

Spanning Tree Protocol sebagai Aplikasi Pohon Merentang

Muhammad Reza Mandala Putra 13509003

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung 40132, Indonesia

13509003@std.stei.itb.ac.id

Sari—Pohon merentang banyak dijumpai aplikasinya dalam topologi jaringan. Salah satu aplikasi dari pohon merentang adalah *Spanning Tree Protocol*. *Spanning Tree Protocol*, disingkat menjadi STP, merupakan bagian dari standard IEEE 802.1 untuk kontrol media akses. *Spanning Tree Protocol* berfungsi sebagai protokol untuk pengaturan koneksi dengan menggunakan algoritma *spanning tree* (pohon merentang) yang disebut dengan STA. Dalam makalah ini akan dijelaskan mengenai *Spanning Tree Protocol*, cara kerja dan perkembangannya.

Kata Kunci—*Spanning Tree Protocol*, *bridge*, graf, pohon.

I. PENDAHULUAN

I.1 Pengertian Graf

Graf adalah suatu struktur diskrit yang terdiri dari vertex (simpul) dan sisi, dimana sisi menghubungkan vertex-vertex yang ada. Secara matematis, graf didefinisikan sebagai berikut:

Graf G didefinisikan sebagai pasangan

himpunan (V,E) yang dalam hal ini:

$V =$ himpunan tidak-kosong dari simpul-simpul (vertices atau node) $=\{v_1, v_2, \dots, v_n\}$ dan

$E =$ himpunan sisi (edges atau arcs) yang menghubungkan sepasang simpul $=\{e_1, e_2, \dots, e_n\}$

Atau dapat ditulis singkat notasi $G=(V,E)$, dengan V tidak boleh kosong, sedangkan E boleh kosong

Jadi, sebuah graf dimungkinkan tidak mempunyai sisi satu buah pun, tetapi simpulnya harus ada, minimal satu.

I.2 Jenis Graf

Graf dapat dikelompokkan menjadi beberapa kategori bergantung pada sudut pandang pengelompokannya. Pengelompokan graf dapat dipandang berdasarkan ada tidaknya sisi ganda atau sisi kalang, berdasarkan jumlah simpul, atau berdasarkan orientasi arah pada sisi.

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis:

1. Graf tak berarah

Graf tak berarah adalah graf yang sisinya tidak mempunyai orientasi arah. Pada graf tak berarah, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. Jadi $(v_j, v_k) = (v_k, v_j)$ adalah sisi

yang sama.

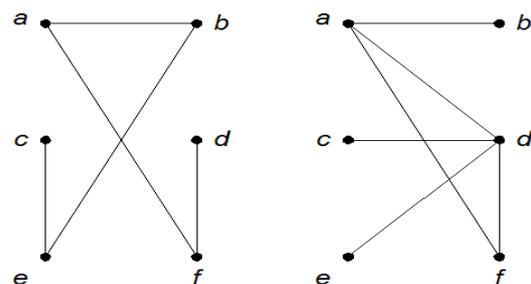
2. Graf berarah

Graf berarah adalah graf yang setiap sisinya diberikan orientasi arah. Jadi $(v_j, v_k) = (v_k, v_j)$ adalah sisi yang berbeda.

I.3 Pengertian Pohon

Pohon suatu graf tak berarah terhubung yang tidak mengandung sirkuit. Karena definisi pohon tersebut diacu dari teori graf, maka sebuah pohon dapat mempunyai sebuah simpul tanpa sebuah sisipun. Dengan kata lain, jika $G = (V,E)$ merupakan sebuah pohon, maka V tidak boleh berupa himpunan kosong, tetapi E boleh merupakan himpunan kosong.

Berdasarkan definisi tersebut, ada dua sifat penting pada pohon, yaitu terhubung dan tidak mengandung sirkuit. Yang dimaksud dengan terhubung adalah pada setiap pasang simpul pada pohon terdapat lintasan yang menghubungkannya. Sedangkan yang dimaksud tidak mengandung sirkuit berarti tidak terdapat lebih dari satu lintasan yang menghubungkan setiap pasang simpul pada pohon.



