

BAB II

JARINGAN INTERNET

2.1 Jaringan Internet

Jaringan internet adalah sebagai sarana penghubung untuk mendapatkan informasi-informasi berupa file-file dalam bentuk dokumen-dokumen, voice (suara), video, dan lain-lain yang dapat diakses kapanpun dan dimana kita berada baik di sekolah, rumah, kantor-kantor pemerintahan dan swasta, perusahaan dan sebagainya. Internet dapat diartikan sebagai jaringan computer luas dan besar mendunia, yaitu menghubungkan pemakai computer dari suatu Negara ke negara lain diseluruh dunia, dimana di dalamnya terdapat berbagai sumber daya informasi dari mulai yang statis hingga yang dinamis dan interaktif dan aplikasi layanannya ada beberapa jenis yaitu :

2.2 FTP (*File Transfer Protocol*)

FTP memungkinkan pengiriman paket atau data secara aman antara client dan server. Client maksudnya sisi yang menginisiasi transfer dan Server maksudnya remote host. Client FTP menghubungi server FTP pada port 21, menggunakan TCP sebagai protocol transport, kemudian port 20 juga akan dibuka untuk pertukaran data. FTP adalah sebuah protokol Internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pentransferan berkas (*file*) komputer antar mesin-mesin dalam sebuah internetwork. FTP merupakan salah satu protokol Internet yang paling awal dikembangkan, dan masih digunakan hingga saat ini untuk melakukan pengunduhan (*download*) dan penggugahan (*upload*) berkas-berkas komputer antara klien FTP dan server FTP. Sebuah Klien FTP merupakan aplikasi yang dapat mengeluarkan perintah-

perintah FTP ke sebuah server FTP, sementara server FTP adalah sebuah Windows Service atau daemon yang berjalan di atas sebuah komputer yang merespons perintah-perintah dari sebuah klien FTP. Perintah-perintah FTP dapat digunakan untuk mengubah direktori, mengubah modus transfer antara biner dan ASCII, menggugah berkas komputer ke server FTP, serta mengunduh berkas dari server FTP.

FTP menggunakan protokol *Transmission Control Protocol* (TCP) untuk komunikasi data antara klien dan server, sehingga di antara kedua komponen tersebut akan dibuatlah sebuah sesi komunikasi sebelum transfer data dimulai. Sebelum membuat koneksi, *port* TCP nomor 21 di sisi server akan “mendengarkan” percobaan koneksi dari sebuah klien FTP dan kemudian akan digunakan sebagai *port* pengatur (*control port*) untuk

1. Membuat sebuah koneksi antara klien dan server
2. Untuk mengizinkan klien untuk mengirimkan sebuah perintah FTP kepada server dan juga
3. Mengembalikan respons server ke perintah tersebut. Sekali koneksi kontrol telah dibuat, maka server akan mulai membuka *port* TCP nomor 20 untuk membentuk sebuah koneksi baru dengan klien untuk mentransfer data aktual yang sedang dipertukarkan saat melakukan pengunduhan dan penggugahan.

FTP hanya menggunakan metode autentikasi standar, yakni menggunakan *username* dan *password* yang dikirim dalam bentuk tidak terenkripsi. Pengguna terdaftar dapat menggunakan *username* dan *password*-nya untuk mengakses, *download*, dan *upload* berkas-berkas yang ia kehendaki. Umumnya, para pengguna terdaftar memiliki akses penuh terhadap beberapa direktori, sehingga mereka dapat membuat berkas, membuat direktori, dan bahkan menghapus berkas. Pengguna

yang belum terdaftar dapat juga menggunakan metode *anonymous login*, yakni dengan menggunakan nama pengguna *anonymous* dan *password* yang diisi dengan menggunakan alamat *e-mail*.

Alokasi *link-sharing* bisa dikonfigurasi statis atau dinamis. Tujuan pertama dari penerapan linksharing adalah setiap kelas akan mendapat *bandwidth* yang dialokasikan, dalam beberapa interval waktu tertentu, saat waktu *congestion*. Untuk kelas dengan alokasi *link-sharing* 0 seperti FTP, *bandwidth* yang diterima ditentukan oleh mekanisme penjadwal pada *gateway* dan mekanisme *link-sharing* tidak “menjamin” *bandwidth* untuk kelas tersebut dalam kondisi trafik penuh.

Tujuan kedua adalah ketika beberapa kelas tidak menggunakan alokasi *bandwidth*, distribusi dari kelebihan *bandwidth* antar kelas lain tidak boleh berubah-ubah tetapi mengikuti aturan tertentu. Floyd dan Jacobson mendefinisikan tujuan dari *link-sharing*:

1. Setiap *interior class* atau *leaf class* harus menerima *link-sharing* sesuai alokasi yang diberikan dalam interval waktu tertentu, sehingga terpenuhi kebutuhannya.
2. Jika semua *leaf* dan *interior class* yang memiliki permintaan telah menerima sedikitnya *link-sharing bandwidth* yang dialokasikan, distribusi dari kelebihan *bandwidth* tidak berubah-ubah tetapi harus mengikuti aturan yang ditentukan.

Pada sistem operasi ADSL yang difungsikan sebagai *router*, semua paket dimasukkan dalam antrian sebelum paket-paket tersebut diteruskan. Mekanisme dalam antrian tersebut diatur oleh *traffic controller* sesuai kebutuhan QoS. *Traffic Controller* hanya memanajemen paket-paket yang akan keluar dari sistem.

