

BAB II

TINJAUAN PUSTAKA

2.1 Umum

Perkembangan teknologi telekomunikasi yang begitu pesat sangat ditujukan untuk mendukung mobilitas manusia yang cukup tinggi. Sehingga munculah beberapa aplikasi multimedia yang digunakan dalam melayani kebutuhan komunikasi secara bergerak. Salah satu aplikasi yang digunakan adalah *video conference*. Teknologi ini memungkinkan dua atau lebih tempat untuk saling berhubungan secara interaktif *via* transmisi *audio* dan *video* melalui dua jurusan yang dilakukan secara serempak. Aplikasi ini bersifat *realtime* dimana *user* seolah-olah bertatap muka langsung melalui *handset* yang digunakan.

Salah satu jaringan yang dapat mendukung layanan *video conference* adalah jaringan LTE (*Long Term Evolution*) dengan menggunakan standar 3GPP release 8 sesuai dengan pengembangan oleh 3GPP (*The 3rd Generation Partnership Project*). Teknologi ini mampu mendukung mobilitas pengguna sistem *video conference*, karena memiliki nilai *latency* yang rendah tidak membutuhkan waktu *buffering* yang lama. Ketersediaan *bandwith* sebesar 100 Mbps pada kanal *downlink* dan 50 Mbps pada kanal *uplink* diharapkan mampu memenuhi kebutuhan *bandwidth* untuk kanal *audio* dan *video* yang sebesar 36 Kbps dalam komunikasi antar *user*. Salah satu teknologi penting pada LTE adalah penggunaan modulasi adaptif. Modulasi adaptif memungkinkan LTE mengatur pola sinyal modulasi sesuai kondisi *signal to noise ratio* (SNR) link radio.

Pada bab ini akan dijelaskan teori penunjang mengenai teknologi LTE, teknik modulasi adaptif, dan sistem *video conference* antara lain seperti pengertian jaringan LTE, teknik dan sistem *video conference* serta parameter-parameter yang digunakan untuk mengetahui performansi *video conference* pada LTE.

2.2 LTE (*Long Term Evolution*)

LTE disebut juga sebagai E-UTRA (*Evolved Universal Terrestrial Radio Access*) dan diperkenalkan oleh 3GPP sebagai release 8. LTE adalah teknologi telekomunikasi seluler yang merupakan pengembangan dari teknologi UMTS/WCDMA/HSPA. LTE menggunakan spesifikasi *air interface* yang baru untuk meningkatkan kecepatan data dibandingkan dengan HSPA.

Perbedaan utama antara LTE dengan teknologi sebelumnya adalah penggunaan teknologi OFDMA pada sisi *downlink* dan SC-FDMA (*Single Carrier Frequency Division Multiple Access*) pada sisi *uplink*, sehingga LTE memiliki kemampuan mengirimkan data dengan kecepatan tinggi mencapai 100 Mbps untuk *downlink* dan 50 Mbps untuk *uplink*. Peningkatan kecepatan ini juga disebabkan kemampuan LTE menggunakan teknologi MIMO. LTE dirancang untuk meningkatkan *throughput*, meningkatkan kapasitas *base station*, mengurangi *latency*, dan beroperasi baik untuk mobilitas penuh.

2.2.1 Spesifikasi Teknis LTE

Standar LTE yang digunakan untuk komunikasi seluler mengacu pada standar yang telah ada yaitu 3GPP *release 8*. Seperti ditunjukkan pada Tabel 2.1

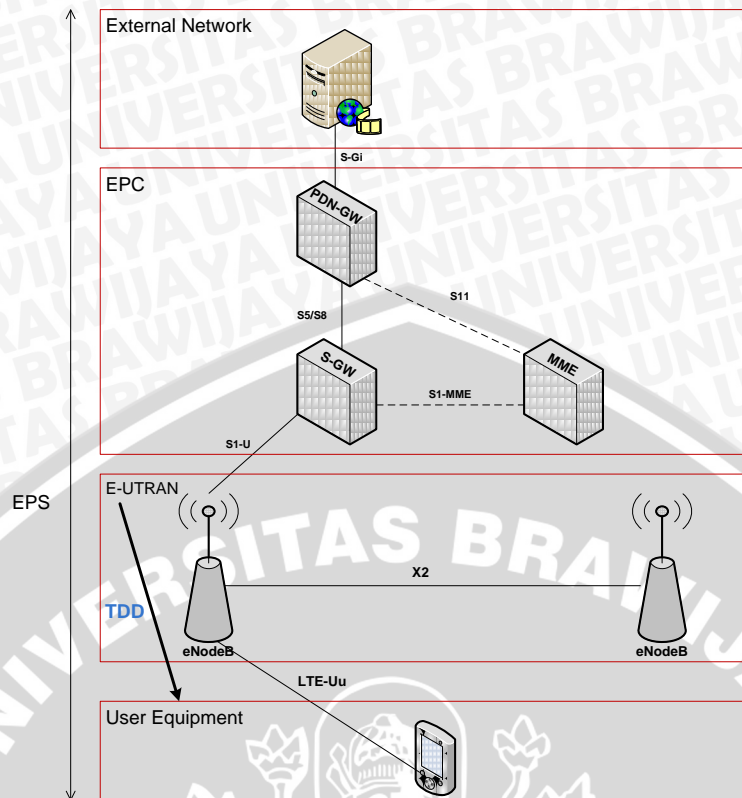
Tabel 2.1 Spesifikasi Teknis LTE

Parameter Spesifikasi Teknis	Jenis/Nilai
<i>Peak data rate</i>	100 Mbit/s for <i>downlink</i> ; 50 Mbit/s for <i>uplink</i>
<i>Average user throughput per MHz than HSPA Release 6</i>	3-4 higher for <i>downlink</i> ; 2-3 higher for <i>uplink</i>
<i>Mobility</i>	0-15 km/h (optimum); 15-120 km/h (<i>high performance guaranteed</i>); 120-350 km/h (<i>connection maintained</i>)
<i>Bandwidth</i>	1.25-20 MHz
<i>Spectrum allocation</i>	Operation in paired spectrum (FDD) and unpaired spectrum (TDD) should be supported
<i>Multiple access</i>	OFDMA (<i>downlink</i>) SC-FDMA (<i>uplink</i>)
MIMO	<i>Downlink</i> 2x2, 4x2, 4x4 <i>Uplink</i> 1x2, 1x4
<i>Modulation</i>	QPSK, 16-QAM, 64-QAM
<i>Latency</i>	5ms user-plane latency at IP layer, for one-way 100 ms control-plane latency from idle to active state
<i>User per cell</i>	At least 200 at 5MHz bandwidth At least 400 at bandwidth > 5MHz

(Sumber : K. Fazel dan S. Kaiser, 2008)

2.2.2 Arsitektur Jaringan LTE

Keseluruhan jaringan LTE disebut juga dengan 3GPP EPS (*Evolved Packet Access*). Arsitektur jaringan LTE ditunjukkan pada Gambar 2.1



Gambar 2.1. Arsitektur LTE

(Sumber : H. Holma dan A. Toskala, 2009)

- UE (*User Equipment*)

UE merupakan perangkat yang digunakan pada sisi pengguna untuk berkomunikasi. UE terdiri dari *Terminal Equipment* (TE) yang berfungsi sebagai terminal radio yang digunakan untuk komunikasi lewat radio dan *Universal Subscriber Identity Module* (USIM) yang berfungsi sebagai bukti dan identitas pengguna serta untuk memperoleh *security key* dalam melindungi *interface* transmisi radio (H. Holma dan A. Toskala, 2009).

- E-UTRAN (*Evolved – UMTS Terrestrial Radio Access Network*)

Jaringan akses LTE yang disebut E-UTRAN adalah jaringan radio akses yang menyediakan koneksi antara UE dan EPC (*Evolved Packet Core*). E-UTRAN hanya terdiri dari eNB (*Evolved Node B*). eNB sendiri adalah unit fisik yang berfungsi untuk mengirim/menerima frekuensi pada sel. Perangkat ini sama dengan BTS (*Base Transceiver Station*) dalam suatu jaringan GSM.

UE berhubungan dengan eNB dan selanjutnya eNB berhubungan dengan EPC serta yang lainnya seperti terlihat pada Gambar 2.1. Untuk trafik pengguna biasa, tidak ada pengontrol terpusat pada E-UTRAN, maka

arsitektur E-UTRAN bisa dikatakan *flat*. [Stefania Sesia, Issam Toufik and Matthew Baker, 2009]

- EPC (*Evolved Packet Access*)

EPC berfungsi untuk melakukan pengontrolan secara menyeluruh dari eNodeB dan pembentukan *bearer*. EPC terdiri dari :

1. PCRF (*Policy Control and Charging Rules Function*)

PCRF berfungsi untuk mengontrol kebijakan pembuatan keputusan dan kemampuan aliran data berdasarkan PCEF (*Policy Control Enforcement Function*) yang terletak pada PDN-GW (*Packet Data Network Gateway*). PCRF menyediakan otorisasi QoS (*Quality Of Service*) yang menentukan bagaimana aliran data tertentu akan diperlakukan dalam PCEF dan memastikan bahwa hal ini sesuai dengan profil pengguna berlangganan.

2. HSS (*Home Subscription Server*)

HSS berfungsi untuk menyimpan data dan informasi mengenai pelanggan agar tersimpan secara permanen. HSS juga berisi rekaman lengkap lokasi terkini dari *user*.

3. PDN-GW (*Packet Data Network Gateway*)

PDN-GW merupakan jangkar untuk mobilitas antara 3GPP dengan teknologi *non-3GPP* seperti WiMAX, 3GPP2 dan WLAN (*Wireless Local Area Network*) melalui beberapa *interface*. PDN-GW menyediakan konektivitas dengan jaringan paket data eksternal. Selain itu PDN-WG bertanggung jawab untuk mengalokasikan alamat IP UE, pengisian aliran data berdasarkan aturan dari PCRF dan menyaring *downlink* paket IP *user* kedalam *bearer* QoS yang berbeda.

4. S-GW (*Serving Gateway*)

Semua pengguna paket IP dikirimkan melalui S-GW, yang berfungsi sebagai jangkar untuk mobilitas data *bearer* bila UE bergerak di antara eNodeB. S-GW juga bertanggung jawab dalam menetapkan *routing* dan meneruskan paket data *user* dengan *interface* S-1, menangani kompresi *header* IP serta enkripsi data *user*.

