

## BAB IX

### TEKNIK SWITCHING

**K**omunikasi voice ataupun data tidak terlepas dari teknik switching. Berikut adalah uraian beberapa teknik switching yang diterapkan. Teknik Switching dikenal ada dua buah yaitu *Circuit Switching* and *Packet Switching*.

#### 8.1 Circuit Switching

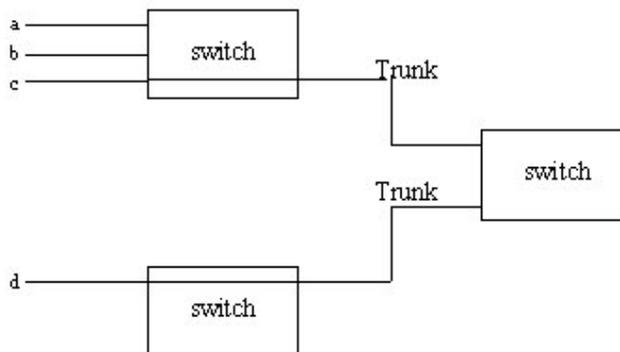
Menerapkan sebuah path komunikasi yang *dedicated* (permanen) antara 2 buah station

- melibatkan tiga fase :
  - *Circuit Establishment*
  - *Signal Transfer* (mungkin analog voice, digitized voice, binary data)
  - *Circuit disconnect*
- kurang efisien karena koneksi tetap *established* walaupun tidak ada data yang ditransfer
- contoh konkret adalah *public telephone network*, PBX (Public Branches eXchange utk gedung)
- tidak complex dalam routing, *flow control*, dan syarat-syarat *error control*

#### 8.2 Routing dalam Circuit Switching

Efisiensi jaringan diperoleh dengan cara meminimisasi switching and kapasitas transmisi. Komponen dalam arsitektur jaringan telekomunikasi umum adalah :

- *pelanggan*
- *local loop* : *link* antara pelanggan dan jaringan. Hampir semuanya menggunakan twisted pair. Panjangnya antara beberapa kilometer dan beberapa puluh kilometer.
- *exchanges* : switching lokal dalam sebuah jaringan.
- Switching Lokal mendukung pelanggan-pelanggan yang dikenal dengan nama *end office* yang biasanya dapat mendukung beribu-ribu pelanggan dalam *local area*.
- *trunks* : cabang-cabang antara *exchanges*. *Trunks* membawa *multiple voice-frequency* dengan menggunakan FDM (*Frequency Division Multiplex*) atau *synchronous TDM* (*Time Division Multiplex*).



**Gambar 8.1** Proses Routing

a dan b koneksi dalam satu buah *end office*, sedangkan c dan d koneksi yang lebih kompleks. Lebih disukai menggunakan *dynamic routing* daripada *static routing*

dikarenakan kondisi traffic yang makin kompleks dan lebih fleksibel. Adapun dalam kelas-kelas dalam *dynamic routing* adalah sebagai berikut :

### 1. Alternate Routing

Adalah routing-routing pilihan yang dapat digunakan antara dua *end office*. Tiap switch diberikan sejumlah route untuk mencapai tiap tujuan. Jika hanya ada satu route dalam tiap pasang *source-destination*, ini disebut dengan *fixed alternate routing*. Yang lebih umum digunakan adalah *dynamic alternate routing*. *Routing decision* didasari atas *status current traffic* (akan ditolak jika sibuk) dan *historical traffic patterns* (urutan-urutan route yang diinginkan).

### 2. Adaptive Routing

Didesain untuk memfungsikan switch dalam mengubah bentuk *traffic* pada sebuah jaringan. Situasi seperti ini, switch yang ada saling bertukar informasi untuk mempelajari kondisi jaringan sehingga tipe routing ini lebih efisien daripada *alternate routing* dalam hal *resourcing* jaringan.