2.3 *Token Bucket Filter (TBF)*

Token Bucket Filter (TBF) adalah penerus paket yang datang pada sebuah rate yang tidak melebihi ketentuan rate yang diberikan. *Token bucket* merupakan suatu definisi formal dari suatu *transfer rate*. Antrian ini memiliki tiga buah komponen: ukuran *burst*, *mean rate*, dan *interval waktu (tc)*. Implementasi TBF terdiri dari sebuah *buffer (bucket)*, yang secara konstan diisi oleh beberapa informasi *virtual* yang dinamakan *token* pada *rate* yang spesifik (*token rate*). Parameter paling penting dari *bucket* adalah ukurannya, yaitu banyaknya token yang dapat disimpan. Setiap *token* yang masuk mengumpulkan satu paket yang datang dari antrian data dan kemudian dihapus dari *bucket*. Dengan menghubungkan *algoritma* ini dengan dua aliran – token dan data, maka akan didapatkan tiga buah kemungkinan :

1. Data yang datang pada TBF memiliki rate yang sama dengan masuknya token. Dalam hal ini, setiap paket yang masuk memiliki token-nya masing-masing dan akan melewati antrian tanpa adanya *delay*.
2. Data yang datang pada TBF memiliki *rate* yang lebih kecil daripada *rate token*. Hanya sebagian dari token yang dihapus pada output pada tiap paket data yang dikirim ke antrian, sehingga token akan menumpuk, memenuhi ukuran bucket. Token yang tidak digunakan kemudian akan dapat digunakan untuk mengirimkan data pada kecepatan yang melampaui *rate token* standar, ini terjadi jika ada ledakan data yang pendek.
3. Data yang datang pada TBF memiliki *rate* yang lebih besar daripada *rate token*. Hal ini berarti *bucket* akan segera kosong dari token, yang akan menyebabkan TBF akan menutup alirannya untuk sementara. Hal inilah yang dinamakan situasi *overlimit*. Jika paket-paket tetap datang, maka paket-paket akan segera di-drop.

2.4 Mekanisme Kerja *Hierarchical Token Bucket*

Pada HTB terdapat parameter *ceil* sehingga kelas akan selalu mendapatkan bandwidth di antara *base link* dan nilai *ceil linknya*. Parameter ini dapat dianggap sebagai Estimator kedua, sehingga setiap kelas dapat meminjam bandwidth selama bandwidth total yang diperoleh memiliki nilai di bawah nilai *ceil*.

Hal ini mudah diimplementasikan dengan cara tidak mengizinkan proses peminjaman bandwidth pada saat kelas telah melampaui *link* ini (keduanya *leaves* dan *interior* dapat memiliki *ceil*). Apabila nilai *ceil* sama dengan nilai *base rate*, maka kelas-kelas tidak diizinkan untuk meminjam bandwidth. Sedangkan jika nilai *ceil* diset dengan nilai tak terbatas atau dengan nilai yang lebih tinggi seperti kecepatan link yang dimiliki, maka akan didapat fungsi yang sama seperti kelas *nonbounded*. Sebagai contoh, seseorang dapat menjamin bandwidth 1 Mbit untuk suatu kelas, dan mengizinkan penggunaan bandwidth sampai dengan 2 Mbit pada kelas tersebut apabila link dalam keadaan *idle* (*seimbang*).

2.5 HTTP (*HyperText Transfer Protocol*)

HTTP merupakan protocol untuk aplikasi web yang menggunakan TCP dengan *port server 80*. Menggunakan komunikasi 2 arah, model *clientserver*. *Client* adalah *browser* yang melakukan *request* dan menerima objek *web*. *Server* adalah *web server* yang mengirim objek sebagai respon terhadap *request client*. HTTP adalah sebuah *protokol* meminta atau menjawab antara *client* dan *server*. Sebuah *client* HTTP seperti *web browser*, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke *port* tertentu di tuan rumah yang jauh (biasanya *port 80*).

2.6 Jaringan SPEEDY

SPEEDY adalah produk yang dikeluarkan oleh PT Telkom. *SPEEDY* merupakan layanan internet berkecepatan tinggi berbasis teknologi ADSL yang memungkinkan terjadinya komunikasi *voice*, *data*, dan video secara bersamaan pada media jaringan akses kabel tembaga. Secara umum, konfigurasi *SPEEDY* sama dengan konfigurasi teknologi ADSL, karena *SPEEDY* merupakan implementasi dari teknologi ADSL. *SPEEDY* mempunyai beberapa kelebihan dibandingkan dengan layanan internet lain. Kelebihan layanan *SPEEDY* antara lain:

Koneksi Internet lebih cepat dibanding menggunakan *analog modem*.

- a. Koneksi ke internet dapat dilakukan setiap saat (*dedicated connection*).
- b. Saluran dapat dipergunakan untuk pembicaraan telepon dan akses internet pada saat yang bersamaan.
- c. Koneksi memiliki sifat respon cepat dan aman.
- d. Memungkinkan dilakukan berbagi informasi dengan pengguna lainnya

Beberapa contoh layanan *SPEEDY* antara lain: *POTS (analog voice)*, *Internet* atau *fast internet*, *transfer file*, *interactive service (online game, tele-shopping, tele-medicine, dll)*, *VoIP (Voice over IP)*, *Broadcast TV*, *VOD*, *Audio* atau *video conference*, dan lain-lain.

