UNIVERSITAS INDONESIA

PERBANDINGAN PERFORMANSI APLIKASI FTP PADA JARINGAN PADA IPv4 DAN IPv6 DENGAN MPLS

SKRIPSI

RENY DWI WIJAYANTI

0706199810

FAKULTAS TEKNIK ELEKTRO

DEPOK

JUNI 2009

UNIVERSITAS INDONESIA

PERBANDINGAN PERFORMANSI APLIKASI FTP PADA JARINGAN PADA IPv4 DAN IPv6 DENGAN MPLS

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

RENY DWI WIJAYANTI

0706199810

FAKULTAS TEKNIK ELEKTRO

DEPOK

JUNI 2009

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk saya nyatakan benar.

Nama : Reny Dwi Wijayanti

NPM : 0706199810

Tanda tangan:

Tanggal : 30 Juni 2009

LEMBAR PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Reny Dwi Wijayanti
NPM : 0706199810
Program Studi : Teknik Elektro
Judul Skripsi : Perbandingan Performansi Aplikasi FTP pada jaringan IPv4 dan IPv6 dengan MPLS

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Ir. Endang Sriningsih MT.,Si ( )

Penguji : Dr. Ir. Anak Agung Putri Ratna M.Eng ( )

Penguji : Prima Dewi Purnamasari ST., MT., MSc ( )

Ditetapkan di : Depok

Tanggal : 30 Juni 2009

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

Ir. ENDANG SRININGSIH, MT.,Si

selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, Juni 2009

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah
ini:

Nama : Reny Dwi Wijayanti
NPM : 0706199810
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :
Perbandingan Performansi Aplikasi FTP pada Jaringan IPv4 dan IPv6 dengan MPLS

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 30 Juni 2009
Yang menyatakan

(.....)

ABSTRAK

Nama : Reny Dwi Wijayanti
Program Studi : Teknik Elektro
Judul : Perbandingan Performansi Aplikasi FTP pada Jaringan IPv4 dan IPv6 dengan MPLS

IPv6 sebagai protokol internet generasi mendatang, diharapkan dapat menjadi teknologi IP masa kini dan mendatang untuk mengatasi segala keterbatasan, hambatan yang dihadapi dalam pengembangan dan penerapan layanan baru. Konvergensi sejauh mungkin ke arah teknologi yang berbasis IP sudah tidak dapat dihindari lagi. Dengan ruang alamat sebesar 128 bit, maka IPv6 meningkatkan jumlah alamat IP yang tersedia untuk layanan baru. Dalam penerapannya, IPv4 pada jaringan MPLS harus dapat diintegrasikan dengan IPv6 untuk kemudian ditingkatkan menjadi IPv6. Pada Skripsi ini dilakukan uji coba performansi jaringan MPLS dalam perbandingannya antara IPv4 dan IPv6 untuk aplikasi FTP. Metode yang dilakukan adalah dengan melakukan studi literatur, perancangan dan implementasi kemudian melakukan pengujian. Parameter-parameter uji yang digunakan adalah delay paket, transfer time dan throughput.
Dari hasil pengujian didapatkan delay MPLS IPv4 lebih kecil 92.65% - 98.3% dibanding MPLS IPv6, transfer time MPLS IPv4 lebih cepat 95.26% - 105.15% dibanding jaringan MPLS IPv6, dan throughput MPLS IPv4 lebih besar 96.17% - 96.35% dibanding MPLS IPv6
Kata kunci:
MPLS, FTP, delay, throughput

ABSTRACT

Name : Reny Dwi Wijayanti
Study Program : Electronic Engineering
Title : MPLS Performances Comparison on IPv4 and IPv6 Packet for FTP Application

