KAJIAN ALGORITMA ROUTING DALAM JARINGAN KOMPUTER

Doro Edi
Jurusan Sistem Informasi
Fakultas Teknologi Informasi, Universitas Kristen Maranatha
Jl. Prof. Drg. Suria Sumantri No. 65 Bandung 40164
Email: doro.edi@eng.maranatha.edu

Abstract

Routing is the act of moving information across an internetwork from a source to a destination. Routing involves two basic activities: determining optimal routing paths and transporting information groups (typically called packets) through an internetwork. In the context of the routing process, the latter of these is referred to as packet switching. Routing algorithms can be differentiated based on several key characteristics. First, the particular goals of the algorithm designer affect the operation of the resulting routing protocol. Second, various types of routing algorithms exist, and each algorithm has a different impact on network and router resources. Finally, routing algorithms use a variety of metrics that affect calculation of optimal routes

Keywords: routing algorithm, static routing, dynamic routing

1. Pendahuluan

Routing merupakan proses dimana sesuatu dibawa dari satu lokasi ke lokasi lainnya. Contoh riil sesuatu yang membutuhkan perutean adalah surat, panggilan telepon, perjalanan kereta api, dan lain sebagainya. Pada suatu jaringan router adalah perangkat yang digunakan untuk merutekan trafik jaringan.

Untuk dapat melakukan perutean, suatu router, atau entitas apapun yang membangun *routing*, melakukan beberapa langkah berikut ini:

- Mengetahui Alamat tujuan – Ke tujuan (alamat) mana sesuatu yang dirutekan dikirim?
- Mengenali sumber-sumber informasi perutean – Dari sumber-sumber (router-router lain) mana saja suatu router dapat mempelajari jalur-jalur menuju tujuan?
- Menemukan rute-rute – Jalur-jalur atau rute-rute mana saja yang mungkin dapat dilalui untuk mencapai alamat tujuan?
- Memilih jalur atau rute – Memilih jalur atau rute terbaik untuk menuju alamat tujuan yang dimaksud.
- Memelihara dan memverifikasi informasi routing – Apakah jalur-jalur ke tujuan yang telah diketahui masih berlaku dan benar?

Pada suatu sistem jaringan komputer, router mempelajari informasi *routing* dari sumber-sumber *routing*-nya yang terletak di dalam tabel *routing* (*routing table*). Router akan berpedoman pada tabel ini untuk menyatakan port mana yang digunakan mem-forward paket-paket yang ditujukan kepadanya.

- Jika jaringan tujuan terhubung langsung dengan router, maka router sudah mengetahui port mana yang digunakan untuk mem-forward paket.
- Jika jaringan tujuan tidak terhubung langsung dengan router, maka router harus mempelajari rute terbaik untuk mem-forward paket ke tujuan.

2. Static Routing dan Dynamic Routing

Secara umum mekanisme koordinasi routing dapat dipelajari oleh router dalam dua metode, yaitu:

- Dimasukkan secara manual oleh administrator jaringan, disebut *Static Routes*.
- Dikumpulkan melalui proses-proses dinamis yang berjalan di jaringan, disebut sebagai *Dynamic Routes*.

2.1. Static Routing

Routing statik (*static route*) adalah pengaturan routing paling sederhana yang dapat dilakukan pada jaringan komputer. *Static route* adalah rute-rute ke *host* atau jaringan tujuan yang dimasukkan secara manual oleh administrator jaringan ke *route table* suatu router. *Static route* mendefinisikan alamat IP *hop router* berikutnya dan *interface* lokal yang digunakan untuk mem-*forward* paket ke tujuan tertentu (*hop router* berikutnya).

Static route memiliki keunggulan untuk menghemat *bandwidth* jaringan karena *static route* tidak membangkitkan trafik *route update* untuk memberikan informasi perubahan rute yang berlaku (sah) saat ini ke router-router lain. Penggunaan routing statik dalam sebuah jaringan yang kecil tentu bukanlah suatu masalah, hanya beberapa entri yang perlu diisikan pada *forwarding table* di setiap router.