5. MME (*Mobility Management Entity*)

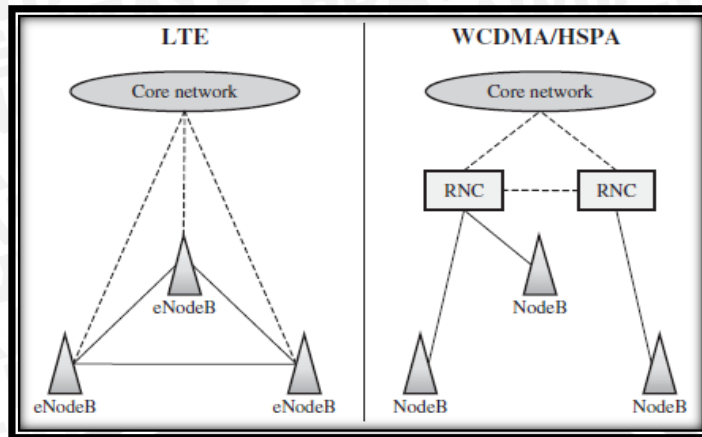
MME adalah *node* kontrol yang memproses sinyal antara UE dan CN (*Core Network*)/EPC. Selain itu MME juga berfungsi sebagai autentifikasi dan keamanan serta *mobility management*. Protokol yang berjalan antara UE dan CN dikenal sebagai protokol *Non-Access Stratum* (NAS). [Stefania Sesia, Issam Toufik dan Matthew Baker, 2009 : 25]

- *Interface* jaringan

1. *Interface* LTE-Uu, yaitu *interface* yang menghubungkan eNodeB dan UE.
2. *Interface* X2, yaitu *interface* yang menghubungkan antar eNodeB.
3. *Interface* S1, yaitu *interface* yang menghubungkan EPC dan eNodeB.
4. *Interface* S1-MME, yaitu *interface* yang menghubungkan MME dan eNodeB.
5. *Interface* S1-U, yaitu *interface* yang menghubungkan S-GW dan eNodeB.
6. *Interface* SGi, yaitu *interface* yang menghubungkan PDN-GW dan jaringan paket data.
7. *Interface* S5/S8, yaitu *interface* yang menghubungkan S-GW dan PDN-GW.
8. *Interface* S11, *interface* yang menghubungkan MME dan S-GW. [Mustafa Ergen, 2009 : 388]

2.2.3 Arsitektur RAN (*Radio Access Network*)

Perbandingan arsitektur RAN (*Radio Access Network*) dari LTE dan WCDMA/HSPA ditunjukkan pada Gambar 2.2. Perbedaan yang signifikan antara WCDMA/HSPA dan LTE adalah dalam LTE tidak mendukung *macro diversity*, yang merupakan permasalahan untuk WCDMA/HSPA. Pada LTE, *Radio Network Controller* (RNC) dihilangkan sehingga dapat mengurangi *latency* di RAN. Hal ini menyebabkan LTE memindahkan kompleksitas ke eNB.

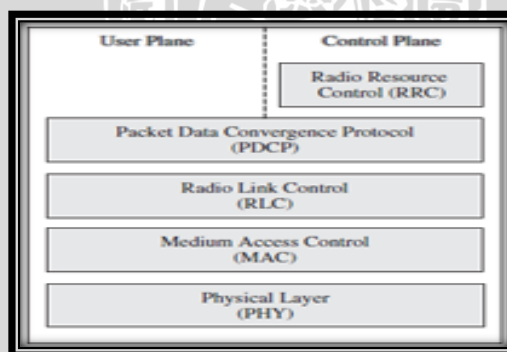


Gambar 2.2 Perbandingan RAN arsitektur antara LTE dan WCDMA/HSPA
(Sumber : K. Fazel dan S. Kaiser, 2008)

Selain pengolahan lapisan fisik, eNB pada LTE juga mempunyai tugas untuk mengatur mobilitas dan manajemen sumber daya radio, yang keduanya termasuk dalam tugas RNC pada WCDMA/HSPA. eNB dalam RAN LTE masing-masing dihubungkan secara langsung satu sama lain dan tugas untuk *handover* diambil oleh eNB. Dalam WCDMA/HSPA, tugas ini dilakukan oleh RNC. [K. Fazel dan S. Kaiser, 2008]

2.2.4 Arsitektur Radio Protokol

Arsitektur radio protokol pada LTE dibagi menjadi *user plane* dan *control plane*. *User plane* membawa semua informasi pengguna dari suara ke data sementara *control plane* digunakan untuk sinyal kontrol yang ditunjukkan pada Gambar 2.3.



Gambar 2.3 User and control plane protocol stack
(Sumber : K. Fazel dan S. Kaiser, 2008)

Semua protokol yang terletak di eNB dan terminal *mobile station* (UE). Fungsi dari masing-masing lapisan protokol dirangkum seperti berikut ini.

- RRC (*Radio Resource Control*)

RRC adalah bagian dari *control plane* yang mengatur kinerja protokol PDCP, RLC, MAC, dan PHY layer 1 dan layer 2. Fungsi utama dari RRC

adalah *admission control*, manajemen *handover* dan QoS, mengatur dan melaporkan pengukuran stasiun terminal.

- PDCP (*Packet Data Convergence Protocol*)

Fungsi utama dari PDCP pada *user plane* adalah mengompres *header IP*, pengiriman data dari *user* dan *chipering*. Pada *control plane*, PDCP bertanggung jawab dalam fungsi pengendalian pengiriman data dan *ciphering*.

- RLC (*Radio Link Control*)

Segmentasi dan *reassembly* paket data dari *layer* yang lebih tinggi serta perbaikan kesalahan melalui ARQ adalah fungsi utama dari RLC. Selain itu, *flow control* antara eNodeB dan *mobile terminal* juga ditangani oleh RLC.

- MAC (*Medium Access Control*)

MAC bertanggung jawab untuk penjadwalan *uplink* dan *downlink*, perbaikan kesalahan melalui HARQ (*hybrid ARQ*), modulasi adaptif, pemetaan antena dan juga berfungsi untuk *mapping* antara kanal logika dan kanal transport. [K. Fazel dan S. Kaiser, 2008 : 220]

- PHY (*Physical Layer*)

Layer fisik LTE membawa data dan mengendalikan informasi antara eNB dan UE dengan cara yang efisien. Layer fisik LTE menggunakan teknologi yang lebih maju seperti OFDMA, SC-FDMA dan MIMO untuk transmisi data.

2.2.5 Alokasi Spektrum Pada LTE

Secara umum, alokasi spektrum pada LTE ditunjukkan pada Tabel 2.2. Berdasarkan ketentuan 3GPP, terdapat 17 *bands* untuk FDD dan 8 *bands* untuk TDD. Saat ini, beberapa *band* telah digunakan oleh teknologi lain dan LTE dapat beroperasi secara bersama-sama dengan teknologi tersebut. [H. Holma dan A. Toskala, 2009 : 8]

Tabel 2.2 Spektrum Frekuensi LTE

LTE Band	Uplink (UL) (MHz)	Downlink (DL) (MHz)	Duplex Mode	Regions
1	1920-1980	2110-2170	FDD	Europa, Asia
2	1850-1910	1930-1990	FDD	Amerika, Asia
3	1710-1785	1805-1880	FDD	Europa, Asia, Amerika
4	1710-1755	2110-2155	FDD	Amerika
5	824-849	869-894	FDD	Amerika
6	830-840	875-885	FDD	Jepang
7	2500-2570	2620-2690	FDD	Europa, Asia

8	880-915	925-960	FDD	Eropa, Asia
9	1749.9-1784.9	1844.9-1879.9	FDD	Jepang
10	1710-1770	2110-2170	FDD	Amerika
11	1427.9-1452.9	1475.9-1500.9	FDD	Jepang
12	698-716	728-746	FDD	Amerika
13	777-787	746-756	FDD	Amerika
14	788-798	758-768	FDD	Amerika
17	704-716	734-746	FDD	-
18	815-830	860-875	FDD	-
19	830-845	875-890	FDD	-
...				
33	1900-1920	1900-1920	TDD	Eropa, Asia
34	2010-2025	2010-2025	TDD	Eropa, Asia
35	1850-1910	1850-1910	TDD	-
36	1930-1990	1930-1990	TDD	-
37	1910-1930	1910-1930	TDD	-
38	2570-2620	2570-2620	TDD	Eropa
39	1880-1920	1880-1920	TDD	Cina
40	2300-2400	2300-2400	TDD	Eropa, Asia

(Sumber : H. Holma dan A. Toskala, 2009)

2.2.6 Modulasi dan Teknik Modulasi Adaptif

Pada LTE, teknik modulasi adaptif mampu menyesuaikan jenis modulasi sesuai dengan kondisi *link* saat itu. Modulasi yang dapat digunakan, yaitu QPSK, 16-QAM, dan 64-QAM

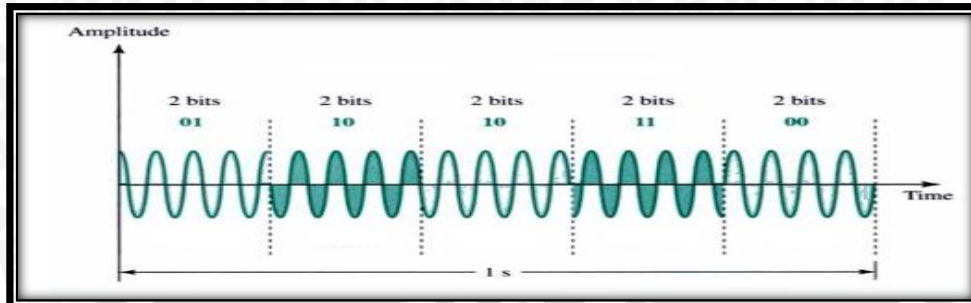
2.2.6.1 Modulasi

Modulasi adalah proses pengkodean informasi dari sumber pesan dengan cara yang sesuai dengan proses transmisi. Pada modulasi digunakan sinyal *carrier* yang memiliki nilai frekuensi yang lebih tinggi dibandingkan dengan nilai frekuensi sinyal pesan. Sinyal pesan disebut sebagai sinyal pemodulasi dan sinyal *carrier* disebut sinyal termodulasi. Pada umumnya modulasi dapat dilakukan dengan mengubah-ubah amplitudo, fasa, atau frekuensi dari sinyal *carrier* sesuai dengan amplitudo sinyal pesan. Akan tetapi pada perkembangannya teknik modulasi sudah dapat mengkombinasikan perubahan amplitudo, fasa, dan frekuensi dalam suatu teknik modulasi. Terdapat dua jenis teknik modulasi yang digunakan pada pentransmisi data digital dalam suatu jaringan LTE yaitu:

a. QPSK

QPSK merupakan teknik modulasi yang merubah atau memodulasikan fasa sinyal *carrier* yang memiliki jumlah fasa sebanyak 4 buah. QPSK mentransmisikan data dua kali lebih cepat karena terdapat dua bit dalam tiap simbolnya. Misalnya fasa 0° merepresentasikan bit 00, fasa 90° merepresentasikan bit 01, fasa 180°

merepresentasikan bit 10, dan fasa 270° merepresentasikan bit 11. Pasangan bit-bit yang merepresentasikan masing-masing fasa disebut dengan dibit. Kelemahan dari QPSK yaitu kurang tahan terhadap *noise* dan rawan terjadi interferensi dari sinyal lain. Gambar 2.4 memperlihatkan konsep dari modulasi QPSK.



Gambar 2.4 Sinyal pada *Quadrature Phase Shift Keying* (QPSK)
(Sumber : Behrouz A. Forouzan, 2000)

QPSK memodulasi fasa dengan 2 bit per simbol sehingga simbol satu dengan simbol yang lain berbeda fasa dengan kelipatan $\pi/2$. Gambar 2.5 di bawah menjelaskan hubungan antara fasa dengan bit pada QPSK yang disebut dengan *constellation* atau *phase state diagram*. [Behrouz A. Forouzan, 2000].

