

Rapid Spanning Tree Protocol untuk Mencegah Loop pada Jaringan Redundan

Muhammad Iqbal Haidar - 13523111¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

iqbalhaidar0603@gmail.com, 13523111@std.stei.itb.ac.id

Abstract— Seiring dengan perkembangan zaman, kebutuhan untuk saling bertukar informasi semakin tinggi. Sebuah jaringan andal diperlukan agar proses pertukaran informasi tidak terganggu. Memberikan redundansi jaringan merupakan salah satu cara untuk memitigasi permasalahan dan meningkatkan keandalan jaringan. Diperlukan sebuah protokol untuk mengatur koneksi mana saja yang dijadikan sebagai jaringan utama dan lainnya sebagai cadangan. *Spanning Tree Protocol* dan *Rapid Spanning Tree Protocol* hadir untuk menjawab tantangan tersebut dengan membuat sebuah *Spanning Tree* dari jaringan redundan dan digunakan sebagai jaringan utama.

Keywords—Broadcast Storm, Jaringan, Redundansi, Spanning Tree.

I. PENDAHULUAN

Di era digital saat ini, pertukaran informasi menjadi hal penting yang dapat mendukung kegiatan manusia sehari-hari. Setiap harinya lebih dari triliunan *byte* data ditransmisikan dari suatu tempat ke tempat lain melalui jaringan internet. Semua informasi tersebut bisa didapatkan dalam genggaman tangan berkat perkembangan teknologi informasi. Oleh sebab itu, teknologi informasi memiliki peranan besar dalam mendorong umat manusia berakselerasi menuju peradaban maju.

Di dunia dengan perkembangan teknologi yang pesat seperti saat ini, inovasi dalam bidang teknologi informasi selalu dinantikan oleh khalayak ramai sebab implementasinya dapat dirasakan secara universal, tidak terbatas pada golongan tertentu saja. Hal tersebut mendorong para ilmuwan serta perusahaan teknologi informasi berlomba-lomba menciptakan inovasi baru yang dapat merevolusi bidang ini untuk perkembangan umat manusia.

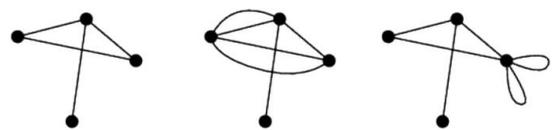
Jaringan komputer adalah cabang ilmu yang mempelajari hubungan komunikasi antara perangkat komputer dengan perangkat komputer lainnya. Ilmu ini mendasari terbentuknya jaringan internet, yakni jaringan yang menghubungkan perangkat komputer di seluruh belahan dunia dan dapat diakses secara publik oleh semua orang. Lebih dari itu ada juga jaringan intranet yang banyak digunakan oleh perusahaan ataupun pemerintahan sebagai sarana berbagi data yang sifatnya tidak boleh diketahui publik. Terlepas dari hal tersebut, keduanya menggunakan protokol yang sama untuk mengatur sebuah jaringan yang dibangun. Protokol atau aturan adalah hal-hal yang disepakati secara bersama-sama sebagai standar global dalam melaksanakan praktik jaringan komputer. Umumnya organisasi yang mengatur hal tersebut adalah *ieee* sebagai organisasi terbesar yang menaungi bidang terkait jaringan komputer.

Makalah ini akan membahas sebuah *protocol* untuk mengatasi permasalahan jaringan komputer khususnya pada jaringan krusial yang membutuhkan redundansi. Sebuah jaringan redundansi adalah jaringan yang memiliki beberapa jalur/*edges* untuk sebuah *node*/perangkat komputer/*vertex*. Hal tersebut dapat menimbulkan kebingungan saat pengiriman data dari node ke node lainnya sebab ada banyak kemungkinan jalur yang dapat digunakan, apabila hal tersebut dibiarkan akan terjadi *traffic jam* pada jaringan tersebut. Oleh karena itu digunakan *Rapid Spanning Tree Protocol* yang akan membentuk sebuah *Spanning Tree* dari jaringan tersebut sehingga permasalahan tersebut dapat diatasi.

