

BAB 2

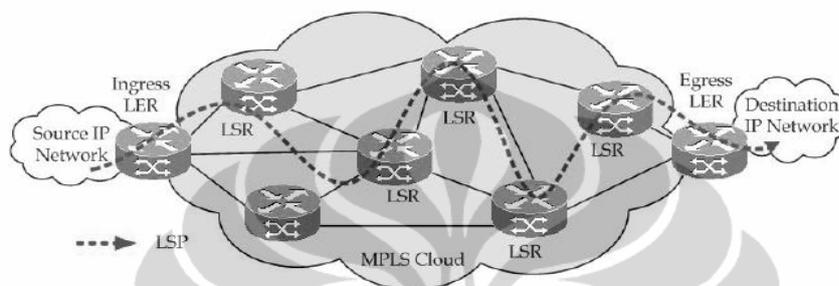
GMPLS DAN SISTEM DRAGON

2.1 Latar Belakang MPLS

Dewasa ini jaringan IP sangat luas penggunaannya, jaringan ini dibentuk dari beberapa komponen jaringan, seperti *server*, *router*, *hub* dan komponen jaringan lainnya yang dibutuhkan. Pencarian *route* dilakukan dengan menggunakan prefik alamat IP. Mekanisme pencarian *route* dilakukan dengan menentukan *interface output* berdasarkan alamat IP yang terdapat di dalam paket dan informasi yang terdapat di dalam *table forwarding* dengan metoda *best matching prefix*. Metoda ini melakukan pemeriksaan prefik *IP Address* dan mencocokkannya dengan *entries* di dalam tabel *forwarding*. Bila ditemukan yang paling cocok yaitu bila kesesuaian jumlah bit paling banyak, maka akan ditemukan *outgoing interface* nya, dan paket IP akan diteruskan oleh *router* ke jaringan yang lain. Selanjutnya *router* membuat dan memperbarui tabel *forwarding* dengan menggunakan protokol *routing*.

Pada MPLS teknik *forwarding* ini disederhanakan, dimana keputusan pengiriman paket didasarkan pada label. Bila paket memasuki jaringan MPLS, *Label Switching Router (LSR)* yang memiliki *database (Label Information Base (LIB))* akan menyediakan label untuk membuat keputusan pengiriman paket. Router MPLS membuat *Forward Equivalency Class (FEC)* yang digunakan untuk merepresentasikan trafik sesuai dengan kebutuhan rekayasa trafik yang sama dan dikaitkan dengan label. Pada *ingress node*, paket-paket *incoming* diperiksa oleh *Label Edge Router (LER)* dan diberikan label. Paket yang telah diberi label kemudian di *forward* melalui *Label Switched Path (LSP)*. Penanganan paket oleh LSR di sepanjang *forwarding path* diperlihatkan pada Gambar 2.1. Keputusan penerusan paket dilakukan oleh masing-masing LSR disepanjang *path* berdasarkan label MPLS pada paket. Dengan mekanisme ini *Label lookup* telah mengeliminasi proses pencocokan prefik terpanjang karena LSR tidak melakukan

pemeriksaan *IP header* paket untuk menentukan lompatan berikutnya. Setiap LSR melakukan *label swapping*, kemudian sebelum lompatan terakhir label dilepaskan dari *IP packet* oleh LSR, yang merupakan *router* terakhir sebelum egress *node*, dan dilanjutkan ke egress LER. Kemudian egress LER merutekan paket ini secara normal [8].



Gambar 2.1: Paradigma Label Switching MPLS [8]

2.2 Evolusi Menuju GMPLS

Sebagaimana yang telah dijelaskan sebelumnya, paradigma label *switching* telah dikembangkan untuk paket IP pada jaringan MPLS, hal ini juga dapat direalisasikan untuk perluasan teknologi *switching* yang berorientasi koneksi. Dengan adanya standar yang dikeluarkan IETF, yaitu *Generalized Multi Protocol Label Switching* (GMPLS) yang merupakan pengembangan dari *traffic engineering signaling protocol* MPLS(RSVP-TE) dan *traffic engineering routing protocols* (OSPF-TE and IS-IS-TE). GMPLS dapat mendukung *Packet Switching Capable* (PSC), *Layer2 Switching Capable* (L2SC), *Time-Division Multiplex* (TDM), *Lambda Switch Capable* (LSC), dan antar muka *Fiber Switch Capable* (FSC) dengan elemen *switching*. Ini berarti bahwa jaringan yang berbasis GMPLS mampu membawa trafik pemakai yang dihasilkan oleh jaringan IP melalui *layer 2* jaringan (misalnya, Ethernet atau ATM) atau TDM *network* (misalnya, *Digital Cross Connects*) ke jaringan *optic* [8].

