*Forward Error Correction*) dengan ARQ (*Automatic Repeat Request*) yang menyimpan informasi data sebelumnya untuk dikirimkan kembali saat terjadi kesalahan atau kegagalan dalam pengiriman, digunakan pada *decoding* selanjutnya. Teknik H-ARQ digunakan untuk retransmisi data yang diterapkan pada lapisan fisik (*Node B*). Retransmisi dapat dilayani langsung oleh *Node B* tanpa adanya keterlibatan dari RNC, sehingga diharapkan akan mengurangi *latency* dan meningkatkan utilitas kanal.

HARQ yang digunakan dalam HSDPA memanfaatkan prosedur *stop and wait*. Pada prosedur ini *transmitter* mengirimkan satu blok TTI (tiga slot) dan menunggu sampai *acknowledgement* (ACK) atau *negative acknowledgement* (NACK) diterima dari UE. Untuk memanfaatkan waktu selama menunggu ACK, proses *stop and wait* HARQ sebanyak N paralel diset untuk UE sehingga beberapa proses yang berbeda dapat

ditransmisikan dalam blok TTI yang berbeda. Bila UE meminta retransmisi untuk data *error* yang diterima sebelumnya, maka saat menerima transmisi yang kedua, UE menkombinasikan informasi dari transmisi sebelumnya dengan transmisi yang kedua sebelum mengkodekannya. Proses HARQ pada HSDPA ditampilkan pada Gambar 2.7.

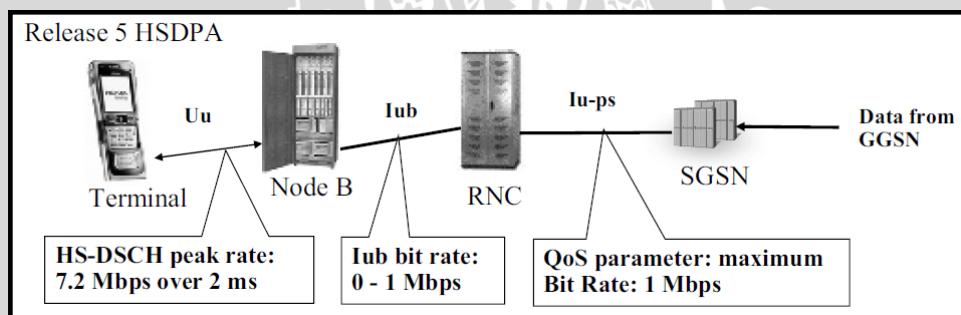


Gambar 2.7 Proses HARQ pada HSDPA

Sumber: Pablo Tapia, 2009:62

2.2.2 Arsitektur Jaringan HSDPA

HSDPA dapat diterapkan pada jaringan WCDMA *release* 1999. Penerapan HSDPA pada jaringan WCDMA *release* 1999, mengakibatkan perubahan pada UTRAN (*UMTS Terrestrial Radio Acces Network*) yang meliputi RNC (*Radio Access Controller*), Node B, Iub *interface* dan UE. Arsitektur jaringan WCDMA *release* 1999 yang menerapkan HDPA ditampilkan pada Gambar 2.8.



Gambar 2.8 Arsitektur Jaringan HSDPA

Sumber: Hari Holma, 2006:26

2.2.2.1 UE (*User Equipment*)/Terminal

UE merupakan perangkat atau terminal pada sisi pelanggan yang berupa *handset* untuk mengirim dan menerima informasi. HSDPA memiliki 12 tipe UE. Karakteristik dari 12 tipe UE berdasarkan maksimal kode HS-DSCH yang diterima, bit per TTI, interval inter-TTI maksimal, *peak data rates*, dan metode modulasi yang didukung, ditampilkan dalam tabel 2.2.

2.2.2.2 Node B

Node B digunakan untuk mentransmisikan data dari lub-*interface* dan Uu-*interface* atau sebaliknya, serta ikut mengontrol sumber radio. Fitur-fitur HSDPA seperti penerapan *shared channel transmission*, H-ARQ, *Fast link Adaptation AMC*, dan *Fast Scheduling* diterminasi langsung di Node B.



2.2.2.3 Iub Interface

Iub-interface berfungsi menghubungkan RNC (*Radio Network Controller*) dengan *Node B* menggunakan NBAP (*Node B Application Part*) signalling. Signalling dari *Node B* ke RNC dan sebaliknya menggunakan protokol NBAP yang dilewatkan melalui Iub-interface. Sumber daya yang dialokasikan untuk *Node B* dikirim dari controlling RNC melalui NBAP: *Physical Shared Channel Reconfiguration Request*. Kode kanalisasi yang dialokasikan untuk transmisi HSDPA juga memerlukan signalling antara RNC ke *Node B*.

Tabel 2.2 Kategori UE pada HSDPA

HS-DSCH category	HS-PDSCH codes	Modulation supported	Min. inter-TTI	Max. TBS [bits]	UE IR buffer size [SML]	Peak data rate [Mbps]
1	5	QPSK and 16-QAM	3	7298	19200	1.2
2	5	QPSK and 16-QAM	3	7298	28800	1.2
3	5	QPSK and 16-QAM	2	7298	28800	1.8
4	5	QPSK and 16-QAM	2	7298	38400	1.8
5	5	QPSK and 16-QAM	1	7298	57600	3.6
6	5	QPSK and 16-QAM	1	7298	67200	3.6
7	10	QPSK and 16-QAM	1	14411	115200	7.2
8	10	QPSK and 16-QAM	1	14411	134400	7.2
9	15	QPSK and 16-QAM	1	20251	172800	10.1
10	15	QPSK and 16-QAM	1	27952	172800	14.0
11	5	QPSK	2	3630	14400	0.9
12	5	QPSK	1	3630	14400	1.8

Sumber: Christope Chevallier, 2006:282

2.2.2.4 RNC (*Radio Network Controller*)

RNC digunakan untuk mengontrol sumber radio dimana *Node B* terhubung dengannya. RNC adalah layanan *access point* untuk semua layanan yang disediakan oleh NSS (*Network and Switching Sub-System*). Perbedaan utama fungsi RNC pada jaringan WCDMA dengan jaringan WCDMA-HSDPA adalah penanganan retransmisi. Semua kanal transport WCDMA diterminasi pada RNC. Oleh karena itu mekanisme retransmisi data paket berada pada layanan RNC. Dengan menerapkan HSDPA, yaitu dengan memperkenalkan HS-DSCH yang langsung diterminasi pada *Node B* yang mengakibatkan retransmisi lebih cepat dan *delay* yang lebih pendek.

2.2.2.5 Core Network

Core network terdiri dari beberapa bagian yaitu:

- SGSN (*Serving GPRS Support Node*)

SGSN mempunyai fungsi seperti MSC/VLR tetapi secara khusus digunakan untuk layanan *Packet Switched (PS)*.

- GGSN (*Gateway GPRS Support Node*)

GGSN berfungsi sebagai *interface* antara jaringan *backbone Packet Switching* dengan jaringan eksternal PDN (*Packet Data Network*), seperti jaringan internet. GGSN akan mengkonversikan paket data dari PDN menjadi PDP (*Packet Data Protocol*) seperti IP atau X.25, kemudian mengirimnya ke jaringan paket data yang dituju, demikian pula sebaliknya.

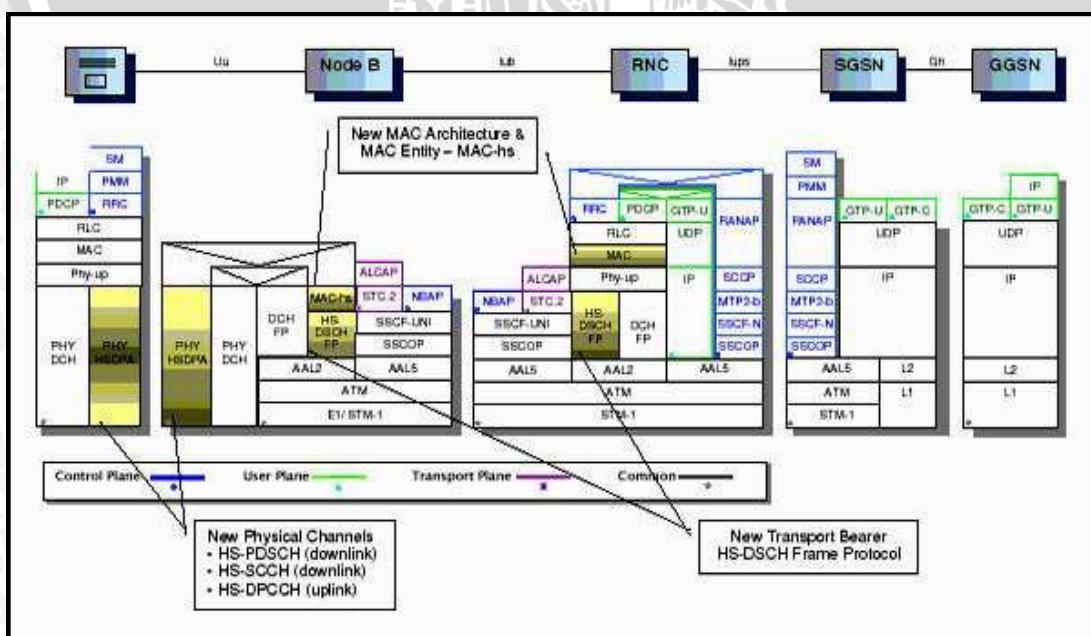
2.2.3 Protokol Stack HSDPA

Fungsionalitas *Node B* berbasis *fast scheduling* merupakan fungsionalitas *layer MAC*, sehingga diperlukan suatu protokol baru pada *Node B*, yaitu MAC-hs (hs untuk *high speed*). Protokol tersebut ditampilkan sebagai bagian dari arsitektur protokol *user plane* seperti Gambar 2.9.

RNC tetap mempertahankan MAC-d (d untuk *dedicated*), namun fungsi yang masih tersisa hanya sebagai *switching* kanal transport karena fungsi yang lain seperti *scheduling* dan prioritas penanganan telah dipindahkan ke MAC-hs. *Layer* di atas MAC yaitu *layer RLC* tidak ada perubahan. Berikut ini adalah penjelasan untuk tiap *layer*.

2.2.3.1 Physical Layer

Physical Layer mendukung transmisi dan penerimaan sinyal antara MS dan BS. *Physical layer* ini mengikuti model referensi OSI (*Open System Interconnection*) layer 1. Unit transmisi *physical layer* disebut dengan paket *layer* fisik.

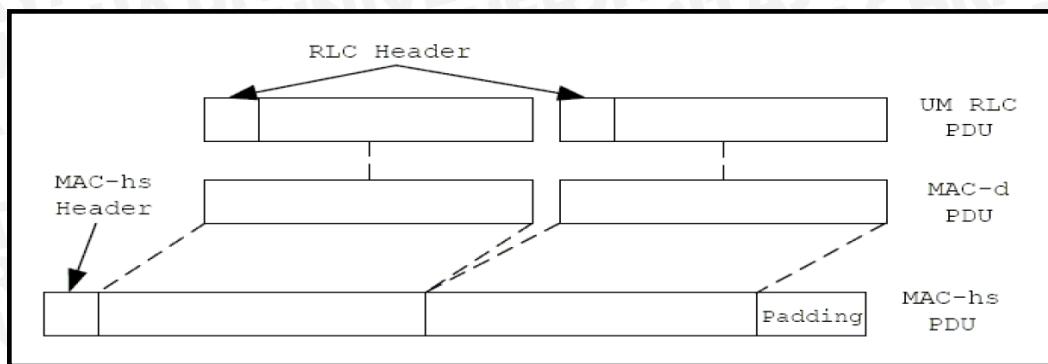


Gambar 2.9 Protokol Stack HSDPA

Sumber: Linda Ekowati, 2008

2.2.3.2 RLC (Radio Link Control)

RLC berfungsi menghubungkan *layer* kedua menuju *layer* di atasnya, menyalin data, transfer data, dan deteksi kesalahan transmisi yang gagal dari *layer* fisik.

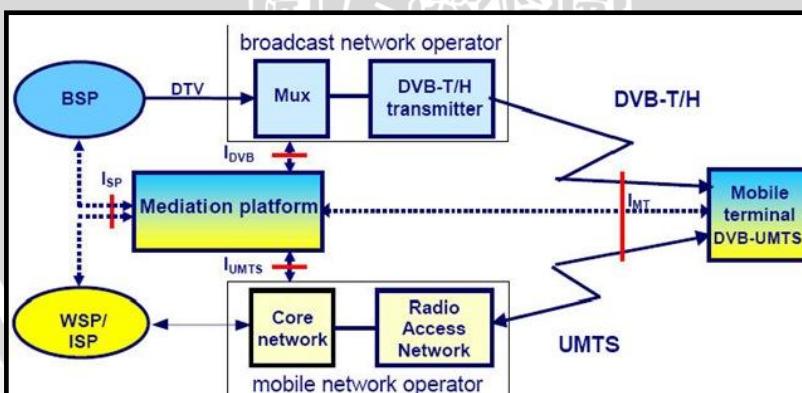


Gambar 2.10 Proses Pembentukan MAC PDU

Sumber: www.wireless.agilent.com, 2009

2.3 MobileTV

MobileTV adalah suatu sistem yang digunakan untuk menerima siaran televisi digital pada telepon seluler, dimana operator HSDPA me-relay siaran TV dari stasiun transmisi melalui sistem terestrial. Televisi ini juga bisa menerima sinyal dari stasiun *repeater* dengan memanfaatkan kembali BTS HSDPA (dan antennanya) sebagai *repeater* S-band, yaitu pada frekuensi 2.170-2.200 MHz. Salah satu teknologi yang digunakan untuk mentransmisikan *Mobile TV* adalah teknologi DVB-H berbasis standar DVB-T yang telah ditetapkan untuk penerimaan siaran TV digital.



Gambar 2.11 Skema Sistem MobileTV Berbasis Digital Video Broadcasting-Handheld
Sumber: Digitag, 2008

Layanan DVB-H pada prinsipnya beroperasi dengan menerima sinyal siaran TV digital yang dioptimalkan untuk perangkat telepon seluler melalui gelombang radio UHF. Perangkat telepon seluler televisi juga akan menerima sinyal dari jaringan seluler untuk mengendalikan tayangan TV sekaligus sebagai saluran interaksi melalui jaringan

seluler yang menggunakan operator jaringan HSDPA, karena membutuhkan kecepatan tinggi seperti aplikasi video.

Sistem DVB-H dapat menggunakan infrastruktur eksisting dari DVB-T dan menyediakan alat penerima *portable* (mudah dibawa), sehingga pengguna telepon seluler dapat bebas bergerak pada saat penerimaan siaran TV digital. DVB-H telah ditetapkan oleh *European Telecommunications Standards Institute* (ETSI) sebagai standar teknologi penyiaran *Mobile TV*.

Spesifikasi DVB-H meliputi:

- a) Time Slicing

DVB-H menggunakan mekanisme dimana data diterima pada suatu waktu, yang disebut dengan *IP datacast carousel*. Pada saat itu penerima tidak aktif untuk beberapa waktu. Dengan demikian level kontrol pensinyalan dimatikan, sehingga dapat menghemat daya hingga 90% atau lebih.

- b) Mode 4-K

DVB-H dapat berkompromi dengan lingkungan antara area SNF yang berkecepatan tinggi pada mode DVB-T 2-K dan area SNF yang berkecepatan rendah pada mode DVB-T 8-K. Dengan demikian DVB-H lebih tahan terhadap interferensi.

- c) MPE-FEC

Merupakan tambahan opsional, tingkat multiplexer. Sebuah *frame* MPE-FEC disusun dari ADT (*Application Data Table*) dan bit-bit paritas RSDT (*Reed-Solomon*) *Data Table* sesuai dengan standar DVB-H. Masing-masing IP *datagram* ditambah dengan *header* sebesar 12 *byte* yang berisi bit-bit informasi pada posisi awal dan akhir dalam *frame* MPE-FEC. Dan ditambahkan juga CRC-32 sebesar 4 *byte*.

2.3.1 RTP (*Real Time Protocol*)

RTP adalah *IP-based protocol* yang menyediakan dukungan *transport data* antara *real-time* seperti video dan audio *stream*, termasuk pada layanan *Mobile TV*. RTP digunakan untuk transport protokol dengan menggunakan UDP, TCP, ST-2, atau yang lainnya. RTP secara tipikal berada di atas IP untuk penggunaan dari *multiplexing* dengan fungsi *checksum*. UDP dipilih sebagai *protocol transport* untuk RTP karena RTP didesain untuk *multicast, connection oriented* TCP tidak memiliki skala yang baik. Selain itu untuk *real-time* data, *reliability* tidak penting untuk pengiriman secara waktu.

2.3.2 UDP (*User Datagram Protocol*)

UDP merupakan protokol yang berada di lapisan transport IP *layer*. Protokol ini bersifat *connectionless* dan *unreliable*. *Connectionless* berarti tidak diperlukan pembentukan hubungan terlebih dahulu untuk mengirimkan data. *Unreliable* berarti data tidak dijamin akan sampai di tujuan yang benar dan dalam kondisi yang benar pula. Protokol *transport* UDP lebih sering diimplementasikan untuk aplikasi yang mengarah pada proses *real-time*, seperti aplikasi multimedia pada layanan MobileTV, dimana rugi-rugi paket data yang lebih kecil akan lebih ditoleransi daripada nilai *delay* yang terjadi pada IP.

2.3.3 IP (*Internet Protocol*)

Protokol IP merupakan inti dari protokol TCP/IP, yang bersifat *unreliable*, *connectionless*, dan *datagram delivery service*. Banyak yang mengistilahkan sebagai *best effort delivery*, yang artinya IP menyediakan *no error checking* atau *tracking*. Jika diperlukan reliabilitas, maka IP harus dipasangkan dengan protokol yang *reliable*, misalnya TCP. IP mempunyai fungsi utama yaitu *connectionless-service*, pemecahan (*fragmentasi*) dan penyatuan dari paket-paket, serta *routing*.

a) Format Header Ipv4

Pada Jaringan EDGE menggunakan IP versi 4. Header IPv4 didesain mempunyai panjang *header* 20 dan memiliki 13 *field* yang lebih untuk menangani kejadian penggandaan paket dan TTL (*time-to-life*) paket. Format header alamat IPv4 ditunjukkan dalam Gambar 2.12.

+	Bits 0–3	4–7	8–15	16–18	19–31
0	Version	Header length	Type of Service (now DiffServ and ECN)		Total Length
32			Identification	Flags	Fragment Offset
64		Time to Live	Protocol		Header Checksum
96			Source Address		
128			Destination Address		
160			Options		
160 or 192+			Data		

Gambar 2.12 Format Header Ipv4

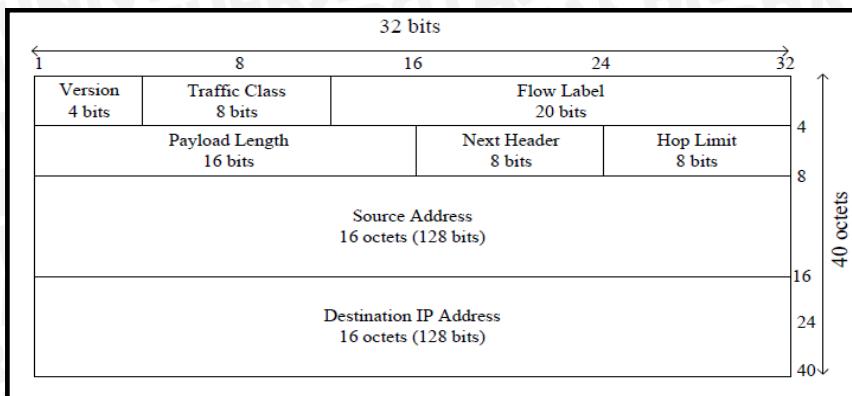
Sumber: www.wikipedia.com, 2008

b) Format Header Ipv6

Pada Jaringan HSDPA menggunakan IP versi 6. Header IPv6 didesain mempunyai panjang *header* 40 byte dan memiliki *field* lebih sedikit dibandingkan dengan IPv4, sehingga *router* melakukan penanganan tiap *field* lebih sedikit



menyebabkan proses *routing* atau aliran data menjadi lebih ringkas. Format *header* alamat IPv6 ditunjukkan dalam Gambar 2.13.



Gambar 2.13 Format *Header* Ipv6

Sumber: Pramono, 2006:11

2.3.4 Codec (*Coder/Decoder*) Mobile TV

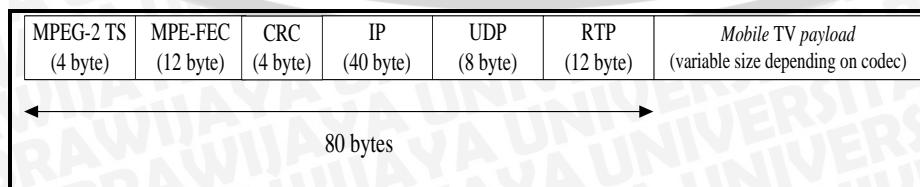
Codec merupakan bagian dari *hardware* maupun *software* yang memproses suatu sinyal (video maupun audio) dan mengonversinya ke dalam format tertentu untuk dikirimkan dengan protokol tertentu, dan pada sisi penerima juga harus menggunakan protokol yang sama. Teknik *codec* yang digunakan untuk aplikasi *mobile* TV dapat dilihat pada Tabel 2.3.

Tabel 2.3 Audio dan Video *Codec* pada Aplikasi *Mobile* TV

Audio CODEC	Bit Rate (kbps)	Maximum Payload (byte)	Delay CODEC (ms)
AMR-WB	6,6-23,85	35	10-20
AMR-WB+	5,2-48	46	20-40
HE-AAC v2	128-320	80	40-80
Video CODEC	Bit Rate (kbps)	Maximum Payload (byte)	Delay CODEC (ms)
H.264/AVC	64-384	254	150-300

Sumber: RFC 4352 and RFC 3984 RTP Payload Format for H.264 Video, 2005

Layanan *Mobile* TV mempunyai paket data seperti terlihat pada Gambar 2.14 terdiri *header* RTP, *header* UDP, *header* IP, *header* MPE-FEC, CRC, *header* MPEG-2 TS dan *payload* yang besarnya bergantung pada jenis *codec* yang dipakai.



Gambar 2.14 Paket Data *Mobile* TV

Sumber: Jenkac et al, J Zhejiang Univ SCIENCE A, 2006: 877



2.3.5 H.264/AVC

H.264/AVC merupakan standar pengkodean video terbaru yang dikembangkan oleh JVT (*Joint Video Team*) dari ISO / IEC MPEG dan ITU-T VCEG. Tujuan utama dari standarisasi H.264/AVC untuk meningkatkan performa kompresi dan penyediaan sebuah penyajian video yang “*network-friendly*” pada aplikasi video telepon, *Mobile TV*, penyimpanan, penyiaran atau *streaming*. H.264/AVC menambahkan berbagai fitur kode baru dan adanya perbaikan mekanisme, yang dapat dua atau tiga kali meningkat efisiensi pengkodean. Selain itu, H.264/AVC memungkinkan ukuran variabel blok partisi, *multiple frame* referensi, dan perkiraan kuartal gerak pixel. (Borko Furht, 2008:136)

2.3.6 AMR-WB (*Adaptive Multi Rate – WideBand*)

Adaptive Multi Rate – WideBand AMR-WB adalah standar pengkodean audio setelah AMR yang menggunakan teknologi yang sama seperti *Algebraic Code Excited Linear Prediction* (ACLEP). AMR-WB beroperasi seperti AMR dengan 9 kecepatan bit yang berbeda. Kecepatan bit terendah yang menyediakan kualitas percakapan terbagus dalam lingkungan bersih adalah 12,65 kbps. Kecepatan bit yang lebih tinggi berguna dalam kondisi latar belakang ber-*noise* dan dalam kasus music.

Kecepatan bit adalah sebagai berikut:

a) Konfigurasi *Mandatory Multi-rate*

- 6,60 kbit/s (digunakan untuk koneksi *circuit switched* GSM dan UMTS; hanya digunakan sementara selama koneksi radio buruk dan tidak dianjurkan pada percakapan *wideband*)
- 8,85 kbit/s (digunakan untuk koneksi *circuit switched* GSM dan UMTS; hanya digunakan sementara selama koneksi radio buruk dan tidak dianjurkan pada percakapan *wideband*; menyediakan kualitas yang sama dengan G.722 pada 48 kbps untuk percakapan jernih)
- 12,65 kbit/s (kecepatan bit utama main; digunakan untuk koneksi *circuit switched* GSM dan UMTS; menawarkan kualitas audio *superior* pada AMR pada dan diatas kecepatan bit ini; menyediakan kualitas yang sama atau lebih baik dari G.722 pada 56 kbps untuk percakapan jernih)

b) Kecepatan bit yang lebih tinggi untuk percakapan dalam lingkungan berlatar belakang *noise*, kombinasi percakapan dan musik, dan konferensi banyak peserta.

- 14,25 kbps

- 15,85 kbps
- 18,25 kbps
- 19,85 kbps
- 23,05 kbps (tidak ditujukan untuk kanal GSM berkecepatan penuh)
- 23,85 kbps (menyediakan kualitas yang sama dengan G.722 pada 64 kbps untuk percakapan jernih; tidak ditujukan untuk kanal GSM berkecepatan penuh)

2.4 Handover

Handover adalah proses pengalihan antara satu kanal tehadap kanal yang lain. *Handover* diperlukan sistem seluler untuk menjamin adanya kontinyuitas komunikasi apabila pelanggan bergerak dari satu *cell* ke *cell* lain. Proses ini sangat penting dalam komunikasi seluler melihat latar belakang pamakaian telepon seluler yang penuh mobilitas, jumlah trafik, dan kualitas sinyal, sehingga tidak terjadi *drop call* atau terputusnya hubungan.

Handover dapat terjadi dalam kaitan dengan pergantian sumber radio yang menyediakan suatu layanan dengan melakukan perubahan yang melibatkan *base station*. Hal ini terjadi khususnya ketika sumber radio menyediakan pergantian layanan dari mode akses radio UTRA (*UMTS Terrestrial Radio Access*) yang satu ke UTRA yang lainnya.

2.4.1 Jenis *Handover* pada Sistem Komunikasi Bergerak

Dengan semakin berkembangnya teknologi *wireless selular*, maka definisi mengenai proses *handover* menjadi semakin meluas tidak sebatas untuk satu jenis teknologi yang digunakan. Oleh sebab itu *handover* sendiri dapat dikategorikan dalam beberapa jenis yang dapat dijelaskan sebagai berikut : (Areej Saleh, 2004 : 17)

2.4.1.1 Berdasarkan Teknologi yang Digunakan:

- a) Horizontal *Handover*, adalah proses *handover* yang terjadi pada sebuah *User Equipment* (UE) yang mengalami perpindahan antar *base station* dengan teknologi yang sama, hal ini juga sering dikenal sebagai *intra-technology handoff*.
- b) Vertical *Handover*, adalah proses *handover* yang terjadi pada sebuah UE yang melibatkan dua *interface* jaringan yang berbeda dengan teknologi yang berbeda pula. *Handover* ini juga dikenal sebagai *inter-technology handoff*.

2.4.1.2 Berdasarkan Domain Administratif:

- a) *Intra-Administrative Domain Handover*, adalah proses *handover* yang terjadi pada sebuah UE yang mengalami perpindahan antar *base station* dengan teknologi yang sama maupun berbeda namun keduanya dikelola dalam satu *domain* administratif.
- b) *Inter-Administrative Domain Handover*, adalah proses *handover* yang terjadi pada sebuah UE yang mengalami perpindahan antar *base station* dengan teknologi yang sama ataupun berbeda namun keduanya dikelola dalam *domain* administratif yang berbeda.

2.4.1.3 Berdasarkan Konektivitas:

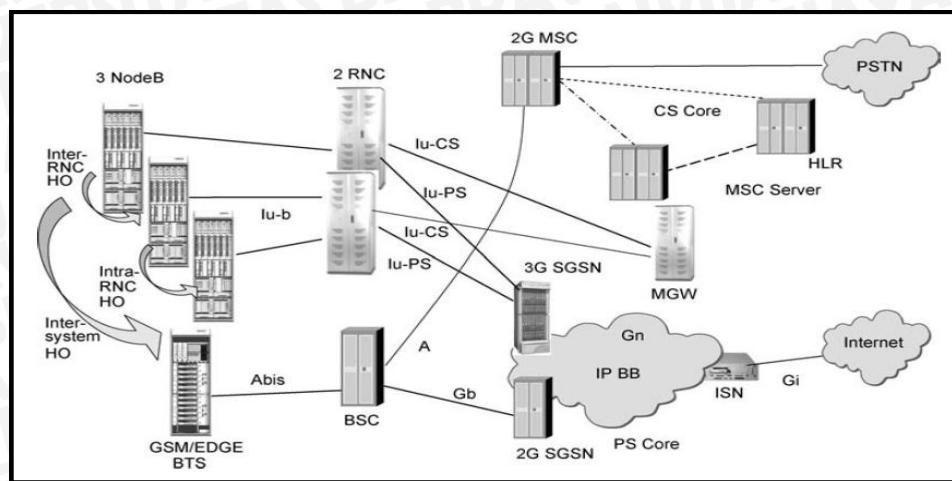
- a) *Hard Handover*, terjadi pada sebuah UE dengan melakukan pemutusan hubungan dengan *base station* yang sebelumnya untuk kemudian melakukan hubungan baru dengan *base station* pada sel yang lain, proses ini lebih dikenal dengan “*break before make*”.
- b) *Soft Handover*, terjadi pada sebuah UE yang melakukan hubungan baru terlebih dahulu saat melakukan perpindahan sel dengan maksud untuk menjaga konektivitasnya, atau dikenal dengan “*make before break*”.
- c) *Softer Handover*, *handover* ini hampir mirip dengan *soft handover*, namun UE mengalami perpindahan konektifitas *radio link* dalam sebuah *base station* yang sama.

2.4.1.4 Berdasarkan Frekuensi yang digunakan:

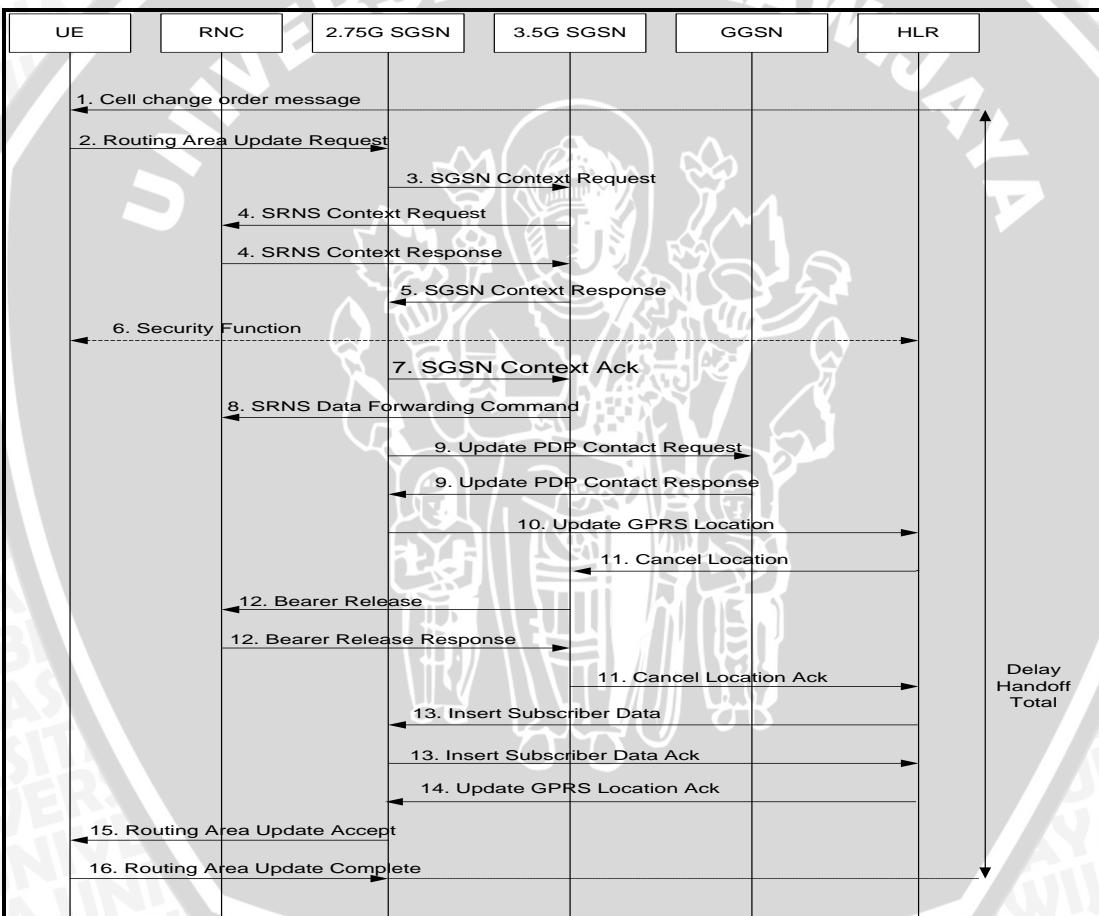
- a) *Intra-Frequency Handover*, adalah proses *handover* yang terjadi pada UE yang mengalami perpindahan antar sel dengan frekuensi kerja yang sama.
- b) *Inter-Frequency Handover*, adalah proses *handover* yang terjadi pada UE yang mengalami perpindahan antar sel dengan frekuensi kerja yang berbeda.

2.4.2 *Handover* Jaringan HSDPA ke EDGE

Handover pada jaringan HSDPA ke EDGE merupakan *handover* dari HS-DSCH ke DCH, yang diilustrasikan pada Gambar 2.15. *Handover* ini diperlukan oleh pelanggan HSDPA yang bergerak dari satu sel dengan jaringan HSDPA ke suatu sel tanpa jaringan HSDPA (Harri Holma, 2006:105). *Handover* ini sering juga disebut sebagai *Intersystem Handover HSDPA to EDGE*.

Gambar 2.15 *Intersystem Handover HSDPA ke EDGE*

Sumber: Pablo Tapia, 2009:119

Gambar 2.16 Prosedur *signaling handover HSDPA ke EDGE*

Sumber: Samsung dan Hari Holma, 2006:204

Tahap-tahap *handover* tersebut dapat dijelaskan sebagai berikut:

- Pada saat RNC mendeteksi bahwa jaringan HSDPA semakin melemah, RNC akan meminta UE untuk melakukan pengukuran target sel dari jaringan EDGE. Jika target sel yang paling bagus dari jaringan EDGE telah diidentifikasi, RNC akan

memerintahkan UE untuk berpindah ke jaringan EDGE dengan pesan *Cell Change Order*.

- b) UE akan mengirim pesan *Routing Area Update Request* ke 2.75G_SGSN.
- c) 2.75G_SGSN mengirimkan SGSN *Context Request* ke 3.5G_SGSN untuk mendapatkan MM (*Mobility Management*) dan PDP *Context* untuk UE.
- d) Pesan SRNS (*Serving Radio Network Subsystem*) dari 3.5G_SGSN dalam bentuk *SRNS Context Request* memberitahukan RNC untuk mulai *buffering* dan tidak mengirim data ke Node B lagi. RNC merespon dengan *SRNS Context Response*.
- e) 3.5G_SGSN merespon dengan SGSN *Context Response*.
- f) Fungsi pengamanan (*Security Function*) dilaksanakan.
- g) Prosedur selesai saat 2.75G_SGSN mengirim pesan SGSN *Context Acknowledge*, dan ini berarti bahwa 2.75G_SGSN telah siap menerima data dari paket sebelumnya.
- h) 3.5G_SGSN mengirim pesan SRNS *Data Forwarding Command* ke RNC, pesan ini memerintahkan RNC untuk memulai pengiriman data melalui 2.75G_SGSN, dan mengirim data tersebut ke UE melalui jaringan EDGE.
- i) 2.75G_SGSN mengirim *Update PDP Context Request* ke GGSN yang terhubung. GGSN memperbarui PDP *Context* field-nya dan mengembalikan *Update PDP Context Response*.
- j) 2.75G_SGSN menginformasikan HLR atas perubahan SGSN dengan mengirimkan *Update GPRS Location* pada HLR.
- k) HLR mengirim *Cancel Location* 3.5G_SGSN. Kemudian 3.5G_SGSN memindahkan MM dan PDP *Context*. Selanjutnya 3.5G_SGSN menyetujui dengan mengirim pesan *Cancel Location Ack*.
- l) 3.5G_SGSN meminta RNC untuk melepaskan sumber radio dengan mengirimkan *Bearer Release*. RNC merespon dengan mengirimkan *Bearer Release Response*
- m) HLR mengirimkan *Insert Subscriber Data* (*Subscription Data*) pada 2.75G_SGSN. 2.75G_SGSN membuat suatu MM *context* untuk UE dan mengirimkan pesan *Insert Subscriber Data Ack* ke HLR.
- n) HLR menyetujui *Update Location* dengan mengirimkan pesan *Update GPRS Location Acknowledge* ke 2.75G_SGSN.
- o) Kemudian 2.75G_SGSN merespon UE dengan mengirim pesan *Routing Area Update Accept*.
- p) UE menyetujui dan memberitahukannya dengan mengirim pesan *Routing Area Update Complete*.

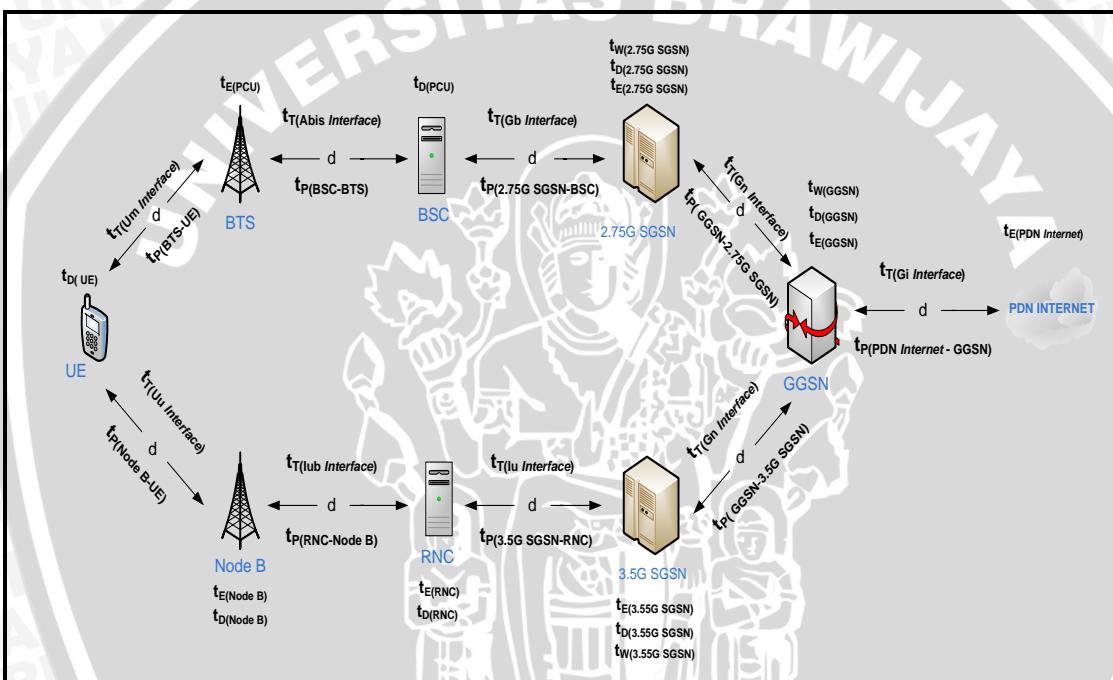
2.5 Parameter Performansi *Handover* dari Jaringan HSDPA ke EDGE

2.5.1 Delay

Definisi umum dari *delay* adalah waktu tunda. Pada aplikasi *Mobile TV*, *delay* sangat berpengaruh pada kualitas layanan. Semakin kecil *delay* yang dihasilkan akan semakin bagus pula kualitas layanan *Mobile TV* tersebut.

2.5.1.1 Delay end-to-end

Dalam jaringan yang berbasiskan *packet switching*, *delay* yang terjadi merupakan penjumlahan *delay-delay* yang ada dalam perjalanan paket dari sumber ke tujuan. *Delay end-to-end* diilustrasikan dalam Gambar 2.17.



Gambar 2.17 Arsitektur *Node-to-node* HSDPA ke EDGE

Sumber: Perancangan

Delay end-to-end dapat dihitung dengan menggunakan persamaan berikut (Anurag Kumar, 2008:65):

$$t_{net} = t_{proc} + t_T + t_p + t_W \quad (s) \quad (2.1)$$

dimana:

t_{net} = *delay end-to-end* pada jaringan EDGE dan HSDPA (s)

t_{proc} = *delay proses* (s)

t_T = *delay transmisi* (s)

t_p = *delay propagasi* (s)

t_W = *delay antrian* (s)



2.5.1.2 Delay Proses

Delay proses merupakan waktu yang dibutuhkan untuk memproses sebuah data agar dapat ditransmisikan dan untuk menentukan kemana data tersebut akan diteruskan. *Delay* proses terdiri dari *delay* enkapsulasi dan dekapsulasi. Proses ini akan terjadi pada semua *node* yang dilalui data dalam arsitektur jaringan, yaitu PDN Internet, GGSN (*Gateway GPRS Support Node*), 2.5G SGSN (*2.5G Service GPRS Support Node*), BSC (*Base Station Controller*), BTS (*Base Transceiver Station*), 3G SGSN (*3G Service GPRS Support Node*), RNC (*Radio Network Controller*), *Node B*, dan UE (*User Equipment*).

a) PDN Internet

Payload data yang telah ditambahkan dengan *header* RTP, *header* UDP, *header* IP, *header* MPE-FEC dan CRC, serta *header* MPEG-2 TS pada *Broadcast Content Provider* akan ditambahkan dengan *header* UDP sehingga diperoleh panjang segmen UDP yaitu berdasarkan persamaan berikut:

$$W_{segmen} = W_{mobile\ TV\ (PDN\ Internet)} + header_{UDP} \quad (byte) \quad (2.2)$$

dengan:

$$W_{segmen} = \text{panjang segmen UDP (byte)}$$

$$W_{mobile\ TV\ (PDN\ Internet)} = \text{panjang paket data aplikasi mobile TV di host mobile TV (byte)}$$

$$header_{UDP} = \text{header UDP (8 byte)}$$

Apabila panjang segmen UDP melebihi MSS (*Maximum Segmen Size*) IP sebesar 65535 byte (rekomendasi IEEE:RFC 0791), maka segmen UDP tersebut akan disegmentasi terlebih dahulu menjadi beberapa *frame*, dan kemudian ditambahkan dengan *header* IP. Apabila segmen UDP yang diterima tidak melebihi MSS IP, maka akan langsung ditambahkan dengan *header* IP menjadi *datagram* IP. Karena IP yang digunakan adalah IPv6, maka besar *header* IP adalah 40 byte. Panjang *datagram* IP diperoleh berdasarkan persamaan berikut (Uke Kurniawan, 2009:39):

$$W_{datagram} = W_{segmen} + header_{IP} \quad (byte) \quad (2.3)$$

Dengan:

$$W_{datagram} = \text{panjang datagram IP (byte)}$$

$$W_{segmen} = \text{panjang segmen UDP (byte)}$$

$$header_{IP} = \text{panjang header IP (40 byte)}$$

Kemudian pada *data link layer*, apabila *datagram IP* yang diterima melebihi MTU (*Maximum Transmission Unit*) *Ethernet* sebesar 1500 byte, maka *datagram IP* tersebut akan disegmentasi terlebih dahulu kemudian ditambahkan *header Ethernet* dan FCS. Apabila *datagram IP* tidak melebihi MTU *Ethernet*, *datagram IP* akan langsung ditambahkan dengan *header Ethernet* dan FCS. Sehingga jumlah total *frame* yang siap ditransmisikan ke GGSN yaitu berdasarkan persamaan berikut:

$$W_{PDN\ Internet} = W_{datagram} + header_{Ethernet} + FCS \quad (\text{byte}) \quad (2.4)$$

Dimana:

$W_{PDN\ Internet}$ = panjang *frame PDN Internet* (byte)

$W_{datagram}$ = panjang *datagram IP* (byte)

$header_{Ethernet}$ = *header Ethernet* (14 byte)

FCS = panjang *Frame Check Sequence* (4 byte)

Sehingga *delay enkapsulasi* pada *PDN Internet* didapatkan berdasarkan persamaan berikut (Uke Kurniawan, 2009:40):

$$t_{E(PDN\ Internet)} = \frac{W_{frame\ PDN\ Internet} \times 8}{V_{PDN\ Internet}} \quad (\text{s}) \quad (2.5)$$

Dengan:

$t_{E(PDN\ Internet)}$ = *delay enkapsulasi* pada *PDN Internet* (s)

$W_{frame\ PDN\ Internet}$ = panjang *frame PDN Internet* (byte)

$V_{PDN\ Internet}$ = kecepatan transmisi data di *PDN Internet* (bps)

b) GGSN (*Gateway GPRS Support Node*)

Pada GGSN paket yang diterima dari *PDN Internet* akan mengalami proses dekapsulasi. *Frame PDN Internet* yang diterima di GGSN akan didekapsulasi untuk mendapatkan data aplikasi sesuai dengan persamaan berikut:

$$W_{mobile\ TV(GGSN)} = W_{frame\ PDN\ Internet} - header_{IP} - header_{UDP} \\ - header_{Ethernet} - FCS \quad (\text{byte}) \quad (2.6)$$

dengan:

$W_{mobile\ TV(GGSN)}$ = panjang data aplikasi *mobile TV* di GGSN (byte)

$W_{frame\ PDN\ Internet}$ = panjang *frame PDN Internet* (byte)

$header_{IP}$ = *header IP* (40 byte)

$header_{UDP}$ = *header UDP* (8 byte)

$header_{Ethernet}$ = *header Ethernet* (14 byte)

FCS = panjang *Frame Check Sequence* (4 byte)

Pada skripsi ini menggunakan GGSN dengan tipe *Cisco 7600 Series Router* dengan teknologi *Gigabit Ethernet* dengan transmisi data sebesar 10 Gbps. Nilai *delay* dekapsulasi pada GGSN didapatkan dengan persamaan berikut (Uke Kurniawan, 2009:40):

$$t_{D(GGSN)} = \frac{W_{mobile\ TV(GGSN)} \times 8}{V_{GGSN}} \quad (s) \quad (2.7)$$

dimana:

$t_{D(GGSN)}$ = *delay* dekapsulasi di GGSN (s)

$W_{mobile\ TV(GGSN)}$ = panjang data aplikasi *mobile TV* di GGSN (byte)

V_{GGSN} = kecepatan transmisi data di GGSN (bps)

Pada GGSN apabila data aplikasi yang diterima melebihi MTU maka disegmentasi menjadi *frame* GGSN sesuai dengan MTU *Ethernet* sebesar 1500 byte. Apabila data aplikasi yang diterima tidak melebihi MTU *Ethernet*, maka *frame* tersebut langsung dienkapsulasi dengan menambahkan *header GTP*, UDP, IP (3GPP TS 29.060 Rel. 5) dan *header Ethernet* serta FCS sebelum ditransmisikan ke SGSN yaitu sesuai dengan persamaan berikut:

$$W_{frame\ GGSN} = W_{mobile\ TV(GGSN)} + header_{GTP} + header_{UDP} + header_{IP} + header_{Ethernet} + FCS \quad (byte) \quad (2.8)$$

dengan:

$W_{frame\ GGSN}$ = panjang *frame* GGSN (byte)

$W_{mobile\ TV(GGSN)}$ = panjang data aplikasi *mobile TV* di GGSN (byte)

$header_{GTP}$ = *header GTP* (8 byte)

$header_{UDP}$ = *header UDP* (8 byte)

$header_{IP}$ = *header IP* (40 byte)

$header_{Ethernet}$ = *header Ethernet* (14 byte)

FCS = panjang *Frame Check Sequence* (4 byte)

Nilai *delay* enkapsulasi yang terjadi di GGSN adalah:

$$t_{E(GGSN)} = \frac{W_{frame\ GGSN} \times 8}{V_{GGSN}} \quad (s) \quad (2.9)$$

dengan:

$t_{E(GGSN)}$ = *delay* enkapsulasi di GGSN (s)

$W_{frame\ GGSN}$ = panjang *frame* di GGSN (byte)

V_{GGSN} = kecepatan transmisi data di GGSN (bps)

c) 2.75G SGSN (2.75 Serving GPRS Support Node)

Pada 2.5G SGSN terjadi proses dekapsulasi terhadap data yang diterima dari GGSN, untuk mendapatkan data aplikasi. Paket data tersebut didekapsulasi dengan persamaan:

$$W_{mobile\ TV\ (2.75G\ SGSN)} = W_{frame\ GGSN} - header_{GTP} - header_{UDP} - header_{IP} - header_{Ethernet} - FCS \quad (2.10)$$

dengan:

$W_{mobile\ TV\ (2.75G\ SGSN)}$ = panjang data aplikasi mobile TV di 2.75G SGSN (byte)

$W_{frame\ GGSN}$ = panjang frame GGSN (byte)

$header_{GTP}$ = header GTP (8 byte)

$header_{UDP}$ = header UDP (8 byte)

$header_{IP}$ = header IP (40 byte)

$header_{Ethernet}$ = header Ethernet (14 byte)

FCS = panjang Frame Check Sequence (4 byte)

Pada skripsi ini 2.75G SGSN menggunakan *combined GSN* dengan kecepatan 15 Mbps. Nilai *delay* dekapsulasi pada 2.75G SGSN diperoleh dengan persamaan:

$$t_D(2.75G\ SGSN) = \frac{W_{mobile\ TV\ (2.75G\ SGSN)} \times 8}{V_{2.75G\ SGSN}} \quad (s) \quad (2.11)$$

dimana:

$t_D(2.75G\ SGSN)$ = *delay* dekapsulasi di 2.75G SGSN (s)

$W_{mobile\ TV\ (2.75G\ SGSN)}$ = panjang data aplikasi mobile TV di 2.75G SGSN (byte)

$V_{2.75G\ SGSN}$ = kecepatan transmisi data di 2.75G SGSN (bps)

Pada 2.75G SGSN, apabila data aplikasi yang diterima melebihi MSS SNDCP (*Sub Network Dependent Convergence Protocol*) maka disegmentasi menjadi *frame* SNDCP, sesuai dengan MSS SNDCP sebesar 1503 byte. Apabila data aplikasi yang diterima tidak melebihi MSS SNDCP, maka *frame* tersebut langsung dienkapsulasi dengan menambahkan *header* SNDCP, LLC, BSSGP, NS. (3GPP TS 29.060 Rel. 5) dan FCS pada *layer* SNDCP untuk deteksi *error* sebelum ditransmisikan ke BSC, sesuai persamaan berikut:

$$W_{frame\ 2.75G\ SGSN} = W_{mobile\ TV\ (2.75G\ SGSN)} + header_{SNDNP} + FCS + header_{LLC} + header_{BSSGP} + header_{NS} \quad (byte) \quad (2.12)$$

dengan:

$W_{frame \ 2.75G \ SGSN}$	= panjang frame di 2.75G SGSN (byte)
$W_{mobile \ TV(2.75G \ SGSN)}$	= panjang data aplikasi mobile TV di 2.75G SGSN (byte)
$header_{SNDCP}$	= header SNDCP (4 byte)
$header_{LLC}$	= header LLC (4 byte)
$header_{BSSGP}$	= header BSSGP (48 byte)
$header_{NS}$	= header NS (4 byte)
FCS	= panjang Frame Check Sequence (4 byte)

Sehingga *delay* enkapsulasi pada 2.75G SGSN didapatkan dengan persamaan:

$$t_{E(2.75G \ SGSN)} = \frac{W_{frame \ 2.75G \ SGSN} \times 8}{V_{2.75G \ SGSN}} \quad (s) \quad (2.13)$$

Dengan:

$t_{E(2.75G \ SGSN)}$	= <i>delay</i> enkapsulasi di 2.75G SGSN (s)
$W_{frame \ 2.75G \ SGSN}$	= panjang frame 2.75G SGSN (byte)
$V_{2.75G \ SGSN}$	= kecepatan transmisi data di 2.75G SGSN (bps)

d) PCU di BSS (*Base Station Subsystem*)

Pada BSC, frame 2.75G SGSN didekapsulasi, dengan cara melepaskan *header-header* yang ada pada proses enkapsulasi di 2.5 SGSN. Sehingga untuk mendapatkan data apliksi diperoleh dengan persamaan berikut :

$$W_{mobile \ TV(BSS)} = W_{frame \ 2.5G \ SGSN} - header_{SNDCP} - FCS - header_{LLC} - header_{BSSGP} - header_{NS} \quad (byte) \quad (2.14)$$

dengan:

$W_{mobile \ TV(BSS)}$	= panjang data aplikasi mobile TV di BSS (byte)
$W_{frame \ 2.5G \ SGSN}$	= panjang frame di 2.5G SGSN (byte)
$header_{SNDCP}$	= header SNDCP (4 byte)
$header_{LLC}$	= header LLC (4 byte)
$header_{BSSGP}$	= header BSSGP (48 byte)
$header_{NS}$	= header NS (4 byte)
FCS	= panjang Frame Check Sequence (4 byte)

Pada BSS, kecepatan kearah Gb *interface* adalah kapasitas maksimum RPP per PCU (14 RPP) dikalikan dengan kecepatan standard E1-link (31x64 Kbps). Sehingga nilai *delay* dekapsulasi pada BSS didapatkan dengan persamaan:

$$t_{D(PCU)} = \frac{W_{mobile \ TV(PCU)} \times 8}{V_{1(PCU)}} \quad (s) \quad (2.15)$$



Dengan:

- $t_{D(PCU)}$ = delay dekapsulasi pada BSS (S)
 $W_{mobile TV(PCU)}$ = panjang data aplikasi mobile TV di BSS (byte)
 $V_{1(PCU)}$ = kecepatan transmisi data di PCU ke arah Gb interface (bps)

Selanjutnya *frame LLC* disegmentasi menjadi RLC/MAC *radio block* untuk dapat ditransmisikan pada Abis *interface* menuju BTS dengan ukuran tetap yaitu 456 bit. Masing masing *Coding Scheme* memiliki nilai RLC data yang berbeda-beda sesuai dengan Tabel 2.4 berikut ini.

Tabel 2.4 Ukuran RLC pada EDGE

MCS	EGPRS RLC Data Unit Size in Bytes	Additional Bit in the Data Part
MCS-1	22	2
MCS-2	28	2
MCS-3	37	2
MCS-4	44	2
MCS-5	56	2
MCS-6	74	2
MCS-7	$2^{*}56$	$2^{*}2$
MCS-8	$2^{*}68$	$2^{*}2$
MCS-9	$2^{*}74$	$2^{*}2$

Sumber: Emmanuel Seurre, 2003

$$N_{frame RLC block} = \frac{W_{frame LLC} \times 8}{RLC_{data unit}} \quad (2.16)$$

dengan:

- $W_{frame LLC}$ = panjang *frame LLC* (byte)
 $RLC_{data unit}$ = panjang maksimum informasi bit pada *coding scheme* (bit)

Sedangkan panjang *frame LLC* didapatkan dengan persamaan:

$$W_{frame LLC} = W_{mobile TV(PCU)} + header_{SNDCP} + FCS + header_{LLC} \text{ (byte)} \quad (2.17)$$

dengan:

- $W_{frame LLC}$ = panjang *frame LLC* (byte)
 $W_{mobile TV(PCU)}$ = panjang *frame* di PCU (byte)
 $header_{SNDCP}$ = *header SNDCP* (4 byte)
 $header_{LLC}$ = *header LLC* (4 byte)
 FCS = panjang *Frame Check Sequence* (4 byte)

Setelah itu tiap segmennya akan dienkapsulasi dengan menambahkan :

RLC/MAC radio block = 3 bit USF + 181 bit RLC data + 40 bit BCS + 4 tail bit. Maka *delay enkapsulasi* yang terjadi di PCU adalah sebagai berikut :



$$t_E (PCU) = \frac{\text{jumlah } h \text{ frame } LLC \times N_{frame RLC block} \times RLC_{radio block}}{V_{2(PCU)}} \quad (\text{s}) \quad (2.18)$$

dengan:

- | | |
|-----------------------|---|
| $t_E (PCU)$ | = delay enkapsulasi di PCU (s) |
| $N_{frame RLC block}$ | = jumlah blok data masing-masing <i>coding scheme</i> |
| $RLC_{radio block}$ | = panjang paket data RLC |
| $V_{2(PCU)}$ | = kecepatan PCU ke arah Abis <i>Interface</i> (bps) |

e) UE (*User Equipment*) pada EDGE

RLC/MAC *radio block* yang diterima UE dari BTS melalui Um *Interface* selanjutnya akan mengalami proses dekapsulasi menjadi paket-paket IP, dengan *delay* dekapsulasi bergantung pada *Coding Scheme* dan jumlah *timeslot* yang digunakan.

$$t_D (2.75G UE) = \frac{\text{jumlah } h \text{ frame IP} \times \text{jumlah } h \text{ frame SNDCP} \times N_{frame RLC block} \times \frac{RLC \text{ data}}{\text{code rate}}}{n \times V_{TS}} \quad (\text{s}) \quad (2.19)$$

dengan:

- | | |
|----------------------------|---|
| $t_D (2.75G UE)$ | = delay dekapsulasi di UE (s) |
| <i>jumlah frame IP</i> | = jumlah segmen <i>frame IP</i> |
| <i>jumlah frame SNDCP</i> | = jumlah segmen <i>frame SNDCP</i> |
| $N_{frame RLC block}$ | = jumlah RLC/MAC <i>radio block</i> |
| <i>RLC data/ code rate</i> | = segmen RLC/MAC <i>radio block</i> yang dikirim (bit) |
| n | = jumlah <i>time slot</i> yang digunakan |
| V_{TS} | = kecepatan data masing-masing <i>coding scheme</i> (bps) |

f) 3.5G SGSN (3.5G *Serving GPRS Support Node*)

Pada SGSN terjadi dekapsulasi terhadap data yang diterima dari GGSN. Paket data tersebut didekapsulasi dengan persamaan:

$$W_{mobile TV(3.5G SGSN)} = W_{frame GGSN} - \text{header}_{GTP} - \text{header}_{IP} - \text{header}_{UDP} - \text{header}_{Ethernet} - FCS \quad (\text{byte}) \quad (2.20)$$

dengan:

- | | |
|----------------------------|--|
| $W_{mobile TV(3.5G SGSN)}$ | = panjang data aplikasi mobile TV 3.5G SGSN (byte) |
| $W_{frame GGSN}$ | = panjang <i>frame</i> di GGSN (byte) |
| header_{GTP} | = <i>header</i> GTP (8 byte) |
| header_{IP} | = <i>header</i> IP (40 byte) |
| header_{UDP} | = <i>header</i> UDP (8 byte) |
| $\text{header}_{Ethernet}$ | = <i>header</i> ethernet (14 byte) |



FCS = panjang *Frame Check Sequence* (4 byte)

Pada skripsi ini menggunakan 3.5G SGSN dengan teknologi *Fast Ethernet* dengan transmisi data sebesar 100 Mbps. Nilai *delay* dekapsulasi yang terjadi di 3.5G SGSN adalah:

$$t_{D(3.5G\ SGSN)} = \frac{W_{mobile\ TV\ (3.5G\ SGSN)} \times 8}{V_{3.5G\ SGSN}} \quad (s) \quad (2.21)$$

dengan:

$t_{D(3.5G\ SGSN)}$ = *delay* dekapsulasi di 3.5G SGSN (s)

$W_{mobile\ TV\ (3.5G\ SGSN)}$ = panjang data aplikasi *mobile TV* di 3.5G SGSN (byte)

$V_{3.5G\ SGSN}$ = kecepatan transmisi data di 3.5G SGSN (bps)

Selanjutnya paket data aplikasi *mobile TV* ditambahkan *header GTP*, *UDP* dan *IP*. Saat memasuki lapisan AAL5, paket data tersebut dibentuk menjadi CPCS PDU (*Common Part Convergence Sublayer Protocol Data Unit*) yang mencakup sebuah *trailer* dengan bidang-bidang berikut (William Stallings, 2000: 381):

- *Padding*, berfungsi untuk penjajaran CPCS PDU menjadi 48 byte (0-47 byte)
- CPCS *User-to-user indication* (1 byte)
- *Common Part Indicator* (1 byte)
- Panjang bidang *payload* PDU CPCS (2 byte)
- *Cyclic Redundancy Check* (4 byte)

Sehingga panjang *frame* menjadi:

$$W_{frame\ AAL5} = W_{mobile\ TV(3.5G\ SGSN)} + header_{GTP} + header_{IP} + header_{UDP} + header_{AAL5} \quad (byte) \quad (2.22)$$

dengan:

$W_{frame\ AAL5}$ = panjang *frame* AAL5 (byte)

$W_{mobile\ TV(3.5G\ SGSN)}$ = panjang data aplikasi *mobile TV* 3.5G SGSN (byte)

$header_{GTP}$ = *header* GTP (8 byte)

$header_{IP}$ = *header* IP (40 byte)

$header_{UDP}$ = *header* UDP (8 byte)

$header_{AAL5}$ = *header* AAL5 (8 byte)

Pada *layer ATM*, *frame* AAL5 tersebut kemudian akan dipecah menjadi blok-blok *payload* PDU SAR (*Segmentation and Reassembly Sublayer*) sebesar 48 byte. Masing-masing PDU SAR 48 byte sesuai dengan sel ATM tunggal (Stalling, 2001:363). Sehingga banyaknya sel ATM yang terbentuk adalah:



$$N_{frame \text{ ATM}} = \frac{W_{frame \text{ AAL5}}}{48 \text{ byte}} \quad (2.23)$$

dengan:

$N_{frame \text{ ATM}}$ = jumlah frame ATM

$W_{frame \text{ AAL5}}$ = panjang total frame AAL5 (byte)

Selanjutnya setiap sel ATM diberi header ATM sebesar 5 byte sehingga panjang frame ATM menjadi 53 byte. Panjang frame di SGSN yang siap ditransmisikan menuju RNC merupakan panjang frame ATM total, yaitu:

$$W_{frame \text{ 3.5GSGSN}} = W_{frame \text{ ATM total}} = N_{frame \text{ ATM}} \times W_{frame \text{ ATM}} \text{ (byte)} \quad (2.24)$$

dengan:

$W_{frame \text{ 3.5GSGSN}}$ = panjang frame 3.5G SGSN (byte)

$W_{frame \text{ ATM total}}$ = panjang frame ATM total (byte)

$N_{frame \text{ ATM}}$ = jumlah frame ATM

$W_{frame \text{ ATM}}$ = panjang frame ATM (53 byte)

Sehingga delay enkapsulasi yang terjadi di 3.5G SGSN dapat dihitung dengan persamaan:

$$t_{E(3.5G \text{ SGSN})} = \frac{W_{frame \text{ 3.5GSGSN}} \times 8}{V_{SGSN-RNS}} \quad (\text{s}) \quad (2.25)$$

$t_{E(3.5G \text{ SGSN})}$ = delay enkapsulasi di 3.5G SGSN (s)

$W_{frame \text{ 3.5GSGSN}}$ = panjang frame 3.5G SGSN (byte)

$V_{SGSN-RNS}$ = kecepatan transmisi data di 3.5G SGSN (bps)

g) RNC (*Radio Network Controller*)

Pada RNC paket yang diterima dari 3.5G SGSN akan mengalami proses dekapsulasi. RNC menggunakan interface STM-1 dengan kecepatan transmisi 155,52 Mbps. Frame ATM total didekapsulasi untuk mendapatkan data aplikasi sesuai dengan persamaan berikut:

$$W_{mobile \text{ TV (RNC)}} = W_{frame \text{ 3.5GSGSN}} - header_{ATM} - header_{GTP} - header_{IP} - header_{UDP} - header_{AAL5} \text{ (byte)} \quad (2.26)$$

dengan:

$W_{mobile \text{ TV (RNC)}}$ = panjang data aplikasi mobile TV di RNC (byte)

$W_{frame \text{ 3.5GSGSN}}$ = panjang frame 3.5G SGSN (byte)

$header_{ATM}$ = panjang header ATM (5 byte)

$header_{GTP}$ = header GTP (8 byte)



$header_{IP}$	= header IP (40 byte)
$header_{UDP}$	= header UDP (8 byte)
$header_{AAL5}$	= header AAL5 (8 byte)

Besar *delay* dekapsulasi yang terjadi di RNC didapatkan dengan persamaan berikut:

$$t_{D(RNC)} = \frac{W_{mobile TV(RNC)} \times 8}{V_{RNC}} \quad (\text{s}) \quad (2.27)$$

dengan:

$t_{D(RNC)}$	= <i>delay</i> dekapsulasi di RNC (s)
$W_{mobile TV(RNC)}$	= panjang data aplikasi <i>mobile TV</i> di RNC (byte)
V_{RNC}	= kecepatan transmisi data di RNC (bps)

Selanjutnya paket data aplikasi tersebut dienkapsulasi pada *layer PDCP* (*Packet Data Convergence Protocol*) dengan diberi *header PDCP* sebesar 1 byte.

$$W_{frame PDCP} = W_{mobile TV(RNC)} + header_{PDCP} \quad (\text{byte}) \quad (2.28)$$

dengan:

$W_{frame PDCP}$	= panjang <i>frame PDCP</i>
$W_{mobile TV(RNC)}$	= panjang data aplikasi <i>mobile TV</i> di RNC (byte)
$header_{PDCP}$	= panjang <i>frame PDCP</i> (1 byte)

Pada *layer RLC*, *frame PDCP* akan disegmentasi menjadi RLC PDU *fixed size* yaitu sebesar 40 byte (Anthony Lo, 2004: 8). Sehingga jumlah *frame RLC* sesuai dengan persamaan berikut:

$$N_{frame RLC} = \frac{W_{frame PDCP}}{40 \text{ byte}} \quad (29)$$

dengan:

$N_{frame RLC}$	= jumlah <i>frame LLC</i>
$W_{frame PDCP}$	= panjang <i>frame PDCP</i> (byte)