DTM (*Dynamic Traffic Management*) yang dikembangkan oleh Northern Telecom menggunakan *central network* untuk mencari *the best alternate route* bergantung dari *congestion* (kepadatan) dalam jaringan tersebut. *Central controller* mengumpulkan status data dari tiap switch untuk mencari alternate route yang diinginkan.

Jaringan dengan menggunakan *circuit-switched* adalah didesain untuk *voice traffic*. Walaupun demikian, *circuit-switched network* juga digunakan dalam komunikasi data dimana akan terjadi :

- untuk *terminal-to-host data connection*, waktu pada line terbuang percuma. Jadi komunikasi data akan tidak efisien jika menggunakan *circuit-switched network*.
- koneksi menyediakan rate yang konstan. Jadi device yang saling terhubung mempunyai rate yang sama saat *transmit* atau *receiving data*. Ini membatasi utilitas dalam jaringan yang banyak terdapat variasi komputer dan terminal.

### 8. 3. Packet Switching

Dalam *Packet Switching*, data yang ditransmisikan dibagi-bagi ke dalam paket-paket kecil. Jika *source* mempunyai *message* yang lebih panjang untuk dikirim, *message* itu akan dipecah ke dalam barisan-barisan paket. Tiap paket berisi data dari user dan *info control*. *Info control* berisi minimal adalah info agar bagaimana paket bisa melalui jaringan dan mencapai alamat tujuan.

Beberapa keuntungan yang diperoleh dari *packet switching* :

- efisiensi *line* sangat tinggi; hubungan *single node-to-node* dapat dishare secara dinamis oleh banyak paket. Paket-paket diqueue dan ditransmisikan secepat mungkin. Secara kontras, dalam *circuit switching*, waktu pada *link node-to-node* adalah dialokasikan terlebih dahulu menggunakan *time-division multiplexing*.
- jaringan packet-switched dapat membuat konversi *data-rate*. Dua buah station yang berbeda *data-ratenya* dapat saling menukar paket.
- ketika traffic mulai padat, beberapa *call* diblok, yang menunjukkan jaringan menolak permintaan koneksi tambahan sampai beban di jaringan menurun. Dalam *packet switched network*, paket masih dapat diterima akan tetapi *delay delivery* bertambah.
- prioritas dapat digunakan. Jadi kalau sebuah node mempunyai sejumlah *queued packet* untuk ditransmisikan, paket dapat ditransmisikan pertama kali berdasarkan prioritas yang lebih tinggi. Paket-paket ini mempunyai delay yang lebih kecil daripada *lower-priority packets*.

#### 8.4. Operasi Internal

Ada dua pendekatan yang berhubungan dengan jaringan, yaitu *datagram* dan *virtual circuit*. Pada *datagram* tiap paket bisa dirutekan berbeda, misalnya station A akan kirim paket 1, 2, dan 3. Route A menuju E ada dua route, maka kemungkinan paket 1 menempuh route yang berbeda dengan paket 2 tergantung dari kepadatan masing-masing jalur. Sedangkan pada *virtual circuit*, sebuah route antara station dikonfigurasi sebelum terjadi transfer data. Ini bukan *dedicated path* seperti dalam *circuit-switching*. Sebuah paket masih disimpan dalam tiap node. Perbedaannya dengan *datagram* adalah node tidak perlu melakukan *routing decision* untuk tiap paket, dilakukan hanya sekali dan berlaku untuk semua paket.