QPSK memodulasi fasa dengan 2 bit per simbol sehingga simbol satu dengan simbol yang lain berbeda fasa dengan kelipatan $\pi/2$ seperti yang dirumuskan pada persamaan berikut : [Rappaport, 2001 : 243]

$$\begin{aligned}
 \text{bit } 00 &= s_1(t) = \sqrt{\frac{2E_s}{T}} \cos\left(2\pi f_c t + \frac{\pi}{4}\right) \\
 \text{bit } 01 &= s_2(t) = \sqrt{\frac{2E_s}{T}} \cos\left(2\pi f_c t + \frac{3\pi}{4}\right) \\
 \text{bit } 11 &= s_3(t) = \sqrt{\frac{2E_s}{T}} \cos\left(2\pi f_c t + \frac{5\pi}{4}\right) \\
 \text{bit } 10 &= s_4(t) = \sqrt{\frac{2E_s}{T}} \cos\left(2\pi f_c t + \frac{7\pi}{4}\right)
 \end{aligned} \tag{2-1}$$

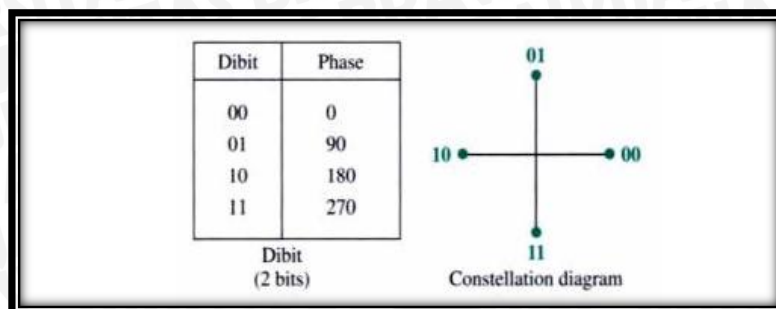
dengan :

$s(t)$ = sinyal QPSK

E_s = energi per simbol

T = durasi simbol

f_c = frekuensi *carrier*

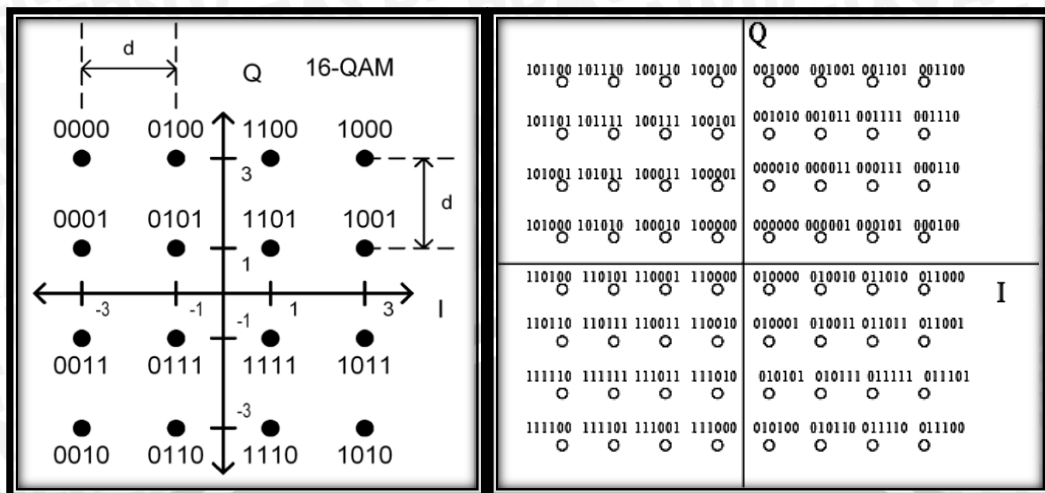


Gambar 2.5 *Quadrature Phase Shift Keying (QPSK) constellation*
(Sumber : Behrouz A. Forouzan, 2000)

b. QAM

QAM merupakan teknik yang mengkombinasikan antara teknik *Amplitude Shift Keying* (ASK) dan teknik *Phase Shift Keying* (PSK). Pada QAM, amplitudo dan *phase* sinyal *carrier* berubah terhadap perubahan amplitudo sinyal informasi. Sehingga sinyal direpresentasikan dalam besaran amplitudo dan pergeseran *phase*. Dikatakan *quadrature* karena jumlah minimum *bit* pada tiap simbolnya adalah 2 *bit*. Sehingga dikenal teknik 4-QAM, 8-QAM, 16-QAM, 32-QAM, 64-QAM, 128-QAM, dan 256-QAM tergantung banyaknya jumlah *bit* dalam satu simbol. Sesuai standar yang telah ditetapkan oleh 3GPP LTE, teknik QAM yang digunakan pada teknologi LTE adalah 16-QAM dan 64-QAM. Semakin banyak jumlah titik konstelasi yang digunakan, semakin besar kemampuan untuk mentransmisikan bit-bit data untuk tiap simbol. [K. Fazel and S. Kaiser, 2008 : 222]

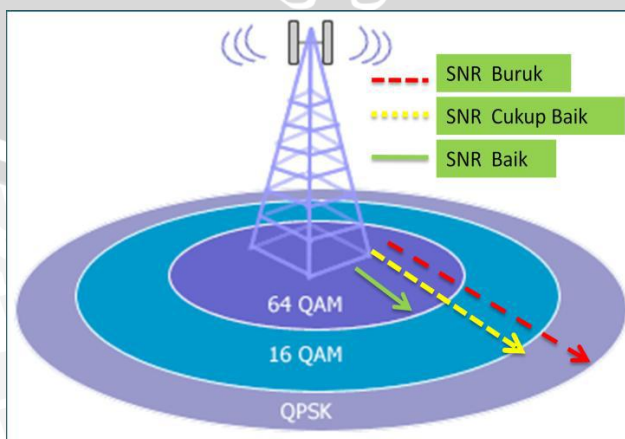
Untuk 16-QAM terdapat 4 bit setiap simbolnya, sehingga disebut juga dengan *quadbit*. Sedangkan untuk 64-QAM terdapat 6 bit pada tiap simbolnya, sehingga disebut juga dengan *hexabit*. Teknik 64-QAM merupakan teknik yang paling cepat mentransmisikan data pada teknologi LTE, karena jumlah bit per simbolnya adalah yang tertinggi dibandingkan dengan teknik modulasi digital yang lain, yaitu 6 bit. Gambar 2.6 menjelaskan hubungan antara amplitudo dan fasa dengan bit pada 16-QAM dan 64-QAM. [Behrouz A. Forouzan, 2000]



Gambar 2.6 (a) 16-QAM constellation, (b) 64-QAM constellation
(Sumber : Behrouz A. Forouzan, 2000)

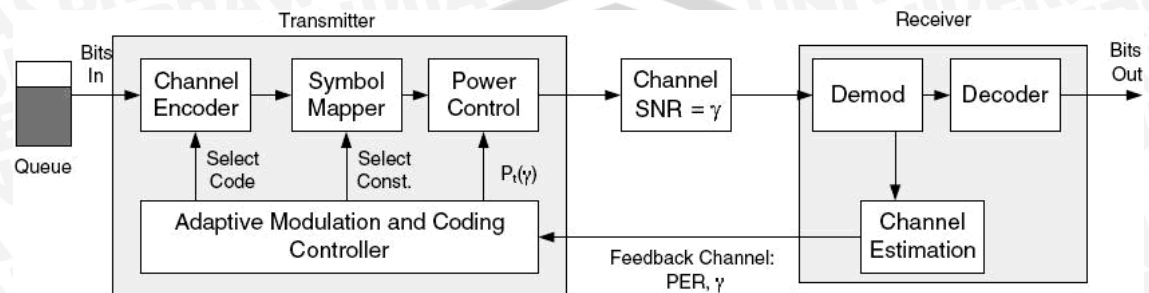
2.2.6.2 Teknik Modulasi Adaptif

Sistem modulasi adaptif melakukan perubahan jenis modulasi sesuai dengan kondisi *link* radio saat itu. Misalkan, saat kondisi *link* radio baik, maka akan meningkatkan nilai SNR sehingga dapat digunakan teknik modulasi yang menghasilkan *bit rate* tertinggi dengan BER yang rendah. Saat *link* radio buruk akan menurunkan nilai SNR sehingga memaksa penggunaan teknik modulasi dengan *bit rate* yang lebih rendah untuk mempertahankan reabilitas *link*. Ketika kondisi link baik maka modulasi 64-QAM akan dipilih untuk digunakan daripada modulasi QPSK karena memiliki *bit rate* lebih cepat. Kondisi ini diperlihatkan pada Gambar 2.18. Pada Gambar tersebut dapat dilihat bahwa semakin dekat daerah layanan dengan antenna pengirim, maka digunakan modulasi dengan level lebih tinggi sehingga digunakan *bit rate* yang lebih tinggi. Modulasi adaptif memungkinkan adanya efisiensi spektrum dan kekebalan transmisi pada kondisi kanal yang bervariasi terhadap waktu.



Gambar 2.7 Penggunaan modulasi adaptif pada kondisi link radio yang berbeda
(Sumber : <http://repository.usu.ac.id/handle/123456789/21436>)

Pada teknik modulasi adaptif *receiver* akan mengirimkan *channel quality indicator* (CQI) berisi level SNR kepada *transmitter* dan nilai tersebut akan dibandingkan dengan nilai *threshold* SNR dan standar BER yang ditetapkan sebelumnya sehingga pengirim akan memutuskan untuk mengubah jenis modulasi pada transmisi berikutnya sesuai dengan informasi yang diterimanya dari penerima. Blok diagram dari sistem modulasi adaptif ditunjukkan pada Gambar 2.8



Gambar 2.8 Blok Diagram Sistem Modulasi Adaptif
(Sumber : <http://repository.usu.ac.id/handle/123456789/21436>)

Tinjau sebuah sistem *single-user* yang berusaha mentransmisikan data secepat mungkin melalui kanal dengan SNR yang berubah-ubah. Tujuan dari pemancar (*transmitter*) adalah untuk mentransmisikan data secepat mungkin, dengan sasaran agar data yang di-demodulasi dan di-dekodekan tetap handal pada sisi penerima (*receiver*). Oleh karena itu kanal *feedback* memegang peranan yang sangat penting untuk modulasi adaptif. Pemancar harus mengetahui “SNR kanal” γ , yang didefinisikan sebagai SNR yang diterima γ_r dibagi dengan daya pancar P_t . Dengan demikian SNR yang diterima $\gamma_r = P_t \cdot \gamma$.

Pada prakteknya, *feedback* (umpan balik) akan menghasilkan sedikit waktu tunda (*delay*) dan mungkin degradasi sistem akibat estimasi kanal yang tidak tepat ataupun akibat terjadi *error* pada kanal *feedback*. Sistem LTE memproteksi dengan kuat kanal *feedback* dengan koreksi *error*, jadi sumber utama degradasi sistem biasanya adalah mobilitas *user*, yang menyebabkan estimasi kanal dengan cepat menjadi tidak terpakai. Secara kasar, *user* dengan kecepatan lebih dari 30 km/jam pada frekuensi pembawa 2100 MHz tidak akan mampu menyediakan informasi status kanal secara akurat dan tepat waktu pada sisi pemancar. Untuk menangani masalah ini, LTE secara opsional mendukung perulangan *preamble* secara lebih sering. *Preamble* ialah *header* subframe yang berisi prosedur lapis fisik, seperti sinkronisasi waktu dan frekuensi, serta informasi awal estimasi kanal. Pada *uplink*, *preamble* pendek, dinamakan *midamble*, dapat digunakan setelah 8, 16, atau 32 simbol. Sedangkan pada *downlink*, *preamble* pendek dapat disisipkan pada permulaan setiap *burst*. Diperkirakan bahwa dengan

menggunakan *midamble* setiap 10 simbol akan memberikan mobilitas hingga 150 km/jam.

2.3 Video Conference

Video conference merupakan layanan komunikasi interaktif jarak jauh yang melibatkan *video* dan *audio* secara *realtime*. Kebutuhan *bandwidth* pada layanan ini cukup besar karena digunakan untuk mengirimkan sinyal *video* dan *audio* secara bersamaan. Sehingga teknik-teknik untuk melakukan kompresi data menjadi sangat strategis untuk memungkinkan penghematan *bandwidth* komunikasi.