Teknologi MPLS dan GMPLS menggunakan *traffic engineering* (TE) untuk mengoptimalkan penggunaan sumber daya jaringan dan meminimalkan kemacetan trafik. Untuk mengontrol *label switching path* (LSP) digunakan *explicit route* yang dapat mengatasi keterbatasan *bandwidth* dan *delay*. Pada

tahun 2005 IETF membentuk *task force* untuk membuat standar *Path Computation Element* (PCE). PCE mampu menghitung *routing path* berdasarkan informasi *network path* dan keterbatasan sumberdaya, baik untuk aplikasi intra maupun inter *domain*. Untuk mencapai fungsi tersebut GMPLS menggunakan protokol-protokol routing (OSPF-TE and IS-IS-TE), protokol pensinyalan (RSVP-TE) dan *link management protocol* (LMP). Protokol-protokol tersebut akan dijelaskan sebagai berikut :

2.2.1 OSPF-TE

OSPF adalah protocol *link state* yang bertugas untuk memelihara topologi dan membantu *router* untuk menemukan jalur terpendek ke *node* tujuan. Protokol ini telah dikembangkan dengan menambahkan kapabilitas jaringan *transport* dengan informasi yang berhubungan dengan ketersediaan *bandwidth*, proteksi *link*, *shared risk link groups*, penyiapan untuk *unnumbered links* sehingga dikenal dengan OSPF-TE. Protokol OSPF-TE memberitahukan tempat tersedianya sumberdaya seperti *bandwidth* untuk *link* tertentu untuk menempatkan LSP. Pada GMPLS juga terdapat informasi *Maximum bandwidth*, *Maximum Reservable bandwidth* dan *Unreserved bandwidth* [12]. Dalam menentukan pemilihan sebuah *route* LSP, perlu juga dipertimbangkan faktor *switching capability* dan *protection capability*.

Selain itu juga terdapat fitur lain yaitu *Shared Risk Link Groups* (SRLG). SRLG terdiri dari beberapa link yang menggunakan sumberdaya fisik yang sama, sehingga kegagalan sumberdaya akan mempengaruhi semua *link*. Informasi ini diberitahukan oleh OSPF-TE pada jaringan GMPLS [13]. Tipe *nodes* dan elemen jaringan yang berbeda juga didukung oleh jaringan GMPLS, hal ini direpresentasikan dengan menggunakan kombinasi *unique router ID* dan *link number* yang disebut “*Unnumbered Link*” [14].

2.2.2 RSVP-TE

Protokol pensinyalan RSVP [RFC 2205] digunakan untuk memesan dan menjaga sumberdaya di dalam jaringan. Protokol ini merupakan protokol *request/response* yang menggunakan *path message* untuk membangun *path* dari

ingress point ke *egress point*, dan *resv message* untuk memesan sumber daya di sepanjang *path*. *Path message* dikirimkan oleh *ingress node* menuju tujuan, dan pesan ini dilewatkan dari *node* ke *node* melalui jaringan hingga mencapai *egress*.

Proses kerja protokol RSVP adalah sebagai berikut, *path message* membawa spesifikasi trafik yang diminta oleh *ingress node*. Setiap *node* pada *path* memproses dan memeriksa *path message* kemudian menyampaikan pesanan layanan yang diminta, dan juga melakukan *updates* terhadap *path message* untuk dikirimkan ke tujuan. *Egress node* menghitung sumberdaya yang akan dipesan untuk memenuhi permintaan dari *ingress node*. Sedangkan *Resv message* dikirimkan oleh *egress node* sebagai balasan dari *path message*. pesan *resv* dilewatkan kembali *hop-by-hop* disepanjang *path* yang dilintasi oleh *Path message*. Ketika pesan *Resv* mencapai *ingress node* setelah melengkapai pengalokasian sumberdayanya, RSVP mengalir secara penuh.

Untuk mendeteksi kegagalan jaringan antara dua *node*, RSVP-TE menggunakan *Hello message*. RSVP-TE juga mengenalkan LSP *coloring* yang digunakan untuk mengambil keputusan routing, terdapat berbagai tipe yaitu salah satu harus tidak boleh dipakai, setidaknya salah satu harus dipakai, semua harus dipakai.

2.2.3 Pengembangan RSVP-TE Untuk GMPLS

Pengembangan RSVP-TE agar dapat bekerja pada lingkungan GMPLS yaitu dengan menambahkan beberapa objek baru untuk membawa informasi tambahan fungsi GMPLS. Objek pertama yaitu *Record Route Object (RRO)* pada *path* dan *Resv messages* digunakan RSVP-TE untuk merekam label-label. Saat GMPLS mendukung LSP dua arah, label pada masing-masing arah harus direkam di dalam RRO. Hal ini dilakukan dengan penambahan sebuah flag pemesanan yang digunakan untuk menunjukkan arah dari label.

Objek lainnya adalah pengurangan *Overhead* Protokol dimana bila terjadi *teardown* maka *PathErr message* dikirimkan, kemudian *initiator (ingress node)* mengirimkan *PathTear messages* disepanjang LSP untuk membebaskan sumberdaya. Sama halnya jika *egress node* ingin mengakhiri LSP, ia harus mengirimkan *ResvTear* dan kemudian menunggu untuk menerima *PathTear request* dari *ingress node*. GMPLS mengembangkan RSVP-TE dengan

menambahkan *flag* baru (*Path State Removed Flag*) [RFC 3473] di dalam *PathErr message*. *Flag* ini digunakan untuk menunjukkan bahwa status telah dirubah oleh *downstream node*.