II. LANDASAN TEORI

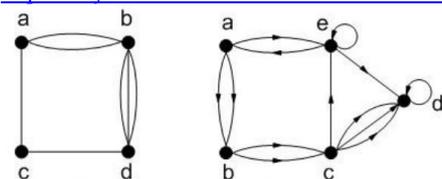
A. Graf

Graf adalah sebuah *tuple* himpunan objek simpul (*vertex*) dan sisi (*edges*) yang dapat dituliskan dalam $G = (V, E)$. V merupakan himpunan objek simpul-simpul yang tidak boleh kosong. Sedangkan E adalah himpunan sisi-sisi yang menghubungkan dua buah simpul dan dibolehkan berupa himpunan kosong. Graf berdasarkan orientasi arah pada sisinya terbagi menjadi dua jenis yakni graf tak-berarah (*undirected*) dan graf berarah (*directed*). Graf tak-berarah adalah graf dimana sisinya tidak memiliki orientasi arah. Sedangkan graf berarah adalah graf dimana sisinya diberikan arah tertentu sehingga dua sisi dapat dikatakan berbeda bergantung arahnya walaupun menghubungkan dua buah simpul yang sama.



Gambar 2.1 Graf

Sumber: <https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/>



Gambar 2.2 Graf Tak Berarah dan Berarah

Sumber: <https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/>

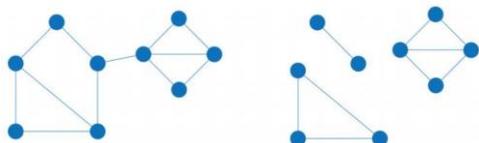
B. Lintasan

Lintasan (*path*) pada sebuah graf adalah kumpulan sisi-sisi yang menghubungkan suatu simpul pada graf dengan simpul

lainnya. Lintasan memiliki sebuah properti yakni panjang lintasan yang merepresentasikan banyaknya sisi pada sebuah lintasan yang menghubungkan sebuah pasangan simpul. Pada graf berarah untuk mencari sebuah lintasan harus memerhatikan orientasi arah pada sisi-sisinya, tidak boleh membuat lintasan yang berlawanan arah dengan arah sisi.

C. Keterhubungan

Sebuah graf tak-berarah dapat dikatakan graf terhubung jika memenuhi kriteria yakni untuk setiap pasangan simpul u dan v terdapat sebuah lintasan yang menghubungkan keduanya. Apabila terdapat satu saja kasus dimana sepasang simpul tidak memiliki lintasan, maka graf tersebut dikatakan graf tak-terhubung. Pada graf berarah untuk mencari tahu apakah graf tersebut terhubung atau tidak cukup menghilangkan orientasi pada setiap sisinya sehingga graf akan menjadi graf tak-berarah baru kemudian tinjau dengan kriteria keterhubungan graf tak-berarah. Sebuah graf berarah dapat dikatakan terhubung kuat apabila untuk setiap pasang simpul pada graf memiliki lintasan berarah bolak-balik yang menghubungkan kedua simpul tersebut. Apabila graf berarah tidak terhubung kuat namun tetap terhubung maka graf tersebut dapat dikatakan graf terhubung lemah.



Gambar 2.3 Graf Terhubung dan Tak Terhubung

Sumber: <https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/>

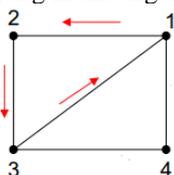


Gambar 2.4 Graf Terhubung Lemah dan Terhubung Kuat

Sumber: <https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/>

D. Sirkuit

Sebuah sirkuit pada graf didefinisikan sebagai suatu lintasan yang menghubungkan sepasang simpul yang sama. Bisa juga dikatakan sirkuit adalah sebuah lintasan pada graf yang memiliki simpul awal dan akhir yang sama. Panjang sirkuit adalah panjang lintasan yang membangun sirkuit tersebut.



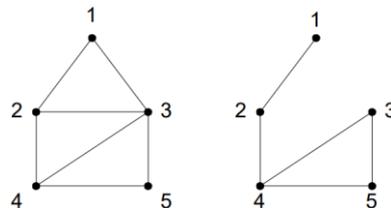
Gambar 2.5 Sirkuit pada Graf

Sumber: <https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/>

E. Upagraf Merentang

Sebuah upagraf (*subgraph*) $G = (V, E)$ didefinisikan sebagai $G' = (V', E')$ dimana $V' \subseteq V$ dan $E' \subseteq E$. Dapat dikatakan bahwa sebuah upagraf adalah sebagian dari graf secara keseluruhan. Upagraf merentang (*spanning subgraph*) adalah sebuah upagraf G' dimana upagraf tersebut mengandung semua

simpul yang dimiliki oleh graf utama G , namun tidak harus memiliki semua sisi dari G .