Selanjutnya setiap *frame RLC* ditambahkan dengan *header RLC* sebesar 2 byte sehingga panjang *frame RLC* menjadi 42 byte. Sehingga panjang *frame RLC* total yang siap diteruskan ke *layer MAC-d* menjadi:

$$W_{frame RLC total} = N_{frame RLC} \times W_{frame RLC} \quad (\text{byte}) \quad (2.30)$$

dengan:

$W_{frame RLC total}$	= panjang <i>frame RLC</i> total (byte)
$N_{frame RLC}$	= jumlah <i>frame LLC</i>
$W_{frame RLC}$	= panjang <i>frame RLC</i> (42 byte)



Pada *layer MAC-d*, RLC PDU akan disegmentasi menjadi MAC-d SDU (*Service Data Unit*). Besarnya MAC-d SDU tergantung pada besarnya RLC PDU yang telah ditetapkan yaitu sebesar 42 byte. Jumlah *frame* MAC-d SDU sesuai dengan persamaan berikut:

$$N_{\text{frame MAC-d}} = \frac{W_{\text{frame RLC total}}}{42 \text{ byte}} \quad (2.31)$$

dengan:

$N_{\text{frame MAC-d}}$ = jumlah *frame* MAC-d

$W_{\text{frame RLC total}}$ = panjang *frame* RLC total (byte)

Sedangkan panjang *frame* MAC-d dapat dihitung berdasarkan persamaan:

$$W_{\text{frame MAC-d}} = \text{header}_{\text{MAC}} + 42 \text{ byte} \quad (\text{byte}) \quad (2.32)$$

dengan:

$W_{\text{frame MAC-d}}$ = panjang *frame* MAC-d (byte)

$\text{header}_{\text{MAC}}$ = header MAC (3 byte)

Pada HS-DSCH FP (*Frame Protocol*), *frame* MAC-d dienkapsulasi dengan menambahkan *header* FP dan CRC sesuai persamaan berikut:

$$W_{\text{frame FP}} = W_{\text{frame MAC-d}} + \text{header}_{\text{FP}} + \text{CRC} \quad (\text{byte}) \quad (2.33)$$

dengan:

$W_{\text{frame FP}}$ = panjang *frame* pada HS-DSCH FP (byte)

$W_{\text{frame MAC-d}}$ = panjang *frame* MAC-d (byte)

$\text{header}_{\text{FP}}$ = header HS-DSCH FP (7 byte)

CRC = Cyclic Redundancy Check (2 byte)

Sehingga panjang *frame* HS-DSCH FP total adalah sesuai dengan persamaan berikut:

$$W_{\text{frame FP total}} = N_{\text{frame MAC-d}} \times W_{\text{frame FP}} \quad (\text{byte}) \quad (2.34)$$

dengan:

$W_{\text{frame FP total}}$ = panjang *frame* HS-DSCH FP total (byte)

$N_{\text{frame MAC-d}}$ = jumlah *frame* MAC-d

$W_{\text{frame FP}}$ = panjang *frame* pada HS-DSCH FP (byte)

Saat memasuki *layer AAL2*, paket data dibentuk menjadi paket CPS (*Common Part Sublayer*) dengan penambahan *header* sebesar 3 byte yang terdiri atas:

- *Channel Identification* (8 bit)
- *Length Indicator* (6 bit)
- *User-to-user Indication* (5 bit)



- *Header Error Control* (5 bit)

Sehingga panjang *frame* menjadi:

$$W_{frame\ AAL2} = W_{frame\ FP\ total} + header_{AAL2} \quad (\text{byte}) \quad (2.35)$$

dengan:

$$W_{frame\ AAL2} = \text{panjang frame AAL2 (byte)}$$

$$W_{frame\ FP\ total} = \text{panjang frame HS-DSCH FP total (byte)}$$

$$header_{AAL2} = \text{header AAL2 (3 byte)}$$

Pada *layer ATM*, *frame AAL2* tersebut kemudian akan dipecah menjadi blok-blok *payload PDU SAR (Segmentation and Reassembly Sublayer)* sebesar 48 byte. Masing-masing *PDU SAR* 48 byte sesuai dengan sel *ATM* tunggal. Sehingga banyaknya sel *ATM* yang terbentuk adalah:

$$N_{frame\ ATM2} = \frac{W_{frame\ AAL2}}{48\ \text{byte}} \quad (2.36)$$

dengan:

$$N_{frame\ ATM2} = \text{jumlah frame ATM pada RNC}$$

$$W_{frame\ AAL2} = \text{panjang frame AAL2 (byte)}$$

Selanjutnya setiap sel *ATM* ditambahkan dengan *header ATM* sebesar 5 byte, sehingga panjang *frame ATM* menjadi 53 byte. Panjang *frame* di RNC yang siap ditransmisikan menuju UE merupakan panjang *frame ATM* total, yaitu:

$$W_{frame\ RNC} = W_{frame\ ATM2\ total} = N_{frame\ ATM2} \times W_{frame\ ATM} \quad (\text{byte}) \quad (2.37)$$

dengan:

$$W_{frame\ RNC} = \text{panjang frame RNC (byte)}$$

$$W_{frame\ ATM2\ total} = \text{panjang frame ATM total pada RNC (byte)}$$

$$N_{frame\ ATM2} = \text{jumlah frame ATM pada RNC}$$

$$W_{frame\ ATM} = \text{panjang frame ATM (53 byte)}$$

Sehingga *delay enkapsulasi* yang terjadi di RNC adalah:

$$t_{E(RNC)} = \frac{W_{frame\ RNC} \times 8}{V_{RNC}} \quad (\text{s}) \quad (2.38)$$

dengan:

$$t_{E(RNC)} = \text{delay enkapsulasi di RNC (s)}$$

$$W_{frame\ RNC} = \text{panjang frame RNC (byte)}$$

$$V_{RNC} = \text{kecpatan transmisi data di RNC (bps)}$$

h) Node B

Pada Node B paket data yang diterima dari RNC didekapsulasi sesuai dengan persamaan berikut:

$$W_{mobile\ TV(Node\ B)} = W_{frame\ RNC} - header_{ATM} - header_{AAL2} \\ - header_{FP} - CRC \quad (2.39)$$

dengan:

$W_{mobile\ TV(Node\ B)}$ = panjang data aplikasi mobile TV di Node B (byte)

$W_{frame\ RNC}$ = panjang frame RNC (byte)

$header_{ATM}$ = header ATM (5 byte)

$header_{AAL2}$ = header AAL2 (3 byte)

$header_{FP}$ = header HS-DSCH FP (7 byte)

CRC = Cyclic Redundancy Check (2 byte)

Delay dekapsulasi yang terjadi di Node B adalah:

$$t_{D(Node\ B)} = \frac{W_{mobile\ TV(Node\ B)} \times 8}{V_{Node\ B}} \quad (s) \quad (2.40)$$

dengan:

$t_{D(Node\ B)}$ = delay dekapsulasi di Node B (s)

$W_{mobile\ TV(Node\ B)}$ = panjang data aplikasi mobile TV di Node B (byte)

$V_{Node\ B}$ = kecepatan transmisi data di Node B (bps)

Pada saat memasuki layer MAC-hs, paket data disegmentasi menjadi paket sebesar 42 byte. Jumlah frame MAC-hs sesuai dengan persamaan berikut:

$$N_{frame\ MAC-hs} = \frac{W_{mobile\ TV(Node\ B)}}{42\ byte} \quad (2.41)$$

dengan:

$N_{frame\ MAC-hs}$ = jumlah frame MAC-hs

$W_{mobile\ TV(Node\ B)}$ = panjang data aplikasi mobile TV di Node B (byte)

Panjang frame MAC-hs dapat dihitung berdasarkan persamaan berikut:

$$W_{frame\ MAC-hs} = header_{MAC} + 42byte \quad (\text{byte}) \quad (2.42)$$

dengan:

$W_{frame\ MAC-hs}$ = panjang frame MAC-hs (byte)

$header_{MAC}$ = header MAC (3 byte)

Sehingga panjang frame total pada Node B didapat dengan persamaan berikut:

$$W_{frame\ Node\ B} = N_{frame\ MAC-hs} \times W_{frame\ MAC-hs} \quad (\text{byte}) \quad (2.43)$$

dengan:

- $W_{frame \ Node \ B}$ = panjang frame Node B total (byte)
 $N_{frame \ MAC-hs}$ = jumlah frame MAC-hs
 $W_{frame \ MAC-hs}$ = panjang frame MAC-hs (byte)

Delay enkapsulasi yang terjadi di Node B adalah:

$$t_{E(Node \ B)} = \frac{W_{frame \ Node \ B} \times 8}{V_{Node \ B}} \quad (\text{s}) \quad (2.44)$$

dengan:

- $t_{E(Node \ B)}$ = delay enkapsulasi pada Node B (s)
 $W_{frame \ Node \ B}$ = panjang frame Node B total (byte)
 $V_{Node \ B}$ = kecepatan transmisi data di Node B (bps)

i) UE (*User Equipment*) pada HSDPA

Pada UE, paket data dari Node B didekapsulasi untuk mendapatkan data aplikasi *mobile TV*. Proses dekapsulasi tersebut sesuai dengan persamaan berikut:

$$W_{mobile \ TV(3.5G \ UE)} = W_{frame \ NodeB} - header_{PDCP} - header_{RLC} - header_{MAC} \quad (\text{byte}) \quad (2.45)$$

dengan:

- $W_{mobile \ TV(3.5G \ UE)}$ = panjang data aplikasi *mobile TV* di UE (byte)
 $W_{frame \ NodeB}$ = panjang frame di Node B (byte)
 $header_{PDCP}$ = header PDCP (1 byte)
 $header_{RLC}$ = header RLC (2 byte)
 $header_{MAC}$ = header MAC (3 byte)

Nilai *delay* dekapsulasi pada UE dapat dihitung dengan persamaan berikut:

$$t_{D(3.5G \ UE)} = \frac{W_{mobile \ TV(3.5G \ UE)} \times 8}{V_{UE}} \quad (\text{s}) \quad (2.46)$$

dengan:

- $t_{D(3.5G \ UE)}$ = delay dekapsulasi di UE (s)
 $W_{mobile \ TV(3.5G \ UE)}$ = panjang data aplikasi *mobile TV* di UE (byte)
 V_{UE} = kecepatan transmisi data di UE (bps)

j) Delay Enkapsulasi dan Dekapsulasi pada Jaringan EDGE

$$t_{E(EDGE)total} = t_{E(PDN \ Internet)} + t_{E(GGSN)} + t_{E(2.75G \ SGSN)} + t_{E(PCU)} \quad (\text{s}) \quad (2.47)$$

$$t_{D(EDGE)total} = t_{D(GGSN)} + t_{D(2.75G \ SGSN)} + t_{D(PCU)} + t_{D(2.75G \ UE)} \quad (\text{s}) \quad (2.48)$$

$$t_{proc \ EDGE} = t_{E(EDGE)total} + t_{D(EDGE)total} \quad (\text{s}) \quad (2.49)$$



dengan:

$t_{proc\ EDGE}$	= delay proses pada jaringan EDGE (s)
$t_{E(EDGE)total}$	= delay enkapsulasi total pada jaringan EDGE (s)
$t_{D(EDGE)total}$	= delay dekapsulasi total pada jaringan EDGE (s)
$t_{E(PDN\ Internet)}$	= delay enkapsulasi di PDN Internet (s)
$t_{E(GGSN)}$	= delay enkapsulasi di GGSN (s)
$t_{E(2.75G\ SGSN)}$	= delay enkapsulasi di 2.75G SGSN (s)
$t_{E(PCU)}$	= delay enkapsulasi di PCU (s)
$t_{D(GGSN)}$	= delay dekapsulasi di GGSN (s)
$t_{D(2.75G\ SGSN)}$	= delay dekapsulasi di 2.75G SGSN (s)
$t_{D(PCU)}$	= delay dekapsulasi di PCU (s)
$t_{D(2.75G\ UE)}$	= delay dekapsulasi di 2.75G UE (s)

k) Delay Enkapsulasi dan Dekapsulasi pada Jaringan HSDPA

$$t_{E(HSDPA)total} = t_{E(PDN\ Internet)} + t_{E(GGSN)} + t_{E(3.55G\ SGSN)} + t_{E(RNC)} + t_{E(Node\ B)} \quad (2.50)$$

$$t_{D(HSDPA)total} = t_{D(GGSN)} + t_{D(3.5G\ SGSN)} + t_{D(RNC)} + t_{D(Node\ B)} + t_{D(3.5G\ UE)} \quad (2.51)$$

$$t_{proc\ HSDPA} = t_{E(HSDPA)total} + t_{D(HSDPA)total} \quad (2.52)$$

dengan:

$t_{proc\ HSDPA}$	= delay proses pada jaringan HSDPA (s)
$t_{E(HSDPA)total}$	= delay enkapsulasi total pada jaringan HSDPA (s)
$t_{D(HSDPA)total}$	= delay dekapsulasi total pada jaringan HSDPA (s)
$t_{E(PDN\ Internet)}$	= delay enkapsulasi pada PDN Internet (s)
$t_{E(GGSN)}$	= delay enkapsulasi pada GGSN (s)
$t_{E(3.55G\ SGSN)}$	= delay enkapsulasi pada 3.5G SGSN (s)
$t_{E(RNC)}$	= delay enkapsulasi pada RNC (s)
$t_{E(Node\ B)}$	= delay enkapsulasi pada Node B (s)
$t_{D(GGSN)}$	= delay dekapsulasi pada GGSN (s)
$t_{D(3.5G\ SGSN)}$	= delay dekapsulasi pada 3.5G SGSN (s)
$t_{D(RNC)}$	= delay dekapsulasi pada RNC (s)



$$\begin{aligned} t_{D(Node\ B)} &= delay\ dekapsulasi\ pada\ Node\ B\ (s) \\ t_{D(3.5G\ UE)} &= delay\ dekapsulasi\ pada\ 3.5G\ UE\ (s) \end{aligned}$$

2.5.1.3 Delay Transmisi

Delay transmisi adalah waktu yang dibutuhkan untuk mentransmisikan bit pertama dari paket data hingga terakhir melalui media transmisi tertentu. Secara umum, *delay* transmisi didefinisikan sebagai: (Mischa Schwartz, 1987 : 132)

$$t_{T(n)} = \frac{l+l'}{c_{link}} \quad (2.53)$$

Dengan:

$$t_{T(n)} = Delay\ transmisi\ (s)$$

$$l = Panjang\ data\ (bit)$$

$$l' = Jumlah\ header\ (bit)$$

$$c_{link} = Kecepatan\ transmisi\ pada\ masing-masing\ saluran\ transmisi\ yang\ menghubungkan\ setiap\ node\ (bps)$$

Proses ini akan terjadi pada semua *interface* yang dilalui data dalam arsitektur jaringan, yaitu PDN *interface*, Gi *interface*, Gn *interface*, Gb *interface*, Abis *interface*, Um *interface*, Iu *interface*, Iub *interface*, dan Uu *interface*.

a) PDN Internet – GGSN (Gi Interface)

Jika jaringan yang menghubungkan PDN *Internet* dengan GGSN diasumsikan sebuah jaringan standar *Ethernet* yang memiliki kecepatan transmisi data 1 Gbps, maka *delay* transmisi yang terjadi pada Gi *interface* adalah sesuai dengan persamaan berikut:

$$t_{T(Gi\ Interface)} = \frac{W_{framePDN_Internet} \times 8}{V_{Ethernet}} \quad (s) \quad (2.54)$$

dengan:

$$t_{T(Gi\ Interface)} = delay\ transmisi\ di\ Gi\ Interface\ (s)$$

$$W_{framePDN_Internet} = panjang\ frame\ PDN\ Internet\ (byte)$$

$$V_{Ethernet} = kecepatan\ transmisi\ pada\ Gi\ Interface\ (bps)$$

b) GGSN – 2.75G SGSN (Gn Interface)

Jika hubungan antara GGSN dan 2.75G SGSN menggunakan *Ethernet* standard 100Base-Tx dengan kecepatan transmisi data sebesar 100 Mbps, maka *delay* transmisi yang terjadi pada Gn *interface* adalah sesuai persamaan berikut :

$$t_{T(Gn\ Interface)} = \frac{W_{frameGGSN} \times 8}{V_{Gn\ Interface}} \quad (s) \quad (2.55)$$



dengan:

$t_{T(Gn\ Interface)}$ = delay transmisi di Gn Interface (s)

$W_{frameGGSN}$ = panjang frame GGSN (byte)

$V_{Gn\ Interface}$ = kecepatan transmisi pada Gn Interface (bps)

c) 2.75G SGSN – BSC (Gb Interface)

Hubungan antara 2.5G SGSN dan BSC menggunakan standar E1-Link dengan kecepatan 2,048 Mbps, sehingga delay transmisi di Gb Interface adalah sesuai persamaan berikut:

$$t_{T(Gb\ Interface)} = \frac{W_{frame\ 2.75G\ SGSN} \times 8}{V_{Gb\ Interface}} \quad (s) \quad (2.56)$$

dengan:

$t_{T(Gb\ Interface)}$ = delay transmisi di Gb Interface (s)

$W_{frame\ 2.75G\ SGSN}$ = panjang frame 2.75G SGSN (byte)

$V_{Gb\ Interface}$ = kecepatan transmisi pada Gb Interface (bps)

d) BSC – BTS (Abis Interface)

Pada hubungan antara BSC dan BTS menggunakan standar 150 GSL dengan kecepatan 16 kbps. Delay transmisi pada Abis Interface adalah sebagai berikut (Uke Kurniawan, 2009:45):

$$t_{T(Abis\ Interface)} = \frac{N_{frame\ RLC\ block} \times RLC_{radio\ block}}{V_{Abis\ Interface}} \quad (s) \quad (2.57)$$

dengan:

$t_{T(Abis\ Interface)}$ = delay transmisi pada Abis Interface (s)

$N_{frame\ RLC\ block}$ = jumlah blok data masing-masing coding scheme

$RLC_{radio\ block}$ = panjang paket data RLC

$V_{Abis\ Interface}$ = kecepatan transmisi pada Abis Interface (bps)

e) BTS – UE (Um Interface)

Hubungan antara BTS dan UE menggunakan CS-1 dengan bit rate 9,05 kbps serta menggunakan 4 timeslot. Sehingga delay transmisi pada Um Interface didapat dengan persamaan :

$$t_{T(Um\ Interface)} = \frac{N_{frame\ RLC\ block} \times RLC_{radio\ block}}{n \times V_{MCS}} \quad (s) \quad (2.58)$$



dengan:

- | | |
|-------------------------|--|
| $t_{T(Um\ Interface)}$ | = delay transmisi pada Um Interface (s) |
| $N_{frame\ RLC\ block}$ | = jumlah blok data masing-masing coding scheme |
| $RLC_{radio\ block}$ | = panjang paket data RLC |
| n | = jumlah slot pada tiap UE |
| V_{MCS} | = kecepatan bit rate pada tiap coding scheme (bps) |

f) GGSN – 3.5G SGSN (Gn Interface)

GGSN dengan 3.5G SGSN dihubungkan dengan menggunakan *Fast Ethernet* yang memiliki kecepatan transmisi data sebesar 100 Mbps. Sehingga *delay* transmisi yang terjadi pada Gn interface adalah sesuai dengan persamaan berikut:

$$t_{T(Gn\ Interface)} = \frac{W_{frame\ GGSN} \times 8}{V_{Gn\ Interface}} \quad (2.59)$$

dengan:

- | | |
|------------------------|---|
| $t_{T(Gn\ Interface)}$ | = delay transmisi di Gn Interface (s) |
| $W_{frame\ GGSN}$ | = panjang frame GGSN (byte) |
| $V_{Gn\ Interface}$ | = kecepatan transmisi pada Gn Interface (bps) |

g) 3.5G SGSN – RNC (Iu Interface)

Iu interface yang menghubungkan antara SGSN dan RNC menggunakan sistem transmisi digital STM-1 yang memberikan kecepatan data sebesar 155,52 Mbps. Sehingga nilai *delay* transmisi didapatkan dengan persamaan:

$$t_{T(Iu\ Interface)} = \frac{W_{frame\ 3.5G\ SGSN} \times 8}{V_{Iu\ Interface}} \quad (2.60)$$

dengan:

- | | |
|-------------------------|---|
| $t_{T(Iu\ Interface)}$ | = delay transmisi di Iu Interface (s) |
| $W_{frame\ 3.5G\ SGSN}$ | = panjang frame 3.5G SGSN (byte) |
| $V_{Iu\ Interface}$ | = kecepatan transmisi pada Iu Interface (bps) |

h) RNC – Node B (Iub Interface)

Hubungan antara RNC dan Node B menggunakan interface sistem transmisi digital STM-1 yang memberikan kecepatan data sebesar 155,52 Mbps. Sehingga nilai *delay* transmisi yang terjadi sesuai dengan persamaan berikut:

$$t_{T(Iub\ Interface)} = \frac{W_{frame\ RNC} \times 8}{V_{Iub\ Interface}} \quad (2.61)$$

dengan:



- $t_{T(Iub\ Interface)}$ = delay transmisi di Iub Interface (s)
 $W_{frame\ RNC}$ = panjang frame RNC (byte)
 $V_{Iu\ Interface}$ = kecepatan transmisi pada Iub Interface (bps)

i) Node B – UE (Uu Interface)

Antara Node B dan UE, *delay* transmisi yang terjadi bergantung pada *data rate* UE serta jumlah slot. Sehingga nilai *delay* transmisi yang terjadi sesuai dengan persamaan:

$$t_{T(Uu\ Interface)} = \frac{W_{frame\ Node\ B} \times 8}{n \times V_{UE}} \quad (2.62)$$

dengan:

- $t_{T(Uu\ Interface)}$ = delay transmisi di Uu Interface (s)
 $W_{frame\ Node\ B}$ = panjang frame Node B (byte)
 n = jumlah slot pada UE
 V_{UE} = kecepatan transmisi pada UE (bps)

Jumlah slot ditentukan oleh *chip rate* HSDPA yaitu sebesar 3,84 Mbps. *Chip rate* sebesar 3,84 Mcps selanjutnya dibagi menjadi radio *frame* 2 ms. Satu slot terdiri dari 2560 *chip*, sehingga banyaknya slot adalah:

$$\begin{aligned} n &= \frac{(cr \times rf)}{2560 \text{ chip}} \\ &= \frac{3,84 \cdot 10^6 \text{ cps} \times 2 \cdot 10^{-3} \text{ s}}{2560 \text{ chip}} = \frac{7680 \text{ chip}}{2560 \text{ chip}} = 3 \text{ slot} \end{aligned} \quad (2.63)$$

dengan:

- cr = *chip rate* (cps)
 rf = *radio frame* (s)

j) Delay Transmisi Total pada Jaringan EDGE

Delay transmisi total yang terjadi pada jaringan EDGE ditentukan berdasarkan persamaan:

$$\begin{aligned} t_{T(EDGE)total} &= t_{T(Gi\ Interface)} + t_{T(Gn\ Interface)} + t_{T(Gb\ Interface)} \\ &\quad + t_{T(Abis\ Interface)} + t_{T(Um\ Interface)} \end{aligned} \quad (2.64)$$

dengan:

- $t_{T(EDGE)total}$ = *delay* transmisi total pada jaringan EDGE (s)
 $t_{T(Gi\ Interface)}$ = *delay* transmisi pada Gi Interface (s)



$t_{T(Gn\ Interface)}$	= delay transmisi pada Gn Interface (s)
$t_{T(Gb\ Interface)}$	= delay transmisi pada Gb Interface (s)
$t_{T(Abis\ Interface)}$	= delay transmisi pada Abis Interface (s)
$t_{T(Um\ Interface)}$	= delay transmisi pada Um Interface (s)

k) Delay Transmisi Total pada Jaringan HSDPA

Delay transmisi total yang terjadi pada jaringan HSDPA ditentukan berdasarkan persamaan:

$$\begin{aligned} t_{T(HSDPA)total} &= t_{T(Gi\ Interface)} + t_{T(Gn\ Interface)} + t_{T(Iu\ Interface)} \\ &\quad + t_{T(Iub\ Interface)} + t_{T(Uu\ Interface)} \text{ (s)} \end{aligned} \quad (2.65)$$

dengan:

$t_{T(HSDPA)total}$	= delay transmisi total pada jaringan HSDPA (s)
$t_{T(Gi\ Interface)}$	= delay transmisi pada Gi Interface (s)
$t_{T(Gn\ Interface)}$	= delay transmisi pada Gn Interface (s)
$t_{T(Iu\ Interface)}$	= delay transmisi pada Iu Interface (s)
$t_{T(Iub\ Interface)}$	= delay transmisi pada Iub Interface (s)
$t_{T(Uu\ Interface)}$	= delay transmisi pada Uu Interface (s)

2.5.1.4 Delay Propagasi

Delay propagasi merupakan waktu perambatan yang dibutuhkan data dari satu *node* ke *node* yang lain melalui suatu media transmisi. Delay propagasi dipengaruhi oleh jarak antar *node* dan media transmisi yang digunakan (kabel coaxial, twisted pair, serat optik, dan sebagainya). Dalam hal ini pada jaringan EDGE, delay propagasi terjadi pada hubungan PDN Internet GGSN-2.5G SGSN-BSC-BTS-UE. Sedangkan pada jaringan WCDMA delay propagasi terjadi pada PDN Internet-GGSN-3G SGSN-RNC-*Node* B-UE.

Pada jaringan EDGE *node* lintasan antara PDN Internet dan BTS menggunakan media *fiber optic*. Sedangkan pada koneksi antara BTS-UE, media transmisi yang digunakan adalah udara dengan cepat rambat gelombang elektromagnetik sebesar 3×10^8 m/s. Untuk jaringan HSDPA *node* lintasan antara PDN Internet dan *Node* B menggunakan media *fiber optic* pula, dengan koneksi *Node* B-UE adalah media udara. Sedangkan referensi jarak antar *node* pada jaringan HSDPA adalah berdasarkan Tabel 2.5.

Tabel 2.5 Jarak tiap *node* pada HSDPA

Node	Jarak (m)
GGSN – SGSN	2000
SGSN – RNC	3000
RNC – Node B	400

Sumber: Linda Ekowati, 2008:68

Besar *delay* propagasi dirumuskan: (Forouzan, Behrouz A, 2000 : 215)

$$t_p = \frac{N_{frame} \times R}{v} \quad (\text{s}) \quad (2.66)$$

Dimana:

- N_{frame} = jumlah *payload frame* yang ditransmisikan
- t_p = *delay* propagasi (s)
- R = jarak pentransmisian (m)
- v = cepat rambat dalam medium (m/s)

2.5.1.5 Delay Antrian

Delay antrian adalah waktu dimana paket data tersebut berada dalam antrian untuk ditransmisikan. *Delay* antrian yang terjadi sangat dinamis sehingga jika antrian kosong dan tidak ada paket data lain maka *delay* antrian tidak terjadi. Model sistem antrian yang digunakan dalam bahasan skripsi ini adalah M/M/1 yang memiliki karakteristik :

- Distribusi waktu antara kedatangan bersifat *Poisson*.
- Distribusi pelayanan bersifat deterministik dengan waktu pelayanan tiap paket sama dan panjang paket konstan.
- Jumlah server 1.
- Disiplin antrian adalah FIFO (*First In First Out*) dan tidak ada paket yang meninggalkan antrian.

Besar *delay* antrian adalah sebagai berikut (I Made Wiryana, 1992:2):

$$t_w = t_{queue} + t_{serve} \quad (\text{s}) \quad (2.67)$$

Dengan:

- t_w = *delay* antrian (s)
 - t_{queue} = waktu tunggu paket (s)
 - t_{serve} = waktu rata-rata pelayanan (s)
- Waktu pelayanan *router* (t_{serv}) dihitung dengan menggunakan persamaan: (Mischa Schwartz, 1987 : 23)



$$t_{serve} = \frac{1}{\mu} \quad (\text{s}) \quad (2.68)$$

Dengan μ adalah kecepatan pelayanan. Nilai kecepatan pelayanan ini diperoleh dari persamaan: (Mischa Schwartz, 1987 : 23)

$$\mu = \frac{C}{m} \quad (\text{paket/s}) \quad (2.69)$$

Dengan:

μ = kecepatan pelayanan dalam *node* (paket/s)

C = kapasitas *link* pada *node* (bps)

m = panjang paket data yang ditransmisikan di *node* (bit)

Waktu tunggu rata-rata (t_{queue}) diperoleh dari (Schwartz, 1987 : 58) :

$$t_{queue} = \frac{\lambda E(\tau^2)}{2(1-\rho)} \quad (\text{s}) \quad (2.70)$$

Parameter ρ merupakan faktor utilitas *link* yang menunjukkan faktor kegunaan *link* dalam sistem. Faktor utilitas mempunyai nilai 0 sampai dibawah 1 ($0 < \rho < 1$). Makin tinggi nilai faktor utilitas menunjukkan tingkat kegunaan sistem yang makin tinggi/padat, sesuai dengan rumus:

$$\lambda = \mu \times \rho \quad (\text{bps}) \quad (2.71)$$

Dimana:

λ = rata-rata laju kedatangan paket (bps)

ρ = faktor utilitas

Sedangkan $E(\tau^2)$ merupakan distribusi waktu pelayanan yang dinyatakan dengan:

$$E(\tau^2) = \sigma^2 + \frac{1}{\mu^2} \quad (\text{s}) \quad (2.72)$$

Total *delay* antrian yang terjadi adalah:

$$t_w = \frac{\lambda}{2\mu(\mu-\lambda)} + \frac{1}{\mu} \quad (\text{s}) \quad (2.73)$$

Pada analisis skripsi ini *delay* antrian di jaringan GPRS terjadi di GGSN dan 2.5G SGSN. Sedangkan untuk jaringan WCDMA *delay* antrian terjadi pada GGSN dan 3G SGSN. Maka total *delay* antrian merupakan penjumlahan dari masing-masing *delay* antrian tersebut.

Delay antrian total pada jaringan EDGE:

$$t_{W(EDGE\ total)} = t_w(GGSN) + t_w(2.75G\ SGSN) \quad (\text{s}) \quad (2.74)$$

Delay antrian total pada jaringan HSDPA:

$$t_{W(HSDPA\ total)} = t_w(GGSN) + t_w(3.5G\ SGSN) \quad (\text{s}) \quad (2.75)$$

2.5.2 Delay Handover

Delay handover merupakan waktu tunda yang terjadi karena adanya proses *signalling* selama *User Equipment* melakukan *handover* atau periode selama proses tukar menukar informasi/*message* saat melakukan *handover*. *Delay* ini dihitung mulai dari *introduction* pertama yang dilakukan oleh *User Equipment* kepada SGSN yang baru sampai dilakukannya *Routing Area* pada *User Equipment* oleh SGSN yang baru. (Eunsoo Shim, 2002:1).

2.5.3 Throughput

Throughput merupakan parameter yang digunakan untuk mengetahui jumlah paket yang diterima dalam keadaan baik terhadap waktu total transmisi yang dibutuhkan dari sumber data ke penerima. Pengiriman data pada jaringan *packet switching* antara 2 stasiun yang melalui beberapa lapisan protokol mempunyai batas kemampuan kapasitas yang bisa dikeluarkan oleh jaringan tersebut. Pada keadaan saturasi, yaitu suatu kondisi dimana selalu tersedia *frame* yang menunggu untuk ditransmisikan, t_v merepresentasikan sebagai waktu transmisi rata-rata *frame* yang diterima dengan benar (Mischa Schwartz, 1987 : 129).

$$\lambda = \frac{\text{jumlah paket data yang diterima}}{t_v} \quad (\text{paket/s}) \quad (2.76)$$

Dengan:

λ = *throughput* (paket/s)

t_v = waktu total transmisi untuk mengirimkan paket yang benar (s)

Throughput yang didapat dengan memperhitungkan probabilitas *packet loss* (ρ) adalah (Mischa Schwartz, 1987 : 129):

$$\lambda = \frac{1}{t_v} = \frac{(1-\rho_{tot})}{t_1[1+(\alpha-1)\rho_{tot}]} \quad (\text{paket/s}) \quad (2.77)$$

Dengan:

λ = *throughput* (paket/s)

t_v = waktu transmisi rata-rata sebuah paket yang diterima dalam keadaan benar (s)

ρ_{tot} = probabilitas *packet loss* total yang terjadi dalam jaringan

t_1 = waktu transmisi sebuah paket data (s)

α = konstanta

Probabilitas *packet loss* total pada aplikasi *mobile TV* dapat dihitung dengan persamaan berikut:



$$\rho_{tot} = 1 - [(1 - \rho_{network})(1 - \rho_{mobile\ tv})] \quad (2.78)$$

Dimana probabilitas *packet loss* pada jaringan HSDPA ($\rho_{network}$) dihitung dari probabilitas *packet loss* yang terjadi pada HSDPA *air interface*, RAN (*node B*, RNC), *core network* (SGSN, GGSN) serta pada PDN *Internet* dengan persamaan:

$$\begin{aligned} \rho_{net} = 1 - & [(1 - \rho_{AI})(1 - \rho_{RNC})(1 - \rho_{Node\ B})(1 - \rho_{SGSN}) \\ & (1 - \rho_{GGSN})(1 - \rho_{PDN\ Internet})] \end{aligned} \quad (2.79)$$

Dengan:

- ρ_{net} = probabilitas *packet loss* jaringan HSDPA
- ρ_{AI} = probabilitas *packet loss* HSDPA *air interface*
- ρ_{RNC} = probabilitas *packet loss* pada RNC
- $\rho_{Node\ B}$ = probabilitas *packet loss* pada Node B
- ρ_{SGSN} = probabilitas *packet loss* pada SGSN
- ρ_{GGSN} = probabilitas *packet loss* pada GGSN
- $\rho_{PDN\ Internet}$ = probabilitas *packet loss* pada PDN *Internet*

Probabilitas *packet loss* pada *air interface*

Probabilitas *packet loss* pada *air interface* HSDPA ditentukan dengan parameter-parameter *link budget* pada Tabel 2.6.

Tabel 2.6. Link Budget Air Interface HSDPA

Kategori	Unit	Nilai
Frekuensi Operasi	MHz	2642,5
Jarak IMR-UE	Km	0,4
EIRP	dBW	57
<i>Interference Margin</i>	dB	9
Rugi-rugi Lintasan Propagasi	dB	126,36
<i>Gain</i> antena Rx	dB	0
<i>Noise Figure</i> UE	dB	8
<i>Implement Losses</i>	dB	3
<i>Noise Power Spectral Density</i>	dBW/Hz	-204,3

Sumber: Yosina Septianingsih, 2008:69

- Daya yang Diterima Kanal Trafik

$$P_{tr} = EIRP - L_p + G_{MS} - I_m \quad (2.80)$$

dengan:

P_{tr} = Daya yang diterima kanal trafik (dBW)

$EIRP$ = EIRP kanal trafik (dBW)



L_p = Rugi-rugi propagasi (dB)

G_{MS} = Gain antena mobile station (dB)

Im = Interference margin (dB)

- Thermal Noise

$$N_o = 10 \log(290 \times 1,38 \cdot 10^{-23}) + NF_{MS} + 40 \quad (2.81)$$

dengan:

No = Thermal noise (dBm/Hz)

Konstanta Boltzmann = $1,38 \cdot 10^{-23}$ J/K

NF_{MS} = Noise figure mobile station (dB)

- Eb/No Kanal Trafik

$$[E_b/N_o]_{Tr} = P_{tr} - 10 \log R_b - 10 \log 10^{0,1N_o} \quad (2.82)$$

dengan:

$[E_b/N_o]_{Tr}$ = Energy per bit-to-noise density ratio kanal trafik (dB)

P_{tr} = Daya yang diterima kanal trafik (dBW)

R_b = Bit rate data (bps)

No = Thermal noise (dBm/Hz)

- Probabilitas Error Kanal Trafik

$$p_e = \frac{e^{-Eb/No}}{\sqrt{\pi(Eb/No)}} \quad (2.83)$$

dengan:

P_e = Probabilitas BER (Bit Error Rate) pada kanal trafik HSDPA

- Probabilitas packet loss air interface HSDPA

$$\rho_{AI} = 1 - (1 - P_e)^{l+l'} \quad (2.84)$$

dengan:

ρ_{AI} = probabilitas packet loss air interface HSDPA

P_e = Probabilitas Bit Error Rate pada kanal trafik HSDPA (BER)

l = Panjang paket data (bit)

l' = Panjang header (bit)

Pada jaringan EDGE, probabilitas packet loss dihitung dengan persamaan berikut:



$$\rho_{net} = 1 - [(1 - \rho_{AI})(1 - \rho_{GERAN})^h(1 - \rho_{SGSN})(1 - \rho_{GGSN})(1 - \rho_{PDN Internet})] \quad (80)$$

ρ_{net} = probabilitas *packet loss* jaringan EDGE

ρ_{AI} = probabilitas *packet loss* EDGE *air interface*

ρ_{GERAN} = probabilitas *packet loss* pada GERAN (BSC dan BTS)

ρ_{SGSN} = probabilitas *packet loss* pada SGSN

ρ_{GGSN} = probabilitas *packet loss* pada GGSN

h = jumlah *hop* pada jaringan GERAN (BSC dan BTS)

Probabilitas *packet loss* pada masing-masing *node* dihitung dari probabilitas *Bit Error Rate* (BER) di jaringan tersebut, dengan persamaan:

$$\rho_{node} = (l - l')\rho_b \quad (2.85)$$

Dengan:

l = panjang paket data (byte)

l' = panjang header (byte)

ρ_b = probabilitas *Bit Error Rate* (BER)

Probabilitas *packet loss* pada *air interface* dapat dihitung dengan persamaan berikut (Anurag Kumar, 2008:75):

$$\rho_{AI} = 1 - (1 - P_e)^{l-l'} \quad (2.86)$$

Dengan:

l = panjang paket data (byte)

l' = panjang header (byte)

P_e = probabilitas *Bit Error Rate* (BER)

Probabilitas *packet loss* pada *mobile TV* dapat dihitung dengan persamaan:

$$\rho_{mobile tv} = P_{size} \cdot \rho_b \quad (2.87)$$

Dengan:

$\rho_{mobile tv}$ = probabilitas *packet loss* pada *mobile TV*

P_{size} = panjang data *mobile TV*

ρ_b = probabilitas *bit error* pada jaringan

Sedangkan waktu minimum yang diperlukan untuk mengetahui paket yang berhasil dikirimkan dapat dinyatakan sebagai berikut:

$$t_T = t_I + t_{out} \quad (s) \quad (2.88)$$

t_T = waktu minimum untuk mengetahui sebuah paket berhasil ditransmisikan (s)



- t_{out} = interval waktu pengiriman sebuah paket dengan pengiriman paket selanjutnya(s)
- t_I = waktu yang dibutuhkan untuk mengirimkan sebuah *frame* (s)

Waktu transmisi *frame* ditentukan dengan persamaan (Schwartz, 1987:132):

$$t_I = \frac{(Pl_{frame} + H_{frame}) \times 8}{C_{trans}} \quad (s) \quad (2.89)$$

Dengan:

- t_I = waktu yang diperlukan untuk mentransmisikan sebuah *frame* (s)
- Pl_{frame} = *payload frame* (s)
- H_{frame} = *header frame* (byte)
- C_{trans} = kecepatan saluran transmisi (bps)

Jaringan EDGE dan HSDPA menggunakan mekanisme *error control Selective Reject Protocol*, sehingga (Schwartz, 1987:129):

$$t_{out} = 2t_p + 2t_I + 2t_{proc} \quad (s) \quad (2.90)$$

Dengan:

- t_{out} = waktu yang dibutuhkan untuk menerima *acknowledge* (s)
- t_p = *delay propagasi* (s)
- t_I = waktu yang diperlukan untuk mentransmisikan sebuah *frame* (s)
- t_{proc} = *delay proses* (s)

Nilai dari t_{proc} dapat dihitung sesuai dengan persamaan:

$$t_{proc} = \frac{t_{ED}}{N_{frame}} + \frac{t_{W\ total}}{N_{frame}} \quad (2.91)$$

dengan:

- t_{ED} = *delay enkapsulasi dan dekapsulasi*
- $t_{W\ total}$ = *delay antrian total*
- N_{frame} = jumlah frame

Dengan mengetahui nilai t_{out} dan t_I dapat dihitung parameter α (Schwartz, 1987:129) dengan persamaan berikut:

$$\alpha = 1 + \frac{t_{out}}{t_I} \quad (2.92)$$

Dimana:

- α = konstanta
- t_{out} = waktu yang dibutuhkan untuk menerima *acknowledge* (s)
- t_I = waktu yang diperlukan untuk mentransmisikan sebuah *frame* (s)



BAB III

METODOLOGI

Kajian dalam skripsi ini bersifat analisis, yaitu analisis terhadap *handover* dari sistem komunikasi HSDPA ke EDGE untuk layanan *mobile TV*. Metodologi penulisan yang digunakan dalam skripsi ini adalah sebagai berikut:

3.1 Studi Literatur

Studi literatur bertujuan untuk mempelajari dan memahami konsep yang mengenai *mobile TV*, *handover* dari jaringan HSDPA ke EDGE, serta parameter performansi jaringan dan beberapa teori penunjang lain yang digunakan dalam penyusunan skripsi ini.

3.2 Metode Pengambilan Data

Data yang digunakan dalam kajian ini berupa data sekunder, yaitu data yang diperoleh dari buku referensi, jurnal, skripsi, *web browsing*, serta rekomendasi dari badan standar telekomunikasi internasional yang berhubungan dengan penerapan teknologi HSDPA dan EDGE. Data-data tersebut diantaranya adalah sebagai berikut:

3.2.1 Pada Jaringan EDGE

a) Spesifikasi

Data spesifikasi mengenai teknologi EDGE diperlukan untuk menganalisis jaringan selular EDGE dan parameter kinerja kualitas pelayanan data (dalam hal ini data *Mobile TV*). Data-data yang digunakan meliputi:

- Arsitektur jaringan EDGE
- *Protocol stack* jaringan EDGE
- Nilai BER ditentukan berdasarkan kelas layanan EDGE adalah 10^{-4} .

b) Pembatasan Variabel Data

Pembatasan variabel data pada jaringan EDGE digunakan untuk membatasi permasalahan sehingga tidak melebihi atau kurang dari format fokus pembahasan yang ingin disampaikan. Dalam skripsi ini, variabel-variabel data untuk jaringan EDGE yang digunakan meliputi:

- MCS (*Modulation Coding Scheme*) yang digunakan adalah MCS 1 samapai 9, yang ditunjukkan pada Tabel 3.1 dan untuk ukuran RLC Data Unit pada EDGE ditunjukkan pada Tabel 3.2.



Tabel 3.1 Modulasi dan *Coding Scheme* pada EDGE

MCS	USF	Precoded USF		Header	HCS	Header Code Rate		Data BCS	Data Code Rate	Data Rate in Kbps
		Header	HCS			Data	BCS			
MCS-1 DL	3	12	28	8	=1/2	178	12	=1/2	8.9	
MCS-1 UL			31	8	=1/2	178	12	=1/2	8.9	
MCS-2 DL	3	12	28	8	=1/2	226	12	0.64	11.3	
MCS-2 UL			31	8	=1/2	226	12	0.64	11.3	
MCS-3 DL	3	12	28	8	=1/2	298	12	0.83	14.9	
MCS-3 UL			31	8	=1/2	298	12	0.83	14.9	
MCS-4 DL	3	12	28	8	=1/2	354	12	=1	17.7	
MCS-4 UL			31	8	=1/2	354	12	=1	17.7	
MCS-5 DL	3	36	25	8	=1/3	450	12	=1/3	22.5	
MCS-5 UL			37	8	=1/3	450	12	=1/3	22.5	
MCS-6 DL	3	36	25	8	=1/3	594	12	=1/2	29.7	
MCS-6 UL			37	8	=1/3	594	12	=1/2	29.7	
MCS-7 DL	3	36	37	8	=1/3	2*450	12	=3/4	45	
MCS-7 UL			46	8	=1/3	2*450	12	=3/4	45	
MCS-8 DL	3	36	37	8	=1/3	2*546	12	0.90	54.6	
MCS-8 UL			46	8	=1/3	2*546	12	0.90	54.6	
MCS-9 DL	3	36	37	8	=1/3	2*594	12	=1	59.4	
MCS-9 UL			46	8	=1/3	2*594	12	=1	59.4	

Sumber : Emmanuel Seurre, 2003

Tabel 3.2 Ukuran RLC Data Unit pada EDGE

MCS	EGPRS RLC Data Unit Size in Bytes	Additional Bit in the Data Part
MCS-1	22	2
MCS-2	28	2
MCS-3	37	2
MCS-4	44	2
MCS-5	56	2
MCS-6	74	2
MCS-7	2*56	2*2
MCS-8	2*68	2*2
MCS-9	2*74	2*2

Sumber : Emmanuel Seurre, 2003

- Faktor utilitas menggunakan nilai $0 < \rho < 1$ dengan selisih 0,1 setiap step.
- Model hubungan end-to-end yang digunakan adalah dari PDN *Internet* - GGSN - 2.75G SGSN - BSC(PCU) - BTS - UE.
- Kecepatan tiap *node*, jarak antar *node*, dan kecepatan tiap *interface* pada jaringan EDGE ditunjukkan dalam Tabel 3.3, Tabel 3.4, serta Tabel 3.5.

Tabel 3.3 Kecepatan Tiap *Node* pada Jaringan EDGE

Node	Kecepatan (bps)
PDN Internet	10^9
GGSN	100.10^6
2.75G SGSN	15.10^6
PCU	$31.64.10^3$

Sumber: Fatimah, 2008

Tabel 3.4 Jarak Antar *Node* pada Jaringan EDGE

Node-to-node	Jarak (m)
PDN Internet – GGSN	1500
GGSN – 2.75G SGSN	1000
2.75G SGSN – BSC	25
BSC – BTS	50
BTS – UE	100

Sumber: Fatimah, 2008

Tabel 3.5 Kecepatan tiap *Interface* pada Jaringan EDGE

Interface	Kecepatan (bps)
Gi interface	10^9
Gn interface	100.10^6
Gb interface	$2.048.10^6$
Abis interface	16.10^3
Um interface	Sesuai bit rate di tiap MCS

Sumber: Fatimah, 2008

3.2.2 Pada Jaringan HSDPA

a) Spesifikasi

Spesifikasi yang digunakan pada jaringan HSDPA adalah sebagai berikut:

- Arsitektur jaringan HSDPA
- *Protocol stack* jaringan HSDPA

b) Pembatasan Variabel Data

Pada skripsi ini, pembatasan variabel data untuk jaringan HSDPA meliputi:

- *Radius cell* yang digunakan adalah untuk type sel *micro* dengan daerah operasi urban dan pusat yaitu sebesar 0,4 km dengan kecepatan data maksimal 384 kbps, sesuai dengan daerah layanan HSDPA.
- Kategori *User Equipment* yang digunakan dalam analisis adalah *User Equipment* kategori 5 sesuai Tabel 3.6.
- Faktor utilisasi menggunakan nilai $0 < \rho < 1$ dengan selisih 0,1 setiap step.
- Model hubungan *end-to-end* yang digunakan adalah dari PDN Internet - GGSN - 2.75G SGSN - RAN (*node B* dan *RNC*) - UE.

Tabel 3.6 Kategori UE pada HSDPA

HS-DSCH category	HS-PDSCH codes	Modulation supported	Min. inter-TTI	Max. TBS [bits]	UE IR buffer size [SML]	Peak data rate [Mbps]
1	5	QPSK and 16-QAM	3	7298	19200	1.2
2	5	QPSK and 16-QAM	3	7298	28800	1.2
3	5	QPSK and 16-QAM	2	7298	28800	1.8
4	5	QPSK and 16-QAM	2	7298	38400	1.8
5	5	QPSK and 16-QAM	1	7298	57600	3.6
6	5	QPSK and 16-QAM	1	7298	67200	3.6
7	10	QPSK and 16-QAM	1	14411	115200	7.2
8	10	QPSK and 16-QAM	1	14411	134400	7.2
9	15	QPSK and 16-QAM	1	20251	172800	10.1
10	15	QPSK and 16-QAM	1	27952	172800	14.0
11	5	QPSK	2	3630	14400	0.9
12	5	QPSK	1	3630	14400	1.8

Sumber: Christope Chevallier, 2006:282

- Kecepatan tiap *node*, jarak antar *node*, dan kecepatan tiap *interface* pada jaringan HSDPA ditunjukkan dalam Tabel 3.7, Tabel 3.8, serta Tabel 3.9.

Tabel 3.7 Kecepatan Tiap *Node* pada Jaringan HSDPA

Node	Kecepatan (bps)
PDN Internet	10^9
GGSN	100.10^6
3.5G SGSN	100.10^6
RNC	$155,52.10^6$
Node B	$155,52.10^6$

Sumber: Indosat dari Skripsi Yosina Septianingsih, 2008

Tabel 3.8 Jarak Antar *Node* pada Jaringan HSDPA

Node-to-node	Jarak (m)
PDN Internet – GGSN	1500
GGSN – 3.5G SGSN	2000
3.5G SGSN – RNC	3000
RNC – Node B	400

Sumber: Indosat dari Skripsi Linda Ekowati, 2008

Tabel 3.9 Kecepatan tiap *Interface* pada Jaringan HSDPA

Interface	Kecepatan (bps)
Gi interface	10^9
Gn interface	100.10^6
Iu interface	$155,52.10^6$
Iub interface	$155,52.10^6$
Uu interface	Sesuai data rate maksimum UE

Sumber: Indosat dari Skripsi Linda Ekowati, 2008



- Parameter *link budget* ditunjukkan pada Tabel 3.10.

Tabel 3.10 Link Budget Air Interface HSDPA

Kategori	Unit	Nilai
Frekuensi Operasi	MHz	2642,5
Jarak IMR-UE	Km	0,4
EIRP	dBW	57
<i>Interference Margin</i>	dB	9
Rugi-rugi Lintasan Propagasi	dB	126,36
<i>Gain</i> antena Rx	dB	0
<i>Noise Figure</i> UE	dB	8
<i>Implement Losses</i>	dB	3
<i>Noise Power Spectral Density</i>	dBW/Hz	-204,3

Sumber: Indosat dari Skripsi Yosina Septianingsih, 2008

3.2.3 Handover antara jaringan HSDPA dan EDGE

a) Spesifikasi

Data yang digunakan pada *Handover* antara Jaringan HSDPA dan EDGE adalah Paket *message* ketika terjadi proses *Signaling Handover* menggunakan ICMP *message* berukuran 8 byte.

b) Pembatasan Variabel Data

- Batasan yang diberikan dalam pembahasan analisis ini, ditekankan pada parameter *delay handover* dan prosedur *handover* yang dibahas. Mekanisme enkripsi dan otentikasi *user* tidak dibahas.
- Efek ping-pong pada jaringan tidak dibahas.

3.2.4 Pada mobile TV

a) Spesifikasi

Data yang diperlukan pada spesifikasi *Mobile TV* merupakan data dari panjang paket *Mobile TV*. Standar yang digunakan untuk menentukan panjang paket *Mobile TV* antara lain:

- *Payload* data aplikasi menggunakan *header RTP*, *header UDP*, *header MPE-FEC*, *CRC*, dan *header MPEG-2 TS*.
- Menggunakan jenis *codec* audio AMR-WB dan *codec* video H.264, sesuai Tabel 3.11.

b) Pembatasan Variabel Data

- Protokol yang digunakan dalam pengiriman paket data *mobile TV* adalah protokol TCP/IP serta menggunakan IPv4 dan IPv6.
- Tidak membahas tentang cara kerja *Mobile TV* secara mendetail dari permintaan layanan pada server sampai dengan ketersediaan layanan *Mobile TV* pada *User Equipment* (UE).

Tabel 3.11 Audio dan Video *Codec* pada Aplikasi *Mobile TV*

Audio CODEC	Bit Rate (kbps)	Maximum Payload (byte)	Delay CODEC (ms)
AMR-WB	6,6-23,85	35	10-20
AMR-WB+	5,2-48	46	20-40
HE-AAC v2	128-320	80	40-80
Video CODEC	Bit Rate (kbps)	Maximum Payload (byte)	Delay CODEC (ms)
H.264/AVC	64-384	254	150-300

Sumber: RFC 4352 and RFC 3984 RTP Payload Format for H.264 Video, 2005

3.3 Metode Analisis dan Pembahasan

Analisis data dalam kajian *handover* dari jaringan HSDPA ke EDGE ini dilakukan dengan mengumpulkan nilai parameter data sekunder yang dibutuhkan dan sesuai dengan standar yang digunakan, untuk kemudian dianalisis yang meliputi:

- Analisis kualitas jaringan EDGE dan HSDPA dengan melakukan perhitungan *delay end-to-end* dan *throughput* jaringan.
- Analisis *delay handover* yang terjadi selama proses *handover* untuk paket data *mobile TV*, beserta pengaruhnya terhadap *throughput* jaringan.

3.3.1 Menentukan Arsitektur Jaringan EDGE dan HSDPA

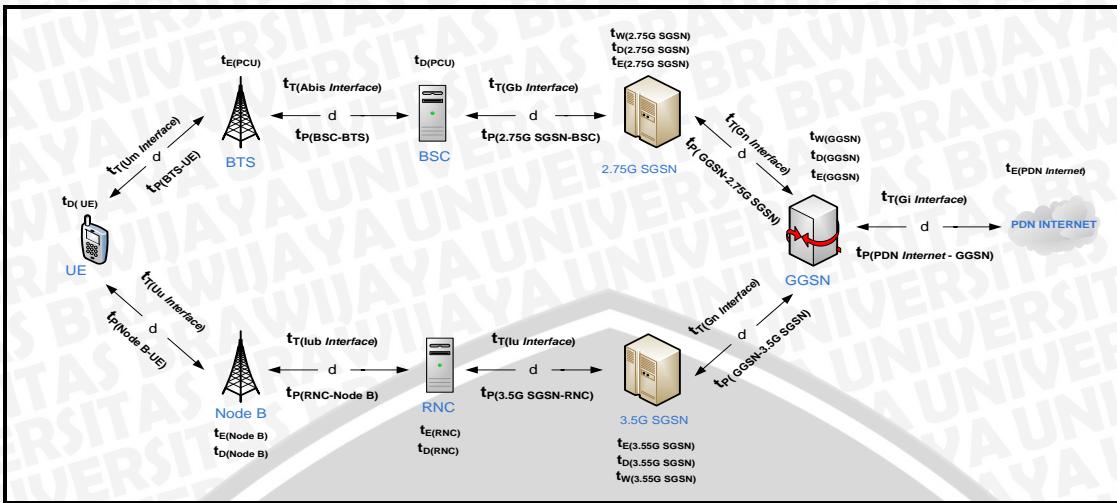
Melakukan studi literatur tentang jaringan EDGE dan HSDPA, beserta spesifikasinya untuk menentukan bagaimana konfigurasi kedua jaringan tersebut.

3.3.2 Menentukan Standar Aplikasi *Mobile TV*

Mencari data sekunder tentang aplikasi *mobile TV* dan spesifikasinya untuk menentukan bagaimana paket data pada aplikasi *mobile TV*.

3.3.3 Menentukan Pemodelan Arsitektur Jaringan HSDPA dan EDGE

Merancang arsitektur *node-to-node* sesuai dengan prosedur pensinyalan *handover* dari jaringan HSDPA ke EDGE. Model rancangan diilustrasikan pada Gambar 3.1

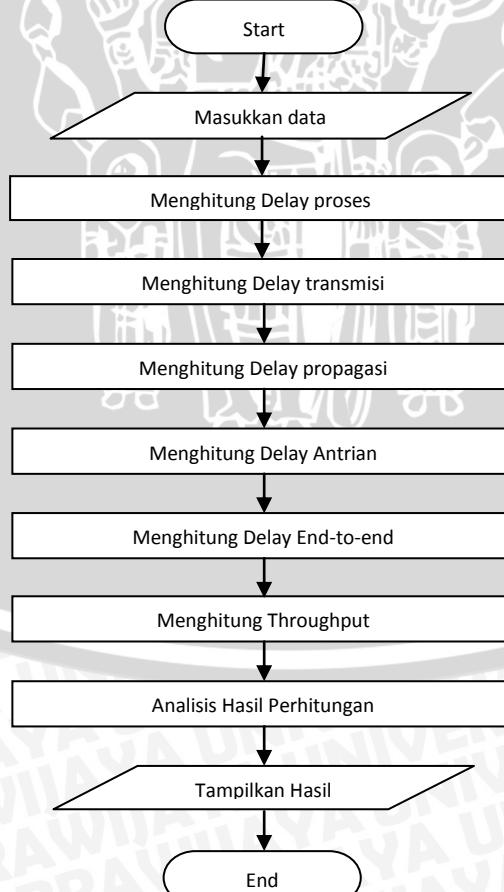


Gambar 3.1 Arsitektur *Node-to-node* HSDPA ke EDGE
Sumber: Perancangan

3.3.4 Menganalisis Performansi Jaringan HSDPA dan EDGE

Perhitungan matematis dilakukan dengan mengumpulkan nilai – nilai parameter yang berupa data sekunder untuk kemudian dianalisis. Perhitungan yang dilakukan meliputi:

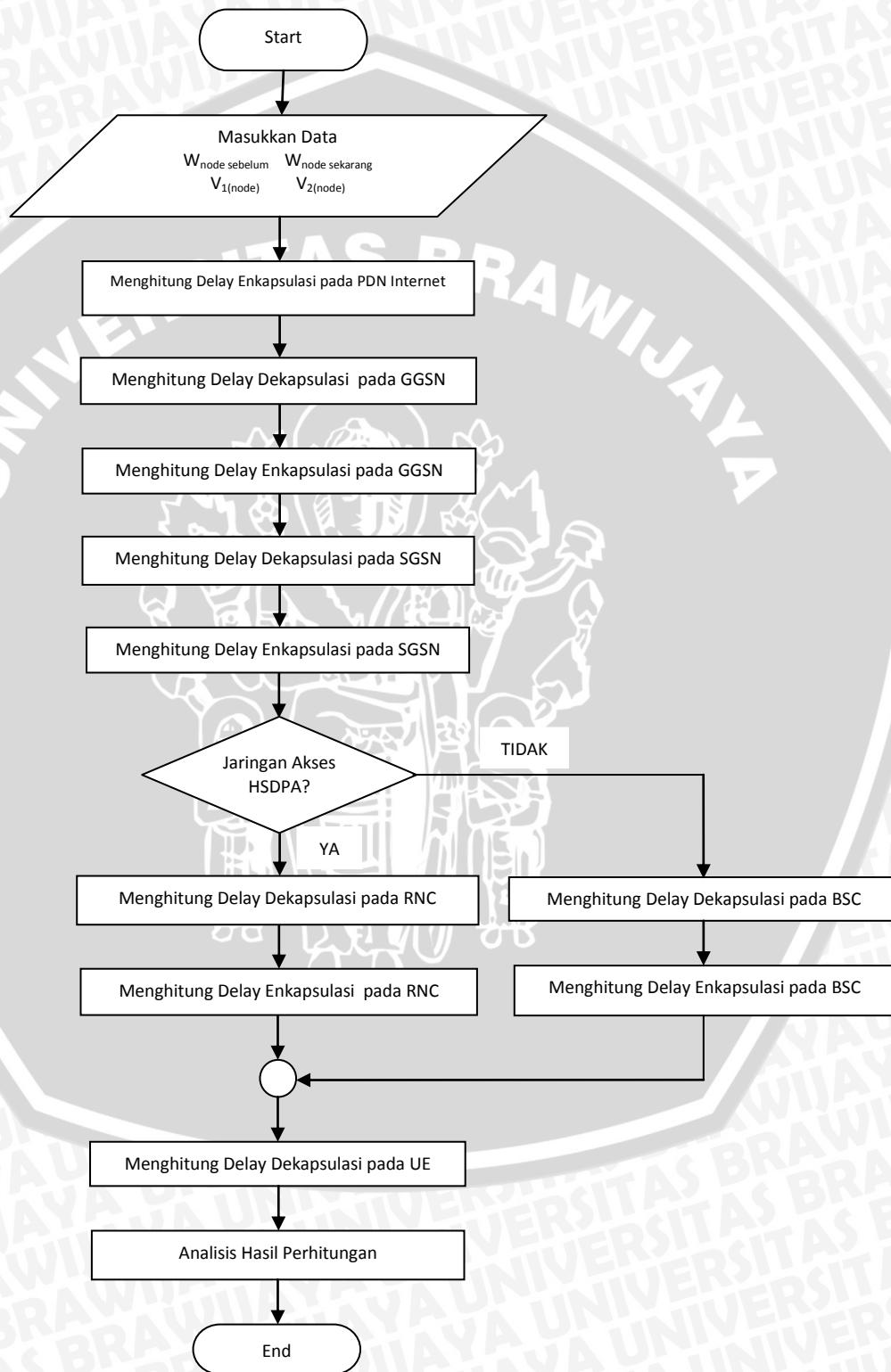
- Perhitungan Performansi Jaringan HSDPA dan EDGE



Gambar 3.2 Diagram Alir Perhitungan Parameter *Handover*

- *Delay proses*

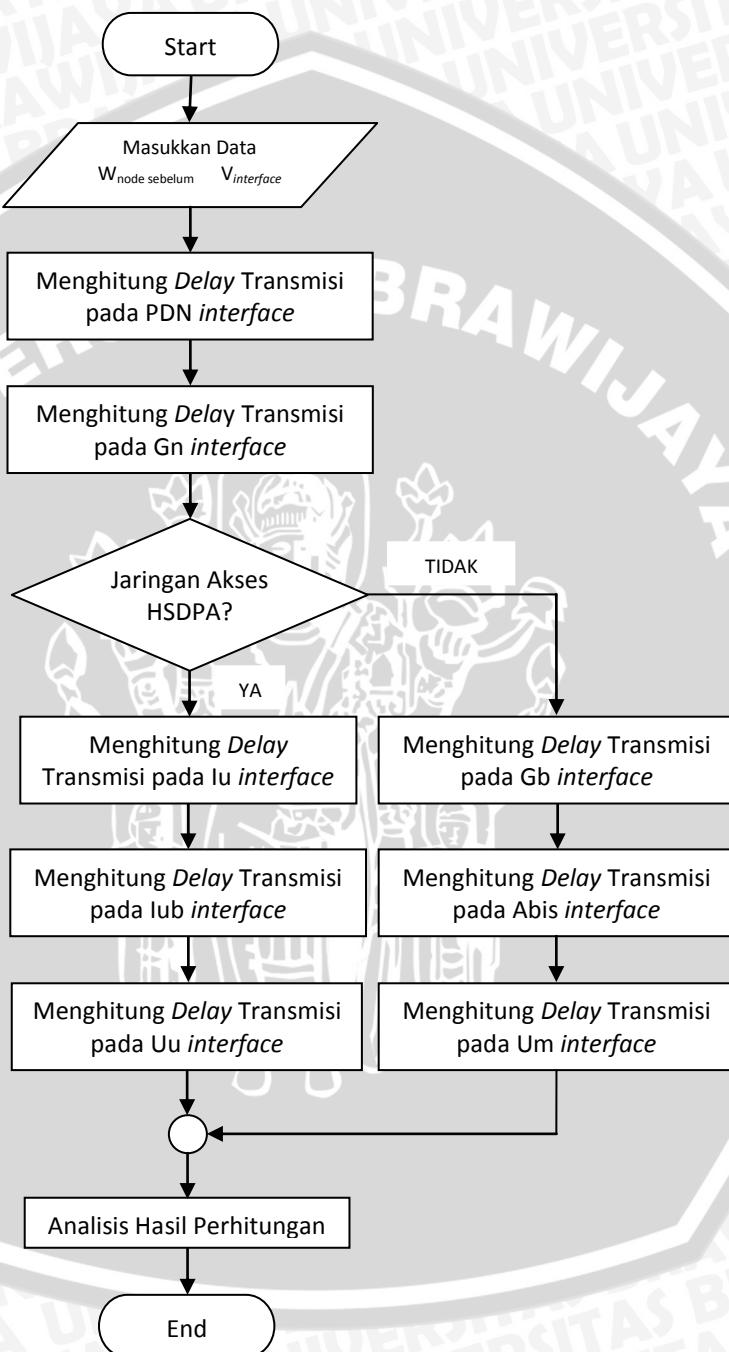
Delay proses terdiri dari *delay enkapsulasi* dan *dekapsulasi* pada tiap *node*. Adapun data yang dibutuhkan adalah panjang data yang digunakan (W_{data}), jumlah *header*, dan kecepatan transmisi data pada tiap *node* (V_{node})



Gambar 3.3 Diagram Alir Perhitungan *Delay Proses*

- *Delay Transmisi*

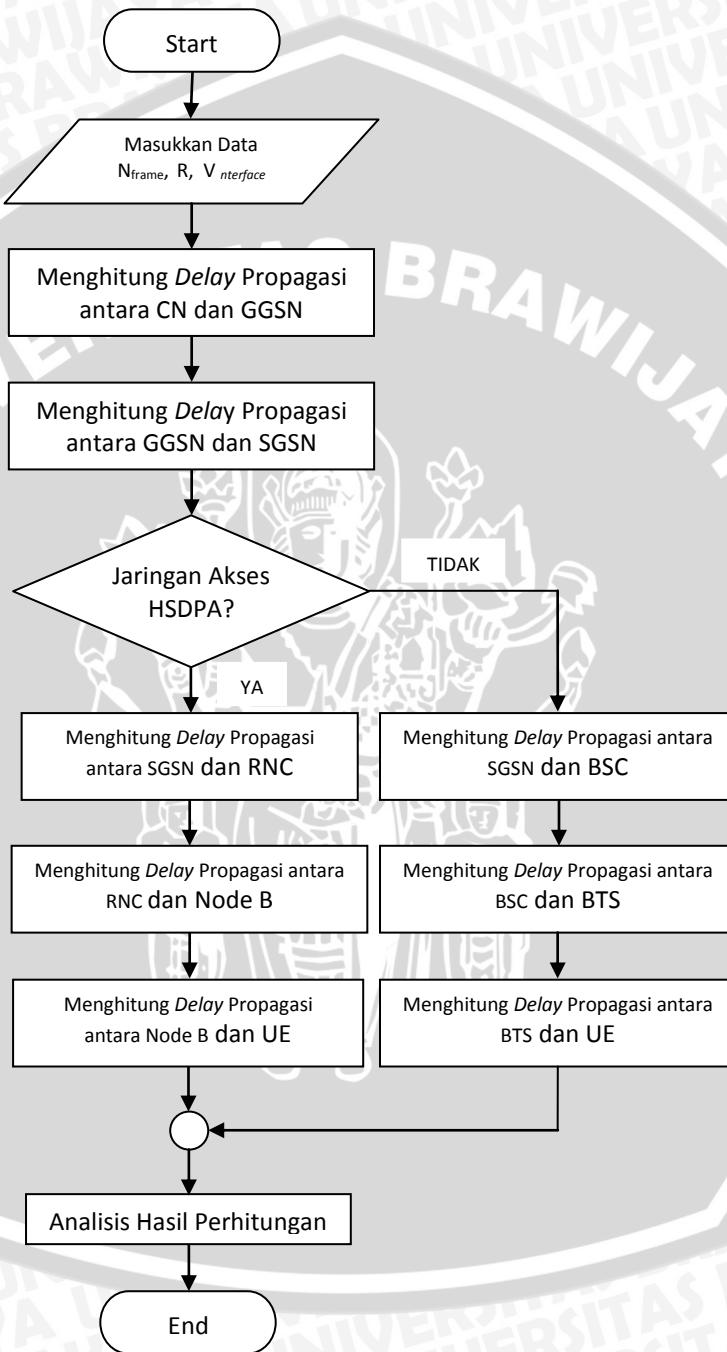
Untuk mendapatkan nilai *delay transmisi*, data yang dibutuhkan adalah panjang data, jumlah *header*, dan kecepatan transmisi pada masing–masing saluran transmisi yang menghubungkan setiap *node* ($v_{interface}$).