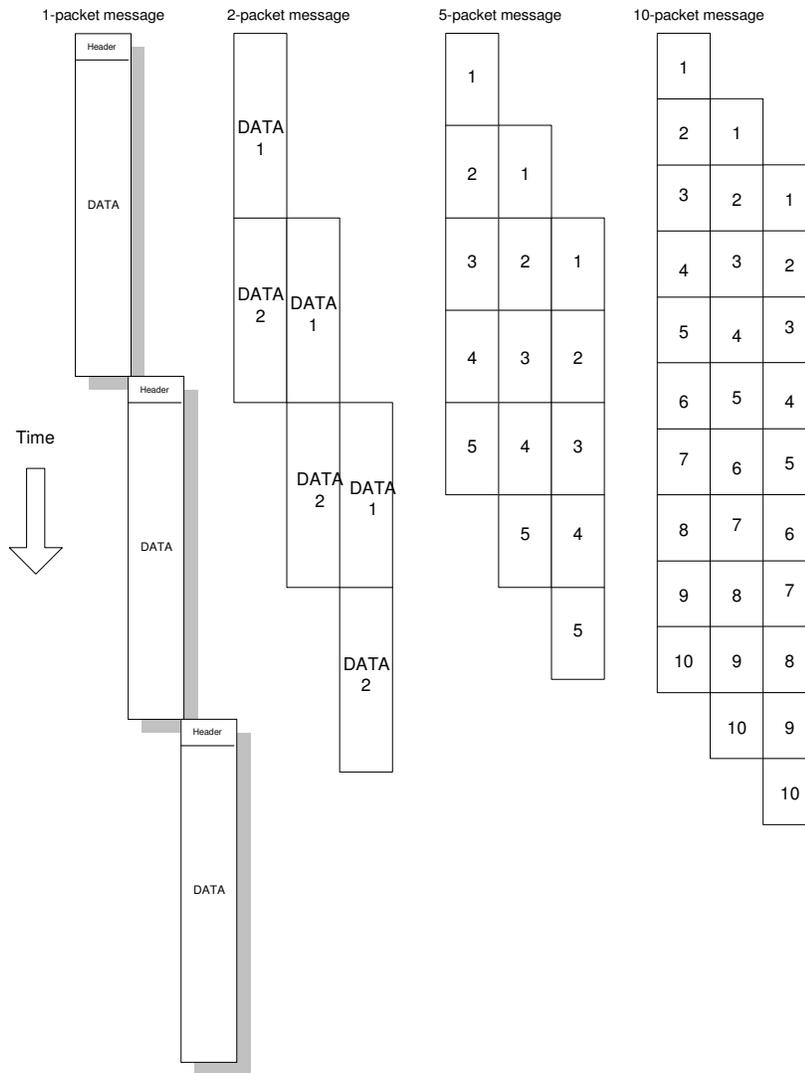
Jika ada dua station yang akan saling menukar data dalam periode waktu tertentu, maka dapat dipastikan keuntungan banyak diperoleh jika menggunakan *virtual circuit*. Pertama, jaringan menyediakan pelayanan yang berhubungan dengan *virtual circuit* termasuk *sequencing and error-control*. *Sequencing* berfungsi apabila semua paket mengambil route yang sama. *Error control* adalah pelayanan untuk meyakinkan semua paket dapat tiba di tujuan, tapi juga tiba dengan paket yang benar-benar diinginkan, tidak ada cacat.

Keuntungan dari datagram adalah *call setup phrase* dapat dihindari. Jadi sebuah station yang mengirim hanya satu atau sedikit paket pengiriman *datagram* akan lebih cepat. Keuntungan yang lain adalah lebih *flexible*, lebih *primitive*. Sebagai contoh, apabila ada satu bagian network yang buntu, maka *datagram* yang dikirim akan mengambil route menjauhi network tersebut. Dengan penggunaan *virtual circuit*, karena paket-paket didefinisikan *routingnya* sebelum dikirim maka hal ini akan menjadi sulit apabila route yang diambil mengalami buntu. Keuntungan ketiga adalah pengiriman *datagram* secara tersirat lebih *reliable*. Pada *virtual circuit*, apabila ada node yang gagal, semua *virtual circuit* yang mendefinisikan lewat node tersebut akan lenyap. Pada *datagram*, paket-paket akan mencari alternatif routing dimana akan mengabaikan node yang gagal. Di *virtual circuit* pada operasi internalnya digunakan *packet-switching*.