Namun tentu dapat dibayangkan bagaimana jika harus melengkapi *forwarding table* di setiap router yang jumlahnya tidak sedikit dalam jaringan yang besar. Apalagi jika untuk mengisi entri-entri di seluruh router di Internet yang jumlahnya banyak sekali dan terus bertambah setiap hari. Jadi penggunaan *static route* cenderung membutuhkan waktu ekstra ketika memanajemen jaringan. Hal ini disebabkan karena sistem administrator harus secara manual meng-*update route table* setiap terjadi perubahan konfigurasi jaringan.

2.2. Dynamic Routing

Routing dinamik adalah cara yang digunakan untuk melepaskan kewajiban mengisi entri-entri *forwarding table* secara manual. Protokol routing mengatur router-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, router-router mengetahui keadaan jaringan yang terakhir dan mampu meneruskan datagram ke arah yang benar.

Routing dinamik yang popular saat ini mengacu pada dua tipe algoritma yang dikenalkan oleh Bellman Ford dengan algoritma *distance vector*nya dan oleh Djikstra dengan algoritma *link state*nya. Cisco kemudian mengembangkan protocol untuk perangkat routernya yang merupakan gabungan dari kedua algoritma tersebut yang diberi nama protocol EIGRP.

2.2.1. Algoritma Distance Vector

Protokol *distance vector* bekerja dengan memberikan router-router kemampuan untuk mempublikasikan semua rute-rute yang diketahui (router bersangkutan) keluar ke seluruh interface yang dimilikinya.

Router yang secara fisik berada pada jaringan yang sama dinamakan *neighbor*. Jika router-router mempublikasikan rute-rute yang diketahuinya melalui seluruh interface-nya, dan seluruh neighbor menerima routing update, maka setiap

router akan juga mengetahui rute-rute yang dapat dilalui ke seluruh subnet suatu jaringan.

Beberapa hal berikut ini akan lebih mempermudah memahami konsep dasar *distance vector*:

- Router secara otomatis akan menambahkan subnet-subnet yang terhubung langsung ke dalam routing table tanpa menggunakan protokol routing.
- Router mengirim *routing update* keluar ke seluruh interface-nya untuk memberitahu rute-rute yang telah diketahuinya.
- Router "memperhatikan" *routing update* yang berasal dari neighbor-nya, sehingga router bersangkutan dapat mempelajari rute-rute baru.
- Informasi routing berupa nomor subnet dan suatu metrik. Metrik mendefinisikan seberapa baik rute bersangkutan. Semakin kecil nilai metrik, semakin baik rute tersebut.
- Jika memungkinkan, router menggunakan broadcast dan multicast untuk mengirim *routing update*. Dengan menggunakan paket broadcast atau multicast, seluruh neighbor dalam suatu LAN dapat menerima informasi routing yang sama untuk sekali update.
- Jika suatu router mempelajari multirute untuk subnet yang sama, router akan memilih rute terbaik berdasarkan nilai metriknya.
- Router mengirim update secara periodik dan menunggu menerima update secara periodik dari router-router neighbor.
- Kegagalan menerima update dari neighbor pada jangka waktu tertentu akan menghasilkan pencabutan router yang semula dipelajari dari neighbor.
- Router berasumsi bahwa rute yang diumumkan oleh suatu router X, router *next-hop* dari rutenya adalah router X tersebut.

Beberapa fitur Protokol *Distance Vector* :

a) Route Poisoning

Routing loop dapat terjadi pada protokol *distance vector routing* ketika router-router memberitahukan bahwa suatu rute berubah dari kondisi valid ke tidak valid. Konvergensi yang lambat akan mengakibatkan router neighbor terlambat mendapat pemberitahuan kondisi tersebut, sehingga router neighbor tetap menganggap rute tersebut valid (dengan hop 1). Ketika router neighbor mengirimkan pemberitahuan keluar ke seluruh interfacenya, router pertama (yang memberitahukan kegagalan hubungan) akan mendapat informasi bahwa hubungan yang tidak valid tersebut dapat dicapai dari router neighbor dengan hop 2. Kedua router akan terus saling memberi informasi rute yang salah tersebut disertai dengan menaikkan informasi hop-nya.