IPv6, as a next generation Internet Protocol, is promised to be the IP technology present and for the next future in order to overcome all of limitation and problems faced along the development and implementation of such new services. Converging as deep as possible to the new technology based on IP is can not be avoided. With a 128 bit of addressing, IPv6 increasing the amount of IP addressing that needed by new services. On the implementation,, IPv4 over MPLS network must be integrated with IPv6 protocol then it can be increased to the full IPv6 network. In this final project, we doing performance comparison testbed over MPLS network in comparison with IPv4 and IPv6 packet for FTP application. This testbed is done by literature study, design and implementation then evaluating the network. The test parameter is delay packet, transfer time and throughput.
The result show that delay MPLS IPv4 92.65% - 98.3% better than MPLS IPv6. Transfer time of MPLS IPv4 95.26% - 105.15% quicker than MPLS IPv6 and MPLS IPv4 throughput 96.17% - 96.35% higher than MPLS IPv6.
Keyword:
MPLS, FTP, delay, throughput

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

IPv6 (*Internet Protocol version* 6) dapat dikatakan sebagai *Internet Protocol Next Neneration* (IPng) karena memang didesain oleh *Internet Engineering Task Force* (IETF) untuk menggantikan protokol internet yang umum dipakai saat ini yaitu *Internet Protocol version* 4 (IPv4). Alamat IPv4 pada dasarnya menggunakan metode pengalamatan berbasis 32 bit, yang berarti mampu mengakomodasi jumlah pengalamatan sampai dengan 2^{32} atau sekitar 4,294 x 10^9. Dalam aplikasinya alamat yang mampu diakomodir kurang dari jumlah tersebut karena pada IPv4 terdapat adanya IP *network* dan IP *broadcast* yang tidak dapat digunakan sebagai alamat *host*. Seiring dengan perkembangan internet yang semakin pesat maka kebutuhan akan jumlah alamat juga makin bertambah. Karena itulah digagas IPv6 untuk mengatasi beberapa kekurangan IPv4.

Alamat IPv6 adalah metode pengalamatan yang menggunakan metode pengalamatan berbasis 128 bit, yang berarti mampu mengakomodasi jumlah pengalamatan sampai dengan 2^{128} atau sekitar 3,402 x 10^{38}. IPv6 dirancang sedemikian rupa agar memiliki kinerja yang lebih handal bila dibandingkan dengan IPv4 seperti dalam pengiriman paket, *security*, *authentication* dan QoS (*Quality Of Service*). Selain itu diharapkan IPv6 juga mampu memberikan fitur-fitur lain yang lebih kompleks yang akan dikembangkan lagi.

Sejak ditemukan pada tahun 1997, *Multi Protocol Label Switching* (MPLS) telah berkembang luas dan telah digunakan oleh banyak *provider* telekomunikasi pada jaringan IPv4 mereka dengan bermacam layanan yang ditawarkan seperti MPLS *Virtual Private Network* (VPN), MPLS *Quality of Service* (QoS), *MPLS Traffic Engineering* (TE), dan lain-lain. Dengan mengusung teknologi *tag-switching*, MPLS mampu menggabungkan keunggulan *switching* di layer 2 dan *routing* di layer 3. MPLS menawarkan mekanisme pengiriman paket data yang sederhana dan cepat dengan melakukan pembacaan label pada tiap paket data yang masuk. Pembacaan paket hanya dilakukan pada saat masuk dan

keluar jaringan MPLS. Berbeda dengan konsep routing konvensional di mana tiap router membaca dan memeriksa alamat tujuan yang terdapat pada paket data.

Perkembangan internet yang sangat pesat tidak menutup kemungkinan terjadinya kekurangan alokasi *host* pada alamat IPv4 saat ini, sedangkan untuk melakukan migrasi secara keseluruhan pada jaringan IPv4 ke IPv6 memerlukan biaya yang cukup besar. Beberapa skenario integrasi telah dikembangkan untuk memanfaatkan infrastruktur IPv4 yang ada dan menambahkan layanan IPv6 tanpa membutuhkan perubahan pada jaringan *backbone*. Salah satunya adalah IPv6 *Provider Edge Router* (6PE) *over* MPLS.