Dari sudut pandang user, tidak akan dapat begitu berbeda apabila provider menggunakan *packet-switched* atau *circuit-switched network*.

#### 8.5. Ukuran Paket

Ada hubungan antara ukuran paket dengan waktu dalam pentransmisi data. Pada gambar terlihat bahwa data apabila dipecah makin kecil membutuhkan waktu lebih cepat, dan tiap paket pecahannya harus disisipi *headernya*. Akan tetapi jika dipecah semakin kecil akan didapatkan waktu transmisi yang lebih besar dari sebelum paket lebih diperkecil lagi. Dalam hal ini harus dipilih pemecahan paket yang optimum.



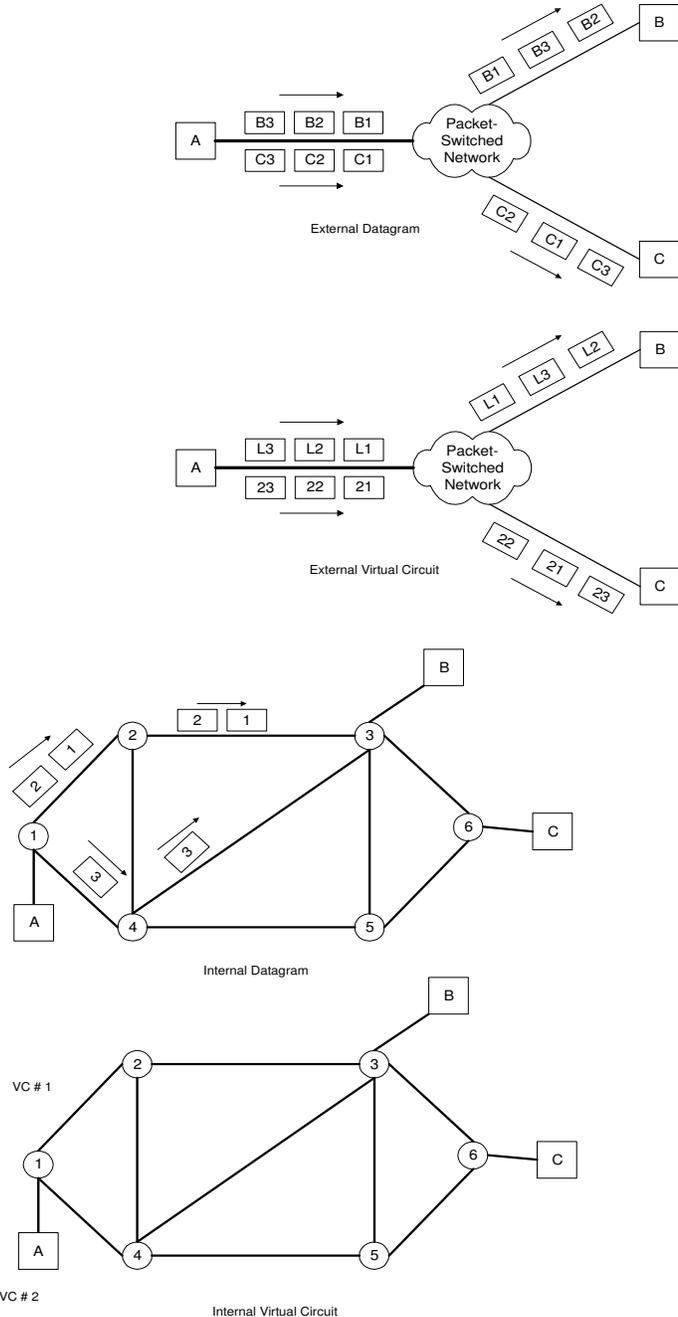
Gambar 8.2 Perbedaan Ukuran-Ukuran Paket

### 8.6. Operasi Internal dan External Service

Hal terpenting dalam *packet-switched network* adalah pemilihan dalam menggunakan *datagram* atau *virtual circuit*. Pada interface antara sebuah station dengan sebuah node network, network harus menyediakan pelayanan *connection-oriented* dan *connection-less*. Pada *connection-oriented*, sebuah station melakukan *call request* untuk membentuk sebuah *logical connection* ke station yang lain. Semua paket yang disajikan ke dalam network diidentifikasi kepunyaan *logical connection* tertentu dan diberi nomor secara berurut.