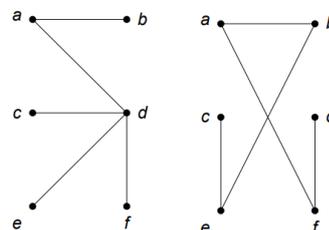


Gambar 2.6 Graf dan Upagraf Merentang

Sumber: <https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/>

F. Pohon

Pohon adalah sebuah *subset* atau kasus khusus dari sebuah graf yang artinya pohon juga termasuk jenis graf dan mewarisi semua sifat-sifat serta terminologi dalam teori graf. Sebuah pohon didefinisikan sebagai graf tak berarah yang memiliki kecirian terhubung serta tidak memiliki sirkuit pada semua lintasannya. Beberapa sifat khusus yang dimiliki oleh sebuah pohon antara lain, memiliki sisi sebanyak $n-1$ sisi dimana n adalah jumlah simpul yang dimiliki pohon, setiap simpul terhubung hanya dengan satu lintasan tunggal, tidak memiliki sirkuit dan penambahan satu sisi pada sebuah pohon akan membuatnya memiliki satu sirkuit.

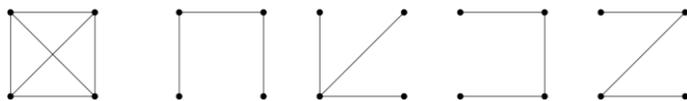


Gambar 2.7 Pohon

Sumber: <https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/>

G. Pohon Merentang

Pohon merentang memiliki definisi yakni sebuah graf yang bercirikan terhubung memiliki upagraf merentang yang berbentuk pohon dengan cara menghapus beberapa sisinya sehingga setiap sepasang simpul pada upagraf merentang dihubungkan dengan hanya satu lintasan tunggal saja. Sebuah pohon merentang yang dibentuk dari sebuah graf terhubung dapat memiliki berbagai macam variasi bentuk bergantung terhadap sisi mana yang dipilih untuk dipertahankan.



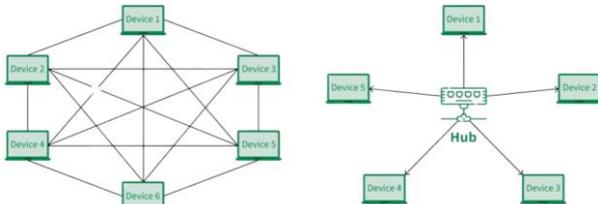
Gambar 2.7 Graf dan Variasi Pohon Merentang

Sumber: <https://informatika.itb.stei.itb.ac.id/~rinaldi.munir/>

H. Topologi Jaringan

Topologi jaringan merupakan cara dalam menyusun hubungan antara suatu komputer dengan komputer lainnya sehingga membentuk suatu kesatuan jaringan yang utuh dan berfungsi sebagaimana harusnya. Pada umumnya sebuah komputer (*node*) dihubungkan dengan komputer lainnya menggunakan kabel UTP yang dijadikan sebagai medium dalam transfer data. Ada berbagai macam jenis topologi jaringan untuk kasus penggunaan berbeda-beda seperti topologi *mesh* yang memiliki redundansi tinggi sebab setiap node memiliki lintasan

husus menuju setiap node lainnya sehingga cocok untuk penggunaan sistem kritis seperti disektor perbankan dan kesehatan. Ada juga topologi *star* yang cocok digunakan untuk perusahaan skala kecil sampai menengah sebab biayanya yang lebih rendah daripada topologi *mesh* dan memiliki skalabilitas yang tinggi sehingga cocok apabila dimasa yang akan datang perusahaan berkembang menjadi lebih besar dan membutuhkan jaringan yang lebih luas.



Gambar 2.8 Topologi Mesh dan Star

Sumber: <https://www.geeksforgeeks.org/types-of-network-topology/>

Sebuah jaringan kecil (*LAN*) yang terdiri atas beberapa node dapat terhubung dengan jaringan kecil ditempat lainnya dengan menggunakan perangkat bernama *switch*. Hal tersebut bermanfaat apabila misalkan sebuah departemen di sebuah perusahaan perlu berbagi data dengan departemen lain di lantai yang berbeda maka cukup menghubungkan kedua jaringan pada departemen tersebut dengan *switch* dan kabel *UTP* lalu mereka seolah-olah berada bersama-sama di suatu jaringan yang lebih besar (*WAN*).

