

BAB II

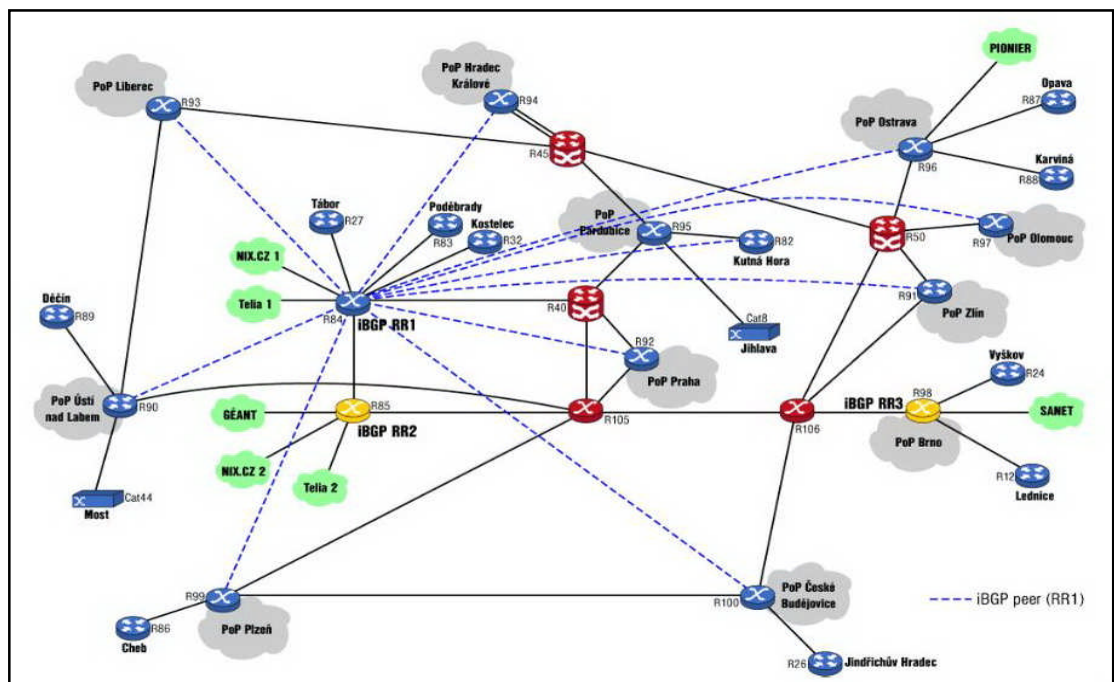
LANDASAN TEORI

2.1 Router

2.1.1 Pengertian Router

Internetwork menggunakan proses *routing* untuk mengirimkan data dari suatu *network* ke *network* yang lain. Untuk menjaga data di dalam jalan yang terbaik ke suatu tujuan, beberapa urutan di dalam *network* sangatlah dibutuhkan.

Route network dalam proses pengiriman data diatasi oleh *protocol routing*. LAN (*Local Area Network*) mempunyai suatu batas *performance* yang bergantung pada ukuran atau kompleksitas dari LAN tersebut.



Gambar 2.1 Network Routing

Data-data dari *device* yang terhubung ke Internet dikirim dalam bentuk datagram, yaitu paket data yang didefinisikan oleh IP. Datagram memiliki alamat tujuan paket data, Internet *Protocol* memeriksa alamat ini untuk menyampaikan datagram dari *device* asal ke *device* tujuan. Jika alamat tujuan datagram tersebut terletak satu jaringan dengan *device* asal, datagram langsung disampaikan kepada

device tujuan tersebut. Jika ternyata alamat tujuan datagram tidak terdapat di jaringan yang sama, datagram disampaikan kepada *router* yang paling tepat (*the best available router*).

IP *Router* (biasa disebut *router* saja) adalah *device* yang melakukan fungsi meneruskan datagram IP pada lapisan jaringan. *Router* memiliki lebih dari satu antarmuka jaringan (*network interface*) dan dapat meneruskan datagram dari satu antarmuka ke antarmuka yang lain. Untuk setiap datagram yang diterima, *router* memeriksa apakah datagram tersebut memang ditujukan ke dirinya. Jika ternyata ditujukan kepada *router* tersebut, datagram disampaikan ke lapisan *transport*.

Jika datagram tidak ditujukan kepada *router* tersebut, yang akan diperiksa adalah *forwarding table* yang dimilikinya untuk memutuskan ke mana seharusnya datagram tersebut ditujukan. *Forwarding table* adalah tabel yang terdiri dari pasangan alamat IP (alamat *host* atau alamat jaringan), alamat *router* berikut, dan antarmuka tempat keluar datagram.

Jika tidak menemukan sebuah baris pun dalam *forwarding table* yang sesuai dengan alamat tujuan, *router* akan memberikan pesan kepada pengirim bahwa alamat yang dimaksud tidak dapat dicapai. Kejadian ini dapat dianalogikan dengan pesan "kembali ke pengirim" pada pos biasa. Sebuah *router* juga dapat memberitahu bahwa dirinya bukan *router* terbaik ke suatu tujuan, dan menyarankan penggunaan *router* lain. Dengan ketiga fungsi yang terdapat pada *router* ini, *host-host* di Internet dapat saling terhubung.

2.1.2 Static Dan Dynamic

Secara umum mekanisme koordinasi *routing* dapat dibagi menjadi dua: *routing* statik dan *routing* dinamik. Pada *routing* statik, *entries* dalam *forwarding table* *router* diisi dan dihapus secara manual, sedangkan pada *routing* dinamik perubahan dilakukan melalui protokol *routing*. *Routing* statik adalah pengaturan *routing* paling sederhana yang dapat dilakukan pada jaringan komputer. Menggunakan *routing* statik murni dalam sebuah jaringan berarti mengisi setiap *entri* dalam *forwarding table* di setiap *router* yang berada di jaringan tersebut.

