BAB II
DASAR TEORI

2.1 *ROUTING [1]*

Routing adalah suatu protokol yang digunakan untuk mendapatkan *route* dari satu jaringan ke jaringan yang lain, informasi *route* ini secara dinamis dapat diberikan ke *router* yang lain ataupun dapat diberikan secara statis ke *router* lain.

Algoritma *routing* merupakan bagian perangkat lunak dan lapisan *network* yang bertanggung jawab terhadap saluran keluaran bagi paket masuk dan harus ditransmisikan. Proses *routing* paket data diperlukan syarat berikut:

- Alamat tujuan yang jelas dan memilih jalur-jalur terbaik dan tercepat.
- Mengidentifikasi informasi dan sumber yaitu *router* mempelajari dan mana informasi berasal, jalur-jalur yang dipilih selanjutnya.
- Menentukan jalur-jalur yang mungkin dilewati yaitu lalu lintas yang dapat ditempuh untuk sampai tujuan.
- Mengatur dan mengkonfirmasilkan informasi *routing* yaitu jalur yang akan digunakan bisa terpercaya atau tidak.

Routing ideal berkriteria ketepatan, ketangguhan, stabilitas, dan optimalisasi. Tujuan utama *routing* adalah *router* tidak mempelajari jalur-jalur terhubung langsung dengannya,

tetapi mengatur bagaimana meneruskan paket data ke jalur yang terhubung tidak langsung.

Routing harus mampu mengatasi perubahan topologi jaringan serta lalu lintas jalur tanpa pembatalan proses pada *host*, selain itu jaringan tidak memerlukan *reboot* jika *router* mengalami tabrakan.

2.1.1 Routing Statis [1]

Routing Statis adalah jenis *routing* yang dikonfigurasi *admin/user* untuk melakukan konfigurasi informasi tentang jarigan yang dituju secara manual. Ciri-ciri dari *routing statis* ini adalah: jalur spesifik jaringan yang akan digunakan ditentukan oleh *admin/user*, selain penetuan jalur yng manual penentuan tabel *routing* juga dilakukan secara manual, *routing statis* biasanya digunakan dalam jaringan yang kecil. Cara kerja *routing* statis dapat dibagi menjadi tiga bagian, yaitu :

- Administator jaringan yang mengkonfirmasi *router*
- *Router* melakukan *routing* berdasarkan informasi dalam tabel *routing*.
- *Routing* statis digunakan untuk melewatkan paket data.

Routing statis memiliki beberap keuntungan seperti :

- Pemeliharan *bandwidth network* karena peng-update-an informasi *router* membutuhkan *broadcasts* yang terus menerus.

- Keamanan *network* karena *statis routing* hanya mengandung informasi yang telah dimasukkan secara manual.

Selain itu, *routing statis* juga memiliki kerugian seperti:

- Tidak ada tolerasi kesalahan. Jika suatu *router* down, maka *statis* tidak akan memperbaharui informasi dan tidak akan menginformasikan ke *router* yang lain.
- Pengembangan *network*. Jika suatu *network* ditambah atau dipindahkan maka *statis routing* harus diperbaharui oleh administrator

2.1.2 ***Routing*** **Dinamis**

Routing dinamis merupakan jenis *routing* yang dapat bekerja secara otomatis sesuai dengan kondisi yang diinginkan dengan mengacu pada parameter tertentu sesuai dengan protokolnya. Biasanya *routing* dinamis ini diterapkan pada sebuah *Personal Computer* (PC) yang dimana PC tersebut berfungsi sebagai *router* dan dibutuhkan *router* yang lain yang sama-sama menggunakan *routing* ini. Pada dymanic *routing* memiliki beberapa jenis *routing* protokol.[10]

2.2 Algoritma Vektor Jarak (*Distance Vector Algorithm*)

Algoritma vektor jaringan mengirim informasi secara periodik dan tabel *routing* salah satu *router* ke tabel lain.

Algoritma ini tidak memperbolehkan *router* untuk mengetahui topologi seluruh interkoneksi jaringan.

Perubahan topologi jaringan yang terjadi adalah perubahan yang harus dikomunikasikan antar *router*. Informasi pembaharuan akan disesuaikan serta dibandingakan dengan tabel miliknya, juga memberi hak pada setiap *router* untuk menjaga tabel sendiri untuk menghasilkan jalur terbaik yang bisa dijangkau ke tujuan. Perubahan pengukuran ditulis pada tabel *routing* dan diperbaharui oleh *router* itu sendiri.

Setiap *router* dapat mengatur sebuah tabel *routing* yang dirangking oleh *router* itu sendiri berdasarkan informasi mengenai perhitungan jarak atau penggunaan antarmuka dan disimpan pada setiap *router* dan *subnet.*

Ciri protokol vektor jarak adalah menemukan jalur terbaik ke tujuan yang tersambung tidak langsung berdasar *matric* akumulasi *router neighbors*, oleh karena itu pengetahuan diperoleh dari informasi *neighbors*nya. *Router* menambah biaya pengantaran ke *router neighbors* agar jarak ke tujuan dilaporkan oleh *router* sebelahnya sehingga menghasilkan ukuran *matric* baru.

2.2.1 Algoritma Keadaan *Link* (*Link-state Algorithm*)

Algoritma keadaan *link* prosesnya memakai sistem perencanaan secara hirarki. *Router* yang bekerja pada suatu

wilayah harus mengetahui detail aturan menentukan jalur paket ke tujuan pada wilayahnya, akan tetapi sebuah *router* tidak diperbolehkan mengetahui sturktur internal *router* wilayah lain.

Protokol *routing* yang menggunakan konsep link state akan bekerja membuat tabel *routing* menurut perhitungannya sendiri dan tidak tergantung perhitungan *router* yang lain. Beberapa hal yang harus Anda pertimbangkan dalam menggunakan konsep ini adalah : Kemampuan processor yang lebih tinggi, karena perhitungannya menggunakan algoritma SPF (*Short Path First*), memori yang besar untuk menampung paket link state, bandwidth yang sedikit lebih besar untuk melakukan proses penerimaan, penyalinan, dan pengiriman paket link state.

Protokol-protokol yang bekerja menggunakan konsep link state antara lain OSPF (*Open Short Path First*) yang biasa digunakan dalam sebuah jaringan besar.

2.3 KONSEP *ROUTING* PROTOKOL [4]

Routing protokol adalah komunikasi antara *router*-*router*, protokol *routing* juga mengijinkan *router* untuk sharing informasi tentang jaringan dan koneksi antar *router*. *Router* menggunakan informasi ini untuk membangun dan memperbaiki tabel *routing*nya.

Router adalah sebuah alat yang mengirimkan paket data melalui sebuah jaringan atau internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai *routing*. *Router* sering

digunakan untuk menghubungkan beberapa jaringan. Baik jaringan yang sama maupun berbeda. *Router* juga digunakan untuk membagi jaringan besar menjadi beberapa buah *subnetwork* (*network-network* kecil). Informasi yang dibutuhkan *router* dalam melakukan *routing* yaitu:

1. Alamat tujuan/ *destination address*
2. Mengenal sumber informasi
3. Menemukan *route*
4. Pemilihan *route*
5. Menjaga informasi *routing*

Routing melibatkan dua aktifitas dasar, yaitu menentukan *path routing* yang paling optimal dan membawa paket informasi melalui suatu jaringan. Alur yang paling optimal diperoleh dari hasil penelusuran algoritma *routing*. Untuk membantu proses penentuan alur, algoritma *routing* menginisialisasi dan memelihara tabel *routing*, yang berisi informasi *routing*. Informasi *routing* bervariasi tergantung pada algoritma *routing* yang digunakan. Informasi *routing* tersbebut merupakan hasil pengukuran standar tertentu yang disebut *metric*.

2.4 *OPEN SHORTEST PATH FIRST* (OSPF)

OSPF bekerja berdasarkan algoritma *Shortest Path First* yang dikembangkan berdasarkan algoritma *Dijkstra*. Sebagai *Interior Gateway protokol* (IGP*). Interior Gateway protokol* atau *Interior Routing* Protokol dikembangkan untuk menghubungkan

router-router dibawah kendali administrator jaringan OSPF mendistribusikan informasi *routing*-nya di dalam *router-router* yang tergabung ke dalam suatu AS. AS adalah jaringan yang dikelola oleh administrator setempat. OSPF menggunakan protokol *routing link-state*, didesain untuk bekerja dengan sangat efisien dalam proses pengiriman update informasi *route*. OSPF merupakan protokol alternatif untuk menutupi kelemahan RIP. OSPF juga merupakan protokol *routing* yang menggunakan prinsip *multipath (multi path protokol*) dapat mempelajari berbagai *route* dan memilih lebih dari satu *route* ke *host* tujuan.