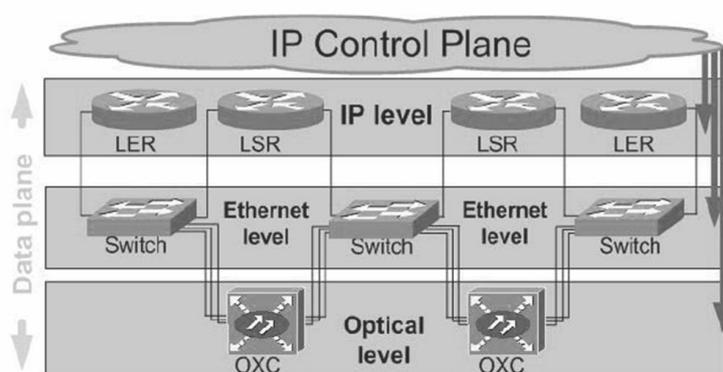
2.2.4 Protokol Manajemen Link

GMPLS menggunakan *Link Management Protocol* (LMP) untuk mengatur *TE link* disepanjang jalur. LMP termasuk *control manajemen* kanal yang digunakan untuk memelihara konektifitas kanal kontrol, *link property*, verifikasi konektifitas jalur fisik dan deteksi kesalahan (dan *error detection*).

2.3 Arsitektur dan Blok Bangunan GMPLS

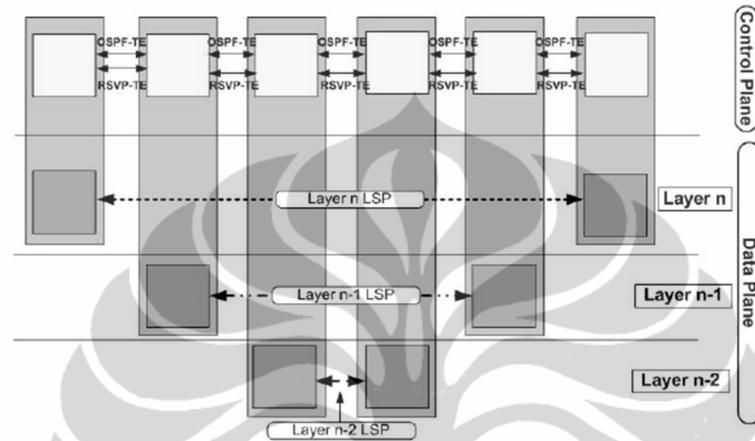
2.3.1 Bidang Kontrol Bersama

Pada standar GMPLS dan MPLS, bidang data dan bidang kontrol dipisahkan, hal ini dimaksudkan untuk meningkatkan efisiensi jaringan. Gambar 2.2 memperlihatkan trafik data disambungkan pada domain optikal, sedangkan kanal kontrol dapat berupa *out of band* atau *in band*. *Out of band* pada domain GMPLS berarti kanal kontrol diletakan pada jaringan fisik eksternal (misalnya. *Ethernet* atau *IP routed network*). Sedangkan pada *in band* kanal kontrol diletakan pada domain GMPLS yang sama tetapi secara logika terpisah dari GMPLS domain.



Gambar 2.2: Arsitektur GMPLS [8]

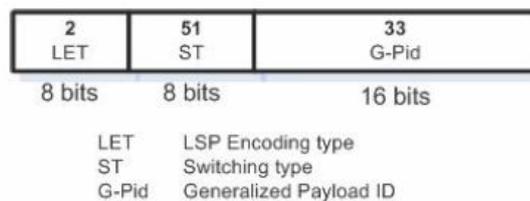
Pada domain GMPLS bidang kontrol didistribusikan ke berbagai tipe perangkat transmisi yang dibangun dengan menggunakan protokol pensinyalan. Bidang kontrol bersama dibangun berdasarkan *hop to hop* pada domain GMPLS diantara semua *node* jaringan untuk mempertukarkan pesan-pesan kontrol dengan menggunakan mekanisme *flooding*.



Gambar 2.3 Pembangunan Bidang Kontrol [8]

2.3.2 Label GMPLS

Untuk mendukung perangkat yang berbeda pada *switching domain*, GMPLS mengenalkan *generalized label*. Label ini berisi tiga bagian yang dinamakan *Encoding Type*, *Switching Type* dan *General Payload Identifier (G-PID)* [RFC 3471]. Masing-masing bagian merefleksikan informasi yang berbeda tentang trafik GMPLS. *Encoding type* merepresentasikan *encoding* dari LSP yang diminta atau digunakan. Kemampuan *switching* dari *link/node* direpresentasikan oleh *switching type*. Sedangkan tipe trafik yang dibawa oleh LSP ditunjukkan oleh GPID. Gambar 2.4 merepresentasikan *header label request* GMPLS untuk *Layer 2 Ethernet LSP*.



Gambar 2.4 GMPLS Label Request Header for Ethernet [8]

Tipe *encoding* LSP mempunyai nilai sesuai RFC 3471 seperti yang diperlihatkan pada tabel 2.1.

Tabel 2.1 : Daftar Tipe *Encoding* LSP sesuai RFC 3471[8]

Value	Type
1	Packet
2	Ethernet
3	ANSI/ETSI PDH
4	Reserved
5	SDH ITU-T G.707 / SONET ANSI T1.105
6	Reserved
7	Digital Wrapper
8	Lambda (photonic)
9	Fiber
10	Reserved
11	FiberChannel

Tabel 2.2 memperlihatkan bahwa terdapat 8 tipe *switching* yang masing-masingnya mempunyai nilai yang berbeda.