I. Switch

Switch adalah suatu perangkat jaringan komputer yang memiliki fungsi utama sebagai pengendali data-data yang sedang ditransmisikan dari suatu komputer menuju komputer lainnya. Cara kerja switch dapat dianalogikan sebagai seorang tukang pos yang bertugas mengirimkan surat dari rumah sang pengirim menuju rumah penerima. Mula-mula switch akan melakukan *Address Resolution Protocol (ARP)* untuk memetakan seluruh perangkat komputer yang berada di dalam jaringan tersebut, kemudian akan terbentuk sebuah *ARP table* yang berisi pemetaan *IP Address* dengan *MAC Address* untuk setiap perangkat dalam jaringan.



Gambar 2.9 Switch

Sumber: Google

Apabila suatu komputer ingin mengirimkan sebuah paket data menuju komputer lainnya maka paket data tersebut akan dikirimkan terlebih dahulu ke *switch* untuk ditinjau alamat penerimanya berdasarkan *ARP table* yang sudah disimpan sebelumnya baru kemudian paket data tersebut diteruskan menuju komputer penerima. Jika sang penerima paket data tidak berada dalam jaringan *switch* tersebut maka paket akan diteruskan menuju *router* ataupun *switch* pada jaringan lainnya untuk diperiksa apakah penerima berada dalam jaringan itu.

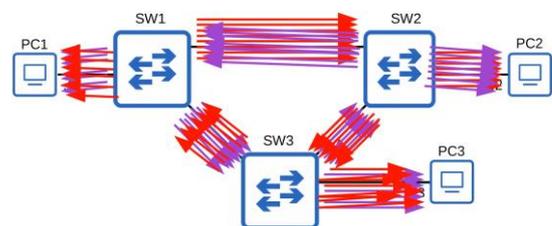
Semakin besar suatu jaringan maka diperlukan *switch* yang semakin canggih untuk menangani arus paket data yang semakin banyak, apabila *switch* tidak mampu dalam menangani hal tersebut maka akan terjadi *traffic jam* yang dapat mengganggu proses transmisi data di dalam jaringan yang berakibat pada

lamanya proses pertukaran data.

III. SPANNING TREE PROTOCOL (STP)

Sebuah jaringan yang andal untuk digunakan kapan pun dibutuhkan oleh hampir semua sektor industri, terlebih lagi industri yang menyangkut kehidupan orang banyak seperti perbankan dan kesehatan. Jaringan pada sektor-sektor tersebut harus bisa berjalan 24/7 sepanjang tahun karena kegagalan dalam sistem hampir tidak memiliki toleransi. Walaupun sekejap saja waktu terbuang untuk memperbaiki kegagalan tersebut dapat memberikan kerugian yang sangat besar bagi perusahaan maupun pelanggan. Oleh sebab itu, di zaman sekarang banyak sekali rekayasawan jaringan komputer yang membangun sebuah jaringan andal dengan memberikan redundansi salah satunya dengan memberikan banyak lintasan yang menghubungkan suatu komputer dengan komputer lainnya. Harapannya apabila suatu lintasan yang menghubungkan sepasang komputer sedang bermasalah, maka dapat digunakan lintasan lain sebagai alternatif penghubung dua komputer tersebut. Dengan begitu jaringan hampir selalu bisa digunakan kapan pun dan dimana pun tanpa harus melakukan *troubleshooting* terlebih dahulu jika muncul permasalahan, hal tersebut dinamakan *zero downtime*.

Namun akibat penggunaan redundansi ini timbul sebuah masalah baru yaitu kemungkinan pesan melakukan *looping* di dalam jaringan sebab banyaknya lintasan yang dapat digunakan. Contohnya ketika *switch* melakukan *Address Resolution Protocol (ARP)* untuk membuat *ARP Table* berisi pemetaan *IP Address* dan *MAC Address* perangkat komputer yang terhubung pada jaringan. Dimana setiap *switch* dalam jaringan mengirimkan *ARP Message* dengan teknik *flooding* yaitu switch mengirimkan *ARP Message* kepada semua perangkat yang terhubung dengannya. Kemudian jika pesan tersebut diterima oleh *switch* lain maka akan diteruskan kembali menggunakan teknik *flooding* dan seterusnya bolak-balik. Akibat dari peristiwa tersebut jika tetap dibiarkan akan menimbulkan sebuah peristiwa bernama *broadcast storm* dimana banyak sekali pesan-pesan yang berkeliaran di dalam jaringan sehingga terjadi *traffic* yang mengakibatkan performa jaringan menurun. Implikasinya pesan/paket data penting yang seharusnya sampai di tujuan dengan tepat waktu justru baru sampai dengan waktu yang lama atau bahkan tidak sampai sama sekali.