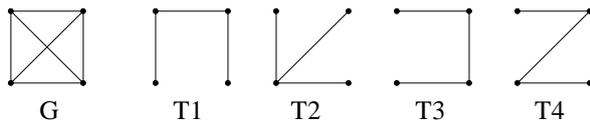
pohon

bukan pohon

Gambar 1. Perbedaan pohon dan bukan pohon

Sama seperti graf, pohon juga mempunyai beberapa jenis. Di antara berbagai jenis tersebut, yang perlu dibahas adalah pohon merentang. Pohon merentang adalah pohon yang dibentuk dari semua vertex dan beberapa sisi. Dimisalkan G adalah graf sederhana. Maka pohon merentang adalah upagraf (*subgraph*) dari graf G yang berupa sebuah pohon berisi setiap vertex G . Pohon

merentang dapat diperoleh dengan cara memutus sirkuit dari graf G seperti pada gambar berikut.



Gambar 2. Pohon-pohon merentang yang dapat dibentuk dari graf G

I.5 Bridge

Bridge merupakan piranti murni yang bekerja pada layer *Data Link* pada model OSI (*Open System Interconnection*) yang menjadi cikal bakal daripada *LAN switch*.

Ada tiga jenis *bridge*, yaitu:

- 1) *Transparent bridge* (untuk jaringan *Ethernet* dan *Token Ring*)
- 2) *Source-routing bridge* (untuk jaringan *Token Ring* saja)
- 3) *Source-routing transparent bridge* (untuk jaringan *Token Ring* saja)

II. SPANNING TREE PROTOCOL

II.1 Pengertian *Spanning Tree Protocol*

Spanning Tree Protocol (STP) adalah layanan yang memungkinkan *LAN switch* dan *LAN bridge* terinterkoneksi secara berlebihan dengan cara menyediakan mekanisme untuk mencegah *loop* yang tidak diinginkan dalam jaringan yang terjadi pada *bridge*.

Tanpa adanya STP Protokol, pada *frame Ethernet* akan terjadi *loop* untuk periode tak terbatas di dalam waktu jaringan dengan *link* berlebihan secara fisik. Untuk mencegah *loop* pada *frame Ethernet*, STP memblokir beberapa *port* dari *frame Ethernet* sehingga hanya satu jalur yang aktif ada di antara setiap pasang segmen LAN

(*collision domain*). Hasil STP adalah *loop frame Ethernet* tidak terbatas yang membuat LAN dapat digunakan. Satu hal yang perlu diperhatikan adalah bahwa *redundant link* tidak untuk keseimbangan beban, karena hanya satu *link* aktif.

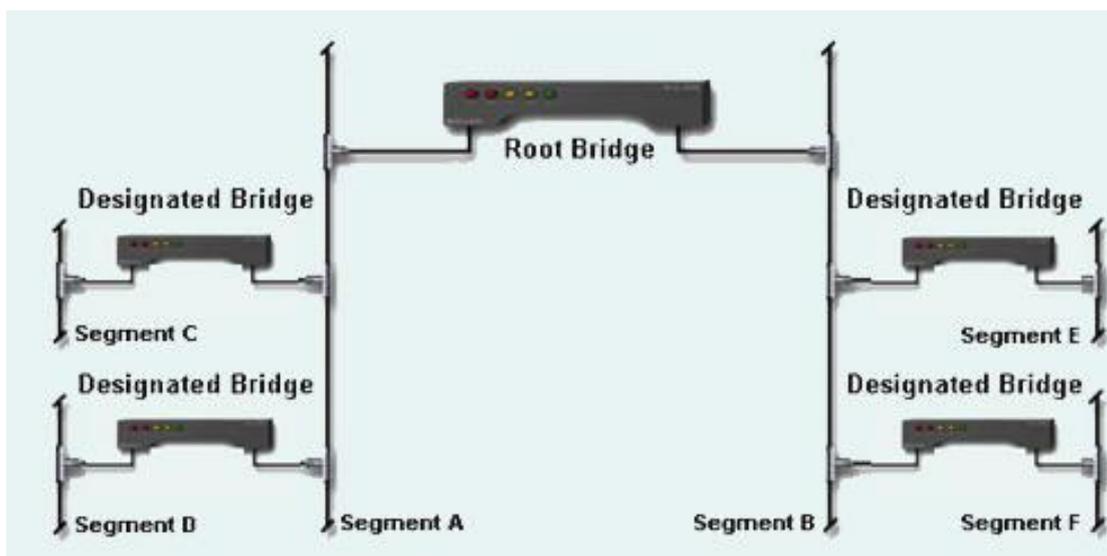
Loop terjadi bila ada rute/jalur alternatif diantara *host-host*. Untuk menyiapkan jalur *back-up*, STP membuat status jalur *back-up* menjadi *stand-by* atau diblok. STP hanya membolehkan satu jalur yang aktif (fungsi pencegahan *loop*) diantara dua *host* namun tetap menyiapkan jalur *back-up* bila jalur utama terputus.