2.6.1 Internet

Internet dapat diartikan sebagai jaringan computer luas dan besar mendunia, yaitu menghubungkan pemakai computer dari suatu Negara ke negara lain diseluruh dunia, dimana di dalamnya terdapat berbagai sumber daya informasi dari mulai yang statis hingga yang dinamis dan interaktif. Sebagai contoh kecil diantaranya:

2.6.2 Internet Protokol (IP)

Internet protokol didesain untuk interkoneksi sistem komunikasi computer pada jaringan *packet switched*. Secara umum *protokol* ini bertugas untuk menangani pendeteksian kesalahan pada saat *transfer data*. Untuk komunikasi datanya, *internet protocol* mengimplementasikan dua fungsi dasar yaitu *addressing* dan *fragmentasi*.

2.6.3 Transmission Control Protocol (TCP)

TCP merupakan salah satu *protocol* yang berada pada lapis *transport*. Konsep dasar cara kerja TCP adalah mengirim dan menerima *segment-segment* informasi dengan panjang dan data bervariasi pada suatu datagram internet. TCP menjamin *reliabilitas* hubungan komunikasi karena melakukan perbaikan terhadap data yang rusak, hilang atau kesalahan kirim. Data dikirim dalam sesi urutan paket. Tetapi penanganan datanya mengalami keterlambatan. *Transmission Control Protocol/Internet Protocol* (TCP/IP) adalah satu set aturan standar komunikasi data yang digunakan dalam proses *transfer data* dari satu komputer ke komputer lain di jaringan komputer tanpa melihat perbedaan jenis *hardware*. Protokol TCP/IP dikembangkan dalam riset pertama kali oleh *Defense Advanced Research Projects Agency* (DARPA) di Amerika Serikat dan paling banyak digunakan saat ini yang implementasinya dalam bentuk perangkat lunak (*software*) di system operasi.

Protokol TCP/IP memiliki standar segmen yang didefinisikan dalam beberapa bagian Berikut:

- a. *Source Port* 16 bit angka yang menunjukkan port asal.
- b. *Destination Port* 16 bit angka yang menunjukkan port tujuan.
- c. *Sequence Number* 32 bit angka yang digunakan untuk memastikan urutan yang benar dari data yang datang.

- d. *Acknowledgment Number* 4 bit untuk menandakan koneksi yang berhasil.
- e. *Data Offset* 32 bit indikasi data pertama.
- f. *Reserved* 6 bit diset 0.
- g. *Window* 16 bit angka dari oktet yang diterima oleh pengirim.
- h. *Checksum* 16 bit *checksum* yang telah dikalkulasi dari *Field header* dan data.
- i. *Urgent Pointer* 16 bit mengindikasikan akhir dari data yang penting.
- j. *Option variable bit maximum TCP segment size*.
- k. *Padding variable bit* memenuhi panjang *header* merupakan kelipatan 32 bit.

Data-data dari *protokol* lapis atas, TCP dan UDP menggunakan *nomor port* (atau *socket*) untuk melewati informasi ke lapis yang lebih atas. *Nomor port* digunakan untuk membedakan aplikasi yang berbeda yang melewati jaringan pada saat yang bersamaan. Suatu komunikasi yang tidak melibatkan suatu aplikasi dengan nomor port yang sudah dikenal, akan diberikan nomor-nomor port yang diambil secara random dari suatu rentang tertentu. sesuai dengan tujuan *QoS*, *administrator* dapat memberikan prioritas trafik tertentu. Suatu jaringan, mungkin saja terdiri dari satu atau beberapa teknologi data *link layer* yang mampu diimplementasikan *QoS* sesuai karakteristik teknologinya, misalnya *Frame Relay*, *Ethernet*, *Token Ring*, *Point-to-Point Protocol* (PPP).

Sebuah jaringan dengan *QoS-enabled* dapat dibuat dengan beberapa teknologi yang berbeda. Teknologi tersebut juga dibangun pada model *QoS* yang berbeda. Sebuah model *QoS* terdiri dari beberapa aspek berikut ini:

- a. *Scope* menetapkan jarak *logic* dimana sebuah model layanan disediakan.
- b. *Granularity* menetapkan satuan terkecil yang diperlakukan oleh sebuah metode layanan.

- c. *Time scale* menetapkan sifat *granularity* dalam satuan waktu dimana sebuah metode layanan disediakan.
- d. *Control model* menetapkan *entity* yang mengambil kontrol terhadap jaringan dan bagaimana cara melakukannya. Sebagai contoh adalah kontrol yang dapat dilakukan pada jaringan atau *end-system*.