OSPF juga merupakan *routing* protokol yang berstandar terbuka. Maksudnya adalah *routing* protokol ini bukan ciptaan dari vendor manapun. Dengan demikian, siapapun dapat menggunakannya, perangkat manapun dapat kompatibel dengannya, dan di manapun *routing* protokol ini dapat diimplementasikan. OSPF merupakan *routing* protokol yang menggunakan konsep hirarki *routing*, artinya OSPF membagi-bagi jaringan menjadi beberapa tingkatan. Tingkatan-tingkatan ini diwujudkan dengan menggunakan sistem pengelompokan *area*. Dengan menggunakan konsep hirarki *routing* ini sistem penyebaran informasinya menjadi lebih teratur dan tersegmentasi, tidak menyebar ke sana ke mari dengan sembarangan. Efek dari keteraturan distribusi *routing* ini adalah jaringan yang penggunaan *bandwidth*-nya lebih efisien, lebih

cepat mencapai konvergensi, dan lebih presisi dalam menentukan *route-route* terbaik menuju ke sebuah lokasi. OSPF merupakan salah satu *routing* protokol yang selalu berusaha untuk bekerja demikian. Teknologi yang digunakan oleh *routing* protokol ini adalah teknologi *link State* yang memang didesain untuk bekerja dengan sangat efisien dalam proses pengiriman update informasi *route*. Hal ini membuat *routing* protokol OSPF menjadi sangat cocok untuk terus dikembangkan menjadi *network* berskala besar. Pengguna OSPF biasanya adalah para administrator jaringan berskala sedang sampai besar. Jaringan dengan jumlah *router* lebih dari sepuluh buah, dengan banyak lokasi-lokasi remote yang perlu juga dijangkau dari pusat, dengan jumlah pengguna jaringan lebih dari lima ratus perangkat komputer, mungkin sudah layak menggunakan *routing* protokol ini. [4]

OSPF digunakan bersamaan dengan IP, maksudnya paket OSPF dikirim bersamaan dengan *header* paket data IP. Setiap *router* OSPF mempunyai *database* yang identik yang menggambarkan topologi suatu *Autonomous System* yang disebut dengan l*ink state database* (*Topological database*). Dari database ini, perhitungan *Shortest Path First* dilakukan untuk membentuk *Routing Table*. Perhitungan ulang terhadap *Shortest Path First* dilakukan apabila terjadi perubahan pada topologi jaringan. OSPF memungkinkan beberapa jaringan untuk dikelompokkan bersama. Pengelompokkan seperti ini dinamakan

dengan *area* dan topologinya tersembunyi dari seluruh AS. Informasi yang tersembunyi ini memungkinkan penurunan *traffic routing*. Dengan menggunakan konsep *area* sistem penyebaran informasinya menjadi lebih teratur dan tersegmentasi. Dengan adanya distribusi *routing* yang teratur, maka penggunaan *bandwidth* akan lebih efisien, lebih cepat mencapai konvergensi, dan lebih presisi dalam menentukan *route* terbaik dalam mengirim paket. [4]

Secara garis besar, protokol *routing* OSPF (Link state) bekerja berdasarkan tahapan sebagai berikut:

a. Selama proses inisialisasi (permulaan), maupun dikarenakan adanya perubahan informasi *routing* berupa perubahan pada topologi jaringan, *router* akan menghasilkan sebuah *Link State Advertisement* (LSA). LSA ini berisikan informasi mengenai semua *link (interface*) pada *router* tersebut.
b. Pada proses selanjutnya, semua *router* akan melakukan pertukaran *link state* dengan mengirimkan paket *Link State Update* (LSU) yang berisikan LSA masing-masing *router*. Proses ini dikenal dengan proses *flooding* pada jaringan. Melalui proses ini, setiap *router* yang menerima LSU dari *router* lain akan menyimpan informasi tersebut ke dalam *Link state (topological*) *database*-nya, kemudian mengumumkan *update* tersebut ke *router* lain.

c. Setelah informasi l*ink state* database pada setiap *router* terbentuk, *router* akan melakukan perhitungan *Shortest Path* ke semua *router* lain pada jaringan dengan menggunakan *Dijkstra algorithm*. Alamat tujuan *cost* dan *hop* selanjutnya untuk mencapai alamat tujuan inilah yang kemudian membentuk tabel *routing* pada *router*.
d. Apabila tidak ada perubahan pada informasi *routing* misalnya: perubahan terhadap *cost* dan *link* pada suatu *router* ataupun terjadi penambahan maupun pengurangan *router* dalam jaringan, *router* akan sangat tenang (tidak terjadi pengiriman informasi *routing*).
e. Apabila terjadi perubahan pada informasi *routing* yang menyebabkan dikirimnya paket LSU maka *router* akan melakukan perhitungan ulang terhadap *Shortest Path* menggunakan *Dijkstra algorithm*. [6]

2.4.1 Cara Kerja OSPF [5]

Berikut adalah sedikit gambaran mengenai prinsip kerja dari OSPF:

- Setiap *router* membuat *Link State Packet* (LSP)
- Kemudian LSP didistribusikan ke semua *neighbour* menggunakan *Link State Advertisement* (LSA) *type* 1 dan menentukan DR dan BDR dalam 1 *Area*.

- Masing-masing *router* menghitung jalur terpendek *(Shortest Path*) ke semua *neighbour* berdasarkan *cost routing*.
- Jika ada perbedaan atau perubahan tabel *routing*, *router* akan mengirimkan LSP ke DR dan BDR melalui alamat *multicast* 224.0.0.6
- LSP akan didistribusikan oleh DR ke *router neighbour* lain dalam 1 *area* sehingga semua *router neighbour* akan melakukan perhitungan ulang jalur terpendek.

2.4.2 Konfigurasi OSPF - *Backbone Area* [5]

OPSF merupakan protokol *routing* yang menggunakan konsep hirarki *routing*, dengan kata lain OSPF mampu membagi-bagi jaringan menjadi beberapa tingkatan. Tingakatan-tingkatan ini diwujudkan dengan menggunakan sistem pengelompokan yaitu *area*.

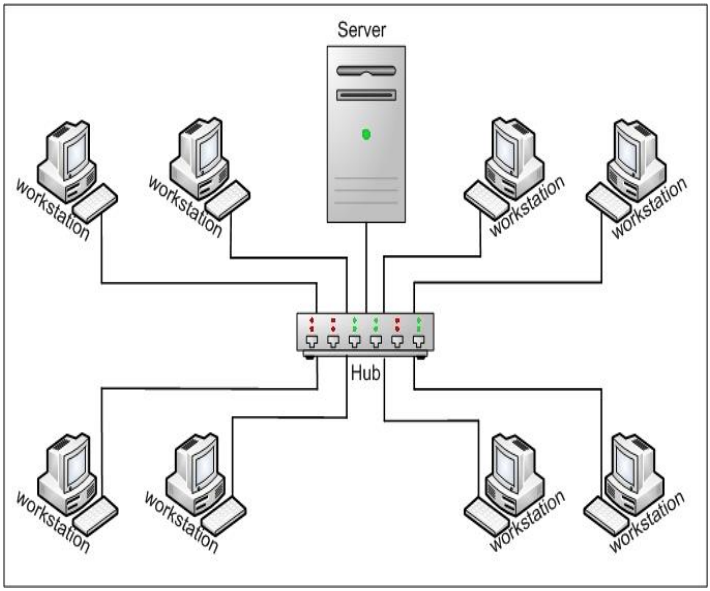
OSPF memiliki beberapa tipe *area* diantaranya:

- ***Bakcbone* - *Area* 0 (*Area* ID 0.0.0.0)** -> Bertanggung jawab mendistribusikan informasi *routing* antara *non-backbone area*. Semua sub-*Area* harus terhubung dengan *backbone* secara logikal.
- ***Standart/Default Area*** : Merupakan sub-*Area* dari *Area* 0. *Area* ini menerima LSA intra-*area* dan *inter-area* dar ABR yang terhubung dengan *area* 0 (*Backbone area*).

- ***Stub Area*** : *Area* yang paling "ujung". *Area* ini tidak menerima *advertise external route* (digantikan *default area*).
- ***Not So Stubby Area*** : Stub *Area* yang tidak menerima *external route* (digantikan *default route*) dari *area* lain tetapi masih bisa mendapatkan *external route* dari *router* yang masih dalam 1 *area*.

2.4.3 IMPLEMENTASI PROTOKOL ROUTING OSPF

Implementasi protokol *routing* OSPF dilakukan dengan menggunakan jaringan komputer dengan konfigurasi jaringan seperti pada gambar 1.



Gambar 2.1 Konfigurasi OSPF [5]

Untuk memulai semua aktivitas OSPF dalam menjalankan pertukaran informasi *routing*, hal pertama yang harus dilakukannya adalah membentuk sebuah komunikasi dengan para *router* lain. *Router* lain yang berhubungan langsung atau yang berada di dalam satu jaringan dengan *router* OSPF tersebut disebut dengan *Neighbour Router* atau *Router Neighbors*. Langkah pertama yang harus dilakukan sebuah *router* OSPF adalah harus membentuk hubungan dengan *Neighbor Router*. *Router* OSPF mempunyai sebuah mekanisme untuk dapat menemukan *router neighbors*-nya dan dapat membuka hubungan. Mekanisme tersebut disebut dengan istilah *Hello protokol*. Dalam membentuk hubungan dengan *neighbors*nya, *router* OSPF akan mengirimkan sebuah paket berukuran kecil secara periodik ke dalam jaringan atau ke sebuah perangkat yang terhubung langsung dengannya. Paket kecil tersebut dinamai dengan istilah *Hello packet*. Pada kondisi standar, *Hello packet* dikirimkan berkala setiap 10 detik sekali dalam media *broadcast multiaccess* dan 30 detik sekali dalam media *Point-to-Point*. *Hello packet* berisikan informasi seputar pernak-pernik yang ada pada *router* pengirim. *Hello packet* pada umumnya dikirim dengan menggunakan *multicast address* untuk menuju ke semua *router* yang menjalankan OSPF (IP *multicast* 224.0.0.5). Semua *router* yang menjalankan OSPF pasti akan mendengarkan protokol hello ini dan juga akan mengirimkan *hello packet*-nya secara berkala. Cara kerja dari o protokol dan

pembentukan neighbour *router* terdiri dari beberapa jenis, tergantung dari jenis media di mana *router* OSPF berjalan.[6]

OSPF memiliki 3 tabel di dalam *router* :

1. *Routing table* biasa juga disebut sebagai *Forwarding database*. *Database* ini berisi the *lowest cost* untuk mencapai *router-router*/*network-network* lainnya. Setiap *router* mempunyai *Routing table* yang berbeda-beda.
2. *Adjecency database, Database* ini berisi semua *router neighbors*nya. Setiap *router* mempunyai *Adjecency database* yang berbeda-beda.
3. *Topological database, Database* ini berisi seluruh informasi tentang *router* yang berada dalam satu jaringannya/*area*nya.