*Logical connection* biasanya merujuk pada sebuah pelayanan *external virtual circuit* yang jauh berbeda dari konsep operasi *internal virtual circuit*. Sedangkan pada pelayanan *connectionless*, jaringan hanya menangani paket secara *independent* dan mungkin tidak ditransmisikan secara berurut. Tipe service seperti ini dikenal dengan nama *external datagram service* yang juga jauh berbeda dari konsep operasi *internal datagram service*. Secara internal, jaringan akan membuat route antara *endpoints (virtual circuit)* atau tidak (*datagram*).

- *External virtual circuit, internal virtual circuit* : Jika user meminta *virtual circuit*, sebuah *dedicated route* yang melintasi dalam jaringan akan dibangun. Semua paket mengikuti route yang sama.
- *External virtual circuit, internal datagram* : Jaringan menangani tiap paket secara terpisah. Jadi, paket-paket yang berbeda dalam *external virtual circuit* yang sama akan mengambil route yang mungkin berbeda.
- *External datagram, internal datagram* : Tiap paket diperlakukan secara bebas dari segi user atau dari segi jaringannya.
- *External datagram, internal virtual circuit*



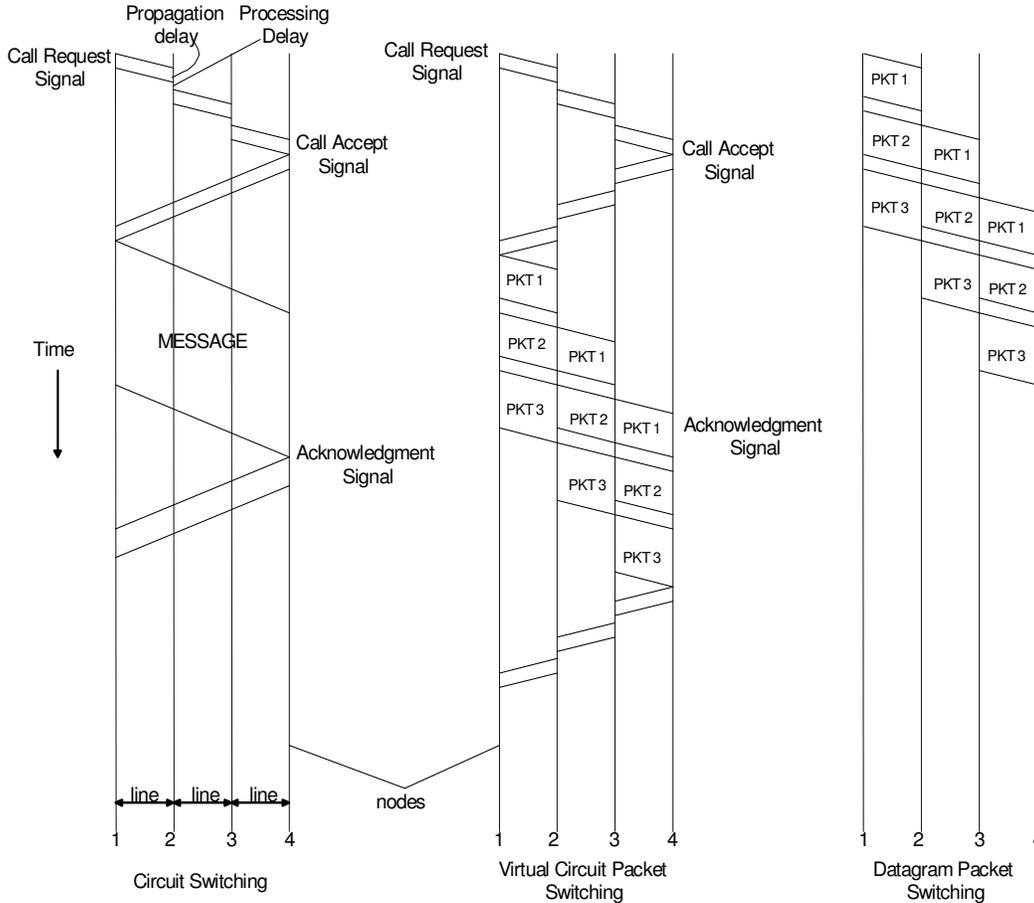
Gambar 8.3 Perbedaan antara External dan Internal Operation