Table 2.2: Daftar Tipe *Switching* Sesuai RFC 3471[8]

Value	Type
1	Packet-Switch Capable-1 (PSC-1)
2	Packet-Switch Capable-2 (PSC-2)
3	Packet-Switch Capable-3 (PSC-3)
4	Packet-Switch Capable-4 (PSC-4)
51	Layer-2 Switching Capable (L2SC)
100	Time-Division-Multiplex Capable (TDM)
150	Lambda-Switch Capable (LSC)
200	Fiber-Switch Capable (FSC)

Sedangkan masing-masing type *switching* tersebut mempunyai *interfaces* yang fungsinya adalah sebagai berikut :

- *Packet switch capable (PSC) interfaces*, fungsinya adalah mengenali batas paket dan meneruskan data berdasarkan isi dari *packet header* (misalnya, *MPLS shim header*)
- *Layer-2 switch capable (L2SC) interfaces*, fungsinya mengenali perbatasan *frame/cell boundaries* dan melakukan *switch* data berdasarkan isi dari *frame/cell header* (misalnya, ATM VPI/VCI)
- *Time-division multiplex capable (TDM) interfaces*, fungsinya melakukan *switch* data berdasarkan *slot* waktu pada siklus pengulangan (misalnya, SONET/SDH DCS & ADM)

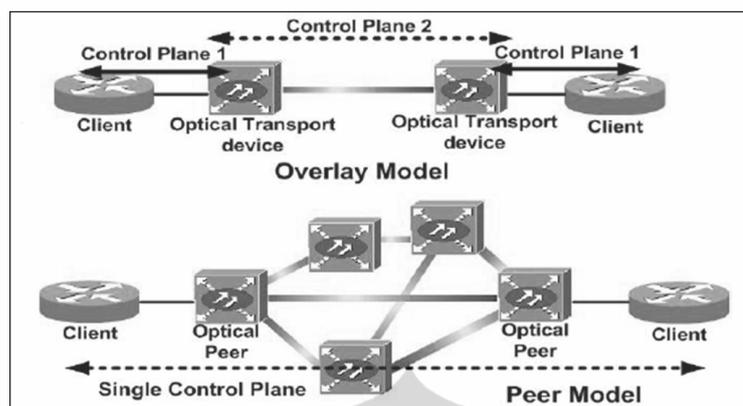
- *Lambda switch capable (LSC) interfaces* melakukan *switch* data berdasarkan pada *wavelength/waveband* ketika data diterima (misalnya, WSXC/*waveband switching* [WBS])
- *Fiber switch capable (FSC) interfaces* melakukan *switch* data berdasarkan posisi data di dalam ruang fisik (misalnya, OXC)

2.3.3 Suggested dan Upstream Labels

Alokasi label GMPLS diputuskan oleh *downstream node* pada *link*, saat *upstream node* mengirimkan *suggested label* ke *downstream node*, *downstream node* kemudian memutuskan apakah label ini dapat diterima atau tidak. Jika diterima, ia memberitahukan label kembali ke *upstream node* dengan menggunakan protokol pensinyalan. *Downstream node* juga menggunakan label ini untuk mengirimkan *traffic* ke *upstream node*. *Upstream node* dapat menyarankan sebuah label, dan dapat juga mengatur pemilihan label pada *downstream nodes*, tapi *downstream* mempunyai hak untuk menolaknya dan memilih sendiri dari ruang label.

LSP pada GMPLS dua arah [RFC 3945] sedangkan pada MPLS satu arah. GMPLS mendukung secara serentak pembangunan LSP *downstream* dan *upstream* dengan menggunakan seperangkat pesan-pesan pensinyalan. Untuk melakukan *setup* LSP dua arah tersebut digunakan label oleh trafik yang mengalir pada arah yang berlawanan (misalnya dari *egress* ke *ingress*) dan didistribusikan oleh *upstream nodes* ke *downstream nodes*. Label ini disebut *upstream lable* [10].

Dua model penerapan layanan GMPLS yaitu model *overlay* dan *peer*. Model *overlay* menyembunyikan rincian *optical core transport network* dari *client* nya dan menggunakan *User-to-Network Interface* (UNI). Pada model ini dioperasikan dua bidang kontrol yang berbeda, satu dioperasikan pada jaringan *transport* inti *optical* dan lainnya antara inti dan *client*, sehingga *Clients* tidak terlihat pada jaringan inti [10]. Model *peer* hanya menggunakan satu bidang kontrol. Pada model ini semua *nodes* mempunyai *complete view* jaringan. Kedua model ini diperlihatkan pada Gambar 2.5.



Gambar 2.5 Model *Overlay* dan *Peer* [8]

2.4 Label Switched Paths (LSP)

LSP adalah *path* yang dibuat pada ingress LER, *intermediate* LSRs dan *egress* LER sesuai dengan FEC yang diinginkan. LSP dikonstruksi sebelum data ditransmisikan. Terdapat dua cara untuk membuat LSP :

1. *Hop-by-hop routing*, masing-masing VLSR disepanjang *path* membuat keputusan kemana meneruskan paket yang diterimanya.
2. *Explicit routing*, *ingress* LER menentukan satu atau lebih VLSR *intermediate* untuk mencapai *egress* LER. *Path* dibangun sesuai keterbatasan QoS seperti *bandwidth*, *delay* untuk memenuhi QoS dan *traffic engineering*.