Sedangkan salah satu model *QoS* yang dapat diaplikasikan adalah *packet scheduling* atau disiplin antrian atau penjadwalan paket dan dalam mengendalikan trafik administrator jaringan bisa memilih beberapa metode tergantung dari situasi pada jaringan *LAN* atau *backbone*. Tiap trafik akan dikendalikan dengan metode tertentu yang akan berdampak pada kecepatan akses, jadi administrator jaringan perlu membaca dan mengerti bagian ini terlebih dahulu, beberapa metode pengendalian trafik sebagai berikut:

2.6.3.1 First in First Out (FIFO)

Teknik antrian FIFO mengacu pada FCFS (*First Come First Server*), paket data yang pertama datang diproses terlebih dahulu. Paket data yang keluar terlebih dahulu di masukan ke dalam antrian FIFO, kemudian dikeluarkan sesuai dengan urutan kedatangan. Pada metode FIFO jika trafik melebihi nilai set maka paket data akan dimasukkan ke antrian, paket data tidak mengalami pembuangan hanya tertunda beberapa saat. Metode FIFO cocok diterapkan pada koneksi internet dengan bandwidth menengah 64kbps tetapi cukup menghabiskan sumber daya *processor* dan memori. Paket data jika melebihi batas konfigurasi akan di masukkan ke dalam antrian dan pada saat jaringan LAN tidak sibuk maka paket data dalam antrian akan dikeluarkan.

2.3.3.2 Stochastic Fairness Queuing (SFQ)

Stochastic Fairness Queuing (SFQ) memiliki kemampuan membagi setiap paket data yang diterima dalam jumlah yang sama rata, setiap paket data yang telah terbagi dimasukkan ke dalam suatu antrian dan menunggu dikeluarkan oleh penjadwalan.

2.6.3.3 Class Based Queue (CBQ)

Teknik klasifikasi paket data yang paling terkenal adalah CBQ, mudah dikonfigurasi, memungkinkan *sharing* bandwidth antar kelas (*class*) dan memiliki fasilitas *user interface*. CBQ adalah suatu *algoritma* yang didasarkan pada pembagian paket ke dalam kelas-kelas dan menjadwalkan paket di dalam antrian dengan suatu transmisi rate tertentu. Jika suatu antrian tidak digunakan, maka bandwidth disediakan untuk antrian lain. CBQ dapat mencapai penyekatan dan pembagian *link bandwidth* dengan struktur golongan. Tiap golongan mempunyai antrian tersendiri dan ditandai, dimana juga membagi bandwidth. CBQ dapat mengatur penggunaan bandwidth dari suatu golongan. Golongan '*child*' dapat di konfigurasi untuk meminjam bandwidth dari golongan induknya selama kelebihan bandwidth tersedia. CBQ mengatur pemakaian bandwidth jaringan yang dialokasikan untuk tiap user, pemakaian bandwidth yang melebihi nilai set akan dipotong (*shaping*), cbq juga dapat diatur untuk *sharing* dan meminjam bandwidth antar class jika diperlukan.

2.6.3.4 Parameter CBQ

1. *avpkt*, jumlah paket rata – rata saat pengiriman
2. *bandwidth*, lebar bandwidth kartu ethernet biasanya 10 – 100Mbit
3. *rate*, kecepatan rata – rata paket data.
4. *cell*, peningkatan paket data yang dikeluarkan ke kartu *ethernet* berdasarkan jumlah *byte*, misalnya 800 ke 808 dengan nilai *cell* 8.

5. *isolated / sharing*, parameter *isolated* mengatur agar bandwidth tidak bisa dipinjam oleh *klas* (class) lain yang sama tingkat / *sibling*. Parameter *sharing* menunjukkan bandwidth kelas (*class*) bisa dipinjam oleh kelas lain.

2.6.3 Karakteristik Qos

Banyak aplikasi yang tidak begitu sensitif terhadap *network congestion*. Sebagai contoh *File Transfer Protocol* (FTP) memiliki toleransi yang besar untuk *network delay* dan terbatasnya bandwidth. Berbeda dengan aplikasi-aplikasi baru seperti *audio* dan *video* yang pada umumnya sensitif terhadap *delay*, dalam hal ini *QoS* dapat digunakan untuk menyediakan jaminan layanan untuk aplikasi-aplikasi tersebut. Ada empat layanan yang digunakan untuk pengukuran parameter *QoS*, berikut ini adalah macam layanan tersebut:

1. *Bandwidth* merupakan *rate transfer* data maksimal yang dapat diteruskan antara dua titik. *Bandwidth* yang dibutuhkan untuk aplikasi suara dipengaruhi oleh *algoritma* kompresi yang digunakan. Sebagai contoh, *header* kompresi *Real Time Protocol* (RTP) dan teknik deteksi aktifitas dapat digunakan untuk mengurangi bandwidth yang digunakan oleh panggilan suara dalam jangkauan 11.2 Kbps sampai 50 Kbps. Aplikasi video membutuhkan koneksi bandwidth yang lebih tinggi dari 20 sampai 64 Kbps untuk kualitas dan resolusi rendah, sebesar 386 Kbps atau lebih untuk kualitas menengah dan sampai 15 Mbps untuk kualitas broadcast.
2. *Delay* merupakan waktu yang dibutuhkan oleh data untuk menempuh jarak dari asal ke tujuan. Delay dari pengirim ke penerima pada dasarnya tersusun atas *hardware latency*, *delay akses*, dan *delay transmisi*. Delay yang paling sering dialami oleh trafik yang lewat adalah *delay transmisi*. Dalam rekomendasi G.114

International Telecommunication Union (ITU) telah menspesifikasikan sebuah jangkauan 0-150 milidetik sebagai delay jaringan yang dapat diterima untuk aplikasi suara.