Pada 6PE memiliki beberapa keuntungan terutama dari segi ekonomi karena tidak diperlukan *upgrade* infrastruktur maupun perubahan konfigurasi pada *backbone* jaringan MPLS IPv4. Perubahan hanya terjadi pada sisi *Provider Edge* yang dapat dilakukan dengan cara meng*upgrade* PE menjadi 6PE (*dual-stack* IPv4 dan IPv6) atau meng*install* 6PE baru. Selain itu, pada *Customer Edge* (CE) juga tidak diperlukan adanya perubahan yang berarti. Hal ini dapat mengurangi biaya pada proses migrasi atau integrasi IPv4 ke IPv6. 6PE juga dapat dikembangkan untuk mendukung layanan VPN untuk IPv6 sebaik pada VPN MPLS IPv4 seperti isolasi trafik dan QoS.

1.2 Tujuan Penulisan

Tujuan penulisan skripsi ini adalah untuk merancang dan menganalisa perbandingan performa jaringan MPLS IPv4 dengan jaringan IPv6 *Provider Edge over* MPLS untuk aplikasi *File Transfer Protocol* (FTP). Performa yang diamati meliputi *delay*, *throughput*, dan *transfer time*.

1.3 Pembatasan Masalah

Masalah yang dibahas dalam skripsi ini adalah perancangan dan pengukuran performa jaringan MPLS IPv4 dengan jaringan IPv6 *Provider Edge over* MPLS (6PE) untuk aplikasi *File Transfer Protocol* (FTP).

1.4 Metode Penulisan

Untuk menyusun skripsi ini, akan dilakukan dengan menggunakan 3 buah *router* yang terhubung langsung dengan konfigurasi sebagai jaringan MPLS IPv4 dan MPLS IPv6 yang diinterkoneksikan dengan komputer untuk aplikasi, yaitu FTP. Untuk pengambilan datanya, dilakukan pengetesan untuk beberapa file dengan ukuran yang berbeda untuk didapat beberapa nilai parameter yaitu *delay*, *thoughput*, dan *transfer time*.

1.5 Sistematika Penulisan

Sistematika penulisan skripsi ini adalah sebagai berikut :

Bab I Berisi tentang latar belakang masalah, tujuan penulisan, pembatasan masalah, dan sistematika penulisan.

Bab II Membahas tentang perkembangan dan gambaran umum mengenai IPv6, MPLS dan 6PE

Bab III Membahas tentang perancangan sistem.

Bab IV Berisi hasil data yang diperoleh dan analisa.

Bab V Berisi penutup dan kesimpulan.

BAB II
IPv6 DAN MPLS

2.1 IPv6

Dalam jaringan komputer dikenal adanya suatu protokol yang mengatur bagaimana suatu node berkomunikasi dengan node lainnya didalam jaringan, protokol tersebut berfungsi sebagai bahasa agar satu komputer dapat berkomunikasi satu dengan yang lainnya. Protokol yang merupakan standar *de facto* dalam jaringan internet yaitu protokol TCP/IP, sehingga dengan adanya TCP/IP komputer yang dengan berbagai jenis *hardware* dan berbagai jenis sistem operasi (*Unix* maupun *Windows*) tetap dapat berkomunikasi.

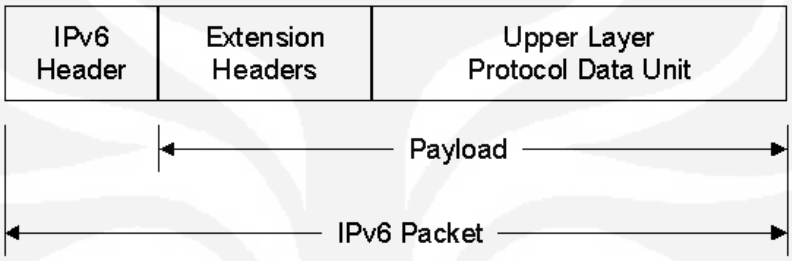
Setiap komputer yang terhubung ke internet setidaknya harus memiliki sebuah IP *address* pada setiap *interface*nya dan IP *address* sendiri harus unik karena tidak boleh ada komputer/*server*/perangkat *network* lainnya yang menggunakan IP *address* yang sama di internet. IP *address* adalah sederetan bilangan biner sepanjang 32 bit (IPv4), yang dipakai untuk mengidentifikasi *host* pada jaringan. IP *address* ini diberikan secara unik pada masing-masing komputer/*host* yang tersambung ke internet. Paket yang membawa data, dimuati IP *address* dari komputer pengirim data, dan IP *address* dari komputer yang dituju, kemudian data tersebut dikirim ke jaringan. Paket ini kemudian dikirim dari *router* ke *router* dengan berpedoman pada IP *address* tersebut, menuju ke komputer yang dituju. Seluruh *host*/komputer yang tersambung ke Internet, dibedakan hanya berdasarkan IP *address* ini, jadi jelaslah bahwa tidak boleh terjadi duplikasi, untuk itu IP *address* ini dibagikan oleh beberapa organisasi yang memiliki otoritas atas pembagian IP *address* tersebut, seperti APNIC (*Asia Pacific Network Information Center*).