Dengan *Route poisoning*, router tidak akan memberitahukan status tidak valid pada suatu rute yang gagal. Tetapi akan tetap memberikan informasi keadaan rute yang gagal dengan status valid. Rute tersebut akan diberi metrik yang sangat besar, sehingga router lain akan menganggap rute tersebut sebagai rute yang tidak valid.

b) Split Horizon

Fitur *Route poisoning* tidak seluruhnya dapat mengatasi kondisi looping. Pada kasus di atas, ketika suatu router memberitahukan suatu rute yang gagal dengan metrik yang sangat besar, router neighbor kemungkinan tidak langsung

mendapat pemberitahuan ini. Jika router neighbor kemudian memberitahu rute yang tidak valid tersebut ke router pertama (yang memberitahukan kegagalan hubungan) bahwa rute tersebut dapat dicapai dari dirinya dengan metrik yang jauh lebih baik, maka kondisi di atas dapat terjadi lagi.

Split horison mengatasi masalah ini dengan memberikan aturan bahwa suatu router yang mendapat pemberitahuan update informasi melalui interface x, tidak akan mengirimkan pemberitahuan yang sama ke interface x pula.

c) Split Horizon with Poison Reverse

Split horizon with poison reserve merupakan varian dari *split horizon*. Pada kondisi stabil, router bekerja dengan fitur *split horizon*. Tetapi ketika suatu rute gagal, router neighbor yang mendapat informasi ini akan mengabaikan aturan *split horizon*, dan kemudian mengirimkan kembali informasi tersebut ke router pertama dengan metrik yang sangat besar pula. Metode ini dapat memastikan bahwa seluruh router mendapat informasi yang benar mengenai kondisi rute tersebut.

d) Hold-Down Timer

Kondisi looping masih tetap terjadi pada jaringan *redundant* (jaringan dengan lebih dari satu jalur) walaupun fitur *split horizon* telah diaktifkan. Hal ini dimungkinkan karena suatu router dalam jaringan dapat memperoleh informasi mengenai rute yang sama melalui lebih dari satu jalur dan router. Oleh karenanya ketika suatu rute diinformasikan tidak valid oleh router bersangkutan, maka router neighbor pada saat yang sama juga mungkin mendapat informasi dari router lain dengan metrik yang masih dapat dijangkau. Informasi rute valid ini (*poison*) kemudian disampaikan ke router pertama, sehingga kondisi looping akan terjadi.

Hold-Down Timer mengatasi masalah ini dengan memberikan aturan bahwa ketika suatu router yang mendapat pemberitahuan suatu rute tidak valid, router tersebut akan mengabaikan informasi rute-rute alternatif ke subnet bersangkutan pada suatu waktu tertentu (*hold-down timer*).

e) Triggered (Flash) Updates

Protokol *distance vektor* biasanya mengirimkan update secara reguler berdasarkan interval waktu tertentu. Oleh karenanya banyak masalah looping terjadi sesaat setelah suatu rute tidak valid. Hal ini disebabkan karena beberapa router tidak segera mendapat informasi ini.

Beberapa router mengatasi masalah ini dengan menggunakan fitur *triggered update* atau *flash update*, dimana router akan segera mengirim pemberitahuan update baru sesaat setelah suatu rute tidak valid. Dengan demikian informasi perubahan status rute dapat segera di-forward-kan secara lebih cepat, sehingga pengaktifan *hold-down timer* di sisi router neighbor juga lebih cepat.

RIP dan IGRP

RIP *(Routing Information Protocol)* dan IGRP *(Interior Gateway Routing Protocol)* merupakan dua standar protokol routing berbasis *distance vector routing protocol.* RIP dan IGRP memiliki banyak kesamaan secara logik. Beberapa perbedaan penting dari kedua protokol routing ini diperlihatkan pada tabel berikut ini:

Tabel 1. Perbedaan antara RIP dan IGRP

Function	RIP	IGRP
Update Timer	30 detik	90 detik
Metric	Hop count	Fungsi bandwidth dan delay (default), Dapat juga berisi reliability, load, dan MTU
Hold-Down Timer	180	280
Flash (Triggered) Updates	Ya	Ya
Mask Sent in Update	Tidak	Tidak
Infinite-metric Value	16	4.294.967.295

IGRP Metric memberikan penghitungan yang lebih baik mengenai seberapa baik rute-rute yang ada dibandingkan RIP metric. IGRP metric dihitung menggunakan pengukuran *bandwidth* dan *delay* pada interface dimana informasi update diterima. Hal ini akan memberikan arti yang lebih baik dibandingkan metrik berdasarkan hop count.