Beberapa jalur aktif antara stasiun menyebabkan *loop* dalam jaringan. Jika *loop* ada dalam topologi jaringan, ada potensi untuk duplikasi pesan. Ketika *loop* terjadi, beberapa *switch* pada stasiun muncul di kedua sisi dari saklar. Kondisi ini membingungkan algoritma *forwarding* dan memungkinkan duplikat *frame Ethernet* untuk diteruskan.

Untuk memberikan redundansi jalur, *Spanning Tree Protocol* mendefinisikan sebuah *pohon* yang merentangkan semua *switch* yang aktif dalam jaringan. *Spanning Tree Protocol* memaksa jalur data tertentu yang berlebihan menjadi *standby state* (diblokir). Jika salah satu segmen jaringan dalam *Spanning Tree Protocol* menjadi tidak tercapai, algoritma *spanning tree* mengonfigurasi kembali topologi *spanning tree* dan membangun kembali *link* dengan mengaktifkan jalur siaga.

II.2 Cara Kerja *Spanning Tree Protocol*

Algoritma *spanning tree/STA* (disebut juga *Spanning Tree Protocol*) secara otomatis menemukan topologi jaringan dan membentuk suatu jalur tunggal yang yang optimal melalui suatu *bridge* jaringan dengan menugasi fungsi-fungsi berikut pada setiap *bridge*. Fungsi *bridge* menentukan bagaimana *bridge* berfungsi dalam hubungannya dengan *bridge* lainnya, dan apakah *bridge* meneruskan *traffic* ke jaringan-jaringannya lainnya atau tidak.



Gambar 3. *Spanning Tree Protocol* - Root bridge

1. Root bridge

Root bridge merupakan *master bridge* atau *controlling bridge*. *Root bridge* secara periodik mem-broadcast konfigurasi *message*. *Message* ini digunakan untuk memilih rute dan rekonfigurasi fungsi-fungsi dari *bridge-bridge* lainnya bila perlu. Hanya ada satu *Root bridge* per jaringan. *Root bridge* dipilih oleh administrator. Saat menentukan *Root bridge*, sebaiknya pilih *Root bridge* yang paling dekat dengan pusat jaringan secara fisik.

2. Designated bridge

Designated bridge adalah *bridge-bridge* lain yang berpartisipasi dalam meneruskan paket melalui jaringan. Mereka dipilih secara otomatis dengan cara saling tukar paket konfigurasi *bridge*. Untuk mencegah terjadinya *bridging loop*, hanya ada satu *Designated bridge* per segment jaringan.

3. Backup bridge

Semua *bridge* redundansi dianggap sebagai *Backup bridge*. *Backup bridge* mendengar *traffic* jaringan dan membangun database *bridge*. Akan tetapi mereka tidak meneruskan paket. *Backup bridge* ini akan mengambil alih fungsi jika suatu *Root bridge* atau *Designated bridge* tidak berfungsi.

Bridge mengirimkan paket khusus yang disebut *Bridge Protocol Data Units* (BPDU) keluar dari setiap *port*. BPDU ini dikirim dan diterima dari *bridge* lainnya digunakan untuk menentukan fungsi-fungsi *bridge*, melakukan verifikasi kalau *bridge* disekitarnya masih berfungsi, dan recovery jika terjadi perubahan topology jaringan.