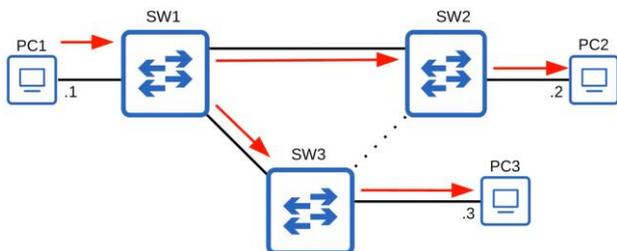


Gambar 3.1 Broadcast Storm pada Jaringan Redundan

Sumber: [Jeremy's IT Lab](#)

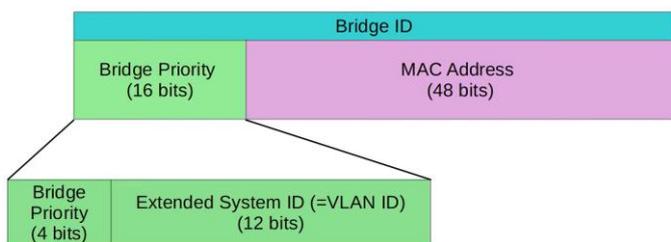
Oleh karena permasalahan itu, dibuatlah sebuah standar baru untuk mengatasi hal tersebut yakni IEEE 802.1D yang membahas tentang *Spanning Tree Protocol (STP)*. Garis besar dari protokol ini adalah membuat sebuah jaringan yang tadinya redundan dengan banyak lintasan beririsan menjadi sebuah *Spanning Tree* saja dimana tidak ada sirkuit dalam jaringan tersebut. Keunggulan dari protokol ini adalah *Spanning Tree* hanya berlaku secara logika saja, sedangkan pada bentuk jaringan fisiknya dimungkinkan tetap memiliki redundansi.

Redundansi ini tidak sia-sia sebab apabila suatu lintasan *Spanning Tree* bermasalah maka sistem akan mencari lintasan baru yang dapat menggantikannya dari redundansi yang ada. Ini merupakan hal yang baik sebab permasalahan *looping* pada jaringan redundan dapat teratasi tanpa harus menghilangkan redundansi itu sendiri sehingga sektor kritis tetap bisa menggunakan jaringan komputer tanpa harus khawatir akan terjadinya masalah. *Spanning Tree Protocol* pada esensinya hanyalah menonaktifkan lintasan yang sekiranya redundan dan mengakibatkan *looping* pada jaringan sehingga jika dilihat secara sekilas akan membentuk mirip sebuah *Spanning Tree* dan pada lintasan yang dinonaktifkan tersebut tidak akan menerima pesan kecuali *BPDU*.



Gambar 3.2 *Spanning Tree* pada Jaringan Redundan
 Sumber: [Jeremy's IT Lab](#)

Dalam prosesnya untuk menentukan bagaimana struktur *Spanning Tree* yang akan dibentuk dari jaringan redundan, terdapat beberapa langkah serta aturan yang harus ditaati menurut IEEE 802.1D agar terjadi standarisasi di seluruh dunia. Mulanya setiap switch akan mengirimkan sebuah *Hello BPDU* setiap dua detik kepada seluruh perangkat yang terhubung dengannya kemudian akan ditentukan sebuah *root node* yaitu switch yang akan menjadi titik acuan pembuatan *Spanning Tree*. Penentuan *root node* ini berdasarkan aturan yakni switch yang memiliki *Bridge ID* terkecil akan dijadikan sebagai *root node*. *Bridge ID* merupakan sebuah *identifikasi* 64 bit yang terdiri atas 48 bit *MAC address* sebuah switch dan 16 bit *priority number* dari switch tersebut, sedangkan *Bridge ID* adalah gabungan penjumlahan antara *MAC Address* + *priority number* nya. *MAC Address* adalah *identifikasi* fisik setiap perangkat jaringan internet dimana hal ini bersifat *fixed* bawaan dari perusahaan pembuat perangkat sehingga tidak dapat diubah. Sedangkan *priority number* adalah sebuah angka yang dapat dimanipulasi oleh rekayasawan dengan rentang 0 - 65536 dengan kelipatan setiap 4096, properti ini berguna jika ingin memaksakan switch tertentu untuk menjadi sebuah *root node* dalam jaringan *Spanning Tree*.