RIP menggunakan penghitungan hop untuk besaran metriknya. Ketika informasi update diterima, metrik dari setiap subnet dalam informasi update merupakan jumlah router yang dilalui oleh informasi antara router penerima dengan setiap subnet. Hal ini dapat dilakukan karena sebelum mengirim informasi update, router akan menambah satu nilai metrinya untuk setiap subnet.

2.2.2. Algoritma Link State

Algoritma dasar kedua yang digunakan dalam proses routing adalah *algoritma link-state*. Algoritma *routing link-state-based* dikenal juga sebagai *shortest path first (SPF)*. Algoritma ini mengelola suatu database kompleks dari informasi topologi. Jika algoritma *distance vector* tidak memiliki informasi spesifik mengenai jaringan-jaringan jauh dan tidak mengetahui router-router jauh, maka *algoritma routing link-state* mengelola secara penuh pengetahuan mengenai jarak router dan bagaimana mereka terhubung.

Routing link-state menggunakan *link-state paket* (LSP), suatu database topologi, algoritma SPF, yang menghasilkan SPF *tree*, dan pada akhirnya akan dihasilkan *routing table* dari jalur dan *port* untuk setiap jaringan.

Routing link-state memiliki keunggulan pada jaringan besar karena beberapa alasan berikut:

- Protokol *link-state* hanya mengirim *update* dari topologi yang berubah saja.
- Periode *update* lebih jarang dibanding protokol *distance vector*.
- Routing *link-state* dapat disegmentasi ke dalam hirarki-hirarki area yang dapat membatasi jangkauan perubahan-perubahan rute.
- Mendukung *classless addressing*.
- Routing *link-state* mengirim *subnet mask* bersama dengan *update routing*.

Protokol *routing link-state* mengurangi trafik *broadcast* karena protokol ini tidak secara periodik melakukan *broadcast* ataupun mengirimkan seluruh isi tabel routing-nya ketika melakukan *broadcast*. Protokol *routing link-state* melakukan pertukaran salinan lengkap tabel rutenya ketika inisialisasi berlangsung. Selajutnya pertukaran *update* rutenya dilakukan secara *multicast* dan hanya pada saat terjadi perubahan (dibangkitkan oleh perubahan topologi). Dengan demikian kondisi ini memungkinkan hanya perubahan saja yang dikirim ke router-router lain, bukan seluruh *route table*-nya.

Berbeda dengan protokol *distance vector*, protokol *link-state* harus menghitung informasi metrik rute yang diterimanya. Router akan menghitung seluruh cost yang berhubungan dengan *link* pada setiap rute untuk mendapatkan metrik rute-rute yang terhubung. Hal ini mengakibatkan router-router yang menggunakan protokol *link-state* bekerja lebih berat dan memerlukan lebih banyak memory serta siklus pemrosessan.

Tabel 2. Perbandingan Protokol *Link-State* dan *Distance Vector*.

Fitur	Link-State	Distance Vector
Convergence Time	Cepat	Lambat, terutama disebabkan oleh fitur *loop-avoidance*
Loop Avoidance	Built in dalam protokol	Membutuhkan fitur tambahan seperti *split horizon*
Memory and CPU Requirements	Bisa besar; diminimalkan dg dsain yg baik	Rendah
Requires Design Effort for Large Networks	Ya	Tidak
Public Standard or Proprietary	OSPF adalah standar publik	RIP terdefinisi secara publik, IGRP tidak

Open Shortest Path First (OSPF)

OSPF adalah protokol routing yang diperuntukkan bagi jaringan IP dengan *Interior Gateway Protocol* (IGP) oleh *working group* dari *Internet Engineering Task Force* (IETF). OSP memiliki dua karakteristk utama, yaitu open standard dan berbasis pada algoritma SPF yang kadangkala direferensikan dengan algoritma Dijkstra (seseorang yang memiliki kontribusi pembuatan algoritma SPF).