Perencanaan jaringan dengan *bridge* menggunakan *Spanning Tree Protocol* memerlukan perencanaan yang hati-hati. Suatu konfigurasi yang optimal menuntut pada aturan-aturan berikut ini :

- Setiap *bridge* seharusnya mempunyai backup (yaitu jalur redundansi antara setiap segmen)
- Paket-paket harus tidak boleh melewati lebih dari dua *bridge* antara segmen-segmen jaringan
- Paket-paket seharusnya tidak melewati lebih dari tiga *bridge* setelah terjadi perubahan topology.

III. PEKEMBANGAN SPANNING TREE PROTOCOL

Dalam beberapa poin berikutnya, kita akan membahas beberapa versi STP berbeda:

• *Spanning Tree Protocol* (STP-802.1d 1998)

Merupakan versi awal dari STP yang hanya mendukung satu contoh *spanning tree* dalam jaringan *bridge*, biasanya disebut sebagai *Common Spanning Tree* (CST). Pada *trunk* 802.1Q yang membawa beberapa VLAN, satu VLAN – biasanya default atau VLAN 1 – akan menentukan topology *forwarding* untuk semua VLAN lainnya. Pada STP, ketika *port* diaktifkan atau ada perubahan dalam topology STP, dapat memakan waktu hingga 50 detik pada network *bridge* untuk jaringan dijematani untuk reconverge.

State pada switch STP :

- **Blocking** – *Port* yang akan menyebabkan *switching loop*, tidak ada *user data* yang dikirim atau diterima tapi ia akan menuju *forwarding mode* jika *link* terpakai lainnya gagal dan algoritma *spanning tree* menentukan port boleh bertransisi ke *forwarding state*. Data BPDU masih diterima dalam blocking state.
 - **Listening** – *Switch* memproses BPDU dan menunggu kemungkinan information yang akan menyebabkannya kembali ke blocking state untuk setiap segmen LAN
 - **Learning** – Saat *port* belum does menyampaikan *frames* (paket), ia akan mempelajari alamat *sources* dari *frames* yang diterima dan menambahkan mereka ke *filtering database* (*switching database*).
 - **Forwarding** - *Port* yang menerima dan mengirim data, operasi normal. STP masih memonitor incoming BPDU yang masuk yang mengindikasikan ia akan kembali ke blocking state untuk mencegah *loop*.
 - **Disabled** - Tidak sepenuhnya bagian dari STP, administrator jaringan secara manual dapat menonaktifkan *port*
- **Rapid Spanning Tree Protocol (RSTP-802.1w/802.1d 2004)**
Rapid Spanning Tree Protocol adalah penyempurnaan dari STP yang sebelumnya. Pada tahun 1998, IEEE memperkenalkan suatu evolusi dari *Spanning Tree Protocol*, yaitu *Rapid Spanning Tree Protocol*. RSTP menyediakan konvergensi *spanning tree* yang lebih cepat. Standar IEEE-2004 802.1d sekarang menggabungkan RSTP dan STP. Sementara STP hanya dapat mengambil 30 sampai 50 detik untuk merespon perubahan topologi, RSTP biasanya mampu menanggapi perubahan dalam waktu 3 kali *Hello time* (default: 6 detik). Yang disebut *Hello time* adalah konfigurasi interval waktu dan penting yang digunakan oleh RSTP untuk beberapa tujuan. Nilai default untuk *Hello time* adalah 2 detik.
- Peran jembatan *port* RSTP :
- **Root** - *forwarding port* yang merupakan *port* terbaik dari *Nonroot-bridge* sampai *Rootbridge*
 - **Designated** - *forwarding port* untuk setiap segmen LAN
 - **Alternate** - Jalur alternatif ke *root bridge*. Jalur ini berbeda daripada menggunakan *root port*.
 - **Backup** - Jalur *backup*/tambahan ke segmen dimana *bridge port* lain sudah terhubung.