Gambar 3.4 Diagram Alir Perhitungan *Delay Transmisi*

- *Delay* propagasi

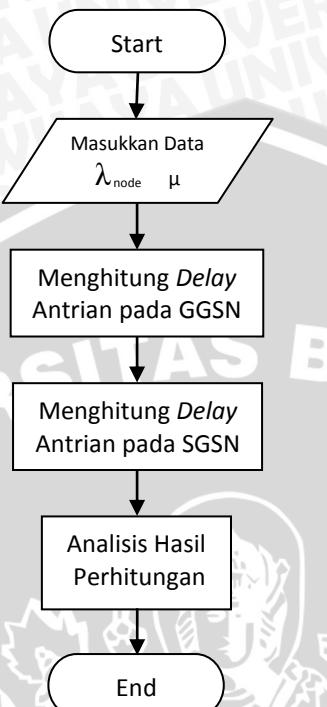
Untuk mendapatkan nilai *delay* propagasi, dibutuhkan data-data jumlah *frame* yang ditransmisikan (N_{frame}), jarak antar *node* (R), dan cepat rambat dalam medium ($v_{interface}$).



Gambar 3.5 Diagram Alir Perhitungan *Delay* Propagasi

- *Delay Antrian*

Data yang dibutuhkan untuk mendapatkan nilai *delay* antrian adalah kecepatan kedatangan paket pada *node* (λ), dan kecepatan pelayanan *node* (μ).

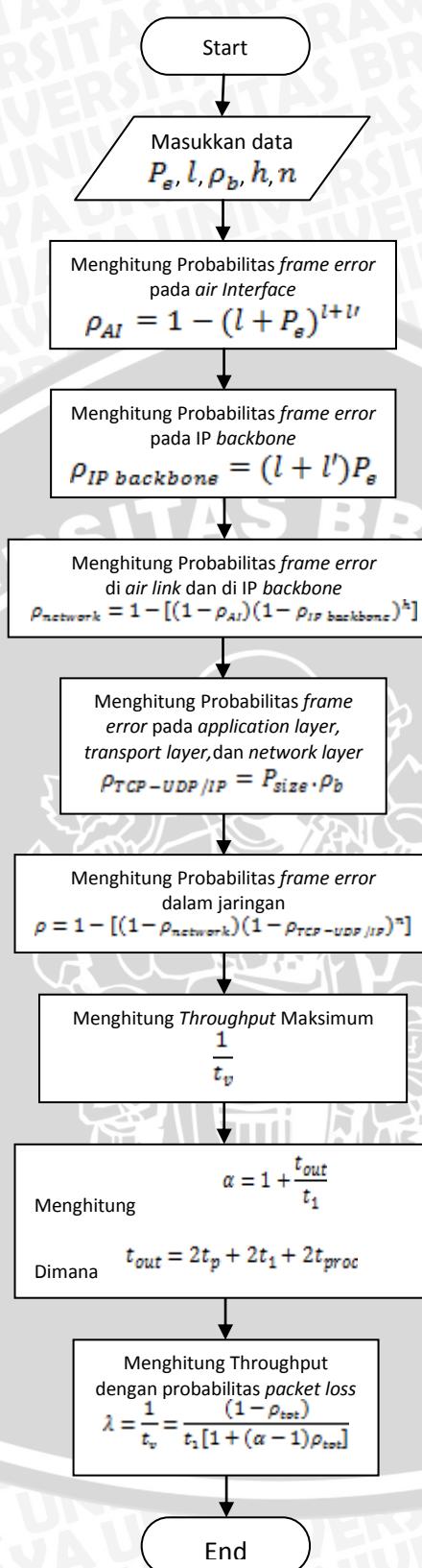


Gambar 3.6 Diagram Alir Perhitungan *Delay Antrian*

- Menghitung nilai *throughput*

Dengan nilai-nilai yang telah didapatkan sebelumnya, maka nilai *throughput* bisa dihitung.





Gambar 3.7 Diagram Alir Perhitungan Throughput

3.3.5 Menentukan Prosedur *Handover* Antara Jaringan HSDPA dan EDGE

Mencari prosedur pensinyalan *handover* dari jaringan HSDPA ke EDGE untuk menganalisis pengaruh *handover* dari sistem komunikasi HSDPA ke EDGE untuk layanan *mobile TV*.

3.3.6 Menganalisis *Handover* dari Jaringan HSDPA ke EDGE

Analisis *delay handover* yang terjadi selama proses *handover* untuk paket data *mobile TV*, beserta pengaruhnya terhadap *throughput* jaringan.

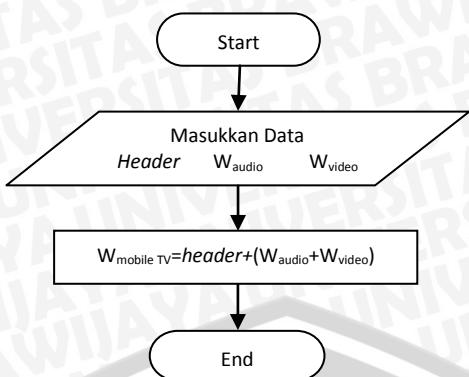


Gambar 3.8 Diagram Alir Perhitungan Parameter *Handover*

- Perhitungan Besar Paket Data *Mobile TV*

Data yang dibutuhkan untuk mendapatkan besar paket data *mobile TV* adalah *header RTP*, *header UDP*, *header IP*, *header MPE-FEC*, *CRC*, *header MPEG-2 TS*, *payload codec* untuk audio (W_{audio}), serta *payload* untuk *codec video* (W_{video}).





Gambar 3.9 Diagram Alir Perhitungan Paket Data Mobile TV

3.4 Pengambilan Kesimpulan dan Saran

Berdasarkan hasil perhitungan dan analisis yang telah dilakukan, maka dapat ditarik kesimpulan sebagai jawaban untuk rumusan masalah.



BAB IV

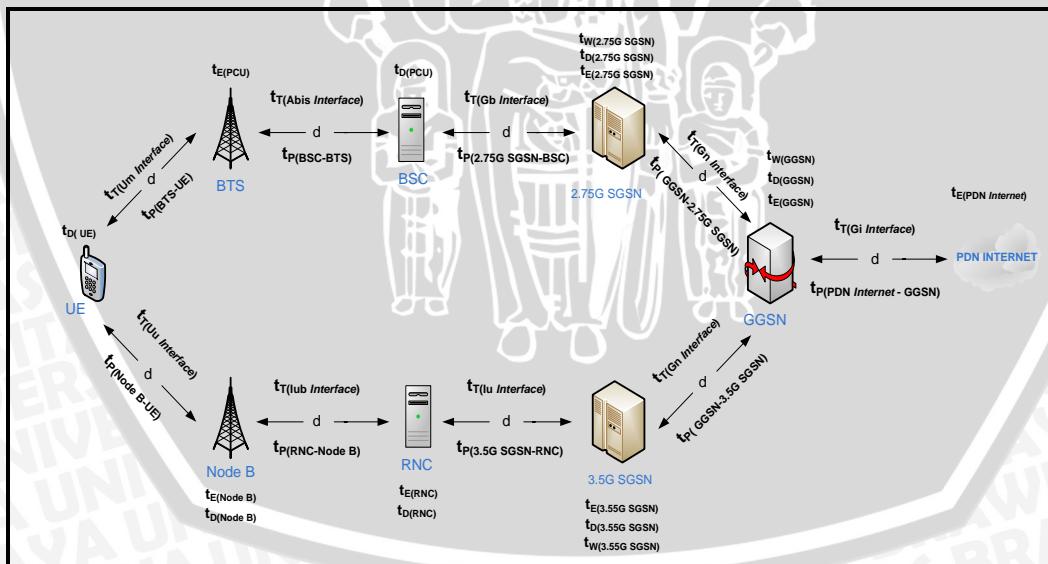
HASIL DAN PEMBAHASAN

4.1 Umum

Pembahasan dan analisis yang dilakukan pada bab ini adalah tentang pengaruh *handover* pada jaringan HSDPA ke EDGE untuk layanan *mobile TV*. Untuk itu ada beberapa tahap pembahasan dan analisis yang dilakukan:

1. Analisis kinerja terhadap jaringan EDGE dan HSDPA yang digunakan, yaitu dengan menghitung nilai masing-masing *delay* (*delay* enkapsulasi/dekapsulasi, *delay* transmisi, *delay* propagasi, *delay* antrian) selama perjalanan paket multimedia *Mobile TV*. Analisis yang akan diperhitungkan pada jaringan EDGE meliputi analisis *delay* dan *throughput* di BTS, BSC, SGSN, GGSN, dan PDN *Internet*. Pada jaringan HSDPA parameter analisis meliputi perhitungan *delay* dan *throughput* pada BTS, *Node B*, SGSN, GGSN, dan PDN *Internet*. Dengan asumsi bahwa paket multimedia *Mobile TV* dilewatkan oleh PDN *Internet* melalui Gi *interface*.
2. Analisis *delay handover* yang terjadi selama proses *handover* berikut pengaruhnya terhadap *throughput*.

Analisis *delay* dalam jaringan dihitung mulai dari PDN *Internet* sampai dengan UE dapat diilustrasikan dalam Gambar 4.1.



Gambar 4.1 Arsitektur Node-to-node HSDPA ke EDGE
Sumber: Perancangan

4.2 Paket Data Aplikasi *Mobile TV*

Aplikasi *mobile TV* pada skripsi ini menggunakan jenis *codec* AMR-WB untuk audio dan H.264 untuk video. Berdasarkan Tabel 2.3, *payload* maksimum untuk H.264 adalah sebesar 254 byte sedangkan AMR-WB memiliki *payload* maksimum sebesar 35

byte. Paket data tersebut ditransmisikan dari *Mobile TV Host* ke jaringan EDGE dan HSDPA.

Payload data aplikasi akan ditambahkan *header RTP*, *header UDP*, *header IP*, *header MPE-FEC*, *CRC*, dan *header MPEG-2 TS*. Karena pada skripsi ini yang digunakan adalah IPv6 sehingga *header* yang digunakan adalah *header IPv6* sebesar 40 byte. Besar *header* tiap paket adalah *header RTP* 12 byte, *header UDP* 8 byte, *header IP* 40 byte, *header MPE-FEC* 12 byte dan *CRC* 4 byte, serta *header MPEG-2 TS* 4 byte. Sehingga total *header* adalah 80 byte.

Besar paket data *Mobile TV* untuk jenis *codec* audio AMR-WB dan *codec* video H.264 yang akan ditransmisikan oleh *Mobile TV Host* adalah sebagai berikut :

$$\begin{aligned} W_{\text{mobile TV}} &= (35 + 254) \text{ byte} + 80 \text{ byte} \\ &= 369 \text{ byte} \end{aligned}$$

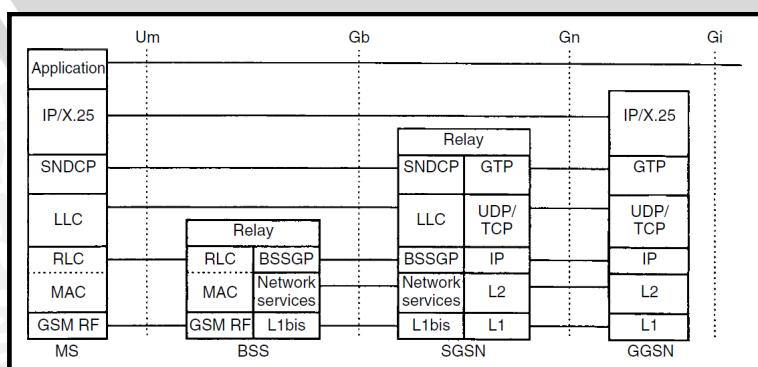
4.3 Performansi Jaringan EDGE/HSDPA

4.3.1 Analisis Delay

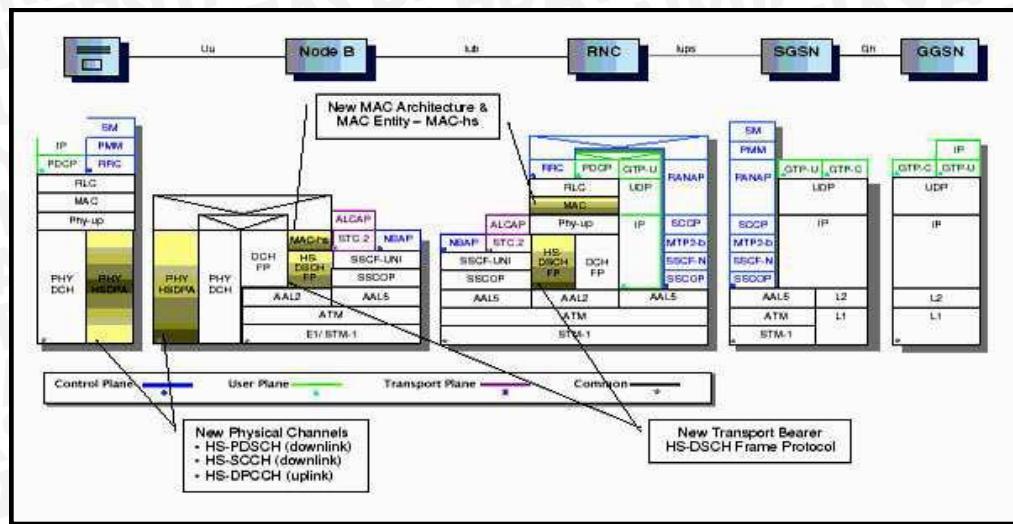
Dalam jaringan berbasis *packet switching*, *delay* yang terjadi merupakan penjumlahan *delay-delay* yang ada dalam perjalanan paket dari sumber ke tujuannya pada setiap *hop*. Pada jaringan HSDPA dan jaringan EDGE, *delay* yang terjadi meliputi *delay proses*, *delay transmisi*, *delay propagasi* dan *delay antrian*.

4.3.1.1 Delay Proses

Pada analisis *delay* proses ini, akan dilakukan perhitungan *delay* enkapsulasi dan dekapsulasi untuk panjang segmen 369 byte pada semua *node* yang dilalui selama proses pentransmision data dari PDN *Internet* sampai pada UE. Perhitungan *delay* enkapsulasi dan dekapsulasi berlaku untuk masing-masing *layer* seperti Gambar 4.2 untuk jaringan EDGE, dan Gambar 4.3 untuk jaringan HSDPA.



Gambar 4.2 Arsitektur Protokol EDGE
Sumber : Timo Halonen, 2003



Gambar 4.3 Arsitektur Protokol HSDPA

Sumber: Linda Ekowati, 2008

a) PDN Internet

Mobile TV merupakan aplikasi *real time*, oleh karena itu pada *transport layer* paket data sepanjang 369 byte akan ditambahkan *header UDP* sehingga diperoleh panjang segmen UDP sesuai dengan persamaan:

$$\begin{aligned} W_{segmen} &= W_{mobile\ TV\ (PDN\ Internet)} + header_{UDP} \\ &= 369\ byte + 8\ byte \\ &= 377\ byte \end{aligned}$$

Layer IP pada jaringan tersebut memiliki MSS (*Maximum Segment Size*) sebesar 65535 byte (rekomendasi IEEE: RFC 0791). Apabila paket data yang masuk melebihi MSS maka akan langsung disegmentasi. Jika panjang segmen tersebut kurang dari MSS IP akan langsung ditambahkan *header IP* menjadi *datagram IP* dan dengan menggunakan persamaan (2.3) diperoleh panjang *datagram IP* ($W_{datagram}$) sebesar :

$$\begin{aligned} W_{datagram} &= W_{segmen} + header_{IP} \\ &= 377\ byte + 40\ byte \\ &= 417\ byte \end{aligned}$$

Datagram IP yang melebihi MTU (*Maximum Transmission Unit*) *Ethernet* sebesar 1500 byte, akan disegmentasi terlebih dahulu. Apabila *datagram IP* tidak melebihi MTU *Ethernet*, *datagram IP* akan langsung ditambahkan dengan *header Ethernet* dan FCS. Selanjutnya dengan menggunakan persamaan (2.4) diperoleh panjang *frame PDN Internet* yang siap untuk ditransmisikan ke GGSN melalui Gi *Interface* sebesar:

$$\begin{aligned} W_{frame\ PDN\ Internet} &= W_{datagram} + header_{Ethernet} + FCS \\ &= 417\ byte + 14\ byte + 4\ byte \\ &= 435\ byte \end{aligned}$$

Pada skripsi ini PDN *Internet* menggunakan standar jenis *Gigabit Ethernet* dengan kecepatan 1 Gbps. Sesuai persamaan (2.5), maka *delay* enkapsulasi yang terjadi di PDN *Internet* yaitu:

$$\begin{aligned} t_{E(PDN\ Internet)} &= \frac{W_{frame\ PDN\ Internet} \times 8}{V_{PDN\ Internet}} \\ &= \frac{(435 \times 8)bit}{1 \times 10^9 bps} \\ &= 3,48 \times 10^{-6} s \end{aligned}$$

b) GGSN (*Gateway GPRS Support Node*)

Frame PDN *Internet* yang diterima di GGSN akan didekapsulasi untuk mendapatkan data aplikasi sesuai dengan persamaan (2.6):

$$\begin{aligned} W_{mobile\ TV(GGSN)} &= W_{frame\ PDN\ Internet} - header_{IP} - header_{UDP} \\ &\quad - header_{Ethernet} - FCS \\ &= 435\ byte - 40\ byte - 8\ byte - 14\ byte - 4\ byte \\ &= 369\ byte \end{aligned}$$

Agar diperoleh data yang asli, data yang diterima akan mengalami proses dekapsulasi. Apabila kecepatan transmisi pada GGSN adalah 100 Mbps, maka berdasarkan persamaan (2.7) didapatkan *delay* dekapsulasi sebesar:

$$\begin{aligned} t_{D(GGSN)} &= \frac{W_{mobile\ TV(GGSN)} \times 8}{V_{GGSN}} \\ &= \frac{(369 \times 8)bit}{100 \times 10^6 bps} \\ &= 2,95 \times 10^{-5} s \end{aligned}$$

Panjang *frame* GGSN yang siap ditransmisikan ke SGSN berdasarkan persamaan (2.8) adalah sebagai berikut:

$$\begin{aligned} W_{frame\ GGSN} &= W_{mobile\ TV(GGSN)} + header_{GTP} + header_{UDP} + header_{IP} \\ &\quad + header_{Ethernet} + FCS \\ &= 369\ byte + 8\ byte + 8\ byte + 40\ byte + 14\ byte + 4\ byte \\ &= 443\ byte \end{aligned}$$

Nilai *delay* enkapsulasi yang terjadi di GGSN berdasarkan persamaan (2.9) adalah:

$$\begin{aligned} t_{E(GGSN)} &= \frac{W_{frame\ GGSN} \times 8}{V_{GGSN}} \\ &= \frac{(443 \times 8)bit}{100 \times 10^6 bps} \\ &= 3,54 \times 10^{-5} s \end{aligned}$$

Data yang telah didekapsulasi akan *reassembly* lagi dalam *datagram* IP dan disegmentasi menjadi *frame Ethernet* untuk dirutekan ke jaringan EDGE atau HSDPA.

c) 2.75 SGSN

Setelah data sampai pada 2.75G SGSN, *frame ethernet* dari GGSN akan didekapsulasi sehingga diperoleh kembali *frame N-PDU*, dimana pada 2.75G SGSN disebut sebagai *frame network layer*. Untuk mendapatkan data aplikasi, frame tersebut didekapsulasi sesuai persamaan (2.10):

$$\begin{aligned} W_{\text{mobile TV (2.75G SGSN)}} &= W_{\text{frame GGSN}} - \text{header}_{\text{GTP}} - \text{header}_{\text{UDP}} \\ &\quad - \text{header}_{\text{IP}} - \text{header}_{\text{Ethernet}} - \text{FCS} \\ &= 443 \text{ byte} - 8 \text{ byte} - 8 \text{ byte} - 40 \text{ byte} - 14 \text{ byte} - 4 \text{ byte} \\ &= 369 \text{ byte} \end{aligned}$$

Delay dekapsulasi pada 2.75G SGSN yang mempunyai kecepatan 15 Mbps, sesuai dengan persamaan (2.11) yaitu:

$$\begin{aligned} t_{D(2.75G SGSN)} &= \frac{W_{\text{mobile TV (2.75G SGSN)}} \times 8}{V_{2.75G SGSN}} \\ &= \frac{(369 \times 8) \text{ bit}}{15 \times 10^6 \text{ bps}} \\ &= 1,968 \times 10^{-4} \text{ s} \end{aligned}$$

Frame N-PDU yang diperoleh dari proses dekapsulasi ini kemudian ditransfer dari Gn *interface* ke Gb *interface* oleh *relay*. Pada *layer SNDNP*, data aplikasi yang melebihi MSS SNDNP akan disegmentasi berdasarkan MSS SNDNP sebesar 1503 byte (standar ETSI). Apabila data aplikasi yang diterima tidak melebihi MSS SNDNP, maka *frame* tersebut dienkapsulasi dengan menambahkan *header SNDNP*, LLC, BSSGP, NS. (3GPP TS 29.060 Rel. 5) dan FCS pada *layer SNDNP* untuk deteksi *error* sebelum ditransmisikan ke BSC, sesuai dengan persamaan (2.12) yaitu:

$$\begin{aligned} W_{\text{frame 2.75G SGSN}} &= W_{\text{mobile TV (2.75G SGSN)}} + \text{header}_{\text{SNDNP}} + \text{FCS} + \text{header}_{\text{LLC}} \\ &\quad + \text{header}_{\text{BSSGP}} + \text{header}_{\text{NS}} \\ &= 369 \text{ byte} + 4 \text{ byte} + 3 \text{ byte} + 4 \text{ byte} + 48 \text{ byte} + 4 \text{ byte} \\ &= 432 \text{ byte} \end{aligned}$$

Besar *delay enkapsulasi* pada 2.75G SGSN berdasarkan persamaan (2.13) adalah:

$$\begin{aligned} t_E(2.75G SGSN) &= \frac{W_{\text{frame 2.75G SGSN}} \times 8}{V_{2.75G SGSN}} \\ &= \frac{(432 \times 8) \text{ bit}}{15 \times 10^6 \text{ bps}} \\ &= 2,304 \times 10^{-4} \text{ s} \end{aligned}$$

d) PCU (BSC)

Sampai di BSC, *frame 2.5G SGSN* didekapsulasi dengan cara melepaskan *header-header* yang ada pada proses enkapsulasi di 2.5 SGSN. Sehingga untuk mendapatkan data aplikasi diperoleh dengan persamaan (2.14) berikut :



$$\begin{aligned}
 W_{mobile TV(BSS)} &= W_{frame 2.5G SGSN} - header_{SNDCP} - FCS - header_{LLC} \\
 &\quad - header_{BSSGP} - header_{NS} \\
 &= 432 \text{ byte} - 4 \text{ byte} - 3 \text{ byte} - 4 \text{ byte} - 48 \text{ byte} - 4 \text{ byte} \\
 &= 369 \text{ byte}
 \end{aligned}$$

Pada BSS, kecepatan kearah Gb *interface* adalah kapasitas maksimum RPP per PCU (14 RPP) dikalikan dengan kecepatan standard E1-link (31x64 Kbps). Sehingga sesuai dengan persamaan (2.15), nilai *delay* dekapsulasi pada BSS adalah:

$$\begin{aligned}
 t_{D(PCU)} &= \frac{W_{mobile TV(PCU)} \times 8}{V_{1(PCU)}} \\
 &= \frac{(370 \times 8) \text{ bit}}{(14 \times 31 \times 64 \times 10^3) \text{ bps}} \\
 &= 1,0628 \times 10^{-4} \text{ s}
 \end{aligned}$$

Selanjutnya *frame* LLC disegmentasi menjadi RLC/MAC *radio block* untuk dapat ditransmisikan pada Abis *interface* menuju BTS dengan ukuran tetap yaitu 456 bit. Masing masing *Coding Scheme* memiliki nilai RLC data yang berbeda-beda sesuai dengan Tabel 2.4. Apabila digunakan MSC (*Modulation Coding Scheme*) 1, maka berdasarkan persamaan (2.16) jumlah *frame* RLC *radio block* adalah sebagai berikut:

$$\begin{aligned}
 N_{frame RLC block} &= \frac{W_{frame LLC} \times 8}{RLC_{data unit}} \\
 &= \frac{(380 \times 8) \text{ bit}}{(22 \times 8) \text{ bit}} \\
 &= 17,2727
 \end{aligned}$$

Dengan panjang *frame* LLC didapatkan dari persamaan (2.17), yaitu:

$$\begin{aligned}
 W_{frame LLC} &= W_{mobile TV(PCU)} + header_{SNDCP} + FCS + header_{LLC} \\
 &= 369 \text{ byte} + 4 \text{ byte} + 3 \text{ byte} + 4 \text{ byte} \\
 &= 380 \text{ byte}
 \end{aligned}$$

Setelah itu tiap segmennya akan dienkapsulasi dengan menambahkan :

RLC/MAC *radio block* = 3 bit USF + 176 bit RLC data + 40 bit BCS + 4 tail bit. Maka *delay* enkapsulasi yang terjadi di PCU adalah sebagai berikut :

$$\begin{aligned}
 t_{E(PCU)} &= \frac{\text{jumlah frame LLC} \times N_{frame RLC block} \times RLC_{radio block}}{V_{2(PCU)}} \\
 &= \frac{1 \times 17,2727 \times 456}{14 \times 150 \times 16 \times 10^3 \text{ bps}} \\
 &= 0,2344 \times 10^{-3} \text{ s}
 \end{aligned}$$

Besar *delay* proses yang terjadi merupakan penjumlahan dari *delay* enkapsulasi dan dekapsulasi yang terjadi di PCU. *Delay* proses PCU untuk MCS-1 adalah sebagai berikut:



$$\begin{aligned}
 t_{proc PCU} &= t_{D(PCU)} + t_{E(PCU)} \\
 &= 1,0628 \times 10^{-4} + 0,2344 \times 10^{-3} \\
 &= 0,3407 \times 10^{-3} \text{ s}
 \end{aligned}$$

Dengan cara yang sama maka akan diperoleh *delay* pada PCU dengan MCS yang berbeda, seperti pada Tabel 4.1.

Tabel 4.1 Hasil Perhitungan *Delay* Proses pada PCU

MCS	Delay Dekapsulasi pada PCU (s)	Delay Enkapsulasi pada PCU (s)	Delay Proses pada PCU (s)
MCS-1	$1,0628 \cdot 10^{-4}$	$0,2344 \cdot 10^{-3}$	$0,3407 \cdot 10^{-3}$
MCS-2	$1,0628 \cdot 10^{-4}$	$0,1842 \cdot 10^{-3}$	$0,2905 \cdot 10^{-3}$
MCS-3	$1,0628 \cdot 10^{-4}$	$0,1394 \cdot 10^{-3}$	$0,2457 \cdot 10^{-3}$
MCS-4	$1,0628 \cdot 10^{-4}$	$0,1172 \cdot 10^{-3}$	$0,2235 \cdot 10^{-3}$
MCS-5	$1,0628 \cdot 10^{-4}$	$0,0921 \cdot 10^{-3}$	$0,1984 \cdot 10^{-3}$
MCS-6	$1,0628 \cdot 10^{-4}$	$0,0697 \cdot 10^{-3}$	$0,1760 \cdot 10^{-3}$
MCS-7	$1,0628 \cdot 10^{-4}$	$0,0460 \cdot 10^{-3}$	$0,1523 \cdot 10^{-3}$
MCS-8	$1,0628 \cdot 10^{-4}$	$0,0379 \cdot 10^{-3}$	$0,1442 \cdot 10^{-3}$
MCS-9	$1,0628 \cdot 10^{-4}$	$0,0348 \cdot 10^{-3}$	$0,1411 \cdot 10^{-3}$

Sumber: Perhitungan

e) UE di Jaringan EDGE

RLC/MAC *radio block* yang diterima UE dari BTS melalui Um *Interface* selanjutnya akan mengalami proses dekapsulasi menjadi *frame* data aslinya, dengan *delay* dekapsulasi bergantung pada MCS sesuai Tabel 2.4 dan jumlah *timeslot* yang digunakan.

Untuk MCS-1, besar *delay* dekapsulasi sesuai persamaan (2.19) adalah:

$$\begin{aligned}
 t_D(2.75G \text{ UE}) &= \frac{\text{jumlah frame IP} \times \text{jumlah frame SNDCP} \times N_{\text{frame RLC block}} \times \frac{\text{RLC data}}{\text{code rate}}}{n \times V_{TS}} \\
 &= \frac{1 \times 1 \times 17,2727 \times \frac{176}{1/2}}{4 \times 8,9 \times 10^3} \\
 &= 0,1708 \text{ s}
 \end{aligned}$$

Dengan cara yang sama, diperoleh nilai *delay* dekapsulasi untuk MCS yang berbeda, seperti pada Tabel 4.2.

Tabel 4.2 Hasil Perhitungan *Delay Dekapsulasi* pada UE di EDGE

MCS	<i>Delay Dekapsulasi pada UE di EDGE (s)</i>
MCS-1	0,1708
MCS-2	0,1051
MCS-3	0,0615
MCS-4	0,0429
MCS-5	0,1013
MCS-6	0,0512
MCS-7	0,0225
MCS-8	0,0155
MCS-9	0,0128

Sumber: Perhitungan

f) 3.5G SGSN

Pada SGSN terjadi dekapsulasi terhadap data yang diterima dari GGSN. Paket data tersebut didekapsulasi sesuai dengan persamaan (2.20):

$$\begin{aligned}
 W_{\text{mobile TV}(3.5G \text{ SGSN})} &= W_{\text{frame GGSN}} - \text{header}_{\text{GTP}} - \text{header}_{\text{IP}} \\
 &\quad - \text{header}_{\text{UDP}} - \text{header}_{\text{Ethernet}} - \text{FCS} \\
 &= 443 \text{ byte} - 8 \text{ byte} - 40 \text{ byte} - 8 \text{ byte} - 14 \text{ byte} - 4 \text{ byte} \\
 &= 369 \text{ byte}
 \end{aligned}$$

Pada skripsi ini menggunakan 3.5G SGSN dengan teknologi *Fast Ethernet* dengan transmisi data sebesar 100 Mbps. Nilai *delay* dekapsulasi yang terjadi di 3.5G SGSN sesuai persamaan (2.21) adalah:

$$\begin{aligned}
 t_{D(3.5G \text{ SGSN})} &= \frac{W_{\text{mobile TV}(3.5G \text{ SGSN})} \times 8}{V_{3.5G \text{ SGSN}}} \\
 &= \frac{(369 \times 8) \text{ bit}}{100 \times 10^6 \text{ bps}} \\
 &= 2,952 \times 10^{-5} \text{ s}
 \end{aligned}$$

Selanjutnya paket data aplikasi *mobile TV* ditambahkan *header GTP*, UDP dan IP. Saat memasuki lapisan AAL5, paket data tersebut dibentuk menjadi CPCS PDU (*Common Part Convergence Sublayer Protocol Data Unit*) sesuai persamaan (2.22) yaitu:

$$\begin{aligned}
 W_{\text{frame AAL5}} &= W_{\text{mobile TV}(3.5G \text{ SGSN})} + \text{header}_{\text{GTP}} + \text{header}_{\text{IP}} + \text{header}_{\text{UDP}} \\
 &\quad + \text{header}_{\text{AAL5}} \\
 &= 369 \text{ byte} + 8 \text{ byte} + 40 \text{ byte} + 8 \text{ byte} + 8 \text{ byte} \\
 &= 433 \text{ byte}
 \end{aligned}$$

Pada layer ATM, frame AAL5 tersebut kemudian akan dipecah menjadi blok-blok *payload PDU SAR* (*Segmentation and Reassembly Sublayer*) sebesar 48 byte. Sehingga banyaknya sel ATM yang terbentuk sesuai persamaan (2.23) adalah:

$$N_{\text{frame ATM}} = \frac{W_{\text{frame AAL5}}}{48 \text{ byte}}$$



$$\begin{aligned}
 &= \frac{433 \text{ byte}}{48 \text{ byte}} \\
 &= 9,0208
 \end{aligned}$$

Terdapat 9 *frame* berisi data 48 *byte* dan 1 *frame* berisi data sebesar $(0,0208 \times 48 \text{ byte} = 0,9984 \approx 1)$. *Frame* yang berisi data 1 *byte* tersebut akan diberi *padding bytes* sebesar $(48 \text{ byte} - 1 \text{ byte} = 47 \text{ byte})$, sehingga terdapat 10 sel ATM. Selanjutnya setiap sel ATM diberi *header ATM* sebesar 5 *byte*, sehingga panjang *frame ATM* menjadi 53 *byte*. Panjang *frame* di SGSN yang siap ditransmisikan menuju RNC merupakan panjang *frame ATM* total sesuai persamaan (2.24) yaitu:

$$\begin{aligned}
 W_{\text{frame } 3.5GSGSN} &= W_{\text{frame ATM total}} = N_{\text{frame ATM}} \times W_{\text{frame ATM}} \\
 &= 10 \times 53 \text{ byte} \\
 &= 530 \text{ byte}
 \end{aligned}$$

Sehingga *delay enkapsulasi* yang terjadi di 3.5G SGSN dapat dihitung dengan persamaan (2.25):

$$\begin{aligned}
 t_{E(3.5G SGSN)} &= \frac{W_{\text{frame } 3.5GSGSN} \times 8}{V_{SGSN-RNS}} \\
 &= \frac{(530 \times 8) \text{ bit}}{155,52 \times 10^6 \text{ bps}} \\
 &= 2,7263 \times 10^{-5} \text{ s}
 \end{aligned}$$

g) RNC

Pada RNC paket yang diterima dari 3.5G SGSN akan mengalami proses dekapsulasi. *Frame ATM* total didekapsulasi untuk mendapatkan data aplikasi sesuai dengan persamaan (2.26):

$$\begin{aligned}
 W_{\text{mobile TV (RNC)}} &= W_{\text{frame } 3.5GSGSN} - \text{header}_{\text{ATM}} - \text{header}_{\text{GTP}} \\
 &\quad - \text{header}_{\text{IP}} - \text{header}_{\text{UDP}} - \text{header}_{\text{AAL5}} \\
 &= 530 \text{ byte} - 5 \text{ byte} - 8 \text{ byte} \\
 &\quad - 40 \text{ byte} - 8 \text{ byte} - 8 \text{ byte} \\
 &= 461 \text{ byte}
 \end{aligned}$$

RNC menggunakan *interface STM-1* dengan kecepatan transmisi 155,52 Mbps. Sehingga besar *delay dekapsulasi* yang terjadi di RNC didapatkan dengan persamaan (2.27) berikut:

$$\begin{aligned}
 t_{D(RNC)} &= \frac{W_{\text{mobile TV(RNC)}} \times 8}{V_{\text{RNC}}} \\
 &= \frac{(461 \times 8) \text{ bit}}{155,52 \times 10^6 \text{ bps}} \\
 &= 2,3714 \times 10^{-5} \text{ s}
 \end{aligned}$$



Selanjutnya paket data aplikasi tersebut dienkapsulasi pada *layer PDCP* (*Packet Data Convergence Protocol*) dengan diberi *header PDCP* sebesar 1 byte, sesuai persamaan (2.28):

$$\begin{aligned} W_{frame \ PDCP} &= W_{mobile \ TV(RNC)} + header_{PDCP} \\ &= 461 \text{ byte} + 1 \text{ byte} \\ &= 462 \text{ byte} \end{aligned}$$

Pada *layer RLC*, *frame PDCP* akan disegmentasi menjadi RLC PDU *fixed size* yaitu sebesar 40 byte, sehingga sesuai dengan persamaan (2.29) jumlah *frame RLC* yaitu:

$$\begin{aligned} N_{frame \ RLC} &= \frac{W_{frame \ PDCP}}{40 \text{ byte}} \\ &= \frac{462 \text{ byte}}{40 \text{ byte}} \\ &= 11,55 \end{aligned}$$

Terdapat 11 *frame RLC* berisi data 40 byte dan 1 *frame* berisi data sebesar $(0,55 \times 40 \text{ byte} = 22 \text{ byte})$. *Frame* yang berisi data 22 byte tersebut akan diberi *padding bytes* sebesar $(40 \text{ byte} - 22 \text{ byte} = 18 \text{ byte})$, sehingga terdapat 12 *frame RLC*. Setiap frame kemudian diberi *header* sebesar 2 byte sehingga panjang *frame* menjadi 42 byte. Panjang *frame RLC* total ditransmisikan menuju *layer MAC-d* sesuai persamaan (2.30) yaitu:

$$\begin{aligned} W_{frame \ RLC \ total} &= N_{frame \ RLC} \times W_{frame \ RLC} \\ &= 12 \times 42 \text{ byte} \\ &= 504 \text{ byte} \end{aligned}$$

Selanjutnya pada *layer MAC-d*, RLC PDU akan disegmentasi menjadi MAC-d SDU (*Service Data Unit*) sebesar 42 byte. Maka jumlah *frame MAC-d SDU* sesuai dengan persamaan (2.31) yaitu:

$$\begin{aligned} N_{frame \ MAC-d} &= \frac{W_{frame \ RLC \ total}}{42 \text{ byte}} \\ &= \frac{504 \text{ byte}}{42 \text{ byte}} \\ &= 12 \end{aligned}$$

Sedangkan panjang *frame MAC-d* dapat dihitung berdasarkan persamaan (2.32):

$$\begin{aligned} W_{frame \ MAC-d} &= header_{MAC} + 42 \text{ byte} \\ &= 3 \text{ byte} + 42 \text{ byte} \\ &= 45 \text{ byte} \end{aligned}$$

Pada HS-DSCH FP (*Frame Protocol*), *frame MAC-d* dienkapsulasi dengan menambahkan *header FP* dan *CRC* sesuai persamaan (2.33):



$$\begin{aligned}
 W_{frame\ FP} &= W_{frame\ MAC-d} + header_{FP} + CRC \\
 &= 45\ byte + 7\ byte + 2\ byte \\
 &= 54\ byte
 \end{aligned}$$

Sehingga panjang *frame* HS-DSCH FP total didapatkan dengan persamaan (2.34):

$$\begin{aligned}
 W_{frame\ FP\ total} &= N_{frame\ MAC-d} \times W_{frame\ FP} \\
 &= 12 \times 54\ byte \\
 &= 648\ byte
 \end{aligned}$$

Saat memasuki *layer* AAL2, paket data dibentuk menjadi paket CPS (*Common Part Sublayer*) dengan penambahan *header* sebesar 3 byte, sehingga sesuai persamaan (2.35) panjang *frame* menjadi:

$$\begin{aligned}
 W_{frame\ AAL2} &= W_{frame\ FP\ total} + header_{AAL2} \\
 &= 648\ byte + 3\ byte \\
 &= 651\ byte
 \end{aligned}$$

Pada *layer* ATM, *frame* AAL2 tersebut kemudian akan dipecah menjadi blok-blok *payload* PDU SAR (*Segmentation and Reassembly Sublayer*) sebesar 48 byte. Masing-masing PDU SAR 48 byte sesuai dengan sel ATM tunggal. Sehingga banyaknya sel ATM yang terbentuk sesuai persamaan (2.36):

$$\begin{aligned}
 N_{frame\ ATM2} &= \frac{W_{frame\ AAL2}}{48\ byte} \\
 &= \frac{651\ byte}{48\ byte} \\
 &= 13,5625
 \end{aligned}$$

Terdapat 13 *frame* berisi data 48 byte dan 1 *frame* berisi data sebesar $(0,5625 \times 48\ byte = 27\ byte)$. *Frame* yang berisi data 27 byte tersebut akan diberi *padding bytes* sebesar $(48\ byte - 27\ byte = 21\ byte)$, sehingga terdapat 14 *frame*. Selanjutnya setiap sel ATM ditambahkan dengan *header* ATM sebesar 5 byte, sehingga panjang *frame* ATM menjadi 53 byte. Panjang *frame* di RNC yang siap ditransmisikan menuju UE merupakan panjang *frame* ATM total sesuai persamaan (2.37) yaitu:

$$\begin{aligned}
 W_{frame\ RNC} &= W_{frame\ ATM2\ total} = N_{frame\ ATM2} \times W_{frame\ ATM} \\
 &= 14 \times 53\ byte \\
 &= 742\ byte
 \end{aligned}$$

Sehingga besarnya *delay* enkapsulasi yang terjadi di RNC diberikan oleh persamaan (2.38):

$$t_{E(RNC)} = \frac{W_{frame\ RNC} \times 8}{V_{RNC}}$$



$$\begin{aligned}
 &= \frac{(742 \times 8)bit}{155,52 \times 10^6 bps} \\
 &= 3,8169 \times 10^{-5}s
 \end{aligned}$$

h) Node B

Pada Node B paket data yang diterima dari RNC didekapsulasi sesuai dengan persamaan (2.39):

$$\begin{aligned}
 W_{mobile TV(Node B)} &= W_{frame RNC} - header_{ATM} - header_{AAL2} - header_{FP} - CRC \\
 &= 742 byte - 5 byte - 3 byte - 7 byte - 2 byte \\
 &= 725 byte
 \end{aligned}$$

Node B menggunakan *interface* sistem transmisi STM-1 dengan kecepatan 155,52 Mbps. Besarnya *delay* dekapsulasi yang terjadi di Node B sesuai dengan persamaan (2.40):

$$\begin{aligned}
 t_{D(Node B)} &= \frac{W_{mobile TV(Node B)} \times 8}{V_{Node B}} \\
 &= \frac{(725 \times 8)bit}{155,52 \times 10^6 bps} \\
 &= 3,7294 \times 10^{-5}s
 \end{aligned}$$

Pada saat memasuki *layer* MAC-hs, paket data disegmentasi menjadi paket sebesar 42 byte, sehingga jumlah *frame* MAC-hs sesuai dengan persamaan (2.41) yaitu:

$$\begin{aligned}
 N_{frame MAC-hs} &= \frac{W_{mobile TV(Node B)}}{42 byte} \\
 &= \frac{725 byte}{42 byte} \\
 &= 17,2619
 \end{aligned}$$

Terdapat 17 *frame* berisi data 42 byte dan 1 *frame* berisi data sebesar ($0,2619 \times 42 byte = 11 byte$). *Frame* yang berisi data 11 byte tersebut akan diberi *padding bytes* sebesar ($42 byte - 11 byte = 31 byte$), sehingga terdapat 18 *frame*. Panjang *frame* MAC-hs dapat dihitung berdasarkan persamaan (2.42):

$$\begin{aligned}
 W_{frame MAC-hs} &= header_{MAC} + 42byte \\
 &= 3 byte + 42 byte \\
 &= 45 byte
 \end{aligned}$$

Sehingga sesuai persamaan (2.43), panjang *frame* total pada Node B yaitu:

$$\begin{aligned}
 W_{frame Node B} &= N_{frame MAC-hs} \times W_{frame MAC-hs} \\
 &= 18 \times 45 byte \\
 &= 810 byte
 \end{aligned}$$

Dengan menggunakan persamaan (2.44) didapatkan *delay* enkapsulasi yang terjadi di Node B sebesar:



$$\begin{aligned}
 t_{E(Node\ B)} &= \frac{W_{frame\ Node\ B} \times 8}{V_{Node\ B}} \\
 &= \frac{(810 \times 8)bit}{155,52 \times 10^6 bps} \\
 &= 4,1667 \times 10^{-5} s
 \end{aligned}$$

i) UE HSDPA

Pada UE, paket data dari Node B didekapsulasi untuk mendapatkan data aplikasi *mobile TV*, sesuai persamaan (2.45):

$$\begin{aligned}
 W_{mobile\ TV(3.5G\ UE)} &= W_{frame\ NodeB} - header_{PDCP} - header_{RLC} - header_{MAC} \\
 &= 810\ byte - 1\ byte - 2\ byte - 3\ byte \\
 &= 804\ byte
 \end{aligned}$$

Dengan mengasumsikan bahwa terminal pengguna adalah terminal kategori 5 sesuai dengan Tabel 2.2, dengan *data rate* maksimum sebesar 3,6 Mbps, maka besar *delay* dekapsulasi yang terjadi di UE dapat dihitung menggunakan persamaan (2.46):

$$\begin{aligned}
 t_{D(3.5G\ UE)} &= \frac{W_{mobile\ TV(3.5G\ UE)} \times 8}{V_{UE}} \\
 &= \frac{(804 \times 8)bit}{3,6 \times 10^6 bps} \\
 &= 0,0018 s
 \end{aligned}$$

j) Delay Proses dari PDN Internet ke UE pada Jaringan EDGE

Pada jaringan EDGE, *delay* enkapsulasi untuk MCS-1 dari PDN *Internet* ke UE sesuai persamaan (2.47) adalah sebagai berikut:

$$\begin{aligned}
 t_{E(EDGE)total} &= t_{E(PDN\ Internet)} + t_{E(GGSN)} + t_{E(2.75G\ SGSN)} + t_{E(PCU)} \\
 &= 3,48 \times 10^{-6} + 3,54 \times 10^{-6} + 2,304 \times 10^{-4} + 0,2344 \times 10^{-3} \\
 &= 0,5073 \times 10^{-3} s
 \end{aligned}$$

Sedangkan *delay* dekapsulasi untuk MCS-1 dari PDN *Internet* ke UE sesuai persamaan (2.48) adalah sebagai berikut:

$$\begin{aligned}
 t_{D(EDGE)total} &= t_{D(GGSN)} + t_{D(2.75G\ SGSN)} + t_{D(PCU)} + t_{D(2.75G\ UE)} \\
 &= 2,95 \times 10^{-6} + 1,968 \times 10^{-4} + 1,0628 \times 10^{-4} + 0,1680 \\
 &= 0,1708 s
 \end{aligned}$$

Sehingga besar *delay* proses dari PDN *Internet* ke *User Equipment* pada jaringan EDGE sesuai persamaan (2.49) adalah:

$$\begin{aligned}
 t_{proc\ EDGE} &= t_{E(EDGE)total} + t_{D(EDGE)total} \\
 &= 0,5210 \times 10^{-3} + 0,1683 \\
 &= 0,1716 s
 \end{aligned}$$

Dengan cara perhitungan yang sama, maka diperoleh besar *delay* proses pada jaringan EDGE untuk MCS-2 sampai MCS-9 dengan ukuran data 369 byte, seperti yang ditampilkan dalam Tabel 4.3.

Pada setiap MCS mempunyai *data rate* yang berbeda, yang akan semakin besar dari MCS-1 hingga MCS-9. Pada perhitungan *delay* enkapsulasi dan dekapsulasi, besar *delay* yang terjadi berbanding terbalik dengan *data rate* yang digunakan, sehingga semakin besar MCS yang digunakan, maka *delay* proses semakin kecil. Dari tabel hasil perhitungan *delay* proses di atas, nilai *delay* proses terbesar terjadi pada MCS-1 dengan nilai *delay* 0,1716 s. Sedangkan nilai *delay* proses terkecil adalah pada MCS-9 yaitu sebesar 0,0134 s.

Tabel 4.3 Hasil Perhitungan *Delay* Proses pada Jaringan EDGE

MCS	<i>Delay</i> Enkapsulasi (s)	<i>Delay</i> Dekapsulasi (s)	<i>Delay</i> Proses (s)
MCS-1	$0,5037 \cdot 10^{-3}$	0,1711	0,1716
MCS-2	$0,4535 \cdot 10^{-3}$	0,1054	0,1059
MCS-3	$0,4087 \cdot 10^{-3}$	0,0618	0,0622
MCS-4	$0,3865 \cdot 10^{-3}$	0,0433	0,0437
MCS-5	$0,3614 \cdot 10^{-3}$	0,1017	0,1020
MCS-6	$0,3390 \cdot 10^{-3}$	0,0515	0,0519
MCS-7	$0,3154 \cdot 10^{-3}$	0,0229	0,0232
MCS-8	$0,3072 \cdot 10^{-3}$	0,0158	0,0161
MCS-9	$0,3042 \cdot 10^{-3}$	0,0131	0,0134

Sumber: Perhitungan

k) *Delay* Proses dari PDN Internet ke UE pada Jaringan HSDPA

Pada jaringan HSDPA, *delay* enkapsulasi pada aliran data dari PDN *Internet* ke UE sesuai persamaan (2.50) adalah sebagai berikut:

$$\begin{aligned} t_{E(HSDPA)total} &= t_{E(PDN\ Internet)} + t_{E(GGSN)} + t_{E(3.5G\ SGSN)} + t_{E(RNC)} + t_{E(Node\ B)} \\ &= 3,48 \times 10^{-6} + 3,54 \times 10^{-6} + 2,7263 \times 10^{-5} + 4,1667 \times 10^{-5} \\ &= 7,5954 \times 10^{-5} \text{ s} \end{aligned}$$

Sedangkan *delay* dekapsulasi yang terjadi sesuai dengan persamaan (2.51) adalah sebagai berikut:

$$\begin{aligned} t_{D(HSDPA)total} &= t_{D(GGSN)} + t_{D(3.5G\ SGSN)} + t_{D(RNC)} + t_{D(Node\ B)} + t_{D(3.5G\ UE)} \\ &= 3,54 \times 10^{-6} + 2,952 \times 10^{-5} + 2,3714 \times 10^{-5} + 3,7294 \times 10^{-5} + 0,0018 \\ &= 0,0019 \text{ s} \end{aligned}$$

Sehingga besar *delay* proses dari PDN *Internet* ke *User Equipment* pada jaringan HSDPA sesuai persamaan (2.52) adalah:

$$\begin{aligned} t_{proc\ HSDPA} &= t_{E(HSDPA)total} + t_{D(HSDPA)total} \\ &= 7,5954 \times 10^{-5} + 0,0019 \\ &= 0,0020 \text{ s} \end{aligned}$$

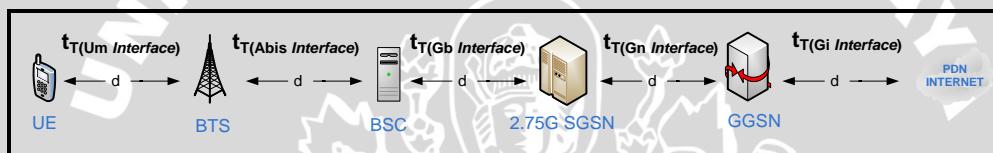
Dari hasil perhitungan dapat diketahui bahwa *delay* proses pada jaringan HSDPA lebih kecil dibandingkan *delay* proses pada jaringan EDGE. Hal ini



disebabkan karena kecepatan transmisi data pada jaringan HSDPA lebih besar dari kecepatan transmisi data pada jaringan EDGE.

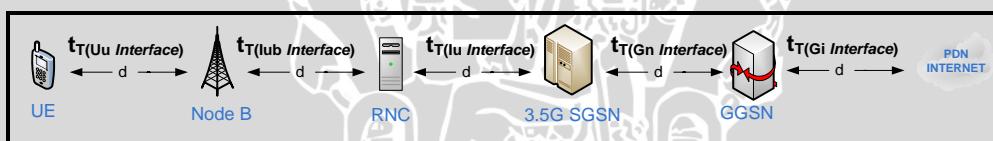
4.3.1.2 Delay Transmisi

Delay transmisi adalah waktu yang dibutuhkan untuk mentransmisikan bit pertama dari paket data hingga terakhir melalui media transmisi tertentu. Dalam analisis delay transmisi ini dilakukan perhitungan semua jalur *direct link*, seperti pada Gambar 4.4 dan Gambar 4.5. Jalur tersebut meliputi PDN *Internet* – GGSN (*Gi interface*), GGSN – 2.75G SGSN (*Gn interface*), 2.75G SGSN – BSC (*Gb interface*), BSC – BTS (*Abis interface*), dan BTS – 2.75G UE (*Um interface*), untuk jaringan EDGE, serta GGSN – 3.5G SGSN (*Gn interface*), 3.5G SGSN – RNC (*Iu interface*), RNC – Node B (*Iub interface*), dan Node B – 3.5G UE (*Uu interface*), untuk jaringan HSDPA.



Gambar 4.4 Arsitektur *Node-to-node Delay* Transmisi pada Jaringan EDGE

Sumber : Perancangan



Gambar 4.5 Arsitektur *Node-to-node Delay* Transmisi pada Jaringan HSDPA

Sumber : Perancangan

a) PDN Internet – GGSN (*Gi Interface*)

Jika PDN *Internet* dengan GGSN dihubungkan dengan jaringan standar *Ethernet* yang memiliki kecepatan transmisi data 1 Gbps, maka *delay* transmisi yang terjadi pada *Gi interface* adalah sesuai dengan persamaan (2.54):

$$\begin{aligned} t_{T(Gi \text{ Interface})} &= \frac{W_{framePDN \text{ Internet}} \times 8}{V_{Ethernet}} \\ &= \frac{(435 \times 8)bit}{1 \times 10^9 bps} \\ &= 3,48 \times 10^{-6} s \end{aligned}$$

b) GGSN – 2.75G SGSN (*Gn Interface*)

Jika hubungan antara GGSN dan 2.75G SGSN menggunakan *Ethernet* standard 100Base-Tx dengan kecepatan transmisi data sebesar 100 Mbps, maka *delay* transmisi yang terjadi pada *Gn Interface* adalah sesuai persamaan (2.55):



$$\begin{aligned}
 t_{T(Gn\ Interface)} &= \frac{W_{frameGGSN} \times 8}{V_{Gn\ Interface}} \\
 &= \frac{(443 \times 8)bit}{100 \times 10^6 bps} \\
 &= 3,544 \times 10^{-5} s
 \end{aligned}$$

c) 2.75G SGSN – BSC (Gb Interface)

Hubungan antara 2.5G SGSN dan BSC menggunakan standar E1-Link dengan kecepatan 2,048 Mbps, sehingga *delay* transmisi di Gb Interface sesuai persamaan (2.56) adalah:

$$\begin{aligned}
 t_{T(Gb\ Interface)} &= \frac{W_{frame\ 2.75G\ SGSN} \times 8}{V_{Gb\ Interface}} \\
 &= \frac{(432 \times 8)bit}{2,048 \times 10^6 bps} \\
 &= 0,0017 s
 \end{aligned}$$

d) BSC – BTS (Abis Interface)

Hubungan antara BSC dan BTS menggunakan standar 150 GSL dengan kecepatan 16 kbps. Besar *delay* transmisi pada Abis Interface sesuai persamaan (2.57) adalah:

$$\begin{aligned}
 t_{T(Abis\ Interface)} &= \frac{N_{frame\ RLC\ block} \times RLC_{radio\ block}}{V_{Abis\ Interface}} \\
 &= \frac{17,2727 \times 456}{150 \times 16 \times 10^3 bps} \\
 &= 0,0033 s
 \end{aligned}$$

Dengan cara yang sama, didapatkan nilai *delay* transmisi pada Abis Interface untuk MCS-2 sampai MCS-9 dengan hasil seperti pada Tabel 4.4.

Tabel 4.4 Hasil Perhitungan *Delay* Transmisi pada Abis Interface

MCS	Delay Transmisi (s)
MCS-1	0,0033
MCS-2	0,0026
MCS-3	0,0020
MCS-4	0,0016
MCS-5	0,0013
MCS-6	0,0010
MCS-7	0,0006
MCS-8	0,0005
MCS-9	0,0005

Sumber: Perhitungan

e) BTS – 2.75G UE (Um Interface)

Hubungan antara BTS dan UE menggunakan MCS-1 dengan *bit rate* 8,9 kbps serta menggunakan 4 timeslot. Sehingga *delay* transmisi pada Um Interface didapat dengan persamaan (2.58).



$$\begin{aligned}
 t_{T(Um\ Interface)} &= \frac{N_{frame\ RLC\ block} \times RLC_{radio\ block}}{n \times V_{MCS}} \\
 &= \frac{17,2727 \times 456}{4 \times 8,9 \times 10^3 bps} \\
 &= 0,2212 s
 \end{aligned}$$

Dengan cara yang sama, didapatkan nilai *delay* transmisi pada Um *Interface* untuk MCS-2 sampai MCS-9 dengan hasil seperti pada Tabel 4.5.

Tabel 4.5 Hasil Perhitungan *Delay* Transmisi pada Um *Interface*

MCS	Delay Transmisi (s)
MCS-1	0,2212
MCS-2	0,1369
MCS-3	0,0786
MCS-4	0,0556
MCS-5	0,0344
MCS-6	0,0197
MCS-7	0,0086
MCS-8	0,0058
MCS-9	0,0049

Sumber: Perhitungan

f) GGSN – 3.5G SGSN (*Gn Interface*)

GGSN dengan 3.5G SGSN dihubungkan dengan menggunakan *Fast Ethernet* yang memiliki kecepatan transmisi data sebesar 100 Mbps. Sehingga *delay* transmisi yang terjadi pada Gn *interface* adalah sesuai dengan persamaan (2.59):

$$\begin{aligned}
 t_{T(Gn\ Interface)} &= \frac{W_{frame\ GGSN} \times 8}{V_{Gn\ Interface}} \\
 &= \frac{(443 \times 8)bit}{100 \times 10^6 bps} \\
 &= 3,544 \times 10^{-5} s
 \end{aligned}$$

g) 3.5G SGSN – RNC (*Iu Interface*)

Iu *interface* yang menghubungkan antara SGSN dan RNC menggunakan sistem transmisi digital STM-1 yang memberikan kecepatan data sebesar 155,52 Mbps. Sehingga nilai *delay* transmisi didapatkan dengan persamaan (2.60):

$$\begin{aligned}
 t_{T(Iu\ Interface)} &= \frac{W_{frame\ 3.5\ G\ SGSN} \times 8}{V_{Iu\ Interface}} \\
 &= \frac{(530 \times 8)bit}{155,52 \times 10^6 bps} \\
 &= 2,7263 \times 10^{-5} s
 \end{aligned}$$

h) RNC – Node B (*Iub Interface*)

Hubungan antara RNC dan Node B menggunakan *interface* sistem trammisi digital STM-1 yang memberikan kecepatan data sebesar 155,52 Mbps. Sehingga nilai *delay* transmisi yang terjadi sesuai dengan persamaan (2.61):

$$\begin{aligned}
 t_{T(Iub\ Interface)} &= \frac{W_{frame\ RNC} \times 8}{V_{Iub\ Interface}} \\
 &= \frac{(742 \times 8)bit}{155,52 \times 10^6 bps} \\
 &= 3,8169 \times 10^{-5} s
 \end{aligned}$$

i) Node B – 3.5G UE (*Uu Interface*)

Antara *Node B* dan *UE*, *delay transmisi* yang terjadi bergantung pada *data rate* *UE* serta jumlah slot. *UE* diasumsikan memiliki *data rate* maksimum sebesar 3,6 Mbps. Sesuai persamaan (2.63) didapatkan jumlah slot pada HSDPA adalah sebanyak 3 slot. Sehingga *delay transmisi* mengacu pada persamaan (2.62) adalah sebesar:

$$\begin{aligned}
 t_{T(Uu\ Interface)} &= \frac{W_{frame\ Node\ B} \times 8}{n \times V_{UE}} \\
 &= \frac{(810 \times 8)bit}{3 \times 3,6 \times 10^6 bps} \\
 &= 6 \times 10^{-4} s
 \end{aligned}$$

j) Delay Transmisi Total pada Jaringan EDGE

Delay transmisi total pada jaringan EDGE merupakan nilai total dari *delay transmisi* pada tiap *interface*. Pada MCS-1 nilai *delay transmisi total* ditentukan berdasarkan persamaan (2.64):

$$\begin{aligned}
 t_{T(EDGE)total} &= t_{T(Gi\ Interface)} + t_{T(Gn\ Interface)} + t_{T(Gb\ Interface)} + t_{T(Abis\ Interface)} \\
 &\quad + t_{T(Um\ Interface)} \\
 &= 3,48 \times 10^{-6} + 3,544 \times 10^{-5} + 0,0017 + 0,0033 + 0,2212 \\
 &= 0,2263 s
 \end{aligned}$$

Dengan cara perhitungan yang sama, nilai *delay transmisi* untuk masing-masing MCS ditunjukkan pada Tabel 4.6.

Hasil perhitungan *delay transmisi total* menunjukkan bahwa besar *delay transmisi* dengan ukuran data aplikasi yang sama, akan berubah nilainya sesuai dengan MCS yang digunakan. Penggunaan MCS-1 menghasilkan *delay transmisi* terbesar, karena memiliki *information field* yang terkecil. Selain itu MCS-1 memiliki *data rate* yang terendah yaitu 8,9 Kbps. Hal ini sesuai dengan persamaan (2.58) dimana *delay transmisi* berbanding lurus dengan *information field* dan berbanding terbalik dengan *data rate*.



Tabel 4.6 Hasil Perhitungan *Delay Transmisi Total* pada EDGE

MCS	<i>Delay Transmisi Total</i> (s)
MCS-1	0,2263
MCS-2	0,1412
MCS-3	0,0823
MCS-4	0,0590
MCS-5	0,0374
MCS-6	0,0224
MCS-7	0,0110
MCS-8	0,0081
MCS-9	0,0071

Sumber: Perhitungan

k) *Delay Transmisi Total* pada Jaringan HSDPA

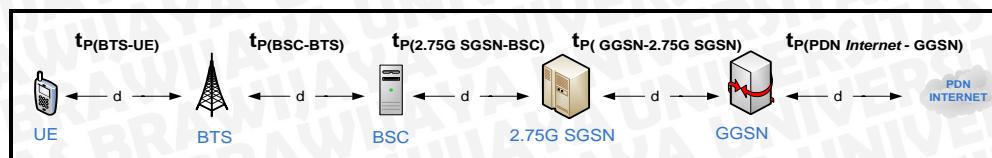
Delay transmisi total pada jaringan HSDPA merupakan nilai total dari *delay transmisi* pada tiap *interface*, sehingga ditentukan berdasarkan persamaan (2.65):

$$\begin{aligned}
 t_{T(HSDPA)\text{total}} &= t_{T(Gi\ Interface)} + t_{T(Gn\ Interface)} + t_{T(Iu\ Interface)} + t_{T(Iub\ Interface)} \\
 &\quad + t_{T(Uu\ Interface)} \\
 &= 3,48 \times 10^{-6} + 3,544 \times 10^{-5} + 2,7263 \times 10^{-5} + 3,8169 \times 10^{-5} + 6 \times 10^{-4} \\
 &= 7,0435 \times 10^{-4} \text{ s}
 \end{aligned}$$

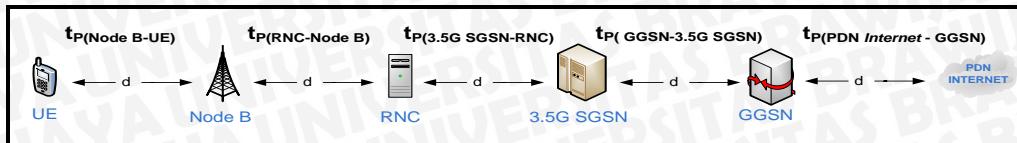
Hasil perhitungan menunjukkan bahwa *delay transmisi* pada jaringan HSDPA lebih kecil dibandingkan dengan *delay transmisi* pada jaringan EDGE. Hal ini disebabkan kecepatan transmisi pada jaringan HSDPA yang lebih besar dari kecepatan transmisi pada jaringan EDGE.

4.3.1.3 *Delay Propagasi*

Dalam analisis ini *delay propagasi* dihitung pada tiap *node*. Pada jaringan EDGE, *delay propagasi* terjadi pada hubungan PDN *Internet*-GGSN-2.5G SGSN-BSC-BTS-UE, seperti ditunjukkan pada Gambar 4.6. Sedangkan pada jaringan WCDMA *delay propagasi* terjadi pada PDN *Internet*-GGSN-3G SGSN-RNC-Node B-UE, yang ditunjukkan pada Gambar 4.7. Antara PDN *Internet* dengan Node B menggunakan media *fiber optic*, serta antara Node B dengan UE dan BTS dengan UE menggunakan media udara.