Kelebihan dari OSPF sebagai berikut

- Tidak menghasilkan *routing loop*
- Mendukung penggunaan beberapa metrik sekaligus
- Dapat menghasilkan banyak jalur ke sebuah tujuan
- Membagi jaringan yang besar mejadi beberapa *area*.
- Waktu yang diperlukan untuk konvergen lebih cepat

Kekurangan dari OSPF sebagai berikut :

- Membutuhkan basis data yang besar
- Lebih rumit

2.5 *INTERMEDIATE SYSTEM TO INTERMEDIATE SYSTEM* (IS-IS)

IS-IS menggunakan metode *link state* sebagai metode pengumpulan *route*nya dan menggunakan algoritma *Shortest Path First* (algoritma *Djikstra*) dalam melakukan perhitungannya. Protokol ini dirancang untuk beroperasi di OSI C*onnectionless Network Service* (CLNS). IS-IS mempunyai prinsip kerja yang mirip dengan protokol OSPF, tetapi berbeda dalam sistem pengalamatan dan struktur hirarki. Sistem pengalamatan yang digunakan IS-IS dalam sistem pengalamatan ciptaan ISO sendiri, yaitu sistem pengalamatan ISO (ISO *Addressing*). Jadi semua perangkat yang ingin digunakan untuk menjalankan IS-IS harus dapat dikonfigurasi dengan alamat ISO. Tetapi karena sistem pengalamatan IP lah yang banyak digunakan, maka sistem pengalamatan ISO juga dibuat kompatibel dengan IP. IS-IS menggunakan *ConnectionLess Network Protokol* (CLNP) *address*, dan ketika CLNP *address* digunakan di *router* maka disebut *Network Service Access Point* (NSAP) dan NSAP ini yang digunakan dalam sistem pengalamatan di IS-IS. [7]

IS-IS protokol dikembangkan oleh *Digital Equipment Corporation* sebagai bagian dari Tahap DECnet V standar oleh ISO pada tahun 1992 sebagai ISO 10589 untuk komunikasi antara perangkat jaringan yang disebut Sistem *Intermediate* oleh

ISO. Tujuan dari IS-IS adalah untuk memungkinkan *routing* datagram menggunakan ISO-OSI dikembangkan tumpukan protokol yang disebut CLNS. IS-IS dikembangkan di sekitar waktu yang sama bahwa Internet Engineering Task Force IETF mengembangkan protokol yang sama disebut OSPF. IS-IS kemudian diperluas untuk mendukung *routing* datagram dalam Internet Protokol (IP), *Network Layer* protokol Internet global. Ini versi IS-IS *routing* protokol kemudian disebut Terpadu IS-IS (RFC 1195). [8]

ISIS merupakan salah satu *routing* protokol IGP *(internal Gateway Protokol*) yang digunakan oleh *network device* dalam hal ini *router* untuk menentukan *best route* (*route* terbaik) untuk meneruskan *traffic* data ke suatu tujuan. IS-IS didevelop oleh DECnet sekitar tahun 1992 dimana pada waktu itu IETF juga sedang mengembangkan protokol OSPF. sebagai IGP *routing* protokol ISIS ber-operasi didalam *Administrative Domain* yang sama. seperti OSPF, protokol IS-IS juga merupakan *link-state* protokol dan sama-sama menggunakan *Djikstra Algorithm* untuk melakukan perhitungan dalam memilih *best path.* Bila dalam protokol OSPF terdapat konsep *Area*, di ISIS terdapat Level, Level disini merupakan batasan dari pengkelompokan *router-router*. terdapat Level 2, Level 1 dan L1/L2 level. **Level 2** merupakan *backbone area* dimana Level 2 ISIS *router* akan saling berbagi informasi bila *router-router* tersebut sama-sama

dikonfigurasi sebagai level 2 *area*. *router* yang dikonfigurasi sebagai **Level 1** makan akan berbagi informasi *routing* bila sama-sama dikonfigurasi Level 1 dan nilai IS-IS *Area* nya sama. sebelum IS-IS *router* dapat saling bertukar informasi, maka *router-router* tersebut haru membentuk adjacency terlebih dahulu.

IS-IS protokol terdapat 6 *state* IS-IS *adjacency*

1. *New*
2. *One-Way*
3. *Initializing*
4. *Up*
5. *Down*
6. *Reject*

New : Proses IS-IS *adjacency* baru dimulai

One-Way : Pada saat IS-IS *router* mengirim IS-IS Hello PDU

state router akan berubah Menjadi *One-Way*, dalam *state* ini lokal *router* belum menerima *hello message* dimana tercantum *address* nya sebagai *neighbor*

Initializing : Lokal *router* menerima *hello message* yang mencantum alamat lokal *router*, pada *state* ini komunikasi 2 arah sudah terbentuk

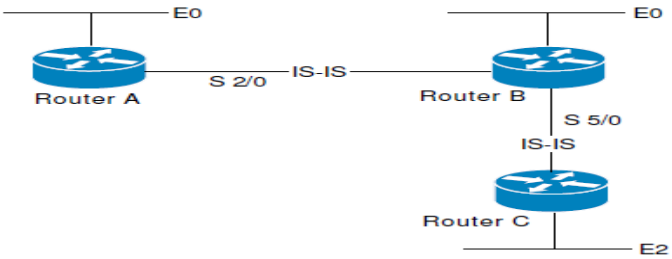
Up : *Adjacency* terbentuk dan pertukaran *database*

sudah dapat dilakukan

Down : *Adjacency* tidak berhasil dibentuk kemungkinan dikarenakan konfigurasi *area* yang tidak cocok, atau dikarenakan kesalahan konfigurasi parameter *hold time* atau parameter *authentikasi*

Reject : Apabila terjadi kesalahan authentikasi maka *state* akan bergulir ke status *Reject*

Apabila dalam OSPF terdapat LSA yang dipertukarkan untuk membentuk OSPF databse, dalam ISIS *network* informasi yang dipertukarkan berbentuk LSP (*Link-state PDU*), pada mpls juga terdapat term LSP (*Label Switching Path*) keduanya adalah hal yang berbeda. *Link-state* PDU berisis informasi tentang *router-router* yang tergabung didalam ISIS *network* dan informasi mengenasi *interface* yang terkoneksi serta metrik nya. *Link-state* PDU ini dibungkus dalam format TLV (*Type Length Value*), dengan format TLV memungkinkan protokol untuk memperluas kemampuan dan fungsinya dengan mudah.



Gambar 2. 2 IS-IS *Routing*

Standarisasi IS-IS adalah ISO 10589 yang menetapkan OSI IS-IS *routing* protokol untuk lalu lintas CLNS

- Sebuah protokol *Link State* dengan hirarki 2 tingkat arsitektur
- Jenis / Panjang / Nilai (TLV) pilihan untuk meningkatkan protokol

RFC 1195 menambahkan dukungan IP

- IS-IS *Intergrated*
- I / IS-IS berjalan di atas *Data Link Layer*

Kemiripan OSPF dan IS-IS

Keduanya menggunakan *Interior Gateway Protokol* (IGP)

- Mereka mendistribusikan informasi *routing* antara *router* milik Otonomi tunggal *System* (AS)
- Dengan dukungan untuk:
- *CIDR* (CIDR)
- Subnet *Variabel* Panjang Masking (VLSM)
- *Authentication*
- *Multi-path*
- *link IP* bernomor

2.5.1 Jenis PDU *Packet* di IS-IS *Routing*

Tumpukan OSI mendefinisikan unit data sebagai protokol data unit (PDU). Sebuah frame karena itu dianggap oleh OSI sebagai *PDU data-link*, dan paket dianggap sebagai PDU

jaringan. Ada empat jenis paket PDU, dan masing-masing jenis dapat Level 1 atau Level 2:

- LSP : suatu LSP adalah PDU yang dikirimkan antara dua *neighbors* IS-IS. LSP berisi informasi tentang *neighbors* dan biaya jalur, termasuk *adjacencies neighbors*, prefiks IP terhubung, *Open System Interconnection* (OSI) sistem akhir, dan alamat daerah. LSP digunakan oleh *router* penerima untuk mempertahankan tabel *routing* mereka.
- IIH : suatu IS-IS *Hello* PDU digunakan untuk membangun dan mempertahankan *adjacencies*.
- PSNP : sebuah parsial nomor urut PDU (PSNP) berisi ringkasan dari hanya sebagian dari LSP dikenal. Sebuah PSNP digunakan untuk mengakui dan meminta *link-state* informasi dengan meminta versi yang lebih baru dari lengkap LSP, atau mengakui penerimaan dari LSP masing-masing.
- CSNP -sebuah nomor urut lengkap PDU (CSNP) berisi ringkasan dari semua LSP yang diketahui oleh *router*.