Penggunaan *routing statik* dalam sebuah jaringan yang kecil tentu bukanlah suatu masalah; hanya beberapa entri yang perlu diisi pada *forwarding table* di

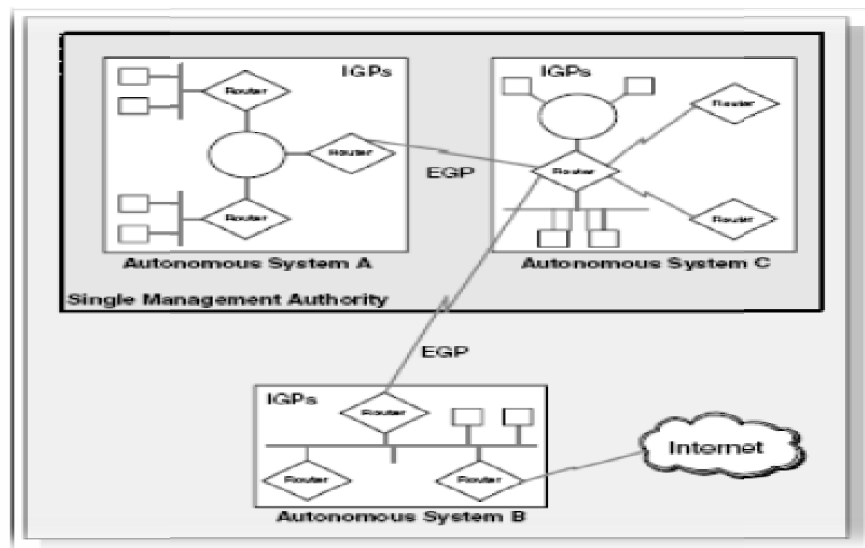
setiap *router*. Namun Anda tentu dapat membayangkan bagaimana jika harus melengkapi *forwarding table* di setiap *router* yang jumlahnya tidak sedikit dalam jaringan yang besar. Apalagi jika Anda ditugaskan untuk mengisi *entries* di seluruh *router* di Internet yang jumlahnya banyak sekali dan terus bertambah setiap hari, tentu akan sangat sulit.

Routing dinamik adalah cara yang digunakan untuk melepaskan kewajiban mengisi *entries forwarding table* secara manual. Protokol *routing* mengatur semua *router* sehingga dapat berkomunikasi satu dengan yang lain dan saling memberikan informasi *routing* yang dapat mengubah isi *forwarding table*, tergantung keadaan jaringannya. Dengan cara ini, semua *router* mengetahui keadaan jaringan yang terakhir dan mampu meneruskan datagram ke arah yang benar.

2.1.3 Interior Routing Protokol

Pada awal 1980-an Internet terbatas pada ARPANET, Satnet (perluasan ARPANET yang menggunakan satelit) dan beberapa jaringan lokal yang terhubung lewat *gateway*. Dalam perkembangannya, Internet memerlukan struktur yang bersifat hirarkis untuk mengantisipasi jaringan yang telah menjadi besar. Internet kemudian dipecah menjadi beberapa *Autonomous System* (AS) dan saat ini Internet terdiri dari ribuan AS. Setiap AS memiliki mekanisme pertukaran dan pengumpulan informasi *routing* sendiri.

Protokol yang digunakan untuk bertukar informasi *routing* dalam AS digolongkan sebagai *interior routing protocol* (IRP). Hasil pengumpulan informasi *routing* ini kemudian disampaikan kepada AS lain dalam bentuk *reachability information*. *Reachability information* yang dikeluarkan oleh sebuah AS berisi informasi mengenai jaringan-jaringan yang dapat dicapai melalui AS tersebut dan menjadi indikator terhubungnya AS ke Internet. Penyampaian *reachability information* antar AS dilakukan menggunakan protokol yang digolongkan sebagai *exterior routing protocol* (ERP).



Gambar 2.2 Autonomous System (AS)

IRP yang dijadikan standar di Internet sampai saat ini adalah *Routing Information Protocol* (RIP) dan *Open Shortest Path First* (OSPF). Di samping kedua protokol ini terdapat juga *protokol routing* yang bersifat *proprietary* tetapi banyak digunakan di Internet, yaitu *Internet Gateway Routing Protocol* (IGRP) dari Cisco System. Protokol IGRP kemudian diperluas menjadi *Extended IGRP* (EIGRP). Semua *protokol routing* di atas menggunakan metrik sebagai dasar untuk menentukan jalur terbaik yang dapat ditempuh oleh datagram. Metrik diasosiasikan dengan "biaya" yang terdapat pada setiap *link*, yang dapat berupa *throughput* (kecepatan data), *delay*, biaya sambungan, dan keandalan *link*.

Dalam Tugas akhir ini akan dibahas *protocol routing* EIGRP adalah yang termasuk *proprietary* Cisco, yang berarti hanya bisa dijalankan pada *router* Cisco, EIGRP bisa jadi merupakan *protokol routing* terbaik didunia jika bukan merupakan *proprietary* Cisco.

Kelebihan utama yang membedakan EIGRP dari *protokol routing* lainnya adalah EIGRP termasuk satu-satunya *protokol routing* yang menawarkan fitur *backup route*, dimana jika terjadi perubahan pada *network*, EIGRP tidak harus melakukan kalkulasi ulang untuk menentukan *route* terbaik karena bisa langsung menggunakan *backup route*.