**Gambar 4.6** Arsitektur *Node-to-node Delay Propagasi* pada Jaringan EDGE

Sumber : Perancangan



Gambar 4.7 Arsitektur *Node-to-node Delay Propagasi* pada Jaringan HSDPA

Sumber : Perancangan

a) *Delay Propagasi* dari PDN *Internet* ke GGSN

Dalam skripsi ini jaringan antara PDN *Internet* dan GGSN merupakan sebuah jaringan *Ethernet* dengan kecepatan 1 Gbps dan menggunakan media transfer *fiber optic* dengan jarak 15000 m. Sehingga *delay* propagasi antara PDN *Internet* dan GGSN didapatkan dengan persamaan (2.66):

$$\begin{aligned} t_{P(PDN\ Internet\ -GGSN)} &= \frac{N_{frame\ PDN\ Internet} \times R}{v} \\ &= \frac{1 \times 15000}{2 \times 10^8} \\ &= 7,5 \times 10^{-5}\ s \end{aligned}$$

b) *Delay Propagasi* dari GGSN ke 2.75G SGSN

Pada GGSN ke 2.75G SGSN dihubungkan dengan media transmisi *fiber optic* dengan jarak antara GGSN ke 2.75G SGSN diasumsikan 1000 m. Nilai *delay* propagasi antara GGSN ke 2.75G SGSN didapatkan dengan persamaan (2.66):

$$\begin{aligned} t_{P(GGSN\ -2.75G\ SGSN)} &= \frac{N_{frame\ GGSN} \times R}{v} \\ &= \frac{1 \times 1000}{2 \times 10^8} \\ &= 5 \times 10^{-6}\ s \end{aligned}$$

c) *Delay Propagasi* dari 2.75G SGSN ke BSC

Pada 2.75G SGSN dan BSC dihubungkan oleh media transmisi *fiber optic* dengan jarak antara 2.75G SGSN dan BSC adalah 25 m. Nilai *delay* propagasi antara 2.75G SGSN dan RNC didapatkan dengan persamaan (2.66) :

$$\begin{aligned} t_{P(2.75G\ SGSN\ -BSC)} &= \frac{N_{frame\ NS} \times R}{v} \\ &= \frac{1 \times 25}{2 \times 10^8} \\ &= 1,255 \times 10^{-7}\ s \end{aligned}$$

d) Delay Propagasi dari BSC ke BTS

Pada koneksi jaringan BSC dan BTS dihubungkan dengan media transmisi *fiber optic* dengan jarak 50 m. Nilai *delay* propagasi antara BSC dan BTS untuk MCS-1 didapatkan dengan persamaan (2.66) :

$$\begin{aligned} t_{P(BSC-BTS)} &= \frac{N_{\text{frame RLC radioblok}} \times R}{v} \\ &= \frac{17,2727 \times 50}{2 \times 10^8} \\ &= 0,4318 \times 10^{-5} \text{ s} \end{aligned}$$

Dengan cara perhitungan yang sama maka didapatkan *delay* propagasi untuk MSC-2 sampai MCS-9 yang ditunjukkan pada Tabel 4.7.

Dari hasil perhitungan *delay* propagasi dari BSC ke BTS didapatkan nilai *delay* propagasi yang berbeda. Hal ini disebabkan jumlah RLC *radio block* yang berbeda pada tiap MCS. Nilai *delay* propagasi terbesar terdapat pada MCS-1, dan *delay* propagasi terkecil terdapat pada MCS-9. Dengan demikian semakin besar jumlah RLC *radio block*, maka nilai *delay* propagasi semakin kecil.

Tabel 4.7 Hasil Perhitungan *Delay* Propagasi BSC ke BTS

MCS	<i>Delay</i> Propagasi (s)
MCS-1	$0,4318 \cdot 10^{-5}$
MCS-2	$0,3393 \cdot 10^{-5}$
MCS-3	$0,2568 \cdot 10^{-5}$
MCS-4	$0,2159 \cdot 10^{-5}$
MCS-5	$0,1696 \cdot 10^{-5}$
MCS-6	$0,1284 \cdot 10^{-5}$
MCS-7	$0,0848 \cdot 10^{-5}$
MCS-8	$0,0699 \cdot 10^{-5}$
MCS-9	$0,0642 \cdot 10^{-5}$

Sumber: Perhitungan

e) Delay Propagasi dari BTS ke UE

Pada skripsi ini BTS dan UE terhubung dengan media udara dengan asumsi bahwa keduanya memiliki jarak 100 m, maka berdasarkan persamaan (2.66) *delay* propagasi antara BTS dan UE adalah sebagai berikut :

$$\begin{aligned} t_{P(BTS-UE)} &= \frac{N_{\text{frame SNDCP}} \times R}{v} \\ &= \frac{17,2727 \times 100}{3 \times 10^8} \\ &= 0,5758 \times 10^{-5} \text{ s} \end{aligned}$$

Dengan cara perhitungan yang sama maka didapatkan *delay* propagasi untuk MSC-2 sampai MCS-9 yang ditunjukkan pada Tabel 4.8.



Dari hasil perhitungan *delay* propagasi dari BTS ke UE, didapatkan analisis bahwa semakin besar MCS, maka nilai *delay* propagasi semakin kecil, karena jumlah RLC *radio block* yang semakin besar.

Tabel 4.8 Hasil Perhitungan *Delay Propagasi* dari BTS ke UE

MCS	Delay Propagasi (s)
MCS-1	$0,5758 \cdot 10^{-5}$
MCS-2	$0,4524 \cdot 10^{-5}$
MCS-3	$0,3423 \cdot 10^{-5}$
MCS-4	$0,2879 \cdot 10^{-5}$
MCS-5	$0,2262 \cdot 10^{-5}$
MCS-6	$0,1712 \cdot 10^{-5}$
MCS-7	$0,1131 \cdot 10^{-5}$
MCS-8	$0,0931 \cdot 10^{-5}$
MCS-9	$0,0856 \cdot 10^{-5}$

Sumber: Perhitungan

f) *Delay Propagasi* dari GGSN ke 3.5G SGSN

Pada GGSN ke 3.5G SGSN dihubungkan dengan media transmisi *fiber optic* dengan jarak antara GGSN ke 3.5G SGSN diasumsikan 2000 m. Mengacu pada persamaan (2.66), maka nilai *delay* propagasi antara GGSN ke 3.5G SGSN adalah sebagai berikut:

$$\begin{aligned} t_{P(GGSN-3.5G SGSN)} &= \frac{N_{frame\ GGSN} \times R}{v} \\ &= \frac{1 \times 2000}{2 \times 10^8} \\ &= 1 \times 10^{-5} s \end{aligned}$$

g) *Delay Propagasi* dari 3.5G SGSN ke RNC

Pada 3.5G SGSN ke RNC dihubungkan dengan media transmisi *fiber optic* dengan jarak antara SGSN ke RNC adalah 3000 m. Berdasarkan persamaan (2.66), maka *delay* propagasi antara 3.5G SGSN ke RNC adalah sebagai berikut:

$$\begin{aligned} t_{P(3.5G SGSN-RNC)} &= \frac{N_{frame\ ATM} \times R}{v} \\ &= \frac{10 \times 3000}{2 \times 10^8} \\ &= 1,5 \times 10^{-4} s \end{aligned}$$

h) *Delay Propagasi* dari RNC ke Node B

Pada Node B dan RNC dihubungkan dengan media transmisi *fiber optic* dan jarak antara RNC ke Node B adalah 400 m. Sehingga nilai *delay* propagasi berdasarkan persamaan (2.66) adalah:

$$\begin{aligned}
 t_{P(RNC-Node\ B)} &= \frac{N_{frame\ ATM2} \times R}{v} \\
 &= \frac{14 \times 400}{2 \times 10^8} \\
 &= 2,8 \times 10^{-5}\ s
 \end{aligned}$$

i) Delay Propagasi dari Node B ke UE

Pada skripsi ini jarak Node B dan UE yang digunakan adalah untuk tipe sel *micro* dengan daerah operasi urban dan pusat kota yaitu sebesar 0,4 km. Sehingga besar *delay propagasi* antara *Node B* ke *UE* berdasarkan persamaan (2.66) adalah:

$$\begin{aligned}
 t_{P(Node\ B-UE)} &= \frac{N_{frame\ MAC-hs} \times R}{v} \\
 &= \frac{18 \times 400}{3 \times 10^8} \\
 &= 2,4 \times 10^{-5}\ s
 \end{aligned}$$

j) Delay Propagasi Total pada Jaringan EDGE

Delay propagasi total pada jaringan EDGE merupakan penjumlahan total *delay propagasi* pada *interface* antara PDN *Internet* dan *UE* pada Jaringan EDGE. *Delay propagasi* pada Jaringan EDGE untuk MCS-1 dapat ditentukan sesuai persamaan berikut:

$$\begin{aligned}
 t_{P(EDGE)} &= t_{P(PDN\ Internet-GGSN)} + t_{P(GGSN-2.75G\ SGSN)} + t_{P(2.75G\ SGSN-BSC)} \\
 &\quad + t_{P(BSC-BTS)} + t_{P(BTS-UE)} \\
 &= 7,5 \times 10^{-5} + 5 \times 10^{-6} + 1,255 \times 10^{-7} + 0,4318 \times 10^{-5} + 0,5758 \times 10^{-5} \\
 &= 0,90202 \times 10^{-4}\ s
 \end{aligned}$$

Dengan cara perhitungan yang sama maka didapatkan *delay propagasi total* pada jaringan EDGE untuk MSC-2 sampai MCS-9 yang ditunjukkan pada Tabel 4.9.

Tabel 4.9 Hasil Perhitungan *Delay Propagasi Total* pada Jaringan EGDE

MCS	Delay Propagasi Total (s)
MCS-1	0,9020.10 ⁻⁴
MCS-2	0,8804.10 ⁻⁴
MCS-3	0,8612.10 ⁻⁴
MCS-4	0,8516.10 ⁻⁴
MCS-5	0,8408.10 ⁻⁴
MCS-6	0,8312.10 ⁻⁴
MCS-7	0,8210.10 ⁻⁴
MCS-8	0,8175.10 ⁻⁴
MCS-9	0,8162.10 ⁻⁴

Sumber: Perhitungan

Dari Tabel 4.9 didapatkan analisis bahwa semakin besar MCS yang digunakan, maka nilai *delay propagasi* akan semakin kecil. Hal ini disebabkan karena setiap MCS mempunyai *data rate* yang berbeda, yang akan semakin besar dari MCS-1 hingga MCS-



9, sehingga besar *delay* yang terjadi berbanding terbalik dengan *data rate* yang digunakan.

k) Delay Propagasi Total pada Jaringan HSDPA

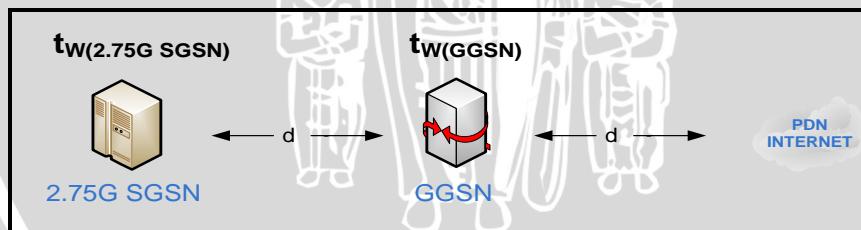
Nilai *delay* propagasi total pada Jaringan HSDPA dapat ditentukan dengan persamaan berikut:

$$\begin{aligned} t_{P(HSDPA)} &= t_{P(PDN \text{ Internet} - GGSN)} + t_{P(GGSN - 3.5G SGSN)} + t_{P(3.5G SGSN - RNC)} \\ &\quad + t_{P(RNC - Node \text{ B})} + t_{P(Node \text{ B} - UE)} \\ &= 7,5 \times 10^{-5} + 1 \times 10^{-5} + 1,5 \times 10^{-4} + 2,8 \times 10^{-5} + 2,4 \times 10^{-5} \\ &= 2,87 \times 10^{-4} \text{ s} \end{aligned}$$

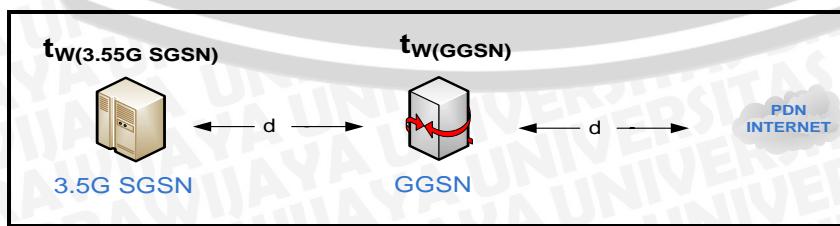
Berdasarkan persamaan untuk *delay* propagasi, nilai *delay* propagasi berbanding lurus dengan jarak pentransmision antar *node*. Semakin besar jarak antar *node*, maka akan semakin besar pula nilai *delay* propagasi, dan sebaliknya. Pada jaringan HSDPA nilai *delay* propagasi lebih kecil dibandingkan dengan *delay* propagasi pada jaringan EDGE. Hal ini disebabkan jarak pentransmision atau area jangkauan pada jaringan HSDPA yang lebih luas dari jaringan EDGE.

4.3.1.4 Delay Antrian

Pada skripsi ini, akan dianalisis *delay* antrian yang terjadi pada GGSN, 2.75G SGSN, dan 3.5G SGSN dengan menggunakan model antrian M/M/1 seperti ditunjukkan pada Gambar 4.8 untuk jaringan EDGE dan Gambar 4.9 untuk Jaringan HSDPA.



Gambar 4.8 Arsitektur *Node-to-node* Delay Antrian pada Jaringan EDGE
Sumber : Perancangan



Gambar 4.9 Arsitektur *Node-to-node* Delay Antrian pada Jaringan HSDPA
Sumber : Perancangan



a) Delay Antrian pada GGSN

Interface yang digunakan pada GGSN berupa *Fast Ethernet* dengan kecepatan 100 Mbps dengan panjang data 443 byte, maka berdasarkan persamaan (2.69) diperoleh nilai kecepatan pelayanan *node* sebesar:

$$\begin{aligned}\mu_{GGSN} &= \frac{C_{GGSN}}{m} \\ &= \frac{100 \times 10^6}{443 \times 8} \\ &= 2,8217 \times 10^4 \text{ paket/s}\end{aligned}$$

Dalam pembahasan ini, nilai faktor utilisasi (ρ) diubah dari nilai 0,1 sampai 0,9 dengan kenaikan sebesar 0,1. Dari persamaan (2.71) diperoleh nilai kecepatan kedatangan paket pada *node* (λ) sebesar:

$$\begin{aligned}\lambda_{GGSN} &= \mu_{GGSN} \times \rho \\ &= 2,8217 \times 10^4 \times 0,1 \\ &= 0,2822 \times 10^4 \text{ bps}\end{aligned}$$

Sehingga dapat ditentukan nilai *delay antrian* pada GGSN berdasarkan pada persamaan (2.73) yaitu:

$$\begin{aligned}t_{w(GGSN)} &= \frac{\lambda_{GGSN}}{\mu_{GGSN}(\mu_{GGSN} - \lambda_{GGSN})} + \frac{1}{\mu_{GGSN}} \\ &= \frac{0,2822 \times 10^4}{2,8217 \times 10^4 (2,8217 \times 10^4 - 0,2822 \times 10^4)} + \frac{1}{2,8217 \times 10^4} \\ &= 0,0394 \times 10^{-3} \text{ s}\end{aligned}$$

Dengan cara yang sama didapatkan nilai *delay antrian* pada GGSN dengan faktor utilisasi (ρ) sebesar 0,2 hingga 0,9, yaitu:

Tabel 4.10 Hasil Perhitungan *Delay Antrian* pada GGSN

Faktor Utilitas	<i>Delay Antrian</i> (s)
0,1	$0,0394 \cdot 10^{-3}$
0,2	$0,0443 \cdot 10^{-3}$
0,3	$0,0506 \cdot 10^{-3}$
0,4	$0,0591 \cdot 10^{-3}$
0,5	$0,0709 \cdot 10^{-3}$
0,6	$0,0886 \cdot 10^{-3}$
0,7	$0,1181 \cdot 10^{-3}$
0,8	$0,1772 \cdot 10^{-3}$
0,9	$0,3544 \cdot 10^{-3}$

Sumber: Perhitungan

b) Delay Antrian pada 2.75G SGSN

Interface yang digunakan pada 2.75G SGSN mempunyai kecepatan 15 Mbps dengan panjang data 432 byte, maka berdasarkan persamaan (2.69) diperoleh nilai kecepatan pelayanan *node* sebesar:

$$\begin{aligned}\mu_{2.75G SGSN} &= \frac{C_{2.75G SGSN}}{m} \\ &= \frac{15 \times 10^6}{432 \times 8} \\ &= 4,3403 \times 10^3 \text{ paket/s}\end{aligned}$$

Dengan nilai faktor utilisasi (ρ) diubah dari nilai 0,1 sampai 0,9 dengan kenaikan sebesar 0,1 maka diperoleh nilai kecepatan kedatangan paket pada *node* (λ) berdasarkan persamaan (2.71) sebesar:

$$\begin{aligned}\lambda_{2.75G SGSN} &= \mu_{2.75G SGSN} \times \rho \\ &= 4,340 \times 10^3 \times 0,1 \\ &= 0,4340 \times 10^3 \text{ bps}\end{aligned}$$

Sehingga dapat ditentukan nilai *delay* antrian pada 2.75G SGSN berdasarkan pada persamaan (2.73) yaitu:

$$\begin{aligned}t_{w2.75G SGSN} &= \frac{\lambda_{2.75G SGSN}}{\mu_{2.75G SGSN}(\mu_{2.75G SGSN} - \lambda_{2.75G SGSN})} + \frac{1}{\mu_{2.75G SGSN}} \\ &= \frac{0,4340 \times 10^3}{4,3403 \times 10^3(4,3403 \times 10^3 - 0,4340 \times 10^3)} + \frac{1}{4,3403 \times 10^3} \\ &= 0,0003 \text{ s}\end{aligned}$$

Dengan cara yang sama didapatkan nilai *delay* antrian pada 2.75G SGSN dengan faktor utilisasi (ρ) sebesar 0,2 hingga 0,9, yaitu:

Tabel 4.11 Hasil Perhitungan *Delay* Antrian pada 2.75G SGSN

Faktor Utilitas	<i>Delay</i> Antrian (s)
0,1	0,0003
0,2	0,0003
0,3	0,0003
0,4	0,0004
0,5	0,0005
0,6	0,0006
0,7	0,0008
0,8	0,0012
0,9	0,0023

Sumber: Perhitungan

c) *Delay* Antrian pada 3.5G SGSN

Teknologi yang digunakan pada 3.5G SGSN berupa *Fast Ethernet* dengan kecepatan 100 Mbps dengan panjang data 530 byte, maka berdasarkan persamaan (2.69) diperoleh nilai kecepatan pelayanan *node* sebesar:

$$\begin{aligned}\mu_{3.5G SGSN} &= \frac{C_{3.5G SGSN}}{m} \\ &= \frac{100 \times 10^6}{530 \times 8} \\ &= 2,3585 \times 10^4 \text{ paket/s}\end{aligned}$$



Dengan nilai faktor utilisasi (ρ) diubah dari nilai 0,1 sampai 0,9 dengan kenaikan sebesar 0,1 maka diperoleh nilai kecepatan kedatangan paket pada *node* (λ) berdasarkan pada persamaan (2.71) sebesar:

$$\begin{aligned}\lambda_{3.5G SGSN} &= \mu_{3.5G SGSN} \times \rho \\ &= 2,3585 \times 10^4 \times 0,1 \\ &= 0,2358 \times 10^4 \text{ bps}\end{aligned}$$

Sehingga dapat ditentukan nilai *delay* antrian pada 3.5G SGSN berdasarkan pada persamaan (2.73) yaitu:

$$\begin{aligned}t_w(3.5G SGSN) &= \frac{\lambda_{3.5G SGSN}}{\mu_{3.5G SGSN}(\mu_{3.5G SGSN} - \lambda_{3.5G SGSN})} + \frac{1}{\mu_{3.5G SGSN}} \\ &= \frac{0,2358 \times 10^4}{2,3585 \times 10^4(2,3585 \times 10^4 - 0,2358 \times 10^4)} + \frac{1}{2,3585 \times 10^4} \\ &= 0,0471 \times 10^{-3} \text{ s}\end{aligned}$$

Dengan cara yang sama didapatkan nilai *delay* antrian pada 3.5G SGSN dengan faktor utilisasi (ρ) sebesar 0,2 hingga 0,9, yaitu:

Tabel 4.12 Hasil Perhitungan *Delay* Antrian pada 3.5G SGSN

Faktor Utilitas	<i>Delay</i> Antrian (s)
0,1	$0,0471 \cdot 10^{-3}$
0,2	$0,0530 \cdot 10^{-3}$
0,3	$0,0606 \cdot 10^{-3}$
0,4	$0,0707 \cdot 10^{-3}$
0,5	$0,0848 \cdot 10^{-3}$
0,6	$0,1060 \cdot 10^{-3}$
0,7	$0,1413 \cdot 10^{-3}$
0,8	$0,2120 \cdot 10^{-3}$
0,9	$0,4240 \cdot 10^{-3}$

Sumber: Perhitungan

Dari hasil-hasil perhitungan tersebut didapatkan *delay* antrian total yang merupakan penjumlahan dari *delay* antrian yang ada pada masing-masing *node*. Pada jaringan EDGE, *delay* antrian total merupakan penjumlahan dari *delay* antrian di GGSN dan 2.75G SGSN. Sedangkan pada jaringan HSDPA, *delay* antrian total merupakan penjumlahan *delay* antrian pada GGSN dan 3.5G SGSN.

Pada jaringan EDGE nilai *delay* antrian total untuk faktor utilisasi 0,1 didapatkan berdasarkan persamaan (2.74) berikut:

$$\begin{aligned}t_W(\text{EDGE total}) &= t_w(\text{GGSN}) + t_w(\text{2.75G SGSN}) \\ &= 0,0394 \times 10^{-3} + 0,0003 \\ &= 0,0003 \text{ s}\end{aligned}$$



Dengan cara yang sama didapatkan nilai *delay* antrian total pada jaringan EDGE dengan faktor utilisasi (ρ) sebesar 0,2 hingga 0,9 yang ditampilkan dalam Tabel 4.13.

Tabel 4.13 Hasil Perhitungan *Delay* Antrian Total pada Jaringan EDGE

Faktor Utilisasi	<i>Delay</i> Antrian GGSN (s)	<i>Delay</i> Antrian 2.75G SGSN (s)	<i>Delay</i> Antrian Total (s)
0,1	$0,0394 \cdot 10^{-3}$	0,0003	0,0003
0,2	$0,0443 \cdot 10^{-3}$	0,0003	0,0003
0,3	$0,0506 \cdot 10^{-3}$	0,0003	0,0004
0,4	$0,0591 \cdot 10^{-3}$	0,0004	0,0004
0,5	$0,0709 \cdot 10^{-3}$	0,0005	0,0005
0,6	$0,0886 \cdot 10^{-3}$	0,0006	0,0007
0,7	$0,1181 \cdot 10^{-3}$	0,0008	0,0009
0,8	$0,1772 \cdot 10^{-3}$	0,0012	0,0013
0,9	$0,3544 \cdot 10^{-3}$	0,0023	0,0027

Sumber: Perhitungan

Pada jaringan HSDPA, nilai *delay* antrian total untuk faktor utilisasi 0,1 didapatkan berdasarkan persamaan (2.75) sebagai berikut:

$$\begin{aligned} t_{W(HSDPA \text{ total})} &= t_w(\text{GGSN}) + t_w(3.5G \text{ SGSN}) \\ &= 0,0394 \times 10^{-3} + 0,0471 \times 10^{-3} \\ &= 0,0865 \times 10^{-3} \text{ s} \end{aligned}$$

Dengan cara yang sama didapatkan nilai *delay* antrian total pada jaringan HSDPA dengan faktor utilisasi (ρ) sebesar 0,2 hingga 0,9 yang ditampilkan dalam Tabel 4.14.

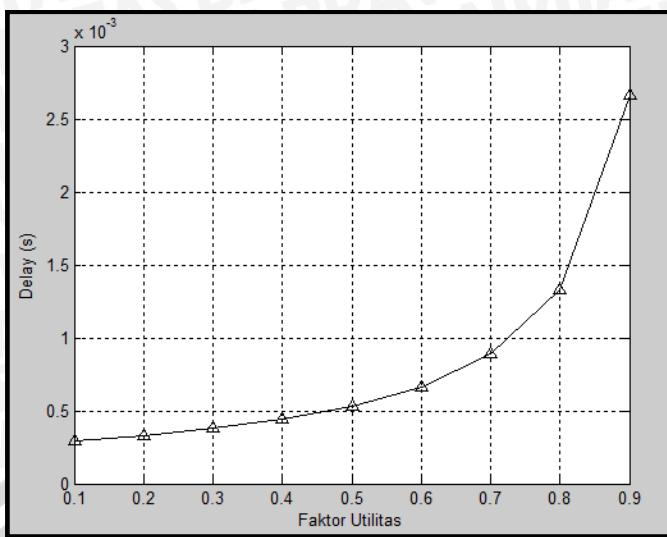
Tabel 4.14 Hasil Perhitungan *Delay* Antrian Total pada Jaringan HSDPA

Faktor Utilisasi	<i>Delay</i> Antrian GGSN (s)	<i>Delay</i> Antrian 3.5G SGSN (s)	<i>Delay</i> Antrian Total (s)
0,1	$0,0374 \cdot 10^{-3}$	$0,0471 \cdot 10^{-3}$	$0,0865 \cdot 10^{-3}$
0,2	$0,0399 \cdot 10^{-3}$	$0,0530 \cdot 10^{-3}$	$0,0973 \cdot 10^{-3}$
0,3	$0,0430 \cdot 10^{-3}$	$0,0606 \cdot 10^{-3}$	$0,1112 \cdot 10^{-3}$
0,4	$0,0473 \cdot 10^{-3}$	$0,0707 \cdot 10^{-3}$	$0,1297 \cdot 10^{-3}$
0,5	$0,0532 \cdot 10^{-3}$	$0,0848 \cdot 10^{-3}$	$0,1557 \cdot 10^{-3}$
0,6	$0,0620 \cdot 10^{-3}$	$0,1060 \cdot 10^{-3}$	$0,1946 \cdot 10^{-3}$
0,7	$0,0768 \cdot 10^{-3}$	$0,1413 \cdot 10^{-3}$	$0,2595 \cdot 10^{-3}$
0,8	$0,1063 \cdot 10^{-3}$	$0,2120 \cdot 10^{-3}$	$0,3892 \cdot 10^{-3}$
0,9	$0,1949 \cdot 10^{-3}$	$0,4240 \cdot 10^{-3}$	$0,7784 \cdot 10^{-3}$

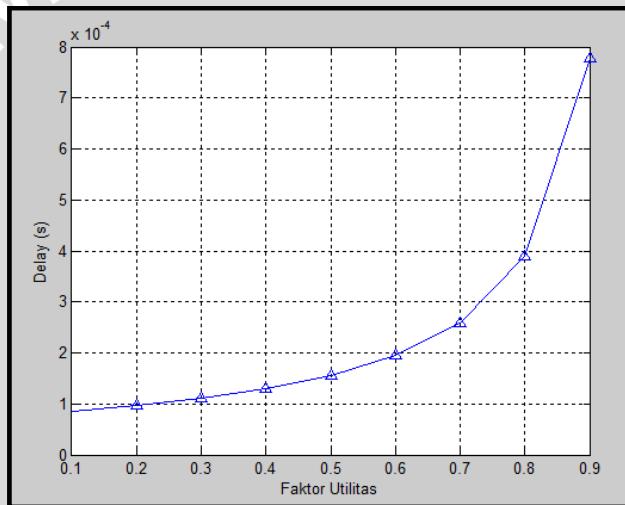
Sumber: Perhitungan

Dari Tabel 4.13 dan 4.14 di atas maka dapat dibuat grafik *delay* antrian total terhadap faktor utilisasi, pada jaringan EDGE dan HSDPA sebagai berikut:





Gambar 4.10 Grafik Hubungan *Delay* Antrian Total Jaringan EDGE terhadap Faktor Utilisasi
Sumber: Analisis



Gambar 4.11 Grafik Hubungan *Delay* Antrian Total Jaringan HSDPA terhadap Faktor Utilisasi
Sumber: Analisis

Berdasarkan analisis yang telah dilakukan baik melalui matematis maupun grafis, maka dapat dijelaskan bahwa:

- Semakin bertambahnya faktor utilisasi, maka nilai *delay* antrian yang terjadi semakin besar. Hal ini disebabkan semakin besar faktor utilisasi, jumlah rata-rata pengguna di antrian semakin banyak, sehingga *delay* yang terjadi semakin besar.
- *Delay* antrian terbesar terdapat pada faktor utilisasi 0,9 pada jaringan EDGE, yaitu sebesar $0,0027\text{ s}$. Sedangkan nilai *delay* antrian terkecil dicapai saat faktor utilisasi 0,1 terdapat pada jaringan HSDPA, yaitu sebesar $0,0865 \cdot 10^{-3}\text{ s}$.

4.3.1.5 Delay Total

Dari hasil perhitungan *delay* proses (enkapsulasi/dekapsulasi), *delay* transmisi, *delay* propagasi, dan *delay* antrian, maka dapat dihitung *delay end-to-end* pada jaringan EDGE maupun jaringan EDGE.

a) *Delay Total* pada Jaringan EDGE

Besar *delay end-to-end* pada jaringan EDGE dengan panjang paket data 369 byte, MCS-1, dan faktor utilisasi 0,1 dapat diperoleh dengan persamaan berikut:

$$\begin{aligned} t_{net}(\text{EDGE}) &= t_{proc}(\text{EDGE}) + t_T(\text{EGDE}) + t_P(\text{EGDE}) + t_W(\text{EGDE}) \\ &= 0,1716 + 0,2263 + 0,90202 \times 10^{-4} + 0,3007 \times 10^{-3} \\ &= 0,3982 \text{ s} \end{aligned}$$

Dengan cara yang sama diperoleh nilai *delay end-to-end* pada jaringan EGDE dengan MCS-2 sampai MCS-9 dan faktor utilisasi (ρ) sebesar 0,2 hingga 0,9, yang ditampilkan dalam Tabel 4.15.

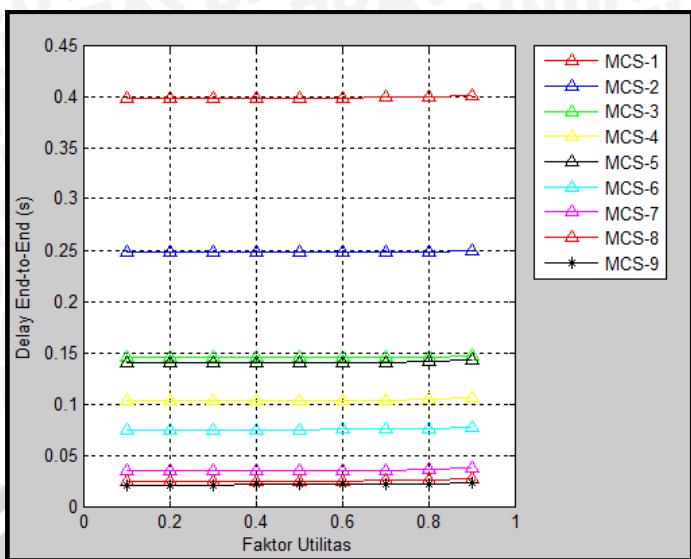
Tabel 4.15 Hasil Perhitungan *Delay Total* pada Jaringan EDGE

Faktor Utilisasi	Delay Total Jaringan EDGE (s)								
	MCS-1	MCS-2	MCS-3	MCS-4	MCS-5	MCS-6	MCS-7	MCS-8	MCS-9
0,1	0,3983	0,2475	0,1448	0,1030	0,1398	0,0746	0,0345	0,0246	0,0210
0,2	0,3983	0,2475	0,1449	0,1031	0,1398	0,0747	0,0345	0,0246	0,0210
0,3	0,3983	0,2476	0,1449	0,1031	0,1399	0,0747	0,0346	0,0247	0,0210
0,4	0,3984	0,2476	0,1450	0,1032	0,1400	0,0748	0,0347	0,0247	0,0211
0,5	0,3985	0,2477	0,1451	0,1033	0,1400	0,0749	0,0347	0,0248	0,0212
0,6	0,3986	0,2478	0,1452	0,1034	0,1402	0,0750	0,0349	0,0249	0,0213
0,7	0,3989	0,2481	0,1454	0,1036	0,1404	0,0752	0,0351	0,0252	0,0215
0,8	0,3993	0,2485	0,1459	0,1041	0,1408	0,0757	0,0355	0,0256	0,0220
0,9	0,4006	0,2498	0,1472	0,1054	0,1422	0,0770	0,0369	0,0269	0,0233

Sumber: Perhitungan

Dari Tabel 4.15 di atas maka dapat dibuat grafik *delay total* pada jaringan EDGE terhadap faktor utilisasi, sebagai berikut:





Gambar 4.12 Grafik Hubungan *Delay Total* Jaringan EDGE terhadap Faktor Utilisasi
Sumber: Analisis

b) *Delay Total* pada Jaringan HSDPA

Besar *delay end-to-end* pada jaringan HSDPA dengan panjang paket data 369 byte, dan faktor utilisasi 0,1 dapat diperoleh dengan persamaan berikut:

$$\begin{aligned}
 t_{net(HSDPA)} &= t_{proc(HSDPA)} + t_T(HSDPA) + t_P(HSDPA) + t_W(HSDPA) \\
 &= 0,0020 + 7,0435 \times 10^{-4} + 2,87 \times 10^{-4} + 0,0822 \times 10^{-3} \\
 &= 0,0031 \text{ s}
 \end{aligned}$$

Dengan cara yang sama diperoleh nilai *delay end-to-end* pada jaringan EGDE dengan paket data 369 byte dan faktor utilisasi (ρ) sebesar 0,2 hingga 0,9 yang ditampilkan dalam Tabel 4.16:

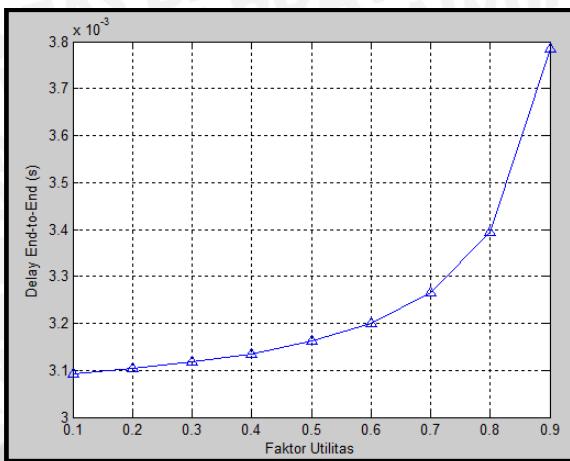
Tabel 4.16 Hasil Perhitungan *Delay Total* pada Jaringan HSDPA

Faktor Utilitas	Delay Total (s)
0,1	0,0031
0,2	0,0031
0,3	0,0031
0,4	0,0031
0,5	0,0032
0,6	0,0032
0,7	0,0033
0,8	0,0034
0,9	0,0038

Sumber: Perhitungan

Dari Tabel 4.16 di atas maka dapat dibuat grafik *delay total* pada jaringan HSDPA terhadap faktor utilisasi, sebagai berikut:





Gambar 4.13 Grafik Hubungan *Delay* Total Jaringan HSDPA terhadap Faktor Utilisasi
Sumber: Analisis

Berdasarkan analisis yang telah dilakukan baik melalui matematis maupun grafis, maka dapat dijelaskan bahwa:

- Pada jaringan EDGE, karena *delay* berbanding terbalik dengan data *rate*, maka pada faktor utilisasi yang sama, semakin besar MCS yang digunakan, data rate semakin besar, sehingga *delay* total pada jaringan semakin kecil.
- Pada Jaringan EDGE maupun HSDPA, semakin besar faktor utilisasi yang digunakan, maka *delay* antrian yang akan semakin besar, sehingga *end-to-end* jaringan juga akan semakin besar.
- Pada jaringan EDGE, nilai *delay* total terbesar terdapat pada MCS-1 dengan faktor utilisasi 0,9 yaitu sebesar 0,3956 s. Sedangkan nilai *delay* total terkecil terdapat pada MCS-9 dengan faktor utilisasi 0,2 yaitu sebesar 0,0920 s.
- Pada jaringan HSDPA, nilai *delay* total terbesar terjadi pada faktor utilisasi 0,9 yaitu sebesar 0,0034 s, sedangkan *delay* total terkecil terdapat pada faktor utilisasi 0,1 sampai 0,6 yaitu sebesar 0,0031 s.

4.3.2 Perhitungan *Throughput*

Analisis *throughput* pada skripsi ini dihitung mulai dari PDN *Internet* hingga UE berdasarkan persamaan (2.76) hingga (2.88).

a) Perhitungan *Throughput* Jaringan EDGE

Untuk ukuran data 369 byte, probabilitas bit *error* pada jaringan EDGE sebesar 10^{-9} , dengan menggunakan persamaan (2.83) maka probabilitas data yang diterima salah dapat dinyatakan dengan :

$$\rho_{mobile\ tv} = P_{size} \times \rho_b$$

$$\begin{aligned} &= 369 \times 8 \times 10^{-9} \\ &= 2,952 \times 10^{-6} \end{aligned}$$

Probabilitas *packet loss* pada jaringan GPRS dihitung dari probabilitas *packet loss* yang terjadi pada EDGE *air interface*, PDN *Internet*, GGSN, 2.75G SGSN, BSC, dan BTS sesuai persamaan (2.80).

Berdasarkan persamaan (2.81), maka probabilitas *packet loss* pada PDN *Internet*, GGSN, 2.75G SGSN, dan GERAN untuk paket data 369 byte yaitu:

$$\begin{aligned}\rho_{PDN\text{intnrt}} &= (l - l')\rho_b \\ &= 435 \times 8 \times 10^{-9} \\ &= 3,48 \times 10^{-6} \\ \rho_{GGSN} &= (l - l')\rho_b \\ &= 443 \times 8 \times 10^{-9}\end{aligned}$$

$$\begin{aligned}\rho_{2.75G\text{SGSN}} &= (l - l')\rho_b \\ &= 432 \times 8 \times 10^{-9} \\ &= 4,456 \times 10^{-6} \\ \rho_{GERAN} &= (l - l')\rho_b \\ &= 380 \times 8 \times 10^{-9} \\ &= 4,04 \times 10^{-6}\end{aligned}$$

Kemudian apabila berdasarkan standar BER yang diberikan oleh ETSI untuk EDGE adalah sebesar 10^{-4} maka dengan menggunakan persamaan (2.82) dapat diperoleh probabilitas *frame error* pada *air interface* sebesar :

$$\begin{aligned}\rho_{AI} &= 1 - (l - P_e)^{l-l'} \\ &= 1 - (1 - 10^{-4})^{456} \\ &= 0,0446\end{aligned}$$

Pada jaringan EDGE, probabilitas *packet loss* dihitung berdasarkan persamaan (2.80):

$$\begin{aligned}\rho_{net(EDGE)} &= 1 - [(1 - \rho_{AI})(1 - \rho_{GERAN})^h(1 - \rho_{SGSN})(1 - \rho_{GGSN})(1 - \rho_{PDN\text{ Internet}})] \\ &= 1 - [(1 - 0,0446)(1 - 4,04 \times 10^{-6})^h(1 - 4,456 \times 10^{-6}) \\ &\quad (1 - 3,544 \times 10^{-6})(1 - 3,48 \times 10^{-6})] \\ &= 0,0046\end{aligned}$$

Probabilitas *packet loss* total pada jaringan EDGE dapat dihitung dengan persamaan (2.78):

$$\begin{aligned}\rho_{tot(EDGE)} &= 1 - [(1 - \rho_{net(EDGE)})(1 - \rho_{mobile\ tv})] \\ &= 1 - [(1 - 0,0046)(1 - 2,952 \times 10^{-6})] \\ &= 0,0446\end{aligned}$$

Waktu minimum transmisi (t_l) untuk MCS-1 dengan data rate 8,9 kbps didapatkan dengan persamaan (2.85):

$$\begin{aligned}t_l &= \frac{(Pl_{frame} + H_{frame}) \times 8}{C_{trans}} \\ &= \frac{456}{8,9 \times 10^3} \\ &= 0,0512\text{ s}\end{aligned}$$



Delay propagasi total dan *delay* proses total untuk satu *frame* dapat dihitung dengan persamaan:

$$\begin{aligned} t_p &= \frac{t_{P(EDGE)}}{N_{frame RLCradioblock}} \\ &= \frac{0,90202 \times 10^{-4} s}{17,2727} \\ &= 0,0522 \times 10^{-4} s \end{aligned}$$

$$\begin{aligned} t_{proses} &= \frac{t_{proc(EDGE)}}{N_{frame RLCradioblock}} + \frac{t_{W(EDGE)}}{N_{frame RLCradioblock}} \\ &= \frac{0,1716}{17,2727} + \frac{0,0003}{17,2727} \\ &= 0,01 s \end{aligned}$$

Sehingga *fixed timeout interval* dapat diperoleh sesuai dengan persamaan (2.86) sebesar:

$$\begin{aligned} t_{out} &= 2t_p + 2t_I + 2t_{proses} \\ &= (2 \times 0,0522 \times 10^{-4}) + (2 \times 0,0512) + (2 \times 0,01) \\ &= 0,1224 s \end{aligned}$$

Maka konstanta dapat diperoleh sesuai dengan persamaan (2.87) yaitu:

$$\begin{aligned} \alpha &= 1 + \frac{t_{out}}{t_I} \\ &= 1 + \frac{0,1221}{0,0512} \\ &= 3,3890 s \end{aligned}$$

Sehingga berdasarkan persamaan (2.77), besar *throughput* pada jaringan EDGE dengan panjang data 369 byte, MCS-1, dan faktor utilisasi 0,1 adalah sebagai berikut:

$$\begin{aligned} \lambda &= \frac{(1 - \rho_{tot})}{t_I[1 + (\alpha - 1)\rho_{tot}]} \\ &= \frac{(1 - 0,0446)}{0,0512[1 + (3,3890 - 1)0,0446]} \\ &= 16,8518 \text{ paket/s} \\ &= 16,8564 \times 456 \\ &= 7,684 \times 10^3 \text{ bps} \end{aligned}$$

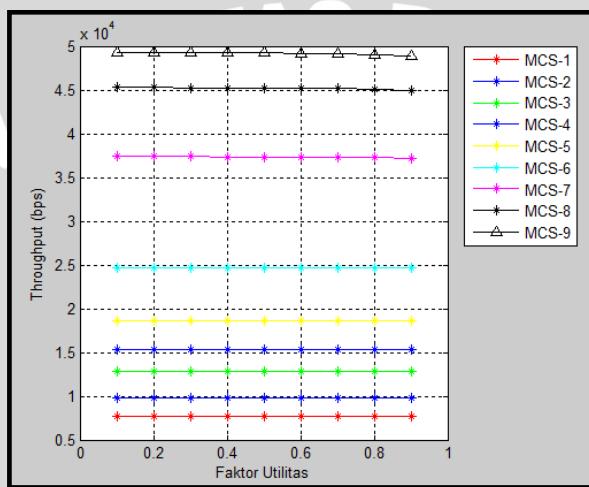
Dengan cara yang sama diperoleh nilai *throughput* pada jaringan EGDE dengan MCS-2 sampai MCS-9 dan faktor utilisasi (ρ) sebesar 0,2 hingga 0,9 seperti pada Tabel 4.17. Dari Tabel 4.17 di atas maka dapat dibuat grafik *throughput* pada jaringan EDGE terhadap faktor utilisasi, yang diperlihatkan pada Gambar 4.14.



Tabel 4.17 Hasil Perhitungan *Throughput* pada Jaringan EDGE

Faktor Utilisasi	Throughput Jaringan EDGE (bps)								
	MCS-1	MCS-2	MCS-3	MCS-4	MCS-5	MCS-6	MCS-7	MCS-8	MCS-9
0,1	$7,6845 \cdot 10^3$	$9,7570 \cdot 10^3$	$1,2860 \cdot 10^4$	$1,5278 \cdot 10^4$	$1,8602 \cdot 10^4$	$2,4711 \cdot 10^4$	$3,7376 \cdot 10^4$	$4,5274 \cdot 10^4$	$4,9277 \cdot 10^4$
0,2	$7,6844 \cdot 10^3$	$9,7569 \cdot 10^3$	$1,2860 \cdot 10^4$	$1,5278 \cdot 10^4$	$1,8602 \cdot 10^4$	$2,4710 \cdot 10^4$	$3,7373 \cdot 10^4$	$4,5269 \cdot 10^4$	$4,9270 \cdot 10^4$
0,3	$7,6844 \cdot 10^3$	$9,7568 \cdot 10^3$	$1,2860 \cdot 10^4$	$1,5278 \cdot 10^4$	$1,8601 \cdot 10^4$	$2,4709 \cdot 10^4$	$3,7369 \cdot 10^4$	$4,5262 \cdot 10^4$	$4,9260 \cdot 10^4$
0,4	$7,6844 \cdot 10^3$	$9,7567 \cdot 10^3$	$1,2860 \cdot 10^4$	$1,5277 \cdot 10^4$	$1,8600 \cdot 10^4$	$2,4708 \cdot 10^4$	$3,7364 \cdot 10^4$	$4,5252 \cdot 10^4$	$4,9248 \cdot 10^4$
0,5	$7,6843 \cdot 10^3$	$9,7566 \cdot 10^3$	$1,2859 \cdot 10^4$	$1,5277 \cdot 10^4$	$1,8599 \cdot 10^4$	$2,4706 \cdot 10^4$	$3,7357 \cdot 10^4$	$4,5239 \cdot 10^4$	$4,9231 \cdot 10^4$
0,6	$7,6842 \cdot 10^3$	$9,7564 \cdot 10^3$	$1,2859 \cdot 10^4$	$1,5276 \cdot 10^4$	$1,8598 \cdot 10^4$	$2,4702 \cdot 10^4$	$3,7345 \cdot 10^4$	$4,5219 \cdot 10^4$	$4,9205 \cdot 10^4$
0,7	$7,6841 \cdot 10^3$	$9,7561 \cdot 10^3$	$1,2858 \cdot 10^4$	$1,5275 \cdot 10^4$	$1,8596 \cdot 10^4$	$2,4697 \cdot 10^4$	$3,7327 \cdot 10^4$	$4,5186 \cdot 10^4$	$4,9163 \cdot 10^4$
0,8	$7,6837 \cdot 10^3$	$9,7555 \cdot 10^3$	$1,2857 \cdot 10^4$	$1,5272 \cdot 10^4$	$1,8591 \cdot 10^4$	$2,4686 \cdot 10^4$	$3,7290 \cdot 10^4$	$4,5120 \cdot 10^4$	$4,9077 \cdot 10^4$
0,9	$7,6828 \cdot 10^3$	$9,7536 \cdot 10^3$	$1,2852 \cdot 10^4$	$1,5265 \cdot 10^4$	$1,8577 \cdot 10^4$	$2,4654 \cdot 10^4$	$3,7178 \cdot 10^4$	$4,5100 \cdot 10^4$	$4,8823 \cdot 10^4$

Sumber: Perhitungan

**Gambar 4.14** Grafik Hubungan *Throughput* Jaringan EDGE terhadap Faktor Utilisasi
Sumber: Analisis**b) Perhitungan *Throughput* pada Jaringan HSDPA**

Pada jaringan HSDPA diketahui bahwa probabilitas *blocking* sebesar 10^{-8} . Maka besarnya probabilitas *packet loss* untuk data *mobile TV* yang terjadi adalah:

$$\begin{aligned} \rho_{\text{mobile TV}} &= P_{\text{size}} \times \rho_b \\ &= 369 \times 8 \times 10^{-8} \\ &= 2,952 \times 10^{-5} \end{aligned}$$

Probabilitas *packet loss* pada jaringan HSDPA dihitung dari probabilitas *packet loss* pada HSDPA *air interface*, RAN (Node B dan RNC), *core network* (GGSN dan 3.5SGSN), dan PDN *Internet* sesuai persamaan (2.81)

$$\begin{aligned} \rho_{\text{PDNintrnt}} &= (l - l')\rho_b \\ &= 435 \times 8 \times 10^{-8} \\ &= 3,48 \times 10^{-5} \end{aligned}$$

$$\begin{aligned} \rho_{\text{GGSN}} &= (l - l')\rho_b \\ &= 443 \times 8 \times 10^{-8} \\ &= 3,544 \times 10^{-5} \end{aligned}$$

$$\begin{aligned} \rho_{\text{3.5G SGSN}} &= (l - l')\rho_b \\ &= 530 \times 8 \times 10^{-8} \\ &= 4,24 \times 10^{-5} \\ \rho_{\text{RNC}} &= (l - l')\rho_b \\ &= 742 \times 8 \times 10^{-8} \\ &= 5,936 \times 10^{-5} \end{aligned}$$

$$\begin{aligned}\rho_{Node\ B} &= (l - l')\rho_b \\ &= 810 \times 8 \times 10^{-8} \\ &= 6,48 \times 10^{-5}\end{aligned}$$

Probabilitas *packet loss* pada *air interface* HSDPA dapat ditentukan berdasarkan parameter – parameter *link budget* pada Tabel 4.18 berikut ini:

Tabel 4.18. Link Budget Air Interface HSDPA

Kategori	Unit	Nilai
Frekuensi Operasi	MHz	2642,5
Jarak IMR-UE	Km	0,4
EIRP	dBW	57
<i>Interference Margin</i>	dB	9
Rugi-rugi Lintasan Propagasi	dB	126,36
<i>Gain</i> antena Rx	dB	0
<i>Noise Figure</i> UE	dB	8
<i>Implement Losses</i>	dB	3
<i>Noise Power Spectral Density</i>	dBW/Hz	-204,3

Sumber: Indosat dari skripsi Yosina Septianingsih, 2008:69

- Daya yang diterima kanal trafik (P_{tr})

Berdasarkan pada tabel 4.18 dan persamaan (2.55) didapatkan nilai daya yang diterima kanal trafik (P_{tr}), yaitu:

$$\begin{aligned}P_{tr} &= EIRP - L_p + G_{MS} - I_m \\ &= 57 - 126,36 + 0 - 9 \\ &= -78,36\ dBW\end{aligned}$$

- *Thermal Noise* (N_o)

Nilai dari *Thermal Noise* (N_o) dapat dihitung berdasarkan pada tabel 4.18 dan persamaan (2.56), yaitu:

$$\begin{aligned}N_o &= 10 \log(290 \times 1,38 \cdot 10^{-23}) + NF_{MS} + 40 \\ &= -203,98 + 8 + 40 \\ &= -155,98\ dBm/Hz\end{aligned}$$

- Eb/No kanal trafik

Nilai Eb/No kanal trafik dapat dihitung berdasarkan pada persamaan (2.57) serta nilai – nilai P_{tr} dan N_o yang telah didapatkan sebelumnya.

$$\begin{aligned}[E_b/N_o]_{Tr} &= P_{tr} - 10 \log R_b - 10 \log 10^{0,1 N_o} \\ &= -78,36 - 10 \log(3,6 \cdot 10^6) - 10 \log 10^{0,1(-155,98)} \\ &= -78,36 - 65,56 + 155,98 \\ &= 12,06\ dB \\ &= 16,07\end{aligned}$$

- Probabilitas *error* kanal trafik

Setelah didapatkan nilai Eb/No, maka probabilitas *error* kanal trafik dapat dihitung sesuai dengan persamaan (2.58), yaitu:

$$\begin{aligned} P_e &= \frac{e^{-Eb/No}}{\sqrt[2]{\pi(Eb/No)}} \\ &= \frac{e^{-16,07}}{\sqrt[2]{(3,14)(16,07)}} \\ &= \frac{1,049 \cdot 10^{-7}}{7,1035} = 4,7288 \cdot 10^{-7} \end{aligned}$$

- Probabilitas *packet loss air interface HSDPA*

Sesuai dengan persamaan (2.59) dan nilai probabilitas *error* kanal trafik, maka didapatkan nilai probabilitas *packet loss air interface HSDPA* yaitu sebesar:

$$\begin{aligned} \rho_{AI} &= 1 - (1 - P_e)^{l+l'} \\ &= 1 - (1 - 4,7288 \cdot 10^{-7})^{810 \times 8} \\ &= 0,0031 \end{aligned}$$

Sehingga dengan persamaan (2.79) dapat diperoleh probabilitas *packet loss* pada jaringan HSDPA.

$$\begin{aligned} \rho_{net(HSDPA)} &= 1 - [(1 - \rho_{AI})(1 - \rho_{RNC})(1 - \rho_{Node_B})(1 - \rho_{SGSN}) \\ &\quad (1 - \rho_{GGSN})(1 - \rho_{PDN\ Internet})] \\ &= 1 - [(1 - 0,0031)(1 - 5,936 \times 10^{-5})(1 - 6,48 \times 10^{-5})(1 - 4,24 \times 10^{-5}) \\ &\quad (1 - 3,544 \times 10^{-5})(1 - 3,48 \times 10^{-5})] \\ &= 0,0032 \end{aligned}$$

Probabilitas *packet loss* total pada jaringan HSDPA dapat dihitung dengan persamaan (2.78):

$$\begin{aligned} \rho_{tot(HSDPA)} &= 1 - [(1 - \rho_{net(HSDPA)})(1 - \rho_{mobile\ tv})] \\ &= 1 - [(1 - 0,0032)(1 - 2,952 \times 10^{-6})] \\ &= 0,0032 \end{aligned}$$

Waktu yang dibutuhkan untuk pentransmision satu *frame* ditentukan melalui persamaan (2.85), yaitu:

$$\begin{aligned} t_I &= \frac{(Pl_{frame} + H_{frame}) \times 8}{C_{trans}} \\ &= \frac{810 \times 8}{3,6 \times 10^6} \\ &= 0,0018\ s \end{aligned}$$

Delay propagasi total dan *delay proses* total untuk satu *frame* dapat dihitung dengan persamaan:

$$t_p = \frac{t_{P(HSDPA)}}{N_{frame}}$$



$$= \frac{2,87 \times 10^{-4} \text{ s}}{18}$$

$$= 1,5944 \times 10^{-5} \text{ s}$$

$$t_{\text{proses}} = \frac{t_{\text{proc (HSDPA)}}}{N_{\text{frame}}} + \frac{t_W(\text{HSDPA})}{N_{\text{frame}}}$$

$$= \frac{0,0020}{18} + \frac{0,0822 \times 10^{-3}}{18}$$

$$= 0,1167 \text{ s}$$

Fixed timeout interval dapat diperoleh sesuai dengan persamaan (2.86) sebesar:

$$t_{\text{out}} = 2t_p + 2t_l + 2t_{\text{proc}}$$

$$= (2 \times 1,5944 \times 10^{-5}) + (2 \times 0,0018) + (2 \times 0,1167 \times 10^{-3})$$

$$= 0,0039 \text{ s}$$

Maka konstanta dapat diperoleh sesuai dengan persamaan (2.87) yaitu:

$$\alpha = 1 + \frac{t_{\text{out}}}{t_l}$$

$$= 1 + \frac{0,0039}{0,0018}$$

$$= 3,1474$$

Sehingga diperoleh nilai *throughput* pada jaringan HSDPA untuk faktor utilisasi 0,1 dan paket data 369 byte, sesuai dengan persamaan (2.77) sebesar:

$$\lambda = \frac{(1 - \rho_{\text{tot}})}{t_l[1 + (\alpha - 1)\rho_{\text{tot}}]}$$

$$= \frac{(1 - 0,0039)}{0,0018[1 + (3,1474 - 1)0,0039]}$$

$$= 549,9370 \text{ paket/s}$$

$$= 549,9370 \times 810 \times 8$$

$$= 3,5636 \times 10^6 \text{ bps}$$

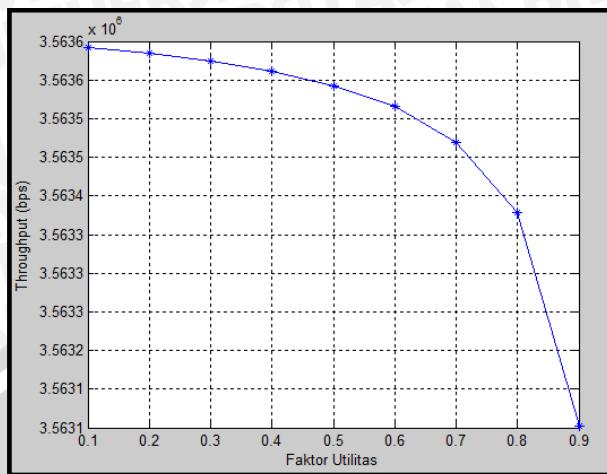
Dengan cara yang sama diperoleh nilai *throughput* pada jaringan HSDPA dengan faktor utilisasi (ρ) sebesar 0,2 hingga 0,9 seperti pada Tabel 4.18.

Tabel 4.19 Hasil Perhitungan *Throughput* pada Jaringan HSDPA

Faktor Utilitas	Throughput (bps)
0,1	$3,5636 \cdot 10^6$
0,2	$3,5636 \cdot 10^6$
0,3	$3,5636 \cdot 10^6$
0,4	$3,5636 \cdot 10^6$
0,5	$3,5635 \cdot 10^6$
0,6	$3,5635 \cdot 10^6$
0,7	$3,5635 \cdot 10^6$
0,8	$3,5634 \cdot 10^6$
0,9	$3,5631 \cdot 10^6$

Sumber: Perhitungan

Dari Tabel 4.18 di atas maka dapat dibuat grafik *throughput* pada jaringan HSDPA terhadap faktor utilisasi, sebagai berikut:



Gambar 4.15 Grafik Hubungan *Throughput* Jaringan HSDPA terhadap Faktor Utilisasi
Sumber: Analisis

Berdasarkan analisis yang dilakukan secara matematis maupun grafis dapat dijelaskan bahwa:

- Pada pengiriman data dari PDN *Internet* ke UE melalui jaringan HSDPA menunjukkan bahwa nilai *throughput* pada jaringan HSDPA dipengaruhi oleh beberapa faktor, antara lain besar data yang dikirimkan, *delay* enkapsulasi dan dekapsulasi, *delay* antrian, faktor utilisasi, serta *packet loss*. Selain itu kenaikan faktor utilisasi akan menyebabkan penurunan nilai *throughput* jaringan, namun penurunan yang terjadi relatif kecil sehingga dapat dikatakan bahwa *throughput* pada jaringan HSDPA relatif konstan untuk semua faktor utilisasi.
- Pada pengiriman data dari PDN *Internet* ke UE melalui jaringan EDGE menunjukkan bahwa nilai *throughput* pada jaringan EDGE dipengaruhi oleh beberapa faktor, antara lain besar data yang dikirimkan, penggunaan *modulation and Coding Scheme* (MCS), *delay* enkapsulasi dan dekapsulasi, *delay* antrian, faktor utilisasi, serta *packet loss*. Kenaikan faktor utilisasi menyebabkan penurunan nilai *throughput* jaringan. Selain itu, semakin besar MCS yang digunakan, maka *throughput* pada jaringan semakin besar. Namun perubahan yang terjadi relatif kecil sehingga pada analisis grafis perubahan *throughput* terlihat tidak begitu jelas, atau dapat dikatakan bahwa *throughput* pada jaringan EDGE relatif konstan untuk semua faktor utilisasi.

4.4 Analisis *Delay Handover*, dan Pengaruhnya

Pada analisis skripsi ini akan dibahas mengenai *delay handover*, serta pengaruhnya pada *throughput* jaringan EDGE. Proses kalkulasi *delay handover* dari jaringan HSDPA ke EDGE dihitung berdasarkan jumlah *message* yang dikirimkan selama proses *handover*. Jumlah *message* yang dikirimkan selama proses *handover* merupakan *message* yang berukuran 64 bit (8 byte) yang dibangkitkan sebagai ICMP (*Internet Control Message Protocol*) *message* dan payload *mobile TV*. *Message* yang akan dikirimkan tersebut akan ditambahkan dengan *header IP* dan 8 byte *original IP payload* (Shivendra Panwar, 1997:43). Sehingga datagram paket untuk *handover* adalah sebagai berikut:

a) Pada Jaringan EDGE

$$\begin{aligned} W_{\text{datagram}} &= W_{\text{ICMP}} + W_{\text{mobileTV}} \\ &= 8 \text{ byte} + 369 \text{ byte} \\ &= 377 \text{ byte} \end{aligned}$$

Panjang *frame original IP payload* adalah:

$$\begin{aligned} W_{\text{original IP payload}} &= W_{\text{datagram}} + \text{original IP payload} \\ &= 377 \text{ byte} + 8 \text{ byte} \\ &= 385 \text{ byte} \end{aligned}$$

Karena jaringan EDGE menggunakan IPv4, maka panjang *frame original IP payload* akan dienkapsulasi dengan menambahkan *header IP* sebesar 20 byte.

$$\begin{aligned} W_{\text{frame IP1}} &= W_{\text{original IP payload}} + \text{header}_{\text{IP}} \\ &= 385 \text{ byte} + 20 \text{ byte} \\ &= 405 \text{ byte} \end{aligned}$$

Panjang *frame IP* tidak melebihi MTU *Ethernet* sebesar 1500 byte, sehingga tidak dilakukan segmentasi. Selanjutnya dihitung panjang *frame SNDNP* dengan terlebih dahulu menambahkan *header SNDNP* sebesar 4 byte pada *frame IP*.

$$\begin{aligned} W_{\text{frame SNDNP}} &= W_{\text{frame IP1}} + \text{header}_{\text{SNDNP}} \\ &= 405 \text{ byte} + 4 \text{ byte} \\ &= 409 \text{ byte} \end{aligned}$$

Karena panjang *frame SNDNP* tidak melebihi MSS SNDNP sebesar 1503 byte, maka tidak perlu disegmentasi. Selanjutnya *frame SNDNP* ditambahkan dengan 4 byte *header LLC* dan 3 byte *header FCS*, untuk menjadi *frame LLC*.

$$\begin{aligned} W_{\text{frame LLC}} &= W_{\text{frame SNDNP}} + \text{header}_{\text{LLC}} + \text{header}_{\text{FCS}} \\ &= 409 \text{ byte} + 4 \text{ byte} + 3 \text{ byte} \\ &= 416 \text{ byte} \end{aligned}$$



Frame LLC akan disegmentasi menjadi RLC/MAC *Radio Block*. Untuk MCS-6, jumlah frame RLC/MAC *radio block* dapat dihitung sebagai berikut:

$$\begin{aligned} N_{\text{frame RLC radioblock}} &= \frac{W_{\text{frame LLC}}}{\text{RLC data unit}} \\ &= \frac{416 \text{ byte}}{74 \text{ byte}} \\ &= 5,6216 \end{aligned}$$

Sehingga *delay* enkapsulasi yang terjadi di 2.75G UE untuk MCS-1 adalah sebagai berikut:

$$\begin{aligned} t_E(2.75G \text{ UE}) &= \frac{\text{jumlah frame IP} \times \text{jumlah frame SNDCP} \times N_{\text{frame RLC block}} \times \text{RLC/MACradioblock}}{n \times V_{TS}} \\ &= \frac{1 \times 1 \times 5,6216 \times 456}{2 \times 29,7 \times 10^3} \\ &= 0,0432 \text{ s} \end{aligned}$$

Dengan cara yang sama, diperoleh nilai *delay* enkapsulasi untuk MCS yang berbeda seperti pada tabel 4.20.