- **Disabled** - Tidak sepenuhnya bagian dari STP, administrator jaringan secara manual dapat menonaktifkan *port*

Karena *Rapid Spanning Tree Protocol* adalah penyempurnaan dari STP yang sebelumnya, RSTP memiliki banyak karakteristik operasi dasar yang sama dengan STP. Namun, ada beberapa perbedaan penting yang terangkum sebagai berikut :

- Deteksi kegagalan pertukaran *root* dilakukan dalam 3 *Hello times*, yang mana terjadi dalam 6 detik jika *default*-nya belum diubah.
- *Port*-nya dapat dikonfigurasi sebagai *port* tepi jika *port*-*port* tersebut terpasang ke LAN yang tidak memiliki *bridge* lain yang terpasang. Transisi *port* tepi ini langsung ke *forwarding state*. RSTP masih terus memantau *port* untuk BPDU jika *bridge* tersambung. RSTP juga dapat dikonfigurasi untuk secara otomatis mendeteksi *port* tepi. Segera setelah *bridge* mendeteksi BPDU datang ke tepi *port*, *port* tersebut menjadi *port* non-tepi.
- Tidak seperti pada STP, RSTP akan merespon BPDU dikirim dari arah *root bridge*. *Bridge* pada RSTP akan "mengusulkan" informasi *spanning tree* ke *port* yang ditunjuk. Jika ada *bridge* RSTP lain menerima informasi ini dan terdeterminasi informasi ini adalah informasi *root* yang superior, ia akan mengeset semua *port* lain untuk melakukan pembuangan informasi tersebut. *Bridge* dapat mengirimkan sebuah "kesepakatan" kepada *bridge* pertama untuk mengonfirmasi informasi superior *spanning tree*-nya. *Bridge* pertama, setelah menerima kesepakatan tersebut, tahu ia dapat dengan cepat mentransisikan *port* itu ke keadaan *forwarding state* melewati transisi *listening/learning state*. Hal ini pada dasarnya menciptakan efek mengalir jauh dari *root bridge* dimana setiap *designated bridge* mengusulkan kepada tetangganya untuk menentukan apakah ia bisa membuat transisi yang cepat. Ini adalah salah satu elemen utama yang memungkinkan RSTP untuk mencapai konvergensi kali lebih cepat dari STP.
- Seperti dijelaskan dalam rincian peran *port* di atas, RSTP mempertahankan cadangan rincian tentang status pembuangan *port*. Hal ini untuk menghindari *timeout* jika *forwarding port* saat

ini gagal atau BPDU tidak diterima pada *root port* akar dalam interval tertentu.

- **Multiple Spanning Tree Protocol (MSTP-802.1s/802.1Q 2003)**

Multiple Spanning Tree Protocol mengonfigurasi *spanning tree* terpisah untuk setiap grup VLAN dan blok semua kecuali salah satu jalan alternatif yang mungkin dalam setiap *spanning tree*.

Jika hanya ada satu Virtual LAN (VLAN) dalam jaringan, STP tunggal bekerja secara tepat. Jika jaringan berisi lebih dari satu VLAN, *logical network* yang dikonfigurasi oleh STP tunggal akan bekerja, tetapi ada kemungkinan untuk membuat lebih baik menggunakan jalur alternatif yang tersedia dengan menggunakan pohon rentang alternatif untuk VLAN yang berbeda atau kelompok VLAN.

MSTP memungkinkan pembentukan *region* MST yang dapat menjalankan beberapa *Multiple Spanning Tree Instances* (MSTI). *Multiple regions* dan *bridge* STP lainnya saling berhubungan menggunakan salah satu *Common Spanning Tree* (CST).

MSTP mencakup semua informasi *spanning tree* dalam format BPDU tunggal. Hal ini tidak hanya mengurangi jumlah BPDU yang diperlukan pada LAN untuk mengomunikasikan informasi *spanning tree* untuk setiap VLAN, tetapi juga memastikan kompatibilitas dengan RSTP dan STP versi awal. MSTP melakukan hal ini dengan melakukan pengkodean tambahan informasi wilayah setelah BPDU RSTP standar sebanyak jumlah pesan MSTI (dari 0 sampai 64 kasus, walaupun dalam prakteknya banyak *bridge* yang kurang mendukung). Masing-masing pesan konfigurasi MSTI menyampaikan informasi *spanning tree* untuk setiap kasus. Setiap kasus dapat diberikan sejumlah VLAN terkonfigurasi dan *frame Ethernet* (paket) yang ditugaskan untuk VLAN tersebut untuk beroperasi pada *spanning tree* kapan pun mereka berada di dalam MST *region*. Untuk menghindari penyampaian seluruh VLAN untuk pemetaan *spanning tree* di setiap BPDU, *bridge* mengkodekan MD5 *digest* dari VLAN mereka ke table kasus dalam BPDU MSTP. *Digest* tersebut kemudian digunakan oleh *bridge* MSTP lain untuk menentukan apakah *bridge* tetangga berada pada MST *region* yang sama seperti *bridge* itu sendiri.