Gambar 3.3 Struktur *Bridge ID*
 Sumber: [Jeremy's IT Lab](#)

Setelah *root node* ditemukan, maka dilanjut dengan langkah berikutnya yakni penentuan *port* mana yang akan dijadikan *root port* dan *designated port*. *Root port* adalah sebuah *port* pada switch yang berfungsi untuk meneruskan/mengirimkan pesan

yang dikirimkan berasal dari *root node*, sedangkan *designated port* adalah sebuah *port* pada switch yang senantiasa bersebelahan/berpasangan dengan *root port*. Pemilihan *root port* dan *designated port* ini krusial dalam pembentukan jaringan menjadi sebuah *Spanning Tree*, oleh karena itu telah ditetapkan standar prioritas pemilihan *root port* pada setiap switch dimana setiap *root port* dipilih berdasarkan

1. *lowest root cost*
2. *lowest neighbor Bridge ID*
3. *lowest neighbor port ID*

Root cost merupakan sebuah konstanta yang merepresentasikan bobot dari setiap lintasan pada jaringan redundan serta telah ditetapkan sebelumnya berdasarkan kecepatan (*bandwidth*) yang mampu ditampung oleh lintasan tersebut. Semakin besar kecepatan sebuah lintasan maka bobot/biaya nya semakin kecil sehingga lebih dipreferensikan sebagai lintasan yang dipilih dalam *Spanning Tree*. *Port ID* merupakan *ID port* yang digunakan switch untuk membentuk lintasan tersebut, perlu diperhatikan bahwa *port ID* yang ditinjau adalah milik tetangga bukan milik diri sendiri. Apabila langkah ini sudah selesai maka *Spanning Tree* sudah terbentuk setengahnya, langkah terakhir yang harus dilakukan adalah menonaktifkan semua lintasan yang tidak terpilih menjadi bagian dari *Spanning Tree*.

Speed	Cost
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2

Tabel 3.1 *Root Cost* pada *Spanning Tree Protocol*

Untuk melakukan penonaktifan sebuah lintasan pada jaringan sesungguhnya adalah memilih sebuah *designated port* yang akan dijadikan sebagai *forwarding state* dan pasangan *port* pada switch lainnya sebagai *non-designated port* yang dijadikan sebagai *blocking state*. Jadi walaupun *port* tersebut dalam *forwarding state* yakni mengirimkan pesan namun oleh *port* dalam *blocking state* tidak akan dihiraukan sehingga *looping* dapat dihindarkan. Pada proses pemilihan *designated port* terdapat beberapa aturan prioritas yang sudah ditetapkan menjadi standar dan wajib diikuti antara lain,

1. *lowest root cost*
2. *lowest Bridge ID*

Setelah pemilihan *designated port* berhasil, maka sistem akan mengubah *port* tersebut menjadi *forwarding state* dan pasangan *port* pada switch lainnya akan menjadi *blocking state*. Saat *Spanning Tree* sudah berhasil terbentuk, maka akan dilanjutkan dengan *ARP* baru kemudian transmisi data dalam jaringan bisa dilaksanakan dengan lancar. Apabila terjadi masalah pada salah satu lintasan *Spanning Tree* maka protokol *STP* akan kembali dijalankan untuk membentuk *Spanning Tree* baru yang tidak memiliki masalah pada lintasannya.

IV. RAPID SPANNING TREE PROTOCOL (RSTP)

Pada esensinya, *Rapid Spanning Tree Protocol (RSTP)* merupakan penyempurnaan lanjutan dari *protocol* sebelumnya yakni *Spanning Tree Protocol*. *RSTP* di standarisasi serta terdokumentasi pada *IEEE 802.1w* yang digunakan sebagai

acuan bagi penyedia perangkat jaringan di seluruh dunia, namun ada juga beberapa perusahaan yang melakukan modifikasi terhadap standar tersebut untuk mengeluarkan versi *proprietary* nya sendiri. Sebagai contoh *cisco* sebagai salah satu perusahaan telekomunikasi terbesar di dunia, melakukan modifikasi tambahan terhadap *RSTP* yang diimplementasikan pada produk *switch* nya. Standar modifikasi yang dilakukan oleh *cisco* bernama *Rapid PVST+* yang memiliki keunggulan dapat melakukan *load balancing* sebab proses *Spanning Tree* dilakukan per *VLAN*. Terlepas dari itu, keduanya sebenarnya memiliki mekanisme yang mirip seperti dengan protokol prekursornya yakni *Spanning Tree Protocol (STP)*.