Tabel 4.20 Hasil Perhitungan *Delay* Enkapsulasi pada UE di EDGE

MCS	Delay Dekapsulasi pada UE di EDGE (s)
MCS-1	0,4844
MCS-2	0,2998
MCS-3	0,1720
MCS-4	0,1218
MCS-5	0,0753
MCS-6	0,0432
MCS-7	0,0188
MCS-8	0,0128
MCS-9	0,0108

Sumber: Perhitungan

Paket data ditransmisikan dari 2.75G UE ke BTS melalui Um *Interface* dengan *delay* transmisi untuk MCS-6 sebesar:

$$\begin{aligned} t_T(\text{Um Interface}) &= \frac{N_{\text{frame RLC block}} \times \text{RLC}_{\text{radio block}}}{n \times V_{MCS}} \\ &= \frac{5,6216 \times 456}{2 \times 29,7 \times 10^3} \\ &= 0,0432 \text{ s} \end{aligned}$$

Apabila jarak antara 2.75G UE dan BTS adalah 100 meter, maka dapat diperoleh *delay* propagasi sebesar:

$$\begin{aligned} t_P(\text{UE-BTS}) &= \frac{N_{\text{frame RLC block}} \times R}{v} \\ &= \frac{5,6216 \times 100}{3 \times 10^8} \\ &= 0,1847 \times 10^{-5} \text{ s} \end{aligned}$$

Delay transmisi untuk data yang dikirim dari BTS akan ditransmisikan ke BSC melalui Abis Interface dengan MCS-6, dapat dihitung sebagai berikut:

$$\begin{aligned} t_{T(\text{Abis Interface})} &= \frac{N_{\text{frame RLC block}} \times RLC_{\text{radio block}}}{V_{\text{Abis Interface}}} \\ &= \frac{5,6216 \times 456}{150 \times 16 \times 10^3 \text{ bps}} \\ &= 0,0011 \text{ s} \end{aligned}$$

Delay propagasi yang terjadi antara BTS dan BSC dengan MCS-6 adalah sebagai berikut:

$$\begin{aligned} t_{P(BTS-BSC)} &= \frac{N_{\text{frame RLC radioblok}} \times R}{v} \\ &= \frac{5,6216 \times 50}{2 \times 10^8} \\ &= 0,1405 \times 10^{-5} \text{ s} \end{aligned}$$

Sehingga *delay* proses uplink pada UE ke BSS untuk MCS-6 adalah sebagai berikut:

$$\begin{aligned} t_{\text{net}(2.75UE-BSC)} &= t_E(2.75G \text{ UE}) + t_{T(Um \text{ Interface})} + t_{P(UE-BTS)} \\ &\quad + t_{T(\text{Abis Interface})} + t_{P(BTS-BSC)} \\ &= 0,0432 + 0,0432 + 0,1847 \times 10^{-5} + 0,0011 + 0,1405 \times 10^{-5} \\ &= 0,0847 \text{ s} \end{aligned}$$

Pada BSS, data yang dikirimkan dari UE dengan MCS-6 akan mengalami proses dekapsulasi dengan perhitungan sebagai berikut:

$$\begin{aligned} t_D(\text{BSS}) &= \frac{\text{jumlah frame LLC} \times N_{\text{frame RLC block}} \times \frac{RLC \text{ data}}{\text{code rate}}}{V_2(\text{PCU})} \\ &= \frac{1 \times 5,6216 \times \frac{592}{1/2}}{14 \times 150 \times 16 \times 10^3 \text{ bps}} \\ &= 0,1981 \times 10^{-3} \text{ s} \end{aligned}$$

Selanjutnya paket yang masuk ke BSS akan dienkapsulasi dengan menambahkan *header* BSSGP sebesar 48 byte.

$$\begin{aligned} W_{\text{frame BSSGP}} &= W_{\text{datagram}} + \text{header}_{\text{BSSGP}} \\ &= 377 \text{ byte} + 48 \text{ byte} \\ &= 425 \text{ byte} \end{aligned}$$

Kemudian paket data dienkapsulasi lagi dengan menambahkan *header* NS sebesar 4 byte dan *header* FCS.

$$\begin{aligned} W_{\text{frame NS}} &= W_{\text{frame BSSGP}} + \text{header}_{\text{NS}} + \text{FCS} \\ &= 425 \text{ byte} + 4 \text{ byte} + 3 \text{ byte} \\ &= 432 \text{ byte} \end{aligned}$$



Panjang *frame* NS tidak melebihi MTU NS sebesar 1600 byte, sehingga tidak dilakukan segmentasi. Maka *delay* enkapsulasi yang terjadi di BSS adalah sebagai berikut:

$$\begin{aligned} t_{E(BSS)} &= \frac{\text{jumlah frame LLC} \times \text{jumlah frame NS} \times W_{\text{frame NS}} \times 8}{V_{1(PCU)}} \\ &= \frac{(1 \times 1 \times 432 \times 8) \text{bit}}{(14 \times 31 \times 64 \times 10^3) \text{bps}} \\ &= 1,2442 \times 10^{-4} \text{s} \end{aligned}$$

Delay transmisi untuk pengiriman melalui Gb *Interface* ke 2.75G SGSN adalah sebesar:

$$\begin{aligned} t_{T(\text{Gb Interface})} &= \frac{W_{\text{frame NS}} \times 8}{V_{\text{Gb Interface}}} \\ &= \frac{(432 \times 8) \text{bit}}{2,048 \times 10^6 \text{bps}} \\ &= 0,0017 \text{s} \end{aligned}$$

Sedangkan *delay* propagasi dari BSC ke 2.75G SGSN adalah:

$$\begin{aligned} t_{P(BSC-2.75G SGSN)} &= \frac{N_{\text{frame NS}} \times R}{v} \\ &= \frac{1 \times 25}{2 \times 10^8} \\ &= 1,255 \times 10^{-7} \text{s} \end{aligned}$$

Sehingga *delay* total proses *uplink* dari BSS ke 2.75G SGSN untuk MCS-6 adalah sebagai berikut:

$$\begin{aligned} t_{net(BSS-2.75G SGSN)} &= t_{D(BSS)} + t_{E(BSS)} + t_{T(\text{Gb Interface})} + t_{P(BSC-2.75G SGSN)} \\ &= 0,2396 \times 10^{-3} + 1,2442 \times 10^{-4} + 0,0017 + 1,255 \times 10^{-7} \\ &= 0,002 \text{s} \end{aligned}$$

Pada 2.75G SGSN, paket data yang diterima dari BSS akan mengalami proses dekapsulasi, dengan besar *delay* dekapsulasi pada 2.75G SGSN adalah sebagai berikut:

$$\begin{aligned} t_{D(2.75G SGSN)} &= \frac{\text{jumlah frame LLC} \times W_{\text{frame NS}} \times \text{jumlah frame NS} \times 8}{V_{2.75G SGSN}} \\ &= \frac{1 \times 432 \times 1 \times 8}{15 \times 10^6} \\ &= 2,304 \times 10^{-4} \text{s} \end{aligned}$$

Selanjutnya *frame* IP akan dienkapsulasi dengan menambahkan *header* GTP, untuk perutean pada jaringan *backbone* EDGE.

$$\begin{aligned} W_{\text{frame GTP}} &= W_{\text{frame IP}} + \text{header}_{\text{GTP}} \\ &= 405 \text{ byte} + 8 \text{ byte} \\ &= 413 \text{ byte} \end{aligned}$$



Karena panjang *frame IP backbone* tidak melebihi MTU Ethernet sebesar 1500 byte, maka tidak perlu disegmentasi. Selanjutnya *frame GTP* ditambahkan dengan 14 byte *header Ethernet* dan FCS, untuk menjadi *frame Ethernet*.

$$\begin{aligned} W_{\text{frame Ethernet}} &= W_{\text{frame GTP}} + \text{header}_{\text{ethernet}} + \text{FCS} \\ &= 413 \text{ byte} + 14 \text{ byte} + 4 \text{ byte} \\ &= 431 \text{ byte} \end{aligned}$$

Karena panjang *frame Ethernet* tidak melebihi MTU Ethernet sebesar 1500 byte, maka tidak perlu disegmentasi. *Delay enkapsulasi* yang terjadi di 2.75G SGSN adalah sebagai berikut:

$$\begin{aligned} t_{E(2.75G \text{ SGSN})} &= \frac{\text{jumlah frame IP backbone} \times \text{jumlah frame Ethernet} \times W_{\text{frame Ethernet}} \times 8}{V_{2.75G \text{ SGSN}}} \\ &= \frac{1 \times 1 \times 431 \times 8}{15 \times 10^6} \\ &= 2,2987 \times 10^{-4} \text{ s} \end{aligned}$$

Besar *delay transmisi* untuk paket data yang dikirimkan ke GGSN melalui Gn *Interface* adalah sebagai berikut:

$$\begin{aligned} t_{T(\text{Gn Interface})} &= \frac{W_{\text{frame Ethernet}} \times 8}{V_{\text{Gn Interface}}} \\ &= \frac{(431 \times 8) \text{ bit}}{100 \times 10^6 \text{ bps}} \\ &= 3,448 \times 10^{-5} \text{ s} \end{aligned}$$

Sedangkan *delay propagasi* dari 2.75G SGSN ke GGSN adalah sebagai berikut:

$$\begin{aligned} t_P(2.75G \text{ SGSN-GGSN}) &= \frac{N_{\text{frame Ethernet}} \times R}{v} \\ &= \frac{1 \times 1000}{2 \times 10^8} \\ &= 5 \times 10^{-6} \text{ s} \end{aligned}$$

Delay antrian yang dianalisis adalah *delay* yang terjadi antara GGSN, 2.75G SGSN, dan 3.5G SGSN dengan menggunakan model antrian M/M/1 sebagaimana telah dijelaskan pada Bab 2. *Interface* yang digunakan pada 2.75G SGSN mempunyai kecepatan 15 Mbps. Dengan panjang data 431 byte, maka diperoleh nilai kecepatan pelayanan node sebesar:

$$\begin{aligned} \mu_{2.75G \text{ SGSN}} &= \frac{C_{2.75G \text{ SGSN}}}{m} \\ &= \frac{15 \times 10^6}{431 \times 8} \\ &= 4,3503 \times 10^3 \text{ paket/s} \end{aligned}$$



Dengan nilai faktor utilisasi (ρ) diubah dari nilai 0,1 sampai 0,9 dengan kenaikan sebesar 0,1 maka diperoleh nilai kecepatan kedatangan paket pada node (λ) berdasarkan pada persamaan (2.71) sebesar:

$$\begin{aligned}\lambda_{2.75G SGSN} &= \mu_{2.75G SGSN} \times \rho \\ &= 4,3503 \times 10^3 \times 0,1 \\ &= 0,4350 \times 10^3 \text{ bps}\end{aligned}$$

Nilai *delay* antrian pada 2.75G SGSN untuk proses *uplink* adalah sebagai berikut:

$$\begin{aligned}t_{w2.75G SGSN} &= \frac{\lambda_{2.75G SGSN}}{\mu_{2.75G SGSN}(\mu_{2.75G SGSN} - \lambda_{2.75G SGSN})} + \frac{1}{\mu_{2.75G SGSN}} \\ &= \frac{0,4350 \times 10^3}{4,3503 \times 10^3(4,3503 \times 10^3 - 0,4350 \times 10^3)} + \frac{1}{4,3503 \times 10^3} \\ &= 0,0003 \text{ s}\end{aligned}$$

Sehingga *delay* total proses *uplink* dari 2.75G SGSN ke GGSN untuk faktor utilisasi 0,1 adalah sebagai berikut:

$$\begin{aligned}t_{net}(2.75G SGSN-GGSN) &= t_D(2.75G SGSN) + t_E(2.75G SGSN) + t_T(Gn Interface) \\ &\quad + t_P(2.75G SGSN-GGSN) + t_{w2.75G SGSN} \\ &= 2,304 \times 10^{-4} + 2,2987 \times 10^{-4} + 3,448 \times 10^{-5} \\ &\quad + 5 \times 10^{-6} + 0,0002 \\ &= 0,0008 \text{ s}\end{aligned}$$

Paket data pada PDN *Internet* akan ditambahkan dengan *header* UDP, sehingga panjang segmen UDP adalah sebagai berikut:

$$\begin{aligned}W_{frame UDP} &= W_{ICMP} + W_{mobileTV} + header_{UDP} \\ &= 8 \text{ byte} + 369 \text{ byte} + 8 \text{ byte} \\ &= 385 \text{ byte}\end{aligned}$$

Panjang segmen tersebut ditambahkan dengan *header* IP menjadi *frame* IP dengan panjang *frame* adalah:

$$\begin{aligned}W_{frame IP2} &= W_{frame UDP} + header_{IP} \\ &= 385 \text{ byte} + 40 \text{ byte} \\ &= 425 \text{ byte}\end{aligned}$$

Untuk mendapatkan *frame* PDN *Internet*, maka pada *frame* IP ditambahkan dengan *header Ethernet* sebesar 14 byte dan FCS.

$$\begin{aligned}W_{frame PDN Internet} &= W_{frame IP2} + header_{Ethernet} + FCS \\ &= 425 \text{ byte} + 14 \text{ byte} + 4 \text{ byte} \\ &= 443 \text{ byte}\end{aligned}$$

Pada skripsi ini PDN *Internet* menggunakan standar *Gigabit Ethernet* dengan kecepatan 1 Gbps. Maka *delay* enkapsulasi yang terjadi di PDN *Internet* untuk pengiriman paket data ke GGSN yaitu:

$$\begin{aligned}
 t_{E(PDN\ Internet)} &= \frac{W_{frame\ PDN\ Internet} \times 8}{V_{PDN\ Internet}} \\
 &= \frac{(443 \times 8)bit}{1 \times 10^9 bps} \\
 &= 3,544 \times 10^{-6} s
 \end{aligned}$$

Delay transmisi untuk pengiriman data dari *PDN Internet* ke *GGSN* melalui *Gi Interface* adalah sebagai berikut:

$$\begin{aligned}
 t_{T(Gi\ Interface)} &= \frac{W_{frame\ PDN\ Internet} \times 8}{V_{Ethernet}} \\
 &= \frac{(443 \times 8)bit}{1 \times 10^9 bps} \\
 &= 3,544 \times 10^{-6} s
 \end{aligned}$$

Delay propagasi dari *PDN Internet* ke *GGSN* adalah sebagai berikut:

$$\begin{aligned}
 t_{P(PDN\ Internet - GGSN)} &= \frac{N_{frame\ PDN\ Internet} \times R}{v} \\
 &= \frac{1 \times 15000}{2 \times 10^8} \\
 &= 7,5 \times 10^{-5} s
 \end{aligned}$$

Delay total proses pentransmisian paket data dari *PDN Internet* ke *GGSN* adalah sebagai berikut:

$$\begin{aligned}
 t_{net(PDN\ internet - GGSN)} &= t_{E(PDN\ Internet)} + t_{T(Gi\ Interface)} + t_{P(PDN\ Internet - GGSN)} \\
 &= 3,544 \times 10^{-6} + 3,544 \times 10^{-6} + 7,5 \times 10^{-5} \\
 &= 8,2088 \times 10^{-5} s
 \end{aligned}$$

Pada *GGSN*, *frame* yang diterima akan didekapsulasi dengan besar *delay* sebagai berikut:

$$\begin{aligned}
 t_{D(GGSN)} &= \frac{W_{datagram} \times 8}{V_{GGSN}} \\
 &= \frac{(377 \times 8)bit}{100 \times 10^6 bps} \\
 &= 3,016 \times 10^{-5} s
 \end{aligned}$$

Jumlah total *frame* yang akan ditransmisikan ke *2.75G SGSN* adalah sebagai berikut:

$$\begin{aligned}
 W_{frame\ IP3} &= W_{ICMP} + W_{mobileTV} + \text{original IP payload} + header_{GTP} \\
 &\quad + header_{UDP} + header_{IP} \\
 &= 8 byte + 369 byte + 8 byte + 8 byte + 8 byte + 20 byte \\
 &= 421 byte
 \end{aligned}$$

$$\begin{aligned}
 W_{frame\ GGSN} &= W_{frame\ IP3} + header_{Ethernet} + FCS \\
 &= 421 byte + 14 byte + 4 byte \\
 &= 439 byte
 \end{aligned}$$



Karena panjang *frame* Ethernet tidak melebihi MTU Ethernet sebesar 1500 byte, maka tidak perlu disegmentasi. Selanjutnya *frame* GGSN dienkapsulasi untuk dikirimkan ke 2.75G SGSN dengan besar *delay* enkapsulasi sebagai berikut:

$$\begin{aligned} t_{E(GGSN)} &= \frac{W_{frame\ GGSN} \times 8}{V_{GGSN}} \\ &= \frac{(439 \times 8)bit}{100 \times 10^6 bps} \\ &= 3,512 \times 10^{-5} s \end{aligned}$$

Besar *delay* transmisi untuk pengiriman paket data dari GGSN ke 2.75G SGSN melalui Gn *Interface* adalah sebagai berikut:

$$\begin{aligned} t_{T(Gn\ Interface)} &= \frac{W_{frame\ Ethernet} \times 8}{V_{Gn\ Interface}} \\ &= \frac{(439 \times 8)bit}{100 \times 10^6 bps} \\ &= 3,512 \times 10^{-5} s \end{aligned}$$

Sedangkan *delay* propagasi dari GGSN ke 2.75G SGSN adalah sebagai berikut:

$$\begin{aligned} t_{P(GGSN-2.75G\ SGSN)} &= \frac{N_{frame\ Ethernet} \times R}{v} \\ &= \frac{1 \times 1000}{2 \times 10^8} \\ &= 5 \times 10^{-6} s \end{aligned}$$

GGSN menggunakan *interface* berupa *Fast Ethernet* dengan kecepatan 100 Mbps, maka berdasarkan persamaan (2.69) diperoleh nilai kecepatan pelayanan node sebesar:

$$\begin{aligned} \mu_{GGSN} &= \frac{C_{GGSN}}{m} \\ &= \frac{100 \times 10^6}{439 \times 8} \\ &= 2,8474 \times 10^4 \text{ paket/s} \end{aligned}$$

Nilai faktor utilisasi (ρ) berubah dari nilai 0,1 sampai 0,9 dengan kenaikan sebesar 0,1. Dari persamaan (2.71) diperoleh nilai kecepatan kedatangan paket pada node (λ) sebesar:

$$\begin{aligned} \lambda_{GGSN} &= \mu_{GGSN} \times \rho \\ &= 2,8474 \times 10^4 \times 0,1 \\ &= 0,28474 \times 10^4 \text{ bps} \end{aligned}$$

Sehingga dapat ditentukan nilai *delay* antrian pada GGSN berdasarkan pada persamaan (2.73) yaitu:

$$t_w(GGSN) = \frac{\lambda_{GGSN}}{\mu_{GGSN}(\mu_{GGSN} - \lambda_{GGSN})} + \frac{1}{\mu_{GGSN}}$$



$$\begin{aligned}
 &= \frac{0,28474 \times 10^4}{2,8474 \times 10^4 (2,8474 \times 10^4 - 0,28474 \times 10^4)} + \frac{1}{2,8474 \times 10^4} \\
 &= 0,0390 \times 10^{-3} \text{ s}
 \end{aligned}$$

Delay total proses downlink dari GGSN ke 2.75G SGSN adalah sebagai berikut:

$$\begin{aligned}
 t_{net(GGSN-2.75GSGSN)} &= t_D(GGSN) + t_E(GGSN) + t_T(Gn\ Interface) + t_P(GGSN-2.75G\ SGSN) + t_w(GGSN) \\
 &= 3,016 \times 10^{-5} + 3,512 \times 10^{-5} + 3,512 \times 10^{-5} + 5 \times 10^{-6} + 0,0371 \times 10^{-3} \\
 &= 0,1444 \times 10^{-3} \text{ s}
 \end{aligned}$$

Besar *delay dekapsulasi* yang terjadi di 2.75G SGSN adalah sebagai berikut:

$$\begin{aligned}
 t_D(2.75G\ SGSN) &= \frac{\text{jumlah frame Ethernet} \times W_{frame\ Ethernet} \times 8}{V_{2.75G\ SGSN}} \\
 &= \frac{1 \times 439, \times 8}{15 \times 10^6} \\
 &= 2,3413 \times 10^{-4} \text{ s}
 \end{aligned}$$

Untuk memperoleh panjang *frame* SNDCP, panjang *frame* yang diterima GGSN akan dienkapsulasi dengan menambahkan *header* SNDCP.

$$\begin{aligned}
 W_{frame\ SNDCP} &= W_{frame\ GGSN} + header_{SNDCP} \\
 &= 439 \text{ byte} + 4 \text{ byte} \\
 &= 443 \text{ byte}
 \end{aligned}$$

Panjang *frame* SNDCP tidak melebihi MSS SNDCP sebesar 1503 byte, sehingga tidak perlu disegmentasi. Selanjutnya *frame* SNDCP dienkapsulasi dengan menambahkan 4 byte *header* LLC dan FCS, untuk menjadi *frame* LLC.

$$\begin{aligned}
 W_{frame\ LLC} &= W_{frame\ SNDCP} + header_{LLC} + FCS \\
 &= 443 \text{ byte} + 4 \text{ byte} + 3 \text{ byte} \\
 &= 450 \text{ byte}
 \end{aligned}$$

Karena panjang *frame* LLC tidak melebihi MSS LLC sebesar 1500 byte, maka tidak perlu disegmentasi. Selanjutnya *frame* LLC dienkapsulasi dengan menambahkan 48 byte *header* BSSGP.

$$\begin{aligned}
 W_{frame\ BSSGP} &= W_{frame\ LLC} + header_{BSSGP} \\
 &= 450 \text{ byte} + 48 \text{ byte} \\
 &= 498 \text{ byte}
 \end{aligned}$$

Pada *layer Network Service*, *frame* BSSGP akan ditransmisikan ke PCU menggunakan NS *Data Unit* dengan menambahkan *header* NS sebesar 4 byte.

$$\begin{aligned}
 W_{frame\ NS} &= W_{frame\ BSSGP} + header_{NS} \\
 &= 498 \text{ byte} + 4 \text{ byte} \\
 &= 502 \text{ byte}
 \end{aligned}$$



Panjang *frame* NS tidak melebihi MSU NS sebesar 1600 *byte*, sehingga tidak perlu disegmentasi. Sehingga *delay* enkapsulasi yang terjadi di 2.75G SGSN pada arah *downlink* adalah sebagai berikut:

$$\begin{aligned} t_{E(2.75G \text{ SGSN})} &= \frac{\text{jumlah frame SNDCP} \times \text{jumlah frame NS} \times W_{\text{frame NS}} \times 8}{V_{2.75G \text{ SGSN}}} \\ &= \frac{1 \times 1 \times 502 \times 8}{15 \times 10^6} \\ &= 2,6773 \times 10^{-4} \text{ s} \end{aligned}$$

Frame LLC hasil dekapsulasi selanjutnya akan disegmentasi menjadi RLC/MAC *Radio Block* untuk ditransmisikan melalui Abis *Interface* menuju BTS. Pada penggunaan MCS-6, maka akan didapat jumlah RLC/MAC *Radio Block* sebagai berikut:

$$\begin{aligned} N_{\text{frame RLC radioblock}} &= \frac{W_{\text{frame LLC}}}{\text{RLC data unit}} \\ &= \frac{450 \text{ byte}}{74 \text{ byte}} \\ &= 6,0811 \end{aligned}$$

Delay transmisi untuk pengiriman paket data dari 2.75G SGSN ke BSC melalui Gb *Interface* adalah sebagai berikut:

$$\begin{aligned} t_{T(\text{Gb Interface})} &= \frac{W_{\text{frame NS}} \times 8}{V_{\text{Gb Interface}}} \\ &= \frac{(502 \times 8) \text{ bit}}{2,048 \times 10^6 \text{ bps}} \\ &= 0,002 \text{ s} \end{aligned}$$

Delay propagasi antara 2.75G SGSN dan BSC adalah sebagai berikut:

$$\begin{aligned} t_{P(2.75G \text{ SGSN-BSC})} &= \frac{N_{\text{frame NS}} \times R}{v} \\ &= \frac{1 \times 25}{2 \times 10^8} \\ &= 1,255 \times 10^{-7} \text{ s} \end{aligned}$$

Delay transmisi pada pengiriman paket data dari BSC ke BTS melalui Abis *Interface* menggunakan MCS-1 adalah sebagai berikut:

$$\begin{aligned} t_{T(\text{Abis Interface})} &= \frac{N_{\text{frame RLC block}} \times RLC_{\text{radio block}}}{V_{\text{Abis Interface}}} \\ &= \frac{6,0811 \times 456}{150 \times 16 \times 10^3 \text{ bps}} \\ &= 0,0012 \text{ s} \end{aligned}$$

Delay propagasi antara BSC dan BTS menggunakan MCS-1 adalah sebagai berikut:

$$t_{P(\text{BSC-BTS})} = \frac{N_{\text{frame RLC radioblok}} \times R}{v}$$



$$\begin{aligned}
 &= \frac{20,4545 \times 50}{2 \times 10^8} \\
 &= 0,5114 \times 10^{-5} \text{ s}
 \end{aligned}$$

Laju pelayanan paket data pada 2.75G SGSN untuk arah *downlink* dapat ditentukan dengan perhitungan berikut:

$$\begin{aligned}
 \mu_{2.75G SGSN} &= \frac{C_{2.75G SGSN}}{m} \\
 &= \frac{15 \times 10^6}{502 \times 8} \\
 &= 3,7351 \times 10^3 \text{ paket/s}
 \end{aligned}$$

Dengan nilai faktor utilisasi (ρ) diubah dari nilai 0,1 sampai 0,9 dengan kenaikan sebesar 0,1 maka diperoleh nilai laju kedatangan paket pada node (λ) 2.75G SGSN sebesar:

$$\begin{aligned}
 \lambda_{2.75G SGSN} &= \mu_{2.75G SGSN} \times \rho \\
 &= 3,7351 \times 10^3 \times 0,1 \\
 &= 0,3735 \times 10^3 \text{ bps}
 \end{aligned}$$

Nilai *delay* antrian pada 2.75G SGSN untuk arah *downlink* adalah sebagai berikut:

$$\begin{aligned}
 t_{w2.75G SGSN} &= \frac{\lambda_{2.75G SGSN}}{\mu_{2.75G SGSN}(\mu_{2.75G SGSN} - \lambda_{2.75G SGSN})} + \frac{1}{\mu_{2.75G SGSN}} \\
 &= \frac{0,4350 \times 10^3}{4,3503 \times 10^3(4,3503 \times 10^3 - 0,4350 \times 10^3)} + \frac{1}{4,3503 \times 10^3} \\
 &= 0,0003 \text{ s}
 \end{aligned}$$

Sehingga *delay* total proses pentransmision paket data dari 2.75G SGSN ke BSS dengan MCS-6 dan faktor utilisasi 0,1 adalah sebagai berikut:

$$\begin{aligned}
 t_{net(2.75G SGSN-BSS)} &= t_D(2.75G SGSN) + t_E(2.75G SGSN) + t_T(Gb Interface) + t_P(2.75G SGSN-BSC) \\
 &\quad + t_T(Abis Interface) + t_P(BSC-BTS) + t_{w2.75G SGSN} \\
 &= 2,3413 \times 10^{-4} + 2,6773 \times 10^{-4} + 0,002 + 1,255 \times 10^{-7} \\
 &\quad + 0,0012 + 0,5114 \times 10^{-5} + 0,0003 \\
 &= 0,0039 \text{ s}
 \end{aligned}$$

Paket data yang diterima BSS akan didekapsulasi dengan *delay* dekapsulasi sebesar:

$$\begin{aligned}
 t_D(BSS) &= \frac{jumlah frame LLC \times jumlah frame NS \times W_{frame NS} \times 8}{V_{1(PCU)}} \\
 &= \frac{(1 \times 1 \times 502 \times 8)bit}{(14 \times 31 \times 64 \times 10^3)bps} \\
 &= 1,4459 \times 10^{-4} \text{ s}
 \end{aligned}$$

Frame LLC hasil dekapsulasi selanjutnya akan disegmentasi menjadi RLC/MAC *Radio Block*. Pada penggunaan MCS-6, maka akan didapat jumlah RLC/MAC *Radio Block* sebagai berikut:



$$\begin{aligned}
 N_{\text{frame RLC/MAC radioblock}} &= \frac{W_{\text{frame RLC/MAC radioblock}}}{\text{RLC data unit}} \\
 &= \frac{450 \text{ byte}}{74 \text{ byte}} \\
 &= 6,0811
 \end{aligned}$$

Delay enkapsulasi pada BSS untuk arah *downlink* adalah sebagai berikut:

$$\begin{aligned}
 t_E(BSS) &= \frac{\text{jumlah frame LLC} \times N_{\text{frame RLC/MAC block}} \times \text{RLC/MAC radioblock}}{V_2(\text{PCU})} \\
 &= \frac{1 \times 20,4545 \times 456}{14 \times 150 \times 16 \times 10^3 \text{ bps}} \\
 &= 0,2776^{-3} \text{ s}
 \end{aligned}$$

Pada pengiriman paket data dari BTS ke UE melalui Um *Interface* menggunakan MCS-6, didapatkan *delay* transmisi sebesar:

$$\begin{aligned}
 t_T(\text{Um Interface}) &= \frac{N_{\text{frame RLC block}} \times \text{RLC radio block}}{n \times V_{\text{MCS}}} \\
 &= \frac{20,4545 \times 456}{2 \times 29,7 \times 10^3} \\
 &= 0,0467 \text{ s}
 \end{aligned}$$

Besar *delay* propagasi antara BTS dan UE adalah sebagai berikut:

$$\begin{aligned}
 t_P(BSS-UE) &= \frac{N_{\text{frame RLC block}} \times R}{v} \\
 &= \frac{6,0811 \times 100}{3 \times 10^8} \\
 &= 0,02027 \times 10^{-5} \text{ s}
 \end{aligned}$$

Sehingga *delay* proses pentransmisian paket data untuk arah *downlink* dari BTS ke UE menggunakan MCS-6 adalah sebagai berikut:

$$\begin{aligned}
 t_{\text{net}(BSS-UE)} &= t_D(BSS) + t_E(BSS) + t_T(\text{Um Interface}) + t_P(BSS-UE) \\
 &= 1,4459 \times 10^{-4} + 0,2776^{-3} + 0,0467 + 0,02027 \times 10^{-5} \\
 &= 0,0469 \text{ s}
 \end{aligned}$$

b) Pada Jaringan HSDPA

$$\begin{aligned}
 W_{\text{datagram}} &= W_{\text{ICMP}} + W_{\text{mobileTV}} \\
 &= 8 \text{ byte} + 369 \text{ byte} \\
 &= 377 \text{ byte}
 \end{aligned}$$

Pada UE, ICMP *message* yang dibangkitkan akan dienkapsulasi dengan menambahkan 8 byte *original IP payload* dan 40 byte *header IP*. Sehingga panjang *frame IP* adalah sebagai berikut:

$$\begin{aligned}
 W_{\text{frame IP}} &= W_{\text{datagram}} + \text{original IP payload} + \text{header}_{\text{IP}} \\
 &= 377 \text{ byte} + 8 \text{ byte} + 40 \\
 &= 425 \text{ byte}
 \end{aligned}$$

Selanjutnya paket data aplikasi tersebut dienkapsulasi pada *layer PDCP* (*Packet Data Convergence Protocol*) dengan menambahkan *header PDCP* sebesar 1 byte.

$$\begin{aligned} W_{frame \ PDCP} &= W_{frame \ IP} + header_{PDCP} \\ &= 425 \text{ byte} + 1 \text{ byte} \\ &= 426 \text{ byte} \end{aligned}$$

Pada *layer RLC*, *frame PDCP* akan disegmentasi menjadi RLC PDU *fixed size* yaitu sebesar 40 byte, sehingga jumlah *frame RLC* adalah sebagai berikut:

$$\begin{aligned} N_{frame \ RLC} &= \frac{W_{frame \ PDCP}}{40 \text{ byte}} \\ &= \frac{426 \text{ byte}}{40 \text{ byte}} \\ &= 10,65 \end{aligned}$$

Terdapat 10 *frame RLC* berisi data 40 byte dan 1 *frame* berisi data sebesar $(0,65 \times 40 \text{ byte} = 26 \text{ byte})$ 26 byte. *Frame* yang berisi data 26 byte tersebut akan diberi *padding bytes* sebesar $(40 \text{ byte} - 26 \text{ byte} = 14 \text{ byte})$ 14 byte, sehingga terdapat 11 *frame RLC*. Setiap *frame* kemudian diberi *header* sebesar 2 byte sehingga panjang *frame* menjadi 42 byte. Panjang *frame RLC* total ditransmisikan menuju *layer MAC-d* adalah sebagai berikut:

$$\begin{aligned} W_{frame \ RLC \ total} &= N_{frame \ RLC} \times W_{frame \ RLC} \\ &= 11 \times 42 \text{ byte} \\ &= 462 \text{ byte} \end{aligned}$$

Selanjutnya pada *layer MAC-d*, RLC PDU akan disegmentasi menjadi MAC-d SDU (*Service Data Unit*) sebesar 42 byte. Maka jumlah *frame MAC-d SDU* adalah:

$$\begin{aligned} N_{frame \ MAC-d} &= \frac{W_{frame \ RLC \ total}}{42 \text{ byte}} \\ &= \frac{462 \text{ byte}}{42 \text{ byte}} \\ &= 11 \end{aligned}$$

Sedangkan panjang *frame MAC-d* dapat dihitung berdasarkan persamaan (2.32):

$$\begin{aligned} W_{frame \ MAC-d} &= header_{MAC} + 42 \text{ byte} \\ &= 3 \text{ byte} + 42 \text{ byte} \\ &= 45 \text{ byte} \end{aligned}$$

Sehingga panjang *frame MAC-d* total adalah sebagai berikut:

$$\begin{aligned} W_{frame \ MAC-d \ total} &= N_{frame \ MAC-d} \times W_{frame \ MAC-d} \\ &= 11 \times 45 \text{ byte} \\ &= 495 \text{ byte} \end{aligned}$$

Delay enkapsulasi untuk arah *uplink* pada 3.5G UE adalah sebagai berikut:

$$\begin{aligned} t_{E(3.5G\ UE)} &= \frac{W_{frame\ MAC-d\ total} \times 8}{V_{UE}} \\ &= \frac{(495 \times 8)bit}{3,6 \times 10^6 bps} \\ &= 0,0011 s \end{aligned}$$

Delay transmisi untuk pengiriman paket data dari 3.5G UE ke Node B melalui Uu *Interface* adalah sebagai berikut:

$$\begin{aligned} t_{T(Uu\ Interface)} &= \frac{W_{frame\ MAC-d\ total} \times 8}{n \times V_{UE}} \\ &= \frac{(495 \times 8)}{3 \times 3,6 \times 10^6 bps} \\ &= 3,6667 \times 10^{-4} s \end{aligned}$$

Delay propagasi antara 3.5G UE dan Node B adalah:

$$\begin{aligned} t_{P(UE-Node\ B)} &= \frac{N_{frame\ MAC-d} \times R}{v} \\ &= \frac{11 \times 400}{3 \times 10^8} \\ &= 1,4667 \times 10^{-5} s \end{aligned}$$

Pada Node B, data akan didekapsulasi sebagai berikut:

$$\begin{aligned} W_{datagram\ (Node\ B)} &= W_{frame\ MAC-d\ total} - header_{MAC} - header_{PDCP} \\ &\quad - original\ IP\ payload - header_{IP} \\ &= 495\ byte - 3\ byte - 1\ byte - 8\ byte - 40\ byte \\ &= 443\ byte \end{aligned}$$

Sehingga *delay* dekapsulasi yang terjadi adalah:

$$\begin{aligned} t_{D(Node\ B)} &= \frac{W_{datagram\ (Node\ B)} \times 8}{V_{Node\ B}} \\ &= \frac{(443 \times 8)bit}{155,52 \times 10^6 bps} \\ &= 2,2788 \times 10^{-5} s \end{aligned}$$

Pada saat memasuki *layer* MAC-hs, paket data disegmentasi menjadi paket sebesar 42 *byte*, sehingga jumlah *frame* MAC-hs sesuai dengan persamaan (2.41) yaitu:

$$\begin{aligned} N_{frame\ MAC-hs} &= \frac{W_{datagram\ (Node\ B)}}{42\ byte} \\ &= \frac{443byte}{42\ byte} \\ &= 10,5476 \end{aligned}$$

Terdapat 10 *frame* berisi data 42 *byte* dan 1 *frame* berisi data sebesar (0,5476 × 42 *byte* = 23 *byte*) 23 *byte*. *Frame* yang berisi data 10 *byte* tersebut akan diberi

padding bytes sebesar ($42\text{ byte} - 23\text{ byte} = 19\text{ byte}$) 19 byte , sehingga terdapat 11 frame . Panjang *frame* MAC-hs adalah sebagai berikut:

$$\begin{aligned}W_{frame\ MAC-hs} &= header_{MAC} + 42\text{ byte} \\&= 3\text{ byte} + 42\text{ byte} \\&= 45\text{ byte}\end{aligned}$$

Sehingga panjang *frame* total pada Node B yaitu:

$$\begin{aligned}W_{frame\ Node\ B} &= N_{frame\ MAC-hs} \times W_{frame\ MAC-hs} \\&= 11 \times 45\text{ byte} \\&= 495\text{ byte}\end{aligned}$$

Dengan menggunakan persamaan (2.44) didapatkan *delay* enkapsulasi yang terjadi di Node B sebesar:

$$\begin{aligned}t_{E(Node\ B)} &= \frac{W_{frame\ Node\ B} \times 8}{V_{Node\ B}} \\&= \frac{(495 \times 8)\text{ byte}}{155,52 \times 10^6\text{ bps}} \\&= 2,5463 \times 10^{-5}\text{ s}\end{aligned}$$

Delay transmisi untuk pengiriman paket data dari Node B ke RNC melalui Iub Interface adalah sebagai berikut:

$$\begin{aligned}t_{T(Iub\ Interface)} &= \frac{W_{frame\ Node\ B} \times 8}{V_{Iub\ Interface}} \\&= \frac{(495 \times 8)\text{ bit}}{155,52 \times 10^6\text{ bps}} \\&= 2,5463 \times 10^{-5}\text{ s}\end{aligned}$$

Delay propagasi antara Node B dan RNC adalah:

$$\begin{aligned}t_{P(RNC-Node\ B)} &= \frac{N_{frame\ MAC-hs} \times R}{v} \\&= \frac{11 \times 400}{2 \times 10^8} \\&= 2,2 \times 10^{-5}\text{ s}\end{aligned}$$

Sehingga *delay* proses pentransmisian paket data dari 3G UE ke RNC adalah sebagai berikut:

$$\begin{aligned}t_{net(UE-RNC)} &= t_{E(3.5G\ UE)} + t_{T(Uu\ Interface)} + t_{P(UE-Node\ B)} + t_{D(Node\ B)} + t_{E(Node\ B)} \\&\quad + t_{T(Iub\ Interface)} + t_{P(RNC-Node\ B)} \\&= 0,0011 + 3,6667 \times 10^{-4} + 1,4667 \times 10^{-5} + 2,2788 \times 10^{-5} \\&\quad + 2,5463 \times 10^{-5} + 2,5463 \times 10^{-5} + 2,2 \times 10^{-5} \\&= 0,0016\text{ s}\end{aligned}$$

Paket data yang sampai di RNC akan didekapsulasi dengan perhitungan sebagai berikut:



$$\begin{aligned}
 W_{\text{datagram (RNC)}} &= W_{\text{frame Node B}} - \text{header}_{\text{MAC}} \\
 &= 495 \text{ byte} - 3 \text{ byte} \\
 &= 492 \text{ byte}
 \end{aligned}$$

Sehingga besar *delay* dekapsulasi yang terjadi di RNC sesuai dengan persamaan (2.27) berikut:

$$\begin{aligned}
 t_{D(\text{RNC})} &= \frac{W_{\text{datagram (RNC)}} \times 8}{V_{\text{RNC}}} \\
 &= \frac{(492 \times 8) \text{ byte}}{155,52 \times 10^6 \text{ bps}} \\
 &= 2,5309 \times 10^{-5} \text{ s}
 \end{aligned}$$

Pada HS-DSCH FP (*Frame Protocol*), datagram pada RNC dienkapsulasi dengan menambahkan *header* FP dan CRC.

$$\begin{aligned}
 W_{\text{frame FP}} &= W_{\text{datagram (RNC)}} + \text{header}_{\text{FP}} + \text{CRC} \\
 &= 492 \text{ byte} + 7 \text{ byte} + 2 \text{ byte} \\
 &= 501 \text{ byte}
 \end{aligned}$$

Saat memasuki *layer* AAL2, paket data dibentuk menjadi paket CPS (*Common Part Sublayer*) dengan penambahan *header* sebesar 3 byte, sehingga panjang *frame* menjadi:

$$\begin{aligned}
 W_{\text{frame AAL2}} &= W_{\text{frame FP total}} + \text{header}_{\text{AAL2}} \\
 &= 501 \text{ byte} + 3 \text{ byte} \\
 &= 504 \text{ byte}
 \end{aligned}$$

Pada *layer* ATM, *frame* AAL2 tersebut kemudian akan dipecah menjadi blok-blok *payload* PDU SAR (*Segmentation and Reassembly Sublayer*) sebesar 48 byte. Masing-masing PDU SAR 48 byte sesuai dengan sel ATM tunggal. Sehingga banyaknya sel ATM yang terbentuk sesuai persamaan (2.36):

$$\begin{aligned}
 N_{\text{frame ATM2}} &= \frac{W_{\text{frame AAL2}}}{48 \text{ byte}} \\
 &= \frac{504 \text{ byte}}{48 \text{ byte}} \\
 &= 10,5
 \end{aligned}$$

Terdapat 10 *frame* berisi data 48 byte dan 1 *frame* berisi data sebesar ($0,5 \times 48 \text{ byte} = 24 \text{ byte}$) 24 byte. *Frame* yang berisi data 24 byte tersebut akan diberi *padding bytes* sebesar ($48 \text{ byte} - 24 \text{ byte} = 24 \text{ byte}$) 24 byte, sehingga terdapat 11 *frame*. Selanjutnya setiap sel ATM ditambahkan dengan *header* ATM sebesar 5 byte, sehingga panjang *frame* ATM menjadi 53 byte. Panjang *frame* di RNC yang siap ditransmisikan menuju 3.5G SGSN adalah sebagai berikut:



$$\begin{aligned}
 W_{frame RNC} &= W_{frame ATM2 total} = N_{frame ATM2} \times W_{frame ATM} \\
 &= 11 \times 53 \text{ byte} \\
 &= 583 \text{ byte}
 \end{aligned}$$

Sehingga besarnya *delay* enkapsulasi yang terjadi di RNC adalah:

$$\begin{aligned}
 t_{E(RNC)} &= \frac{W_{frame RNC} \times 8}{V_{RNC}} \\
 &= \frac{(583 \times 8) \text{ byte}}{155,52 \times 10^6 \text{ bps}} \\
 &= 2,999 \times 10^{-5} \text{ s}
 \end{aligned}$$

Delay transmisi untuk pengiriman paket data dari RNC ke 3.5G SGSN melalui *Iu Interface* adalah sebagai berikut:

$$\begin{aligned}
 t_{T(Iu Interface)} &= \frac{W_{frame RNC} \times 8}{V_{Iu Interface}} \\
 &= \frac{(583 \times 8) \text{ byte}}{155,52 \times 10^6 \text{ bps}} \\
 &= 2,999 \times 10^{-5} \text{ s}
 \end{aligned}$$

Delay propagasi antara RNC dan 3.5G SGSN adalah:

$$\begin{aligned}
 t_{P(RNC-3.5G SGSN)} &= \frac{N_{frame ATM2} \times R}{v} \\
 &= \frac{11 \times 3000}{2 \times 10^8} \\
 &= 1,65 \times 10^{-4} \text{ s}
 \end{aligned}$$

Sehingga *delay* proses total untuk arah *uplink* pada pentransmisian paket data dari RNC ke 3.5G SGSN didapat dengan perhitungan berikut ini.

$$\begin{aligned}
 t_{net(RNC-3.5G SGSN)} &= t_{D(RNC)} + t_{E(RNC)} + t_{T(Iu Interface)} + t_{P(RNC-3.5G SGSN)} \\
 &= 2,5309 \times 10^{-5} + 2,999 \times 10^{-5} + 2,999 \times 10^{-5} + 1,65 \times 10^{-4} \\
 &= 2,5029 \times 10^{-4} \text{ s}
 \end{aligned}$$

Paket data yang sampai di 3.5G SGSN akan didekapsulasi dengan perhitungan sebagai berikut:

$$\begin{aligned}
 W_{datagram(3.5G SGSN)} &= W_{frame RNC} - header_{ATM} - header_{AAL2} - header_{FP} - CRC \\
 &= 583 \text{ byte} - 5 \text{ byte} - 3 \text{ byte} - 7 \text{ byte} - 2 \text{ byte} \\
 &= 566
 \end{aligned}$$

Sehingga besar *delay* dekapsulasi yang terjadi di 3.5G SGSN adalah sebagai berikut:

$$\begin{aligned}
 t_{D(3.5G SGSN)} &= \frac{W_{datagram(3.5G SGSN)} \times 8}{V_{3.5G SGSN}} \\
 &= \frac{(566 \times 8) \text{ bit}}{100 \times 10^6 \text{ bps}} \\
 &= 4,528 \times 10^{-5} \text{ s}
 \end{aligned}$$

Selanjutnya *frame* pada 3.5G SGSN ditambahkan *header* GTP, UDP dan IP. Saat memasuki lapisan AAL5, paket data tersebut dibentuk menjadi CPCS PDU (*Common Part Convergence Sublayer Protocol Data Unit*) sesuai persamaan (2.22) yaitu:

$$\begin{aligned} W_{\text{frame AAL5}} &= W_{\text{datagram (3.5G SGSN)}} + \text{header}_{\text{GTP}} + \text{header}_{\text{IP}} + \text{header}_{\text{UDP}} + \text{header}_{\text{AAL5}} \\ &= 566 \text{ byte} + 8 \text{ byte} + 40 \text{ byte} + 8 \text{ byte} + 8 \text{ byte} \\ &= 630 \text{ byte} \end{aligned}$$

Pada *layer ATM*, *frame AAL5* tersebut kemudian akan dipecah menjadi blok-blok *payload PDU SAR (Segmentation and Reassembly Sublayer)* sebesar 48 byte. Sehingga banyaknya sel ATM yang terbentuk sesuai persamaan (2.23) adalah:

$$\begin{aligned} N_{\text{frame ATM}} &= \frac{W_{\text{frame AAL5}}}{48 \text{ byte}} \\ &= \frac{630 \text{ byte}}{48 \text{ byte}} \\ &= 13,125 \end{aligned}$$

Terdapat 13 *frame* berisi data 48 byte dan 1 *frame* berisi data sebesar $(0,125 \times 48 \text{ byte} = 6 \text{ byte})$. *Frame* yang berisi data 1 byte tersebut akan diberi *padding bytes* sebesar $(48 \text{ byte} - 6 \text{ byte} = 42 \text{ byte})$ 42 byte, sehingga terdapat 14 sel ATM. Selanjutnya setiap sel ATM diberi *header ATM* sebesar 5 byte sehingga panjang *frame ATM* menjadi 53 byte. Panjang *frame* di SGSN yang siap ditransmisikan menuju GGSN adalah sebagai berikut:

$$\begin{aligned} W_{\text{frame 3.5GSGSN}} &= W_{\text{frame ATM total}} = N_{\text{frame ATM}} \times W_{\text{frame ATM}} \\ &= 14 \times 53 \text{ byte} \\ &= 742 \text{ byte} \end{aligned}$$

Sehingga *delay enkapsulasi* yang terjadi di 3.5G SGSN dapat dihitung dengan persamaan (2.25):

$$\begin{aligned} t_{E(3.5G SGSN)} &= \frac{W_{\text{frame 3.5GSGSN}} \times 8}{V_{\text{SGSN-RNS}}} \\ &= \frac{(742 \times 8) \text{ byte}}{155,52 \times 10^6 \text{ bps}} \\ &= 3,8169 \times 10^{-5} \text{ s} \end{aligned}$$

Delay transmisi untuk pengiriman paket data dari 3.5G SGSN ke GGSN melalui Gn *Interface* adalah sebagai berikut:

$$t_{T(\text{Gn Interface})} = \frac{W_{\text{frame 3.5GSGSN}} \times 8}{V_{\text{Gn Interface}}}$$



$$\begin{aligned}
 &= \frac{(742 \times 8) \text{ byte}}{100 \times 10^6 \text{ bps}} \\
 &= 5,936 \times 10^{-5} \text{ s}
 \end{aligned}$$

Delay propagasi antara 3.5G SGSN dan GGSN adalah sebagai berikut:

$$\begin{aligned}
 t_{P(3.5G \text{ SGSN-GGSN})} &= \frac{N_{frame \text{ ATM}} \times R}{v} \\
 &= \frac{14 \times 2000}{2 \times 10^8} \\
 &= 1,4 \times 10^{-4} \text{ s}
 \end{aligned}$$

Teknologi yang digunakan pada 3.5G SGSN berupa *Fast Ethernet* dengan kecepatan 100 Mbps, maka dapat diperoleh kecepatan pelayanan node sebesar:

$$\begin{aligned}
 \mu_{3.5G \text{ SGSN}} &= \frac{C_{3.5G \text{ SGSN}}}{m} \\
 &= \frac{100 \times 10^6}{742 \times 8} \\
 &= 1,6846 \times 10^4 \text{ paket/s}
 \end{aligned}$$

Dengan nilai faktor utilisasi (ρ) diubah dari nilai 0,1 sampai 0,9 dengan kenaikan sebesar 0,1 maka diperoleh nilai kecepatan kedatangan paket pada node (λ) sebesar:

$$\begin{aligned}
 \lambda_{3.5G \text{ SGSN}} &= \mu_{3.5G \text{ SGSN}} \times \rho \\
 &= 1,6846 \times 10^4 \times 0,1 \\
 &= 0,1686 \times 10^4 \text{ bps}
 \end{aligned}$$

Nilai *delay* antrian pada 3.5G SGSN adalah sebagai berikut:

$$\begin{aligned}
 t_w(3.5G \text{ SGSN}) &= \frac{\lambda_{3.5G \text{ SGSN}}}{\mu_{3.5G \text{ SGSN}}(\mu_{3.5G \text{ SGSN}} - \lambda_{3.5G \text{ SGSN}})} + \frac{1}{\mu_{3.5G \text{ SGSN}}} \\
 &= \frac{0,1686 \times 10^4}{1,6846 \times 10^4(1,6846 \times 10^4 - 0,1686 \times 10^4)} + \frac{1}{1,6846 \times 10^4} \\
 &= 0,0001 \text{ s}
 \end{aligned}$$

Sehingga *delay* total proses pentransmisian paket data pada arah *uplink* dari 3.5G SGSN ke GGSN dengan faktor utilisasi 0,1 adalah sebagai berikut:

$$\begin{aligned}
 t_{net}(3.5G \text{ SGSN-GGSN}) &= t_D(3.5G \text{ SGSN}) + t_E(3.5G \text{ SGSN}) + t_{T(Gn \text{ Interface})} \\
 &\quad + t_{P(3.5G \text{ SGSN-GGSN})} + t_w(3.5G \text{ SGSN}) \\
 &= 4,528 \times 10^{-5} + 3,8169 \times 10^{-5} + 5,936 \times 10^{-5} \\
 &\quad + 1,4 \times 10^{-4} + 0,0001 \\
 &= 0,0004 \text{ s}
 \end{aligned}$$

Pada proses pentransmisian dari GGSN ke HLR untuk tujuan registrasi dan manajemen, paket data yang sampai di HLR akan mengalami proses dekapsulasi sebagai berikut:

$$\begin{aligned}
 W_{datagram \text{ (HLR)}} &= W_{frame \text{ 3.5GSGSN}} - header_{ATM} - header_{GTP} - header_{IP} \\
 &\quad - header_{UDP} - header_{AAL5} \\
 &= 742 \text{ byte} - 5 \text{ byte} - 8 \text{ byte} - 40 \text{ byte} - 8 \text{ byte} - 8 \text{ byte} \\
 &= 673 \text{ byte}
 \end{aligned}$$



Sehingga besar *delay* dekapsulasi yang terjadi di HLR adalah sebagai berikut:

$$\begin{aligned} t_{D(HLR)} &= \frac{W_{\text{datagram (HLR)}} \times 8}{V_{GGSN}} \\ &= \frac{(673 \times 8) \text{bit}}{100 \times 10^6 \text{bps}} \\ &= 5,384 \times 10^{-5} \text{s} \end{aligned}$$

Pada HLR *frame* akan disegmentasi untuk diterapkan MAP (*Mobile Application Part*) yang akan mengirimkan informasi pensinyalan yang berhubungan dengan *update* lokasi dan *handover* yang terjadi. Dalam proses tersebut *frame* akan dienkapsulasi dengan penambahan *header TCAP* (*Transaction Capabilities Application Part*), SCCP (*Signalling Connection Control Part*), dan CRC. Sehingga akan diperoleh panjang *frame* sebagai berikut:

$$\begin{aligned} W_{\text{frame HLR}} &= W_{\text{datagram (HLR)}} + \text{header}_{\text{MAP}} + \text{header}_{\text{TCAP}} + \text{header}_{\text{SCCP}} + \text{CRC} \\ &= 673 \text{ byte} + 13 \text{ byte} + 13 \text{ byte} + 13 \text{ byte} + 3 \text{ byte} \\ &= 715 \text{ byte} \end{aligned}$$

Delay enkapsulasi pada HLR adalah sebesar:

$$\begin{aligned} t_{E(HLR)} &= \frac{W_{\text{frame HLR}} \times 8}{V_{3.5G SGSN}} \\ &= \frac{(715 \times 8) \text{bit}}{100 \times 10^6 \text{bps}} \\ &= 5,72 \times 10^{-5} \text{s} \end{aligned}$$

Delay transmisi antara GGSN dan HLR yang dihubungkan oleh Gr *Interface* adalah sebesar:

$$\begin{aligned} t_{T(\text{Gr Interface})} &= \frac{W_{\text{frame HLR}} \times 8}{V_{\text{Ethernet}}} \\ &= \frac{(715 \times 8) \text{bit}}{1 \times 10^9 \text{bps}} \\ &= 5,72 \times 10^{-6} \text{s} \end{aligned}$$

Antara GGSN dan HLR dihubungkan dengan media udara dengan jarak 1 km. *Delay* propagasi antara GGSN dan HLR adalah sebagai berikut:

$$\begin{aligned} t_{P(\text{GGSN-HLR})} &= \frac{R}{v} \\ &= \frac{1000 \text{ m}}{3 \times 10^8 \text{ m/s}} \\ &= 3,3333 \times 10^{-6} \text{s} \end{aligned}$$

Sehingga *delay* proses total pentransmisi paket data dari GGSN ke HLR didapat dengan perhitungan berikut ini.



$$\begin{aligned}
 t_{net(HLR-GGSN)} &= t_{D(HLR)} + t_{E(HLR)} + t_{T(Gr\ Interface)} + t_{P(GGSN-HLR)} \\
 &= 5,384 \times 10^{-5} + 5,72 \times 10^{-5} + 5,72 \times 10^{-6} + 3,3333 \times 10^{-6} \\
 &= 1,2009 \times 10^{-4} \text{ s}
 \end{aligned}$$

Pada proses pentransmisian paket data dari HLR ke GGSN untuk arah *downlink* akan terjadi proses enkapsulasi, sehingga panjang *frame* pada HLR adalah sebesar:

$$\begin{aligned}
 W_{frame\ HLR} &= W_{datagram(HLR)} + header_{MAP} + header_{TCAP} + header_{SCCP} + CRC \\
 &= 673 \text{ byte} + 13 \text{ byte} + 13 \text{ byte} + 13 \text{ byte} + 3 \text{ byte} \\
 &= 715 \text{ byte}
 \end{aligned}$$

Sehingga *delay* enkapsulasi pada HLR adalah sebesar:

$$\begin{aligned}
 t_{E(HLR)} &= \frac{W_{frame\ HLR} \times 8}{V_{3.5G\ SGSN}} \\
 &= \frac{(715 \times 8) \text{ bit}}{100 \times 10^6 \text{ bps}} \\
 &= 5,72 \times 10^{-5} \text{ s}
 \end{aligned}$$

Delay transmisi antara HLR dan GGSN yang dihubungkan oleh *Gr Interface* adalah sebesar:

$$\begin{aligned}
 t_{T(Gr\ Interface)} &= \frac{W_{frame\ HLR} \times 8}{V_{Ethernet}} \\
 &= \frac{(715 \times 8) \text{ bit}}{1 \times 10^9 \text{ bps}} \\
 &= 5,72 \times 10^{-6} \text{ s}
 \end{aligned}$$

Antara HLR dan GGSN dihubungkan dengan media udara dengan jarak 1 km. *Delay* propagasi antara HLR dan GGSN adalah sebagai berikut:

$$\begin{aligned}
 t_{P(HLR-GGSN)} &= \frac{R}{v} \\
 &= \frac{1000 \text{ m}}{3 \times 10^8 \text{ m/s}} \\
 &= 3,3333 \times 10^{-6} \text{ s}
 \end{aligned}$$

Sehingga *delay* proses total pentransmisian paket data dari HLR ke GGSN adalah sebagai berikut:

$$\begin{aligned}
 t_{net(HLR-GGSN)} &= t_{E(HLR)} + t_{T(Gr\ Interface)} + t_{P(GGSN-HLR)} \\
 &= 5,72 \times 10^{-5} + 5,72 \times 10^{-6} + 3,3333 \times 10^{-6} \\
 &= 1,2009 \times 10^{-4} \text{ s}
 \end{aligned}$$

Pada GGSN data akan didekapsulasi dengan *delay* dekapsulasi sebesar:

$$t_{D(GGSN)} = \frac{W_{datagram} \times 8}{V_{GGSN}}$$



$$\begin{aligned}
 &= \frac{(377 \times 8)bit}{100 \times 10^6 bps} \\
 &= 3,016 \times 10^{-5} s
 \end{aligned}$$

Panjang *frame* GGSN yang siap ditransmisikan ke 3.5G SGSN adalah sebagai berikut:

$$\begin{aligned}
 W_{frame\ GGSN} &= W_{datagram} + header_{GTP} + header_{UDP} + header_{IP} \\
 &\quad + header_{Ethernet} + FCS \\
 &= 377 byte + 8 byte + 8 byte + 40 byte + 14 byte + 4 byte \\
 &= 451 byte
 \end{aligned}$$

Nilai *delay* enkapsulasi yang terjadi di GGSN adalah:

$$\begin{aligned}
 t_{E(GGSN)} &= \frac{W_{frame\ GGSN} \times 8}{V_{GGSN}} \\
 &= \frac{(451 \times 8)bit}{100 \times 10^6 bps} \\
 &= 3,608 \times 10^{-5} s
 \end{aligned}$$

Delay transmisi untuk pengiriman paket data dari GGSN ke 3.5G SGSN melalui Gn *Interface* untuk arah *downlink* adalah sebesar:

$$\begin{aligned}
 t_{T(Gn\ Interface)} &= \frac{W_{frame\ GGSN} \times 8}{V_{Gn\ Interface}} \\
 &= \frac{(451 \times 8)byte}{100 \times 10^6 bps} \\
 &= 3,608 \times 10^{-5} s
 \end{aligned}$$

Sedangkan *delay* propagasi antara GGSN dan 3.5G SGSN adalah sebagai berikut:

$$\begin{aligned}
 t_{P(GGSN-3.5G\ SGSN)} &= \frac{N_{frame\ GGSN} \times R}{v} \\
 &= \frac{1 \times 2000}{2 \times 10^8} \\
 &= 1 \times 10^{-5} s
 \end{aligned}$$

Interface yang digunakan pada GGSN berupa *Fast Ethernet* dengan kecepatan 100 Mbps dengan panjang data 451 byte, maka dapat diperoleh nilai kecepatan pelayanan *node* sebesar:

$$\begin{aligned}
 \mu_{GGSN} &= \frac{C_{GGSN}}{m} \\
 &= \frac{100 \times 10^6}{451 \times 8} \\
 &= 2,7716 \times 10^4 \text{ paket/s}
 \end{aligned}$$

Jika nilai faktor utilisasi (ρ) diubah dari nilai 0,1 sampai 0,9 dengan kenaikan sebesar 0,1 maka diperoleh nilai kecepatan kedatangan paket pada node (λ) sebesar:

$$\begin{aligned}
 \lambda_{GGSN} &= \mu_{GGSN} \times \rho \\
 &= 2,7716 \times 10^4 \times 0,1 \\
 &= 0,2772 \times 10^4 bps
 \end{aligned}$$



Sehingga dapat ditentukan nilai *delay* antrian pada GGSN sebagai berikut:

$$\begin{aligned} t_{w(GGSN)} &= \frac{\lambda_{GGSN}}{\mu_{GGSN}(\mu_{GGSN} - \lambda_{GGSN})} + \frac{1}{\mu_{GGSN}} \\ &= \frac{0,2772 \times 10^4}{2,7716 \times 10^4 (2,7716 \times 10^4 - 0,2772 \times 10^4)} + \frac{1}{2,7716 \times 10^4} \\ &= 0,0401 \times 10^{-3} \text{ s} \end{aligned}$$

Delay proses total pentransmisian paket data dari GGSN ke 3.5G SGSN didapat dengan perhitungan berikut ini.

$$\begin{aligned} t_{net(GGSN-3.5G SGSN)} &= t_{D(GGSN)} + t_{E(GGSN)} + t_{T(Gn Interface)} \\ &\quad + t_{P(GGSN-3.5G SGSN)} + t_{w(GGSN)} \\ &= 3,016 \times 10^{-5} + 3,608 \times 10^{-5} + 3,608 \times 10^{-5} \\ &\quad + 1 \times 10^{-5} + 0,0381 \times 10^{-3} \\ &= 0,1524 \times 10^{-3} \text{ s} \end{aligned}$$

Paket data yang diterima di 3.5G SGSN akan didekapsulasi dengan *delay* sebesar:

$$\begin{aligned} t_{D(3.5G SGSN)} &= \frac{W_{datagram} \times 8}{V_{3.5G SGSN}} \\ &= \frac{(377 \times 8) \text{ bit}}{100 \times 10^6 \text{ bps}} \\ &= 3,016 \times 10^{-5} \text{ s} \end{aligned}$$

Selanjutnya paket data ditambahkan *header* GTP, UDP dan IP. Saat memasuki lapisan AAL5, paket data tersebut dibentuk menjadi CPCS PDU (*Common Part Convergence Sublayer Protocol Data Unit*), sehingga panjang *frame* adalah:

$$\begin{aligned} W_{frame AAL5} &= W_{datagram} + header_{GTP} + header_{IP} + header_{UDP} + header_{AAL5} \\ &= 377 \text{ byte} + 8 \text{ byte} + 40 \text{ byte} + 8 \text{ byte} + 8 \text{ byte} \\ &= 441 \text{ byte} \end{aligned}$$

Pada *layer* ATM, *frame* AAL5 tersebut kemudian akan dipecah menjadi blok-blok *payload* PDU SAR (*Segmentation and Reassembly Sublayer*) sebesar 48 byte. Sehingga banyaknya sel ATM yang terbentuk adalah:

$$\begin{aligned} N_{frame ATM} &= \frac{W_{frame AAL5}}{48 \text{ byte}} \\ &= \frac{441 \text{ byte}}{48 \text{ byte}} \\ &= 9,1875 \end{aligned}$$

Terdapat 9 *frame* berisi data 48 byte dan 1 *frame* berisi data sebesar $(0,1875 \times 48 \text{ byte} = 9 \text{ byte})$. *Frame* yang berisi data 1 byte tersebut akan diberi *padding bytes* sebesar $(48 \text{ byte} - 9 \text{ byte} = 39 \text{ byte})$ 39 byte, sehingga terdapat 10 sel ATM. Selanjutnya setiap sel ATM diberi *header* ATM sebesar 5 byte sehingga panjang *frame*

ATM menjadi 53 byte. Panjang *frame* di SGSN yang siap ditransmisikan menuju RNC merupakan panjang *frame* ATM total yaitu:

$$\begin{aligned} W_{\text{frame } 3.5G \text{ SGSN}} &= W_{\text{frame } \text{ATM total}} = N_{\text{frame } \text{ATM}} \times W_{\text{frame } \text{ATM}} \\ &= 10 \times 53 \text{ byte} \\ &= 530 \text{ byte} \end{aligned}$$

Sehingga *delay* enkapsulasi yang terjadi di 3.5G SGSN adalah sebagai berikut:

$$\begin{aligned} t_{E(3.5G \text{ SGSN})} &= \frac{W_{\text{frame } 3.5G \text{ SGSN}} \times 8}{V_{\text{SGSN-RNS}}} \\ &= \frac{(530 \times 8) \text{ byte}}{155,52 \times 10^6 \text{ bps}} \\ &= 2,7263 \times 10^{-5} \text{ s} \end{aligned}$$

Delay transmisi yang terjadi pada Iu *Interface* adalah sebagai berikut:

$$\begin{aligned} t_{T(\text{Iu Interface})} &= \frac{W_{\text{frame } 3.5G \text{ SGSN}} \times 8}{V_{\text{Iu Interface}}} \\ &= \frac{(530 \times 8) \text{ byte}}{155,52 \times 10^6 \text{ bps}} \\ &= 2,7263 \times 10^{-5} \text{ s} \end{aligned}$$

Delay propagasi antara 3.5G SGSN dan RNC adalah:

$$\begin{aligned} t_{P(3.5G \text{ SGSN-RNC})} &= \frac{N_{\text{frame } \text{ATM}} \times R}{v} \\ &= \frac{10 \times 3000}{2 \times 10^8} \\ &= 1,5 \times 10^{-4} \text{ s} \end{aligned}$$

Teknologi yang digunakan pada 3.5G SGSN berupa *Fast Ethernet* dengan kecepatan 100 Mbps dengan panjang data 530 byte, maka diperoleh nilai kecepatan pelayanan *node* sebesar:

$$\begin{aligned} \mu_{3.5G \text{ SGSN}} &= \frac{C_{3.5G \text{ SGSN}}}{m} \\ &= \frac{100 \times 10^6}{530 \times 8} \\ &= 2,3585 \times 10^4 \text{ paket/s} \end{aligned}$$

Dengan faktor utilisasi (ρ) diubah dari nilai 0,1 sampai 0,9 dengan kenaikan sebesar 0,1 maka diperoleh nilai kecepatan kedatangan paket pada node (λ) sebesar:

$$\begin{aligned} \lambda_{3.5G \text{ SGSN}} &= \mu_{3.5G \text{ SGSN}} \times \rho \\ &= 2,3585 \times 10^4 \times 0,1 \\ &= 0,23585 \times 10^4 \text{ bps} \end{aligned}$$

Sehingga dapat ditentukan nilai *delay* antrian pada 3.5G SGSN yaitu:

$$t_w(3.5G \text{ SGSN}) = \frac{\lambda_{3.5G \text{ SGSN}}}{\mu_{3.5G \text{ SGSN}} (\mu_{3.5G \text{ SGSN}} - \lambda_{3.5G \text{ SGSN}})} + \frac{1}{\mu_{3.5G \text{ SGSN}}}$$

$$\begin{aligned}
 &= \frac{0,23585 \times 10^4}{2,3585 \times 10^4(2,3585 \times 10^4 - 0,23585 \times 10^4)} + \frac{1}{2,3585 \times 10^4} \\
 &= 0,0464 \times 10^{-3} \text{ s}
 \end{aligned}$$

Delay proses total pentransmisian paket data dari 3.5G SGSN ke RNC didapat dengan perhitungan berikut ini.

$$\begin{aligned}
 t_{net}(3.5G\ SGSN-RNC) &= t_D(3.5G\ SGSN) + t_E(3.5G\ SGSN) + t_T(Iu\ Interface) \\
 &\quad + t_P(3.5G\ SGSN-RNC) + t_w(3.5G\ SGSN) \\
 &= 3,016 \times 10^{-5} + 2,7263 \times 10^{-5} + 2,7263 \times 10^{-5} \\
 &\quad + 1,5 \times 10^{-4} + 0,0444 \times 10^{-3} \\
 &= 0,2811 \times 10^{-3} \text{ s}
 \end{aligned}$$

Pada RNC paket yang diterima dari 3.5G SGSN akan mengalami proses dekapsulasi dengan persamaan berikut:

$$\begin{aligned}
 W_{datagram\ (RNC)} &= W_{frame\ 3.5GSGSN} - header_{ATM} - header_{GTP} - header_{IP} \\
 &\quad - header_{UDP} - header_{AAL5} \\
 &= 530 \text{ byte} - 5 \text{ byte} - 8 \text{ byte} - 40 \text{ byte} - 8 \text{ byte} - 8 \text{ byte} \\
 &= 461 \text{ byte}
 \end{aligned}$$

Besar *delay* dekapsulasi yang terjadi di RNC adalah:

$$\begin{aligned}
 t_{D(RNC)} &= \frac{W_{datagram\ (RNC)} \times 8}{V_{RNC}} \\
 &= \frac{(461 \times 8) \text{ bit}}{155,52 \times 10^6 \text{ bps}} \\
 &= 2,3714 \times 10^{-5} \text{ s}
 \end{aligned}$$

Selanjutnya paket data aplikasi tersebut dienkapsulasi pada *layer* PDCP (*Packet Data Convergence Protocol*) dengan diberi *header* PDCP sebesar 1 byte.

$$\begin{aligned}
 W_{frame\ PDCP} &= W_{datagram\ (RNC)} + header_{PDCP} \\
 &= 461 \text{ byte} + 1 \text{ byte} \\
 &= 462 \text{ byte}
 \end{aligned}$$