2.2 EIGRP

2.2.1 Pengertian EIGRP

Enhanced Interior Gateway Routing Protocol (EIGRP) adalah sebuah routing protokol proprietary Cisco longgar didasarkan pada IGRP asli. EIGRP termasuk jenis protokol distance-vector canggih, dengan optimasi meminimalkan ketidakstabilan routing yang terjadi setelah perubahan topologi, serta penggunaan kekuatan bandwidth di router. EIGRP secara otomatis mendukung router yang akan mendistribusikan informasi ke router tetangga dengan mengubah metrik EIGRP 32 bit ke 24 bit metrik IGRP. Kebanyakan dari optimisasi routing didasarkan pada penyebaran Update Algoritma (DUAL) yang menjamin operasi bebas looping dan menyediakan suatu mekanisme untuk konvergensi cepat. EIGRP adalah protokol routing yang termasuk proprietary Cisco, yang berarti hanya bisa dijalankan pada router Cisco, EIGRP bisa jadi merupakan protokol routing terbaik didunia jika bukan merupakan proprietary Cisco.

Dahulu EIGRP digambarkan dalam materi pemasaran berbagai Cisco sebagai hibrid seimbang routing protokol, Hal ini dikarenakan EIGRP menggabungkan fitur terbaik dari link state dan protokol distance-vektor. Menurut definisi, protokol routing adalah jarak vektor didasarkan pada bentuk distribusi dari algoritma Bellman-Ford untuk menemukan jalan terpendek. Mereka bekerja dengan menukarkan vektor dari jarak ke semua tujuan yang diketahui pada setiap node. Tidak ada informasi lebih lanjut dari topologi yang pernah dipertukarkan. Dengan demikian, setiap node mengetahui tentang semua tujuan dalam jaringan dan jarak ke tempat tujuan.

Classful Routing Protocol adalah penerapan subnet secara penuh atau default /24, /16, dan /8 artinya penggunaan kelas full dikonsep ini. Classful routing protocols juga ialah suatu protokol dimana protokol ini tidak membawa routing mask information ketika update routing atau routing advertisements. Ia hanya membawa informasi IP address saja, dan menggunakan informasi default mask sebagai mask-nya. Dynamic routing Classfull : Rip V1, IGRP. Classfull merupakan metode pembagian IP address berdasarkan kelas dimana IP address (yang berjumlah sekitar 4 milyar) dibagi kedalam lima kelas yakni :

1. *Address kelas A : 1 bit pertama IP Address-nya "0"*
2. *Address kelas B : 2 bit pertama IP Address-nya "10"*
3. *Address kelas C : 3 bit pertama IP Address-nya "110"*
4. *Address kelas D : 4 bit pertama IP Address-nya "1110"*
5. *Address kelas E : 4 bit pertama IP Address-nya "1111"*

Kelemahan dari classful routing protocols ialah tak dapat men-support VLSM.

Classless Routing Protocol artinya kita dapat menggunakan semua subnet yang dapat digunakan tetapi tidak dapat menggunakan metode VLSM pada penerapannya. Dynamic routing Classless : IS-IS, Rip V2, OSPF, EIGRP, dan BGP.

2.2.2 Kelebihan EIGRP

Kelebihan utama yang membedakan EIGRP dari protokol routing lainnya adalah EIGRP termasuk satu-satunya protokol routing yang menawarkan fitur backup route, dimana jika terjadi perubahan pada network, EIGRP tidak harus melakukan kalkulasi ulang untuk menentukan route terbaik karena bisa langsung menggunakan backup route. Kalkulasi ulang route terbaik dilakukan jika backup route juga mengalami kegagalan.

Berikut adalah fitur-fitur kelebihan yang dimiliki EIGRP :

1. *Waktu convergence yang cepat.*
2. *Mendukung VLSM (Variable Length Subnet Masking) dan subnet-subnet yang discontinuous (tidak bersebelahan/berurutan).*
3. *Partial updates, tidak seperti RIP yang selalu mengirimkan keseluruhan tabel routing dalam pesan Update, EIGRP menggunakan partial updates atau triggered update yang berarti hanya mengirimkan update jika terjadi perubahan pada network (misal: ada network yang down).*
4. *Termasuk protokol routing distance vector tingkat lanjut (Advanced Distance Vector).*
5. *Summarization dapat dilakukan dimana saja dan kapan saja. Pada OSPF summarization hanya bisa dilakukan di ABR dan ASBR.*
6. *Mendukung multiple protokol network.*
7. *Desain network yang fleksibel.*

8. *Multicast dan unicast, EIGRP saling berkomunikasi dengan tetangga (neighbor) nya secara multicast (224.0.0.10) dan tidak mem-broadcast-nya.*
9. *Manual summarization, EIGRP dapat melakukan summarization dimana saja.*
10. *Menjamin 100% topologi routing yang bebas looping.*
11. *Mudah dikonfigurasi untuk WAN dan LAN.*
12. *Load balancing via jalur dengan cost equal dan unequal, yang berarti EIGRP dapat menggunakan 2 link atau lebih ke suatu network destination dengan koneksi bandwidth (cost metric) yang berbeda, dan melakukan load sharing pada link-link tersebut dengan beban yang sesuai yang dimiliki oleh link masing-masing, dengan begini pemakaian bandwidth pada setiap link menjadi lebih efektif, karena link dengan bandwidth yang lebih kecil tetap digunakan dan dengan beban yang sepadan juga.*

EIGRP mengkombinasikan kelebihan-kelebihan yang dimiliki oleh protokol routing link-state dan distance vector. Tetapi pada dasarnya EIGRP adalah protokol distance vector karena router-router yang menjalankan EIGRP tidak mengetahui road map/topologi network secara menyeluruh seperti pada protokol link-state.

EIGRP mudah dikonfigurasi seperti pendahulunya (IGRP) dan dapat diadaptasikan dengan variasi topologi network. Penambahan fitur-fitur protokol link-state seperti neighbor discovery membuat EIGRP menjadi protokol distance vector tingkat lanjut.