Versi IP yang saat ini telah dipakai secara meluas di internet adalah *Internet Protocol version* 4 (IPv4). IPv4 memiliki panjang bit 32 bit, dengan jumlah *host* 2^{32} = 4294967296 alamat *host*. Jumlah ini pun pada kenyataannya tidak mencapai 4 miliar dikarenakan beberapa limitasi seperti adanya IP *network* dan IP *broadcast*. Perkembangan internet yang sangat pesat saat ini menyebabkan ketersediaan alokasi alamat IP versi 4 semakin berkurang. Untuk itulah IETF

(*Internet Engineering Task Force*) mendesain alamat IP baru yang disebut IPv6 (*Internet Protocol version* 6) yang memiliki panjang bit 128 bit dengan jumlah alamat *host* yang cukup besar yaitu $2^{128} = 3{,}4x10^{38}$ alamat *host*.

2.1.1 Struktur IPv6

Paket data IPv6 terdiri dari komponen-komponen seperti pada Gambar 2.1 di bawah:



Gambar 2.1 Struktur Paket Data IPv6 [5]

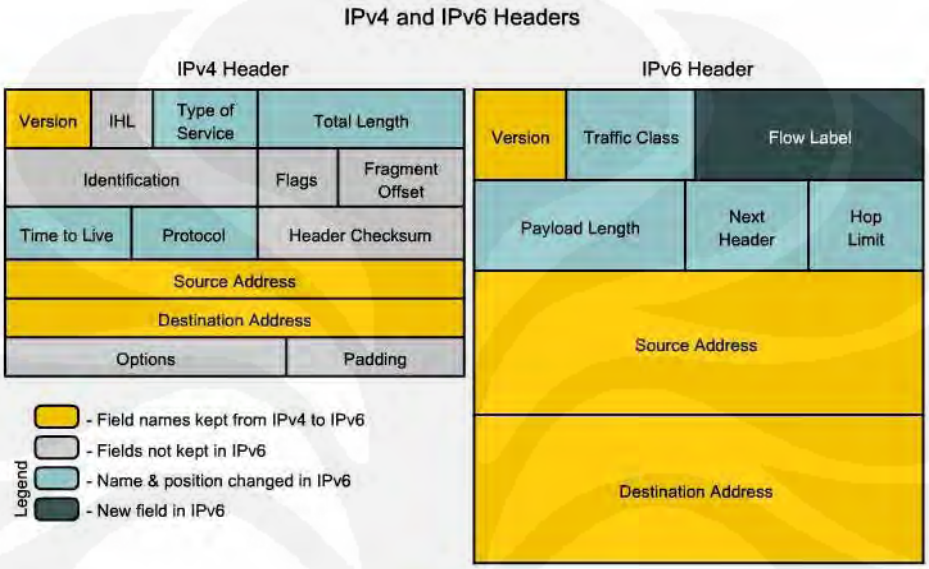
1. *Header* IPv6

Header IPv6 ini akan selalu ada dengan ukuran yang tetap yaitu 40 bytes. *Header* ini merupakan penyederhanaan dari *header* IPv4 dengan menghilangkan bagian yang tidak diperlukan atau jarang digunakan dan menambahkan bagian yang menyediakan dukungan yang lebih bagus untuk komunikasi masa depan yang sebagian besar adalah trafik *real-time*. Beberapa perbandingan kunci dari *header* IPv4 dan IPv6:

a. Jumlah *header field* berkurang dari 12 (termasuk *option*) pada *header* IPv4 menjadi 8 pada *header* IPv6.
b. Jumlah *header field* yang harus diproses oleh *router* antara (*intermediate router*) turun dari 6 menjadi 4 yang membuat proses *forwarding* paket IPv6 normal menjadi lebih efisien.
c. *Header field* yang jarang terpakai seperti *field*s *supporting fragmentation* dan *option* pada *header* IPv4 dipindahkan ke *extension header* IPv6.
d. Ukuran *header* IPv6 memang bertambah dua kalinya, yaitu dari 20 bytes pada *header* minimum IPv4 menjadi tetap sebesar 40 bytes. Namun keuntungannya

adalah *header* untuk pengalamatan menjadi 4 kali lebih panjang dari IPv4 (dari 32 menjadi 128 bit) yang menyebabkan tersedianya jumlah alamat yang jauh lebih besar.[5]

Perbandingan arsitektur header IPv4 dan IPv6 dapat dilihat pada Gambar 2.2 di bawah.



Gambar 2.2 Perbandingan Header IPv4 dengan Header IPv6 [4]

2. *Extension header*s

Header dan *extension header* pada IPv6 ini menggantikan *header* dan *option* pada IPv4. Tidak seperti *option*s pada IPv4, *extension header*s IPv6 tidak memiliki ukuran maksimum dan dapat diperluas untuk melayani kebutuhan komunikasi data di IPv6. Jika pada *header* IPv4 semua *option* akan dicek dan diproses jika ada maka pada *extension header*s IPv6 hanya ada satu yang harus diproses yaitu *Hop-by-Hop Options*. Hal ini akan meningkatkan kecepatan pemrosesan *header* IPv6 dan meningkatkan kinerja *forwarding* paket IPv6. *Extension header* yang harus didukung oleh setiap titik IPv6 yaitu :

- *Hop-by-Hop Options header*
- *Destination Options header*
- *Routing header*
- *Fragment header*
- *Authentication header*

- *Encapsulating Security Payload header*

3. *Protocol Data Unit* (PDU) dari *layer* yang lebih tinggi (upper *layer*)

Protocol Data Unit (PDU) *layer* yang lebih tinggi pada dasarnya terdiri dari *header* protokol *layer* yang lebih tinggi dan *payload* yang terkandung di dalamnya misalnya saja TCP, UDP atau ICMPv6. [5]

2.1.2 Pengalamatan IPv6

Dalam IPv6, alamat 128-bit akan dibagi ke dalam 8 blok berukuran 16-bit, yang dapat dikonversikan ke dalam bilangan heksadesimal berukuran 4-digit. Setiap blok bilangan heksadesimal tersebut akan dipisahkan dengan tanda titik dua (:). Karenanya, format notasi yang digunakan oleh IPv6 juga sering disebut dengan *colon-hexadecimal format*, berbeda dengan IPv4 yang menggunakan *dotted-decimal format*. [12]

Berikut ini adalah contoh alamat IPv6 dalam bentuk bilangan biner:
00100001110110100000000011010011000000000000000000101111001110110000001010101010000000001111111111111110001010001001110001011010

Untuk menterjemahkan ke dalam bentuk notasi *colon-hexadecimal format*, angka-angka biner di atas harus dibagi ke dalam 8 buah blok berukuran 16-bit:
0010000111011010 0000000011010011 0000000000000000 0010111100111011 0000001010101010 0000000011111111 1111111000101000 1001110001011010

Setiap blok berukuran 16-bit tersebut harus dikonversikan ke dalam bilangan heksadesimal dan setiap bilangan heksadesimal tersebut dipisahkan dengan menggunakan tanda titik dua. Hasil konversinya adalah sebagai berikut:
21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A

2.1.2.1 Penyederhanaan Bentuk Alamat

Alamat di atas juga dapat disederhanakan lagi dengan membuang angka 0 pada awal setiap blok yang berukuran 16-bit di atas, dengan menyisakan satu digit terakhir [12]. Dengan membuang angka 0, alamat di atas disederhanakan menjadi:

21DA:D3:0:2F3B:2AA:FF:FE28:9C5A

Konvensi pengalamatan IPv6 juga mengizinkan penyederhanaan alamat lebih jauh lagi, yakni dengan membuang banyak karakter 0, pada sebuah alamat yang banyak angka 0-nya. Jika sebuah alamat IPv6 yang direpresentasikan dalam notasi *colon-hexadecimal* format mengandung beberapa blok 16-bit dengan angka 0, maka alamat tersebut dapat disederhanakan dengan menggunakan tanda dua buah titik dua (**::**). Untuk menghindari kebingungan, penyederhanaan alamat IPv6 dengan cara ini sebaiknya hanya digunakan sekali saja di dalam satu alamat, karena kemungkinan nantinya pengguna tidak dapat menentukan berapa banyak bit 0 yang direpresentasikan oleh setiap tanda dua titik dua (**::**) yang terdapat dalam alamat tersebut. Tabel 2.1 berikut mengilustrasikan cara penggunaan hal ini.

Tabel 2.1 Contoh Penyederhanaan Alamat IPv6

Alamat asli	Alamat asli yang disederhanakan	Alamat setelah dikompres
FE80:0000:0000:0000:02AA:00FF:FE9A:4CA2	FE80:0:0:0:2AA:FF:FE9A:4CA2	FE80::2AA:FF:FE9A:4CA2
FF02:0000:0000:0000:0000:0000:0000:0002	FF02:0:0:0:0:0:0:2	FF02::2

Untuk menentukan berapa banyak bit bernilai 0 yang dibuang (dan digantikan dengan tanda dua titik dua) dalam sebuah alamat IPv6, dapat dilakukan dengan menghitung berapa banyak blok yang tersedia dalam alamat tersebut, yang kemudian dikurangkan dengan angka 8, dan angka tersebut dikalikan dengan 16. Sebagai contoh, alamat FF02::2 hanya mengandung dua blok alamat (blok FF02 dan blok 2). Maka, jumlah bit yang dibuang adalah (8-2) x 16 = 96 buah bit.

2.1.2.2 Format Prefix

Dalam IPv4, sebuah alamat dalam notasi *dotted-decimal format* dapat direpresentasikan dengan menggunakan angka *prefix* yang merujuk kepada *subnet mask*. IPv6 juga memiliki angka *prefix*, tapi tidak digunakan untuk merujuk kepada *subnet mask*, karena memang IPv6 tidak mendukung *subnet mask*.

Prefix adalah sebuah bagian dari alamat IP, di mana bit-bit memiliki nilai-nilai yang tetap atau bit-bit tersebut merupakan bagian dari sebuah rute atau *subnet identifier*. *Prefix* dalam IPv6 direpesentasikan dengan cara yang sama seperti halnya *prefix* alamat IPv4, yaitu [alamat]/[angka panjang *prefix*]. Panjang

prefix menentukan jumlah bit terbesar paling kiri yang membuat *prefix* subnet. Sebagai contoh, *prefix* sebuah alamat IPv6 dapat direpresentasikan sebagai berikut:

3FFE:2900:D005:F28B::/64

Pada contoh di atas, 64 bit pertama dari alamat tersebut dianggap sebagai *prefix* alamat, sementara 64 bit sisanya dianggap sebagai *interface* ID. [12]

2.1.2.3 Jenis-Jenis Alamat IPv6

Jika pada IPv4 dikenal adanya pengkelasan yaitu kelas A,B,C,D dan E, maka pada IPv6 tidak dikenal istilah pengkelasan, hanya IPv6 menyediakan 3 jenis pengalamatan yaitu: *Unicast, Anycast* dan *Multicast*.

Alamat *unicast* yaitu alamat yang menunjuk pada sebuah alamat *interface* atau *host,* digunakan untuk komunikasi satu lawan satu. Pada alamat *unicast* dibagi 3 jenis lagi yaitu: alamat link lokal, alamat site lokal dan alamat global. Alamat link lokal adalah alamat yang digunakan di dalam satu link yaitu jaringan lokal yang saling tersambung dalam satu level. Alamat site lokal setara dengan alamat privat pada IPv4, yang dipakai terbatas di dalam satu site sehingga terbatas penggunaannya hanya didalam satu site sehingga tidak dapat digunakan untuk mengirimkan alamat diluar site ini. Alamat global adalah alamat yang dipakai misalnya untuk *Internet Service Provider* (ISP).