3. *Delay Variation* merupakan variasi delay *end-to-end* paket-paket yang diterima. Dimana pengiriman paket dilakukan secara *continuous stream*.
4. *Losses*, dimana kemungkinan hilangnya paket saat proses pengiriman. Dua aplikasi yang sensitif terhadap *loss* adalah *voice* dan *high definition video*

2.6.4 UDP (*User Datagram Protokol*)

User Datagram Protocol (UDP) adalah salah satu protokol lapisan transpor TCP/IP yang mendukung komunikasi yang tidak andal (*unreliable*), tanpa koneksi (*connectionless*) antara host-host dalam jaringan yang menggunakan TCP/IP. UDP yang merupakan salah satu protocol utama diatas IP adalah transport protocol yang lebih sederhana dibanding dengan TCP. UDP menyediakan layanan "*best effort*" (tidak handal), tidak berurutan, *unicast* atau *multicast* sehingga segment UDP bisa hilang. Data dikirim sebagai urutan *byte-byte (datagram)* dalam satu paket saja. Pengiriman data sangat cepat, sehingga delay yang ada hanya *delay propagasi* saja. UDP digunakan pada *DNS, VoIP*, dan lain-lain.

UDP memiliki karakteristik-karakteristik berikut:

1. *Connectionless* (tanpa koneksi): Pesan-pesan UDP akan dikirimkan tanpa harus dilakukan proses negosiasi koneksi antara dua host yang hendak berkar informasi.
2. *Unreliable* (tidak andal): Pesan-pesan UDP akan dikirimkan sebagai datagram tanpa adanya nomor urut atau pesan *acknowledgment*. Protokol lapisan aplikasi yang berjalan di atas UDP harus melakukan pemulihan terhadap pesan-pesan

yang hilang selama transmisi. Umumnya, protokol lapisan aplikasi yang berjalan di atas UDP mengimplementasikan layanan keandalan mereka masing-masing, atau mengirim pesan secara periodik atau dengan menggunakan waktu yang telah didefinisikan.

3. UDP menyediakan mekanisme untuk mengirim pesan-pesan ke sebuah protokol lapisan aplikasi atau proses tertentu di dalam sebuah host dalam jaringan yang menggunakan TCP/IP. Header UDP berisi *field Source Process Identification* dan *Destination Process Identification*.
4. UDP menyediakan penghitungan *checksum* berukuran 16-bit terhadap keseluruhan pesan UDP.

UDP tidak menyediakan layanan-layanan *antar-host* berikut:

1. UDP tidak menyediakan mekanisme penyanggaan (*buffering*) dari data yang masuk ataupun data yang keluar. Tugas *buffering* merupakan tugas yang harus diimplementasikan oleh protokol lapisan aplikasi yang berjalan di atas UDP.
2. UDP tidak menyediakan mekanisme *segmentasi* data yang besar ke dalam segmen-segmen data, seperti yang terjadi dalam protokol TCP. Karena itulah, protokol lapisan aplikasi yang berjalan di atas UDP harus mengirimkan data yang berukuran kecil (tidak lebih besar dari nilai *Maximum Transfer Unit/MTU*) yang dimiliki oleh sebuah antarmuka di mana data tersebut dikirim. Karena, jika ukuran paket data yang dikirim lebih besar dibandingkan nilai MTU, paket data yang dikirimkan bisa saja terpecah menjadi beberapa *fragmen* yang akhirnya tidak jadi terkirim dengan benar.
3. UDP tidak menyediakan mekanisme *flow-control*, seperti yang dimiliki oleh TCP.

Fungsi UDP sebagai berikut:

1. Protokol yang “ringan” : Untuk menghemat sumber daya memori dan prosesor, beberapa protokol lapisan aplikasi membutuhkan penggunaan protokol yang ringan yang dapat melakukan fungsi-fungsi spesifik dengan saling bertukar pesan.
2. Protokol lapisan aplikasi yang mengimplementasikan layanan keandalan : Jika protokol lapisan aplikasi menyediakan layanan transfer data yang andal, maka kebutuhan terhadap keandalan yang ditawarkan oleh TCP pun menjadi tidak ada.
3. Protokol yang tidak membutuhkan keandalan. Contoh protokol ini adalah protokol *Routing Information Protocol* (RIP).
4. Transmisi broadcast: Karena UDP merupakan protokol yang tidak perlu membuat koneksi terlebih dahulu dengan sebuah *host* tertentu, maka transmisi broadcast pun dimungkinkan. Sebuah protokol lapisan aplikasi dapat mengirimkan paket data ke beberapa tujuan dengan menggunakan alamat *multicast* atau *broadcast*. Hal ini kontras dengan protokol TCP yang hanya dapat mengirimkan transmisi *one-to-one*.

UDP berbeda dengan TCP yang memiliki satuan paket data yang disebut dengan segmen, melakukan pengepakan terhadap data ke dalam pesan-pesan UDP (*UDP Messages*). Sebuah pesan UDP berisi header UDP dan akan dikirimkan ke protokol lapisan selanjutnya (*lapisan internetwork*) setelah mengepaknya menjadi datagram IP. *Enkapsulasi* terhadap pesan-pesan UDP oleh *protokol IP* dilakukan dengan menambahkan *header IP* dengan protokol IP nomor 17 (0×11). Pesan UDP dapat memiliki besar maksimum 65507 byte: 65535 (216)-20 (ukuran terkecil dari header IP)-

8 (ukuran dari header UDP) byte. Datagram IP yang dihasilkan dari proses *enkapsulasi* tersebut, akan dienkapsulasi kembali dengan menggunakan *header* dan *trailer protokol* lapisan *Network Interface* yang digunakan oleh *host* tersebut.