Pemilihan akan *virtual circuit* dengan *datagram* tergantung dari desain objek untuk komunikasi jaringan dan faktor-faktor *cost* secara detailnya.

Untuk *external service* :

- *datagram service* memberikan penggunaan yang efisien dari jaringan dimana tidak ada *call setup*. Ini akan cocok untuk penggunaan beberapa aplikasi real time.
- *virtual circuit service* dapat menyediakan *end-to-end sequencing* dan *error control*. Ini akan cocok untuk aplikasi seperti *file transfer* dan *remote access terminal*.

Perbandingan antara *Circuit Switching* dengan *Packet Switching* :



**Gambar 8.1** Perbedaan antara *Circuit Switching* dengan *Packet Switching*

Pada gambar 8.1 dimisalkan ada 4 node, node 1 sebagai *source address* dan node 4 sebagai *destination address*. Untuk *circuit switching* ada sejumlah delay sebelum *message* dikirim, yaitu untuk *call request*, lalu jika *destination station* tidak sibuk, sinyal *accepted* dikirim dari *destination address*. Proses ini tidak berlangsung setelah koneksi telah disetup. *Virtual circuit switching* hampir sama dengan *circuit switching*. Berbeda dengan *circuit switching*, *call acceptance* akan memakan waktu (delay) walaupun koneksi telah *established*. Hal itu karena paket itu mengalami antrian dan harus menunggu untuk retransmisi. Sekali *virtual circuit* established, *message* akan dikirim dalam bentuk paket-paket. Maka *virtual circuit* tidak akan lebih cepat dari *circuit switching*.

*Datagram packet switching* tidak membutuhkan *call setup*. Jadi untuk *message* pendek akan lebih cepat dari *virtual circuit packet switching* dan mungkin juga *circuit switching*. Selama tiap *datagram* diroute secara bebas, proses untuk tiap *datagram* di tiap node mungkin lebih panjang dari paket-paket *virtual circuit*. Jadi untuk *message* yang panjang-panjang, teknik *virtual circuit* mungkin diutamakan.

**Tabel 8.1.** Perbedaan antara Circuit Switching dengan Packet Switching

<b>Circuit Switching</b>	<b>Virtual-Circuit Packet Switching</b>	<b>Datagram Packet Switching</b>
Dedicated transmission path	No dedicated path	No dedicated path
Continous transmission of data	Transmission of packets	Transmission of packets
Fast enough for interactive	Fast enough for interactive	Fast enough for interactive
Messages are not stored	Packets stored until delivered	Packets may be stored until delivered
The path is established for entire conversation	Route established for entire conversation	Route established for each packet
Call setup delay; negligible transmission delay	Call setup delay; packet transmission delay	Packet transmission delay
Busy signal if called party busy	Sender notified of connection denial	Sender may be notified if packet not delivered
Overload may block call setup; no delay for established calls	Overload may block call setup; increases packet delay	Overload increases packet delay
User responsible for message loss protection	Network may be responsible for packet sequences	Network may be responsible for individual packets
Usually no speed or code conversion	Speed and code conversion	Speed and code conversion
Fixed-bandwidth transmission	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call setup	Overhead bits in each packet	Overhead bits in each packet