Proses pemilihan *root node*, *root ports* dan *designated ports* pada *Rapid Spanning Tree Protocol (RSTP)* memiliki alur serta urutan prioritas yang sama dengan *Spanning Tree Protocol (STP)*. Namun ada beberapa bagian yang diperbaharui untuk menyesuaikan dengan perkembangan teknologi yang berlangsung seperti halnya pada pembobotan lintasan. *RSTP* mampu melakukan konvergensi *Spanning Tree* lebih cepat ketimbang *STP* ketika topologi ataupun perangkat yang terhubung berubah. Hal tersebut disebabkan karena pada protokol ini tidak ada *listening state* saat proses memindahkan sebuah *port* menjadi *forwarding state*, melainkan digantikan langsung dengan mekanisme negosiasi *bridge-bridge handshake*. *Improvement* ini membuat dapat menghemat waktu dari awalnya 30-50 detik untuk konvergensi pada *STP* menjadi beberapa milidetik saja pada *RSTP*, sehingga jaringan akan menjadi lebih resistan dan responsif terhadap permasalahan yang muncul.

Sebab perkembangan teknologi jaringan yang sangat pesat, standarisasi *port cost* pada *RSTP* juga harus diperbaharui dari *STP* sebelumnya. Mulanya *port cost* hanya dihitung sampai pada tingkat *bandwidth 10 Gbps* dan sekarang di-*extend* hingga *1 Tbps*, apabila terdapat *bandwidth* yang melebihi batas maksimal maka *port cost* yang digunakan tetap dengan *bandwidth* maksimal yakni *1 Tbps*. Hal ini diperlukan supaya *Spanning Tree* yang terbentuk merupakan *Minimum Spanning Tree* sehingga performa jaringan menjadi maksimal.

Speed	Cost
10 Mbps	2,000,000
100 Mbps	200,000
1 Gbps	20,000
10 Gbps	2,000
100 Gbps	200
1 Tbps	20

Tabel 3.1 Root Cost pada Rapid Spanning Tree Protocol

Port states pada *Rapid Spanning Tree* hanya ada tiga jenis sesuai dengan peruntukannya masing-masing. Ketiga jenis *port states* tersebut antara lain,

- **Discarding** - ini merupakan *port states* yang bersifat stabil dimana berfungsi sebagai *port* yang memblokir suatu lintasan yang tidak digunakan dalam *Spanning Tree* sehingga tidak terjadi *looping* dalam jaringan.
- **Learning** - ini merupakan *port states* yang bersifat

transisional dimana hal ini terjadi ketika *Spanning Tree* baru mau dibentuk. Pada fase ini *port* mengirim dan menerima pesan *Hello BPDU* untuk menentukan bagaimana bentuk geometri dari *Spanning Tree*

- **Forwarding** - ini merupakan *port states* yang bersifat stabil dimana berfungsi sebagai *port* yang membuka jalan untuk paket agar bisa dikirim melalui lintasan *Spanning Tree* yang sudah terbentuk sebelumnya menuju perangkat lain.

Sebelumnya pada *Spanning Tree Protocol (STP)* terdapat tiga *port roles* yang masing-masing memiliki fungsi tertentu, pada *RSTP* akan dipecah lagi *port roles non-designated port* menjadi *alternate port* dan *backup port*.