2.5.2 LSP-Related Interval dan Exponential Backoff Timers

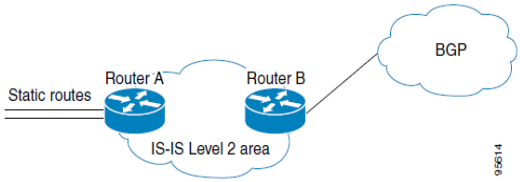
Berikut *timer* dan *interval* berhubungan dengan LSP yang dihasilkan oleh IS-IS *router*.

- LSP Refresh Interval : *Specifies* jumlah detik (0-65535) *router* akan menunggu sebelum merefresh LSP sendiri.
- Maksimum LSP lifetime --*Specifies* nilai seumur hidup di *header* LSP.

Berikut *timer backoff* eksponensial yang digunakan oleh di IS-IS untuk mengontrol perhitungan SPF, *Partial Route Calculation* (PRC), dan generasi pada LSP :

- Interval RRC --Specifies jumlah detik antara dua PRCS berturut-turut. Ketika perubahan yang tidak mempengaruhi topologi.
- Interval generasi LSP --*Specifies* jumlah detik antara menciptakan versi baru dari yang diberikan LSP pada basis per-*node*.
    - Interval SPF : Specifies jumlah detik antara dua perhitungan SPF berturut-turut.

Dalam skenario menggunakan tag *route*, mengkonfigurasi beberapa perintah pada satu *router* dan perintah lainnya pada *router* lain. Sebagai contoh konfigurasi memiliki peta *route* yang cocok pada tag dan menetapkan tag yang berbeda pada *router* di tepi jaringan, dan pada *router* yang berbeda dalam mengkonfigurasi redistribusi *route* berdasarkan tag di peta *route* yang berbeda.

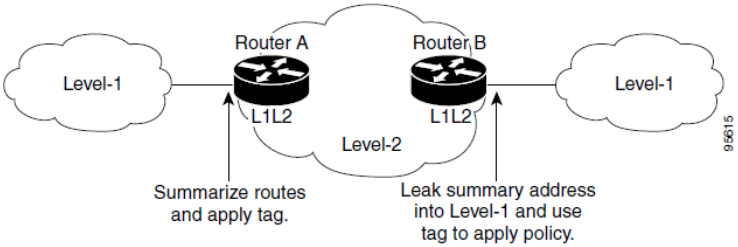


Gambar 2. 3 Meredistribusikan kembali IS-IS *route* menggunakan peta *route*

Gambar di atas menggambarkan sebuah flat Level 2 daerah IS-IS. Di tepi kiri adalah *route* statis dari *Router* A ke mencapai beberapa *prefiks* IP. *Router* A meredistribusi *route* statis menjadi IS-IS. *Router* B berjalan BGP dan meredistribusi IS-IS *route* ke BGP dan kemudian menggunakan tag untuk menerapkan kebijakan administratif yang berbeda berdasarkan nilai-nilai tag yang berbeda.

2.5.3 Tagging a Summary Address and Applying a Route Map

Gambar di bawah ini menggambarkan dua 1 daerah Tingkat dan satu Level 2 daerah antara mereka. *Router* A dan *Router* B adalah Level 1 / Level 2 *router* tepi di Level 2 daerah. Di tepi *Router* A, alamat ringkasan dikonfigurasi untuk mengurangi jumlah alamat IP dimasukkan ke dalam Level 2 IS-IS basis data. Pada *Router* B, alamat Ringkasan yang diteruskan ke daerah Level 1, dan kebijakan administrasi diterapkan berdasarkan nilai tag



.Gambar 2. 4 *Summary Address*

2.5.4 ISIS Metric

Metrik default dari ISIS *routing* protokol adalah 10 (baik itu *interface* Level 1 maupun Level 2) untuk *loopback interface* memiliki *default metric* 0, nilai metrik yang disupport oleh *IS Reachbility TLV, IP Internal Reachability TLV* dan *IP external Reachability TLV* maksimum adalah 63. dengan nilai tersebut tidak dapat men-*support Traffic Engineering*, agar bisa men-*support Traffic Engineering* di gunakanlah IS *Extended Reachability TLV* dan *IP Extended Reachbility* yang menggunakan 24-*bit metric* yang memiliki nilai maksimum 16777215.

Overload

Pada ISIS juga di kenal fitur *Overload* yang amat bermanfaat bila terjadi gangguan link ataupun mau dilakukan *maintenance* terhadap *router* tersebut. *router* yang di *set overload* maka ISIS akan membuat *traffic* tidak akan melalui *router* yang diset *overload* (*router* yang diset *overload* akan dihapus dari *Topology*), seting *overload* dapat bersifat *permanent*, maupun *temporary* (diset berapa lama *router* tersebut akan dianggap *overload*)

2.5.5 SISTEM PENGALAMATAN IS-IS [10]

IS-IS merupakan *routing* protokol yang diciptkana oleh *Interational Standarization Organization* (ISO). Tujuan diciptakan IS-IS oleh ISO adalah agar *routing* protokol ini

menjadi sebuah standar terbuka yang dapat digunakan oleh semua perangkat jaringan. Namun kenyataan yang lebih banyak digunakan adalah semua protokol dan sistem pengalamatan yang diciptakan berdasarkan organisasi standar *Open System interconnection* (OSI). Sistem pengalamatan IP yang selama ini dikenal di seluruh dunia dan *routing* protokol seperti OSPF diciptakan berdasarkan standarisasi dari OSI ini. Dengan demikian IS-IS tidak menggunakan sistem pengalamaan berdasarkan nomor IP. Sistem pengalamatan yang digunakan adalah sistem pengalamatan ciptaan ISO sendiri (ISO *Addressing*). Perangkat yang digunakan untuk menjalankan IS-IS harus dikonfigurasi dengan alamat ISO. Sistem pengalamatan ISO juga dibuat kompatibel dengan IP. Dalam penerapan pada sebuah *router* yang menjalankan IS-IS digunakan untuk membawa informasi *route* dalam *format* IP maka dari itu sebuah *router* yang tergabung dalam jaringan ini harus diberi alamat ISO untuk dapat mengirim dan menerima informasi ini.

2.5.6 STRUKTUR HIRARKI PENGALAMATAN

Routing protokol jenis *link state* menggunakan konsep *area* dalam sistem pengalamatannya sehingga jaringannya membentuk sebuah hirarki yang teratur.. Sistem *area* dalam IS-IS diberikan untuk keseluruhan perangkat *router*, artinya sebuah *router* hanya akan tergabung dengan sebuah *area* saja, tidak bisa tergabung kedalam banyak *area*. Hal ini dikarenakan peraturan

nomor-nomor pada *area* IS-IS hanya diberikan pada alamat ISO nya saja, dimana alamat tersebut biasanya hanya diberikan satu buah pada setiap *router*. *Router* yang berada dalam *area* sama baik OSPF maupun IS-IS dapat langsung saling berkomunikasi. IS-IS merupakan salah satu *routing* protokol *Link-State*, Tidak seperti OSPF , yang dikembangkan dan distandarisasi oleh *Internet Engineering Task Force* (IETF) , IS - IS adalah protokol ANSI ISO dan pada awalnya didasarkan pada Teknologi *Digital Equipment* Perusahaan DECNET Tahap V *Network*.

Pada IS- IS semua *router* menempatkan informasi dalam PDU *link*- lain yang diterima ke dalam *database link-state* mereka , dan semua *router* memiliki *routing table* yang sama dari topologi jaringan . IS - IS menjalankan algoritma SPF pada informasi dalam database *link-state* untuk menentukan jalur terpendek ke setiap tujuan pada jaringan, menempatkan pasangan tujuan / *next- hop* yang dihasilkan dari perhitungan SPF ke *database* IS- IS *routing* . Tidak seperti protokol lain yang biasanya berjalan pada TCP , UDP , atau IP, yaitu *OSI Layer* 3 atau Layer 4 protokol , IS - IS berjalan secara langsung pada data *link* layer (Layer 2) . [10]

Alamat IS- IS terdiri dari tiga bagian :

contoh : **49.0002.0010.0100.1001.0010.00**

NSAP addressing structure : AFI.Area ID.System ID.NSEL

1. *AFI* : tiga *byte* pertama adalah ID daerah . *Byte* pertama dari contoh ini - 49 - adalah *Authority and Format Identifier* (AFI). Nilai AFI 49 setara dengan RFC 1918 *address space* untuk protokol IP .