Proses dasar pembelajaran rute-rute OSPF untuk pertamakalinya umumnya:

- Setiap router menemukan neighbor melalui setiap interface-nya. Daftar setiap neighbor di simpan dalam tabel neighbor.
- Setiap router menggunakan protokol tertentu untuk bertukar informasi topologi (LSA) dengan neighbor-nya.
- Setiap router menyimpan informasi topologi yang dipelajarinya dalam database topologi.
- Setiap router menjalankan algoritma SPF pada database topologinya untuk menghitung rute-rute terbaik dari setiap subnet di database.
- Setiap router menyimpan rute-rute terbaik ke setiap subnet ke dalam tabel routing-nya

Beberapa fitur Protokol *link state* :

a. Steady-State Operation

Tidak seperti protokol *distance vector*, protokol *link-state* menjaga hubungan dengan neighbor melalui pengiriman paket-paket kecil secara tak berkala dan jarang (kadang-kadang). OSPF menyebut paket kecil ini dengan *Hello packets*. Hello packet secara sederhana mengidentifikasi subnet dan keaktifan link serta router neighbor.

Ketika router gagal menerima paket Hellos dari neighbor pada suatu interval tertentu (dinamakan *dead interval*), router akan mempercayai bahwa router bersangkutan mengalami kegagalan dan menandainya dengan "down" pada database topologi-nya. Kemudian router berhenti menerima paket Hello dan mulai menjalankan Dijkstra untuk menghitung kembali rute-rute baru.

b. Loop Avoidance

Algoritma SPF mencegah loop yang secara natural telah dilakukan bersamaan dengan pemrosessan database topologi, sehingga tidak diperlukan *fitur loop-avoidance* seperti *split horizon, poison reserve, hold down timer*, dan lain sebagainya.

c. Scalling OSPF Through Hierarchical Design

Pada jaringan besar dengan ratusan router, waktu konvergensi OSPF dapat melambat, dan membutuhkan banyak memory, serta pembebanan prosessor. Masalah ini dapat diringkas sebagai berikut:

- Pada topologi database yang besar dibutuhkan lebih banyak memory dalam setiap router.
- Pemrosessan database topologi yang besar dengan algoritma SPF membutuhkan daya pemrosesan yang bertambah secara eksponensial sebanding dengan ukuran database topologi.
- Satu perubahan status interface (up ke down atau down ke up) memaksa setiap router untuk menjalankan SPF lagi.

Meskipun demikian, tidak ada definisi yang tepat untuk mendeskripsikan "jaringan besar". Sebagai patokan (sangat umum, bergantung pada desain, model, router, dan lain-lain), untuk jaringan dengan paling sedikit 50 router dan 100 subnet, fitur *OSPF scalability* seharusnya digunakan untuk mengurangi problem di atas.

d. OSPF Area

Penggunaan OSPF area dapat memecahkan banyak (tidak semuanya) permasalahan mendasar ketika menjalankan OSPF pada jaringan besar. OSPF area memecah-mecah jaringan sehingga router dalam satu area lebih sedikit mengetahui informasi topologi mengenai subnet pada area lainnya. Dengan database topologi yang lebih kecil, router akan mengkonsumsi memory dan proses yang lebih sedikit.

OSPF menggunakan istilah *Area Border Router* (ABR) untuk mendeskripsikan suatu router yang berada diantara dua area (perbatasan). Suatu ABR memiliki database topologi untuk kedua area tersebut dan menjalankan SPF ketika status *link* berubah pada salah satu area. Penggunaan area tidak selamanya mengurangi kebutuhan memory dan sejumlah penghitungan SPF untuk router ABR.

e. Stub Area

OSPF mengijinkan pendefinisian suatu area sebagai stub area, sehingga dapat mengurangi ukuran database topologi. OSPF juga mengijinkan varian area lain yang dapat mengurangi ukuran database topologi, dimana juga akan mempercepat pemrosessan algoritma SPF.

Tipe area terbaru saat ini adalah *Totally Not-So-Stubby Area* (TNSSA).