EIGRP menggunakan DUAL (Diffusing Update Algorithm) sebagai mesin utama yang menjalankan lingkungan EIGRP, DUAL, dan dapat dibandingkan dengan algoritma SPF Dijkstra pada OSPF.

Fitur – fitur utama EIGRP adalah :

1. *Partial updates : EIGRP tidak mengirimkan update secara periodik seperti yang dilakukan oleh RIP, tetapi EIGRP mengirimkan update hanya jika terjadi perubahan route/metric (triggered update). Update*

yang dikirimkan hanya berisi informasi tentang *route* yang mengalami perubahan saja. Pengiriman pesan *update* ini juga hanya ditujukan sebatas pada semua *router* yang membutuhkan informasi perubahan tersebut saja. Hasilnya EIGRP menghabiskan *bandwidth* yang lebih sedikit daripada IGRP. Hal ini juga membedakan EIGRP dengan protokol *link-state* yang mengirimkan *update* kepada semua *router* dalam satu area.

2. Mendukung *multiple* protokol *network*: EIGRP mendukung protokol IP, AppleTalk, dan Novell NetWare IPX dengan memanfaatkan modul-modul yang tidak bergantung pada protokol tertentu.

Fitur EIGRP lain yang patut diperhatikan adalah sebagai berikut:

1. Koneksi dengan semua jenis data *link* dan topologi tanpa memerlukan konfigurasi lebih lanjut, protokol *routing* lain seperti OSPF, menggunakan konfigurasi yang berbeda untuk protokol *layer 2* (Data *Link*) yang berbeda, misalnya *Ethernet* dan *Frame Relay*. EIGRP beroperasi dengan efektif pada lingkungan LAN dan WAN. Dukungan WAN untuk *link point-to-point* dan topologi *non-broadcast multiaccess* (NBMA) merupakan standar EIGRP.
2. *Metric* yang canggih: EIGRP menggunakan algoritma yang sama dengan IGRP untuk menghitung metrik tetapi menggambarkan nilai-nilai dalam format 32-bit. EIGRP mendukung *load balancing* untuk metrik yang tidak seimbang (*unequal*), yang memungkinkan *engineer* untuk mendistribusikan trafik dalam *network* dengan lebih baik.
3. *Multicast and unicast*: EIGRP menggunakan multicast dan unicast sebagai ganti broadcast. *Address multicast* yang digunakan adalah 224.0.0.10.

2.2.3 Metric yang digunakan oleh EIGRP

Protokol *routing* digolongkan berdasarkan cara mereka memilih jalur terbaik dan cara mereka menghitung *metric* (metrik) suatu jalur (*route*). *Metric* adalah suatu ukuran yang digunakan untuk menentukan nilai *cost* dari suatu *route*

menuju *network* tertentu. Semakin kecil metrik suatu *route network* semakin bagus dan akan menjadi pilihan utama dalam pemilihan *route* terbaik.

EIGRP menggunakan metrik yang canggih, yaitu menggunakan algoritma yang sama dengan IGRP untuk menghitung metrik tetapi menggambarkan nilai-nilai dalam format 32-bit. EIGRP mendukung load balancing untuk metrik yang tidak seimbang (unequal), yang memungkinkan engineer untuk mendistribusikan trafik dalam network dengan lebih baik.

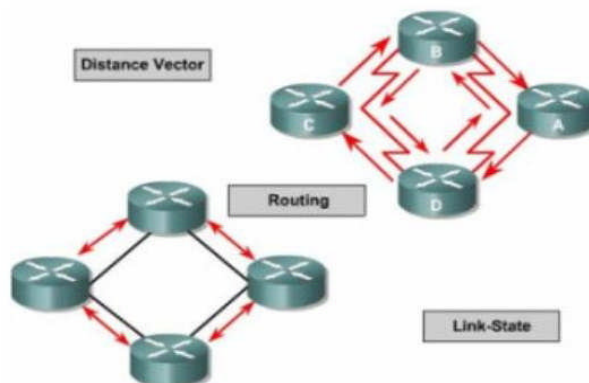
EIGRP menggunakan gabungan metrik yang sama seperti pada IGRP untuk menentukan jalur terbaik, hanya saja metrik EIGRP dikalikan 256. EIGRP secara *default* hanya menggunakan 2 kriteria metrik berikut:

1. *Bandwidth.*
2. *Delay : total lama delay interface sepanjang jalur.*

2.2.4 Algoritma Routing yang Digunakan EIGRP :

Klasifikasi Algoritma Routing :

1. *Global yang artinya :*
Semua router memiliki informasi lengkap mengenai topologi link cost. Contohnya adalah algoritma link-state.
2. *Desentralisasi yang artinya :*
 - a. *Router mengetahui koneksi fisik atau link cost ke tetangga,*
 - b. *Terjadi pengulangan proses komputasi dan mempertukarkan,*
 - c. *informasinya ke router tetangganya, contohnya adalah algoritma distance vector.*