- **Root port** - Adalah sebuah *port* pada *switch* yang digunakan untuk menerima pesan yang dikirim oleh *root node* untuk kemudian diproses oleh *switch* penerima. *Root port* ini dapat dibilang merupakan *port* pada *switch* yang terhubung dengan *switch root node*. Perlu dicatat bahwa sebuah *root node* tidak mempunyai sama sekali *root port*.
- **Designated port** - Adalah sebuah *port* pada *switch* yang digunakan untuk mengirim pesan dari *root node* menuju *switch* lain di dalam jaringan. *Designated port* ini merupakan pasangan dari sebuah *root port* sehingga keduanya dapat membentuk sebuah lintasan yang termasuk ke dalam struktur geometri *Spanning Tree*. Perlu dicatat bahwa semua *port* pada *root node* memiliki *role* yang sama yakni *designated port*
- **Alternate port** - Adalah sebuah *port* pada *switch* dengan state sebagai *discarding port* yang menerima pesan *BPDU superior* dari *switch* lainnya dalam jaringan. Fungsi dari *port* ini adalah sebagai *backup* apabila sebuah *root port* mengalami kendala maka *alternate port* ini akan segera langsung menggantikannya tanpa harus ada *transitional state*.
- **Backup port** - Adalah sebuah *port* pada *switch* dengan state sebagai *discarding port* yang menerima pesan *BPDU superior* dari *interface* lainnya pada *switch* yang sama. Umumnya ini ada pada sebuah *hub* jaringan, namun karena *hub* sudah tidak lagi digunakan tergantikan oleh *switch* maka *port* ini akan jarang untuk ditemui. Fungsi dari *backup port* ini adalah untuk sebagai *backup* apabila sebuah *designated port* mengalami kendala masalah.

Pada *RSTP* terdapat tiga macam jenis koneksi (lintasan) yang mungkin, masing-masingnya merepresentasikan sesuatu hal yang berbeda-beda. Ketiga jenis koneksi tersebut adalah

- **Edge** - Ini merupakan sebuah *port* yang terhubung langsung dengan komputer sehingga pada *link* ini tidak diperlukan negosiasi untuk membuatnya, cukup otomatis dijadikan *forwarding state*.
- **Point-to-point** - Ini merupakan sebuah koneksi yang menghubungkan suatu *switch* dengan *switch* lainnya dalam jaringan sehingga untuk menginisiasinya memerlukan sebuah negosiasi agar tidak terjadi *loop*. *Link* ini beroperasi dalam *full duplex mode*.
- **Shared** - Ini merupakan sebuah koneksi yang menghubungkan sebuah *hub*, koneksi ini sudah jarang ditemui sebab *hub* tidak lagi digunakan dan telah tergantikan oleh *switch*. *Link* ini beroperasi dalam *half duplex mode*.

V. KESIMPULAN

Berdasarkan pembahasan di atas dapat ditarik kesimpulan bahwa teori graf khususnya *tree* memiliki aplikasi yang luas di kehidupan sehari-hari. Dalam bidang jaringan komputer, *tree* digunakan untuk menghindari sebuah masalah *loop* yang mungkin terjadi pada jaringan redundan. *Spanning Tree Protocol (STP)* dan *Rapid Spanning Tree Protocol (RSTP)* merupakan salah satu contoh penerapan *tree* yang mengatasi permasalahan tersebut. Bukan tidak mungkin, dimasa yang akan datang bermunculan implementasi-implementasi lain yang memanfaatkan teori graf/*tree* untuk memecahkan persoalan khususnya dibidang jaringan komputer.

UCAPAN TERIMA KASIH

Puji dan syukur penulis ucapkan kepada Tuhan Yang Maha Esa atas berkat, rahmat, dan hidayat-Nya sehingga penulisan makalah ini bisa terselesaikan. Penulis juga ingin menyampaikan ucapan terima kasih kepada orang tua yang senantiasa memberikan semangat dan dukungan selama proses pengerjaan makalah ini. Penulis juga ingin mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T., selaku dosen pengampu mata kuliah matematika diskrit (IF1220) kelas K-1, yang telah menjadi pembimbing, memberikan ilmu dan pengetahuan terkait materi sebagai modal dalam mengembangkan makalah ini. Ucapan terima kasih juga disampaikan kepada semua rekan dan pihak lain yang turut mendukung proses penulisan makalah ini, baik secara moral maupun material, walaupun tidak dapat disebutkan satu per satu.

REFERENSI

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf> Diakses pada tanggal 27 Desember 2024.
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/22-Graf-Bagian3-2024.pdf> Diakses pada tanggal 27 Desember 2024.
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf> Diakses pada tanggal 27 Desember 2024.
- [4] <https://www.geeksforgeeks.org/types-of-network-topology/> Diakses pada tanggal 27 Desember 2024.
- [5] <https://www.jeremysitlab.com/ccna-resources> Diakses pada tanggal 27 Desember 2024.
- [6] <https://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/10556-16.html> Diakses pada tanggal 27 Desember 2024.
- [7] <https://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/24062-146.html> Diakses pada tanggal 27 Desember 2024.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bekasi, 27 Desember 2024



Muhammad Iqbal Haidar - 13523111