Pada *layer* RLC, *frame* PDCP akan disegmentasi menjadi RLC PDU *fixed size* yaitu sebesar 40 byte, sehingga jumlah *frame* RLC yaitu:

$$\begin{aligned}
 N_{frame\ RLC} &= \frac{W_{frame\ PDCP}}{40 \text{ byte}} \\
 &= \frac{462 \text{ byte}}{40 \text{ byte}} \\
 &= 11,55
 \end{aligned}$$

Terdapat 11 *frame* RLC berisi data 40 byte dan 1 *frame* berisi data sebesar ($0,55 \times 40 \text{ byte} = 22 \text{ byte}$) 22 byte. *Frame* yang berisi data 22 byte tersebut akan diberi *padding bytes* sebesar ($40 \text{ byte} - 22 \text{ byte} = 18 \text{ byte}$) 18 byte, sehingga

terdapat 12 frame RLC. Setiap frame kemudian diberi *header* sebesar 2 byte sehingga panjang frame menjadi 42 byte. Panjang frame RLC total yang ditransmisikan menuju layer MAC-d yaitu:

$$\begin{aligned} W_{frame \ RLC \ total} &= N_{frame \ RLC} \times W_{frame \ RLC} \\ &= 12 \times 42 \text{ byte} \\ &= 504 \text{ byte} \end{aligned}$$

Selanjutnya pada layer MAC-d, RLC PDU akan disegmentasi menjadi MAC-d SDU (*Service Data Unit*) sebesar 42 byte. Maka jumlah frame MAC-d SDU adalah:

$$\begin{aligned} N_{frame \ MAC-d} &= \frac{W_{frame \ RLC \ total}}{42 \text{ byte}} \\ &= \frac{504 \text{ byte}}{42 \text{ byte}} \\ &= 12 \end{aligned}$$

Sedangkan panjang frame MAC-d adalah sebagai berikut:

$$\begin{aligned} W_{frame \ MAC-d} &= header_{MAC} + 42 \text{ byte} \\ &= 3 \text{ byte} + 42 \text{ byte} \\ &= 45 \text{ byte} \end{aligned}$$

Pada HS-DSCH FP (*Frame Protocol*), frame MAC-d dienkapsulasi dengan menambahkan *header* FP dan CRC.

$$\begin{aligned} W_{frame \ FP} &= W_{frame \ MAC-d} + header_{FP} + CRC \\ &= 45 \text{ byte} + 7 \text{ byte} + 2 \text{ byte} \\ &= 54 \text{ byte} \end{aligned}$$

Sehingga panjang frame HS-DSCH FP total yaitu:

$$\begin{aligned} W_{frame \ FP \ total} &= N_{frame \ MAC-d} \times W_{frame \ FP} \\ &= 12 \times 54 \text{ byte} \\ &= 648 \text{ byte} \end{aligned}$$

Saat memasuki layer AAL2, paket data dibentuk menjadi paket CPS (*Common Part Sublayer*) dengan penambahan *header* sebesar 3 byte, sehingga panjang frame menjadi:

$$\begin{aligned} W_{frame \ AAL2} &= W_{frame \ FP \ total} + header_{AAL2} \\ &= 648 \text{ byte} + 3 \text{ byte} \\ &= 651 \text{ byte} \end{aligned}$$

Pada layer ATM, frame AAL2 tersebut kemudian akan dipecah menjadi blok-blok payload PDU SAR (*Segmentation and Reassembly Sublayer*) sebesar 48 byte. Masing-masing PDU SAR 48 byte sesuai dengan sel ATM tunggal. Sehingga banyaknya sel ATM yang terbentuk adalah:



$$\begin{aligned}
 N_{frame \text{ ATM}2} &= \frac{W_{frame \text{ AAL2}}}{48 \text{ byte}} \\
 &= \frac{651 \text{ byte}}{48 \text{ byte}} \\
 &= 13,5625
 \end{aligned}$$

Terdapat 13 *frame* berisi data 48 *byte* dan 1 *frame* berisi data sebesar $(0,5625 \times 48 \text{ byte} = 27 \text{ byte})$ 27 *byte*. *Frame* yang berisi data 27 *byte* tersebut akan diberi *padding bytes* sebesar $(48 \text{ byte} - 27 \text{ byte} = 21 \text{ byte})$ 21 *byte*, sehingga terdapat 14 *frame*. Selanjutnya setiap sel ATM ditambahkan dengan *header ATM* sebesar 5 *byte*, sehingga panjang *frame* ATM menjadi 53 *byte*. Panjang *frame* di RNC yang siap ditransmisikan menuju UE merupakan panjang *frame* ATM total yaitu:

$$\begin{aligned}
 W_{frame \text{ RNC}} &= W_{frame \text{ ATM2 total}} = N_{frame \text{ ATM2}} \times W_{frame \text{ ATM}} \\
 &= 14 \times 53 \text{ byte} \\
 &= 742 \text{ byte}
 \end{aligned}$$

Sehingga besarnya *delay enkapsulasi* yang terjadi di RNC sebesar:

$$\begin{aligned}
 t_{E(RNC)} &= \frac{W_{frame \text{ RNC}} \times 8}{V_{RNC}} \\
 &= \frac{(742 \times 8) \text{ bit}}{155,52 \times 10^6 \text{ bps}} \\
 &= 3,8169 \times 10^{-5} \text{ s}
 \end{aligned}$$

Delay transmisi yang terjadi pada Iub *Interface* adalah sebagai berikut:

$$\begin{aligned}
 t_{T(Iub \text{ Interface})} &= \frac{W_{frame \text{ RNC}} \times 8}{V_{Iub \text{ Interface}}} \\
 &= \frac{(742 \times 8) \text{ bit}}{155,52 \times 10^6 \text{ bps}} \\
 &= 3,8169 \times 10^{-5} \text{ s}
 \end{aligned}$$

Delay propagasi antara RNC dan Node B adalah:

$$\begin{aligned}
 t_{P(RNC-Node \text{ B})} &= \frac{N_{frame \text{ ATM2}} \times R}{v} \\
 &= \frac{14 \times 400}{2 \times 10^8} \\
 &= 2,8 \times 10^{-5} \text{ s}
 \end{aligned}$$

Pada Node B paket data yang diterima dari RNC didekapsulasi dengan perhitungan sebagai berikut:

$$\begin{aligned}
 W_{datagram \text{ (Node B)}} &= W_{frame \text{ RNC}} - header_{ATM} - header_{AAL2} - header_{FP} - CRC \\
 &= 742 \text{ byte} - 5 \text{ byte} - 3 \text{ byte} - 7 \text{ byte} - 2 \text{ byte} \\
 &= 725 \text{ byte}
 \end{aligned}$$

Besarnya *delay dekapsulasi* yang terjadi di Node B yaitu:

$$\begin{aligned}
 t_{D(Node\ B)} &= \frac{W_{datagram\ (Node\ B)} \times 8}{V_{Node\ B}} \\
 &= \frac{(725 \times 8)bit}{155,52 \times 10^6 bps} \\
 &= 3,7294 \times 10^{-5}s
 \end{aligned}$$

Pada saat memasuki *layer MAC-hs*, paket data disegmentasi menjadi paket sebesar 42 byte, sehingga jumlah *frame* *MAC-hs* adalah:

$$\begin{aligned}
 N_{frame\ MAC-hs} &= \frac{W_{datagram\ (Node\ B)}}{42\ byte} \\
 &= \frac{725\ byte}{42\ byte} \\
 &= 17,2619
 \end{aligned}$$

Terdapat 17 *frame* berisi data 42 byte dan 1 *frame* berisi data sebesar ($0,2619 \times 42\ byte = 11\ byte$) 11 byte. *Frame* yang berisi data 11 byte tersebut akan diberi *padding bytes* sebesar ($42\ byte - 11\ byte = 31\ byte$) 31 byte, sehingga terdapat 18 *frame*. Panjang *frame* *MAC-hs* dapat dihitung berdasarkan persamaan (2.42):

$$\begin{aligned}
 W_{frame\ MAC-hs} &= header_{MAC} + 42byte \\
 &= 3\ byte + 42\ byte \\
 &= 45\ byte
 \end{aligned}$$

Sehingga panjang *frame* total pada Node B yaitu:

$$\begin{aligned}
 W_{frame\ Node\ B} &= N_{frame\ MAC-hs} \times W_{frame\ MAC-hs} \\
 &= 18 \times 45\ byte \\
 &= 810\ byte
 \end{aligned}$$

Delay enkapsulasi yang terjadi di Node B sebesar:

$$\begin{aligned}
 t_{E(Node\ B)} &= \frac{W_{frame\ Node\ B} \times 8}{V_{Node\ B}} \\
 &= \frac{(810 \times 8)bit}{155,52 \times 10^6 bps} \\
 &= 4,1667 \times 10^{-5}s
 \end{aligned}$$

Delay transmisi yang terjadi pada Uu *Interface* adalah sebagai berikut:

$$\begin{aligned}
 t_{T(Uu\ Interface)} &= \frac{W_{frame\ Node\ B} \times 8}{n \times V_{UE}} \\
 &= \frac{(810 \times 8)bit}{3 \times 3,6 \times 10^6 bps} \\
 &= 6 \times 10^{-4}s
 \end{aligned}$$

Delay propagasi antara Node B dan UE adalah:

$$\begin{aligned}
 t_{P(Node\ B-UE)} &= \frac{N_{frame\ MAC-hs} \times R}{v} \\
 &= \frac{18 \times 400}{3 \times 10^8} \\
 &= 2,4 \times 10^{-5} s
 \end{aligned}$$

Sehingga *delay* total proses pentransmisian paket data pada arah *downlink* dari RAN ke UE sebagai berikut:

$$\begin{aligned}
 t_{net(RAN-UE)} &= t_{D(RNC)} + t_{E(RNC)} + t_{T(Iub\ Interface)} + t_{P(RNC-Node\ B)} + t_{D(Node\ B)} \\
 &\quad + t_{E(Node\ B)} + t_{T(Uu\ Interface)} + t_{P(Node\ B-UE)} \\
 &= 2,3714 \times 10^{-5} + 3,8169 \times 10^{-5} + 3,8169 \times 10^{-5} + 2,8 \times 10^{-5} \\
 &\quad + 3,7294 \times 10^{-5} + 4,1667 \times 10^{-5} + 6 \times 10^{-4} + 2,4 \times 10^{-5} \\
 &= 8,3101 \times 10^{-4} s
 \end{aligned}$$

Berdasarkan perhitungan *delay handover* pada jaringan EDGE dan HSDPA di atas, maka dapat diperoleh kebutuhan *delay* untuk proses *handover* dari HSDPA ke EDGE dengan menggunakan MCS-6 dan faktor utilisasi 0,1, yang ditunjukkan pada Tabel 4.21 berikut ini.

Tabel 4.21 Hasil Perhitungan *Delay Handover* untuk MCS-6 dengan Faktor Utilisasi 0,1

No	Spesifikasi	Arah Pentransmisian	Kebutuhan <i>Delay</i> (s)
1.	Cell Change Order Message	RNC - UE	$8,3101 \cdot 10^{-4}$
2.	Routing Area Update Request	UE - 2,75G SGSN	0,0894
3.	SGSN Context Request	2,75G SGSN - 3,5G SGSN	0,0013
4.	SRNS Context Request	3,5G SGSN - RAN	$0,3312 \cdot 10^{-3}$
5.	SRNS Context Response	RAN - 3,5G SGSN	$2,5029 \cdot 10^{-4}$
6.	SGSN Context Response	3,5G SGSN - 2,75G SGSN	0,0009
7.	Security Function	HLR - UE	0,0015
8.	SGSN Context Acknowledge	2,75G SGSN - 3,5G SGSN	0,0013
9.	SRNS Data Forwarding Command	3,5G SGSN - RAN	$0,3312 \cdot 10^{-3}$
10.	Update PDP Contact Request	2,75G SGSN - GGSN	0,0011
11.	Update PDP Contact Response	GGSN - 2,75G SGSN	$0,1932 \cdot 10^{-3}$
12.	Update GPRS Location	2,75G SGSN - HLR	0,0012
13.	Cancel Location	3,5G SGSN - HLR	0,0009
14.	Bearer Release	3,5G SGSN - RAN	$0,3312 \cdot 10^{-3}$
15.	Bearer Release Response	RAN - 3,5G SGSN	$2,5029 \cdot 10^{-4}$
16.	Cancel Location Acknowledge	3,5G SGSN - HLR	0,0009
17.	Insert Subscriber Data	HLR - 2,75G SGSN	$0,3133 \cdot 10^{-3}$
18.	Insert Subscriber Data Ack.	2,75G SGSN - HLR	0,0012
19.	Update GPRS Location Ack.	HLR - 2,75G SGSN	$0,3313 \cdot 10^{-3}$
20.	Routing Area Update Accept	2,75G SGSN - UE	0,0508
21.	Routing Area Update Complete	UE - 2,75G SGSN	0,0894
<i>Delay Handover Total</i>			0,2397

Sumber: Perhitungan

Dengan cara yang sama, maka didapatkan *delay handover* total menggunakan MCS-1 sampai MCS-9 dan faktor utilisasi 0,1 sampai 0,9. Tabel 4.22 menunjukkan hasil perhitungan *delay handover* total tersebut.

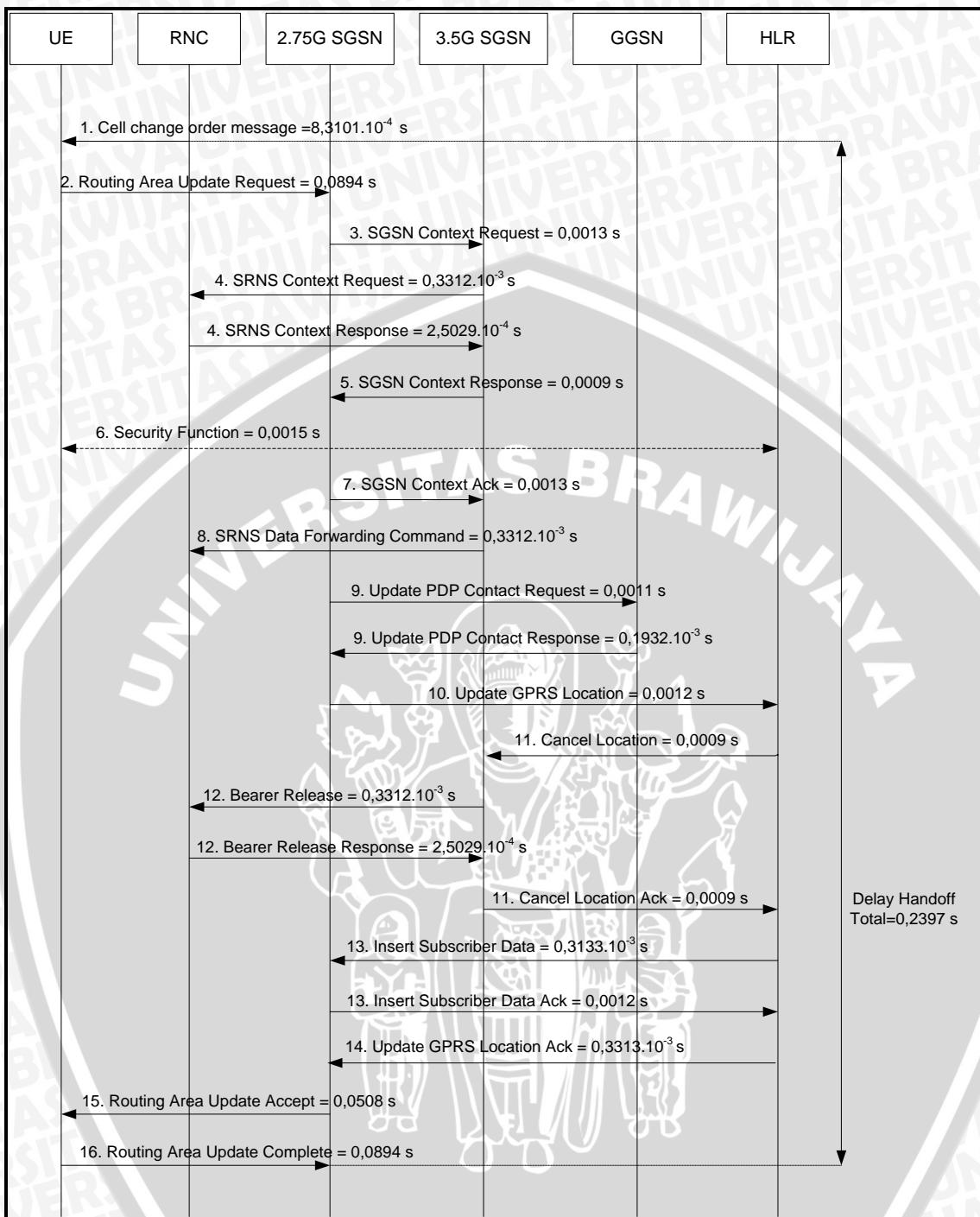
Tabel 4.22 Hasil Perhitungan *Delay Handover*

Faktor Utilisasi	Delay Handover EDGE (s)								
	MCS-1	MCS-2	MCS-3	MCS-4	MCS-5	MCS-6	MCS-7	MCS-8	MCS-9
0,1	2,4900	1,5492	0,8979	0,6414	0,4042	0,2397	0,1147	0,0836	0,0733
0,2	2,4904	1,5496	0,8982	0,6417	0,4045	0,2400	0,1151	0,0839	0,0737
0,3	2,4908	1,5500	0,8987	0,6422	0,4050	0,2405	0,1155	0,0844	0,0741
0,4	2,4915	1,5507	0,8993	0,6428	0,4056	0,2411	0,1162	0,0850	0,0748
0,5	2,4923	1,5515	0,9002	0,6437	0,4065	0,2420	0,1170	0,0859	0,0756
0,6	2,4936	1,5528	0,9015	0,6450	0,4078	0,2433	0,1183	0,0872	0,0769
0,7	2,4958	1,5550	0,9037	0,6472	0,4100	0,2455	0,1205	0,0894	0,0791
0,8	2,5002	1,5594	0,9080	0,6515	0,4143	0,2498	0,1249	0,0937	0,0835
0,9	2,5132	1,5724	0,9211	0,6646	0,4274	0,2629	0,1379	0,1068	0,0965

Sumber: Perhitungan

Delay handover pada prosedur proses *handover* dari HSDPA ke EDGE dengan menggunakan MCS-6 dan faktor utilisasi 0,1 ditunjukkan pada Gambar 4.16





Gambar 4.16 Delay Handover pada Prosedur Pensinyalan Handover dari HSDPA ke EDGE
Sumber: Analisis

Berdasarkan perhitungan *delay handover* pada jaringan EDGE dan HSDPA di atas, maka didapat *delay proses total*, *delay transmisi total*, *delay propagasi total*, serta *delay antrian total* sebagai berikut:

- *Delay proses total* pada proses *handover*, merupakan penjumlahan *delay enkapsulasi* dan *delay dekapsulasi* saat proses *handover* berlangsung. Hasil perhitungan *delay proses* saat *handover* ditunjukkan pada tabel 4.23.

Tabel 4.23 Hasil Perhitungan *Delay* Proses dengan Pengaruh *Handover*

MCS	<i>Delay</i> Proses Saat <i>Handover</i> (s)
MCS-1	0,9748
MCS-2	0,6054
MCS-3	0,3498
MCS-4	0,2492
MCS-5	0,1566
MCS-6	0,0921
MCS-7	0,0433
MCS-8	0,0311
MCS-9	0,0271

Sumber: Perhitungan

- *Delay* transmisi total pada proses *handover*, merupakan nilai total dari *delay* transmisi pada tiap *interface* saat terjadi *handover*. Hasil perhitungan *delay* transmisi saat *handover* ditunjukkan pada tabel 4.24.

Tabel 4.24 Hasil Perhitungan *Delay* Transmisi dengan Pengaruh *Handover*

MCS	<i>Delay</i> Transmisi Saat <i>Handover</i> (s)
MCS-1	1,5113
MCS-2	0,9399
MCS-3	0,5442
MCS-4	0,3883
MCS-5	0,2438
MCS-6	0,1437
MCS-7	0,0676
MCS-8	0,0486
MCS-9	0,0423

Sumber: Perhitungan

- *Delay* propagasi total pada proses *handover*, merupakan penjumlahan total *delay* propagasi pada *interface* pada saat *handover*. Tabel 4.25 menunjukkan hasil perhitungan *delay* propagasi total saat *handover*.
- *Delay* antrian total pada proses *handover*, merupakan penjumlahan total *delay* antrian pada PND Internet, GGSN, dan SGSN pada jaringan EDGE maupun HSDPA saat *handover*. Tabel 4.26 menunjukkan hasil perhitungan *delay* propagasi total saat *handover*.

Tabel 4.25 Hasil Perhitungan *Delay* Propagasi Total dengan Pengaruh *Handover*

MCS	<i>Delay</i> Transmisi Saat <i>Handover</i> (s)
MCS-1	0,001587
MCS-2	0,001579
MCS-3	0,001573
MCS-4	0,001570
MCS-5	0,001566
MCS-6	0,001563
MCS-7	0,001559
MCS-8	0,001558
MCS-9	0,001557

Sumber: Perhitungan

Tabel 4.26 Hasil Perhitungan *Delay Antrian Total* dengan Pengaruh *Handover*

Faktor Utilisasi	<i>Delay Antrian Total</i> (s)
0,1	0,0023
0,2	0,0027
0,3	0,0031
0,4	0,0038
0,5	0,0046
0,6	0,0059
0,7	0,0081
0,8	0,0125
0,9	0,0255

Sumber: Perhitungan

Perhitungan selanjutnya setelah didadapkan *delay handover* adalah perhitungan pengaruh *delay handover* terhadap perubahan *throughput* jaringan. Berdasarkan hasil analisis *delay antrian total* jaringan EDGE pada tabel 4.13 maka proses perhitungan dapat dijelaskan sebagai berikut:

Dalam proses *handover*, proses pentransmisi data akan mengalami perpindahan dari jaringan HSDPA dengan *throughput* $3,5634 \cdot 10^3$ kbps, ke jaringan EDGE dengan *throughput* 76,836 kbps. Pengaruh *delay handover* pada *throughput* di jaringan EDGE untuk MCS-1 dengan faktor utilisasi 0,9 dapat dianalisis dengan perhitungan berikut ini.

Probabilitas data yang diterima salah dapat dinyatakan dengan :

$$\begin{aligned}\rho_{mobile\ tv} &= P_{size} \times \rho_b \\ &= 377 \times 8 \times 10^{-9} \\ &= 3,016 \times 10^{-6}\end{aligned}$$

Probabilitas *packet loss* pada PDN Internet, GGSN, 2.75G SGSN, dan GERAN yaitu:

$$\begin{aligned}\rho_{PDN\ intrnt} &= (l - l')\rho_b \\ &= 443 \times 8 \times 10^{-9} \\ &= 3,544 \times 10^{-6}\end{aligned}$$

$$\begin{aligned}\rho_{GGSN} &= (l - l')\rho_b \\ &= 439 \times 8 \times 10^{-9}\end{aligned}$$

$$\begin{aligned}\rho_{2.75G\ SGSN} &= (l - l')\rho_b \\ &= 502 \times 8 \times 10^{-9} \\ &= 4,016 \times 10^{-6}\end{aligned}$$

$$\begin{aligned}\rho_{GERAN} &= (l - l')\rho_b \\ &= 502 \times 8 \times 10^{-9} \\ &= 4,016 \times 10^{-6}\end{aligned}$$

Probabilitas *frame error* pada *air interface* sebesar :

$$\begin{aligned}\rho_{AI} &= 1 - (l - P_e)^{l-l'} \\ &= 1 - (1 - 10^{-4})^{456} \\ &= 0,0446\end{aligned}$$

Pada jaringan EDGE, probabilitas *packet loss* adalah sebesar:

$$\begin{aligned}\rho_{net\ (EDGE)} &= 1 - [(1 - \rho_{AI})(1 - \rho_{GERAN})^h(1 - \rho_{SGSN})(1 - \rho_{GGSN})(1 - \rho_{PDN\ Internet})] \\ &= 1 - [(1 - 0,0446)(1 - 4,016 \times 10^{-6})^1(1 - 4,016 \times 10^{-6}) \\ &\quad (1 - 3,512 \times 10^{-6})(1 - 3,544 \times 10^{-6})] \\ &= 0,0446\end{aligned}$$

Probabilitas *packet loss* total pada jaringan EDGE adalah sebagai berikut:



$$\begin{aligned}\rho_{tot(EDGE)} &= 1 - [(1 - \rho_{net(EDGE)})(1 - \rho_{mobile\ tv})] \\ &= 1 - [(1 - 0,0446)(1 - 3,016 \times 10^{-6})] \\ &= 0,0446\end{aligned}$$

Berdasarkan hasil analisis *delay* total saat terjadi *handover* pada tabel 4.23 sampai 4.26, maka dapat diperoleh:

Waktu minimum transmisi (t_l) untuk MCS-6 dengan data *rate* 8,9 kbps adalah:

$$\begin{aligned}t_l &= \frac{(Pl_{frame} + H_{frame}) \times 8}{C_{trans}} \\ &= \frac{456}{29,7 \times 10^3} \\ &= 0,0154\ s\end{aligned}$$

Delay propagasi total dan *delay* proses total untuk MCS-6 dan faktor utilisasi 0,1 yaitu:

$$\begin{aligned}t_p &= \frac{t_{P(EDGE)}}{N_{frame\ RLCradioblock}} \\ &= \frac{0,0016\ s}{6,0811} \\ &= 0,2570 \times 10^{-3}\ s\end{aligned}$$

$$\begin{aligned}t_{proses} &= \frac{t_{proc(HO)}}{N_{frame\ RLCradioblock}} + \frac{t_{W(HO)}}{N_{frame\ RLCradioblock}} \\ &= \frac{0,0921}{6,0811} + \frac{0,0023}{6,0811} \\ &= 0,0155\ s\end{aligned}$$

Sehingga *fixed timeout interval* adalah sebesar:

$$\begin{aligned}t_{out} &= 2t_p + 2t_l + 2t_{proses} \\ &= (2 \times 0,2570 \times 10^{-3}) + (2 \times 0,0154) + (2 \times 0,0155) \\ &= 0,0623\ s\end{aligned}$$

Maka konstanta dapat diperoleh sesuai dengan persamaan (2.87) yaitu:

$$\begin{aligned}\alpha &= 1 + \frac{t_{out}}{t_l} \\ &= 1 + \frac{0,0623}{0,0154} \\ &= 5,0556\ s\end{aligned}$$

Sehingga besar *throughput* pada jaringan EDGE dengan menggunakan MCS-6, dan faktor utilisasi 0,1 akibat terjadinya proses *handover* adalah sebagai berikut:

$$\begin{aligned}\lambda &= \frac{(1 - \rho_{tot})}{t_l[1 + (\alpha - 1)\rho_{tot}]} \\ &= \frac{(1 - 0,0446)}{0,0154[1 + (5,0556 - 1)0,0446]} \\ &= 52,6953\ paket/s\end{aligned}$$



$$\begin{aligned}
 &= 52,6953 \times 456 \\
 &= 2,4029 \times 10^4 \text{ bps}
 \end{aligned}$$

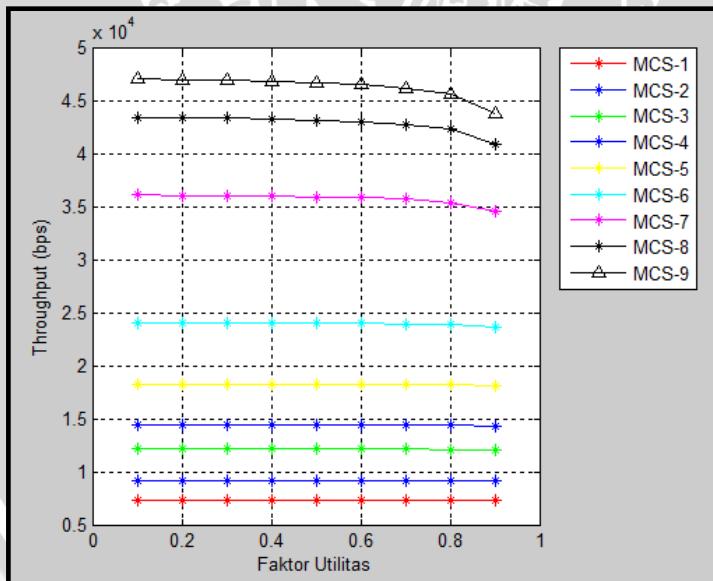
Dengan cara perhitungan yang sama, didapatkan nilai *throughput* untuk masing-masing MCS dengan faktor utilisasi 0,1 sampai 0,9 dengan kenaikan 0,1 pada tiap step, yang ditunjukkan dalam tabel 4.27.

Tabel 4.27 Perubahan nilai *Throughput* karena Pengaruh *Handover*

Faktor Utilisasi	<i>Throughput</i> Jaringan EDGE <i>Handover</i> (bps)								
	MCS-1	MCS-2	MCS-3	MCS-4	MCS-5	MCS-6	MCS-7	MCS-8	MCS-9
0,1	$7,2521 \cdot 10^3$	$9,2038 \cdot 10^3$	$1,2126 \cdot 10^4$	$1,4393 \cdot 10^4$	$1,8263 \cdot 10^4$	$2,4029 \cdot 10^4$	$3,6055 \cdot 10^4$	$4,3395 \cdot 10^4$	$4,6994 \cdot 10^4$
0,2	$7,2519 \cdot 10^3$	$9,2034 \cdot 10^3$	$1,2125 \cdot 10^4$	$1,4392 \cdot 10^4$	$1,8260 \cdot 10^4$	$2,4022 \cdot 10^4$	$3,6031 \cdot 10^4$	$4,3352 \cdot 10^4$	$4,6941 \cdot 10^4$
0,3	$7,2517 \cdot 10^3$	$9,2029 \cdot 10^3$	$1,2124 \cdot 10^4$	$1,4390 \cdot 10^4$	$1,8256 \cdot 10^4$	$2,4013 \cdot 10^4$	$3,6000 \cdot 10^4$	$4,3298 \cdot 10^4$	$4,6871 \cdot 10^4$
0,4	$7,2513 \cdot 10^3$	$9,2023 \cdot 10^3$	$1,2122 \cdot 10^4$	$1,4387 \cdot 10^4$	$1,8250 \cdot 10^4$	$2,4001 \cdot 10^4$	$3,5959 \cdot 10^4$	$4,3226 \cdot 10^4$	$4,6780 \cdot 10^4$
0,5	$7,2509 \cdot 10^3$	$9,2013 \cdot 10^3$	$1,2120 \cdot 10^4$	$1,4384 \cdot 10^4$	$1,8243 \cdot 10^4$	$2,3984 \cdot 10^4$	$3,5901 \cdot 10^4$	$4,3126 \cdot 10^4$	$4,6652 \cdot 10^4$
0,6	$7,2502 \cdot 10^3$	$9,1999 \cdot 10^3$	$1,2117 \cdot 10^4$	$1,4378 \cdot 10^4$	$1,8232 \cdot 10^4$	$2,3959 \cdot 10^4$	$3,5816 \cdot 10^4$	$4,2976 \cdot 10^4$	$4,6461 \cdot 10^4$
0,7	$7,2490 \cdot 10^3$	$9,1976 \cdot 10^3$	$1,2112 \cdot 10^4$	$1,4369 \cdot 10^4$	$1,8214 \cdot 10^4$	$2,3917 \cdot 10^4$	$3,5674 \cdot 10^4$	$4,2728 \cdot 10^4$	$4,6146 \cdot 10^4$
0,8	$7,2468 \cdot 10^3$	$9,1929 \cdot 10^3$	$1,2101 \cdot 10^4$	$1,4351 \cdot 10^4$	$1,8177 \cdot 10^4$	$2,3833 \cdot 10^4$	$3,5394 \cdot 10^4$	$4,2242 \cdot 10^4$	$4,5530 \cdot 10^4$
0,9	$7,2399 \cdot 10^3$	$9,1788 \cdot 10^3$	$1,2069 \cdot 10^4$	$1,4298 \cdot 10^4$	$1,8067 \cdot 10^4$	$2,3586 \cdot 10^4$	$3,4579 \cdot 10^4$	$4,0847 \cdot 10^4$	$4,3776 \cdot 10^4$

Sumber: Perhitungan

Berdasarkan tabel 4.27, maka didapatkan grafik perbandingan nilai *throughput* pada jaringan EDGE yang tidak dipengaruhi oleh proses *handover* untuk MCS-1 sampai MCS-9 dengan faktor utilisasi 0,1 sampai 0,9, yang ditunjukkan pada Gambar 4.58.



Gambar 4.17 Grafik Hubungan *Throughput Handover* Jaringan EDGE terhadap Faktor Utilisasi

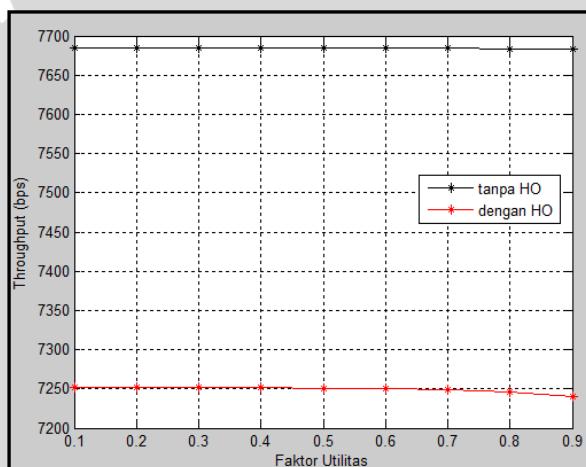
Sumber: Analisis

Berdasarkan analisis yang dilakukan secara matematis maupun grafis dapat dijelaskan bahwa pada pengiriman data dari PDN *Internet* ke UE melalui jaringan EDGE pada saat terjadi *handover* menunjukkan bahwa nilai *throughput* dipengaruhi oleh beberapa faktor, antara lain besar data yang dikirimkan, penggunaan *modulation*

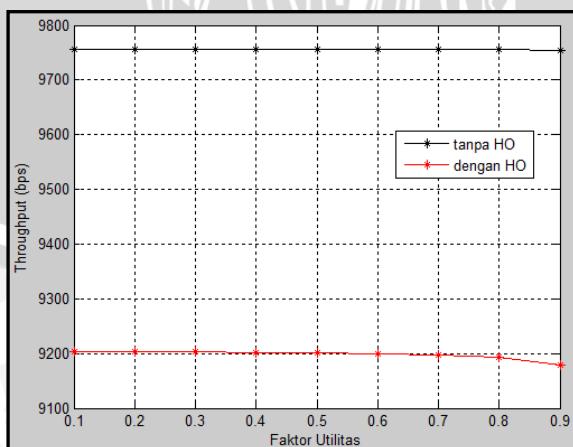
and Coding Scheme (MCS), *delay* yang terjadi saat *handover* berlangsung sesuai dengan prosesur pensinyalan *handover* dari HSDPA ke EDGE (*delay* enkapsulasi dan dekapsulasi, *delay* antrian, serta *delay* transmisi), faktor utilisasi, dan *packet loss*.

Semakin besar MCS yang digunakan, maka *throughput* akibat proses *handover* semakin besar. Nilai *throughput* pada jaringan EDGE dengan pengaruh *handover* akan semakin kecil dengan bertambahnya faktor utilisasi. Akan tetapi karena perubahan yang terjadi relatif kecil, pada analisis grafis perubahan *throughput* terlihat tidak begitu jelas.

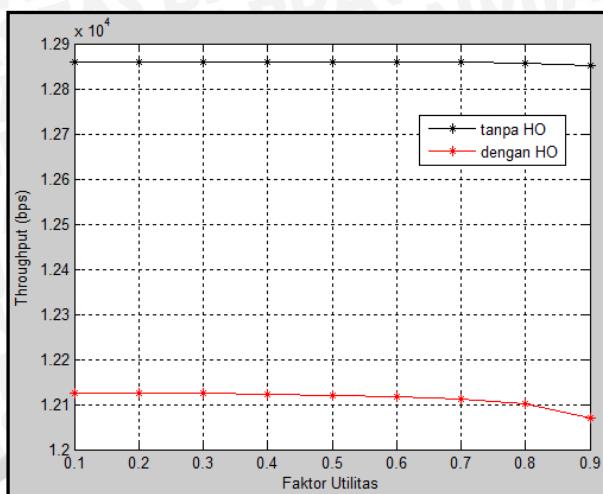
Berdasarkan tabel 4.17 dan 4.27, maka didapatkan grafik perbandingan nilai *throughput* pada jaringan EDGE yang tidak dipengaruhi oleh proses *handover* dengan *throughput* akibat pengaruh proses *handover*, untuk MCS-1 sampai MCS-9 dengan faktor utilisasi 0,1 sampai 0,9, yang ditunjukkan pada Gambar 4.18 sampai Gambar 4.26.



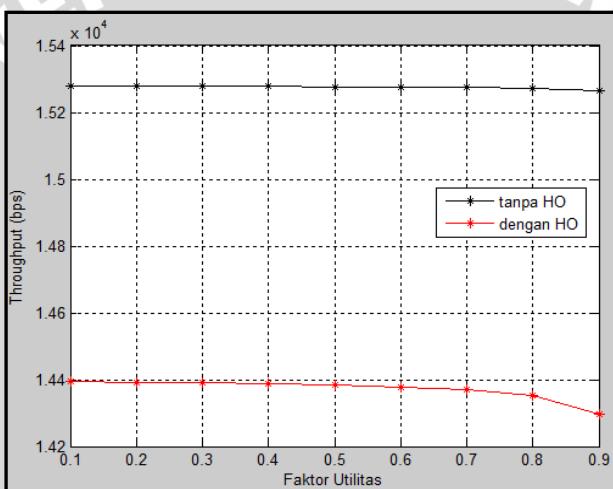
Gambar 4.18 Grafik Perbandingan *Throughput* Jaringan EDGE MCS-1 dengan Pengaruh *Handover* dan Tanpa Pengaruh *Handover* terhadap Faktor Utilisasi
Sumber: Analisis



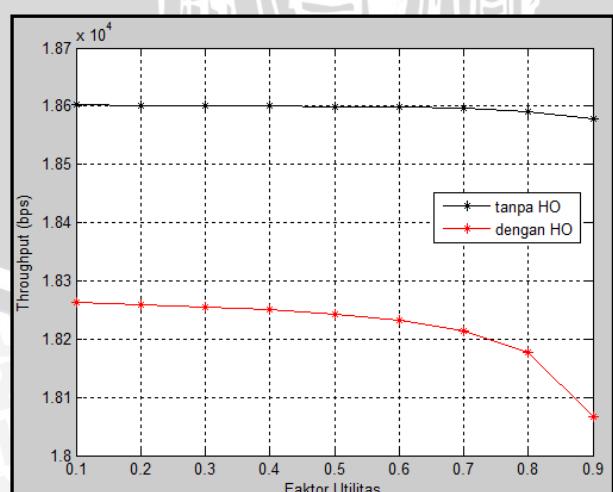
Gambar 4.19 Grafik Perbandingan *Throughput* Jaringan EDGE MCS-2 dengan Pengaruh *Handover* dan Tanpa Pengaruh *Handover* terhadap Faktor Utilisasi
Sumber: Analisis



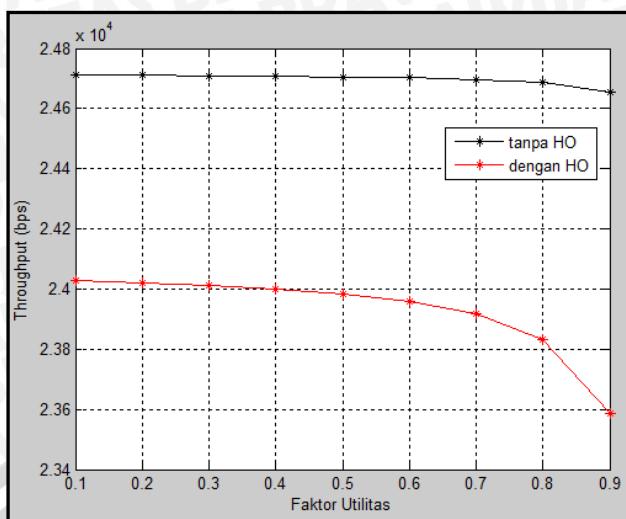
Gambar 4.20 Grafik Perbandingan *Throughput* Jaringan EDGE MCS-3 dengan Pengaruh *Handover* dan Tanpa Pengaruh *Handover* terhadap Faktor Utilisasi
Sumber: Analisis



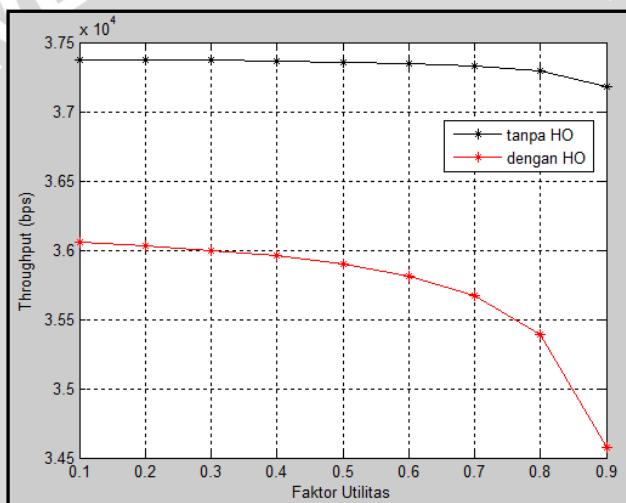
Gambar 4.21 Grafik Perbandingan *Throughput* Jaringan EDGE MCS-4 dengan Pengaruh *Handover* dan Tanpa Pengaruh *Handover* terhadap Faktor Utilisasi
Sumber: Analisis



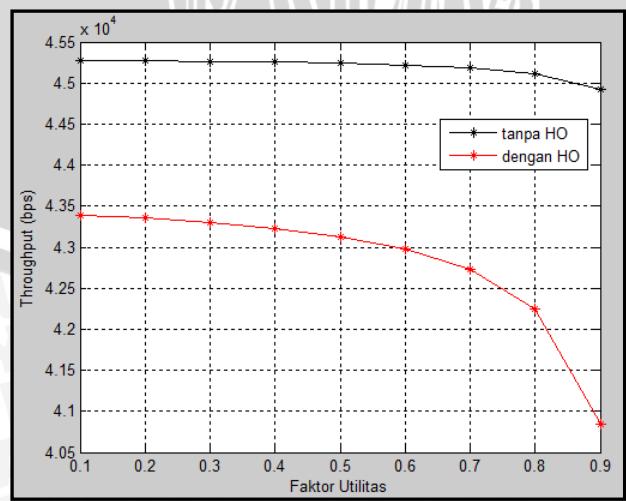
Gambar 4.22 Grafik Perbandingan *Throughput* Jaringan EDGE MCS-5 dengan Pengaruh *Handover* dan Tanpa Pengaruh *Handover* terhadap Faktor Utilisasi
Sumber: Analisis



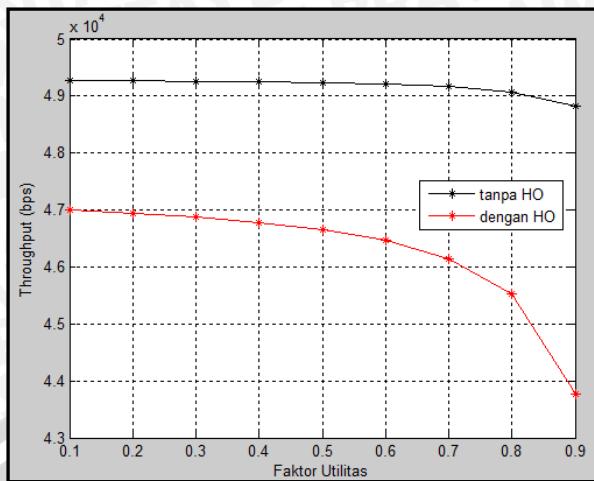
Gambar 4.23 Grafik Perbandingan *Throughput* Jaringan EDGE MCS-6 dengan Pengaruh *Handover* dan Tanpa Pengaruh *Handover* terhadap Faktor Utilisasi
Sumber: Analisis



Gambar 4.24 Grafik Perbandingan *Throughput* Jaringan EDGE MCS-7 dengan Pengaruh *Handover* dan Tanpa Pengaruh *Handover* terhadap Faktor Utilisasi
Sumber: Analisis



Gambar 4.25 Grafik Perbandingan *Throughput* Jaringan EDGE MCS-8 dengan Pengaruh *Handover* dan Tanpa Pengaruh *Handover* terhadap Faktor Utilisasi
Sumber: Analisis



Gambar 4.26 Grafik Perbandingan *Throughput* Jaringan EDGE MCS-9 dengan Pengaruh *Handover* dan Tanpa Pengaruh *Handover* terhadap Faktor Utilisasi
Sumber: Analisis

Berdasarkan analisis baik secara matematis maupun secara grafis, maka dapat dijelaskan bahwa:

- Hasil perhitungan menunjukkan bahwa nilai *delay handover* menuju jaringan EDGE terbesar terjadi pada MCS-1 dengan faktor utilisasi 0,9 yaitu sebesar 2,5132 detik. Sedangkan nilai *delay handover* terkecil terdapat pada MCS-9 dengan faktor utilisasi 0,1 yaitu 0,0733 detik.
- Pada MCS-1 sampai MCS-5 jaringan EDGE tidak memenuhi standar untuk penerapan aplikasi *mobile TV*, karena *delay* yang terjadi lebih besar dari *delay* toleransi untuk layanan *mobile TV* yaitu 350 ms. Sehingga *handover* hanya dapat dilakukan pada MCS-6 sampai MCS-9.
- Berdasarkan perhitungan dan analisa grafik pada Gambar 4.59 sampai Gambar 4.67 menunjukkan adanya penurunan yang signifikan pada nilai *throughput* di jaringan EDGE sebagai pengaruh *delay handover* selama terjadinya proses *handover* dari jaringan HSDPA ke EDGE.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan perhitungan dan analisis tentang pengaruh *handover* pada jaringan HSDPA ke EDGE untuk layanan *mobile TV*, maka diperoleh kesimpulan sebagai berikut:

1. Pada jaringan EDGE, *delay end-to-end* terbesar terdapat pada penggunaan MCS-1 dengan faktor utilisasi 0,9 yaitu 0,4006 detik, sedangkan *delay total* terkecil terjadi pada MCS-9 untuk faktor utilisasi 0,1 yaitu 0,0210 detik. Sehingga semakin besar MCS, *delay end-to-end* semakin kecil.
2. Pada jaringan HSDPA, didapatkan *delay total* terbesar terdapat pada faktor utilisasi 0,9 yaitu 0,0038 detik, sedangkan *delay total* terkecil terdapat pada faktor utilisasi 0,1 yaitu 0,0031 detik. Sehingga semakin besar faktor utilisasi, maka *delay end-to-end* juga semakin besar.
3. Pada jaringan EDGE saat tidak terjadi *handover*, aplikasi *mobile TV* tidak dapat diterapkan untuk penggunaan MCS-1, karena *delay* yang terjadi melebihi batas toleransi untuk layanan *mobile TV* yaitu 350 ms. Sedangkan pada jaringan HSDPA *delay total* yang terjadi masih layak untuk penerapan aplikasi *mobile TV* dengan AMR-WB sebagai *codec* audio dan H.264 sebagai *codec* video.
4. Pada Jaringan EDGE, nilai *throughput* terbesar terjadi pada MCS-9 dengan faktor utilisasi 0,1 yaitu sebesar $4,9277 \cdot 10^4$ bps. Sedangkan nilai *throughput* terkecil terdapat pada penggunaan MCS-1 dengan faktor utilisasi 0,9 yaitu sebesar $7,6828 \cdot 10^3$ bps. Pada jaringan HSDPA kenaikan faktor utilisasi yang menyebabkan bertambahnya *delay*, sehingga nilai *throughput* terbesar terdapat pada faktor utilisasi 0,1 sebesar $3,5636 \cdot 10^6$ bps, sedangkan *throughput* terkecil terjadi pada faktor utilisasi 0,9 sebesar $3,5631 \cdot 10^6$ bps.
5. Berdasarkan proses *handover* dari jaringan HSDPA ke EDGE melalui standar prosedur-prosedur yang telah disebutkan, dengan mengabaikan *throughput* pada MCS-1 sampai MCS-5, dikarenakan *delay* yang tidak dapat ditoleransi untuk layanan *mobile TV*, maka dapat disimpulkan sebagai berikut:
 - a) Hasil analisis menunjukkan bahwa nilai *delay handover* menuju jaringan EDGE terbesar terjadi pada MCS-6 dengan faktor utilisasi 0,9 yaitu sebesar 0,2629 detik. Sedangkan nilai *delay handover* terkecil terdapat pada MCS-9 dengan faktor utilisasi 0,1 yaitu 0,0733 detik. Dengan demikian proses *handover* dari

jaringan HSDPA ke EDGE dengan aplikasi *mobile TV* yang menggunakan AMR-WB sebagai *codec* audio dan H.264 sebagai *codec* video, masih dapat dilakukan untuk MCS-6 sampai MCS-9, karena *delay* yang terjadi tidak melebihi batas toleransi untuk layanan *mobile TV* yaitu 350 ms.

- b) Proses *handover* menuju MCS-6 hingga MCS-9 menunjukkan adanya penurunan yang signifikan pada nilai *throughput* di jaringan EDGE sebagai pengaruh *delay handover* selama terjadinya proses *handover* dari jaringan HSDPA ke EDGE. Hal ini dikarenakan nilai *delay* yang berbanding terbalik dengan *throughput*. Semakin besar *delay* yang dibutuhkan pada proses *handover*, maka nilai *throughput* yang terjadi pada sistem yang dituju akan semakin kecil.

5.2 Saran

Saran yang diberikan berdasarkan analisis yang telah dilakukan pada skripsi ini adalah:

1. Pembahasan dapat dikembangkan pada *handover* untuk tujuan sebaliknya, yaitu dari *handover* dari jaringan EDGE ke HSDPA.
2. Analisis lebih lanjut dapat dikembangkan dengan memperhitungkan efek ping-pong jaringan.
3. Analisis dalam skripsi ini dapat dikembangkan dengan menggunakan jenis *codec* audio dan video yang berbeda pada *mobile TV*.



DAFTAR PUSTAKA

- Anonim. 2003. *Nokia HSDPA Solution*. White Paper. Nokia.
- Anonim. 2009. *Technical Overview and Performance of HSDPA and Mobile WiMax*. White Paper. Ericsson.
- Ekowati R, Linda. 2008. *Penerapan HSDPA pada Jaringan WCDMA*. Malang: Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, 2008.
- Faria et. al. 2006. *DVB-H: Digital Video Broadcasting Services to Handheld Devices White Paper*. http://www.dvb-h.org/technology/white_paper_dvb-h.pdf. (diakses tanggal 30 Januari 2010).
- Fatimah. 2008. *Pengaruh Handover pada Sistem Komunikasi HSDPA ke GPRS untuk Layanan MoblieTV*. Malang: Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, 2008.
- Halonen, Timo. 2003. *GSM, GPRS, and EDGE Performance*. England: Wiley and Sons, Ltd.
- Harri Holma and Antii Toskala. 2004. *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*. England: John Wiley and Sons, Ltd.
- Harri Holma and Antii Toskala. 2006. *HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communication*. England: John Wiley and Sons, Ltd.
- Kumar, Amitabh. 2007. *Mobile TV: DVB-H, DMB, 3G Systems and Rich Media Applications*. Burlington: Focal Press.
- Kumar, Anurag. 2008. *Wireless Networking*. Burlington: Morgan Kaufmann Publishers.
- Kurniawan, Uke. 2008. *Konsep Teknologi Seluler*. Jakarta: Penerbit Informatika.
- Rysavy, Peter. 2006. *EDGE, HSPA, & LTE*. White Paper. 3G Americas.
- Seurre, Emmanuel. 2003. *EDGE for Mobile Internet*. Norwood: Artech house.
- Schwartz, Mischa. 1987. *Telecommunication Network*. Addison-Wesley.
- Sun, Zhili. 2005. *Satelite Networking Principles and Protocols*. England: John Wiley and Sons, Ltd.
- Tapia, Pablo. 2009. *HSPA Performance and Evolution*. England: John Wiley and Sons, Ltd.

LAMPIRAN

UNIVERSITAS BRAWIJAYA



LISTING PROGRAM PERHITUNGAN DELAY

```
% PERHITUNGAN DELAY PROSES ENKAPSULASI DAN DEKAPSULASI
W_mob_tv=369;
h_UDP=8;
h_IP=40;
h_ether=14;
FCS=4;
V_PDN_int=10^9;
V_GGSN=10^8;
h_GTP=8;
V_ESGSN=15*(10^6);
h_SNDCP=4;
FCS_1=3;
h_LLC=4;
h_BSSGP=48;
h_NS=4;
V_PCU=14*31*64000;
MCS=[176, 224, 296, 352, 448, 592, 896, 1088, 1184]
jml_frame_LLC=1;
RLC_radioblk=456;
V_PCU2=14*150*16000
code_rate=[1/2, 0.64, 0.83, 1, 1/3, 1/2, 3/4, 0.90, 1]
V_EUE=4.*[8900, 11300, 14900, 17700, 22500, 29700, 45000, 54600, 59400]

V_HSGSN=10^8;
h_AAL5=8;
W_frame_ATM=53;
V_HSGSN_RNC=155.52*(10^6);
h_ATM=5;
V_RNC=155.52*(10^6);
h_PDCP=1;
RLC_PDU_fixsize=40;
W_frame_RLC=42;
MACd_SDU=42;
h_MAC=3;
h_FP=7;
CRC=2;
h_AAL2=3;
V_nodeB=155.52*(10^6);
h_PDCP=1;
h_RLC=2;
V_HUE=3.6*(10^6);

W_semen=W_mob_tv+h_UDP;
W_datagram=W_semen+h_IP;
W_PDN_int=W_datagram+h_ether+FCS;
t_E_PDNint=(W_PDN_int*8)/V_PDN_int
t_D_GGSN=(W_mob_tv*8)/V_GGSN
W_frame_GGSN=W_mob_tv+h_GTP+h_UDP+h_IP+h_ether+FCS;
t_E_GGSN=(W_frame_GGSN*8)/V_GGSN

t_D_ESGSN=(W_mob_tv*8)/V_ESGSN
W_frame_ESGSN=W_mob_tv+h_SNDCP+FCS_1+h_LLC+h_BSSGP+h_NS;
t_E_SGSN=(W_frame_ESGSN*8)/V_ESGSN
t_D_PCU=(W_mob_tv*8)/V_PCU
W_frame_LLC=W_mob_tv+h_SNDCP+FCS_1+h_LLC;
N_frame_RLC=(W_frame_LLC*8)./MCS;
t_E_PCU=(jml_frame_LLC*N_frame_RLC*RLC_radioblk)/V_PCU2
```

```

t_P_PCU=t_D_PCU+t_E_PCU
t_D_EUE=(jml_frame_LLC.*N_frame_RLC.* (MCS./code_rate)) ./V_EUE

t_D_HSGSN=(W_mob_tv*8)/V_HSGSN;
W_frame_AAL5=W_mob_tv+h_GTP+h_IP+h_UDP+h_AAL5;
N_frame_ATM=W_frame_AAL5/48;
N_frameATM_tot=10;
W_frame_HSGSN=N_frameATM_tot*W_frame_ATM;
t_E_HSGSN=(W_frame_HSGSN*8)/V_HSGSN_RNC;
W_mobtv_RNC=W_frame_HSGSN-h_ATM-h_GTP-h_IP-h_UDP-h_AAL5;
t_D_RNC=(W_mobtv_RNC*8)/V_RNC;
W_frame_PDCP=W_mobtv_RNC+h_PDCP;
N_frame_HRLC=W_frame_PDCP/RLC_PDU_fixsize;
N_frameHRLC_tot=12;
W_frameRLC_tot=N_frameHRLC_tot*W_frame_RLC;
N_frame_MACd=W_frameRLC_tot/MACd_SDU;
W_frame_MACd=h_MAC+42;
W_frame_FP=W_frame_MACd+h_FP+CRC;
W_frameFP_tot=N_frame_MACd*W_frame_FP;
W_frame_AAL2=W_frameFP_tot+h_AAL2;
N_frame_ATM2=W_frame_AAL2/48;
N_frameATM2_tot=14;
W_frame_RNC=N_frameATM2_tot*W_frame_ATM;
t_E_RNC=(W_frame_RNC*8)/V_RNC;
W_mobtv_nodeB=W_frame_RNC-h_ATM-h_AAL2-h_FP-CRC;
t_D_nodeB=(W_mobtv_nodeB*8)/V_nodeB;
N_frame_MAChs=W_mobtv_nodeB/42;
N_frameMAChs_tot=18;
W_frame_MAChs=h_MAC+42;
W_frame_nodeB=N_frameMAChs_tot*W_frame_MAChs;
t_E_nodeB=(W_frame_nodeB*8)/V_nodeB;
W_mobtv_HUE=W_frame_nodeB-h_PDCP-h_RLC-h_MAC;
t_D_HUE=(W_mobtv_HUE*8)/V_HUE

```

%PERHITUNGAN DELAY TRANSMISI

```

W_frame_PDNint=435;
V_eth=10^9;
W_frame_GGSN=443;
V_Gn=10^8;
W_frame_ESGSN=432;
V_Gb=2.048*(10^6);
N_frame_RLCradio=[17.2727,13.5714,10.2703,8.6364,6.7857,5.1351,3.3929,
2.7941,2.5676];
RLC_radioblok=456;
V_Abis=150*16000;
V_MCS=[8900,11300,14900,17700,22500,29700,45000,54600,59400];
W_frame_HSGSN=530;
V_Iu=155.52*(10^6);
W_frame_RNC=742;
V_Iub=155.52*(10^6);
W_frame_nodeB=810;
V_HUE=3.6*(10^6);

t_T_Gi=(W_frame_PDNint*8)/V_eth;
t_T_Gn=(W_frame_GGSN*8)/V_Gn;
t_T_Gb=(W_frame_ESGSN*8)/V_Gb;
t_T_Abis=(RLC_radioblok.*N_frame_RLCradio)/V_Abis;
t_T_Um=(RLC_radioblok.*N_frame_RLCradio)./(4*V_MCS);
t_T_Iu=(W_frame_HSGSN*8)/V_Iu;
t_T_Iub=(W_frame_RNC*8)/V_Iub;

```



```
t_T_Uu=(W_frame_nodeB*8) / (3*V_HUE);
%PERHITUNGAN DELAY PROPAGASI

v_FO=2*(10^8);
v_UDR=3*(10^8);
N_frame_PDNint=1;
R_1=15000;
N_frame_GGSN=1;
R_2=1000;
N_frame_NS=1;
R_3=25;
N_frame_RLC=[17.2727,13.5714,10.2703,8.6364,6.7857,5.1351,3.3929,2.794
1,2.5676];
R_4=50;
N_frame_SNDCP=[17.2727,13.5714,10.2703,8.6364,6.7857,5.1351,3.3929,2.7
941,2.5676];
R_5=100;

R_6=2000;
N_frame_ATM=10;
R_7=3000;
N_frame_ATM2=14;
R_8=400;
N_frame_MAChs=18;
R_9=400;

t_Prop_PDN_GGSN=(N_frame_PDNint*R_1)/v_FO;
t_Prop_GGSN_ESGSN=(N_frame_GGSN*R_2)/v_FO;
t_Prop_ESGSN_BSC=(N_frame_NS*R_3)/v_FO;
t_Prop_BSC_BTS=(R_4.*N_frame_RLC)/v_FO;
t_Prop_BTS_UE=(R_5.*N_frame_SNDCP)/v_UDR;

t_Prop_GGSN_HSGSN=(N_frame_GGSN*R_6)/v_FO;
t_Prop_HSGSN_RNC=(N_frame_ATM*R_7)/v_FO;
t_Prop_RNC_nodeB=(N_frame_ATM2*R_8)/v_FO;
t_Prop_nodeB_UE=(N_frame_MAChs*R_9)/v_UDR;

%PERHITUNGAN DELAY ANTRIAN

p=[0.1:0.1:0.9]

C_GGSN=10^8;
L_GGSN=443*8
u_GGSN = C_GGSN/L_GGSN;
lambda_GGSN = u_GGSN*p;
x_1 = u_GGSN-lambda_GGSN;
y_1 = lambda_GGSN./ [u_GGSN * [x_1]];
z_1 = 1 / u_GGSN;
tw_1 = y_1 + z_1;

C_ESGSN=15*(10^6);
L_ESGSN=432*8
u_ESGSN = C_ESGSN/L_ESGSN;
lambda_ESGSN = u_ESGSN*p
x_2 = u_ESGSN-lambda_ESGSN;
y_2 = lambda_ESGSN./ [u_ESGSN * [x_2]];
z_2 = 1 / u_ESGSN;
tw_2 = y_2 + z_2

C_HSGSN=10^8;
```

```

L_HSGSN=530*8
u_HSGSN = C_HSGSN/L_HSGSN
lambda_HSGSN = u_HSGSN*p
x_3 = u_HSGSN-lambda_HSGSN;
y_3 = lambda_HSGSN./ [u_HSGSN * [x_3]];
z_3 = 1 / u_HSGSN;
tw_3 = y_3 + z_3;

%PERHITUNGAN DELAY TOTAL JARINGAN HSDPA DAN EDGE

t_E_EDGE_tot=t_E_PDNint+t_E_GGSN+t_E_SGSN+t_E_PCU
t_D_EDGE_tot=t_D_GGSN+t_D_ESGSN+t_D_PCU+t_D_EUE
t_proc_EDGE=t_E_EDGE_tot+t_D_EDGE_tot

t_E_HSDPA=t_E_PDNint+t_E_GGSN+t_E_HSGSN+t_E_nodeB;
t_D_HSDPA=t_D_GGSN+t_D_HSGSN+t_D_RNC+t_D_nodeB+t_D_HUE;
t_proc_HSDPA=t_E_HSDPA+t_D_HSDPA

t_T_EDGE=t_T_Gi+t_T_Gn+t_T_Gb+t_T_Abis+t_T_Um
t_T_HSDPA=t_T_Gi+t_T_Gn+t_T_Iu+t_T_Iub+t_T_Uu

t_Prop_EDGE=t_Prop_PDN_GGSN+t_Prop_GGSN_ESGSN+t_Prop_ESGSN_BSC+t_Prop_BSC_BTS+t_Prop_BTS_UE
t_Prop_HSDPA=t_Prop_PDN_GGSN+t_Prop_GGSN_HSGSN+t_Prop_HSGSN_RNC+t_Prop_RNC_nodeB+t_Prop_nodeB_UE

tw_EDGE=tw_1+tw_2
tw_HSDPA=tw_1+tw_3

t_total_EDGE1_1=t_proc_EDGE(1,1)+t_T_EDGE(1,1)+t_Prop_EDGE(1,1)+tw_EDGE(1,1)
t_total_EDGE1_2=t_proc_EDGE(1,2)+t_T_EDGE(1,2)+t_Prop_EDGE(1,2)+tw_EDGE(1,1)
t_total_EDGE1_3=t_proc_EDGE(1,3)+t_T_EDGE(1,3)+t_Prop_EDGE(1,3)+tw_EDGE(1,1)
t_total_EDGE1_4=t_proc_EDGE(1,4)+t_T_EDGE(1,4)+t_Prop_EDGE(1,4)+tw_EDGE(1,1)
t_total_EDGE1_5=t_proc_EDGE(1,5)+t_T_EDGE(1,5)+t_Prop_EDGE(1,5)+tw_EDGE(1,1)
t_total_EDGE1_6=t_proc_EDGE(1,6)+t_T_EDGE(1,6)+t_Prop_EDGE(1,6)+tw_EDGE(1,1)
t_total_EDGE1_7=t_proc_EDGE(1,7)+t_T_EDGE(1,7)+t_Prop_EDGE(1,7)+tw_EDGE(1,1)
t_total_EDGE1_8=t_proc_EDGE(1,8)+t_T_EDGE(1,8)+t_Prop_EDGE(1,8)+tw_EDGE(1,1)
t_total_EDGE1_9=t_proc_EDGE(1,9)+t_T_EDGE(1,9)+t_Prop_EDGE(1,9)+tw_EDGE(1,1)

t_total_EDGE2_1=t_proc_EDGE(1,1)+t_T_EDGE(1,1)+t_Prop_EDGE(1,1)+tw_EDGE(1,2)
t_total_EDGE2_2=t_proc_EDGE(1,2)+t_T_EDGE(1,2)+t_Prop_EDGE(1,2)+tw_EDGE(1,2)
t_total_EDGE2_3=t_proc_EDGE(1,3)+t_T_EDGE(1,3)+t_Prop_EDGE(1,3)+tw_EDGE(1,2)
t_total_EDGE2_4=t_proc_EDGE(1,4)+t_T_EDGE(1,4)+t_Prop_EDGE(1,4)+tw_EDGE(1,2)
t_total_EDGE2_5=t_proc_EDGE(1,5)+t_T_EDGE(1,5)+t_Prop_EDGE(1,5)+tw_EDGE(1,2)
t_total_EDGE2_6=t_proc_EDGE(1,6)+t_T_EDGE(1,6)+t_Prop_EDGE(1,6)+tw_EDGE(1,2)