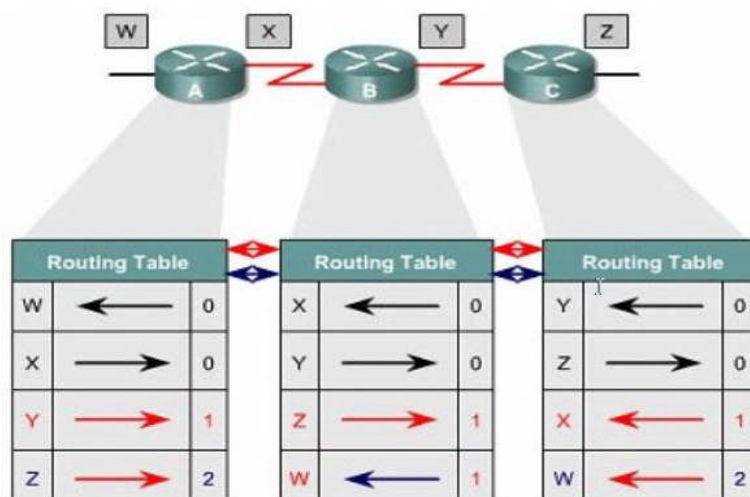
Gambar 2.3 Algoritma Routing

2.2.4.1 Distance Vector

Algoritma *routing distance vector* secara periodik menyalin *table routing* dari *router* ke *router*. Perubahan *table routing* ini di-*update* antar *router* yang saling berhubungan saat terjadi perubahan topologi. Setiap *router* menerima *table routing* dari *router* tetangga yang terhubung secara langsung. Proses *routing* ini disebut juga dengan *routing* Bellman-Ford atau Ford-Fulkerson. *Routing* vektor jarak beroperasi dengan membiarkan setiap *router* menjaga tabel (sebuah vektor) memberikan jarak yang terbaik yang dapat diketahui ke setiap tujuan dan saluran yang dipakai menuju tujuan tersebut. Tabel-tabel ini di-*update* dengan cara saling bertukar informasi dengan *router* tetangga.

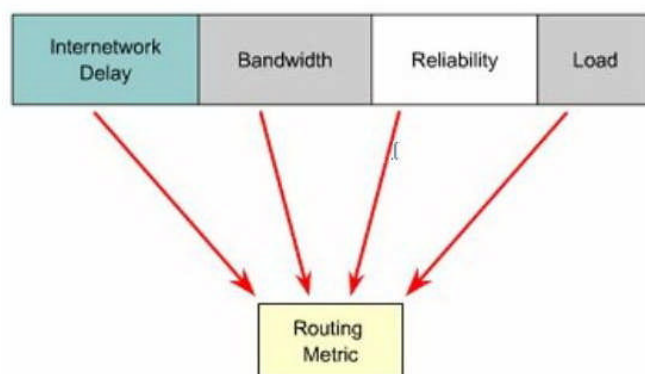
Routing distance vector bertujuan untuk menentukan arah atau vektor dan jarak ke *link* lainnya di suatu *internetwork*. Sedangkan *link-state* bertujuan untuk menciptakan kembali topologi yang benar pada suatu *internetwork*. Misal, *router* Y menerima tabel informasi estimasi dari *router* X, dimana terdapat X_i , yang menyatakan estimasi waktu yang dibutuhkan oleh X untuk sampai ke *router* i. Bila Y mengetahui *delay* ke X sama dengan m milidetik, Y juga mengetahui bahwa Y dapat mencapai *router* i dalam $X_i + m$ milidetik. Struktur data tabel Distance Vector :

1. Setiap *node (router)* memilikinya,
2. Baris digunakan menunjukkan tujuan yang mungkin,
3. Kolom digunakan menunjukkan untuk setiap *node* tetangga secara langsung,
4. Sebagai contoh : pada *router* X, untuk tujuan Y melalui tetangga Z.
5. Pembentukan tabel *routing* dilakukan dengan cara tiap-tiap *router* saling bertukar informasi *routing* dengan *router* yang terhubung secara langsung.
6. Proses pertukaran informasi dilakukan secara periodik, misal setiap 45 detik.



Gambar 2.4 Proses Petukaran Informasi

Update table routing dilakukan ketika terjadi perubahan topologi jaringan. Sama dengan proses discovery, proses *update* perubahan topologi step-by-step dari *router* ke *router*. Gambar diatas menunjukkan algoritma distance vector memanggil ke semua *router* untuk mengirim ke isi table *routing*nya. Table *routing* berisi informasi tentang total path cost yang ditentukan oleh metric dan alamat logic dari *router* pertama dalam jaringan yang ada di isi table *routing*, seperti skema oleh gambar di bawah ini.



Gambar 2.5 Tabel Routing Metric

Analogi *distance vector* dapat dianalogikan dengan jalan tol. Tanda yang menunjukkan titik ke tujuan dan menunjukkan jarak ke tujuan. Dengan adanya

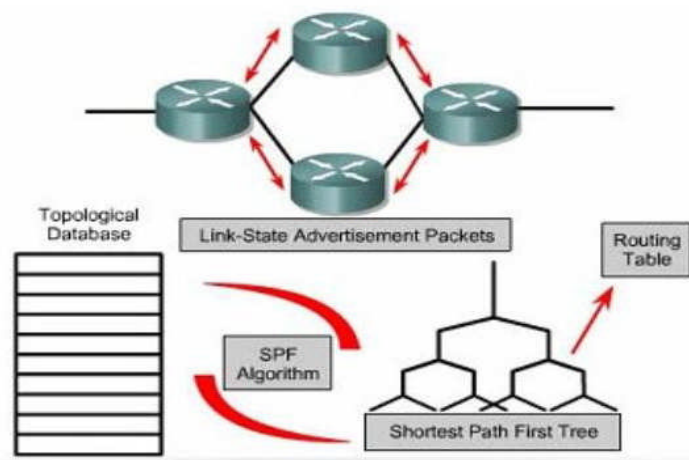
tanda-tanda seperti itu pengendara dapat dengan mudah mengetahui perkiraan jarak yang akan ditempuh untuk mencapai tujuan. Dan tentunya jarak terpendek adalah rute yang terbaik.

Analogi *distance vector* dapat dianalogikan dengan jalan tol. Tanda yang menunjukkan titik ke tujuan dan menunjukkan jarak ke tujuan. Dengan adanya tanda-tanda seperti itu pengendara dapat dengan mudah mengetahui perkiraan jarak yang akan ditempuh untuk mencapai tujuan. Dan tentunya jarak terpendek adalah rute yang terbaik.