AFI Value	Address Domain
39	ISO Data Country Code (DCC)
45	E.164
47	ISO 6523 International Code Designator (ICD)
49	Locally administered (private)

Gambar 2.5 *AFI value of addressing domain*

2. *Area* ID : - 0001 atau 0002 - mewakili IS- IS *level* nomor 1 atau nomor 2, AFI dan *Area* ID disebut IDP (*Initial Domain Part*). Misalnya 47.0005 untuk *U.S Civilian Goverement*.
3. System ID : Mengidentifikasi node (*router*) pada jaringan . Identifier sistem setara dengan *host* atau bagian alamat pada alamat IP
4. NSEL (*NSAP Selector*) : *value* pada *router* harus 0(00), selain itu bukan termasuk IS (*Intermediate System/Router*). NSEL dengan *value* 0 disebut juga NET (*Network Entity title), system ID* dan NSEL disebut juga dengan DSP (*Domain Selector Part*)

NSAP Address

49.0002.0000.0c12.3456.00

IDI		DSP	
AFI	Area ID	System ID	NSEL
49	0002	0000.0c12.3456	00
3 bytes		6 bytes	1 byte

Gambar 2.6 *NSAP addressing*

Apabila IP dijadikan sebagai System ID, maka seperti berikut :

IP 10.1.2.3

1st step: Menambahkan angka men jadi tiga digit untuk tiap octet, contoh: 010.001.002.003

2nd step: Menggeser titiknya supaya jadi 3 octet (IP address kan 4 octet), contoh: 0100.0100.2003

3rd step: Memasukan **49.0002.0100.0100.2003.00 pada konfigurasi IS-IS NET (router id-nya IS-IS)**

2.5.7 Perbandingan IS-IS dan OSPF

Berikut ini adalah perbandingan konsep antara *routing* OSPF dengan IS-IS

OSPF	**ISIS**
- *Host*	- *End System*
- *Router*	- *Intermediate System*
- *Link*	- *Circuit*
- *Packet*	- *Protokol Data Unit*
- *Designated Router*	- *Designated IS*
- *Backup DR*	- *N/A (no BDIS is used*
- *Link State Advertisement*	- *Link state PDU*
- *Hello packet*	- *IIH PDU*
- *Database Description*	- *Complete number PDU (CSNP)*
- *Area*	- *Sub domain area*
- *Non backbone area*	- *Level-1 area*
- *Backbone area*	- *Level-2 sub domain (backbone)*
- *Area border router*	- *L1 L2 router*
- *Automonous system boundary router*	- *Any IS*

- OSPF menggunakan IP Protokol 89 sebagai transportasi

Data Link Header	*Ip Header*	*OSPF Header*	*OSPF Data*

- IS-IS langsung dikemas dalam Layer 2

Data Link Header	*IS-IS Header*	*IS-IS Data*

Dalam pemilihan IGP

- OSPF
- Semua jaringan harus memiliki daerah 0 (*backbone*)
- Setelan ISP dengan *core* berkecepatan tinggi di jaringan menghubungkan daerah PoPs.
- IS-IS
- *Router level* L2 harus dihubungkan melalui *backbone*
- Lebih fleksibel dibandingkan OSPF, tapi lebih rawan terhadap kesalahan (*error*)

Ada beberapa dasar pertimbangan dalam penggunaan *routing* OSPF dan IS-IS

1. Keamanan
   - IS-IS berjalan pada *link layer*
   - Tidak mungkin menyusupi IGP menggunakan IP seperti dengan OSPF
   - Tidak tergantung pada pengalamatan IP

- IS-IS NSAP skema pengalamatan menghindari ketergantungan IP seperti pada OSPF

2. Keandalan
   - IS-IS telah lama digunakan oleh mayoritas ISP terbesar dunia ini
   - Kepercayaan bahwa vendor peralatan lebih memperhatikan ISIS dalam kehandalan, skalabilitas, dan fitur.
3. Migrasi ke IPv6
   - Menambahkan IPv6 pada OSPFv2 dan OSPFv3 di dalam jaringan
   - IS-IS hanya memerlukan penambahan IPv6 pada *address family*
   - Sebagian besar jaringan mengoperasikan topologi tunggal untuk IPv4 dan IPv6

2.5.8. Persamaan dan Perbedaan antara OSPF dengan IS-IS

Persamaan OSPF dan IS-IS :

- Keduanya menggunakan *Interior Gateway Protocols*
- Menggunakan *link state*
- Menggunakan algoritma djisktra

Perbedaan OSPF dan IS-IS :

- Pada OSPF menggunakan metode TCI/IP milik IETF, sedangkan IS-IS menggunakan metode OSI (ISO/IEC 10589:2002)
- ISIS menggunakan *service* OSI layer 2 yang bernama CLNS (*ConnectionLess Network Service*) untuk *adjency*-nya
- OSPF menggunakan *service*-nya IP dan UDP
- IP protokol ISIS adalah CLNP (*ConnectionLess Network Protocol)*
- IP *addresing* ISIS adalah NSAP (*Network Service Access point)*
- OSPD terkenal dengan istilah *area*
- ISIS terkenal dengan istilah *level*
- OSPF menghubungkan menggunakan ABR (*Area Boundary Router)*
- ISIS menghubungkan *inter-level* perlu L2L1, tidak seperti OSPF yang semua *router* harus konek ke *area backbone* yang sama
- Pada OSPF terdapat DR (*Designated Router*)/BDR (*Backup Designated Router*) sedangkan pada IS-IS terdapat DIS (*Designated IS*) tapi tidak terdapat *backup* DIS

- Pada OSPF tidak terdapat konfigurasi untuk mengganti DR/DBR secara otomatis tanpa harus di *shutdown* terlebih dahulu.
- Pada IS-IS terdapat mekanisme konfigurasi untuk mengganti DIS secara otomatis yang disebut *Preemptive*.[10]

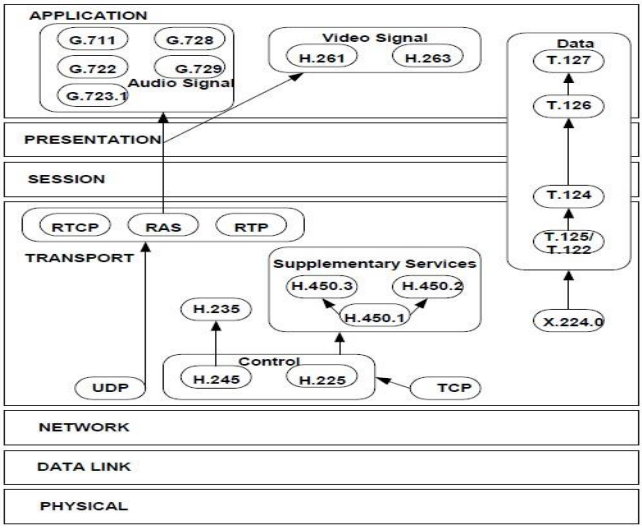
2.6 Videoconferencing pada Jaringan IP [11]

Layanan *videoconferencing* melalui jaringan IP, baik pada lingkungan industri, perusahaan, maupun akademis berkembang sangat cepat. Dengan popularitas internet yang tinggi, teknologi seperti *email*, *browser*, layanan *chat*, telah dikembangkan dan memfasilitasi komunikasi dan interaksi melalui internet. Oleh karena itu, gagasan untuk mengirimkan layanan *audio* dan *video* melalui internet pun tercetuskan, dan keuntungan dari pengiriman *audio*/*video* melalui internet telah menjadi jelas, hingga saat ini Dengan munculnya teknologi seperti H.323 (pada tahun 1996), SIP dan VRVS, menyediakan berbagai layanan yang berkaitan dengan *videoconferencing* berbasis internet. Saat ini, teknologi *videoconferencing* telah diaplikasikan dalam konferensi-konferensi rutin, *Distance Learning*, *Telecommuting* dan *Telemedicine*. Dengan *bandwidth* internet yang menjadi lebih murah dan lebih besar dari

waktu ke waktu, masa depan *videoconferencing* melalui Internet memang mempunyai prospek yang cerah .

2.6.1 Protokol H.323 [11]

H.323 merupakan payung standar dari *International Telecommunication Union* (ITU) yang mendefinisikan bagaimana komunikasi multimedia *real-time*, seperti *audio* dan *videoconferencing*, dapat dipertukarkan di jaringan paket *switch* (internet) yang tidak memberikan jaminan *quality of service* (QoS). Standar H.323 menyediakan dasar untuk layanan *audio*, *video*, dan komunikasi data melalui jaringan berbasis IP, termasuk internet.



Gambar 2.7. H.323 *protocol stack* [11]

Standar-standar ITU yang dipayungi oleh protokol H.323 antara lain:

<table>
<tr><td>Video H.261, H.263 (optional)</td><td>Data T.120 (optional)</td></tr>
<tr><td>Audio G.711, G.722, G.728, G.723,
G.729 (mandatory)</td><td rowspan="2">Security H.235 (optional)</td></tr>
<tr><td>Call Signaling H.225.0</td></tr>
<tr><td>Call Control H.245</td><td rowspan="2">Supplementary Services H.450</td></tr>
<tr><td>Multipoint H.323 (optional)</td><td>CS services H.246 (optional)</td></tr>
</table>

Beberapa protokol yang berhubungan dengan *signaling* antara lain:

- RAS : Bertugas dalam pengaturan registrasi, admisi, dan status.
- Q.931 : Mengatur *setup* panggilan dan *termination (call signaling).*
- H.245 : Bertugas dalam negosiasi penggunaan dan kemampuan kanal *(control signaling).*
- H.235 : Untuk keamanan dan autentikasi.

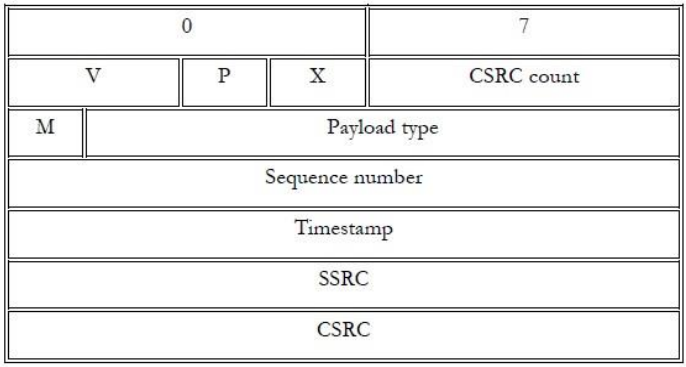
Standar *real-time transport protocol* dikembangkan oleh *Audio–Video Transport Working Group* yang mengacu pada *Internet Engineering Task Force* (IETF). Standarnya didokumentasikan ke dalam RFC 1889 dan RFC 1890.

Untuk pengimplementasian layanan aplikasi *real-time* harus menggunakan dua protokol yaitu :

1. RTP : Menyediakan layanan transportasi paket data secara *real-time.*
2. RTCP : Mengawasi kualitas layanan yang disediakan pada RTP *session* yang sudah ada.