2.2.3. Balanced Hybrid Routing Protocol

Cisco menggunakan istilah *balanced hybrid* untuk mendeskripsikan protokol routing yang dipakai oleh EIGRP (*enhanced* IGRP). Hal ini dikarenakan EIGRP memiliki beberapa fitur seperti protokol *distance vector* dan protokol *link-state*.

EIGRP menggunakan formula berbasis bandwidth dan delay untuk menghitung metrik yang bersesuaian dengan suatu rute. Formula ini mirip dengan yang digunakan oleh IGRP, tetapi jumlahnya dikalikan dengan 256 untuk mengakomodasi perhitungan ketika nilai bandwidth yang digunakan sangat tinggi.

EIGRP melakukan konvergensi secara cepat ketika menghindari loop. EIGRP tidak melakukan perhitungan-perhitungan rute seperti yang dilakukan oleh protokol *link-state*. Hal ini menjadikan EIGRP tidak membutuhkan desain eksta, sehingga hanya memerlukan lebih sedikit memory dan proses dibandingkan protokol *link-state*.

Konvergensi EIGRP lebih cepat dibandingkan dengan protokol *distance vector*. Hal ini terutama disebabkan karena EIGRP tidak memerlukan fitur *loop-avoidance* yang pada kenyataannya menyebabkan konvergensi protokol *distance vector* melambat. Hanya dengan mengirim sebagian dari *routing update* (setelah seluruh informasi routing dipertukarkan), EIGRP mengurangi pembebanan di jaringan.

Salah satu kelemahan utama EIGRP adalah protokol ini Cisco-*proprietary,* sehingga jika diterapkan pada jaringan multivendor diperlukan suatu fungsi yang disebut *route redistribution*. Fungsi ini akan menangani proses pertukaran rute router diantara dua protokol *link-state* (OSPF dan EIGRP).

Tabel 3. Fitur EIGRP dibandingkan dengan OSPF dan IGRP.

Fitur	EIGRP	IGRP	OSPF
Mengenali router tetangga sebelum mempertukarkan informasi routing	Ya	Tidak	Ya
Membangun tabel topologi selain menambahkan route kedalam tabel routing	Ya	Tidak	Ya
Cepat berkonvergensi	Ya	Tidak	Ya
Secara default menggunakan metrik yang didasarkan bandwidth dan delay	Ya*	Ya	Tidak
Mengirimkan seluruh informasi routing pada setiap siklus *routing update*	Tidak	Ya	Tidak
Membutuhkan fitur *distance vector loop-avoidance*	Tidak	Ya	Tidak
Standar publik	Tidak	Tidak	Ya

*EIGRP menggunakan metrik yang sama seperti IGRP, kecuali penskalaan metrik dikalikan dengan 256.

3. Kesimpulan

- Untuk jaringan berskala kecil algoritma routing yang sesuai adalah routing secara statik karena lebih menghemat bandwidth sedangkan untuk jaringan berskala besar lebih tepat menggunakan *dynamic routing*.
- Protokol RIP banyak digunakan karena kesederhanaan dalam mengimplementasikannya.
- Algoritma *link state* lebih baik dibandingkan algoritma *distance vector* dilihat dari sisi waktu konvergensi dan tidak adanya *routing loop* di dalam jaringan.
- Algoritma EIGRP yang dikembangkan Cisco sudah menggabungkan kelebihan dari algoritma *link state* dan algoritma *distance vector*, tetapi teknologi ini tidak banyak didukung oleh vendor router yang lain (Cisco *proprietary*).

Daftar Pustaka

1. Andrea, S. 1989. Computer Networks. Prentice Hall.
2. Computer Network Research Group, ITB, Mei 1999, oleh Adnan Basamalah, Lutfi Wisbiono Arif, Joko Yulianto.
3. Keiser, GE. 1989. LAN. McGraw-Hill
4. Stage 1 Intelligent Network Service Descriptions, Divisi RisTI TELKOM, Bandung, 1997.
5. Stallings, W. Data and Computer Communication third Edition. Maxwell Maxmilian International.
6. Tannembaum, A.S. 1996. Computer Network, Prentice Hall.