Alamat *anycast* adalah alamat yang menunjukkan beberapa *interface* (biasanya node yang berbeda). Paket yang dikirimkan ke alamat ini akan dikirimkan ke salah satu alamat *interface* yang paling dekat dengan *router*. Alamat *anycast* tidak mempunyai alokasi khusus, karena jika beberapa node/*interface* diberikan *prefix* yang sama maka alamat tersebut sudah merupakan alamat *anycast.*

Alamat *multicast* adalah alamat yang menunjukkan beberapa *interface* (biasanya untuk node yang berbeda). Paket yang dikirimkan ke alamat ini akan dikirimkan ke semua *interface* yang ditunjukkan oleh alamat ini. Alamat *multicast* ini didesain untuk menggantikan alamat *broadcast* pada IPv4 yang banyak mengkonsumsi bandwidth. [4]

2.1.3 Perbandingan IPv4 dengan IPv6

Pada tabel 2.2 di bawah akan disebutkan beberapa perbedaan antara IPv4 dengan IPv6.

Tabel 2.2 Perbandingan IPv4 dengan IPv6 [5]

IPv4	IPv6
Panjang alamat 32 bit (4 bytes)	Panjang alamat 128 bit (16 bytes)
Dikonfigurasi secara manual atau DHCP IPv4	Tidak arus dikonfigurasi secara manual, bisa menggunakan *address autoconfiguration*
Dukungan terhadap IPSec opsional	Dukungan terhadap IPSec dibutuhkan
Fragmentasi dilakukan oleh pengirim dan pada *router*, menurunkan kinerja *router*	Fragmentasi dilakukan hanya oleh pengirim
Tidak mensyaratkan ukuran paket pada *link-layer* dan harus bisa menyusun kembali paket berukuran 576 byte	Paket link-*layer* harus mendukung ukuran paket 1280 byte dan harus bisa menyusun kembali paket berukuran 1500 byte
Checksum termasuk pada *header*	*Cheksum* tidak masuk dalam *header*
Header mengandung *option*	Data opsional dimasukkan seluruhnya ke dalam *extensions header*
Menggunakan ARP *Request* secara *broadcast* untuk menterjemahkan alamat IPv4 ke alamat *link-layer*	ARP *Request* telah digantikan oleh *Neighbor Solicitation* secara *multicast*
Untuk mengelola keanggotaan grup pada subnet lokal digunakan *Internet Group Management Protocol* (IGMP)	IGMP telah digantikan fungsinya oleh *Multicast Listener Discovery* (MLD)

2.2 Multi Ptotocol Label Switching (MPLS)

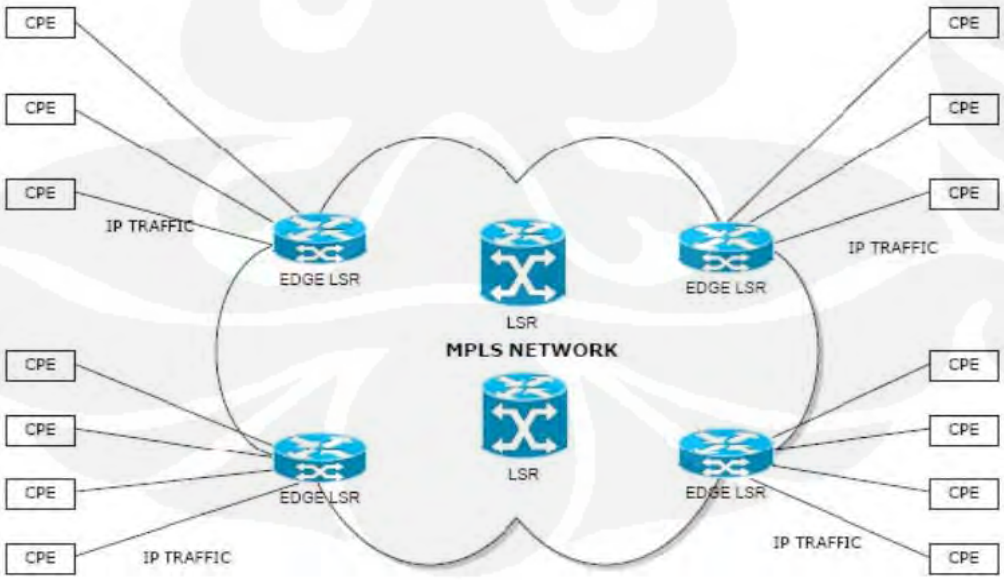
Teknologi ATM dan *frame-relay* bersifat *connection-oriented,* yaitu setiap *virtual circuit* harus di-*setup* dengan protokol persinyalan sebelum transmisi. IP