Sebagai gambaran sebuah kanal gambar (*video*) yang baik tanpa dikompresi akan mengambil *bandwidth* sekitar 9 Mbps, sedangkan sebuah kanal suara/*audio* yang baik tanpa di kompresi akan mengambil *bandwidth* sekitar 64 Kbps. Dari gambaran di atas dapat diasumsikan bahwa kebutuhan minimal *bandwidth* yang diperlukan untuk mengirim gambar dan suara adalah 9,064 Mbps, memang akan memakan *bandwidth* yang sangat lebar. Namun dengan teknik kompresi yang ada kanal *video* bisa dikompresi hingga menjadi 30 Kbps dan kanal suara menjadi 6 Kbps. Sehingga untuk kebutuhan *multiuser* akan dibutuhkan multi *bandwidth* juga, artinya untuk melakukan *video conference* dibutuhkan *bandwidth* minimal 32-36 Kbps dikali dengan jumlah *user* yang melakukan *video conference* dalam satu jaringan.

2.3.1 Kompresi Audio

Beberapa teknik kompresi data *audio* ditunjukkan pada Tabel 2.3.

Table 2.3 Kompresi suara

Kompresi	Bandwidth (Kbps)	Delay(ms)
G.711 PCM	64	0.125
G.726 ADPCM	32	0.125
G.728 LD-CELP	16	0.625
G.729 CS-ACELP	8	10
G.729 x2 Encoding	8	10
G.729 x3 Encoding	8	10
G.729a CS-ACELP	8	10
G.723.1 MPMLQ	6.3	30
G.723.1 ACELP	5.3	30

Sumber : Onno W.Purbo, 2003:2

Kolom Kbps memperlihatkan berapa lebar *bandwidth* yang diambil untuk mengirimkan suara yang dikompres menggunakan teknik kompresi tertentu. *Delay* (ms) adalah waktu yang dibutuhkan untuk melakukan kompresi. Sehingga digunakan standar

G.711 untuk mengkompres suara kita agar menghemat *bandwidth* saluran komunikasi pada jaringan.

2.3.2 Kompresi Video

Pada teknik kompresi *video* ada tiga buah saluran standar yang umum digunakan, dalam pengiriman *video* melalui saluran komunikasi yang sempit, yaitu:

- H.261 – biasanya menggunakan kanal ISDN dengan kecepatan $p \times 64$ Kbps, dimana p adalah 1, 2, 3, ..., 30.
- H.263 – diarahkan untuk mengirimkan gambar *video* berkecepatan rendah mulai dari 20-30 Kbps ke atas.
- H.264 – efisiensi kompresi yang lebih baik dalam hal kompresi *video* dan fleksibilitas yang lebih besar dalam melakukan kompresi.

H.264 dikenal sebagai teknologi kompresi MPEG-4 generasi ke-10 dimana teknologi kompresi ini menggunakan format MPEG-4, namun memiliki sejumlah perbaikan dimana teknologi kompresi H.264 memiliki kemampuan untuk memprediksi perpindahan antar gambar *video* hingga 32 kemungkinan. Kelebihan inilah yang membuat teknologi kompresi H.264 menghasilkan kompresi data yang lebih baik dari pada MPEG-4 sehingga membuat teknologi kompresi ini paling ideal dalam *video conference*. Beberapa hal yang perlu diperhatikan adalah:

- Jika kita menggunakan *video* hitam-putih, maka akan memakan *bandwidth* lebih kecil dari pada jika kita melakukan konferensi menggunakan *video* berwarna.
- Jika kita menggunakan kecepatan pengiriman fps (*frame per second*) *video* yang rendah, maka akan memakan *bandwidth* yang rendah dibandingkan fps (*frame per second*) yang tinggi.

Video yang cukup baik biasanya dikirim dengan kecepatan fps (*frame per second*), yaitu sekitar 30 fps. Jika dikirimkan tanpa kompresi, sebuah *video* dengan 30 fps akan mengambil *bandwidth* kira-kira 9 Mbps, amat sangat besar untuk ukuran kanal komunikasi data.

2.3.3 Coder Decoder (CODEC)

Salah satu komponen yang terpenting dalam *video conference* adalah peralatan *codec* yang menggunakan teknik penyamplingan sinyal analog untuk dirubah menjadi

sinyal digital lalu mereduksi lebar pita sinyal sesuai dengan kebutuhan. Algoritma sebagai proses pengkodean sinyal-sinyal informasi sehingga lebar pita sinyal tersebut dapat direduksi, dipakai pada alat *codec* ini untuk pengkompresian data yang telah didapat dari hasil sampling.

Pada dasarnya perencanaan kecepatan data keluaran dari *codec* ini dapat disesuaikan dengan kebutuhan, akan tetapi ada satu rekomendasi dari badan Internasional CCITT (*The International Telegraph and Telephone Consultative Commite*) No. H.261 Tahun 1990 yang memberikan saran bagi para perekayasa sistem untuk membuat *codec video* bagi pelayanan *audio visual* pada kecepatan $p \times 64$ Kbps dengan nilai p bilangan bulat berkisar antara 1 – 30.

Terdapat dua jenis CODEC yang biasanya dapat digunakan, yaitu (Andreas Handoyo, dkk, 2009: 1):

- *Lossy* CODEC

CODEC ini akan mengurangi kualitas data dengan mengurangi ukuran data (kompresi). Pada umumnya, CODEC ini digunakan untuk menyimpan data pada media penyimpanan yang berukuran terbatas seperti CD-ROM dan DVD. *Lossy* CODEC ini biasanya digunakan untuk *conference*, karena *bandwidth* jaringan yang terbatas. Contoh: Windows Media Video, H.264.

- *Lossless* CODEC

Pada *lossless* CODEC ini, kualitas data yang dihasilkan tidak akan berkurang. Tetapi, ukuran data yang dihasilkan oleh *lossless* CODEC ini akan lebih besar dibandingkan dengan *lossy codec*. Pada umumnya, *lossless* CODEC ini digunakan pada *video* yang masih memerlukan *editing*, karena dalam proses *editing* dilakukan *encode-decode* berulang kali, sehingga jika menggunakan *lossy* CODEC, kualitas *video* akan jauh menurun dibandingkan dengan *video* aslinya. Contoh: CorePNG, huffyuv, *Apple Lossless Audio Codec*.

Jenis CODEC yang digukana pada layanan *video conference* adalah G.711 untuk *audio* dan H.264 untuk *video*, beberapa macam teknik coding lain yang sering adalah sebagai berikut seperti pada tabel 2.4

Tabel 2.4 Audio dan video Coding

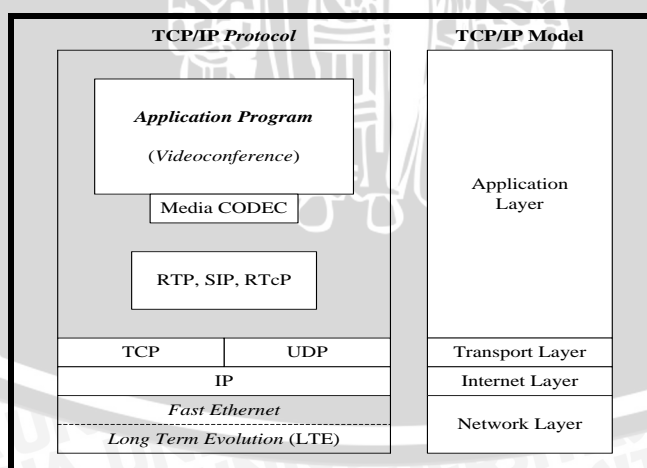
Audio CODEC	Bit rate	Delay CODEC
G. 711	64 Kbps	0,75 ms
G.26	32 Kbps	1 ms
G.723.1	5,3/6,3 Kbps	30 ms
G.728	16 Kbps	3-5 ms
G.729	8 Kbps	10 ms
Video CODEC	Bit rate	Delay CODEC
H.261	p x 64 Kbps	20-150 ms
H.263	10-384 Kbps	20-150 ms
H.264/AVC	64-384 Kbps	150-300 ms

Sumber: Cisco System, 2006

2.3.4 Protokol video conference

Protokol merupakan sekumpulan aturan yang mendefinisikan beberapa fungsi seperti pembuatan hubungan, mengirim pesan, data, informasi atau *file* yang harus dipenuhi oleh pengirim dan penerima agar suatu sesi komunikasi dapat berlangsung dengan baik dan benar. Protokol yang digunakan dalam pelaksanaan *video conference* adalah H.323. Protokol ini mendefinisikan format, urutan *message* yang dikirim dan yang diterima oleh entitas jaringan, serta tindakan (*action*) yang harus diambil bila mengirimkan atau menerima *message*.

Pada aplikasi *video conference* secara *realtime*, keutuhan data bukanlah prioritas utama karena pentransmisian kembali paket tersebut yang hilang akan memperburuk kualitas penerimaan di tujuan, karena *audio* dan *video* bersifat sensitif terhadap waktu.



Gambar 2.9 Struktur Layer TCP/IP

(Sumber: William Stalling: 21, 2002)

Dari gambar di atas terlihat mekanisme protokol pada *video conference* dikelompokkan menjadi beberapa bagian. Dari paling bawah yaitu IP yang berfungsi



sebagai *internet layer*, kemudian TCP dan UDP berfungsi sebagai *transport layer*, dan RTP berfungsi sebagai *application layer*.

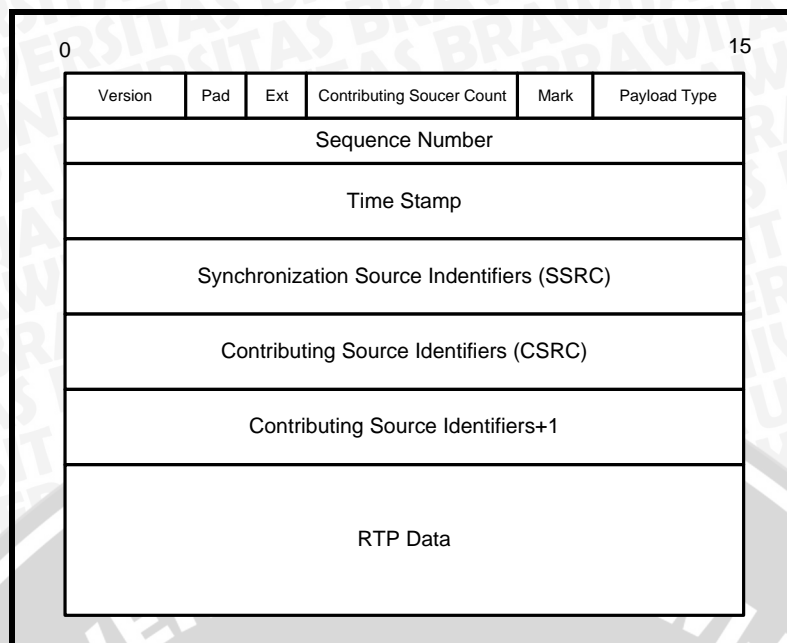
Protokol yang digunakan pada pelaksanaan *video conference* adalah RTP/UDP/IP sesuai ITU-T protokol. Protokol ini digunakan karena pada sistem pelaksanaan *video conference* yang ada pada saat ini secara global telah menggunakan protokol tersebut. Untuk menjamin transmisi paket yang *real time*, RTP digunakan sebagai tambahan pada UDP. RTP bekerja diatas UDP dan IP dan biasanya disebut RTP/UDP/IP dalam melakukan mekanisme transport.

- RTP (*Real Time Protocol*)

RTP merupakan protokol yang menyediakan dukungan *transport data real time* secara *end to end*. RTP terdiri dari dua bagian yaitu *header* dan *data*. RTP memberikan mekanisme transport yang memungkinkan terjadinya sinkronisasi *multimedia data stream* dari berbagai aplikasi misalkan *video* dan *suara*.

Apabila user bekerja dengan aplikasi *video* dan *suara* maka akan terdapat dua data *stream* yaitu *video* dan *suara*, dengan RTP maka paket yang berbeda tersebut akan ditandai sesuai dengan *payload* masing-masing aplikasi sehingga pada user tujuan paket data tersebut dapat disinkronkan (mendukung sinkronisasi dari berbagai media *conference*). *Payload* dari paket RTP berupa format *video* maupun *audio*.