```

```
t_total_EDGE2_7=t_proc_EDGE(1,7)+t_T_EDGE(1,7)+t_Prop_EDGE(1,7)+tw_EDGE(1,2)
t_total_EDGE2_8=t_proc_EDGE(1,8)+t_T_EDGE(1,8)+t_Prop_EDGE(1,8)+tw_EDGE(1,2)
t_total_EDGE2_9=t_proc_EDGE(1,9)+t_T_EDGE(1,9)+t_Prop_EDGE(1,9)+tw_EDGE(1,2)

t_total_EDGE3_1=t_proc_EDGE(1,1)+t_T_EDGE(1,1)+t_Prop_EDGE(1,1)+tw_EDGE(1,3)
t_total_EDGE3_2=t_proc_EDGE(1,2)+t_T_EDGE(1,2)+t_Prop_EDGE(1,2)+tw_EDGE(1,3)
t_total_EDGE3_3=t_proc_EDGE(1,3)+t_T_EDGE(1,3)+t_Prop_EDGE(1,3)+tw_EDGE(1,3)
t_total_EDGE3_4=t_proc_EDGE(1,4)+t_T_EDGE(1,4)+t_Prop_EDGE(1,4)+tw_EDGE(1,3)
t_total_EDGE3_5=t_proc_EDGE(1,5)+t_T_EDGE(1,5)+t_Prop_EDGE(1,5)+tw_EDGE(1,3)
t_total_EDGE3_6=t_proc_EDGE(1,6)+t_T_EDGE(1,6)+t_Prop_EDGE(1,6)+tw_EDGE(1,3)
t_total_EDGE3_7=t_proc_EDGE(1,7)+t_T_EDGE(1,7)+t_Prop_EDGE(1,7)+tw_EDGE(1,3)
t_total_EDGE3_8=t_proc_EDGE(1,8)+t_T_EDGE(1,8)+t_Prop_EDGE(1,8)+tw_EDGE(1,3)
t_total_EDGE3_9=t_proc_EDGE(1,9)+t_T_EDGE(1,9)+t_Prop_EDGE(1,9)+tw_EDGE(1,3)

t_total_EDGE4_1=t_proc_EDGE(1,1)+t_T_EDGE(1,1)+t_Prop_EDGE(1,1)+tw_EDGE(1,4)
t_total_EDGE4_2=t_proc_EDGE(1,2)+t_T_EDGE(1,2)+t_Prop_EDGE(1,2)+tw_EDGE(1,4)
t_total_EDGE4_3=t_proc_EDGE(1,3)+t_T_EDGE(1,3)+t_Prop_EDGE(1,3)+tw_EDGE(1,4)
t_total_EDGE4_4=t_proc_EDGE(1,4)+t_T_EDGE(1,4)+t_Prop_EDGE(1,4)+tw_EDGE(1,4)
t_total_EDGE4_5=t_proc_EDGE(1,5)+t_T_EDGE(1,5)+t_Prop_EDGE(1,5)+tw_EDGE(1,4)
t_total_EDGE4_6=t_proc_EDGE(1,6)+t_T_EDGE(1,6)+t_Prop_EDGE(1,6)+tw_EDGE(1,4)
t_total_EDGE4_7=t_proc_EDGE(1,7)+t_T_EDGE(1,7)+t_Prop_EDGE(1,7)+tw_EDGE(1,4)
t_total_EDGE4_8=t_proc_EDGE(1,8)+t_T_EDGE(1,8)+t_Prop_EDGE(1,8)+tw_EDGE(1,4)
t_total_EDGE4_9=t_proc_EDGE(1,9)+t_T_EDGE(1,9)+t_Prop_EDGE(1,9)+tw_EDGE(1,4)

t_total_EDGE5_1=t_proc_EDGE(1,1)+t_T_EDGE(1,1)+t_Prop_EDGE(1,1)+tw_EDGE(1,5)
t_total_EDGE5_2=t_proc_EDGE(1,2)+t_T_EDGE(1,2)+t_Prop_EDGE(1,2)+tw_EDGE(1,5)
t_total_EDGE5_3=t_proc_EDGE(1,3)+t_T_EDGE(1,3)+t_Prop_EDGE(1,3)+tw_EDGE(1,5)
t_total_EDGE5_4=t_proc_EDGE(1,4)+t_T_EDGE(1,4)+t_Prop_EDGE(1,4)+tw_EDGE(1,5)
t_total_EDGE5_5=t_proc_EDGE(1,5)+t_T_EDGE(1,5)+t_Prop_EDGE(1,5)+tw_EDGE(1,5)
t_total_EDGE5_6=t_proc_EDGE(1,6)+t_T_EDGE(1,6)+t_Prop_EDGE(1,6)+tw_EDGE(1,5)
t_total_EDGE5_7=t_proc_EDGE(1,7)+t_T_EDGE(1,7)+t_Prop_EDGE(1,7)+tw_EDGE(1,5)
t_total_EDGE5_8=t_proc_EDGE(1,8)+t_T_EDGE(1,8)+t_Prop_EDGE(1,8)+tw_EDGE(1,5)
```

t_total_EDGE5_9=t_proc_EDGE(1,9)+t_T_EDGE(1,9)+t_Prop_EDGE(1,9)+tw_EDGE(1,5)

t_total_EDGE6_1=t_proc_EDGE(1,1)+t_T_EDGE(1,1)+t_Prop_EDGE(1,1)+tw_EDGE(1,6)

t_total_EDGE6_2=t_proc_EDGE(1,2)+t_T_EDGE(1,2)+t_Prop_EDGE(1,2)+tw_EDGE(1,6)

t_total_EDGE6_3=t_proc_EDGE(1,3)+t_T_EDGE(1,3)+t_Prop_EDGE(1,3)+tw_EDGE(1,6)

t_total_EDGE6_4=t_proc_EDGE(1,4)+t_T_EDGE(1,4)+t_Prop_EDGE(1,4)+tw_EDGE(1,6)

t_total_EDGE6_5=t_proc_EDGE(1,5)+t_T_EDGE(1,5)+t_Prop_EDGE(1,5)+tw_EDGE(1,6)

t_total_EDGE6_6=t_proc_EDGE(1,6)+t_T_EDGE(1,6)+t_Prop_EDGE(1,6)+tw_EDGE(1,6)

t_total_EDGE6_7=t_proc_EDGE(1,7)+t_T_EDGE(1,7)+t_Prop_EDGE(1,7)+tw_EDGE(1,6)

t_total_EDGE6_8=t_proc_EDGE(1,8)+t_T_EDGE(1,8)+t_Prop_EDGE(1,8)+tw_EDGE(1,6)

t_total_EDGE6_9=t_proc_EDGE(1,9)+t_T_EDGE(1,9)+t_Prop_EDGE(1,9)+tw_EDGE(1,6)

t_total_EDGE7_1=t_proc_EDGE(1,1)+t_T_EDGE(1,1)+t_Prop_EDGE(1,1)+tw_EDGE(1,7)

t_total_EDGE7_2=t_proc_EDGE(1,2)+t_T_EDGE(1,2)+t_Prop_EDGE(1,2)+tw_EDGE(1,7)

t_total_EDGE7_3=t_proc_EDGE(1,3)+t_T_EDGE(1,3)+t_Prop_EDGE(1,3)+tw_EDGE(1,7)

t_total_EDGE7_4=t_proc_EDGE(1,4)+t_T_EDGE(1,4)+t_Prop_EDGE(1,4)+tw_EDGE(1,7)

t_total_EDGE7_5=t_proc_EDGE(1,5)+t_T_EDGE(1,5)+t_Prop_EDGE(1,5)+tw_EDGE(1,7)

t_total_EDGE7_6=t_proc_EDGE(1,6)+t_T_EDGE(1,6)+t_Prop_EDGE(1,6)+tw_EDGE(1,7)

t_total_EDGE7_7=t_proc_EDGE(1,7)+t_T_EDGE(1,7)+t_Prop_EDGE(1,7)+tw_EDGE(1,7)

t_total_EDGE7_8=t_proc_EDGE(1,8)+t_T_EDGE(1,8)+t_Prop_EDGE(1,8)+tw_EDGE(1,7)

t_total_EDGE7_9=t_proc_EDGE(1,9)+t_T_EDGE(1,9)+t_Prop_EDGE(1,9)+tw_EDGE(1,7)

t_total_EDGE8_1=t_proc_EDGE(1,1)+t_T_EDGE(1,1)+t_Prop_EDGE(1,1)+tw_EDGE(1,8)

t_total_EDGE8_2=t_proc_EDGE(1,2)+t_T_EDGE(1,2)+t_Prop_EDGE(1,2)+tw_EDGE(1,8)

t_total_EDGE8_3=t_proc_EDGE(1,3)+t_T_EDGE(1,3)+t_Prop_EDGE(1,3)+tw_EDGE(1,8)

t_total_EDGE8_4=t_proc_EDGE(1,4)+t_T_EDGE(1,4)+t_Prop_EDGE(1,4)+tw_EDGE(1,8)

t_total_EDGE8_5=t_proc_EDGE(1,5)+t_T_EDGE(1,5)+t_Prop_EDGE(1,5)+tw_EDGE(1,8)

t_total_EDGE8_6=t_proc_EDGE(1,6)+t_T_EDGE(1,6)+t_Prop_EDGE(1,6)+tw_EDGE(1,8)

t_total_EDGE8_7=t_proc_EDGE(1,7)+t_T_EDGE(1,7)+t_Prop_EDGE(1,7)+tw_EDGE(1,8)

t_total_EDGE8_8=t_proc_EDGE(1,8)+t_T_EDGE(1,8)+t_Prop_EDGE(1,8)+tw_EDGE(1,8)

t_total_EDGE8_9=t_proc_EDGE(1,9)+t_T_EDGE(1,9)+t_Prop_EDGE(1,9)+tw_EDGE(1,8)



```

t_total_EDGE9_1=t_proc_EDGE(1,1)+t_T_EDGE(1,1)+t_Prop_EDGE(1,1)+tw_EDG
E(1,9)
t_total_EDGE9_2=t_proc_EDGE(1,2)+t_T_EDGE(1,2)+t_Prop_EDGE(1,2)+tw_EDG
E(1,9)
t_total_EDGE9_3=t_proc_EDGE(1,3)+t_T_EDGE(1,3)+t_Prop_EDGE(1,3)+tw_EDG
E(1,9)
t_total_EDGE9_4=t_proc_EDGE(1,4)+t_T_EDGE(1,4)+t_Prop_EDGE(1,4)+tw_EDG
E(1,9)
t_total_EDGE9_5=t_proc_EDGE(1,5)+t_T_EDGE(1,5)+t_Prop_EDGE(1,5)+tw_EDG
E(1,9)
t_total_EDGE9_6=t_proc_EDGE(1,6)+t_T_EDGE(1,6)+t_Prop_EDGE(1,6)+tw_EDG
E(1,9)
t_total_EDGE9_7=t_proc_EDGE(1,7)+t_T_EDGE(1,7)+t_Prop_EDGE(1,7)+tw_EDG
E(1,9)
t_total_EDGE9_8=t_proc_EDGE(1,8)+t_T_EDGE(1,8)+t_Prop_EDGE(1,8)+tw_EDG
E(1,9)
t_total_EDGE9_9=t_proc_EDGE(1,9)+t_T_EDGE(1,9)+t_Prop_EDGE(1,9)+tw_EDG
E(1,9)

t_EDGE1=[t_total_EDGE1_1 t_total_EDGE2_1 t_total_EDGE3_1
t_total_EDGE4_1 t_total_EDGE5_1 t_total_EDGE6_1 t_total_EDGE7_1
t_total_EDGE8_1 t_total_EDGE9_1]
t_EDGE2=[t_total_EDGE1_2 t_total_EDGE2_2 t_total_EDGE3_2
t_total_EDGE4_2 t_total_EDGE5_2 t_total_EDGE6_2 t_total_EDGE7_2
t_total_EDGE8_2 t_total_EDGE9_2]
t_EDGE3=[t_total_EDGE1_3 t_total_EDGE2_3 t_total_EDGE3_3
t_total_EDGE4_3 t_total_EDGE5_3 t_total_EDGE6_3 t_total_EDGE7_3
t_total_EDGE8_3 t_total_EDGE9_3]
t_EDGE4=[t_total_EDGE1_4 t_total_EDGE2_4 t_total_EDGE3_4
t_total_EDGE4_4 t_total_EDGE5_4 t_total_EDGE6_4 t_total_EDGE7_4
t_total_EDGE8_4 t_total_EDGE9_4]
t_EDGE5=[t_total_EDGE1_5 t_total_EDGE2_5 t_total_EDGE3_5
t_total_EDGE4_5 t_total_EDGE5_5 t_total_EDGE6_5 t_total_EDGE7_5
t_total_EDGE8_5 t_total_EDGE9_5]
t_EDGE6=[t_total_EDGE1_6 t_total_EDGE2_6 t_total_EDGE3_6
t_total_EDGE4_6 t_total_EDGE5_6 t_total_EDGE6_6 t_total_EDGE7_6
t_total_EDGE8_6 t_total_EDGE9_6]
t_EDGE7=[t_total_EDGE1_7 t_total_EDGE2_7 t_total_EDGE3_7
t_total_EDGE4_7 t_total_EDGE5_7 t_total_EDGE6_7 t_total_EDGE7_7
t_total_EDGE8_7 t_total_EDGE9_7]
t_EDGE8=[t_total_EDGE1_8 t_total_EDGE2_8 t_total_EDGE3_8
t_total_EDGE4_8 t_total_EDGE5_8 t_total_EDGE6_8 t_total_EDGE7_8
t_total_EDGE8_8 t_total_EDGE9_8]
t_EDGE9=[t_total_EDGE1_9 t_total_EDGE2_9 t_total_EDGE3_9
t_total_EDGE4_9 t_total_EDGE5_9 t_total_EDGE6_9 t_total_EDGE7_9
t_total_EDGE8_9 t_total_EDGE9_9]

t_total_HSDPA=t_proc_HSDPA + t_T_HSDPA + t_Prop_HSDPA + tw_HSDPA

plot(p,t_EDGE1, '-^r', p,t_EDGE2, '-^b', p,t_EDGE3, '-^g', p,t_EDGE4, '-^y',
     p,t_EDGE5, '-^k', p,t_EDGE6, '-^c', p,t_EDGE7, '-^m', p,t_EDGE8, '-^r',
     p,t_EDGE9, '-*k')
grid on
xlabel ('Faktor Utilitas')
ylabel ('Delay End-to-End (s)')
h=legend

plot(p,t_total_HSDPA, '-^b')
grid on

```

```
xlabel ('Faktor Utilitas')
ylabel ('Delay End-to-End (s)')
```

LISTING PERHITUNGAN THROUGHPUT

```
%PERHITUNGAN THROUGHPUT
```

```
%JARINGAN EDGE
```

```
Pe=10^-9
```

```
Pe_BER=10^-4
```

```
p_mobtv=W_mob_tv*8*(Pe)
```

```
p_PDN=W_frame_PDNint*8*(Pe)
```

```
p_GGSN=W_frame_GGSN*8*(Pe)
```

```
p_ESGSN=W_frame_ESGSN*8*(Pe)
```

```
p_GERAN=380*8*(Pe)
```

```
p_AI= 1 - (1- Pe_BER)^(456)
```

```
p_net_EDGE = 1 - ((1 - p_AI)*((1 - p_GERAN)^2)*(1 - p_GGSN)*(1 - p_ESGSN)*(1 - p_PDN))
```

```
P_net_E_tot=1-(1-p_net_EDGE)*(1-p_mobtv))
```

```
TI=456./V_MCS
```

```
TP=t_Prop_EDGE./N_frame_RLC
```

```
T_proses1=(t_proc_EDGE./N_frame_RLC)+(tw_EDGE(1,1)./N_frame_RLC)
```

```
T_proses2=(t_proc_EDGE./N_frame_RLC)+(tw_EDGE(1,2)./N_frame_RLC)
```

```
T_proses3=(t_proc_EDGE./N_frame_RLC)+(tw_EDGE(1,3)./N_frame_RLC)
```

```
T_proses4=(t_proc_EDGE./N_frame_RLC)+(tw_EDGE(1,4)./N_frame_RLC)
```

```
T_proses5=(t_proc_EDGE./N_frame_RLC)+(tw_EDGE(1,5)./N_frame_RLC)
```

```
T_proses6=(t_proc_EDGE./N_frame_RLC)+(tw_EDGE(1,6)./N_frame_RLC)
```

```
T_proses7=(t_proc_EDGE./N_frame_RLC)+(tw_EDGE(1,7)./N_frame_RLC)
```

```
T_proses8=(t_proc_EDGE./N_frame_RLC)+(tw_EDGE(1,8)./N_frame_RLC)
```

```
T_proses9=(t_proc_EDGE./N_frame_RLC)+(tw_EDGE(1,9)./N_frame_RLC)
```

```
T_out1=(2*TP)+(2*TI)+(2*T_proses1)
```

```
T_out2=(2*TP)+(2*TI)+(2*T_proses2)
```

```
T_out3=(2*TP)+(2*TI)+(2*T_proses3)
```

```
T_out4=(2*TP)+(2*TI)+(2*T_proses4)
```

```
T_out5=(2*TP)+(2*TI)+(2*T_proses5)
```

```
T_out6=(2*TP)+(2*TI)+(2*T_proses6)
```

```
T_out7=(2*TP)+(2*TI)+(2*T_proses7)
```

```
T_out8=(2*TP)+(2*TI)+(2*T_proses8)
```

```
T_out9=(2*TP)+(2*TI)+(2*T_proses9)
```

```
a_x=1./TI
```

```
a_y_1=a_x.*T_out1
```

```
a_y_2=a_x.*T_out2
```

```
a_y_3=a_x.*T_out3
```

```
a_y_4=a_x.*T_out4
```

```
a_y_5=a_x.*T_out5
```

```
a_y_6=a_x.*T_out6
```

```
a_y_7=a_x.*T_out7
```

```
a_y_8=a_x.*T_out8
```

```
a_y_9=a_x.*T_out9
```

```
a1=1+a_y_1
```

```
a2=1+a_y_2
```

```
a3=1+a_y_3
```

```
a4=1+a_y_4
```



a5=1+a_y_5
a6=1+a_y_6
a7=1+a_y_7
a8=1+a_y_8
a9=1+a_y_9

T_1= 1 - P_net_E_tot
T_2_1=((1 + (a1-1).*P_net_E_tot).*TI)
T_2_2=((1 + (a2-1).*P_net_E_tot).*TI)
T_2_3=((1 + (a3-1).*P_net_E_tot).*TI)
T_2_4=((1 + (a4-1).*P_net_E_tot).*TI)
T_2_5=((1 + (a5-1).*P_net_E_tot).*TI)
T_2_6=((1 + (a6-1).*P_net_E_tot).*TI)
T_2_7=((1 + (a7-1).*P_net_E_tot).*TI)
T_2_8=((1 + (a8-1).*P_net_E_tot).*TI)
T_2_9=((1 + (a9-1).*P_net_E_tot).*TI)

T1=T_1./T_2_1
T2=T_1./T_2_2
T3=T_1./T_2_3
T4=T_1./T_2_4
T5=T_1./T_2_5
T6=T_1./T_2_6
T7=T_1./T_2_7
T8=T_1./T_2_8
T9=T_1./T_2_9

T_bps1=T1*456
T_bps2=T2*456
T_bps3=T3*456
T_bps4=T4*456
T_bps5=T5*456
T_bps6=T6*456
T_bps7=T7*456
T_bps8=T8*456
T_bps9=T9*456

T1_bps=[T_bps1(1,1) T_bps2(1,1) T_bps3(1,1) T_bps4(1,1) T_bps5(1,1)
T_bps6(1,1) T_bps7(1,1) T_bps8(1,1) T_bps9(1,1)]
T2_bps=[T_bps1(1,2) T_bps2(1,2) T_bps3(1,2) T_bps4(1,2) T_bps5(1,2)
T_bps6(1,2) T_bps7(1,2) T_bps8(1,2) T_bps9(1,2)]
T3_bps=[T_bps1(1,3) T_bps2(1,3) T_bps3(1,3) T_bps4(1,3) T_bps5(1,3)
T_bps6(1,3) T_bps7(1,3) T_bps8(1,3) T_bps9(1,3)]
T4_bps=[T_bps1(1,4) T_bps2(1,4) T_bps3(1,4) T_bps4(1,4) T_bps5(1,4)
T_bps6(1,4) T_bps7(1,4) T_bps8(1,4) T_bps9(1,4)]
T5_bps=[T_bps1(1,5) T_bps2(1,5) T_bps3(1,5) T_bps4(1,5) T_bps5(1,5)
T_bps6(1,5) T_bps7(1,5) T_bps8(1,5) T_bps9(1,5)]
T6_bps=[T_bps1(1,6) T_bps2(1,6) T_bps3(1,6) T_bps4(1,6) T_bps5(1,6)
T_bps6(1,6) T_bps7(1,6) T_bps8(1,6) T_bps9(1,6)]
T7_bps=[T_bps1(1,7) T_bps2(1,7) T_bps3(1,7) T_bps4(1,7) T_bps5(1,7)
T_bps6(1,7) T_bps7(1,7) T_bps8(1,7) T_bps9(1,7)]
T8_bps=[T_bps1(1,8) T_bps2(1,8) T_bps3(1,8) T_bps4(1,8) T_bps5(1,8)
T_bps6(1,8) T_bps7(1,8) T_bps8(1,8) T_bps9(1,8)]
T9_bps=[T_bps1(1,9) T_bps2(1,9) T_bps3(1,9) T_bps4(1,9) T_bps5(1,9)
T_bps6(1,9) T_bps7(1,9) T_bps8(1,9) T_bps9(1,9)]

plot(p,T1_bps,'-*r',p,T2_bps,'-*b', p,T3_bps,'-*g', p,T4_bps,'-*b',
p,T5_bps,'-*y', p,T6_bps,'-*c', p,T7_bps,'-*m', p,T8_bps,'-*k',
p,T9_bps,'-*k')
grid on
xlabel ('Faktor Utilitas')

```

ylabel ('Throughput (bps)')

%JARINGAN HSDPA
Pe2=10^-8
p_HSGSN=W_frame_HSGSN*8*(Pe2)
p_RNC=W_frame_RNC*8*(Pe2)
p_nodeB=W_frame_nodeB*8*(Pe2)

EIRP_dBw = 57;
Lp_dB = 126.36;
G_MS_dB = 0;
Im_dB = 9;
NF_MS_dB = 8;
Rb = 3600000;
Ptr = EIRP_dBw - Lp_dB + G_MS_dB - Im_dB
No = 10 * log10 (290* 1.38 *10 ^-23) + NF_MS_dB + 40
EbNo_Tr= Ptr - 10 * log10(Rb) - 10 * log10(10^(0.1 * No))
Pe_H_AI = exp(-EbNo_Tr) / (2 * sqrt(pi *EbNo_Tr))

p_H_AI = 1 - (1- Pe_H_AI)^(W_frame_nodeB*8)

p_net_HSDPA = 1 - ((1 - p_H_AI)*(1 - p_nodeB)*(1 - p_RNC)*(1 - p_GGSN)*(1 - p_HSGSN)*(1 - p_PDN))
P_net_H_tot=1-(1-p_net_HSDPA)*(1-p_mobtv))

TI_H=(W_frame_nodeB*8)/V_HUE
TP_H=t_Prop_HSDPA/N_frameMAChs_tot
T_H_proses=(t_proc_HSDPA/N_frameMAChs_tot)+(tw_HSDPA/N_frameMAChs_tot)

t_H_out=(2*TI_H)+(2*TP_H)+(2*T_H_proses)
a_H=1+(t_H_out/TI_H)

T_H_1= 1 - P_net_H_tot
T_H_2=((1 + (a_H-1).*P_net_H_tot).*TI_H)
T_H=T_H_1./T_H_2
T_H_bps=T_H*W_frame_nodeB*8

plot(p,T_H_bps,'-*b')
grid on
xlabel ('Faktor Utilitas')
ylabel ('Throughput (bps)')

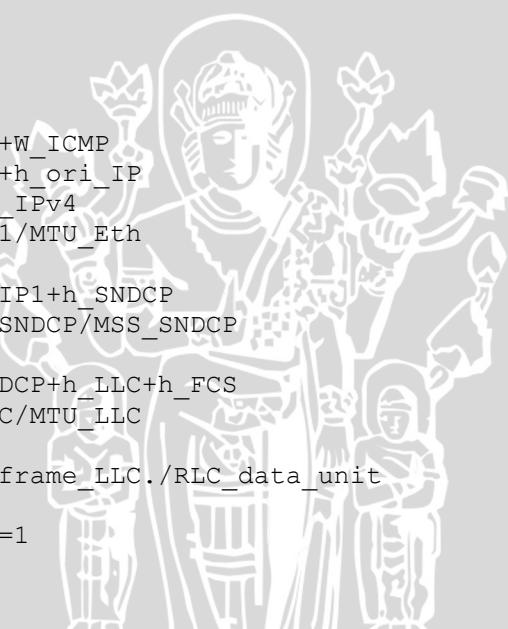
```

LISTING PERHITUNGAN SAAT HANDOVER

```

W_mob_tv=369;
W_ICMP=8;
h_ori_IP=8;
h_IPv4=20;
MTU_Eth=1500;
h_SNDCP=4;
MSS_SNDCP=1503;
h_LLC=4;
h_FCS=3;
MTU_LLC=1500;
RLC_data_unit=[22,28,37,44,56,74,112,136,148];
RLC_data_unit_bit=8*[22,28,37,44,56,74,112,136,148]
RLC_MAC_radioblk=456
V_EUE2=[8900,11300,14900,17700,22500,29700,45000,54600,59400]

```



```
v_UDR=3*(10^8)
R_UE_BTS=100
V_Abis=150*16000
R_BTS_BSC=50
v_FO=2*(10^8)
code_rate=[1/2, 0.64, 0.83, 1, 1/3, 1/2, 3/4, 0.90, 1]
V_PCU2=14*150*16000
h_BSSGP=48;
h_NS=4;
FCS=3;
V_PCU1=14*31*64000;
V_Gb=2.048*(10^6);
R_ESGSN_BSC=25;
V_ESGSN=15*(10^6);
h_GTP=8;
h_Ethernet=14;
FCS2=4;
V_Gn=10^8;
R_ESGSN_GGSN=1000;
h_UDP=8;
h_IP=40;
V_PDNint=10^9;
V_Ethernet=10^9;
R_PDNint_GGSN=15000;
V_GGSN=10^8;

W_datagram_HO=W_mob_tv+W_ICMP
W_ori_IP=W_datagram_HO+h_ori_IP
W_frame_IP1=W_ori_IP+h_IPv4
n_frame_IP1=W_frame_IP1/MTU_Eth
n_frame_IP1_fix=1
W_frame_SNDCP=W_frame_IP1+h_SNDCP
n_frame_SNDCP=W_frame_SNDCP/MSS_SNDCP
n_frame_SNDCP_fix=1
W_frame_LLC=W_frame_SNDCP+h_LLC+h_FCS
n_frame_LLC=W_frame_LLC/MTU_LLC
n_frame_LLC_fix=1
n_frame_RLCradioblk=W_frame_LLC./RLC_data_unit
n_frame_NS_fix=1
n_frame_IPbackbone_fix=1
n_frame_Ethernet_fix=1
n_frame_PDNint_fix=1

t_E_EUE=(n_frame_IP1_fix*n_frame_SNDCP_fix.*n_frame_RLCradioblk*RLC_MAC_radioblk)./(2*V_EUE2)
t_T_Um=(RLC_MAC_radioblk.*n_frame_RLCradioblk)./(2*V_EUE2)
t_Prop UE BTS=(R_UE_BTS.*n_frame_RLCradioblk)/v_UDR
t_T_Abis=(RLC_MAC_radioblk.*n_frame_RLCradioblk)/V_Abis
t_Prop BTS_BSC=(R_BTS_BSC.*n_frame_RLCradioblk)/v_FO

t_net_EUE_BTS=t_E_EUE+t_T_Um+t_Prop UE BTS+t_T_Abis+t_Prop BTS_BSC

t_D_BSS=(n_frame_LLC_fix.*n_frame_RLCradioblk.* (RLC_data_unit_bit./code_rate))/V_PCU2

W_frame_BSSGP=W_datagram_HO+h_BSSGP
W_frame_NS=W_frame_BSSGP+h_NS+FCS

t_E_BSS=(n_frame_LLC_fix*n_frame_NS_fix*W_frame_NS*8)/V_PCU1
t_T_Gb=(W_frame_NS*8)/V_Gb
```

```

t_Prop_BSC_ESGSN=(n_frame_NS_fix*R_ESGSN_BSC)/v_FO

t_net_BSS_ESGSN=t_D_BSS+t_E_BSS+t_T_Gb+t_Prop_BSC_ESGSN

t_D_ESGSN=(n_frame_LLC_fix*W_frame_NS*n_frame_NS_fix*8)/V_ESGSN

W_frame_GTP=W_frame_IP1+h_GTP
W_frame_Eth=W_frame_GTP+h_Ethernet+FCS2

t_E_ESGSN=(n_frame_IPbackbone_fix*n_frame_Ethernet_fix*W_frame_Eth*8)/
V_ESGSN
t_T_Gn_up=(W_frame_Eth*8)/V_Gn
t_Prop_ESGSN_GGSN=(n_frame_Ethernet_fix*R_ESGSN_GGSN)/v_FO

p=[0.1:0.1:0.9]

C_ESGSN=15*(10^6)
L_ESGSN=431*8
u_ESGSN = C_ESGSN/L_ESGSN
lambda_ESGSN = u_ESGSN*p
x_1 = u_ESGSN-lambda_ESGSN
y_1 = lambda_ESGSN./[u_ESGSN * [x_1]]
z_1 = 1 / u_ESGSN
tw_ESGSN = y_1 + z_1

t_net_ESGSN_GGSN=t_D_ESGSN+t_E_ESGSN+t_T_Gn_up+t_Prop_ESGSN_GGSN+tw_ES
GSN

W_frame_UDP=W_datagram_HO+h_UDP
W_frame_IP2=W_frame_UDP+h_IP
W_frame_PDNint=W_frame_IP2+h_Ethernet+FCS2

t_E_PDNint=(W_frame_PDNint*8)/V_PDNint
t_T_PDNint=(W_frame_PDNint*8)/V_Ethernet
t_Prop_PDNint_GGSN=(n_frame_PDNint_fix*R_PDNint_GGSN)/v_FO

t_net_PDNint_GGSN=t_E_PDNint+t_T_PDNint+t_Prop_PDNint_GGSN

t_D_GGSN=(W_datagram_HO*8)/V_GGSN

w_frame_IP3=W_datagram_HO+h_ori_IP+h_GTP+h_UDP+h_IPv4
W_frame_GGSN=w_frame_IP3+h_Ethernet+FCS2

t_E_GGSN=(W_frame_GGSN*8)/V_GGSN
t_T_Gn_down=(W_frame_GGSN*8)/V_Gn
t_Prop_GGSN_ESGSN=(n_frame_Ethernet_fix*R_ESGSN_GGSN)/v_FO

C_GGSN=10^8;
L_GGSN=W_frame_GGSN*8
u_GGSN = C_GGSN/L_GGSN
lambda_GGSN = u_GGSN*p
x_2 = u_GGSN-lambda_GGSN;
y_2 = lambda_GGSN./[u_GGSN * [x_2]];
z_2 = 1 / u_GGSN;
tw_GGSN = y_2 + z_2

t_net_GGSN_ESGSN=t_D_GGSN+t_E_GGSN+t_T_Gn_down+t_Prop_GGSN_ESGSN+tw_GG
SN

```

```

t_D_ESGSN_down=(n_frame_Ethernet_fix*W_frame_GGSN*8)/V_ESGSN

W_frame SNDCP_down=W_frame_GGSN+h SNDCP
W_frame LLC_down=W_frame SNDCP_down+h LLC+h FCS
W_frame_BSSGP=W_frame_LLC_down+h_BSSGP
W_frame_NS_down=W_frame_BSSGP+h_NS
n_frame_RLCradioblk_down=W_frame_LLC_down./RLC_data_unit

t_E_ESGSN_down=(n_frame_SNDCP_fix*n_frame_NS_fix*W_frame_NS_down*8)/V_ESGSN
t_T_Gb_down=(W_frame_NS_down*8)/V_Gb
t_Prop_ESGSN_BSC=(n_frame_NS_fix*R_ESGSN_BSC)/v_FO
t_T_Abis_down=(RLC_MAC_radioblk.*n_frame_RLCradioblk_down)/V_Abis
t_Prop_BSC_BTS=(R_BTS_BSC.*n_frame_RLCradioblk_down)/v_FO

C_ESGSN_down=15*(10^6)
L_ESGSN_down=W_frame_NS_down*8
u_ESGSN_down = C_ESGSN_down/L_ESGSN_down
lambda_ESGSN_down = u_ESGSN_down*p
x_3 = u_ESGSN_down-lambda_ESGSN_down
y_3 = lambda_ESGSN_down./ [u_ESGSN_down * [x_3]]
z_3 = 1 / u_ESGSN_down
tw_ESGSN_down = y_3 + z_3

t_net_ESGSN_BSS_1_1=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,1)+t_Prop_BSC_BTS(1,1)+tw_ESGSN_down(1,1)
t_net_ESGSN_BSS_1_2=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,1)+t_Prop_BSC_BTS(1,1)+tw_ESGSN_down(1,2)
t_net_ESGSN_BSS_1_3=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,1)+t_Prop_BSC_BTS(1,1)+tw_ESGSN_down(1,3)
t_net_ESGSN_BSS_1_4=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,1)+t_Prop_BSC_BTS(1,1)+tw_ESGSN_down(1,4)
t_net_ESGSN_BSS_1_5=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,1)+t_Prop_BSC_BTS(1,1)+tw_ESGSN_down(1,5)
t_net_ESGSN_BSS_1_6=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,1)+t_Prop_BSC_BTS(1,1)+tw_ESGSN_down(1,6)
t_net_ESGSN_BSS_1_7=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,1)+t_Prop_BSC_BTS(1,1)+tw_ESGSN_down(1,7)
t_net_ESGSN_BSS_1_8=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,1)+t_Prop_BSC_BTS(1,1)+tw_ESGSN_down(1,8)
t_net_ESGSN_BSS_1_9=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,1)+t_Prop_BSC_BTS(1,1)+tw_ESGSN_down(1,9)

t_net_ESGSN_BSS_2_1=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,2)+t_Prop_BSC_BTS(1,2)+tw_ESGSN_down(1,1)
t_net_ESGSN_BSS_2_2=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,2)+t_Prop_BSC_BTS(1,2)+tw_ESGSN_down(1,2)
t_net_ESGSN_BSS_2_3=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,2)+t_Prop_BSC_BTS(1,2)+tw_ESGSN_down(1,3)
t_net_ESGSN_BSS_2_4=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,2)+t_Prop_BSC_BTS(1,2)+tw_ESGSN_down(1,4)
t_net_ESGSN_BSS_2_5=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,2)+t_Prop_BSC_BTS(1,2)+tw_ESGSN_down(1,5)
t_net_ESGSN_BSS_2_6=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,2)+t_Prop_BSC_BTS(1,2)+tw_ESGSN_down(1,6)
t_net_ESGSN_BSS_2_7=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,2)+t_Prop_BSC_BTS(1,2)+tw_ESGSN_down(1,7)
t_net_ESGSN_BSS_2_8=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_ESGSN_BSC+t_T_Abis_down(1,2)+t_Prop_BSC_BTS(1,2)+tw_ESGSN_down(1,8)

```



```
t_net_ESGSN_BSS_9_3=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_E
SGSN_BSC+t_T_Abis_down(1,9)+t_Prop_BSC_BTS(1,9)+tw_ESGSN_down(1,3)
t_net_ESGSN_BSS_9_4=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_E
SGSN_BSC+t_T_Abis_down(1,9)+t_Prop_BSC_BTS(1,9)+tw_ESGSN_down(1,4)
t_net_ESGSN_BSS_9_5=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_E
SGSN_BSC+t_T_Abis_down(1,9)+t_Prop_BSC_BTS(1,9)+tw_ESGSN_down(1,5)
t_net_ESGSN_BSS_9_6=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_E
SGSN_BSC+t_T_Abis_down(1,9)+t_Prop_BSC_BTS(1,9)+tw_ESGSN_down(1,6)
t_net_ESGSN_BSS_9_7=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_E
SGSN_BSC+t_T_Abis_down(1,9)+t_Prop_BSC_BTS(1,9)+tw_ESGSN_down(1,7)
t_net_ESGSN_BSS_9_8=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_E
SGSN_BSC+t_T_Abis_down(1,9)+t_Prop_BSC_BTS(1,9)+tw_ESGSN_down(1,8)
t_net_ESGSN_BSS_9_9=t_D_ESGSN_down+t_E_ESGSN_down+t_T_Gb_down+t_Prop_E
SGSN_BSC+t_T_Abis_down(1,9)+t_Prop_BSC_BTS(1,9)+tw_ESGSN_down(1,9)

t_net_ESGSN_BSS_1=[t_net_ESGSN_BSS_1_1 t_net_ESGSN_BSS_1_2
t_net_ESGSN_BSS_1_3 t_net_ESGSN_BSS_1_4 t_net_ESGSN_BSS_1_5
t_net_ESGSN_BSS_1_6 t_net_ESGSN_BSS_1_7 t_net_ESGSN_BSS_1_8
t_net_ESGSN_BSS_1_9]
t_net_ESGSN_BSS_2=[t_net_ESGSN_BSS_2_1 t_net_ESGSN_BSS_2_2
t_net_ESGSN_BSS_2_3 t_net_ESGSN_BSS_2_4 t_net_ESGSN_BSS_2_5
t_net_ESGSN_BSS_2_6 t_net_ESGSN_BSS_2_7 t_net_ESGSN_BSS_2_8
t_net_ESGSN_BSS_2_9]
t_net_ESGSN_BSS_3=[t_net_ESGSN_BSS_3_1 t_net_ESGSN_BSS_3_2
t_net_ESGSN_BSS_3_3 t_net_ESGSN_BSS_3_4 t_net_ESGSN_BSS_3_5
t_net_ESGSN_BSS_3_6 t_net_ESGSN_BSS_3_7 t_net_ESGSN_BSS_3_8
t_net_ESGSN_BSS_3_9]
t_net_ESGSN_BSS_4=[t_net_ESGSN_BSS_4_1 t_net_ESGSN_BSS_4_2
t_net_ESGSN_BSS_4_3 t_net_ESGSN_BSS_4_4 t_net_ESGSN_BSS_4_5
t_net_ESGSN_BSS_4_6 t_net_ESGSN_BSS_4_7 t_net_ESGSN_BSS_4_8
t_net_ESGSN_BSS_4_9]
t_net_ESGSN_BSS_5=[t_net_ESGSN_BSS_5_1 t_net_ESGSN_BSS_5_2
t_net_ESGSN_BSS_5_3 t_net_ESGSN_BSS_5_4 t_net_ESGSN_BSS_5_5
t_net_ESGSN_BSS_5_6 t_net_ESGSN_BSS_5_7 t_net_ESGSN_BSS_5_8
t_net_ESGSN_BSS_5_9]
t_net_ESGSN_BSS_6=[t_net_ESGSN_BSS_6_1 t_net_ESGSN_BSS_6_2
t_net_ESGSN_BSS_6_3 t_net_ESGSN_BSS_6_4 t_net_ESGSN_BSS_6_5
t_net_ESGSN_BSS_6_6 t_net_ESGSN_BSS_6_7 t_net_ESGSN_BSS_6_8
t_net_ESGSN_BSS_6_9]
t_net_ESGSN_BSS_7=[t_net_ESGSN_BSS_7_1 t_net_ESGSN_BSS_7_2
t_net_ESGSN_BSS_7_3 t_net_ESGSN_BSS_7_4 t_net_ESGSN_BSS_7_5
t_net_ESGSN_BSS_7_6 t_net_ESGSN_BSS_7_7 t_net_ESGSN_BSS_7_8
t_net_ESGSN_BSS_7_9]
t_net_ESGSN_BSS_8=[t_net_ESGSN_BSS_8_1 t_net_ESGSN_BSS_8_2
t_net_ESGSN_BSS_8_3 t_net_ESGSN_BSS_8_4 t_net_ESGSN_BSS_8_5
t_net_ESGSN_BSS_8_6 t_net_ESGSN_BSS_8_7 t_net_ESGSN_BSS_8_8
t_net_ESGSN_BSS_8_9]
t_net_ESGSN_BSS_9=[t_net_ESGSN_BSS_9_1 t_net_ESGSN_BSS_9_2
t_net_ESGSN_BSS_9_3 t_net_ESGSN_BSS_9_4 t_net_ESGSN_BSS_9_5
t_net_ESGSN_BSS_9_6 t_net_ESGSN_BSS_9_7 t_net_ESGSN_BSS_9_8
t_net_ESGSN_BSS_9_9]

t_D_BSS_down=(n_frame_LLC_fix*n_frame_NS_fix*W_frame_NS_down*8)/V_PCU1
t_E_BSS_down=(n_frame_LLC_fix*n_frame_RLCradioblk_down*RLC_MAC_radioblk)/V_PCU2
t_T_Um_down=(RLC_MAC_radioblk.*n_frame_RLCradioblk_down)./(2*V_EUE2)
t_Prop_BSS_UE=(R UE_BTS.*n_frame_RLCradioblk_down)/v_UDR

t_net_BSS_UE=t_D_BSS_down+t_E_BSS_down+t_T_Um_down+t_Prop_BSS_UE

h_PDCP=1;
```

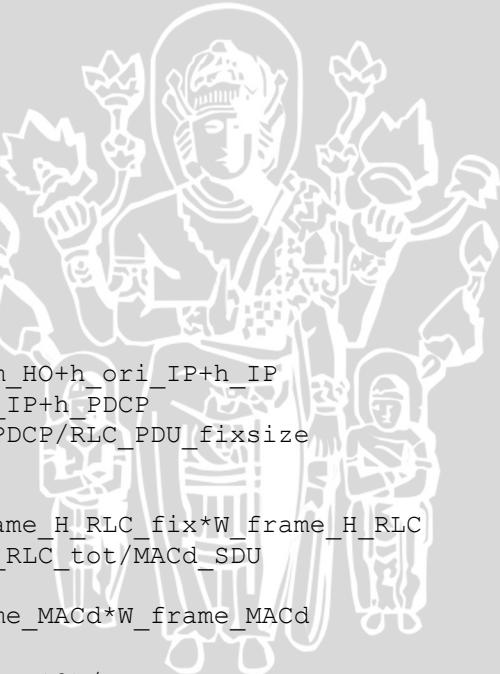
RLC_PDU_fixsize=40;
MACd_SDU=42;
h_MAC=3;
V_HUE=3.6*(10^6);
n=3;
R_UE_NodeB=400;
V_NodeB=155.52*(10^6);
V_Iub=155.52*(10^6);
R_NodeB_RNC=400;
V_RNC=155.52*(10^6);
h_FP=7;
CRC=2;
h_AAL2=3;
h_ATM=5;
V_Iu=155.52*(10^6);
R_RNC_HSGSN=3000;
V_HSGSN=100*(10^6);
h_GTP=8;
h_UDP=8;
h_AAL5=8;
V_HSGSN2=155.52*(10^6);
V_Gn=100*(10^6);
R_HSGSN_GGSN=2000;
h_MAP=13;
h_TCAP=13;
h_SCCP=13;
CRC_HLR=3;
V_Gr=10^9;
R_GGSN_HLR=1000;
n_frame_GGSN=1;
h_RLC=2;
MACd_SDU=42;
PDU_SAR=48;

W_frame_H_IP=W_datagram_H0+h_ori_IP+h_IP
W_frame_PDCP=W_frame_H_IP+h_PDCP
n_frame_H_RLC=W_frame_PDCP/RLC_PDU_fixsize
n_frame_H_RLC_fix=11
W_frame_H_RLC=42
W_frame_H_RLC_tot=n_frame_H_RLC_fix*W_frame_H_RLC
n_frame_MACd=W_frame_H_RLC_tot/MACd_SDU
W_frame_MACd=h_MAC+42
W_frame_MACd_tot=n_frame_MACd*W_frame_MACd

t_E_HUE=(W_frame_MACd_tot*8)/V_HUE
t_T_Uu=(W_frame_MACd_tot*8)/(n*V_HUE)
t_Prop UE_NodeB=(n_frame_MACd*R_UE_NodeB)/v_UDR

W_datagram_NodeB=W_frame_MACd_tot-h_MAC-h_PDCP-h_ori_IP-h_IP
t_D_NodeB=(W_datagram_NodeB*8)/V_NodeB

n_frame_MACHs=W_datagram_NodeB/42
n_frame_MACHs_fix=11
W_frame_MACHs=h_MAC+42
W_frame_MACHs_tot=n_frame_MACHs_fix*W_frame_MACHs
t_E_NodeB=(W_frame_MACHs_tot*8)/V_NodeB
t_T_Iub=(W_frame_MACHs_tot*8)/V_Iub
t_Prop_NodeB_RNC=(n_frame_MACHs_fix*R_NodeB_RNC)/v_FO



```

t_net_HUE_RNC=t_E_HUE+t_T_Uu+t_Prop_UE_NodeB+t_D_NodeB+t_E_NodeB+t_T_I
ub+t_Prop_NodeB_RNC

W_datagram_RNC=W_frame_MAChs_tot-h_MAC
t_D_RNC=(W_datagram_RNC*8)/V_RNC

W_frame_FP=W_datagram_RNC+h_FP+CRC
W_frame_AAL2=W_frame_FP+h_AAL2
n_frame_ATM2=W_frame_AAL2/48
n_frame_ATM2_fix=11
W_frame_ATM2=48+h_ATM
W_frame_RNC=n_frame_ATM2_fix*W_frame_ATM2

t_E_RNC=(W_frame_RNC*8)/V_RNC
t_T_Iu=(W_frame_RNC*8)/V_Iu
t_Prop_RNC_HSGSN=(n_frame_ATM2_fix*R_RNC_HSGSN)/v_FO

t_net_RNC_HSGSN=t_D_RNC+t_E_RNC+t_T_Iu+t_Prop_RNC_HSGSN

W_datagram_HSGSN=W_frame_RNC-h_ATM-h_AAL2-h_FP-CRC
t_D_HSGSN=(W_datagram_HSGSN*8)/V_HSGSN

W_frame_AAL5=W_datagram_HSGSN+h_GTP+h_IP+h_UDP+h_AAL5
n_frame_ATM=W_frame_AAL5/48
n_frame_ATM_fix=14
W_frame_ATM=48+h_ATM
W_frame_HSGSN=n_frame_ATM_fix*W_frame_ATM
t_E_HSGSN=(W_frame_HSGSN*8)/V_HSGSN2
t_T_Gn=(W_frame_HSGSN*8)/V_Gn
t_prop_HSGSN_GGSN=(n_frame_ATM_fix*R_HSGSN_GGSN)/v_FO

C_HSGSN=10^8;
L_HSGSN=W_frame_HSGSN*8
u_HSGSN = C_HSGSN/L_HSGSN
lambda_HSGSN = u_HSGSN*p
x_4 = u_HSGSN-lambda_HSGSN;
y_4 = lambda_HSGSN./[u_HSGSN * [x_4]];
z_4 = 1 / u_HSGSN;
tw_4 = y_3 + z_4

t_net_HSGSN_GGSN=t_D_HSGSN+t_E_HSGSN+t_T_Gn+t_prop_HSGSN_GGSN+tw_4

w_datagram_HLR=W_frame_HSGSN-h_ATM-h_GTP-h_IP-h_UDP-h_AAL5
t_D_HLR=(w_datagram_HLR*8)/V_GGSN

W_frame_HLR=w_datagram_HLR+h_MAP+h_TCAP+h_SCCP+CRC_HLR
t_E_HLR=(W_frame_HLR*8)/V_GGSN
t_T_Gr=(W_frame_HLR*8)/V_Gr
t_Prop_GGSN_HLR=R_GGSN_HLR/v_UDR

t_net_GGSN_HLR=t_D_HLR+t_E_HLR+t_T_Gr+t_Prop_GGSN_HLR

t_net_HLR_GGSN=t_net_GGSN_HLR

t_D_GGSN_down=(W_datagram_HO*8)/V_GGSN
W_frame_GGSN_down=W_datagram_HO+h_GTP+h_UDP+h_IP+h_Ethernet+FCS2
t_E_GGSN_down=(W_frame_GGSN_down*8)/V_GGSN
t_T_Gn_down=(W_frame_GGSN_down*8)/V_Gn
t_prop_GGSN_HSGSN_down=(n_frame_GGSN*R_HSGSN_GGSN)/v_FO

```

```
C_GGSN=10^8;
L_GGSN_down=W_frame_GGSN_down*8
u_GGSN_down = C_GGSN/L_GGSN_down
lambda_GGSN_down = u_GGSN_down*p
x_5 = u_GGSN_down-lambda_GGSN_down;
y_5 = lambda_GGSN_down./ [u_GGSN_down * [x_5]];
z_5 = 1 / u_GGSN_down;
tw_GGSN_down = y_5 + z_5

t_net_GGSN_HSGSN=t_D_GGSN_down+t_E_GGSN_down+t_T_Gn_down+t_prop_GGSN_H
SGSN_down+tw_GGSN_down

t_D_HSGSN_down=(W_datagram_HO*8)/V_HSGSN
W_frame_AAL5_down=W_datagram_HO+h_GTP+h_IP+h_UDP+h_AAL5
n_frame_ATM_down=W_frame_AAL5_down/48
n_frame_ATM_down_fix=10;
W_frame_ATM_down=48+h_ATM;
W_frame_HSGSN_down=n_frame_ATM_down_fix*W_frame_ATM_down
t_E_HSGSN_down=(W_frame_HSGSN_down*8)/V_HSGSN2
t_T_Iu_down=(W_frame_HSGSN_down*8)/V_Iu
t_Prop_HSGSN_RNC_down=(n_frame_ATM_down_fix*R_RNC_HSGSN)/v_FO

C_HSGSN=10^8;
L_HSGSN_down=W_frame_HSGSN_down*8
u_HSGSN_down = C_HSGSN/L_HSGSN_down
lambda_HSGSN_down = u_HSGSN_down*p
x_6 = u_HSGSN_down-lambda_HSGSN_down;
y_6 = lambda_HSGSN_down./ [u_HSGSN_down * [x_5]];
z_6 = 1 / u_HSGSN_down;
tw_6 = y_6 + z_6

t_net_HSGSN_RNC_down=t_D_HSGSN_down+t_E_HSGSN_down+t_T_Iu_down+t_Prop_
HSGSN_RNC_down+tw_6

W_datagram_RNC_down=W_frame_HSGSN_down-h_ATM-h_GTP-h_IP-h_UDP-h_AAL5
t_D_RNC_down=(W_datagram_RNC_down*8)/V_RNC

W_frame_PDCP_down=W_datagram_RNC_down+h_PDCP
n_frame_RLC_down=W_frame_PDCP_down/40
n_frame_RLC_down_fix=12
W_frame_RLC_down=40+h_RLC
W_frame_RLC_tot_down=n_frame_RLC_down_fix*W_frame_RLC_down
n_frame_MACd_down=W_frame_RLC_tot_down/MACd_SDU
W_frame_MACd_down=h_MAC+MACd_SDU
W_frame_FP_down=W_frame_MACd_down+h_FP+CRC
W_frame_FP_tot_down=n_frame_MACd_down*W_frame_FP_down
W_frame_AAL2_down=W_frame_FP_tot_down+h_AAL2
n_frame_ATM2_down=W_frame_AAL2_down/PDU_SAR
n_frame_ATM2_down_fix=14;
W_frame_RNC_down=n_frame_ATM2_down_fix*W_frame_ATM2
t_E_RNC_down=(W_frame_RNC_down*8)/V_RNC

t_T_Iub_down=(W_frame_RNC_down*8)/V_Iub
t_Prop_RNC_NodeB_down=(n_frame_ATM2_down_fix*R_NodeB_RNC)/v_FO

W_datagram_NodeB_down=W_frame_RNC_down-h_ATM-h_AAL2-h_FP-CRC
t_D_NodeB_down=(W_datagram_NodeB_down*8)/V_NodeB

n_frame_MACHs_down=W_datagram_NodeB_down/42
n_frame_MACHs_down_fix=18
```

W_frame_MACHs_down=h_MAC+42
W_frame_NodeB_down=n_frame_MACHs_down_fix*W_frame_MACHs_down
t_E_NodeB_down=(W_frame_NodeB_down*8)/V_NodeB

t_T_Uu_down=(W_frame_NodeB_down*8)/(n*V_HUE)
t_Prop_NodeB_UE_down=(n_frame_MACHs_down_fix*R UE_NodeB)/v_UDR

t_net_RAN_UE=t_D_RNC_down+t_E_RNC_down+t_T_Iub_down+t_Prop_RNC_NodeB_d
own+t_D_NodeB_down+t_E_NodeB_down+t_T_Uu_down+t_Prop_NodeB_UE_down

RNC_UE=t_net_RAN_UE
UE_ESGSN_1=t_net_EUE_BTS+t_net_BSS_ESGSN
ESGSN_HSGSN_1=t_net_ESGSN_GGSN+t_net_GGSN_HSGSN
HSGSN_RAN_1=t_net_HSGSN_RNC_down
RAN_HSGSN_1=t_net_RNC_HSGSN
HSGSN_ESGSN=t_net_HSGSN_GGSN+t_net_GGSN_ESGSN
HLR_UE=t_net_HLR_GGSN+t_net_GGSN_HSGSN+t_net_HSGSN_RNC_down+t_net_RAN_UE
ESGSN_HSGSN_2=t_net_ESGSN_GGSN+t_net_GGSN_HSGSN
HSGSN_RAN_2=t_net_HSGSN_RNC_down
ESGSN_GGSN=t_net_ESGSN_GGSN
GGSN_ESGSN=t_net_GGSN_ESGSN
ESGSN_HLR_1=t_net_ESGSN_GGSN+t_net_GGSN_HLR
HSGSN_HLR_1=t_net_HSGSN_GGSN+t_net_GGSN_HLR
HSGSN_RAN_3=t_net_HSGSN_RNC_down
RAN_HSGSN_2=t_net_RNC_HSGSN
HSGSN_HLR_2=t_net_HSGSN_GGSN+t_net_GGSN_HLR
HLR_ESGSN_1=t_net_HLR_GGSN+t_net_GGSN_ESGSN
ESGSN_HLR_2=t_net_ESGSN_GGSN+t_net_GGSN_HLR
HLR_ESGSN_2=t_net_HLR_GGSN+t_net_GGSN_ESGSN

ESGSN_UE_1_1=t_net_ESGSN_BSS_1_1+t_net_BSS_UE(1,1)
ESGSN_UE_1_2=t_net_ESGSN_BSS_1_2+t_net_BSS_UE(1,1)
ESGSN_UE_1_3=t_net_ESGSN_BSS_1_3+t_net_BSS_UE(1,1)
ESGSN_UE_1_4=t_net_ESGSN_BSS_1_4+t_net_BSS_UE(1,1)
ESGSN_UE_1_5=t_net_ESGSN_BSS_1_5+t_net_BSS_UE(1,1)
ESGSN_UE_1_6=t_net_ESGSN_BSS_1_6+t_net_BSS_UE(1,1)
ESGSN_UE_1_7=t_net_ESGSN_BSS_1_7+t_net_BSS_UE(1,1)
ESGSN_UE_1_8=t_net_ESGSN_BSS_1_8+t_net_BSS_UE(1,1)
ESGSN_UE_1_9=t_net_ESGSN_BSS_1_9+t_net_BSS_UE(1,1)

ESGSN_UE_2_1=t_net_ESGSN_BSS_2_1+t_net_BSS_UE(1,2)
ESGSN_UE_2_2=t_net_ESGSN_BSS_2_2+t_net_BSS_UE(1,2)
ESGSN_UE_2_3=t_net_ESGSN_BSS_2_3+t_net_BSS_UE(1,2)
ESGSN_UE_2_4=t_net_ESGSN_BSS_2_4+t_net_BSS_UE(1,2)
ESGSN_UE_2_5=t_net_ESGSN_BSS_2_5+t_net_BSS_UE(1,2)
ESGSN_UE_2_6=t_net_ESGSN_BSS_2_6+t_net_BSS_UE(1,2)
ESGSN_UE_2_7=t_net_ESGSN_BSS_2_7+t_net_BSS_UE(1,2)
ESGSN_UE_2_8=t_net_ESGSN_BSS_2_8+t_net_BSS_UE(1,2)
ESGSN_UE_2_9=t_net_ESGSN_BSS_2_9+t_net_BSS_UE(1,2)

ESGSN_UE_3_1=t_net_ESGSN_BSS_3_1+t_net_BSS_UE(1,3)
ESGSN_UE_3_2=t_net_ESGSN_BSS_3_2+t_net_BSS_UE(1,3)
ESGSN_UE_3_3=t_net_ESGSN_BSS_3_3+t_net_BSS_UE(1,3)
ESGSN_UE_3_4=t_net_ESGSN_BSS_3_4+t_net_BSS_UE(1,3)
ESGSN_UE_3_5=t_net_ESGSN_BSS_3_5+t_net_BSS_UE(1,3)
ESGSN_UE_3_6=t_net_ESGSN_BSS_3_6+t_net_BSS_UE(1,3)
ESGSN_UE_3_7=t_net_ESGSN_BSS_3_7+t_net_BSS_UE(1,3)
ESGSN_UE_3_8=t_net_ESGSN_BSS_3_8+t_net_BSS_UE(1,3)
ESGSN_UE_3_9=t_net_ESGSN_BSS_3_9+t_net_BSS_UE(1,3)



ESGSN UE 4_1=t net ESGSN BSS 4_1+t net BSS UE (1, 4)
ESGSN UE 4_2=t net ESGSN BSS 4_2+t net BSS UE (1, 4)
ESGSN UE 4_3=t net ESGSN BSS 4_3+t net BSS UE (1, 4)
ESGSN UE 4_4=t net ESGSN BSS 4_4+t net BSS UE (1, 4)
ESGSN UE 4_5=t net ESGSN BSS 4_5+t net BSS UE (1, 4)
ESGSN UE 4_6=t net ESGSN BSS 4_6+t net BSS UE (1, 4)
ESGSN UE 4_7=t net ESGSN BSS 4_7+t net BSS UE (1, 4)
ESGSN UE 4_8=t net ESGSN BSS 4_8+t net BSS UE (1, 4)
ESGSN UE 4_9=t net ESGSN BSS 4_9+t net BSS UE (1, 4)

ESGSN UE 5_1=t net ESGSN BSS 5_1+t net BSS UE (1, 5)
ESGSN UE 5_2=t net ESGSN BSS 5_2+t net BSS UE (1, 5)
ESGSN UE 5_3=t net ESGSN BSS 5_3+t net BSS UE (1, 5)
ESGSN UE 5_4=t net ESGSN BSS 5_4+t net BSS UE (1, 5)
ESGSN UE 5_5=t net ESGSN BSS 5_5+t net BSS UE (1, 5)
ESGSN UE 5_6=t net ESGSN BSS 5_6+t net BSS UE (1, 5)
ESGSN UE 5_7=t net ESGSN BSS 5_7+t net BSS UE (1, 5)
ESGSN UE 5_8=t net ESGSN BSS 5_8+t net BSS UE (1, 5)
ESGSN UE 5_9=t net ESGSN BSS 5_9+t net BSS UE (1, 5)

ESGSN UE 6_1=t net ESGSN BSS 6_1+t net BSS UE (1, 6)
ESGSN UE 6_2=t net ESGSN BSS 6_2+t net BSS UE (1, 6)
ESGSN UE 6_3=t net ESGSN BSS 6_3+t net BSS UE (1, 6)
ESGSN UE 6_4=t net ESGSN BSS 6_4+t net BSS UE (1, 6)
ESGSN UE 6_5=t net ESGSN BSS 6_5+t net BSS UE (1, 6)
ESGSN UE 6_6=t net ESGSN BSS 6_6+t net BSS UE (1, 6)
ESGSN UE 6_7=t net ESGSN BSS 6_7+t net BSS UE (1, 6)
ESGSN UE 6_8=t net ESGSN BSS 6_8+t net BSS UE (1, 6)
ESGSN UE 6_9=t net ESGSN BSS 6_9+t net BSS UE (1, 6)

ESGSN UE 7_1=t net ESGSN BSS 7_1+t net BSS UE (1, 7)
ESGSN UE 7_2=t net ESGSN BSS 7_2+t net BSS UE (1, 7)
ESGSN UE 7_3=t net ESGSN BSS 7_3+t net BSS UE (1, 7)
ESGSN UE 7_4=t net ESGSN BSS 7_4+t net BSS UE (1, 7)
ESGSN UE 7_5=t net ESGSN BSS 7_5+t net BSS UE (1, 7)
ESGSN UE 7_6=t net ESGSN BSS 7_6+t net BSS UE (1, 7)
ESGSN UE 7_7=t net ESGSN BSS 7_7+t net BSS UE (1, 7)
ESGSN UE 7_8=t net ESGSN BSS 7_8+t net BSS UE (1, 7)
ESGSN UE 7_9=t net ESGSN BSS 7_9+t net BSS UE (1, 7)

ESGSN UE 8_1=t net ESGSN BSS 8_1+t net BSS UE (1, 8)
ESGSN UE 8_2=t net ESGSN BSS 8_2+t net BSS UE (1, 8)
ESGSN UE 8_3=t net ESGSN BSS 8_3+t net BSS UE (1, 8)
ESGSN UE 8_4=t net ESGSN BSS 8_4+t net BSS UE (1, 8)
ESGSN UE 8_5=t net ESGSN BSS 8_5+t net BSS UE (1, 8)
ESGSN UE 8_6=t net ESGSN BSS 8_6+t net BSS UE (1, 8)
ESGSN UE 8_7=t net ESGSN BSS 8_7+t net BSS UE (1, 8)
ESGSN UE 8_8=t net ESGSN BSS 8_8+t net BSS UE (1, 8)
ESGSN UE 8_9=t net ESGSN BSS 8_9+t net BSS UE (1, 8)

ESGSN UE 9_1=t net ESGSN BSS 9_1+t net BSS UE (1, 9)
ESGSN UE 9_2=t net ESGSN BSS 9_2+t net BSS UE (1, 9)
ESGSN UE 9_3=t net ESGSN BSS 9_3+t net BSS UE (1, 9)
ESGSN UE 9_4=t net ESGSN BSS 9_4+t net BSS UE (1, 9)
ESGSN UE 9_5=t net ESGSN BSS 9_5+t net BSS UE (1, 9)
ESGSN UE 9_6=t net ESGSN BSS 9_6+t net BSS UE (1, 9)
ESGSN UE 9_7=t net ESGSN BSS 9_7+t net BSS UE (1, 9)
ESGSN UE 9_8=t net ESGSN BSS 9_8+t net BSS UE (1, 9)
ESGSN UE 9_9=t net ESGSN BSS 9_9+t net BSS UE (1, 9)