2.5 Sistem DRAGON

DRAGON merupakan proyek penelitian dan pengembangan teknologi untuk menyediakan sumberdaya jaringan yang dinamis berbasis *interdomain* melalui teknologi jaringan yang beragam. Dengan proyek ini dapat diciptakan layanan *transport end to end* yang dinamis, transparan dan dapat dikelola pada aplikasi *high-end e-Science*. DRAGON telah mengembangkan komponen perangkat lunak yang diperlukan untuk bidang kontrol IP. Bidang kontrol ini dapat menyediakan layanan yang cepat antar domain yang berkaitan dengan otentikasi, otorisasi, akuntansi, penjadwalan, dan *end system*.

2.5.2 Arsitektur Jaringan

Arsitektur jaringan DRAGON terdiri dari sejumlah jaringan domain otonom, masing-masing secara sepihak dapat menetapkan kebijakan pengelolaan lalu lintas internal untuk pengaturan *bi-lateral peering* dengan domain eksternal lainnya yang diinginkan dan disepakati bersama. Arsitekturnya didasarkan pada *switching* dan *forwarding node* yang mendukung hirarki label GMPLS. Hierarki label GMPLS terdiri dari label paket (misalnya MPLS), label TDM (misalnya SONET), label panjang gelombang, dan label serat optik.

Label Switch Router (LSR) merupakan *switching node* yang beroperasi pada lalu lintas label GMPLS. Masing-masing LSR menjalankan protokol *intra-domain routing* GMPLS (GMPLS-OSPF) dan protokol pemesanan sumberdaya GMPLS (GMPLS-RSVP). Kedua protocol ini dapat mengontrol pembentukan *Label Switched Paths (LSP)*, sehingga dapat menyediakan koneksi *end-to-end* untuk layanan multi lalu lintas di *transportation layer* yang konsisten dengan jenis label yang telah disepakati. Arsitektur DRAGON menyediakan komponen-komponen, *Network Aware Resouce Broker (NARB)*, *Virtual LSR (VLSR)*, dan *Application Specific Topology Definition Language (ASTDL)*.

Dalam rangka membangun LSP *end-to-end* antar-domain, pertukaran informasi jaringan harus sesuai dengan label yang telah disepakati. Konsep-konsep yang diusulkan adalah :

- NARB berfungsi untuk membangun end-to-end antar-domain LSP, pertukaran informasi jaringan harus sesuai dengan label masing-masing.
- VLSR berfungsi untuk menerjemahkan protokol GMPLS standar dalam perangkat protokol khusus, untuk memungkinkan rekonfigurasi perangkat non-MPLS.
- ASTDL berfungsi untuk memformalkan layanan jaringan dan menyederhanakan definisi dan deskripsi topologi jaringan kompleks yang digunakan.
- Pengembangan alat dan *middleware library* untuk antarmuka aplikasi pengguna dengan *routing*, *signaling*, dan kerangka keamanan yang diusulkan arsitektur DRAGON [1].

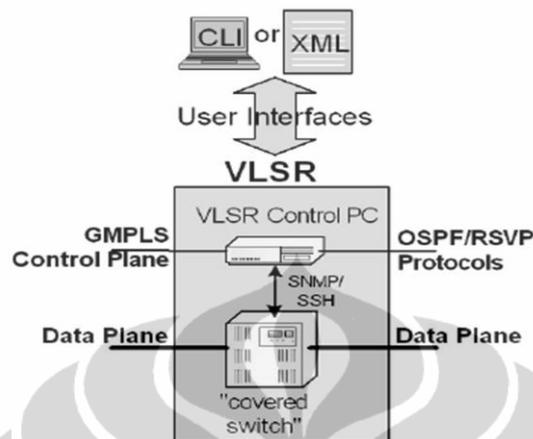
2.5.2 Implementasi VLSR pada GMPLS

VLSR menyelenggarakan mekanisme untuk mengintegrasikan perangkat dan jaringan non GMPLS dengan jaringan GMPLS dengan menyiapkan layanan *end-to-end* GMPLS. VLSR mentranslasikan protokol standar GMPLS ke protokol perangkat tertentu, sehingga memungkinkan untuk melakukan rekonfigurasi ulang perangkat non-GMPLS. VLSR merupakan kombinasi PC yang menjalankan GMPLS berbasis perangkat lunak bidang kontrol dan *switch fabric*. PC VLSR terdiri dari bidang kontrol yang meliputi OSPF-TE dan RSVP-TE dan bertindak sebagai *proxy agent* untuk perangkat non-GMPLS.

Penggunaan utama VLSR pada proyek DRAGON adalah untuk mengontrol *switch* ethernet melalui bidang kontrol GMPLS. Disamping itu VLSR juga telah diadaptasi untuk mengontrol *switch* TDM dan Optikal. Aplikasi VLSR untuk lingkungan Ethernet merupakan modifikasi OSPF-TE dan RSVP-TE sehingga dapat menyiapkan rangkaian Ethernet didasarkan pada konfigurasi VLAN. Pada masing-masing *switch Ethernet*, VLSR mentranslasikan pesan-pesan pensinyalan RSVP-TE kedalam perintah-perintah *switch local* dan membuat *VLAN-ports* yang dilengkapi dengan jaminan *bandwidth*. VLSR PC menggunakan kombinasi RFC 2674 SNMP dan *Command Line Interface (CLI)* untuk mengontrol *Ethernet switch fabric* secara lokal.