Dalam header IP dari sebuah pesan UDP, *field Source IP Address* akan diset ke antarmuka host yang mengirimkan pesan UDP yang bersangkutan; sementara *field Destination IP Address* akan diset ke alamat IP unicast dari sebuah host tertentu, alamat *IP broadcast*, atau alamat *IP multicast*.

2.6.6 VoIP (*Voice over Internet Protocol*)

Voice over Internet Protocol (VoIP) merupakan suatu sistem yang menggunakan jaringan internet (IP) untuk mengirimkan paket data suara dari suatu tempat ke tempat lainnya. VoIP mentransmisikan sinyal suara dengan mengubahnya ke dalam bentuk digital, dan dikelompokkan menjadi paket-paket data yang dikirim dengan menggunakan platform IP (*Internet Protocol*). jenis perangkat telekomunikasi Voip adalah dua buah komputer terhubung dengan internet, dan teknologi ini memungkinkan percakapan suara jarak jauh melalui media internet. Data suara diubah menjadi kode digital dan dialirkan melalui jaringan yang mengirimkan paket-paket data, jaringan yang di maksud bukan lewat sirkuit analog telepon biasa. Syarat-syarat dasar untuk mengadakan koneksi VoIP adalah komputer yang terhubung ke internet, mempunyai kartu suara yang dihubungkan dengan speaker dan mikropon. Dengan dukungan perangkat lunak khusus, kedua pemakai komputer bisa saling terhubung dalam koneksi VoIP satu sama lain.

Hubungan tersebut dalam bentuk pertukaran *file*, suara dan gambar, sedangkan penekanan utama VOIP adalah hubungan keduanya dalam bentuk suara. Sekarang ini Voip mengalami *evolusi*, tidak hanya dalam bentuk komputer saja yang bisa terhubung,

tetapi pesawat telpon bisa juga bisa terhubung dengan jaringan *voip*. Perkembangan *voip* terjadi pada tahun 2000, dengan nama *voip* merdeka. Teknologi yang di gunakan adalah H.323. Pada tahun 2005 *voip* mengeluarkan jenis baru berbasis *Session Initiation Protocol* (SIP), yang menggantikan H.323. Pada tahun 2006 infrastruktur VoIP SIP dikenal sebagai *VoIP Rakyat*.

Keuntungan menggunakan jaringan *voip* diantaranya:

1. Biaya lebih rendah untuk sambungan langsung jarak jauh. Penekanan utama dari *VoIP* adalah biaya. Dengan dua lokasi yang terhubung dengan internet maka biaya percakapan menjadi sangat rendah.
2. Memanfaatkan infrastruktur jaringan data yang sudah ada untuk suara. Berguna jika perusahaan sudah mempunyai jaringan. Jika memungkinkan jaringan yang ada bisa dibangun jaringan *VoIP* dengan mudah. Tidak diperlukan tambahan biaya bulanan untuk penambahan komunikasi suara.
3. Penggunaan bandwidth yang lebih kecil daripada telepon biasa. Dengan majunya teknologi penggunaan bandwidth untuk voice sekarang ini menjadi sangat kecil. Teknik pemampatan data memungkinkan suara hanya membutuhkan sekitar 8kbps bandwidth.
4. Memungkinkan digabung dengan jaringan telepon lokal yang sudah ada. Dengan adanya gateway bentuk jaringan *VoIP* yang ada dikantor. Komunikasi antar kantor bisa menggunakan pesawat telepon biasa.
5. Berbagai bentuk jaringan *VoIP* bisa digabungkan menjadi jaringan yang besar. Contoh di Indonesia adalah *VoIP Rakyat*.
6. Variasi penggunaan peralatan yang ada, misal dari *PC* sambung ke telepon biasa, *IP phone handset*.

Kelemahan menggunakan jaringan voip diantaranya:

1. Kualitas suara tidak sejernih Telkom. Merupakan efek dari kompresi suara dengan bandwidth kecil maka akan ada penurunan kualitas suara dibandingkan jaringan *PSTN konvensional*. Namun jika koneksi internet yang digunakan adalah koneksi internet pita-lebar / broadband seperti Telkom Speedy, maka kualitas suara akan jernih - bahkan lebih jernih dari sambungan Telkom dan tidak terputus-putus.
2. Ada jeda dalam berkomunikasi. Proses perubahan data menjadi suara, jeda jaringan, membuat adanya jeda dalam komunikasi dengan menggunakan *VoIP*. Kecuali jika menggunakan koneksi *Broadband* (lihat di poin atas).
3. *Regulasi* dari pemerintah RI membatasi penggunaan untuk disambung ke jaringan milik Telkom.
4. Jika belum terhubung secara 24 jam ke internet perlu janji untuk saling berhubungan.
5. Jika memakai internet dan komputer dibelakang NAT (*Network Address Translation*), maka dibutuhkan konfigurasi khusus untuk membuat *VoIP* tersebut berjalan
6. Tidak pernah ada jaminan kualitas jika *VoIP* melewati internet.
7. Peralatan relatif mahal.
8. Berpotensi menyebabkan jaringan terhambat atau Stuck. Jika pemakaian *VoIP* semakin banyak, maka ada potensi jaringan data yang ada menjadi penuh jika tidak diatur dengan baik. Pengaturan bandwidth adalah perlu agar jaringan di perusahaan tidak menjadi jenuh akibat pemakaian *VoIP*.
9. Penggabungan jaringan tanpa dikoordinasi dengan baik akan menimbulkan kekacauan dalam sistem