- **Per-VLAN Spanning Tree (PVST)**

Dalam lingkungan *Ethernet switch* dimana beberapa Virtual LAN berada, *spanning tree* dapat digunakan per Virtual LAN. Nama Cisco (perusahaan multinasional Amerika yang memproduksi barang-barang elektronik) untuk ini adalah *Per VLAN Spanning Tree* (PVST dan PVST +, merupakan protokol standar yang digunakan oleh *switch* Cisco). Baik protocol PVST dan PVST+ merupakan protokol hak milik Cisco dan mereka tidak dapat digunakan pada *switch* bagian ketiga. Meskipun jaringan *Force10* dan *Extreme* mendukung PVST +, jaringan *Extreme* melakukannya dengan dua keterbatasan (kurangnya dukungan pada *port* dimana VLAN ditandai / asli dan juga pada VLAN dengan ID 1). PVST hanya bekerja dengan ISL (protokol hak milik Cisco untuk enkapsulasi VLAN) karena tertanamnya ID *spanning tree*. PVST+ dapat menembus melintasi daerah MSTP.

- **Per-VLAN Spanning Tree Plus/Rapid-PVST+ (PVST+/RPVST+)**

PVST + adalah yang versi Cisco dari *spanning tree*. Dalam PVST +, VLAN masing-masing akan memiliki *spanning tree* sendiri. PVST + memiliki waktu konvergensi yang sama sebagai STP versi awal. Sejak konvergensi waktu PVST+ adalah sub-optimal, Cisco memasukkan fitur RSTP ke PVST + yang mereka sebut RPVST. Namun, RPVST + masih mengiklan pada alamat *multicast* dan dicadangkan untuk semua VLAN kecuali VLAN 1.

IV. KESIMPULAN

Kesimpulan yang didapat dalam pembahasan masalah *spanning tree protocol* pada makalah ini adalah sebagai berikut :

- 1) *Spanning Tree Protocol* merupakan salah satu aplikasi dari pohon merentang
- 2) Cara kerja *Spanning Tree Protocol* adalah menggunakan algoritma *spanning tree* yang secara otomatis menemukan topologi jaringan dan membentuk suatu jalur tunggal yang yang optimal melalui suatu *bridge* jaringan dengan menugasi fungsi-fungsi berikut pada setiap *bridge*
- 3) *Spanning Tree Protocol* mengalami perkembangan dari masa ke masa. Ada beberapa versi *Spanning Tree Protocol* dari semenjak awal ditemukannya hingga sekarang, antara lain *Spanning Tree Protocol* itu sendiri, *Rapid Spanning Tree Protocol*, *Multiple Spanning Tree Protocol*, *Per-VLAN Spanning Tree*, dan *Per-VLAN Spanning Tree Plus*.

REFERENSI

- [1] Munir, Rinaldi, "Matematika Diskrit", 2nd ed. Bandung: Informatika Bandung, 2009, hal. 353–486.
- [2] en.wikipedia.org/wiki/Spanning_tree_protocol
Tanggal akses 12 Desember 2010
- [3] www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/sw_ntman/cwsimain/cws2/cwsiug2/vlan2/stpapp.htm
Tanggal akses 12 Desember 2010

- [4] www.cs.umd.edu/~shankar/417-F01/Slides/chapter5b-aus.ppt
Tanggal akses 12 Desember 2010
- [5] www.juniper.net/us/en/local/pdf/implementation-guides/8010002-en.pdf
Tanggal akses 12 Desember 2010
- [6] www.sysneta.com/spanning-tree-protocol
Tanggal akses 12 Desember 2010

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.

Bandung, 12 Desember 2010

ttd

Muhammad Reza Mandala Putra 13509003