Gambar 2.6 menggambarkan konfigurasi VLSR. Informasi yang diberitahukan oleh VLSR OSPF-TE dikumpulkan di dalam *traffic engineering database* oleh NARB. NARB adalah entitas yang merepresentasikan sebuah domain dan beroperasi sebagai *listener* protokol *intradomain routing*. NARB dilengkapi dengan RCE menggunakan informasi *traffic engineering database* untuk *komputasi path*. Hasil komputasi Ethernet circuit yang diminta berupa *Explicit Route Object (ERO)* yang disediakan untuk *RSVP-TE signaling engine* untuk menyiapkan LSP. Fungsi NARB lainnya adalah melakukan *inter-domain traffic engineering routing exchanges*. Terdapat satu NARB per *domain* yang menjaga hubungannya dengan *domain* NARB tetangganya. Semua NARB mempertukarkan informasi topology untuk komputasi dan pensinyalan interdomain dapat dilakukan. Pertukaran topology didasarkan pada topologi

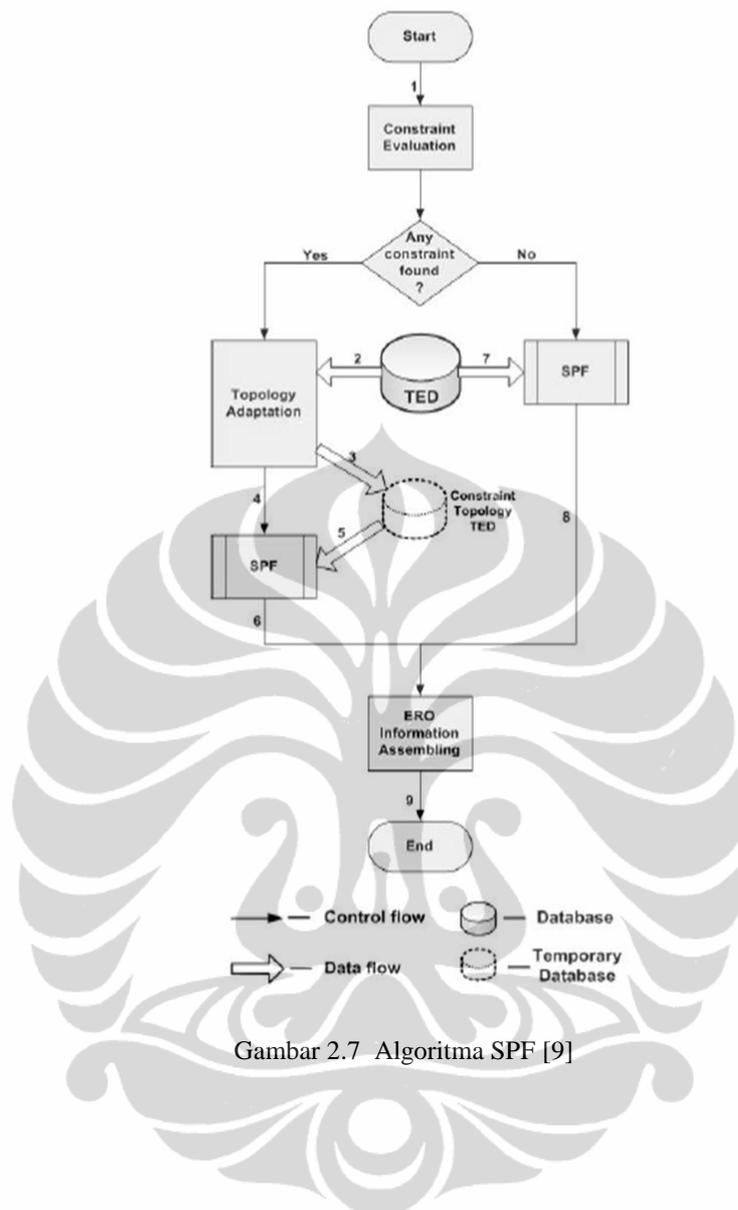
aktual atau *abstracted topologies* yang ditentukan konfigurasi administrative [16].



Gambar 2.6 Arsitektur VLSR [16]

2.5.2.1 DRAGON OSPF-TE

OSPF merupakan kontrol *routing link-state* yang dikembangkan untuk jaringan IP. OSPF mengirimkan *Link State Advertisements (LSAs)* ke semua *router* lainnya yang berada pada area yang sama. Pada setiap *router* OSPF menjaga *database* yang berisi topologi jaringan. Berdasarkan informasi yang ada di *database*, tabel *routing* dihitung dengan pengkonstruksian jalur terdekat. Penghitungan dilakukan dengan mengakumulasikan informasi kondisi *link* dengan menggunakan algoritma SPF untuk menghitung jalur terdekat disetiap *node*. OSPF menghitung kembali jalur terdekat bila terjadi perubahan topologi jaringan. Proyek *DRAGON* telah mengembangkan paket perangkat lunak *routing open source* GNU Zebra, termasuk fungsi-fungsi GMPLS yang diperlukan. Distribusi GNU Zebra adalah perangkat protokol yang melingkupi berbagai kontrol jaringan seperti RIP, OSPF, BGP dan lainnya. Perangkat lunak DRAGON OSPF-TE dikembangkan dari modul daemon OSPF GNU Zebra. Bagan alir algoritma SPF dapat dilihat pada Gambar 2.7