2.6.2 RTP [11]

Protokol *real-time transport* (RTP) menyediakan fungsi *transport* jaringan *end-to- end* yang cocok untuk pengiriman aplikasi data *real-time* seperti *audio, video* atau data simulasi, melalui jaringan *multicast* atau *unicast.*



Gambar 2.8. RTP *fixed header* [11]

RTP tidak menggunakan *address resource reservation* dan tidak menjamin QoS untuk layanan *real-time*. Transportasi

data ditambah dengan *control protocol* (RTCP) untuk memonitor pengiriman data dalam cara yang terukur ke jaringan *multicast* yang besar, dan untuk memberikan fungsi minimal dari kontrol dan identifikasi. RTP dan RTCP didesain untuk independen dari *layer transport* dan *network*.

Format dari RTP *header* ditunjukkan dalam gambar II.2.

Ke**terangan:**

V

Versi

Mengidentifikasi versi RTP.

P

Padding. Jika diatur adanya *padding*, paket berisi satu atau lebih oktet *padding* tambahan diakhir, yang bukan merupakan bagian dari *payload*.

X

Extension bit. Jika diatur adanya *extension*, maka *fixed header* diikuti dengan satu *header* tambahan, dengan format yang telah ditetapkan.

CRC *count*

Berisi sejumlah CSRC *identifiers* yang mengikuti *fixed header*.

M

Marker. Sebagai penanda batas *frame* dalam suatu aliran paket.

Payload type

Mengidentifikasi format dari *payload* RTP dan menentukan interpretasinya berdasarkan aplikasi.

Sequence number

Bertambah satu untuk setiap paket data RTP yang dikirim, dan dapat digunakan oleh penerima untuk mendeteksi *packet loss* dan untuk mengembalikan urutan paket.

Timestamp

Menrefleksikan *sampling* dari oktet pertama dalam paket data RTP. *Sampling* harus dilakukan dalam waktu yang bertahap secara konstan dan linier untuk memungkinkan sinkronisasi dan perhitungan *jitter*. Resolusi waktu harus cukup

untuk akurasi sinkronisasi yang diinginkan dan untuk mengukur paket kedatangan

jitter.

H.245

H.245 berfungsi sebagai protokol *control signaling*. H.245 adalah saluran kontrol protokol yang digunakan misalnya pada sesi komunikasi H.323 dan H.324, dan melibatkan transmisi sinyal non-telepon. Protokol ini mengatur masalah

conference control dan *capability exchange messages*. *Capability exchange* dibutuhkan agar kedua pihak (atau semua pihak) yang ikut dalam suatu *conference* bisa membuat persetujuan mengenai media *stream* apa yang akan digunakan serta berbagai parameter *call control* lainnya.

H.261

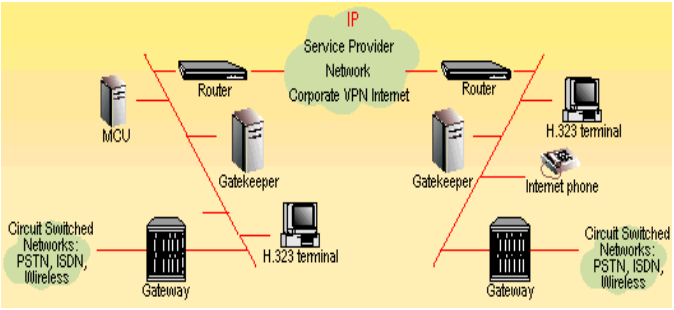
H.261 adalah standar *video coding* yang dikeluarkan oleh ITU. H.261 dirancang untuk kecepatan data yang merupakan kelipatan dari 64 Kbps, dan kadang-kadang disebut p x 64 Kbps (dimana p variabel dalam kisaran 1 – 30). Kecepatan datanya sesuai dengan kecepatan jalur ISDN, yang memang pada mulanya merupakan alasan *video codec* ini dirancang. H.261 *video streaming transport* menggunakan protokol *transport real-time*, RTP, dengan satu atau beberapa protokol yang membawa RTP.

2.7 Arsitektur Sistem [11]

Videoconferencing dengan protokol H.323 dapat menggunakan dua mode yaitu: *point-to-point* and *multipoint*. Arsitektur *videoconferencing* H.323 *point-to-point* diperlihatkan pada gambar II.6. Pengguna dalam panggilan *point-to-point* terhubung satu sama lain dengan menggunakan IP *address* atau alias salah satu pengguna. Dalam kaitan untuk memiliki beberapa pengguna dalam sebuah konferensi *video*, maka digunakan *multipoint*

videoconferencing. Arsitektur *videoconferencing* H.323 *multipoint*, seperti yang ditunjukkan pada gambar II.7 mengandung 4 komponen utama, yaitu:

- Terminal
- *Gateway*
- *Gatekeeper*
- *Multipoint Control Unit* (MCU)

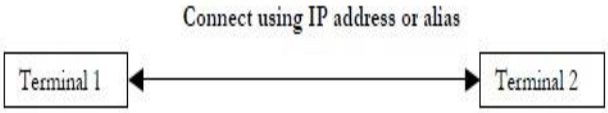


Gambar 2.9. Komponen-komponen sistem protokol H.323

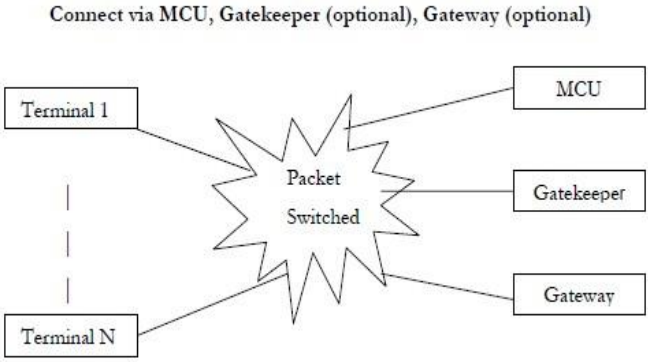
Terminal dapat merupakan perangkat yang berdiri sendiri *(stand-alone)* atau PC yang menjalankan sebuah aplikasi yang mampu melakukan komunikasi multimedia *bi-directional*. *Gateway* digunakan untuk melakukan panggilan antara dua jaringan yang berbeda. Sebuah *gatekeeper* merupakan komponen opsional yang

digunakan untuk memberikan layanan seperti pengalamatan, otentikasi terminal, penagihan dan mengelola *bandwidth*.

MCUs memberikan dukungan untuk konferensi tiga atau lebih terminal H.323 *(endpoint).* Semua terminal yang berpartisipasi dalam konferensi membuat sambungan dengan MCU. Dalam protokol H.323, sebuah MCU terdiri dari *Multipoint Controller* (MC) dan nol atau lebih *Multipoint Processor* (MPs). MC bertugas antar terminal untuk mengetahui *audio* atau *video coder / decoder* (CODEC) apa yang digunakan, dan juga mengelola sumber daya konferensi, dengan menentukan yang mana, jika ada, dari *audio* dan *video stream* tersebut akan di-*multicast*, namun MC tidak berhubungan langsung dengan salah satu media *stream*. Hal ini diserahkan kepada MP, yang me-*multiplex*, men-*switch* dan memproses *audio*, *video* dan/atau bit data. MC dapat terletak dalam *gatekeeper*, *gateway*, terminal atau MCU.



Gambar 2.10. Arsitektur *videoconferencing* H.323 *point-to-point* [6]



Gambar 2.11. Arsitektur *videoconferencing* H.323 *multipoint* [6]

2.8 Faktor-Faktor yang Mempengaruhi Sistem [11]

Ada beberapa faktor yang mempengaruhi performansi dari sistem *videoconferencing* H.323. Faktor-faktor tersebut dibagi menjadi 3 kategori, diantaranya:

1. Faktor manusia
2. Faktor perangkat
3. Faktor jaringan

Faktor manusia berhubungan dengan persepsi kualitas dari *audio*/*video* itu sendiri. Meskipun penilaian mengenai kualitas adalah masalah yang sangat subyektif, namun tetap ada batasnya, yaitu kualitas *audio/video* yang seperti apa yang dapat diterima oleh setiap orang. *Human error* berkaitan dengan kelalaian atau kurangnya pelatihan

juga dapat menjadi penyebab penurunan performa, yang mempengaruhi kualitas *audio*/*video*. Sebagai contoh, jika ada pengguna yang sedang tidak berbicara, tidak mematikan *(muting)* mikrofon selama konferensi berlangsung akan menyebabkan *noise* yang tidak diinginkan masuk ke dalam keseluruhan *videoconferencing*. Bagian II.4.1 berhubungan dengan persepsi manusia tentang kualitas *videoconferencing* secara lebih rinci.

Perangkat seperti H.323 *endpoints*, MCUs, *router*, *firewall*, *network address translator* (NATs) dan perangkat lain seperti modem, juga mempengaruhi kualitas *videoconferencing*. Sejauh mana H.323 *endpoints* dapat mempengaruhi kualitas *videoconferencing* tergantung pada *codec*, sistem operasi, kecepatan prosesor dan kapasitas memori pada *endpoint*. MCUs dan *router* mempengaruhi kualitas *videoconferencing*, juga termasuk ke dalam faktor jaringan, seperti keseluruhan *delay end-to-end*, *jitter* dan *packet loss*. Sehingga merupakan hal yang cukup sulit untuk membedakan dengan jelas pengklasifikasian terhadap faktor perangkat dan faktor jaringan.