2.2.4.2 *Link-state*

Algoritma *link-state* juga dikenal dengan algoritma Dijkstra atau algoritma *shortest path first* (SPF). Algoritma ini memperbaiki informasi *database* dari informasi topologi. Algoritma *distance vector* memiliki informasi yang tidak spesifik tentang *distance network* dan tidak mengetahui jarak *router*. Sedangkan algoritma *link-state* memperbaiki pengetahuan dari jarak *router* dan bagaimana mereka inter-koneksi. Beberapa fitur yang dimiliki oleh *routing link-state* adalah:

1. *Link-state advertisement* (LSA) – paket kecil dari informasi *routing* yang dikirim antar *router*.
2. *Topological database* – kumpulan informasi yang dari LSA-LSA.
3. *SPF algorithm* – hasil perhitungan pada *database* sebagai hasil dari pohon SPF.
4. *Routing table* – adalah daftar rute dan *interface*.



Gambar 2.6 Topologi *Link-state Routing*

Konsep *Link-state* dimana dasar algoritma *routing* yang lain adalah algoritma *link-state*. Algoritma *link-state* biasa disebut sebagai algoritma Dijkstra atau algoritma *Shortest Path First* (SPF).

1. Setiap *router* mempunyai peta jar,
2. *Router* menentukan rute ke setiap tujuan di jar berdasarkan peta jar tersebut,
3. Peta jaringan disimpan *router* dalam bentuk *database* sebagai hasil dari pertukaran info *link-state* antara *router-router* bertetangga di jar tersebut,
4. Setiap record dalam *database* menunjukkan status sebuah jalur dijar (*link-tate*),
5. Menerapkan algoritma Dijkstra,
6. Topologi jaringan dan *link cost* diketahui oleh semua *node router*,
7. Dilakukan dengan cara mem-broadcast informasi *link-state*,
8. Semua *node* memiliki informasi yang sama,
9. Menghitung cost terkecil dari satu *node* ke *node* lainnya,
10. Memberikan tabel rute untuk *router* tersebut setelah iterasi sebanyak n, diketahui *link cost* terkecil untuk n tujuan.

2.2.5 Cara Kerja EIGRP

Istilah-istilah algoritma DUAL

1. Memilih jalur/*route* untuk mencapai suatu *network* dengan ongkos paling rendah, dan bebas looping.
2. AD (*advertised distance*), menggambarkan seberapa jauh sebuah *network* dari *neighbour*, merupakan ongkos (*mteric*) antara *router next-hop* dengan *network destination*.
3. FD (*feasible distance*), menggambarkan seberapa jauh sebuah *network* dari *router*, merupakan ongkos (*metric*) antara *router* dengan *router next-hop* ditambah dengan AD dari *router next-hop*.
4. Ongkos paling rendah = FD paling rendah.

5. *Successor*, adalah jalur utama untuk mencapai suatu *network* (*route* terbaik), merupakan *router* next-hop dengan Ongkos paling rendah dan jalur bebas *looping*.
6. *Feasible Successor*, adalah jalur backup dari *successor* (AD dari *feasible successor* harus lebih kecil daripada FD dari *successor*)

EIGRP menggunakan dan memelihara 3 jenis tabel. Tabel *neighbour* untuk mendaftarkan semua *router* *neighbour*, tabel topologi untuk mendaftarkan semua entri *route* untuk setiap *network destination* yang didapatkan dari setiap *neighbour*, dan tabel *routing* yang berisi jalur/*route* terbaik untuk mencapai ke setiap *destination*.

2.2.5.1 Tabel Neighbour

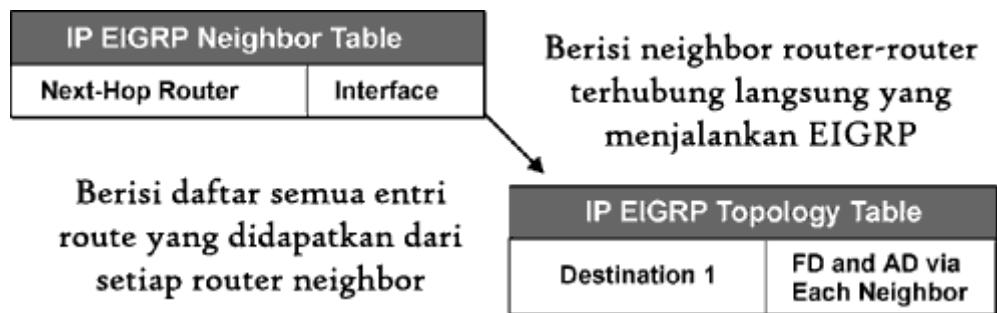
IP EIGRP Neighbor Table	
Next-Hop Router	Interface

Gambar 2.7 Neighbour Table

Ketika *router* menemukan dan menjalin hubungan *adjacency* (ketetanggaan) dengan *neighbour* baru, maka *router* akan menyimpan *address router neighbour* beserta *interface* yang dapat menghubungkan dengan *neighbour* tersebut sebagai satu entri dalam tabel *neighbour*. Tabel *neighbour* EIGRP dapat diperbandingkan dengan *database adjacency* yang digunakan oleh protokol *routing link-state* yang keduanya mempunyai tujuan yang sama: untuk melakukan komunikasi 2 arah dengan setiap *neighbour* yang terhubung langsung.

Ketika *neighbour* mengirimkan paket *hello*, ia akan menyertakan informasi *hold time*, yakni total waktu sebuah *router* dianggap sebagai *neighbour* yang dapat dijangkau dan operasional. Jika paket *hello* tidak diterima sampai *hold time* berakhir, algoritma DUAL akan menginformasikan terjadinya perubahan topologi.