RTP mengidentifikasi tipe data yang ditransmisikan, menentukan urutan paket data yang seharusnya ditampilkan dan menampilkan lalu mensinkronkan stream media dari berbagai sumber.



Gambar 2.10 Format Header RTP
(Sumber: <http://www.Juniper.net>)

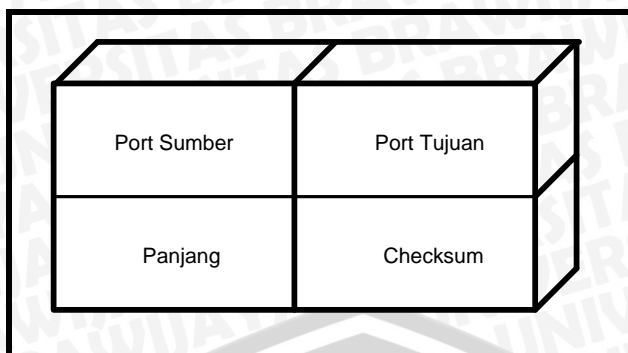
- **UDP (User Datagram Protokol)**

UDP merupakan mekanisme pengiriman *datagram* dari satu aplikasi ke aplikasi lain. UDP berfungsi untuk menyisipkan *field number port* sumber dan tujuan untuk layanan multiplexing. Software UDP dalam komputer dapat menghantarkan pesan ke penerima yang tepat. Tidak ada umpan balik untuk mengontrol tingkat informasi yang dimiliki pesan.

Digunakan UDP karena alasan keringanan, kecepatan, dan reliabilitas transportasi data diantara *host-host* TCP/IP serta UDP memberikn transmisi kanal yang bersifat *nonreliable*. Protokol ini bersifat *connectionless* dan *unreliable*. *Connectionless* berarti tidak diperlukan suatu bentuk hibungan terlebih dahulu untuk mengirimkan data. *Unreliable* berarti pada protokol ini data tidak dijamin akan sampai pada tujuan yang benar dan dalam kondisi yang benar pula. Aplikasi dapat mentolerir adanya sebagian fraksi paket yang hilang dan bahwa reliabilitas transfer data tidak begitu kritis.

Pada sebuah pengiriman *datagram* aplikasi yang menggunakan UDP harus mengetahui identitas tujuan yang bukan hanya berupa alamat IP, untuk itu UDP menyediakan identifikasi yang berupa port baik itu *source* maupun *destination* pada *headernya*.



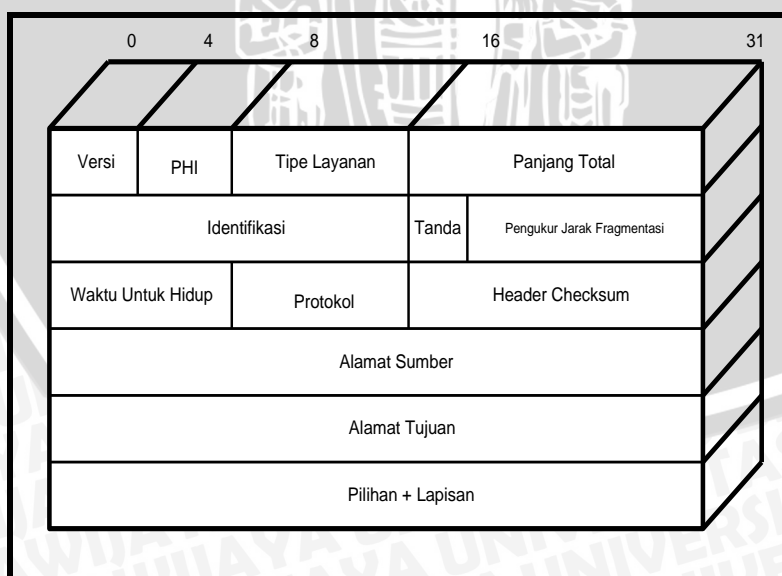


Gambar 2.11 Format Header UDP
(Sumber: William Stalligs, 2002: 274)

- IP (*Internet Protocol*)

IP merupakan protokol yang dapat digunakan untuk komunikasi sepanjang jaringan apapun yang terkoneksi baik *Local Area Network* (LAN) maupun *Wide Area Network* (WAN). IP merupakan protokol jaringan yang terdiri dari informasi pengalaman dan beberapa kontrol yang memungkinkan suatu data diroutekan.

IP merupakan protokol yang bersifat *unreliable*, *best efforts* dan *connectionless*. IP mempunyai dua fungsi dasar yaitu menyediakan hubungan yang bersifat *connectionless*, *best effort* pada saat pengiriman paket dan menangani fragmentasi dari *datagram* untuk mendukung *datalink* dengan besar *maximum-transmissions unit* (MTU) yang berbeda-beda. Seperti ditunjukkan pada gambar 2.11



Gambar 2.12 Format Header IP
(Sumber: William Stalligs, 2002: 167)

Dimana IP memiliki beberapa fungsi sebagai berikut :

1. Mendefinisikan *datagram* yang merupakan unit transmisi elementer di internet.
2. Mendefinisikan skema pengalamatan internet.
3. Melewatkan data antara *Network Access Layer* dan *Host to Host Transport Layer*.
4. *Routing datagram* ke remote *host*.
5. Menjalankan fragmentasi dan penyusunan kembali *datagram*.

2.3.5 Perhitungan *payload video conference*

Pada aplikasi *video conference*, paket data yang ditransmisikan dibedakan atas paket data *audio* dan *video*, dimana tiap paket tersebut mempunyai besar *payload* yang berbeda. Untuk menghitung besarnya *payload audio* maupun *video* digunakan persamaan berikut:

$$P_{LA/V} = B_{CODEC} \times \text{frame rate} \quad (2-2)$$

dengan:

$P_{LA/V}$: *Payload audio/video (byte)*

B_{CODEC} : *Bandwidth CODEC (Kbps)*

frame rate : Waktu pembentukan tiap *frame (s)*

Jumlah paket *audio* dan *video* yang akan diencode dipengaruhi oleh besar *payload* maksimum dari *video* maupun *audio*, sehingga paket *audio* dan *video* dapat dihitung dengan persamaan:

$$P_{A/V} = \frac{P_{LA/V}}{P_{LA/V \max}} \quad (2-3)$$

Dengan:

$P_{LA/V \max}$: *payload maksimum (byte)*

Besarnya paket data *video conference* merupakan penjumlahan *payload* paket *audio* dan *video* dengan *header* NALU, RTP, UDP, dan IPv6, sehingga dapat dihitung dengan persamaan:

$$\text{Paket data} = P_{LA/V} + (P_{A/V} \times (NALU + H_{RTP} + H_{UDP} + H_{IP})) \quad (2-4)$$

Sehingga panjang paket *video conference* sesuai dengan persamaan berikut:

$$W_{\text{data}} = P_{\text{audio}} + P_{\text{video}} \quad (2-5)$$

dengan :

W_{data} : panjang paket *video conference (byte)*



P_{LA} : panjang paket *audio* (byte)

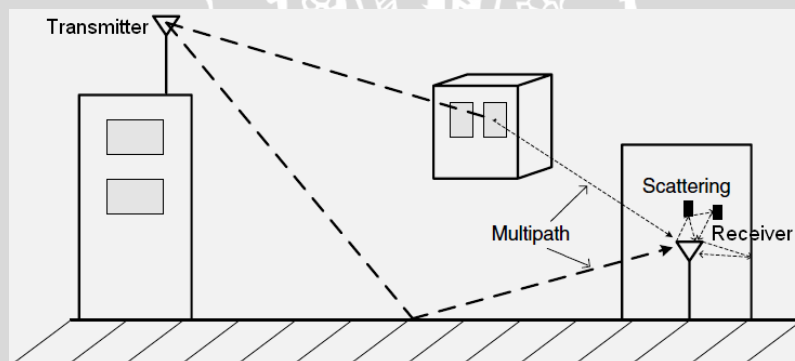
P_{LV} : panjang paket *video* (byte)

2.4 Perhitungan *Link Budget* Teknik Modulasi Adaptif

Perhitungan *link budget* merupakan perhitungan level daya yang dilakukan untuk memastikan bahwa level daya penerimaan lebih besar atau sama dengan level daya *threshold* ($RSL \geq R_{th}$). Tujuannya untuk menjaga keseimbangan *gain* dan *loss* untuk mencapai SNR yang diinginkan di *receiver*. Perhitungan *link budget* juga berguna untuk menghitung luas daerah jangkauan sinyal dari *base station*, seberapa jauh sinyal masih dapat diterima oleh pelanggan dengan baik.

2.4.1 Perhitungan Rugi-Rugi Propagasi

Kanal radio untuk sistem komunikasi *wireless* dibedakan untuk kondisi LOS dan NLOS. Pada keadaan LOS, sinyal merambat langsung melalui udara tanpa melewati suatu *obstacle* atau hambatan dari pengirim ke penerima.



Gambar 2.13 Propagasi NLOS

(Sumber : <http://repository.usu.ac.id/handle/123456789/21436>)

Pada kondisi NLOS seperti ditunjukkan Gambar 2.12, sinyal yang ditangkap di penerima (*receiver*) adalah sinyal yang telah mengalami proses refleksi, *scattering*, dan difraksi. Sinyal datang yang ditangkap penerima merupakan gabungan dari sinyal langsung, *multipath*, energi hamburan, dan sinyal propagasi yang telah terdifraksi. Sinyal ini mempunyai *delay* pola sebaran yang berbeda, redaman, polarisasi, dan kestabilan relatif dari sinyal langsung. Untuk mengetahui rugi-rugi propagasi pada kondisi NLOS dapat digunakan berbagai macam model propagasi NLOS antara lain model propagasi Okumura – Hata, COST-231 Hata, SUI (*Stanford University Interim*), ECC-33 dan Walfish – Ikegami. Model propagasi ini telah dikembangkan menyesuaikan karakter lingkungan RF (*Radio Frequency*) dan memperkirakan kuat sinyal RF. Model tersebut diperoleh dari pengukuran empiris di lapangan yang

digunakan untuk memprediksikan cakupan area dalam skala besar pada sistem komunikasi radio *broadband wireless access* (BWA). Pada Tugas Akhir ini akan digunakan model propagasi ECC-33 *Path Loss Model*.