Gambar 2.7 Algoritma SPF [9]

2.5.2.2 DRAGON RSVP-TE

RSVP adalah protokol pensinyalan yang memungkinkan pengirim dan penerima berkomunikasi dalam melakukan *setting reserved highway* untuk melakukan transmisi data dengan QoS yang diinginkan. Aplikasi ini dapat dijalankan pada *IP end systems*. Proyek DRAGON dikembangkan dalam bentuk *open source KOM RSVP Engine* menyelenggarakan implementasi dari RSVP. Modifikasi yang dilakukan oleh DRAGON adalah sebagai berikut :

- Penambahan ekstensi GMPLS TE untuk RSVP
- Perluasan KOM-RSVP API untuk melewati konfigurasi spesifik dari VLSR untuk RSVP daemon.

- Penambahan fungsionalitas sehingga dapat mengontrol *Ethernet switches* melalui *Simple Network Management Protocol* (SNMP).
- Penambahan fungsionalitas sehingga dapat mengontrol *EoS subnets* melalui TL1.
- Mendukung pensinyalan *VLSR-to-VLSR*.

2.5.2.3 *Network Aware Resource Broker* (NARB)

NARB merupakan komponen bidang kontrol pada DRAGON berupa *agent* yang merepresentasikan *Autonomous* atau *Administrative Domain* (AD). NARB sebagai *path computation engine* dari *end-systems* atau perangkat lain bertugas untuk mencari ketersediaan *traffic engineered paths* antara pasangan *node* sumber dan tujuan. Proyek DRAGON memisahkan fungsi NARB dan *Resource Computation Element* (RCE) dimana NARB melakukan fungsi-fungsi *higher-level* seperti *domain level topology abstraction*, *inter-domain path computation*, *inter-domain routing* dan *end-to-end Label Switched Path (LSP) management*. Sedangkan RCE menyediakan basis data sumberdaya dasar untuk NARB dan layanan komputasi *path*.

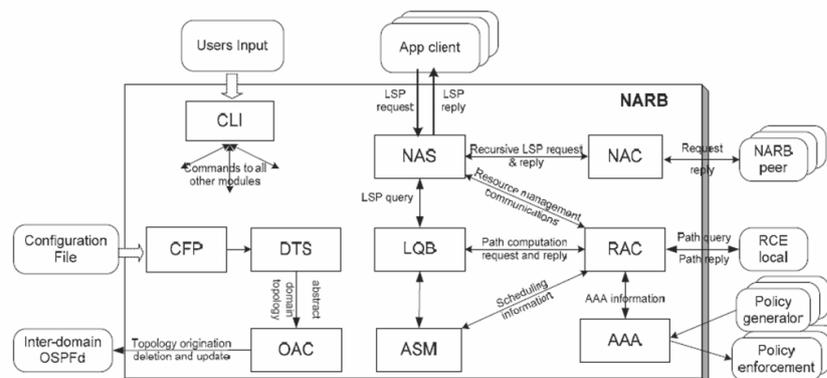
Pasangan NARB melintasi *domains* dan mempertukarkan informasi topology untuk mengaktifkan *inter-domain path computation* dan *end-to-end LSP provisioning*. Pertukaran topologi antar domain didasarkan pada topologi aktual yang didapatkan dari RCE dengan menggunakan algoritma OSPF. NARB memiliki kapabilitas sebagai berikut :

- *Intra-domain routing listener*
- *Intra-domain path computation*
- *Inter-domain routing*
- *Domain level topology abstraction (in support of inter-domain routing)*
- *Inter-domain path computation*
- *Label Switched Path (LSP) management*

NARB dirancang dalam bentuk perangkat lunak yang berdiri sendiri dapat menyediakan layanan banyak *client* secara serentak. Masing-masing NARB *client* melakukan sambungan dengan *common server port* dan menjaga hubungan TCP selama hubungan terjadi. NARB membuat sesi *internal server* untuk masing-

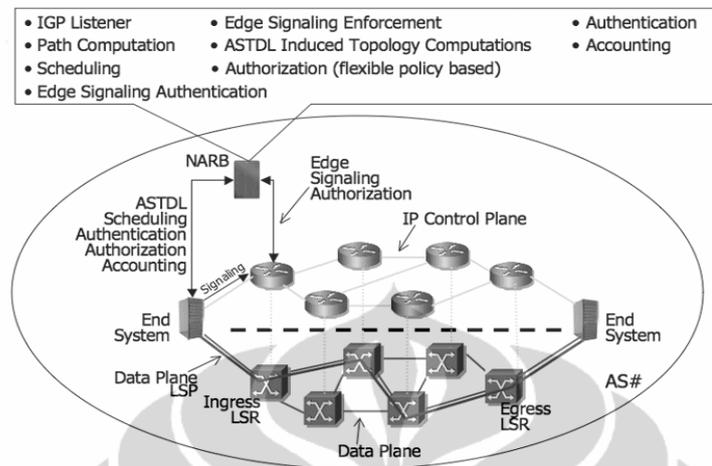
masing *client*. Server NARB menentukan *Application Programming Interface* (API) untuk berkomunikasi dengan clientnya. NARB juga menjaga koneksi TCP dengan RCE, OSPF *daemons* dan *peer* NARB. Struktur perangkat lunak dan proses penyiapan LSP, ditunjukkan pada Gambar 2.8. Pada gambar terlihat struktur NARB terdiri dari beberapa modul, yaitu :