2.2.5.2 Table Topologi



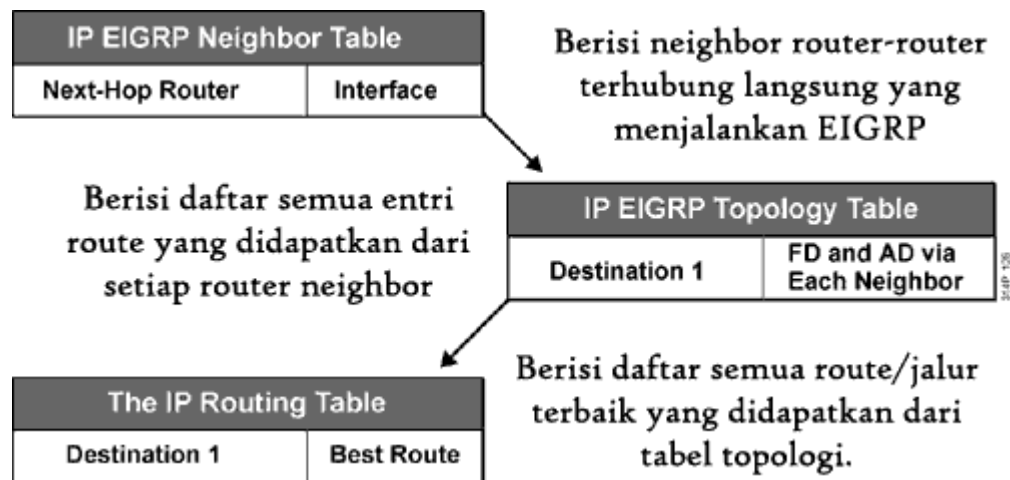
Gambar 2.8 Topologi Table

Ketika *router* menemukan *neighbour* baru, maka *router* akan mengirimkan sebuah *update* mengenai *route-route* yang ia ketahui kepada *neighbour* baru tersebut dan juga sebaliknya menerima informasi yang sama dari *neighbour*. *Update-update* ini lah yang akan membangun tabel topologi. Tabel topologi berisi informasi semua *network destination* yang di advertise oleh *router neighbour*. Jika *neighbour* meng-advertise *route* ke suatu *network destination*, maka *neighbour* tersebut harus menggunakan *route* tersebut untuk meneruskan paket.

Tabel topologi di *update* setiap kali ada perubahan pada *network* yang terhubung langsung atau pada *interface* atau ada pemberitahuan perubahan pada suatu jalur dari *router neighbour*.

Entri pada tabel topologi untuk suatu *destination* dapat berstatus aktif atau pasif. *Destination* akan berstatus pasif jika *router* tidak melakukan komputasi ulang, dan berstatus aktif jika *router* masih melakukan komputasi ulang. Jika selalu tersedia *feasible successor* maka *destination* tidak akan pernah berada pada status aktif dan terhindar dari komputasi ulang. Status yang diharapkan untuk setiap *network destination* adalah status *pasif*.

2.2.5.3 Routing Table



Gambar 2.9 Routing Table

Router akan membandingkan semua FD untuk mencapai *network* tertentu dan memilih jalur/*route* dengan FD paling rendah dan meletakkannya pada tabel *routing*, jalur/*route* inilah yang disebut *successor route*. FD untuk jalur/*route* yang terpilih akan menjadi metrik EIGRP untuk mencapai *network* tersebut dan disertakan dalam tabel *routing*.

2.2.6 Paket-Paket EIGRP

EIGRP saling berkomunikasi dengan tetangganya secara *multicast* (224.0.0.10) dan menggunakan 5 jenis pesan (*message*) dalam berhubungan dengan *neighbour*-nya :

1. **Hello:** Router-Router menggunakan paket Hello untuk menjalin hubungan *neighbour*. Paket-paket dikirimkan secara *multicast* dan tidak membutuhkan.
2. **Update:** Untuk mengirimkan *update* informasi *routing*. Tidak seperti RIP yang selalu mengirimkan keseluruhan tabel *routing* dalam pesan *Update*, EIGRP menggunakan *triggered update* yang berarti hanya mengirimkan *update* jika terjadi perubahan pada *network* (mis: ada *network* yang down). Paket *update* berisi informasi perubahan jalur/*route*. *Update-update* ini dapat berupa *unicast* untuk *router* tertentu atau *multicast* untuk beberapa *router* yang terhubung.

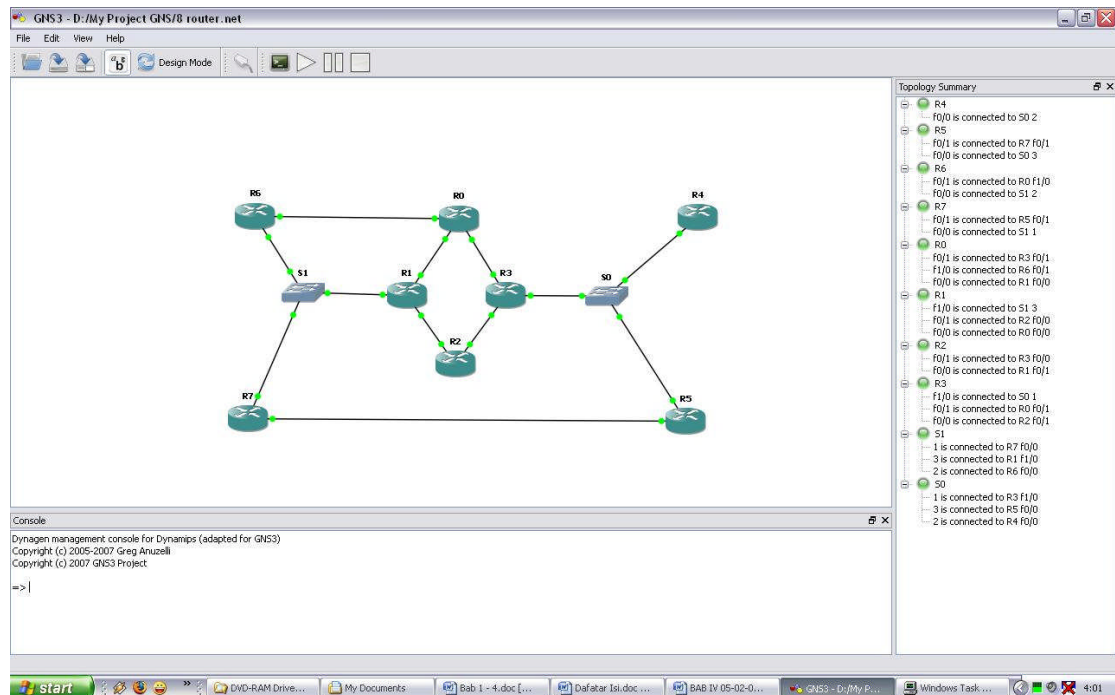
3. **Query:** Untuk menanyakan suatu *route* kepada tetangga. Biasanya digunakan saat setelah terjadi kegagalan/down pada salah satu *route network*, dan tidak terdapat *feasible successor* untuk *route/jalur* tersebut. *router* akan mengirimkan pesan Query untuk memperoleh informasi *route* alternatif untuk mencapai *network* tersebut, biasanya dalam bentuk multicast tapi bisa juga dalam bentuk unicast untuk beberapa kasus tertentu.
4. **Reply:** Respon dari pesan **Query**.
5. **ACK:** Untuk memberikan acknowledgement (pengakuan/konfirmasi) atas pesan *Update*, Query, dan Reply.