2.4.1.1 ECC-33 Path Loss Model

Meskipun Hata-Okumura Model secara luas digunakan untuk band UHF, ketelitiannya masih dipertanyakan untuk frekuensi yang lebih tinggi. Teknik pendekatan yang berbeda kemudian dikembangkan oleh *Electronic Communication Committee* (ECC) dari pengukuran awal yang dilakukan oleh Okumura dan mengubahnya sehingga sangat mendekati sistem *Fixed Wireless Access* (FWA). Model *path loss* yang dikembangkan tersebut dikenal sebagai ECC-33 Model. *ECC-33 Path Loss Model* didefinisikan sebagai:

$$L_{propagasi}[dB] = A_{fs} + A_{bm} - G_t - G_r \quad (2-6)$$

dimana A_{fs} , A_{bm} , G_b , G_r berturut-turut adalah *free space attenuation*, *basic median path loss*, faktor *gain* ketinggian antenna pemancar (*Base Station*), dan faktor *gain* ketinggian antenna penerima. Faktor-faktor ini didefinisikan sebagai:

$$A_{fs} = 92,4 + 20\log(d) + 20\log(f) \quad (2-7)$$

$$A_{bm} = 20,41 + 9,38\log(d) + 7,894\log(f) + 9,56[\log(f)]^2 \quad (2-8)$$

$$G_t = \log(h_t/200)\{13,958 + 5,8[\log(d)]^2\} \quad (2-9)$$

$$G_r = [42,57 + 13,7\log(f)][\log(h_r) - 0,585] \quad (2-10)$$

dimana :

f = frekuensi (GHz)

d = jarak pemancar dan penerima (km)

h_t = ketinggian antenna pemancar (m)

h_r = ketinggian antenna penerima (m)

2.4.2 Perhitungan EIRP (*Effective Isotropic Radiated Power*)

EIRP merupakan besaran yang menyatakan kekuatan daya pancar suatu antenna di bumi, dapat dihitung dengan rumus:

$$EIRP = P_{tx} + G_{tx} - L_{tx} \quad (2-11)$$

dimana :

P_{tx} = daya pancar (dBm)

G_{tx} = penguatan antenna pemancar (dBi)

L_{tx} = rugi-rugi pada pemancar (dB)

2.4.3 Perhitungan RSL (*Receive Signal Level*)

RSL (*Receive Signal Level*) adalah level sinyal yang diterima di penerima dan nilainya harus lebih besar dari sensitivitas perangkat penerima ($RSL \geq R_{ss}$). Sensitivitas perangkat penerima merupakan kepekaan suatu perangkat pada sisi penerima yang dijadikan ukuran *threshold*. Nilai RSL dapat dihitung dengan:

$$RSL = EIRP - L_{propagasi} + Grx - Lrx \quad (2-12)$$

dimana :

$EIRP$ = *Effective Isotropic Radiated Power* (dBm)

$L_{propagasi}$ = rugi-rugi gelombang saat berpropagasi (dB)

Grx = penguatan antena penerima (dBi)

Lrx = rugi-rugi saluran penerima (dB)

2.4.4 Perhitungan Rss (*Receiver Sensitivity*)

Sensitivitas perangkat penerima merupakan kepekaan suatu perangkat pada sisi penerima yang dijadikan ukuran *threshold*. Perhitungan Rss dilakukan dengan menggunakan persamaan:

$$R_{ss}[dBm] = 10 \log(kTB) + SNR_{RX} + N_F \quad (2-13)$$

dimana

k = konstanta Boltzman, $1,38 \times 10^{-23}$ J/°K

T = suhu ruang, 290° K

B = *bandwidth* (Hz)

SNR_{RX} = *signal to noise ratio threshold* (dB)

N_F = *noise figure* (dB)

2.5 Parameter Performansi *Video Conference* pada Jaringan LTE

Untuk mengetahui performansi dari sistem LTE, mulai dari saat pengiriman di pengirim sampai dengan di penerima, dibutuhkan beberapa parameter yang akan digunakan dalam analisis. Beberapa parameter performansi yang akan dibahas antara lain *delay end to end*, probabilitas *packet loss* dan *throughput*.

2.5.1 *Delay end to end*

Delay end to end video conference merupakan jumlah *delay* CODEC aplikasi *video conference* dengan *delay* jaringan dimana aplikasi tersebut berjalan. Sehingga, total *delay end to end video conference* pada jaringan LTE menggunakan mode *duplex* TDD dapat dihitung dengan persamaan:

$$t_{\text{end-to-end}} = t_{\text{CODEC}} + t_{\text{tot}} \quad (2-14)$$

dengan :

$t_{\text{end-to-end}}$: *delay end to end video conference* (s)

t_{tot} : *delay jaringan LTE* (s)

t_{CODEC} : *delay CODEC video conference* (s)

a. Delay CODEC

Delay CODEC ini terjadi pada sisi *encoder* terdiri dari waktu untuk mengakumulasi *sample* suara ke dalam *frame* suara, waktu untuk mengkompresi paket suara, waktu untuk memuat *frame* suara ke dalam paket dan mentransfer paket tersebut ke jaringan *transport* dan *delay hardware* yang bersifat tetap. Atau waktu yang dibutuhkan untuk mengkompresi satu blok *sample* PCM.

Coding delay adalah waktu yang dibutuhkan untuk mengkonversi sinyal non digital menjadi pola bit sinyal digital. *Delay* ini tergantung pada algoritma *coding* yang digunakan pada perangkat lunak atau perangkat keras yang digunakan. *Coding delay* juga dapat terjadi pada aplikasi-aplikasi yang tidak membutuhkan proses transformasi sinyal, misalkan pada proses kompresi data.

Delay CODEC pada aplikasi *video conference* dapat dihitung dengan persamaan sebagai berikut:

$$t_{\text{CODEC}} = t_a + t_v \quad (2-15)$$

dengan:

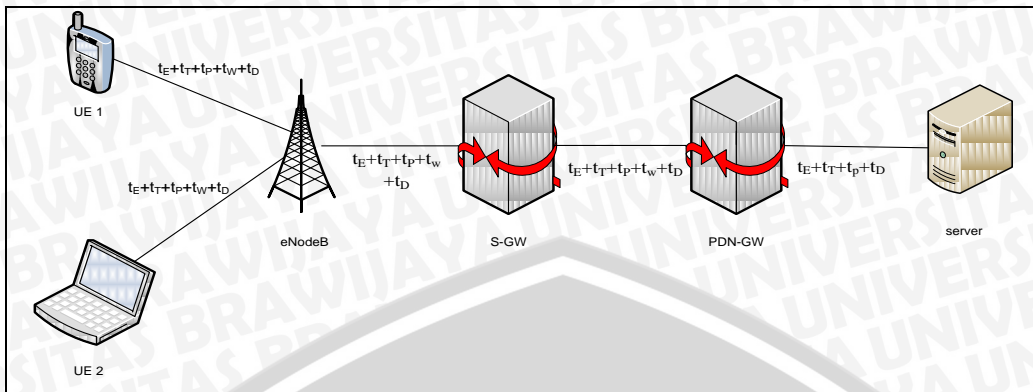
t_{CODEC} : *delay CODEC aplikasi video conference* (ms)

t_a : *delay CODEC audio G.711* (ms)

t_v : *delay CODEC video H.264* (ms)



b. *Delay end to end video conference pada jaringan LTE*



Gambar 2.14 Delay end to end pada jaringan LTE

Sumber : Perencanaan

Delay end to end berdasarkan Gambar 2.13 dihitung dengan menggunakan persamaan:

$$t_{TOT} = t_{proc} + t_T + t_P + t_W \quad (2-16)$$

dengan:

t_{TOT} : *delay end to end* pada jaringan LTE (s)

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

t_T : *delay transmisi* (s)

t_P : *delay propagasi* (s)

t_W : *delay antrian* (s)

- **Delay Proses**

Delay proses merupakan waktu yang dibutuhkan untuk memproses paket data dan untuk menentukan kemana data tersebut akan diteruskan. *Delay proses* pada jaringan LTE meliputi *delay enkapsulasi* dan *delay dekapsulasi*.

Delay enkapsulasi adalah waktu yang dibutuhkan untuk menambahkan keseluruhan *header* pada sebuah paket sehingga paket data tersebut dapat tepat sampai ke tujuan. Sedangkan *delay dekapsulasi* adalah waktu yang dibutuhkan untuk melepaskan keseluruhan *header* dari sebuah paket. Besarnya *delay dekapsulasi* dan *delay enkapsulasi* dapat diperoleh dengan menggunakan persamaan di bawah ini (Onno W. Purbo, et al., 2001: 24):

$$t_E = \frac{W_{frame} - L}{C} \times 8 \quad (2-17)$$

$$t_D = \frac{W_{frame} - L}{C} \times 8 \quad (2-18)$$

dengan:

- t_E = *delay* enkapsulasi (s)
- t_D = *delay* dekapsulasi (s)
- W_{frame} = panjang *frame* (byte)
- L = panjang paket data di *node* (byte)
- C = kecepatan pemrosesan data (bps)

Delay enkapsulasi terjadi pada *server*, PDN-GW, S-GW, dan eNodeB. Besarnya *delay* enkapsulasi didapatkan dengan menggunakan persamaan:

$$t_E = t_{E1} + t_{E2} + t_{E3} + t_{E4} \quad (2-19)$$

dengan:

- t_E = *delay* enkapsulasi total
- t_{E1} = *delay* enkapsulasi pada *server*
- t_{E2} = *delay* enkapsulasi pada PDN-GW
- t_{E3} = *delay* enkapsulasi pada S-GW
- t_{E4} = *delay* enkapsulasi pada eNodeB

Sedangkan *delay* dekapsulasi terjadi di setiap *node* pada jaringan LTE, yaitu PDN-GW, S-GW, dan eNodeB, dan UE. Besarnya *delay* dekapsulasi dapat dihitung menggunakan persamaan:

$$t_D = t_{D1} + t_{D2} + t_{D3} + t_{D4} \quad (2-20)$$

dengan:

- t_D = *delay* dekapsulasi total
- t_{D1} = *delay* dekapsulasi pada PDN-GW
- t_{D2} = *delay* dekapsulasi pada S-GW
- t_{D3} = *delay* dekapsulasi pada eNodeB
- t_{D4} = *delay* dekapsulasi pada UE

Nilai *delay* proses pada jaringan LTE untuk *video conference* dapat diperoleh dengan menggunakan persamaan:

$$t_{\text{proc}} = t_E + t_D \quad (2-21)$$

dengan:

- t_{proc} = *delay* proses
- t_E = *delay* enkapsulasi total
- t_D = *delay* dekapsulasi total

➤ **Server**

Pada server terjadi proses enkapsulasi sebelum paket data ditransmisikan ke PDN-GW. Paket data *video conference* diberi penambahan header IP. Jumlah total *frame* pada *server* yang dapat dikirimkan ke PDN-GW sesuai dengan persamaan:

$$W_{frame\ server} = W_{VC-size} + H_{UDP} + H_{IP} + H_{ethernet} + FCS \quad (2-22)$$

Apabila panjang paket *video conference* melebihi *Maximum Transfer Unit* (MTU) *Ethernet* sebesar 1500 *byte*, maka paket *video conference* akan mengalami fragmentasi menjadi beberapa buah *frame* sesuai dengan persamaan:

$$N_{frame\ ethernet} = \frac{W_{VC-size}}{MTU\ ethernet} \quad (2-23)$$

dengan:

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

$$MTU_{ethernet} = MTU\ Ethernet\ (1500\ byte)$$

Sehingga jika panjang paket *video conference* melebihi *Maximum Transfer Unit* (MTU) *Ethernet*, jumlah total *frame* pada *server* yang dapat dikirimkan ke PDN-GW sesuai dengan persamaan:

$$W_{frame\ server} = W_{VC-size} + [N_{frame\ Ethernet} \times (H_{UDP} + H_{IP} + H_{ethernet} + FCS)] \quad (2-24)$$

Dalam skripsi ini, pada *server* digunakan standar *interface Gigabit Ethernet* dengan kecepatan 1 Gbps. Sehingga *delay* enkapsulasi pada *server* didapatkan dengan persamaan berikut:

$$t_{E1} = \frac{W_{frame\ server} - W_{VC-size}}{C_{server}} \quad (2-25)$$

dengan:

$$t_{E1} = \text{delay enkapsulasi pada } server\ (s)$$

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

$$C_{server} = \text{kecepatan pemrosesan data pada } server\ (bps)$$

➤ **Packet Data Network Gateway (PDN-GW)**

Pada PDN-GW, paket data yang diterima dari *server* mengalami proses dekapsulasi. Kemudian, sebelum paket data aplikasi *video conference* ditransmisikan ke S-GW, dilakukan proses enkapsulasi terlebih dahulu.