- *Configuration File Processing* (CFP), modul ini membaca dan menginterpretasikan perintah-perintah konfigurasi dari berkas konfigurasi. Perintah-perintah konfigurasi *typical* termasuk ringkasan deskripsi topologi dan metoda untuk berkomunikasi dengan OSPFd dan pasangan NARB.
- *Domain Topology Summarization* (DTS), modul ini mentranslasikan DTS ke OSPF-TE *Link States Advertisements* (LSAs), menjaga dan memperbaharui *originated domain topology*.
- OSPF API Client (OAC), adalah *GNU Zebra OSPF client* untuk berkomunikasi dengan *inter-domain GNU Zebra OSPF daemon*.
- NARB API Server (NAS), modul ini membuat sesi server NARB, melayani *NARB clients* dan menginterpretasikan pesan-pesan NARB API.
- LSP *Query Broker* (LQB), modul ini menginterpretasikan tugas-tugas LSP *client* dan melakukan logika komputasi *inter-domain* LSP dengan dukungan RCE local dan pasangan NARB.
- RCE API *Client Module* (RAC), modul ini berkomunikasi dengan *local RCE server* untuk melakukan komputasi *path* dan fungsi manajemen sumber daya.
- NARB API *Client Server* (NAS), modul ini berkomunikasi dengan *peer NARB servers* melalui NARB API, khususnya untuk *inter-domain routing*.



Gambar 2.8 : Struktur Perangkat Lunak NARB [3]

Fungsi-fungsi intradomain *NARB* dapat dilihat pada gambar 2.9.

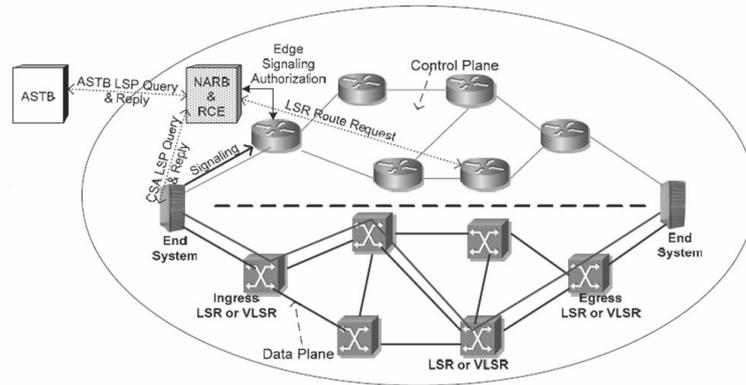


Gambar 2.9 : Intradomain NARB dan Fungsinya [17]

2.6.2.4 Path Computation Element (PCE)

PCE adalah *entitas* yang mampu menghitung *path* atau *route* berdasarkan topologi jaringan dan keterbatasan *bandwidth* yang tersedia. PCE dapat diterapkan pada intra area, inter area, inter *Autonomous System* (AS) dan inter-layer TE. PCE secara fungsi terpisah dari *Label Switch Router* (LSR) [2]. Proses kerja perhitungan *path* adalah sebagai berikut, pertama LSR meminta PCE untuk melakukan perhitungan *path*. Sebelum pensinyalan pembentukan LSP, PCE merespon LSR dengan hasil perhitungan *path*, LSR disebut *Path Computation Client* (PCC). PCE terdiri dari fungsi *path computation*, IGP, TE database, dan PCE-PCE *interface*. PCE mengumpulkan informasi topologi TE dengan fungsi IGP dan menyimpannya didalam TED. PCE digunakan karena beragamnya kebijakan TE carrier dan optimisasi global [2].

Pada sistem DRAGON, PCE disebut juga dengan *Resource Computation Element* (RCE). Sebagaimana dilihat pada Gambar 2.8, NARB/RCE menerima permintaan pembentukan LSP dari CSA, ASTB atau LSR/VLSR. NARB/RCE juga berpartisipasi dalam penyediaan LSP dengan *reserving*, *scheduling* dan *updating resource states* dalam *database* sumberdayanya.



Gambar 2.10 : Konfigurasi NARB/RCE pada Jaringan DRAGON [17].

2.7 Skalabilitas Jaringan

Skalabilitas jaringan menunjukkan kemampuan jaringan dalam menangani perkembangan jumlah *resource*. Skalabilitas merupakan *issue* penting pada sistem elektronik, database, routers, and networking. Sistem scalable adalah sistem yang dapat memperbaiki kinerjanya setelah penambahan *hardware*, sesuai dengan kapasitas yang ditambahkan. Pada GMPLS hal-hal yang harus diatasi adalah jumlah label yang terbatas, jumlah *link* yang sangat banyak, sehingga skalabilitas protokol *Link state* sangat diperlukan [18]