2.3 Dynamips + GNS3

Sudah menjadi rahasia umum kalau harga *router* itu sangat mahal buat perorangan, namun untuk sebuah perusahaan hal ini penting dan menjadi investasi yang bisa mendatangkan keuntungan besar dari berbagai efisiensi. Dengan semakin besarnya suatu perusahaan maka kebutuhan IT akan terus meningkat. Terutama kebutuhan *network* untuk mengalirkan system dari satu tempat ke tempat lain. Disinilah peran infrastruktur jaringan yang handal dan *network engineer* yang *capable* diperlukan.

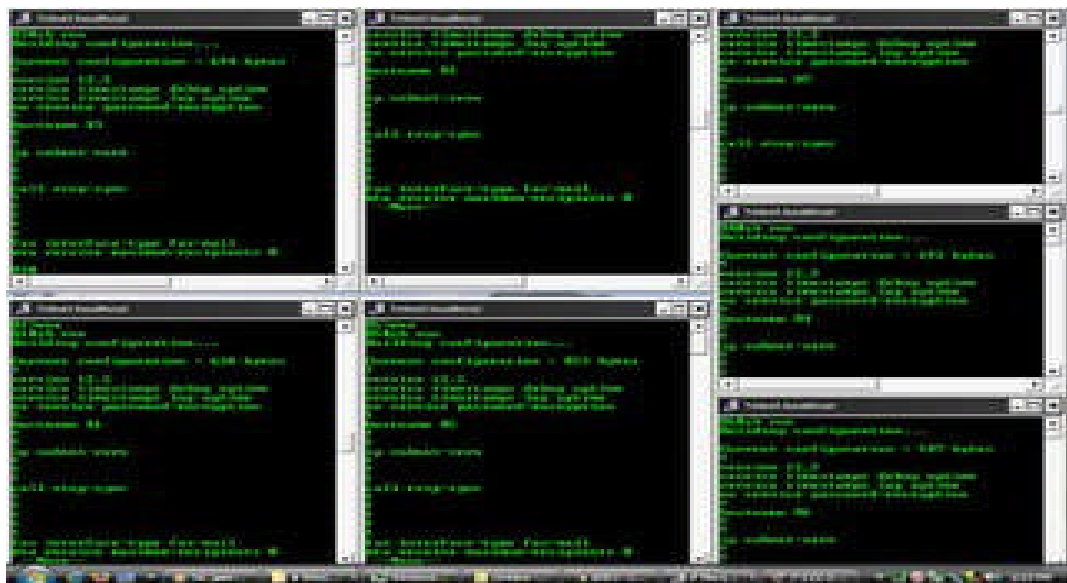
Biasanya infrastruktur *network* itu banyak macamnya, mulai dari *router*, *switch*, hub, FO, UTP sampai RJ45. Disinilah peran seorang *network engineer* untuk mendesain, *maintenance*, *improvement* & *solving problem* jika terjadi hal-hal yang tidak diinginkan. Tapi yang terpenting untuk seseorang yang ingin menjadi *network engineer* adalah bagaimana cara belajar *network* sebelum bekerja di suatu perusahaan. Dengan harga *router* yang mahal maka dibutuhkan suatu simulator yg handal dan mendekati dengan kondisi *real*. Disinilah peran *dynamips* dapat bekerja seolah-olah meng-*console* beberapa *router* sekaligus dengan IOS yang ter-*update* dari berbagai jenis *cisco devices*.

Agar lebih mirip lagi maka diperlukan *software* GUI yang *user friendly* yaitu GNS3. Ini adalah tampilan GNS3. Kita tinggal buat topologi yang diinginkan, kemudian me masukan IOS dari masing2 *device* dan koneksi antar *device*.



Gambar 2.10 Tampilan GNS3

Gambar diatas membuat topologi dari 7 routers. *Notebook* yang digunakan adalah HP NC4400 dengan spesifikasi Processor Core 2 Duo (@ 1,67 GHz), RAM 1,5 Gb, WinXP, HD 60 GB. Untuk mengaktifkan ke 7 router tersebut *notebook* ini cukup handal & bisa beroperasi dengan cepat. Kemudian tampilan dari ke 7 router tersebut dalam CLI seperti dibawah ini.



Gambar 2.11 Tampilan CLI