Pada proses dekapsulasi, paket data aplikasi *video conference* didapatkan sesuai dengan persamaan:

$$W_{VC\ PDN-GW} = W_{frame\ server} - [N_{frame\ ethernet} \times (H_{Ethernet} + H_{UDP} + H_{IP} + FCS)] \quad (2-26)$$

Dengan mengasumsikan PDN-GW menggunakan standar *Fast Ethernet* dengan kecepatan 100 Mbps, maka besarnya *delay* dekapsulasi pada PDN-GW didapatkan sesuai dengan persamaan:

$$t_{D1} = \frac{W_{frameserver} - W_{VCPDN-GW}}{C_{PDN-GW}} \times 8 \quad (2-27)$$

dengan:

t_{D1} = *delay* dekapsulasi pada PDN-GW (s)

$W_{VC PDN-GW}$ = panjang paket data video *conference* di PDN-GW (byte)

C_{PDN-GW} = kecepatan pemrosesan data di PDN-GW (bps)

Besar MSS dapat diperoleh sesuai dengan persamaan berikut:

$$MSS = MTU - H_{GTP} - H_{UDP} - H_{IP} \quad (2-28)$$

Selanjutnya, paket data yang melebihi MSS akan disegmentasi menggunakan persamaan berikut:

$$N_{datagram} = \frac{W_{VCPDN-GW}}{MSS} \quad (2-29)$$

Kemudian, paket data aplikasi video *conference* dienkapsulasi dengan penambahan *header* GTP (*GPRS Tunneling Protocol*), UDP (*User Datagram Protocol*), dan IP (*Internet Protocol*) sesuai dengan persamaan berikut:

$$W_{datagram} = W_{VC PDN-GW} + [N_{datagram} \times (H_{GTP} + H_{UDP} + H_{IP})] \quad (2-30)$$

Karena panjang datagram IP melebihi *Maximum Transfer Unit* (MTU) *Ethernet* (1500 byte), maka datagram IP akan mengalami fragmentasi menjadi beberapa buah frame sesuai dengan persamaan:

$$N_{frameEthernet} = \frac{W_{datagram}}{MTU_{ethernet}} \quad (2-31)$$

Sehingga, jumlah total *frame* pada PDN-GW yang dapat dikirimkan ke S-GW sesuai dengan persamaan:

$$W_{frame PDN-GW} = W_{datagram} + [N_{frame Ethernet} \times (H_{ethernet} + FCS)] \quad (2-32)$$

Sehingga *delay* enkapsulasi pada PDN-GW didapatkan dengan persamaan:

$$t_{E2} = \frac{W_{framePDN-GW} - W_{VCPDN-GW}}{C_{PDN-GW}} \times 8 \quad (2-33)$$

dengan:

t_{E2} = *delay* enkapsulasi pada PDN-GW (s)

$W_{frame PDN-GW}$ = panjang *frame* di PDN-GW (byte)

C_{PDN-GW} = kecepatan pemrosesan data di PDN-GW (bps)

➤ **Serving Gateway (S-GW)**

Pada S-GW, paket data yang diterima dari PDN-GW mengalami proses dekapsulasi. Kemudian, sebelum paket data aplikasi *video conference* ditransmisikan ke eNodeB, dilakukan proses enkapsulasi terlebih dahulu.

Pada proses dekapsulasi, paket data aplikasi *video conference* didapatkan sesuai dengan persamaan:

$$W_{VCS-GW} = W_{frame\ PDN-GW} - [N_{datagram} \times (H_{GTP} + H_{UDP} + H_{IP})] - [N_{frame\ ethernet} \times (H_{Ethernet} + FCS)] \quad (2-34)$$

Dengan mengasumsikan S-GW menggunakan standar *Fast Ethernet* dengan kecepatan 100 Mbps, maka besarnya *delay* dekapsulasi pada S-GW didapatkan sesuai dengan persamaan:

$$t_{D2} = \frac{W_{frame\ PDN-GW} - W_{VCS-GW}}{C_{S-GW}} \times 8 \quad (2-35)$$

dengan:

t_{D2} = *delay* dekapsulasi pada S-GW (s)

W_{VCS-GW} = panjang paket data *video conference* di S-GW (byte)

C_{S-GW} = kecepatan pemrosesan data di S-GW (bps)

Besar MSS dapat diperoleh sesuai dengan persamaan berikut:

$$MSS = MTU - H_{GTP} - H_{UDP} - H_{IP} \quad (2-36)$$

Selanjutnya, paket data yang melebihi MSS akan disegmentasi menggunakan persamaan berikut

$$N_{datagram} = \frac{W_{VCS-GW}}{MSS} \quad (2-37)$$

Kemudian, paket data aplikasi *video conference* dienkapsulasi dengan penambahan *header* GTP (*GPRS Tunneling Protocol*), UDP (*User Datagram Protocol*), dan IP (*Internet Protocol*) sesuai dengan persamaan berikut:

$$W_{datagram} = W_{VCS-GW} + [N_{datagram} \times (H_{GTP} + H_{UDP} + H_{IP})] \quad (2-38)$$

Karena panjang datagram IP melebihi *Maximum Transfer Unit (MTU) Ethernet* (1500 *byte*), maka datagram IP akan mengalami fragmentasi menjadi beberapa buah frame sesuai dengan persamaan:

$$N_{frameEthernet} = \frac{W_{datagram}}{MTU_{ethernet}} \quad (2-39)$$

Sehingga jumlah total *frame* pada S-GW yang dapat dikirimkan ke eNodeB sesuai dengan persamaan:

$$W_{frame\ S-GW} = W_{datagram} + [N_{frame\ Ethernet} \times (H_{ethernet} + FCS)] \quad (2-40)$$

Sehingga *delay* enkapsulasi pada S-GW didapatkan dengan persamaan:

$$t_{E3} = \frac{W_{frameS-GW} - W_{VCS-GW}}{C_{S-GW}} \times 8 \quad (2-41)$$

dengan:

t_{E3} = *delay* enkapsulasi pada S-GW (s)

$W_{frame\ S-GW}$ = panjang *frame* di S-GW (byte)

C_{S-GW} = kecepatan pemrosesan data di S-GW (bps)

➤ **Evolved Node B (eNodeB)**

Pada eNodeB, paket data yang diterima dari S-GW mengalami proses dekapsulasi. Kemudian, sebelum paket data aplikasi video *conference* ditransmisikan ke UE, dilakukan proses enkapsulasi terlebih dahulu.

Pada proses dekapsulasi, paket data aplikasi video *conference* didapatkan sesuai dengan persamaan:

$$W_{VC\ eNodeB} = W_{frame\ S-GW} - [N_{datagram} \times (H_{GTP} + H_{UDP} + H_{IP})] - [N_{frame\ ethernet} \times (H_{ethernet} + FCS)] \quad (2-42)$$

eNodeB menggunakan *interface* STM-1 (*Synchronous Transport Module-1*) dengan kecepatan 155,52 Mbps. Maka besarnya *delay* dekapsulasi didapatkan dengan menggunakan persamaan:

$$t_{D3} = \frac{W_{frameS-GW} - W_{VCeNodeB}}{C_{eNodeB}} \times 8 \quad (2-43)$$

dengan:

t_{D3} = *delay* dekapsulasi pada eNodeB (s)

$W_{VC\ eNodeB}$ = panjang paket data *video conference* di eNodeB (byte)

C_{eNodeB} = kecepatan pemrosesan data di eNodeB (bps)

Pada layer PDCP (*Packet Data Convergence Protocol*), paket datagram IP *video conference* dienkapsulasi dengan penambahan *header* sebesar 1 *byte*, yaitu *header* PDU. Banyaknya paket IP pada eNodeB dapat diperoleh sebagai berikut:

$$N_{datagram} = \frac{W_{VCeNodeB}}{MTU} \quad (2-44)$$

Pada layer PDCP, panjang frame dapat dihitung dengan menggunakan persamaan:

$$W_{frame\ PDCP} = W_{VC\ eNodeB} + [N_{datagram} \times H_{PDCP}] \quad (2-45)$$

Selanjutnya pada layer RLC (*Radio Link Control*), *frame* PDCP disegmentasi menjadi RLC PDU *fixed size* sebesar 40 *byte*. Sehingga jumlah *frame* RLC sesuai dengan persamaan

$$N_{frameRLC} = \frac{W_{framePDCP}}{40byte} \quad (2-46)$$

dengan:

$N_{frame\ RLC}$ = jumlah *frame* RLC

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

Setiap *frame* selanjutnya diberi *header* sebesar 2 *byte*, sehingga panjang setiap *frame* RLC sebesar 42 *byte*. Panjang *frame* RLC total yang siap diteruskan ke layer MAC (*Medium Access Control*) sesuai dengan persamaan:

$$W_{frame\ RLC\ total} = N_{frame\ RLC} \times W_{frame\ RLC} \quad (2-47)$$

Pada layer MAC, RLC disegmentasi menjadi MAC SDU (*Service Data Unit*) sebesar 42 *byte*. Jumlah *frame* SDU sesuai dengan persamaan:

$$N_{frameMAC} = \frac{W_{frameRLCtotal}}{42byte} \quad (2-48)$$

dengan:

$N_{frame\ MAC}$ = jumlah *frame* MAC

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

Sedangkan panjang *frame* MAC dihitung dengan menggunakan persamaan:

$$W_{frameMAC} = H_{MAC} + 42\ byte \quad (2-49)$$

Panjang *frame* yang siap ditransmisikan menuju UE merupakan panjang *frame* MAC total sesuai dengan persamaan:

$$W_{frame\ eNodeB} = N_{frame\ MAC} \times W_{frame\ MAC} \quad (2-50)$$



Sehingga besar *delay* dekapsulasi yang terjadi pada eNodeB didapatkan dengan menggunakan persamaan:

$$t_{E4} = \frac{W_{frameeNodeB} - W_{VSeNodeB}}{C_{eNodeB}} \times 8 \quad (2-51)$$

dengan:

- t_{E4} = *delay* enkapsulasi pada eNodeB (s)
- $W_{VC\ eNodeB}$ = panjang paket data video *conference* di eNodeB (byte)
- C_{eNodeB} = kecepatan pemrosesan data di eNodeB (bps)

➤ **User Equipment (UE)**

Pada UE, paket data yang diterima dari eNodeB mengalami proses dekapsulasi. Proses dekapsulasi pada UE sesuai dengan persamaan:

$$W_{VC\ UE} = W_{frame\ eNodeB} - (N_{datagram} \times H_{PDCP}) - (N_{frame\ RLC} \times H_{RLC}) - (N_{frame\ MAC} \times H_{MAC}) \quad (2-52)$$

Dengan mengasumsikan *data rate* maksimum sebesar 3,6 Mbps, maka besarnya *delay* dekapsulasi yang terjadi di UE didapatkan dengan persamaan:

$$t_{D4} = \frac{W_{frameeNodeB} - W_{VCUE}}{C_{UE}} \times 8 \quad (2-53)$$

dengan:

- t_{D4} = *delay* dekapsulasi pada UE (s)
- $W_{VC\ UE}$ = panjang paket data video *conference* pada UE (byte)
- C_{UE} = kecepatan pemrosesan data pada UE (bps)

• **Delay Transmisi**

Delay transmisi adalah waktu yang dibutuhkan untuk meletakkan sebuah paket data ke media transmisi. Dipengaruhi ukuran paket data dan kecepatan transmisi. *Delay* transmisi dapat ditentukan dengan persamaan (Mischa Schwartz, 1987 : 132):

$$t_T = \frac{W}{C} \quad (2-54)$$

Sehingga *delay* transmisi total ditentukan dengan persamaan:

$$t_{T\ tot} = t_{T1} + t_{T2} + t_{T3} + t_{T4} \quad (2-55)$$

dimana:

- $t_{T\ tot}$ = *delay* transmisi total (s)