UE_ESGSN_2=t_net_EUE_BTS+t_net_BSS_ESGSN

```
t_HO_tot_1_1=RNC_UE+UE_ESGSN_1(1,1)+ESGSN_HGSN_1(1,1)+HGSN_RAN_1(1,1)
+RAN_HGSN_1+HGSN_ESGSN(1,1)+HLR_UE(1,1)+ESGSN_HGSN_2(1,1)+HGSN_RA
N_2(1,1)+ESGSN_GGSN(1,1)+GGSN_ESGSN(1,1)+ESGSN_HLR_1(1,1)+HGSN_HLR_1(
1,1)+HGSN_RAN_3(1,1)+RAN_HGSN_2+HGSN_HLR_2(1,1)+HLR_ESGSN_1(1,1)+ES
GSN_HLR_2(1,1)+HLR_ESGSN_2(1,1)+ESGSN_UE_1_1+UE_ESGSN_2(1,1)
t_HO_tot_1_2=RNC_UE+UE_ESGSN_1(1,1)+ESGSN_HGSN_1(1,2)+HGSN_RAN_1(1,2)
+RAN_HGSN_1+HGSN_ESGSN(1,2)+HLR_UE(1,2)+ESGSN_HGSN_2(1,2)+HGSN_RA
N_2(1,2)+ESGSN_GGSN(1,2)+GGSN_ESGSN(1,2)+ESGSN_HLR_1(1,2)+HGSN_HLR_1(
1,2)+HGSN_RAN_3(1,2)+RAN_HGSN_2+HGSN_HLR_2(1,2)+HLR_ESGSN_1(1,2)+ES
GSN_HLR_2(1,2)+HLR_ESGSN_2(1,2)+ESGSN_UE_1_2+UE_ESGSN_2(1,1)
t_HO_tot_1_3=RNC_UE+UE_ESGSN_1(1,1)+ESGSN_HGSN_1(1,3)+HGSN_RAN_1(1,3)
+RAN_HGSN_1+HGSN_ESGSN(1,3)+HLR_UE(1,3)+ESGSN_HGSN_2(1,3)+HGSN_RA
N_2(1,3)+ESGSN_GGSN(1,3)+GGSN_ESGSN(1,3)+ESGSN_HLR_1(1,3)+HGSN_HLR_1(
1,3)+HGSN_RAN_3(1,3)+RAN_HGSN_2+HGSN_HLR_2(1,3)+HLR_ESGSN_1(1,3)+ES
GSN_HLR_2(1,3)+HLR_ESGSN_2(1,3)+ESGSN_UE_1_3+UE_ESGSN_2(1,1)
t_HO_tot_1_4=RNC_UE+UE_ESGSN_1(1,1)+ESGSN_HGSN_1(1,4)+HGSN_RAN_1(1,4)
+RAN_HGSN_1+HGSN_ESGSN(1,4)+HLR_UE(1,4)+ESGSN_HGSN_2(1,4)+HGSN_RA
N_2(1,4)+ESGSN_GGSN(1,4)+GGSN_ESGSN(1,4)+ESGSN_HLR_1(1,4)+HGSN_HLR_1(
1,4)+HGSN_RAN_3(1,4)+RAN_HGSN_2+HGSN_HLR_2(1,4)+HLR_ESGSN_1(1,4)+ES
GSN_HLR_2(1,4)+HLR_ESGSN_2(1,4)+ESGSN_UE_1_4+UE_ESGSN_2(1,1)
t_HO_tot_1_5=RNC_UE+UE_ESGSN_1(1,1)+ESGSN_HGSN_1(1,5)+HGSN_RAN_1(1,5)
+RAN_HGSN_1+HGSN_ESGSN(1,5)+HLR_UE(1,5)+ESGSN_HGSN_2(1,5)+HGSN_RA
N_2(1,5)+ESGSN_GGSN(1,5)+GGSN_ESGSN(1,5)+ESGSN_HLR_1(1,5)+HGSN_HLR_1(
1,5)+HGSN_RAN_3(1,5)+RAN_HGSN_2+HGSN_HLR_2(1,5)+HLR_ESGSN_1(1,5)+ES
GSN_HLR_2(1,5)+HLR_ESGSN_2(1,5)+ESGSN_UE_1_5+UE_ESGSN_2(1,1)
t_HO_tot_1_6=RNC_UE+UE_ESGSN_1(1,1)+ESGSN_HGSN_1(1,6)+HGSN_RAN_1(1,6)
+RAN_HGSN_1+HGSN_ESGSN(1,6)+HLR_UE(1,6)+ESGSN_HGSN_2(1,6)+HGSN_RA
N_2(1,6)+ESGSN_GGSN(1,6)+GGSN_ESGSN(1,6)+ESGSN_HLR_1(1,6)+HGSN_HLR_1(
1,6)+HGSN_RAN_3(1,6)+RAN_HGSN_2+HGSN_HLR_2(1,6)+HLR_ESGSN_1(1,6)+ES
GSN_HLR_2(1,6)+HLR_ESGSN_2(1,6)+ESGSN_UE_1_6+UE_ESGSN_2(1,1)
t_HO_tot_1_7=RNC_UE+UE_ESGSN_1(1,1)+ESGSN_HGSN_1(1,7)+HGSN_RAN_1(1,7)
+RAN_HGSN_1+HGSN_ESGSN(1,7)+HLR_UE(1,7)+ESGSN_HGSN_2(1,7)+HGSN_RA
N_2(1,7)+ESGSN_GGSN(1,7)+GGSN_ESGSN(1,7)+ESGSN_HLR_1(1,7)+HGSN_HLR_1(
1,7)+HGSN_RAN_3(1,7)+RAN_HGSN_2+HGSN_HLR_2(1,7)+HLR_ESGSN_1(1,7)+ES
GSN_HLR_2(1,7)+HLR_ESGSN_2(1,7)+ESGSN_UE_1_7+UE_ESGSN_2(1,1)
t_HO_tot_1_8=RNC_UE+UE_ESGSN_1(1,1)+ESGSN_HGSN_1(1,8)+HGSN_RAN_1(1,8)
+RAN_HGSN_1+HGSN_ESGSN(1,8)+HLR_UE(1,8)+ESGSN_HGSN_2(1,8)+HGSN_RA
N_2(1,8)+ESGSN_GGSN(1,8)+GGSN_ESGSN(1,8)+ESGSN_HLR_1(1,8)+HGSN_HLR_1(
1,8)+HGSN_RAN_3(1,8)+RAN_HGSN_2+HGSN_HLR_2(1,8)+HLR_ESGSN_1(1,8)+ES
GSN_HLR_2(1,8)+HLR_ESGSN_2(1,8)+ESGSN_UE_1_8+UE_ESGSN_2(1,1)
t_HO_tot_1_9=RNC_UE+UE_ESGSN_1(1,1)+ESGSN_HGSN_1(1,9)+HGSN_RAN_1(1,9)
+RAN_HGSN_1+HGSN_ESGSN(1,9)+HLR_UE(1,9)+ESGSN_HGSN_2(1,9)+HGSN_RA
N_2(1,9)+ESGSN_GGSN(1,9)+GGSN_ESGSN(1,9)+ESGSN_HLR_1(1,9)+HGSN_HLR_1(
1,9)+HGSN_RAN_3(1,9)+RAN_HGSN_2+HGSN_HLR_2(1,9)+HLR_ESGSN_1(1,9)+ES
GSN_HLR_2(1,9)+HLR_ESGSN_2(1,9)+ESGSN_UE_1_9+UE_ESGSN_2(1,1)

t_HO_tot_2_1=RNC_UE+UE_ESGSN_1(1,2)+ESGSN_HGSN_1(1,1)+HGSN_RAN_1(1,1)
+RAN_HGSN_1+HGSN_ESGSN(1,1)+HLR_UE(1,1)+ESGSN_HGSN_2(1,1)+HGSN_RA
N_2(1,1)+ESGSN_GGSN(1,1)+GGSN_ESGSN(1,1)+ESGSN_HLR_1(1,1)+HGSN_HLR_1(
1,1)+HGSN_RAN_3(1,1)+RAN_HGSN_2+HGSN_HLR_2(1,1)+HLR_ESGSN_1(1,1)+ES
GSN_HLR_2(1,1)+HLR_ESGSN_2(1,1)+ESGSN_UE_2_1+UE_ESGSN_2(1,2)
t_HO_tot_2_2=RNC_UE+UE_ESGSN_1(1,2)+ESGSN_HGSN_1(1,2)+HGSN_RAN_1(1,2)
+RAN_HGSN_1+HGSN_ESGSN(1,2)+HLR_UE(1,2)+ESGSN_HGSN_2(1,2)+HGSN_RA
N_2(1,2)+ESGSN_GGSN(1,2)+GGSN_ESGSN(1,2)+ESGSN_HLR_1(1,2)+HGSN_HLR_1(
1,2)+HGSN_RAN_3(1,2)+RAN_HGSN_2+HGSN_HLR_2(1,2)+HLR_ESGSN_1(1,2)+ES
GSN_HLR_2(1,2)+HLR_ESGSN_2(1,2)+ESGSN_UE_2_2+UE_ESGSN_2(1,2)
t_HO_tot_2_3=RNC_UE+UE_ESGSN_1(1,2)+ESGSN_HGSN_1(1,3)+HGSN_RAN_1(1,3)
+RAN_HGSN_1+HGSN_ESGSN(1,3)+HLR_UE(1,3)+ESGSN_HGSN_2(1,3)+HGSN_RA
N_2(1,3)+ESGSN_GGSN(1,3)+GGSN_ESGSN(1,3)+ESGSN_HLR_1(1,3)+HGSN_HLR_1(
1,3)+HGSN_RAN_3(1,3)+RAN_HGSN_2+HGSN_HLR_2(1,3)+HLR_ESGSN_1(1,3)+ES
GSN_HLR_2(1,3)+HLR_ESGSN_2(1,3)+ESGSN_UE_1_3+UE_ESGSN_2(1,1)
```





1, 3) + HSGSN_RAN_3(1, 3) + RAN_HSGSN_2 + HSGSN_HLR_2(1, 3) + HLR_ESGSN_1(1, 3) + ESGSN_HLR_2(1, 3) + HLR_ESGSN_2(1, 3) + ESGSN_UE_2_3 + UE_ESGSN_2(1, 2)
t_HO_tot_2_4=RNC_UE+UE_ESGSN_1(1, 2)+ESGSN_HGSN_1(1, 4)+HSGSN_RAN_1(1, 4)+RAN_HSGSN_1+HSGSN_ESGSN(1, 4)+HLR_UE(1, 4)+ESGSN_HGSN_2(1, 4)+HSGSN_RAN_2(1, 4)+ESGSN_GGSN(1, 4)+GGSN_ESGSN(1, 4)+ESGSN_HLR_1(1, 4)+HSGSN_RAN_3(1, 4)+RAN_HSGSN_2+HSGSN_HLR_2(1, 4)+HLR_ESGSN_1(1, 4)+ESGSN_HLR_2(1, 4)+HLR_ESGSN_2(1, 4)+ESGSN_UE_2_4+UE_ESGSN_2(1, 2)
t_HO_tot_2_5=RNC_UE+UE_ESGSN_1(1, 2)+ESGSN_HGSN_1(1, 5)+HSGSN_RAN_1(1, 5)+RAN_HSGSN_1+HSGSN_ESGSN(1, 5)+HLR_UE(1, 5)+ESGSN_HGSN_2(1, 5)+HSGSN_RAN_2(1, 5)+ESGSN_GGSN(1, 5)+GGSN_ESGSN(1, 5)+ESGSN_HLR_1(1, 5)+HSGSN_RAN_3(1, 5)+RAN_HSGSN_2+HSGSN_HLR_2(1, 5)+HLR_ESGSN_1(1, 5)+ESGSN_HLR_2(1, 5)+HLR_ESGSN_2(1, 5)+ESGSN_UE_2_5+UE_ESGSN_2(1, 2)
t_HO_tot_2_6=RNC_UE+UE_ESGSN_1(1, 2)+ESGSN_HGSN_1(1, 6)+HSGSN_RAN_1(1, 6)+RAN_HSGSN_1+HSGSN_ESGSN(1, 6)+HLR_UE(1, 6)+ESGSN_HGSN_2(1, 6)+HSGSN_RAN_2(1, 6)+ESGSN_GGSN(1, 6)+GGSN_ESGSN(1, 6)+ESGSN_HLR_1(1, 6)+HSGSN_HLR_1(1, 6)+HSGSN_RAN_3(1, 6)+RAN_HSGSN_2+HSGSN_HLR_2(1, 6)+HLR_ESGSN_1(1, 6)+ESGSN_HLR_2(1, 6)+HLR_ESGSN_2(1, 6)+ESGSN_UE_2_6+UE_ESGSN_2(1, 2)
t_HO_tot_2_7=RNC_UE+UE_ESGSN_1(1, 2)+ESGSN_HGSN_1(1, 7)+HSGSN_RAN_1(1, 7)+RAN_HSGSN_1+HSGSN_ESGSN(1, 7)+HLR_UE(1, 7)+ESGSN_HGSN_2(1, 7)+HSGSN_RAN_2(1, 7)+ESGSN_GGSN(1, 7)+GGSN_ESGSN(1, 7)+ESGSN_HLR_1(1, 7)+HSGSN_HLR_1(1, 7)+HSGSN_RAN_3(1, 7)+RAN_HSGSN_2+HSGSN_HLR_2(1, 7)+HLR_ESGSN_1(1, 7)+ESGSN_HLR_2(1, 7)+HLR_ESGSN_2(1, 7)+ESGSN_UE_2_7+UE_ESGSN_2(1, 2)
t_HO_tot_2_8=RNC_UE+UE_ESGSN_1(1, 2)+ESGSN_HGSN_1(1, 8)+HSGSN_RAN_1(1, 8)+RAN_HSGSN_1+HSGSN_ESGSN(1, 8)+HLR_UE(1, 8)+ESGSN_HGSN_2(1, 8)+HSGSN_RAN_2(1, 8)+ESGSN_GGSN(1, 8)+GGSN_ESGSN(1, 8)+ESGSN_HLR_1(1, 8)+HSGSN_HLR_1(1, 8)+HSGSN_RAN_3(1, 8)+RAN_HSGSN_2+HSGSN_HLR_2(1, 8)+HLR_ESGSN_1(1, 8)+ESGSN_HLR_2(1, 8)+HLR_ESGSN_2(1, 8)+ESGSN_UE_2_8+UE_ESGSN_2(1, 2)
t_HO_tot_2_9=RNC_UE+UE_ESGSN_1(1, 2)+ESGSN_HGSN_1(1, 9)+HSGSN_RAN_1(1, 9)+RAN_HSGSN_1+HSGSN_ESGSN(1, 9)+HLR_UE(1, 9)+ESGSN_HGSN_2(1, 9)+HSGSN_RAN_2(1, 9)+ESGSN_GGSN(1, 9)+GGSN_ESGSN(1, 9)+ESGSN_HLR_1(1, 9)+HSGSN_HLR_1(1, 9)+HSGSN_RAN_3(1, 9)+RAN_HSGSN_2+HSGSN_HLR_2(1, 9)+HLR_ESGSN_1(1, 9)+ESGSN_HLR_2(1, 9)+HLR_ESGSN_2(1, 9)+ESGSN_UE_2_9+UE_ESGSN_2(1, 2)

t_HO_tot_3_1=RNC_UE+UE_ESGSN_1(1, 3)+ESGSN_HGSN_1(1, 1)+HSGSN_RAN_1(1, 1)+RAN_HSGSN_1+HSGSN_ESGSN(1, 1)+HLR_UE(1, 1)+ESGSN_HGSN_2(1, 1)+HSGSN_RAN_2(1, 1)+ESGSN_GGSN(1, 1)+GGSN_ESGSN(1, 1)+ESGSN_HLR_1(1, 1)+HSGSN_HLR_1(1, 1)+HSGSN_RAN_3(1, 1)+RAN_HSGSN_2+HSGSN_HLR_2(1, 1)+HLR_ESGSN_1(1, 1)+ESGSN_HLR_2(1, 1)+HLR_ESGSN_2(1, 1)+ESGSN_UE_3_1+UE_ESGSN_2(1, 3)
t_HO_tot_3_2=RNC_UE+UE_ESGSN_1(1, 3)+ESGSN_HGSN_1(1, 2)+HSGSN_RAN_1(1, 2)+RAN_HSGSN_1+HSGSN_ESGSN(1, 2)+HLR_UE(1, 2)+ESGSN_HGSN_2(1, 2)+HSGSN_RAN_2(1, 2)+ESGSN_GGSN(1, 2)+GGSN_ESGSN(1, 2)+ESGSN_HLR_1(1, 2)+HSGSN_HLR_1(1, 2)+HSGSN_RAN_3(1, 2)+RAN_HSGSN_2+HSGSN_HLR_2(1, 2)+HLR_ESGSN_1(1, 2)+ESGSN_HLR_2(1, 2)+HLR_ESGSN_2(1, 2)+ESGSN_UE_3_2+UE_ESGSN_2(1, 3)
t_HO_tot_3_3=RNC_UE+UE_ESGSN_1(1, 3)+ESGSN_HGSN_1(1, 3)+HSGSN_RAN_1(1, 3)+RAN_HSGSN_1+HSGSN_ESGSN(1, 3)+HLR_UE(1, 3)+ESGSN_HGSN_2(1, 3)+HSGSN_RAN_2(1, 3)+ESGSN_GGSN(1, 3)+GGSN_ESGSN(1, 3)+ESGSN_HLR_1(1, 3)+HSGSN_HLR_1(1, 3)+HSGSN_RAN_3(1, 3)+RAN_HSGSN_2+HSGSN_HLR_2(1, 3)+HLR_ESGSN_1(1, 3)+ESGSN_HLR_2(1, 3)+HLR_ESGSN_2(1, 3)+ESGSN_UE_3_3+UE_ESGSN_2(1, 3)
t_HO_tot_3_4=RNC_UE+UE_ESGSN_1(1, 3)+ESGSN_HGSN_1(1, 4)+HSGSN_RAN_1(1, 4)+RAN_HSGSN_1+HSGSN_ESGSN(1, 4)+HLR_UE(1, 4)+ESGSN_HGSN_2(1, 4)+HSGSN_RAN_2(1, 4)+ESGSN_GGSN(1, 4)+GGSN_ESGSN(1, 4)+ESGSN_HLR_1(1, 4)+HSGSN_HLR_1(1, 4)+HSGSN_RAN_3(1, 4)+RAN_HSGSN_2+HSGSN_HLR_2(1, 4)+HLR_ESGSN_1(1, 4)+ESGSN_HLR_2(1, 4)+HLR_ESGSN_2(1, 4)+ESGSN_UE_3_4+UE_ESGSN_2(1, 3)
t_HO_tot_3_5=RNC_UE+UE_ESGSN_1(1, 3)+ESGSN_HGSN_1(1, 5)+HSGSN_RAN_1(1, 5)+RAN_HSGSN_1+HSGSN_ESGSN(1, 5)+HLR_UE(1, 5)+ESGSN_HGSN_2(1, 5)+HSGSN_RAN_2(1, 5)+ESGSN_GGSN(1, 5)+GGSN_ESGSN(1, 5)+ESGSN_HLR_1(1, 5)+HSGSN_HLR_1(1, 5)+HSGSN_RAN_3(1, 5)+RAN_HSGSN_2+HSGSN_HLR_2(1, 5)+HLR_ESGSN_1(1, 5)+ESGSN_HLR_2(1, 5)+HLR_ESGSN_2(1, 5)+ESGSN_UE_3_5+UE_ESGSN_2(1, 3)
t_HO_tot_3_6=RNC_UE+UE_ESGSN_1(1, 3)+ESGSN_HGSN_1(1, 6)+HSGSN_RAN_1(1, 6)+RAN_HSGSN_1+HSGSN_ESGSN(1, 6)+HLR_UE(1, 6)+ESGSN_HGSN_2(1, 6)+HSGSN_RAN_2(1, 6)+ESGSN_GGSN(1, 6)+GGSN_ESGSN(1, 6)+ESGSN_HLR_1(1, 6)+HSGSN_HLR_1(1, 6)+HSGSN_RAN_3(1, 6)+RAN_HSGSN_2+HSGSN_HLR_2(1, 6)+HLR_ESGSN_1(1, 6)+ESGSN_HLR_2(1, 6)+HLR_ESGSN_2(1, 6)+ESGSN_UE_3_6+UE_ESGSN_2(1, 3)

1, 6) + HSGSN_RAN_3(1, 6) + RAN_HSGSN_2 + HSGSN_HLR_2(1, 6) + HLR_ESGSN_1(1, 6) + ESGSN_HLR_2(1, 6) + HLR_ESGSN_2(1, 6) + ESGSN_UE_3_6 + UE_ESGSN_2(1, 3)
t_HO_tot_3_7=RNC_UE+UE_ESGSN_1(1, 3)+ESGSN_HGSN_1(1, 7)+HSGSN_RAN_1(1, 7)+RAN_HSGSN_1+HSGSN_ESGSN(1, 7)+HLR_UE(1, 7)+ESGSN_HGSN_2(1, 7)+HSGSN_RAN_2(1, 7)+ESGSN_GGSN(1, 7)+GGSN_ESGSN(1, 7)+ESGSN_HLR_1(1, 7)+HSGSN_RAN_3(1, 7)+RAN_HSGSN_2+HSGSN_HLR_2(1, 7)+HLR_ESGSN_1(1, 7)+HSGSN_HLR_1(1, 7)+HSGSN_RAN_2(1, 7)+HLR_ESGSN_2(1, 7)+ESGSN_UE_3_7+UE_ESGSN_2(1, 3)
t_HO_tot_3_8=RNC_UE+UE_ESGSN_1(1, 3)+ESGSN_HGSN_1(1, 8)+HSGSN_RAN_1(1, 8)+RAN_HSGSN_1+HSGSN_ESGSN(1, 8)+HLR_UE(1, 8)+ESGSN_HGSN_2(1, 8)+HSGSN_RAN_2(1, 8)+ESGSN_GGSN(1, 8)+GGSN_ESGSN(1, 8)+ESGSN_HLR_1(1, 8)+HSGSN_HLR_1(1, 8)+HSGSN_RAN_3(1, 8)+RAN_HSGSN_2+HSGSN_HLR_2(1, 8)+HLR_ESGSN_1(1, 8)+ESGSN_HLR_2(1, 8)+HLR_ESGSN_2(1, 8)+ESGSN_UE_3_8+UE_ESGSN_2(1, 3)
t_HO_tot_3_9=RNC_UE+UE_ESGSN_1(1, 3)+ESGSN_HGSN_1(1, 9)+HSGSN_RAN_1(1, 9)+RAN_HSGSN_1+HSGSN_ESGSN(1, 9)+HLR_UE(1, 9)+ESGSN_HGSN_2(1, 9)+HSGSN_RAN_2(1, 9)+ESGSN_GGSN(1, 9)+GGSN_ESGSN(1, 9)+ESGSN_HLR_1(1, 9)+HSGSN_HLR_1(1, 9)+HSGSN_RAN_3(1, 9)+RAN_HSGSN_2+HSGSN_HLR_2(1, 9)+HLR_ESGSN_1(1, 9)+ESGSN_HLR_2(1, 9)+HLR_ESGSN_2(1, 9)+ESGSN_UE_3_9+UE_ESGSN_2(1, 3)

t_HO_tot_4_1=RNC_UE+UE_ESGSN_1(1, 4)+ESGSN_HGSN_1(1, 1)+HSGSN_RAN_1(1, 1)+RAN_HSGSN_1+HSGSN_ESGSN(1, 1)+HLR_UE(1, 1)+ESGSN_HGSN_2(1, 1)+HSGSN_RAN_2(1, 1)+ESGSN_GGSN(1, 1)+GGSN_ESGSN(1, 1)+ESGSN_HLR_1(1, 1)+HSGSN_HLR_1(1, 1)+HSGSN_RAN_3(1, 1)+RAN_HSGSN_2+HSGSN_HLR_2(1, 1)+HLR_ESGSN_1(1, 1)+ESGSN_HLR_2(1, 1)+HLR_ESGSN_2(1, 1)+ESGSN_UE_4_1+UE_ESGSN_2(1, 4)
t_HO_tot_4_2=RNC_UE+UE_ESGSN_1(1, 4)+ESGSN_HGSN_1(1, 2)+HSGSN_RAN_1(1, 2)+RAN_HSGSN_1+HSGSN_ESGSN(1, 2)+HLR_UE(1, 2)+ESGSN_HGSN_2(1, 2)+HSGSN_RAN_2(1, 2)+ESGSN_GGSN(1, 2)+GGSN_ESGSN(1, 2)+ESGSN_HLR_1(1, 2)+HSGSN_HLR_1(1, 2)+HSGSN_RAN_3(1, 2)+RAN_HSGSN_2+HSGSN_HLR_2(1, 2)+HLR_ESGSN_1(1, 2)+ESGSN_HLR_2(1, 2)+HLR_ESGSN_2(1, 2)+ESGSN_UE_4_2+UE_ESGSN_2(1, 4)
t_HO_tot_4_3=RNC_UE+UE_ESGSN_1(1, 4)+ESGSN_HGSN_1(1, 3)+HSGSN_RAN_1(1, 3)+RAN_HSGSN_1+HSGSN_ESGSN(1, 3)+HLR_UE(1, 3)+ESGSN_HGSN_2(1, 3)+HSGSN_RAN_2(1, 3)+ESGSN_GGSN(1, 3)+GGSN_ESGSN(1, 3)+ESGSN_HLR_1(1, 3)+HSGSN_HLR_1(1, 3)+HSGSN_RAN_3(1, 3)+RAN_HSGSN_2+HSGSN_HLR_2(1, 3)+HLR_ESGSN_1(1, 3)+ESGSN_HLR_2(1, 3)+HLR_ESGSN_2(1, 3)+ESGSN_UE_4_3+UE_ESGSN_2(1, 4)
t_HO_tot_4_4=RNC_UE+UE_ESGSN_1(1, 4)+ESGSN_HGSN_1(1, 4)+HSGSN_RAN_1(1, 4)+RAN_HSGSN_1+HSGSN_ESGSN(1, 4)+HLR_UE(1, 4)+ESGSN_HGSN_2(1, 4)+HSGSN_RAN_2(1, 4)+ESGSN_GGSN(1, 4)+GGSN_ESGSN(1, 4)+ESGSN_HLR_1(1, 4)+HSGSN_HLR_1(1, 4)+HSGSN_RAN_3(1, 4)+RAN_HSGSN_2+HSGSN_HLR_2(1, 4)+HLR_ESGSN_1(1, 4)+ESGSN_HLR_2(1, 4)+HLR_ESGSN_2(1, 4)+ESGSN_UE_4_4+UE_ESGSN_2(1, 4)
t_HO_tot_4_5=RNC_UE+UE_ESGSN_1(1, 4)+ESGSN_HGSN_1(1, 5)+HSGSN_RAN_1(1, 5)+RAN_HSGSN_1+HSGSN_ESGSN(1, 5)+HLR_UE(1, 5)+ESGSN_HGSN_2(1, 5)+HSGSN_RAN_2(1, 5)+ESGSN_GGSN(1, 5)+GGSN_ESGSN(1, 5)+ESGSN_HLR_1(1, 5)+HSGSN_HLR_1(1, 5)+HSGSN_RAN_3(1, 5)+RAN_HSGSN_2+HSGSN_HLR_2(1, 5)+HLR_ESGSN_1(1, 5)+ESGSN_HLR_2(1, 5)+HLR_ESGSN_2(1, 5)+ESGSN_UE_4_5+UE_ESGSN_2(1, 4)
t_HO_tot_4_6=RNC_UE+UE_ESGSN_1(1, 4)+ESGSN_HGSN_1(1, 6)+HSGSN_RAN_1(1, 6)+RAN_HSGSN_1+HSGSN_ESGSN(1, 6)+HLR_UE(1, 6)+ESGSN_HGSN_2(1, 6)+HSGSN_RAN_2(1, 6)+ESGSN_GGSN(1, 6)+GGSN_ESGSN(1, 6)+ESGSN_HLR_1(1, 6)+HSGSN_HLR_1(1, 6)+HSGSN_RAN_3(1, 6)+RAN_HSGSN_2+HSGSN_HLR_2(1, 6)+HLR_ESGSN_1(1, 6)+ESGSN_HLR_2(1, 6)+HLR_ESGSN_2(1, 6)+ESGSN_UE_4_6+UE_ESGSN_2(1, 4)
t_HO_tot_4_7=RNC_UE+UE_ESGSN_1(1, 4)+ESGSN_HGSN_1(1, 7)+HSGSN_RAN_1(1, 7)+RAN_HSGSN_1+HSGSN_ESGSN(1, 7)+HLR_UE(1, 7)+ESGSN_HGSN_2(1, 7)+HSGSN_RAN_2(1, 7)+ESGSN_GGSN(1, 7)+GGSN_ESGSN(1, 7)+ESGSN_HLR_1(1, 7)+HSGSN_HLR_1(1, 7)+HSGSN_RAN_3(1, 7)+RAN_HSGSN_2+HSGSN_HLR_2(1, 7)+HLR_ESGSN_1(1, 7)+ESGSN_HLR_2(1, 7)+HLR_ESGSN_2(1, 7)+ESGSN_UE_4_7+UE_ESGSN_2(1, 4)
t_HO_tot_4_8=RNC_UE+UE_ESGSN_1(1, 4)+ESGSN_HGSN_1(1, 8)+HSGSN_RAN_1(1, 8)+RAN_HSGSN_1+HSGSN_ESGSN(1, 8)+HLR_UE(1, 8)+ESGSN_HGSN_2(1, 8)+HSGSN_RAN_2(1, 8)+ESGSN_GGSN(1, 8)+GGSN_ESGSN(1, 8)+ESGSN_HLR_1(1, 8)+HSGSN_HLR_1(1, 8)+HSGSN_RAN_3(1, 8)+RAN_HSGSN_2+HSGSN_HLR_2(1, 8)+HLR_ESGSN_1(1, 8)+ESGSN_HLR_2(1, 8)+HLR_ESGSN_2(1, 8)+ESGSN_UE_4_8+UE_ESGSN_2(1, 4)
t_HO_tot_4_9=RNC_UE+UE_ESGSN_1(1, 4)+ESGSN_HGSN_1(1, 9)+HSGSN_RAN_1(1, 9)+RAN_HSGSN_1+HSGSN_ESGSN(1, 9)+HLR_UE(1, 9)+ESGSN_HGSN_2(1, 9)+HSGSN_RAN_2(1, 9)+ESGSN_GGSN(1, 9)+GGSN_ESGSN(1, 9)+ESGSN_HLR_1(1, 9)+HSGSN_HLR_1(1, 9)

$t_{HO_tot_5_1} = RNC_UE + UE_ESGSN_1(1,5) + ESGSN_HGSN_1(1,1) + HGSN_RAN_1(1,1)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,1) + HLR_UE(1,1) + ESGSN_HGSN_2(1,1) + HGSN_RAN_2(1,1)$
 $+ ESGSN_GGSN(1,1) + GGSN_ESGSN(1,1) + ESGSN_HLR_1(1,1) + HGSN_HLR_1(1,1)$
 $+ HGSN_RAN_3(1,1) + RAN_HGSN_2 + HGSN_HLR_2(1,1) + HLR_ESGSN_1(1,1) + ESGSN_HLR_2(1,1)$
 $+ HLR_ESGSN_2(1,1) + ESGSN_UE_5_1 + UE_ESGSN_2(1,5)$
 $t_{HO_tot_5_2} = RNC_UE + UE_ESGSN_1(1,5) + ESGSN_HGSN_1(1,2) + HGSN_RAN_1(1,2)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,2) + HLR_UE(1,2) + ESGSN_HGSN_2(1,2) + HGSN_RAN_2(1,2)$
 $+ ESGSN_GGSN(1,2) + GGSN_ESGSN(1,2) + ESGSN_HLR_1(1,2) + HGSN_HLR_1(1,2)$
 $+ HGSN_RAN_3(1,2) + RAN_HGSN_2 + HGSN_HLR_2(1,2) + HLR_ESGSN_1(1,2) + ESGSN_HLR_2(1,2)$
 $+ HLR_ESGSN_2(1,2) + ESGSN_UE_5_2 + UE_ESGSN_2(1,5)$
 $t_{HO_tot_5_3} = RNC_UE + UE_ESGSN_1(1,5) + ESGSN_HGSN_1(1,3) + HGSN_RAN_1(1,3)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,3) + HLR_UE(1,3) + ESGSN_HGSN_2(1,3) + HGSN_RAN_2(1,3)$
 $+ ESGSN_GGSN(1,3) + GGSN_ESGSN(1,3) + ESGSN_HLR_1(1,3) + HGSN_HLR_1(1,3)$
 $+ HGSN_RAN_3(1,3) + RAN_HGSN_2 + HGSN_HLR_2(1,3) + HLR_ESGSN_1(1,3) + ESGSN_HLR_2(1,3)$
 $+ HLR_ESGSN_2(1,3) + ESGSN_UE_5_3 + UE_ESGSN_2(1,5)$
 $t_{HO_tot_5_4} = RNC_UE + UE_ESGSN_1(1,5) + ESGSN_HGSN_1(1,4) + HGSN_RAN_1(1,4)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,4) + HLR_UE(1,4) + ESGSN_HGSN_2(1,4) + HGSN_RAN_2(1,4)$
 $+ ESGSN_GGSN(1,4) + GGSN_ESGSN(1,4) + ESGSN_HLR_1(1,4) + HGSN_HLR_1(1,4)$
 $+ HGSN_RAN_3(1,4) + RAN_HGSN_2 + HGSN_HLR_2(1,4) + HLR_ESGSN_1(1,4) + ESGSN_HLR_2(1,4)$
 $+ HLR_ESGSN_2(1,4) + ESGSN_UE_5_4 + UE_ESGSN_2(1,5)$
 $t_{HO_tot_5_5} = RNC_UE + UE_ESGSN_1(1,5) + ESGSN_HGSN_1(1,5) + HGSN_RAN_1(1,5)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,5) + HLR_UE(1,5) + ESGSN_HGSN_2(1,5) + HGSN_RAN_2(1,5)$
 $+ ESGSN_GGSN(1,5) + GGSN_ESGSN(1,5) + ESGSN_HLR_1(1,5) + HGSN_HLR_1(1,5)$
 $+ HGSN_RAN_3(1,5) + RAN_HGSN_2 + HGSN_HLR_2(1,5) + HLR_ESGSN_1(1,5) + ESGSN_HLR_2(1,5)$
 $+ HLR_ESGSN_2(1,5) + ESGSN_UE_5_5 + UE_ESGSN_2(1,5)$
 $t_{HO_tot_5_6} = RNC_UE + UE_ESGSN_1(1,5) + ESGSN_HGSN_1(1,6) + HGSN_RAN_1(1,6)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,6) + HLR_UE(1,6) + ESGSN_HGSN_2(1,6) + HGSN_RAN_2(1,6)$
 $+ ESGSN_GGSN(1,6) + GGSN_ESGSN(1,6) + ESGSN_HLR_1(1,6) + HGSN_HLR_1(1,6)$
 $+ HGSN_RAN_3(1,6) + RAN_HGSN_2 + HGSN_HLR_2(1,6) + HLR_ESGSN_1(1,6) + ESGSN_HLR_2(1,6)$
 $+ HLR_ESGSN_2(1,6) + ESGSN_UE_5_6 + UE_ESGSN_2(1,5)$
 $t_{HO_tot_5_7} = RNC_UE + UE_ESGSN_1(1,5) + ESGSN_HGSN_1(1,7) + HGSN_RAN_1(1,7)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,7) + HLR_UE(1,7) + ESGSN_HGSN_2(1,7) + HGSN_RAN_2(1,7)$
 $+ ESGSN_GGSN(1,7) + GGSN_ESGSN(1,7) + ESGSN_HLR_1(1,7) + HGSN_HLR_1(1,7)$
 $+ HGSN_RAN_3(1,7) + RAN_HGSN_2 + HGSN_HLR_2(1,7) + HLR_ESGSN_1(1,7) + ESGSN_HLR_2(1,7)$
 $+ HLR_ESGSN_2(1,7) + ESGSN_UE_5_7 + UE_ESGSN_2(1,5)$
 $t_{HO_tot_5_8} = RNC_UE + UE_ESGSN_1(1,5) + ESGSN_HGSN_1(1,8) + HGSN_RAN_1(1,8)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,8) + HLR_UE(1,8) + ESGSN_HGSN_2(1,8) + HGSN_RAN_2(1,8)$
 $+ ESGSN_GGSN(1,8) + GGSN_ESGSN(1,8) + ESGSN_HLR_1(1,8) + HGSN_HLR_1(1,8)$
 $+ HGSN_RAN_3(1,8) + RAN_HGSN_2 + HGSN_HLR_2(1,8) + HLR_ESGSN_1(1,8) + EGSN_HLR_2(1,8)$
 $+ HLR_ESGSN_2(1,8) + ESGSN_UE_5_8 + UE_ESGSN_2(1,5)$
 $t_{HO_tot_5_9} = RNC_UE + UE_ESGSN_1(1,5) + ESGSN_HGSN_1(1,9) + HGSN_RAN_1(1,9)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,9) + HLR_UE(1,9) + ESGSN_HGSN_2(1,9) + HGSN_RAN_2(1,9)$
 $+ ESGSN_GGSN(1,9) + GGSN_ESGSN(1,9) + ESGSN_HLR_1(1,9) + HGSN_HLR_1(1,9)$
 $+ HGSN_RAN_3(1,9) + RAN_HGSN_2 + HGSN_HLR_2(1,9) + HLR_ESGSN_1(1,9) + EGSN_HLR_2(1,9)$
 $+ HLR_ESGSN_2(1,9) + ESGSN_UE_5_9 + UE_ESGSN_2(1,5)$

 $t_{HO_tot_6_1} = RNC_UE + UE_ESGSN_1(1,6) + ESGSN_HGSN_1(1,1) + HGSN_RAN_1(1,1)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,1) + HLR_UE(1,1) + ESGSN_HGSN_2(1,1) + HGSN_RAN_2(1,1)$
 $+ ESGSN_GGSN(1,1) + GGSN_ESGSN(1,1) + ESGSN_HLR_1(1,1) + HGSN_HLR_1(1,1)$
 $+ HGSN_RAN_3(1,1) + RAN_HGSN_2 + HGSN_HLR_2(1,1) + HLR_ESGSN_1(1,1) + EGSN_HLR_2(1,1)$
 $+ HLR_ESGSN_2(1,1) + ESGSN_UE_6_1 + UE_ESGSN_2(1,6)$
 $t_{HO_tot_6_2} = RNC_UE + UE_ESGSN_1(1,6) + ESGSN_HGSN_1(1,2) + HGSN_RAN_1(1,2)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,2) + HLR_UE(1,2) + ESGSN_HGSN_2(1,2) + HGSN_RAN_2(1,2)$
 $+ ESGSN_GGSN(1,2) + GGSN_ESGSN(1,2) + ESGSN_HLR_1(1,2) + HGSN_HLR_1(1,2)$
 $+ HGSN_RAN_3(1,2) + RAN_HGSN_2 + HGSN_HLR_2(1,2) + HLR_ESGSN_1(1,2) + EGSN_HLR_2(1,2)$
 $+ HLR_ESGSN_2(1,2) + ESGSN_UE_6_2 + UE_ESGSN_2(1,6)$
 $t_{HO_tot_6_3} = RNC_UE + UE_ESGSN_1(1,6) + ESGSN_HGSN_1(1,3) + HGSN_RAN_1(1,3)$
 $+ RAN_HGSN_1 + HGSN_ESGSN(1,3) + HLR_UE(1,3) + ESGSN_HGSN_2(1,3) + HGSN_RAN_2(1,3)$

N_2(1,3)+ESGSN_GGSN(1,3)+GGSN_ESGSN(1,3)+ESGSN_HLR_1(1,3)+HSGSN_HLR_1(1,3)+HSGSN_RAN_3(1,3)+RAN_HSGSN_2+HSGSN_HLR_2(1,3)+HLR_ESGSN_1(1,3)+ESGSN_HLR_2(1,3)+HLR_ESGSN_2(1,3)+ESGSN_UE_6_3+UE_ESGSN_2(1,6)
 t_HO_tot_6_4=RNC_UE+UE_ESGSN_1(1,6)+ESGSN_HGSN_1(1,4)+HGSN_RAN_1(1,4)+RAN_HSGSN_1+HGSN_ESGSN(1,4)+HLR_UE(1,4)+ESGSN_HGSN_2(1,4)+HGSN_RAN_2(1,4)+ESGSN_GGSN(1,4)+GGSN_ESGSN(1,4)+ESGSN_HLR_1(1,4)+HGSN_HLR_1(1,4)+HGSN_RAN_3(1,4)+RAN_HSGSN_2+HGSN_HLR_2(1,4)+HLR_ESGSN_1(1,4)+ESGSN_HLR_2(1,4)+HLR_ESGSN_2(1,4)+ESGSN_UE_6_4+UE_ESGSN_2(1,6)
 t_HO_tot_6_5=RNC_UE+UE_ESGSN_1(1,6)+ESGSN_HGSN_1(1,5)+HGSN_RAN_1(1,5)+RAN_HSGSN_1+HGSN_ESGSN(1,5)+HLR_UE(1,5)+ESGSN_HGSN_2(1,5)+HGSN_RAN_2(1,5)+ESGSN_GGSN(1,5)+GGSN_ESGSN(1,5)+ESGSN_HLR_1(1,5)+HGSN_HLR_1(1,5)+HGSN_RAN_3(1,5)+RAN_HSGSN_2+HGSN_HLR_2(1,5)+HLR_ESGSN_1(1,5)+ESGSN_HLR_2(1,5)+HLR_ESGSN_2(1,5)+ESGSN_UE_6_5+UE_ESGSN_2(1,6)
 t_HO_tot_6_6=RNC_UE+UE_ESGSN_1(1,6)+ESGSN_HGSN_1(1,6)+HGSN_RAN_1(1,6)+RAN_HSGSN_1+HGSN_ESGSN(1,6)+HLR_UE(1,6)+ESGSN_HGSN_2(1,6)+HGSN_RAN_2(1,6)+ESGSN_GGSN(1,6)+GGSN_ESGSN(1,6)+ESGSN_HLR_1(1,6)+HGSN_HLR_1(1,6)+HGSN_RAN_3(1,6)+RAN_HSGSN_2+HGSN_HLR_2(1,6)+HLR_ESGSN_1(1,6)+ESGSN_HLR_2(1,6)+HLR_ESGSN_2(1,6)+ESGSN_UE_6_6+UE_ESGSN_2(1,6)
 t_HO_tot_6_7=RNC_UE+UE_ESGSN_1(1,6)+ESGSN_HGSN_1(1,7)+HGSN_RAN_1(1,7)+RAN_HSGSN_1+HGSN_ESGSN(1,7)+HLR_UE(1,7)+ESGSN_HGSN_2(1,7)+HGSN_RAN_2(1,7)+ESGSN_GGSN(1,7)+GGSN_ESGSN(1,7)+ESGSN_HLR_1(1,7)+HGSN_HLR_1(1,7)+HGSN_RAN_3(1,7)+RAN_HSGSN_2+HGSN_HLR_2(1,7)+HLR_ESGSN_1(1,7)+ESGSN_HLR_2(1,7)+HLR_ESGSN_2(1,7)+ESGSN_UE_6_7+UE_ESGSN_2(1,6)
 t_HO_tot_6_8=RNC_UE+UE_ESGSN_1(1,6)+ESGSN_HGSN_1(1,8)+HGSN_RAN_1(1,8)+RAN_HSGSN_1+HGSN_ESGSN(1,8)+HLR_UE(1,8)+ESGSN_HGSN_2(1,8)+HGSN_RAN_2(1,8)+ESGSN_GGSN(1,8)+GGSN_ESGSN(1,8)+ESGSN_HLR_1(1,8)+HGSN_HLR_1(1,8)+HGSN_RAN_3(1,8)+RAN_HSGSN_2+HGSN_HLR_2(1,8)+HLR_ESGSN_1(1,8)+ESGSN_HLR_2(1,8)+HLR_ESGSN_2(1,8)+ESGSN_UE_6_8+UE_ESGSN_2(1,6)
 t_HO_tot_6_9=RNC_UE+UE_ESGSN_1(1,6)+ESGSN_HGSN_1(1,9)+HGSN_RAN_1(1,9)+RAN_HSGSN_1+HGSN_ESGSN(1,9)+HLR_UE(1,9)+ESGSN_HGSN_2(1,9)+HGSN_RAN_2(1,9)+ESGSN_GGSN(1,9)+GGSN_ESGSN(1,9)+ESGSN_HLR_1(1,9)+HGSN_HLR_1(1,9)+HGSN_RAN_3(1,9)+RAN_HSGSN_2+HGSN_HLR_2(1,9)+HLR_ESGSN_1(1,9)+ESGSN_HLR_2(1,9)+HLR_ESGSN_2(1,9)+ESGSN_UE_6_9+UE_ESGSN_2(1,6)

t_HO_tot_7_1=RNC_UE+UE_ESGSN_1(1,7)+ESGSN_HGSN_1(1,1)+HGSN_RAN_1(1,1)+RAN_HSGSN_1+HGSN_ESGSN(1,1)+HLR_UE(1,1)+ESGSN_HGSN_2(1,1)+HGSN_RAN_2(1,1)+ESGSN_GGSN(1,1)+GGSN_ESGSN(1,1)+ESGSN_HLR_1(1,1)+HGSN_HLR_1(1,1)+HGSN_RAN_3(1,1)+RAN_HSGSN_2+HGSN_HLR_2(1,1)+HLR_ESGSN_1(1,1)+ESGSN_HLR_2(1,1)+HLR_ESGSN_2(1,1)+ESGSN_UE_7_1+UE_ESGSN_2(1,7)
 t_HO_tot_7_2=RNC_UE+UE_ESGSN_1(1,7)+ESGSN_HGSN_1(1,2)+HGSN_RAN_1(1,2)+RAN_HSGSN_1+HGSN_ESGSN(1,2)+HLR_UE(1,2)+ESGSN_HGSN_2(1,2)+HGSN_RAN_2(1,2)+ESGSN_GGSN(1,2)+GGSN_ESGSN(1,2)+ESGSN_HLR_1(1,2)+HGSN_HLR_1(1,2)+HGSN_RAN_3(1,2)+RAN_HSGSN_2+HGSN_HLR_2(1,2)+HLR_ESGSN_1(1,2)+ESGSN_HLR_2(1,2)+HLR_ESGSN_2(1,2)+ESGSN_UE_7_2+UE_ESGSN_2(1,7)
 t_HO_tot_7_3=RNC_UE+UE_ESGSN_1(1,7)+ESGSN_HGSN_1(1,3)+HGSN_RAN_1(1,3)+RAN_HSGSN_1+HGSN_ESGSN(1,3)+HLR_UE(1,3)+ESGSN_HGSN_2(1,3)+HGSN_RAN_2(1,3)+ESGSN_GGSN(1,3)+GGSN_ESGSN(1,3)+ESGSN_HLR_1(1,3)+HGSN_HLR_1(1,3)+HGSN_RAN_3(1,3)+RAN_HSGSN_2+HGSN_HLR_2(1,3)+HLR_ESGSN_1(1,3)+ESGSN_HLR_2(1,3)+HLR_ESGSN_2(1,3)+ESGSN_UE_7_3+UE_ESGSN_2(1,7)
 t_HO_tot_7_4=RNC_UE+UE_ESGSN_1(1,7)+ESGSN_HGSN_1(1,4)+HGSN_RAN_1(1,4)+RAN_HSGSN_1+HGSN_ESGSN(1,4)+HLR_UE(1,4)+ESGSN_HGSN_2(1,4)+HGSN_RAN_2(1,4)+ESGSN_GGSN(1,4)+GGSN_ESGSN(1,4)+ESGSN_HLR_1(1,4)+HGSN_HLR_1(1,4)+HGSN_RAN_3(1,4)+RAN_HSGSN_2+HGSN_HLR_2(1,4)+HLR_ESGSN_1(1,4)+ESGSN_HLR_2(1,4)+HLR_ESGSN_2(1,4)+ESGSN_UE_7_4+UE_ESGSN_2(1,7)
 t_HO_tot_7_5=RNC_UE+UE_ESGSN_1(1,7)+ESGSN_HGSN_1(1,5)+HGSN_RAN_1(1,5)+RAN_HSGSN_1+HGSN_ESGSN(1,5)+HLR_UE(1,5)+ESGSN_HGSN_2(1,5)+HGSN_RAN_2(1,5)+ESGSN_GGSN(1,5)+GGSN_ESGSN(1,5)+ESGSN_HLR_1(1,5)+HGSN_HLR_1(1,5)+HGSN_RAN_3(1,5)+RAN_HSGSN_2+HGSN_HLR_2(1,5)+HLR_ESGSN_1(1,5)+ESGSN_HLR_2(1,5)+HLR_ESGSN_2(1,5)+ESGSN_UE_7_5+UE_ESGSN_2(1,7)
 t_HO_tot_7_6=RNC_UE+UE_ESGSN_1(1,7)+ESGSN_HGSN_1(1,6)+HGSN_RAN_1(1,6)+RAN_HSGSN_1+HGSN_ESGSN(1,6)+HLR_UE(1,6)+ESGSN_HGSN_2(1,6)+HGSN_RAN_2(1,6)+ESGSN_GGSN(1,6)+GGSN_ESGSN(1,6)+ESGSN_HLR_1(1,6)+HGSN_HLR_1(1,6)+HGSN_RAN_3(1,6)+RAN_HSGSN_2+HGSN_HLR_2(1,6)+HLR_ESGSN_1(1,6)+ESGSN_HLR_2(1,6)+HLR_ESGSN_2(1,6)+ESGSN_UE_7_6+UE_ESGSN_2(1,6)



N_2(1,6)+ESGSN_GGSN(1,6)+GGSN_ESGSN(1,6)+ESGSN_HLR_1(1,6)+HSGSN_HLR_1(1,6)+HSGSN_RAN_3(1,6)+RAN_HSGSN_2+HSGSN_HLR_2(1,6)+HLR_ESGSN_1(1,6)+ESGSN_HLR_2(1,6)+HLR_ESGSN_2(1,6)+ESGSN_UE_7_6+UE_ESGSN_2(1,7)
t_HO_tot_7_7=RNC_UE+UE_ESGSN_1(1,7)+ESGSN_HGSN_1(1,7)+HGSN_RAN_1(1,7)+RAN_HSGSN_1+HGSN_ESGSN(1,7)+HLR_UE(1,7)+ESGSN_HGSN_2(1,7)+HGSN_RAN_2(1,7)+ESGSN_GGSN(1,7)+GGSN_ESGSN(1,7)+ESGSN_HLR_1(1,7)+HGSN_HLR_1(1,7)+HGSN_RAN_3(1,7)+RAN_HSGSN_2+HSGSN_HLR_2(1,7)+HLR_ESGSN_1(1,7)+ESGSN_HLR_2(1,7)+HLR_ESGSN_2(1,7)+ESGSN_UE_7_7+UE_ESGSN_2(1,7)
t_HO_tot_7_8=RNC_UE+UE_ESGSN_1(1,7)+ESGSN_HGSN_1(1,8)+HGSN_RAN_1(1,8)+RAN_HSGSN_1+HGSN_ESGSN(1,8)+HLR_UE(1,8)+ESGSN_HGSN_2(1,8)+HGSN_RAN_2(1,8)+ESGSN_GGSN(1,8)+GGSN_ESGSN(1,8)+ESGSN_HLR_1(1,8)+HGSN_HLR_1(1,8)+HGSN_RAN_3(1,8)+RAN_HSGSN_2+HSGSN_HLR_2(1,8)+HLR_ESGSN_1(1,8)+ESGSN_HLR_2(1,8)+HLR_ESGSN_2(1,8)+ESGSN_UE_7_8+UE_ESGSN_2(1,7)
t_HO_tot_7_9=RNC_UE+UE_ESGSN_1(1,7)+ESGSN_HGSN_1(1,9)+HGSN_RAN_1(1,9)+RAN_HSGSN_1+HGSN_ESGSN(1,9)+HLR_UE(1,9)+ESGSN_HGSN_2(1,9)+HGSN_RAN_2(1,9)+ESGSN_GGSN(1,9)+GGSN_ESGSN(1,9)+ESGSN_HLR_1(1,9)+HGSN_HLR_1(1,9)+HGSN_RAN_3(1,9)+RAN_HSGSN_2+HSGSN_HLR_2(1,9)+HLR_ESGSN_1(1,9)+ESGSN_HLR_2(1,9)+HLR_ESGSN_2(1,9)+ESGSN_UE_7_9+UE_ESGSN_2(1,7)
t_HO_tot_8_1=RNC_UE+UE_ESGSN_1(1,8)+ESGSN_HGSN_1(1,1)+HGSN_RAN_1(1,1)+RAN_HSGSN_1+HGSN_ESGSN(1,1)+HLR_UE(1,1)+ESGSN_HGSN_2(1,1)+HGSN_RAN_2(1,1)+ESGSN_GGSN(1,1)+GGSN_ESGSN(1,1)+ESGSN_HLR_1(1,1)+HGSN_HLR_1(1,1)+HGSN_RAN_3(1,1)+RAN_HSGSN_2+HSGSN_HLR_2(1,1)+HLR_ESGSN_1(1,1)+ESGSN_HLR_2(1,1)+HLR_ESGSN_2(1,1)+ESGSN_UE_8_1+UE_ESGSN_2(1,8)
t_HO_tot_8_2=RNC_UE+UE_ESGSN_1(1,8)+ESGSN_HGSN_1(1,2)+HGSN_RAN_1(1,2)+RAN_HSGSN_1+HGSN_ESGSN(1,2)+HLR_UE(1,2)+ESGSN_HGSN_2(1,2)+HGSN_RAN_2(1,2)+ESGSN_GGSN(1,2)+GGSN_ESGSN(1,2)+ESGSN_HLR_1(1,2)+HGSN_HLR_1(1,2)+HGSN_RAN_3(1,2)+RAN_HSGSN_2+HSGSN_HLR_2(1,2)+HLR_ESGSN_1(1,2)+ESGSN_HLR_2(1,2)+HLR_ESGSN_2(1,2)+ESGSN_UE_8_2+UE_ESGSN_2(1,8)
t_HO_tot_8_3=RNC_UE+UE_ESGSN_1(1,8)+ESGSN_HGSN_1(1,3)+HGSN_RAN_1(1,3)+RAN_HSGSN_1+HGSN_ESGSN(1,3)+HLR_UE(1,3)+ESGSN_HGSN_2(1,3)+HGSN_RAN_2(1,3)+ESGSN_GGSN(1,3)+GGSN_ESGSN(1,3)+ESGSN_HLR_1(1,3)+HGSN_HLR_1(1,3)+HGSN_RAN_3(1,3)+RAN_HSGSN_2+HSGSN_HLR_2(1,3)+HLR_ESGSN_1(1,3)+ESGSN_HLR_2(1,3)+HLR_ESGSN_2(1,3)+ESGSN_UE_8_3+UE_ESGSN_2(1,8)
t_HO_tot_8_4=RNC_UE+UE_ESGSN_1(1,8)+ESGSN_HGSN_1(1,4)+HGSN_RAN_1(1,4)+RAN_HSGSN_1+HGSN_ESGSN(1,4)+HLR_UE(1,4)+ESGSN_HGSN_2(1,4)+HGSN_RAN_2(1,4)+ESGSN_GGSN(1,4)+GGSN_ESGSN(1,4)+ESGSN_HLR_1(1,4)+HGSN_HLR_1(1,4)+HGSN_RAN_3(1,4)+RAN_HSGSN_2+HSGSN_HLR_2(1,4)+HLR_ESGSN_1(1,4)+ESGSN_HLR_2(1,4)+HLR_ESGSN_2(1,4)+ESGSN_UE_8_4+UE_ESGSN_2(1,8)
t_HO_tot_8_5=RNC_UE+UE_ESGSN_1(1,8)+ESGSN_HGSN_1(1,5)+HGSN_RAN_1(1,5)+RAN_HSGSN_1+HGSN_ESGSN(1,5)+HLR_UE(1,5)+ESGSN_HGSN_2(1,5)+HGSN_RAN_2(1,5)+ESGSN_GGSN(1,5)+GGSN_ESGSN(1,5)+ESGSN_HLR_1(1,5)+HGSN_HLR_1(1,5)+HGSN_RAN_3(1,5)+RAN_HSGSN_2+HSGSN_HLR_2(1,5)+HLR_ESGSN_1(1,5)+ESGSN_HLR_2(1,5)+HLR_ESGSN_2(1,5)+ESGSN_UE_8_5+UE_ESGSN_2(1,8)
t_HO_tot_8_6=RNC_UE+UE_ESGSN_1(1,8)+ESGSN_HGSN_1(1,6)+HGSN_RAN_1(1,6)+RAN_HSGSN_1+HGSN_ESGSN(1,6)+HLR_UE(1,6)+ESGSN_HGSN_2(1,6)+HGSN_RAN_2(1,6)+ESGSN_GGSN(1,6)+GGSN_ESGSN(1,6)+ESGSN_HLR_1(1,6)+HGSN_HLR_1(1,6)+HGSN_RAN_3(1,6)+RAN_HSGSN_2+HSGSN_HLR_2(1,6)+HLR_ESGSN_1(1,6)+ESGSN_HLR_2(1,6)+HLR_ESGSN_2(1,6)+ESGSN_UE_8_6+UE_ESGSN_2(1,8)
t_HO_tot_8_7=RNC_UE+UE_ESGSN_1(1,8)+ESGSN_HGSN_1(1,7)+HGSN_RAN_1(1,7)+RAN_HSGSN_1+HGSN_ESGSN(1,7)+HLR_UE(1,7)+ESGSN_HGSN_2(1,7)+HGSN_RAN_2(1,7)+ESGSN_GGSN(1,7)+GGSN_ESGSN(1,7)+ESGSN_HLR_1(1,7)+HGSN_HLR_1(1,7)+HGSN_RAN_3(1,7)+RAN_HSGSN_2+HSGSN_HLR_2(1,7)+HLR_ESGSN_1(1,7)+ESGSN_HLR_2(1,7)+HLR_ESGSN_2(1,7)+ESGSN_UE_8_7+UE_ESGSN_2(1,8)
t_HO_tot_8_8=RNC_UE+UE_ESGSN_1(1,8)+ESGSN_HGSN_1(1,8)+HGSN_RAN_1(1,8)+RAN_HSGSN_1+HGSN_ESGSN(1,8)+HLR_UE(1,8)+ESGSN_HGSN_2(1,8)+HGSN_RAN_2(1,8)+ESGSN_GGSN(1,8)+GGSN_ESGSN(1,8)+ESGSN_HLR_1(1,8)+HGSN_HLR_1(1,8)+HGSN_RAN_3(1,8)+RAN_HSGSN_2+HSGSN_HLR_2(1,8)+HLR_ESGSN_1(1,8)+ESGSN_HLR_2(1,8)+HLR_ESGSN_2(1,8)+ESGSN_UE_8_8+UE_ESGSN_2(1,8)
t_HO_tot_8_9=RNC_UE+UE_ESGSN_1(1,8)+ESGSN_HGSN_1(1,9)+HGSN_RAN_1(1,9)+RAN_HSGSN_1+HGSN_ESGSN(1,9)+HLR_UE(1,9)+ESGSN_HGSN_2(1,9)+HGSN_RAN_2(1,9)+ESGSN_GGSN(1,9)+GGSN_ESGSN(1,9)+ESGSN_HLR_1(1,9)+HGSN_HLR_1(1,9)+HGSN_RAN_3(1,9)+RAN_HSGSN_2+HSGSN_HLR_2(1,9)+HLR_ESGSN_1(1,9)+ESGSN_HLR_2(1,9)+HLR_ESGSN_2(1,9)+ESGSN_UE_8_9+UE_ESGSN_2(1,8)

N_2(1,9)+ESGSN_GGSN(1,9)+GGSN_ESGSN(1,9)+ESGSN_HLR_1(1,9)+HGSN_HLR_1(1,9)+HGSN_RAN_3(1,9)+RAN_HGSN_2+HGSN_HLR_2(1,9)+HLR_ESGSN_1(1,9)+ESGSN_HLR_2(1,9)+HLR_ESGSN_2(1,9)+ESGSN UE_8_9+UE_ESGSN_2(1,8)

t HO_tot_9_1=RNC_UE+UE_ESGSN_1(1,9)+ESGSN_HGSN_1(1,1)+HGSN_RAN_1(1,1)+RAN_HGSN_1+HGSN_ESGSN(1,1)+HLR_UE(1,1)+ESGSN_HGSN_2(1,1)+HGSN_RAN_2(1,1)+ESGSN_GGSN(1,1)+GGSN_ESGSN(1,1)+ESGSN_HLR_1(1,1)+HGSN_RAN_3(1,1)+RAN_HGSN_2+HGSN_HLR_2(1,1)+HLR_ESGSN_1(1,1)+ESGSN_HLR_2(1,1)+HLR_ESGSN_2(1,1)+ESGSN UE_9_1+UE_ESGSN_2(1,9)
t HO_tot_9_2=RNC_UE+UE_ESGSN_1(1,9)+ESGSN_HGSN_1(1,2)+HGSN_RAN_1(1,2)+RAN_HGSN_1+HGSN_ESGSN(1,2)+HLR_UE(1,2)+ESGSN_HGSN_2(1,2)+HGSN_RAN_2(1,2)+ESGSN_GGSN(1,2)+GGSN_ESGSN(1,2)+ESGSN_HLR_1(1,2)+HGSN_HLR_1(1,2)+HGSN_RAN_3(1,2)+RAN_HGSN_2+HGSN_HLR_2(1,2)+HLR_ESGSN_1(1,2)+ESGSN_HLR_2(1,2)+HLR_ESGSN_2(1,2)+ESGSN UE_9_2+UE_ESGSN_2(1,9)
t HO_tot_9_3=RNC_UE+UE_ESGSN_1(1,9)+ESGSN_HGSN_1(1,3)+HGSN_RAN_1(1,3)+RAN_HGSN_1+HGSN_ESGSN(1,3)+HLR_UE(1,3)+ESGSN_HGSN_2(1,3)+HGSN_RAN_2(1,3)+ESGSN_GGSN(1,3)+GGSN_ESGSN(1,3)+ESGSN_HLR_1(1,3)+HGSN_HLR_1(1,3)+HGSN_RAN_3(1,3)+RAN_HGSN_2+HGSN_HLR_2(1,3)+HLR_ESGSN_1(1,3)+ESGSN_HLR_2(1,3)+HLR_ESGSN_2(1,3)+ESGSN UE_9_3+UE_ESGSN_2(1,9)
t HO_tot_9_4=RNC_UE+UE_ESGSN_1(1,9)+ESGSN_HGSN_1(1,4)+HGSN_RAN_1(1,4)+RAN_HGSN_1+HGSN_ESGSN(1,4)+HLR_UE(1,4)+ESGSN_HGSN_2(1,4)+HGSN_RAN_2(1,4)+ESGSN_GGSN(1,4)+GGSN_ESGSN(1,4)+ESGSN_HLR_1(1,4)+HGSN_HLR_1(1,4)+HGSN_RAN_3(1,4)+RAN_HGSN_2+HGSN_HLR_2(1,4)+HLR_ESGSN_1(1,4)+ESGSN_HLR_2(1,4)+HLR_ESGSN_2(1,4)+ESGSN UE_9_4+UE_ESGSN_2(1,9)
t HO_tot_9_5=RNC_UE+UE_ESGSN_1(1,9)+ESGSN_HGSN_1(1,5)+HGSN_RAN_1(1,5)+RAN_HGSN_1+HGSN_ESGSN(1,5)+HLR_UE(1,5)+ESGSN_HGSN_2(1,5)+HGSN_RAN_2(1,5)+ESGSN_GGSN(1,5)+GGSN_ESGSN(1,5)+ESGSN_HLR_1(1,5)+HGSN_HLR_1(1,5)+HGSN_RAN_3(1,5)+RAN_HGSN_2+HGSN_HLR_2(1,5)+HLR_ESGSN_1(1,5)+ESGSN_HLR_2(1,5)+HLR_ESGSN_2(1,5)+ESGSN UE_9_5+UE_ESGSN_2(1,9)
t HO_tot_9_6=RNC_UE+UE_ESGSN_1(1,9)+ESGSN_HGSN_1(1,6)+HGSN_RAN_1(1,6)+RAN_HGSN_1+HGSN_ESGSN(1,6)+HLR_UE(1,6)+ESGSN_HGSN_2(1,6)+HGSN_RAN_2(1,6)+ESGSN_GGSN(1,6)+GGSN_ESGSN(1,6)+ESGSN_HLR_1(1,6)+HGSN_HLR_1(1,6)+HGSN_RAN_3(1,6)+RAN_HGSN_2+HGSN_HLR_2(1,6)+HLR_ESGSN_1(1,6)+ESGSN_HLR_2(1,6)+HLR_ESGSN_2(1,6)+ESGSN UE_9_6+UE_ESGSN_2(1,9)
t HO_tot_9_7=RNC_UE+UE_ESGSN_1(1,9)+ESGSN_HGSN_1(1,7)+HGSN_RAN_1(1,7)+RAN_HGSN_1+HGSN_ESGSN(1,7)+HLR_UE(1,7)+ESGSN_HGSN_2(1,7)+HGSN_RAN_2(1,7)+ESGSN_GGSN(1,7)+GGSN_ESGSN(1,7)+ESGSN_HLR_1(1,7)+HGSN_HLR_1(1,7)+HGSN_RAN_3(1,7)+RAN_HGSN_2+HGSN_HLR_2(1,7)+HLR_ESGSN_1(1,7)+ESGSN_HLR_2(1,7)+HLR_ESGSN_2(1,7)+ESGSN UE_9_7+UE_ESGSN_2(1,9)
t HO_tot_9_8=RNC_UE+UE_ESGSN_1(1,9)+ESGSN_HGSN_1(1,8)+HGSN_RAN_1(1,8)+RAN_HGSN_1+HGSN_ESGSN(1,8)+HLR_UE(1,8)+ESGSN_HGSN_2(1,8)+HGSN_RAN_2(1,8)+ESGSN_GGSN(1,8)+GGSN_ESGSN(1,8)+ESGSN_HLR_1(1,8)+HGSN_HLR_1(1,8)+HGSN_RAN_3(1,8)+RAN_HGSN_2+HGSN_HLR_2(1,8)+HLR_ESGSN_1(1,8)+ESGSN_HLR_2(1,8)+HLR_ESGSN_2(1,8)+ESGSN UE_9_8+UE_ESGSN_2(1,9)
t HO_tot_9_9=RNC_UE+UE_ESGSN_1(1,9)+ESGSN_HGSN_1(1,9)+HGSN_RAN_1(1,9)+RAN_HGSN_1+HGSN_ESGSN(1,9)+HLR_UE(1,9)+ESGSN_HGSN_2(1,9)+HGSN_RAN_2(1,9)+ESGSN_GGSN(1,9)+GGSN_ESGSN(1,9)+ESGSN_HLR_1(1,9)+HGSN_HLR_1(1,9)+HGSN_RAN_3(1,9)+RAN_HGSN_2+HGSN_HLR_2(1,9)+HLR_ESGSN_1(1,9)+ESGSN_HLR_2(1,9)+HLR_ESGSN_2(1,9)+ESGSN UE_9_9+UE_ESGSN_2(1,9)

t HO_tot_1=[t HO_tot_1_1 t HO_tot_1_2 t HO_tot_1_3 t HO_tot_1_4 t HO_tot_1_5 t HO_tot_1_6 t HO_tot_1_7 t HO_tot_1_8 t HO_tot_1_9]
t HO_tot_2=[t HO_tot_2_1 t HO_tot_2_2 t HO_tot_2_3 t HO_tot_2_4]
t HO_tot_2_5 t HO_tot_2_6 t HO_tot_2_7 t HO_tot_2_8 t HO_tot_2_9]
t HO_tot_3=[t HO_tot_3_1 t HO_tot_3_2 t HO_tot_3_3 t HO_tot_3_4]
t HO_tot_3_5 t HO_tot_3_6 t HO_tot_3_7 t HO_tot_3_8 t HO_tot_3_9]
t HO_tot_4=[t HO_tot_4_1 t HO_tot_4_2 t HO_tot_4_3 t HO_tot_4_4]
t HO_tot_4_5 t HO_tot_4_6 t HO_tot_4_7 t HO_tot_4_8 t HO_tot_4_9]
t HO_tot_5=[t HO_tot_5_1 t HO_tot_5_2 t HO_tot_5_3 t HO_tot_5_4]
t HO_tot_5_5 t HO_tot_5_6 t HO_tot_5_7 t HO_tot_5_8 t HO_tot_5_9]

```
t_HO_tot_6=[t_HO_tot_6_1 t_HO_tot_6_2 t_HO_tot_6_3 t_HO_tot_6_4  
t_HO_tot_6_5 t_HO_tot_6_6 t_HO_tot_6_7 t_HO_tot_6_8 t_HO_tot_6_9]  
t_HO_tot_7=[t_HO_tot_7_1 t_HO_tot_7_2 t_HO_tot_7_3 t_HO_tot_7_4  
t_HO_tot_7_5 t_HO_tot_7_6 t_HO_tot_7_7 t_HO_tot_7_8 t_HO_tot_7_9]  
t_HO_tot_8=[t_HO_tot_8_1 t_HO_tot_8_2 t_HO_tot_8_3 t_HO_tot_8_4  
t_HO_tot_8_5 t_HO_tot_8_6 t_HO_tot_8_7 t_HO_tot_8_8 t_HO_tot_8_9]  
t_HO_tot_9=[t_HO_tot_9_1 t_HO_tot_9_2 t_HO_tot_9_3 t_HO_tot_9_4  
t_HO_tot_9_5 t_HO_tot_9_6 t_HO_tot_9_7 t_HO_tot_9_8 t_HO_tot_9_9]  
  
plot(p,t_HO_tot_1,'-^r', p,t_HO_tot_2,'-^b', p,t_HO_tot_3,'-^g',  
p,t_HO_tot_4,'-^y', p,t_HO_tot_5,'-^k', p,t_HO_tot_6,'-^c',  
p,t_HO_tot_7,'-^m', p,t_HO_tot_8,'-^r', p,t_HO_tot_9,'-^k')  
grid on  
xlabel ('Faktor Utilitas')  
ylabel ('Delay End-to-End HO (s)')
```