2.7 Teknologi ADSL (*Asymmetric Digital Subscriber Line*)

Adalah suatu teknologi *MODEM* yang memiliki kecepatan pentransferan data 1.5 Mbps sampai 8 Mbps untuk mendukung implementasi layanan multimedia pada jaringan *broadband* dengan menggunakan satu pair kabel tembaga dan berupa teknik-teknik yang dipakai untuk memasukan informasi dalam suatu gelombang pembawa, biasanya gelombang sinus. Umumnya alat ini digunakan untuk merubah sinyal analog menjadi digital dan sebaliknya. misalnya untuk menghubungkan antara dua komputer melalui *dial-up* dengan menggunakan *line* telepon dalam mengakses data melalui jaringan atau internet.

Disebut *asymmetric* karena *rate* (kecepatan transmisi) dari arah *downstream* (sentral ke pelanggan) lebih besar dari arah *upstream* (pelanggan ke sentral), atau dapat dikatakan bahwa kecepatan transmisi dari arah *downstream* berbeda dengan dari arah *upstream*. *Bit rate downstream* 1,5-8 Mbps, *upstream* 16-640 Kbps. Adanya perbedaan kecepatan transmisi antara sisi *downstream* dan *upstream* dikarenakan kebutuhan koneksi internet lebih banyak digunakan untuk mengambil data (*download*) dari jaringan utama dibandingkan dengan pengiriman informasi (*upload*). Perbedaan antara *modem konvensional* dengan modem ADSL pada dasarnya dikarenakan perbedaan penggunaan frekuensi untuk mengirimkan sinyal atau data. Pada modem konvensional frekuensi yang digunakan di bawah 4 KHz, sedangkan pada modem ADSL digunakan frekuensi di atas 4 KHz.

Kelebihan modem ADSL yang lainnya adalah dari segi *line codingnya* yaitu menggunakan teknik *modulasi multicarrier* atau lebih dikenal dengan istilah DMT (*Discrete Multitone*). DMT mampu mengalokasikan bandwidth untuk transmisi data sehingga transmisi dari tiap sub kanal lebih maksimal. Teknik *multiplexing* yang

digunakan pada teknologi ADSL adalah melalui FDM (*Frekuensi Division Multiplexing*) atau *Echo Cancellation*. Cara kerja teknologi ADSL hanya berupa proses “*dial-up connection*”, bukan proses “*call set-up*” seperti jaringan *fixed telephone*, harus melalui proses dial tone dulu. Ketika ada permintaan dari *user* (pelanggan di rumah) untuk akses internet, maka *modem* ADSL sisi sentral akan langsung memprosesnya (dipisahkan apakah informasi yang diminta berupa data atau suara, alat pemisahanya disebut *splitter*). Selanjutnya informasi tersebut akan dilewatkan melalui MDF-RK-DP hingga KTB, kemudian di sisi pelanggan informasi data tersebut masuk ke *splitter* lagi, jika informasinya berupa akses internet (data) maka akan dimasukkan ke modem ADSL sisi pelanggan diteruskan ke *PC user*, jika berupa suara dari *splitter* langsung ke telepon, jika yang diminta video dari *splitter* masuk ke modem ADSL lalu masuk ke *Set Top Box* (STB) baru ke layar TV.

Beberapa keuntungan menggunakan teknologi ADSL adalah:

1. Menggunakan jaringan kabel tembaga existing atau kabel tembaga baru sehingga menghemat investasi penggelaran jaringan baru.
2. Mudah dalam proses instalasi
3. Dibandingkan dengan 56k modem, ADSL mampu menawarkan kecepatan hingga 125x lebih cepat.
4. Tidak perlu dial-up lagi, begitu komputer hidup, koneksi langsung tersambung.
5. ADSL memberikan kemampuan *Internet* dan *Voice* atau *Fax* secara simultan. Ini berarti kita dapat surfing internet dan menggunakan *Telepon* atau *Fax* pada saat bersamaan. Ini akan memberikan kepuasan untuk menikmati *High-Speed Internet Access* tanpa kehilangan kontak telepon dengan relasi.

6. Karena koneksi dilakukan dengan kabel sendiri, maka setiap pelanggan mendapatkan masing-masing koneksi *point-to-point* ke internet. Sehingga kestabilan koneksi dan keamanan lebih terjamin.

Akan tetapi ADSL juga memiliki kekurangan diantaranya :

1. Jarak yang terlalu jauh dari STO akan menurunkan kualitas sambungan dan menurunkan kecepatan.
2. Kabel tembaga tua dapat menurunkan kualitas sambungan dan menurunkan kecepatan.
3. Koneksi *asimetris* berarti waktu *upload* akan lebih lama daripada *download*.
4. Layanan ini tidak terdapat di semua wilayah

Ada beberapa perlengkapan yang dibutuhkan untuk menyediakan layanan-layanan ADSL. Komponen-komponen yang digunakan beserta fungsinya adalah sebagai berikut:

2.7.1 Transport System

Komponen ini menyediakan interface transmisi backbone untuk system DSLAM (*Digital Subscriber Line Access Multiplexer*). peralatan ini menyediakan *interface*, seperti T1/E1, T3/E3, OC-1, OC-3, STS-1, dan STS-3.