- W = panjang *frame* pada *node* (bit)
- C = kecepatan transmisi (bps)
- t_{T1} = *delay* transmisi pada *server* – PDN-GW (s)
- t_{T2} = *delay* transmisi pada PDN-GW – S-GW (s)
- t_{T3} = *delay* transmisi pada S-GW – eNodeB (s)
- t_{T4} = *delay* transmisi pada eNodeB -UE (s)

- **Delay Propagasi**

Delay propagasi adalah waktu yang dibutuhkan untuk merambatkan paket data melalui media transmisi dari *server* ke UE. *Delay* propagasi ditentukan dengan persamaan (Forouzan, Behrouz A, 2000 : 215):

$$t_p = \frac{N_{frame} \times R}{v} \quad (2-56)$$

Sehingga *delay* propagasi total ditentukan dengan persamaan:

$$t_{p\ tot} = t_{p1} + t_{p2} + t_{p3} + t_{p4} \quad (2-57)$$

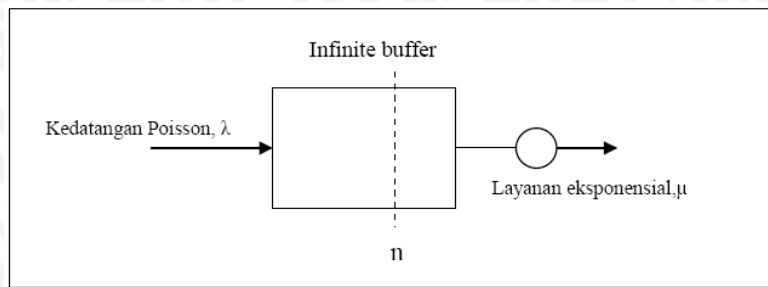
dimana:

- $t_{p\ tot}$ = *delay* propagasi (s)
- N_{frame} = jumlah *frame* pada *node*
- R = jarak antar *node* (m)
- v = cepat rambat gelombang di cahaya = 3×10^8 (m/s)
- t_{p1} = *delay* propagasi pada *server* – PDN-GW (s)
- t_{p2} = *delay* propagasi pada PDN-GW – S-GW (s)
- t_{p3} = *delay* propagasi pada S-GW - eNodeB (s)
- t_{p4} = *delay* propagasi pada eNodeB -UE (s)

- **Delay Antrian**

Delay antrian adalah waktu dimana paket data berada dalam antrian untuk ditransmisikan. Selama waktu ini, paket data menunggu hingga paket yang lain selesai ditransmisikan. *Delay* antrian dapat dihitung dengan menggunakan model antrian M/M/1. M pertama menunjukkan kedatangan *Poisson*, M kedua berarti distribusi waktu pelayanan eksponensial, dan 1 menunjukkan 1 menunjukkan jumlah server yang akan melayani pelanggan.

Disiplin antrian yang digunakan adalah FIFO (*First In First Out*). Bentuk model antrian M/M/1 dapat dilihat pada Gambar 2.14.



Gambar 2.15 Model antrian M/M/1
(Sumber: Mischa Schwartz, 1987:31)

Delay antrian terjadi pada setiap *node* pada jaringan LTE, yaitu PDN-GW, S-GW, eNodeB, dan UE. Besarnya *delay* antrian pada *node* ditentukan dengan persamaan (Mischa Schwartz, 1987 : 42):

$$t_w = t_{queue} + t_{serv} \tag{2-58}$$

dengan:

t_w = *delay* antrian (s)

t_{queue} = waktu tunggu paket pada *node* (s)

t_{serv} = waktu rata-rata pelayanan *node* (s)

Perhitungan waktu rata-rata pelayanan antrian data di *node* (t_{serv}) dihitung dengan menggunakan persamaan (Mischa Schwartz, 1987 : 23):

$$t_{serv} = \frac{L}{C} \tag{2-59}$$

dengan:

μ = kecepatan pelayanan *node* (bps)

L = panjang paket data di *node* (bit)

C = kecepatan transmisi pada *node* (bps)

Sedangkan waktu tunggu paket pada *router* (t_{queue}) dihitung dengan menggunakan persamaan (Mischa Schwartz, 1987 : 42):

$$t_{queue} = \frac{\lambda}{\mu^2(1-\rho)} \tag{2-60}$$

Untuk performansi sistem antrian, ditunjukkan dalam bentuk ρ (faktor utilisasi), yang nilainya diasumsikan berubah-ubah dengan kenaikan tertentu. Faktor utilisasi bernilai lebih besar dari 0 dan lebih kecil dari 1. Hal ini disebabkan karena jika $\rho > 1$ berarti rata-rata kedatangan melampaui rata-rata pelayanan atau server tidak dapat



menjaga rata-rata kedatangan dan panjang antrian yang bertambah tanpa batas. Besarnya faktor utilisasi dapat dihitung dengan menggunakan persamaan:

$$\rho = \frac{\lambda}{\mu} \implies \lambda = \mu\rho \quad (2-61)$$

Sehingga dari persamaan 2.59 dan 2.60 *delay* antrian dapat ditentukan dengan persamaan:

$$t_w = \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu} \quad (2-62)$$

dengan:

- t_{queue} = waktu tunggu paket pada *node* (s)
- λ = jumlah kedatangan user pada *node*
- ρ = faktor utilitas ($0 < \rho < 1$)
- μ = jumlah pelayanan pada *node*

Sehingga *delay* antrian total ditentukan dengan persamaan:

$$t_{w \text{ tot}} = t_{w1} + t_{w2} + t_{w3} + t_{w4} \quad (2-63)$$

dengan:

- $t_{w \text{ tot}}$ = *delay* antrian total (s)
- t_{w1} = *delay* antrian pada PDN-GW (s)
- t_{w2} = *delay* antrian pada S-GW (s)
- t_{w3} = *delay* antrian pada eNodeB (s)
- t_{w4} = *delay* antrian pada UE (s)

2.5.2 Probabilitas *packet loss video conference* pada jaringan LTE

Packet loss adalah jumlah paket yang hilang dibandingkan dengan paket yang diterima benar. Umumnya perangkat jaringan memiliki *buffer* untuk menampung data yang diterima. Jika terjadi tabrakan yang cukup banyak, *buffer* akan penuh, dan data baru tidak dapat diterima. Paket yang hilang ini harus ditransmisi ulang, yang akan membutuhkan waktu tambahan.

Probabilitas *packet loss* total performansi aplikasi *video conference* pada suatu jaringan, ditentukan berdasarkan pada *packet loss* aplikasi *video conference* dan *packet loss* jaringan.

$$\rho_{\text{tot}} = 1 - [(1 - \rho_{\text{LTE}})(1 - \rho_{\text{VC}})] \quad (2-64)$$

dengan:

- ρ_{tot} = *packet loss* total

P_{LTE} = probabilitas *packet loss* pada jaringan LTE

P_{VC} = probabilitas *packet loss* aplikasi *video conference*

Prosentase *packet loss* ditentukan dengan persamaan:

$$\text{packet loss (\%)} = \frac{N_{\text{packet loss}}}{N_{\text{paket}} + N_{\text{packet loss}}} \times 100\% \quad (2-65)$$

dengan:

$N_{\text{packet loss}}$ = jumlah paket yang hilang

N_{paket} = jumlah paket yang diterima dengan benar

BER (*bit error rate*) atau dengan sebutan lain probabilitas *error* bit merupakan nilai ukur kualitas sinyal yang diterima untuk sistem transmisi data digital. BER juga dapat didefinisikan sebagai perbandingan jumlah bit *error* terhadap total bit yang diterima. Nilai BER untuk modulasi QPSK pada jaringan LTE dapat dihitung dengan persamaan (Andrea Goldsmith, 2005:167):

$$P_{be} = Q\left(\sqrt{2 \frac{E_b}{N_o}}\right) \quad (2-66)$$

dengan:

P_{be} = BER LTE pada saat transmisi (tanpa satuan)

$\frac{E_b}{N_o}$ = rasio energi *bit* terhadap kerapatan *noise* pada saat transmisi (dB)

Nilai Q dapat diperoleh dengan distribusi Gaussian menggunakan persamaan (John G. Proakis, 2000: 40):

$$Q(x) = \frac{1}{2} \operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right) \quad (2-67)$$

BER pada sistem dengan teknik modulasi M-QAM dapat dihitung dengan persamaan : [E.Glatz, 2001]

$$P_{be} = \frac{4}{K} \left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{3K E_b}{M-1 N_o}}\right) \quad (2-68)$$

dimana, $\operatorname{erfc}(x) \approx \frac{1}{\sqrt{\pi x}} e^{-x^2}$

dengan :

P_{be} = probabilitas bit *error*

K = *bits/symbol*

M = jumlah simbol sesuai teknik modulasi

$\frac{E_b}{N_o}$ = rasio *energy bit* terhadap *noise* (dB)

$$\operatorname{erfc}(x) \approx \frac{1}{\sqrt{\pi x}} \cdot e^{-x^2} \quad (2-69)$$

dengan:

erfc = *error function complementary*

E_b/N_o adalah suatu parameter yang berhubungan dengan SNR yang biasanya digunakan untuk menentukan laju data digital dan sebagai ukuran mutu standar untuk kinerja sistem komunikasi digital. Hubungan SNR dengan E_b/N_o ditunjukkan dalam persamaan:

$$\frac{E_b}{N_o} \text{ (dB)} = \frac{S}{N} + 10 \log \frac{B}{R} \quad (2-70)$$

dengan :

$\frac{E_b}{N_o}$ = rasio *energy bit* terhadap *noise* (dB)

SNR = *signal to noise ratio* (dB)

B = *bandwidth* (Hz)

R = laju data (bps)

Besarnya pengaruh redaman sinyal terhadap sinyal yang ditransmisikan dapat dinyatakan dengan perbandingan antara sinyal dengan *noise* (SNR) yang dinyatakan dalam persamaan berikut (E. Glatz, 1999):

$$SNR_{(dB)} = RSL(dBm) - N_o(dBm) \quad (2-71)$$

dengan :

SNR = *signal to noise ratio* (dB)

RSL = daya pada penerima (dBm)

N_o = daya *noise* saluran transmisi (dBm)

Sedangkan untuk perhitungan daya *noise* dinyatakan dalam persamaan berikut (E. Glatz, 1999):

$$N_{(dBm)} = 10 \log k \cdot T + 10 \log B + NF \quad (2-72)$$

dengan :

N = daya *noise* saluran transmisi (dBm)

k = konstanta Boltzman ($1,38 \times 10^{-23}$ J/K)

T = suhu *absolute* (300° K)

NF = *noise figure* (11,2 dB)

B = *bandwidth* (Hz)

2.5.3 Throughput

Throughput merupakan salah satu parameter yang menunjukkan kinerja dari suatu komunikasi data, yaitu menunjukkan jumlah data yang diterima dengan benar pada penerima setelah melewati kanal komunikasi. Nilai *throughput* dihitung pada proses pentransmisi sinyal dari *server* hingga informasi diterima pada UE. *Throughput* juga dapat diartikan sebagai kecepatan maksimal jaringan saat tidak ada data yang hilang pada saat pentransmisi, dijelaskan dalam persamaan berikut :

$$\lambda = \frac{1-\rho}{t_t[1+(\alpha-1)\rho]} \quad (2-73)$$

Parameter α dapat dihitung dengan persamaan : [Mischa Schwartz, 1987:129]

$$t_{out} = 2t_p + 2t_t \quad (2-74)$$

$$\alpha = 1 + \frac{t_{out}}{t_t} = 1 + \frac{2t_p + 2t_t}{t_t} = 3 + \frac{2t_p}{t_t} \quad (2-75)$$

dengan :

λ = *throughput* (Mbps)

ρ = probabilitas paket salah

t_t = waktu untuk mentransmisikan sebuah paket (detik)

α = konstanta propagasi