Untuk penggelaran layanan *videoconferencing*, kita harus memperhatikan karakteristik dua faktor penting. Pertama adalah ketersediaan *bandwidth*, dan kedua adalah *delay end-to-end*. Jumlah aktual sesi *videoconferencing* yang dapat didukung jaringan dibatasi oleh kedua faktor tersebut.

Tergantung pada jaringan yang diteliti, ketersediaan *bandwidth* dan *delay* dapat menjadi faktor kunci yang dominan dalam menentukan jumlah sesi *videoconferencing* yang dapat didukung . Sesi *videoconferencing* terdiri dari dua *stream bidirectional* yang independen, yaitu suara dan *video* . *Bandwidth* yang diperlukan untuk panggilan suara dalam arah apapun adalah sebesar 50 pps *(packet per second)* atau 90,4 Kbps dengan *packet overhead*. *Codec* G.711 menyampel 20 ms suara per paket. Oleh karena itu 50 paket perlu ditransmisikan setiap detiknya, dimana setiap paket terdiri dari 160 sampel suara dalam kaitan untuk menghasilkan 8000 sampel suara per detik. Setiap paket dikirim dalam *frame Ethernet*. Dengan setiap paket berukuran 160 bytes, ditambah *header* dari *layer* protokol. Dalam *header* ini termasuk RTP + UDP + IP + *Ethernet* dengan ukuran 12 + 8 + 20 + 26 *bytes* secara berurutan. Sehingga total ada 226 *bytes* atau 1808 bits, yang perlu ditransmisikan setiap detiknya atau 90,4 Kbps dalam satu arah. Sehingga untuk dua arah (*uplink* dan *downlink*) memerlukan *bandwidth* untuk satu panggilan sebesar 100 pps atau 180,8 Kbps. Sedangkan untuk *video*, dalam penelitian ini menggunakan paket *video* yang diasumsikan berukuran tetap sebesar 2559 *bytes* dengan *frame rate* 30 fps, hal ini akan memberikan kecepatan 614 Kbps untuk trafik *video*. *Bandwidth* sebesar 614 Kbps

merupakan hasil perkalian besar paket $2559 * 30 * 8 = 614$ Kbps. Dengan tambahan *payload* 66 bytes *layer headers* , *bandwidth* yang diperlukan untuk layanan *video call* 630 Kbps. Untuk dua arah, *bandwidth* yang dibutuhkan untuk satu sesi panggilan *video call* adalah 60 pps, atau 1,26 Mbps dengan asumsi *symmetric flow*. Oleh karena itu *bandwidth* yang diperlukan untuk satu sesi layanan *videoconferencing* adalah 160 pps, atau 1,44 Mbps.

Bagian II.4.2 dan II.4.3 menggambarkan isu-isu mengenai kinerja trafik *audio*/*video* H.323 berkaitan dengan *delay end-to-end* keseluruhan, *jitter* dan *packet loss*.

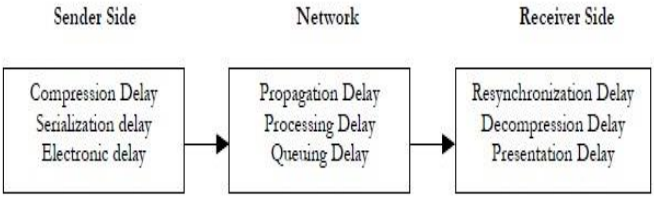
2.8.1 *Human Perception* [11]

Pengalaman paling umum dalam interaksi manusia terhadap sistem komunikasi adalah panggilan telepon lokal melalui jaringan telepon umum, yang memiliki *delay* satu arah sekitar 12 ms [10]. Agar manusia dapat berinteraksi secara alami, telah diamati bahwa *delay* 0 - 150 ms dalam komunikiasi *audio* menghasilkan interaktivitas yang baik dan *delay* satu arah hingga 400 ms masih dapat ditoleransi [11]. Persepsi visual manusia juga terbatas. Informasi visual yang ditangkap oleh mata adalah sekitar 800 Mbits dimana otak memproses 1/100 dari informasi tersebut . Oleh karena itu jika terjadi penurunan kualitas gambar *video*, orang awam hampir tidak akan menyadarinya. Antara *audio* dan

video, *latency* dari *audio* lebih tidak bisa toleransi karena menyebabkan *choppiness* dan *breakup* dalam *audio playback* di penerima. Meskipun ITU merekomendasikan *latency* dua arah maksimum untuk komunikasi suara yang dapat diterima adalah 300 ms, namun terlihat bahwa pengguna bersedia menerima *delay* sebesar 400 ms atau lebih jika solusi tersebut dapat mengurangi biaya. *Delay* diatas 600 ms ditolak oleh 40% dari pengguna telepon . Faktor-faktor tersebut membantu menunjukkan redudansi dalam *stream audio* dan *video* yang merupakan inspirasi untuk mengembangkan teknologi kompresi *audio/video*.

2.8.2 *Delay End-to-end*

Ada banyak elemen yang berkontribusi terhadap *delay end-to-end* dalam sistem *videoconferencing* H.323. Elemen-elemen ini perlu diidentifikasi dan solusi yang sesuai harus dikembangkan untuk menjaga agar *delay end-to-end* total di bawah terikat diinginkan.



Gambar 2.12. Berbagai elemen *delay* dari *delay end-to-end*

Untuk mendapatkan layanan natural *interactive videoconferencing*, batas atas *delay end-to-end* (terkadang disebut *latency*) untuk paket *video* dan suara harus dijaga seminimal mungkin. Pada dasarnya, *delay* dapat dibagi menjadi 3 komponen (gambar II.8), diantaranya: (i) *voice sampling* atau *frame grabbing*, *encoding*, kompresi, *delay* paketisasi pada pengirim, (ii) propagasi, *delay* transmisi dan *delay* antrian pada jaringan, dan (iii) *buffering*, dekompresi, depaketisasi, *decoding*, dan *delay playback* pada penerima. Menurut rekomendasi ITU G.114 , ketika *delay* kurang dari 150 ms, sebagian besar aplikasi interaktif, baik *speech* maupun *non-speech*, akan mengalami interaksi percakapan yang natural.

Untuk layanan suara, *delay end-to-end* terkadang disebut juga dengan M2E atau *Mouth-to-Ear delay* Menentukan *delay* yang lebih konservatif sebesar 100 ms untuk trafik suara dan *video* untuk memungkinkan interaksi natural/alami manusia. Pada *videoconferencing*, tidak ada *delay* terpisah antara trafik suara dan *video*, dimana suara dan *video* disinkronisasi dengan apa yang disebut dengan *"lipsync"*. Menurut eksperimen dari , perbedaan *delay* antara trafik suara dan *video* sebaiknya kurang dari 80 ms untuk mendapatkan interaksi dan impresi natural dari percakapan manusia. Untuk batas atas *end-to-end delay* dari paket suara dan *video*, dalam penelitian ini menggunakan 100 ms yang

dapat dibedakan menjadi 80 ms untuk *delay* jaringan, dan 20 ms untuk *delay* antar *workstation* pengirim dan penerima.

2.8.3 *Jitter* dan *Packet Loss* [11]

Jitter terjadi karena operasi internal dari komponen-komponen di dalam jaringan. Antrian dan *buffering* data di dalam jaringan, *packet rerouting*, *packet loss*, *multiplexing* jaringan merupakan beberapa faktor yang dapat menyebabkan *jitter*. *Jitter* juga dapat terjadi di *endpoints* yang berada di sumber trafik jaringan. *Jitter* ini disebut *insertion jitter* yang terjadi ketika sebuah paket tertunda sebelum ditempatkan pada *slot* transmisi karena transmisi sebelumnya belum lengkap. Ukuran paket juga mempengaruhi besarnya *insertion jitter*. Ukuran paket yang panjang meningkatkan *delay* secara keseluruhan karena *packet-processing overhead*. Ini adalah salah satu alasan bahwa aplikasi multimedia memiliki karakteristik ukuran paket yang kecil. Untuk mengurangi *jitter* pada sisi pengirim, perangkat *playback buffer* dapat digunakan di *endpoints*.

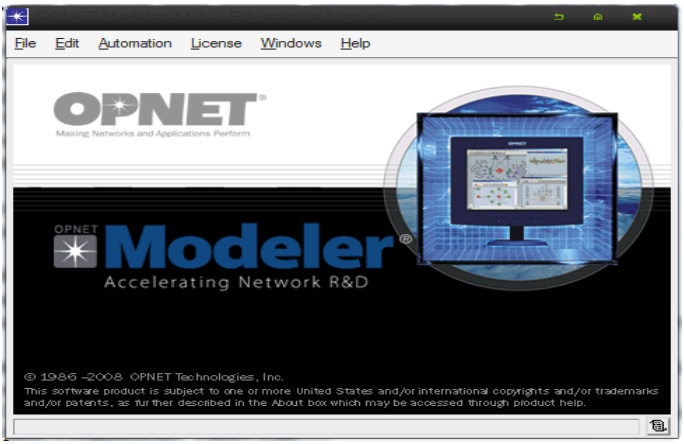
Hal lain yang mempengaruhi jumlah panggilan yang dapat didukung oleh suatu jaringan data adalah *packet loss*. *Packet loss* pada layanan *videoconferencing* harus di bawah 1% sesuai dengan , dan oleh karena itu *packet loss* dapat menjadi kendala ketiga yang memainkan peran kunci dalam menentukan jumlah panggilan dapat didukung oleh jaringan selain *bandwidth* dan *delay*. *Packet loss* dapat

disebabkan oleh perubahan pada *inter-arrival time* dari paket *audio* berkaitan dengan proses *intermediate router* sepanjang jalur pengiriman paket. Variasi yang tajam dalam nilai *jitter* menyebabkan peningkatan yang signifikan dalam *packet loss*. Penelitian telah menunjukkan bahwa perubahan 1% dalam *packet loss* satu arah setara dengan perubahan dari 220 ms dalam *delay* satu arah . Saling keterikatan di pengaturan parameter jaringan seperti ini merupakan salah satu tantangan bagi seorang insinyur jaringan untuk mencapai keseimbangan pada jaringan.