2.7.2 Local Access Network

Local Access Network menggunakan *local carrier inter-CO network* sebagai pondasi. *Switch ATM*, *Frame Relay*, dan *router* dapat digunakan untuk mengakses jaringan. Saat ini, ATM adalah sistem yang paling efisien.

2.8 Jaringan Traffic Control

Istilah *traffic control* dapat didefinisikan sebagai pengaturan lalu lintas aliran paket dari berbagai layanan dengan pengalokasian yang tepat dari suatu bandwidth untuk mendukung kebutuhan atau keperluan aplikasi atau suatu layanan jaringan. Istilah *bandwidth* dapat didefinisikan sebagai kapasitas atau daya tampung suatu channel komunikasi (medium komunikasi) untuk dapat dilewati sejumlah *traffic* informasi atau data dalam satuan waktu tertentu. Umumnya bandwidth dihitung dalam satuan bit, kbit atau bps (*byte per second*). Pengalokasian bandwidth yang tepat dapat menjadi salah satu metode dalam memberikan jaminan kualitas suatu layanan jaringan (*QoS = Quality Of Services*).

Pada pokoknya, setiap skema spengatur trafik meliputi pengatur antrian. Sejumlah besar disiplin antrian disarankan untuk mencapai persyaratan seperti kelayakan, perlindungan, batas performansi, kemudahan implementasi atau administrasi. Antrian yang berhubungan dengan blok-blok fungsional pada suatu router dalam tiap blok fungsional akan dibutuhkan untuk membangun layanan tertentu, namun tidak juga selalu diperlukan untuk layanan lainnya. Kenyataannya, pada kebanyakan *router* yang digunakan saat ini tidak menggunakan seluruh blok fungsional.

Paket-paket berdatangan pada satu *interface router (ingress interface)*, lalu diteruskan ke *interface lainnya (egress interface)*. Sebuah *router* mempunyai blok fungsional pada *ingress interface* untuk menjaga paket-paket yang datang, namun blok fungsional yang utama berada di *egress interface*. Fungsi dari tiap blok dijelaskan sebagai berikut:

1. *Classifier* adalah Paket *classifier* menggolongkan paket-paket berdasarkan dari isi dari beberapa bagian pada suatu header paket (e.g , pengalamatan dan nomor-nomor *port*). Paket-paket disesuaikan dengan aturan tertentu untuk proses lebih lanjut.
2. *Meters* adalah Trafics meters yang mengukur sifat aliran trafik (e.g., *bandwidth*, menghitung paket). Hasil pengukuran karakteristik disimpan sebagai flow state dan digunakan oleh fungsi lain.
3. *Markers* adalah *packet markers* mengumpulkan nilai akurat beberapa bagian dari *header paket*. Nilai yang dicatat menjadi prioritas, informasi *kongesti*, tipe aplikasi, atau tipe informasi lainnya.
4. *Droppers* adalah *droppers* menggantikan beberapa atau seluruh paket dalam aliran trafik dengan tujuan untuk membatasi panjang antrian, atau sebagai informasi *kongesti* yang lengkap.
5. *Queues* adalah *buffer-buffer* terhingga untuk menyimpan paket-paket *backlogged*. Suatu disiplin antrian mungkin dapat mempunyai antrian ganda untuk kelas *trafik* yang berbeda.
6. *Schedulers* adalah memilih paket untuk dikirimkan dari paket *backlogged* diantrian.
7. *Shapers* adalah menunda beberapa atau seluruh paket di aliran *trafik* dengan maksud untuk membatasi laju puncak dari aliran. Suatu *shaper* biasanya memiliki ukuran *buffer* terhingga, dan paket mungkin saja ditukar jika tidak ada ruang yang cukup untuk menahan paket yang ditunda.

Mekanisme shaper lain yang populer adalah *token bucket* yang membolehkan ledakan (*burst*) kecil dengan ukuran ledakan yang dapat di konfigurasi. Token bucket dapat mengakomodasi aliran-aliran trafik dengan karakteristik ledakan sehingga lebih sesuai untuk *trafik internet* saat ini. Tujuan lain disiplin antrian adalah menghindari



2.12.1 Bandwith Manajemen

1. Bandwidth Management (*Traffic Control/Shaping*) adalah suatu istilah yang ditujukan pada suatu subsistem antrian packet dalam/pada suatu jaringan atau *network devices*. Secara singkat *traffic control/shaping* adalah suatu usaha mengontrol traffic jaringan sehingga bandwidth lebih optimal dan performa network lebih terjamin. Rumusan untuk mendapatkan hasil bandwith yang ingin dicapai dapat dicari dengan persamaan (2.2)

$$\frac{\text{Bandwith(Telkom)}}{\text{Hasil Troughput sebulan}} = \frac{\text{Bandwith Total}}{\text{Standar troughput(Telkom)}} \dots\dots\dots(2.2)$$

2.12.2 Traffic metering

Traffic metering digunakan untuk mengukur paket yang melewati jaringan pada kondisi tertentu, kemudian hasilnya diperlukan untuk membuat skala bandwidth pada *QoS* atau memberikan penanganan lebih lanjut terhadap trafik paket data.