2.9 OPNET MODELER 14.5

OPNET Modeler adalah sebuah *network simulator* yang dirancang oleh *OPNET Technologies Inc*. OPNET Modeler mengakselerasikan R&D *network*, mengurangi *time-to-market*, dan meningkatkan kualitas produk. Dengan menggunakan simulasi, *network designers* dapat mengurangi biaya penelitian dan memastikan kualitas produk yang optimal. Teknologi terbaru OPNET Modeler menyediakan sebuah lingkungan untuk mendesain protokol dan teknologi juga menguji dan mendemonstrasikan dengan skenario yang realistik sebelum diproduksi. OPNET Modeler digunakan perusahaan perlengkapan jaringan terbesar di dunia untuk meningkatkan desain dari *network devices*, teknologi seperti VoIP, TCP, OSPFv3, MPLS, IPv6 dan lain-lainnya.[10]

Pada opnet memiliki banyak modul-modul yang disesuaikan dengan kebutuhan pengguna. Modul-modul yang banyak inilah yang menjadi mempermudah para pengguna untuk melakukan pemodelan, mendesain sebuah jaringan. Tampilan dari aplikasi OPNET Modeler ini seperti yang ada pada gambar 2.13.



Gambar 2.13 Tampilan Simulator OPNET Modeler Edisi 14.5[8]

2.10 Parameter Quality of Service

OPNET Modeler Edisi 14.5 akan membantu untuk proses simulasi jaringan dengan menggunakan kedua *routing* seperti yang sudah ditentukan pada judul skripsi ini. Pada opnet ada beberapa parameter yang akan diamati seperti

waktu konvergensi, *jiiter*, *throughput*, *packet delay*, dan *packet loss.*

2.10.1 Waktu konvergensi

Waktu konvergensi merupakan total waktu yang dibutuhkan oleh sebuah *router* selesai melakukan konvergensi, diantaranya menghitung jalur terbaik dan memperbarui tabel *routing*. Nilai konvergensi dapat diketahui ketika terdapat perubahan pada jaringan.

2.10.2 Packet loss

Packet loss adalah banyaknya paket yang hilang selama proses transmisi ke tujuan. Paket hilang terjadi ketika satu atau lebih paket data yang melewati suatu jaringan gagal mencapai tujuannya.

Rumus perhitungan *packet loss* adalah sebagai berikut :

$$\textbf{Packet Loss} = \frac{Pd}{Ps} x100\% \; 0 \leq t \leq T \text{.....(2.1)}$$

Keterangan :

Pd = Paket yang mengalami *drop* (paket), Ps = Paket yang dikirim (paket),

T = Waktu simulasi (detik), t = Waktu pengambilan sampel (detik).

2.10.3 Jitter

Jitter adalah variasi *delay*, yaitu perbedaan selang waktu kedatangan antar paket di terminal tujuan. *Jitter* dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya *congestion* dengan demikian nilai *jitter*-nya akan semakin besar.

Rumus perhitungan *jiiter* adalah : [14]

$$\textbf{\textit{Jitter}} = (Ri - Si) - (R_{i+1} - S_{i+1}) \quad \text{.....(2.2)}$$

Keterangan :

R = Received Time (Waktu paket data saat diterima atau datang)

S = Sent Start Time (Waktu awal pengiriman paket data)

Jitter memiliki Standarisasi seperti pada tabel 2.5 berikut ini.

Tabel 2.1 Standarisasi Penilian *Jitter*

No	Kategori	Besar *Jitter*
1	Sangat Bagus	0 ms
2	Bagus	1-75 ms
3	Sedang	76-125 ms
4	Jelek	126-225 ms

2.10.4 Throughput

Throughput adalah jumlah *bit* atau paket dari suatu unit data yang diterima dengan benar oleh penerima. Faktor yang menetukan *throughput* adalah seberapa besar data yang ditransfer, topologi jaringan yang dipakai, jumlah *user* serta kualitas dari perangkat.

Rumus untuk mencar *throughput* adalah :

$$\textbf{Troughput} = \frac{\sum \textbf{Packet Delivered}}{\sum \textbf{Packet Arrival Time} - \textbf{Packet Start Time}} \quad \text{.....(2.3)}$$

Keterangan :

Packets Delivered = Paket data yang terkirim

Packets Arrival Time = Waktu kedatangan paket data

Packets Start Time = Waktu awal dikirimkannya paket data

2.10.5 *Delay*

Delay adalah waktu yang dibutuhkan oleh sebuah paket data terhitung dari saat pengiriman oleh *transmitter* sampai saat diterima oleh *receiver*.

Rumus perhitungan *delay* adalah sebagai berikut:[14]

$$\textbf{Delay} = \sum \textbf{Packet Arrival Time} - \textbf{Packet Start Time} \quad \text{.....(2.4)}$$

Keterangan :

Packets Arrival Time = Waktu kedatangan paket data.

Packets Start Time = Waktu awal dikirimkannya paket data.

Delay memiliki Standarisasi seperti pada tabel 2.6 berikut ini.

Tabel 2.2 Standarisasi Penilian *Delay*[15]

No	Kategori	Besar *Delay*
1	Sangat Bagus	< 150 ms
2	Bagus	151-300 ms
3	Sedang	300-450 ms
4	Jelek	> 450 ms

Tabel 2.3 Target kinerja untuk aplikasi *audio* dan *video*[16]

Medium	Application	Degree of symmetry	Typical data rates	Key perfomance parameters and target values			
				One-way delay	Delay variation	Information loss (Note 2)	Other
Audio	Conversational voice	Two-way	4-64 kbit/s	<150 ms preferred (Note 1) <400 ms limit (Note 1)	<1 ms	< 3% packet loss ratio (PLR)	
Audio	Voice messaging	Primarily one-way	4-32 kbit/s	<1 s for playback <2 s for record	<1 ms	< 3% PLR	
Audio	High quality streaming *audio*	Primarily one-way	16-128 kbit/s (Note 3)	<10 s	<<1 ms	< 1% PLR	
Video	*Video*phone	Two-way	16-384 kbit/s	<150 ms preferred (note 4) <400 ms limit		< 1% PLR	
Video	One-way	One-way	16-384 kbit/s	<10 s		< 1% PLR	Lip synch: <80 ms

NOTE 1 – Assumes adequate echo control
NOTE 2 – Exact values depend on specific codec, but assumes use of a packet loss concealment algorithm to minimise
effect of packet loss.
NOTE 3 – Quality is very dependent on codec type and bit-rate.
NOTE 4 – These values are to be considered as long-term target values which may not be met by current technology.

Tabel 2.4 memberikan indikasi target kinerja yang sesuai untuk aplikasi data. Berdasarkan standar *International Telecommunication Union – Telecommunication* (ITU-T) G.1010 parameter *packet loss* atau *information loss*, *delay*, dan *jitter* (*variation delay*) untuk aplikasi data dapat dilihat pada tabel 2.4.

Tabel 2.4 Target kinerja untuk aplikasi data [16]

Medium	Application	Degree of symmetry	Typical amount of data	Key perfomance parameters and target values		
				One-way delay	Delay variation	Information loss (Note 2)
Data	Web-browsing - HTML	Primarily one-way	~ 10 KB	Preferred < 2 s /page Acceptable < 4 s /page	N.A.	Zero
Data	Bulk data transfer /retrieval	Primarily one-way	10 KB - 10 MB	Preferred < 15 s Acceptable < 60 s	N.A.	Zero
Data	Transaction services - high priority e.g. e-commerce, ATM	Two-way	< 10 KB	Preferred < 2 s Acceptable < 4 s	N.A.	Zero
Data	Command /control	Two-way	~ 1 KB	< 250 ms	N.A.	Zero
Data	Still image	One-way	< 100 KB	Preferred < 15 s Acceptable <	N.A.	Zero
Data	Interactive games	Two-way	< 1 KB	< 200 ms	N.A.	Zero
Data	Telnet	Two-way (asymmetr	< 1 KB	< 200 ms	N.A.	Zero
Data	E-mail (server access)	Primarily one-way	< 10 KB	Preferred < 2 s Acceptable < 4	N.A.	Zero

Tabel 2.4 Target kinerja untuk aplikasi data (lanjutan) [16]

Medium	Application	Degree of symmetry	Typical amount of data	Key perfomance parameters and target values		
				One-way delay	Delay variation	Information loss (Note 2)
Data	E-mail (server to server transfer)	Primarily one-way	< 10 KB	Can be several minutes	N.A.	Zero
Data	Fax ("real time")	Primarily one-way	~ 10 KB	<30s/page	N.A.	< 10-6 BER
Data	Fax (store & forward)	Primarily one-way	~ 10 KB	Can be several minutes	N.A.	< 10-6 BER
Data	Low priority transactions	Primarily one-way	< 10 KB	< 30 s	N.A.	Zero
Data	Usenet	Primarily one-way	Can be 1 MB or more	Can be several minutes	N.A.	Zero

NOTE – In some cases, it may be more appropriate to consider these values as response times.