bersifat *connectionless*, di mana protokol *routing* menentukan arah pengiriman paket dengan bertukar info *routing*. MPLS mewakili konvergensi kedua pendekatan ini.

MPLS (*Multi Protocol Label Switching*) adalah arsitektur *network* yang didefinisikan oleh IETF untuk memadukan mekanisme label *swapping* di *layer* 2 dengan *routing* di *layer* 3 untuk mempercepat pengiriman paket. Arsitektur MPLS dipaparkan dalam RFC-3031.

Keuntungan lain adalah tidak diperlukannya kerumitan teknis seperti enkapsulasi ke dalam AAL dan pembentukan sel-sel ATM, yang masing-masing menambah *delay*, menambah *header*, dan memperbesar kebutuhan *bandwidt*h. MPLS tidak memerlukan hal-hal itu.

Network MPLS terdiri atas sirkit yang disebut *label-switched path* (LSP), yang menghubungkan titik-titik yang disebut *label-switched router* (LSR). LSR pertama dan terakhir disebut *ingress* dan *egress*. Setiap LSP dikaitkan dengan sebuah *forwarding equivalence class* (FEC), yang merupakan kumpulan paket yang menerima perlakukan *forwarding* yang sama di sebuah LSR. FEC diidentifikasikan dengan pemasangan label. Gambar jaringan MPLS dapat dilihat pada Gambar 2.3 di bawah.



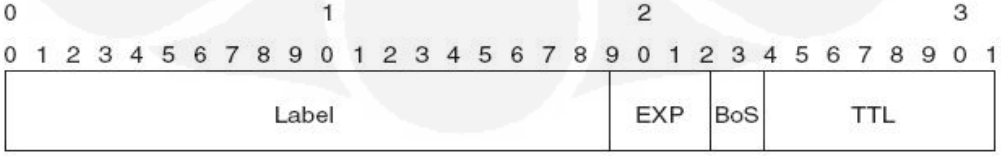
Gambar 2.3 Jaringan MPLS [6]

Untuk membentuk LSP, diperlukan suatu protokol persinyalan. Protokol ini menentukan *forwarding* berdasarkan label pada paket. Label yang pendek dan berukuran tetap mempercepat proses *forwarding* dan mempertinggi fleksibilitas pemilihan *path*. Hasilnya adalah *network* datagram yang bersifat lebih *connection-oriented*.

2.2.1 Arsitektur Jaringan MPLS

Multi Protocol Label Switching (MPLS) merupakan sebuah teknik yang menggabungkan kemampuan manajemen *switching* yang ada dalam teknologi ATM dengan fleksibilitas *network layer* yang dimiliki teknologi IP. Konsep utama MPLS adalah teknik peletakan "label" dalam setiap paket yang dikirim melalui jaringan ini. MPLS bekerja dengan cara memberi label paket-paket data, untuk menentukan rute dan prioritas paket tersebut. Label tersebut akan memuat informasi penting yang berhubungan dengan informasi *routing* suatu paket, di antaranya berisi tujuan paket serta prioritas paket mana yang harus dikirimkan terlebih dahulu. Teknik ini biasa disebut dengan *label switching*. Dengan informasi label *switching* yang didapat dari *routing network layer*, setiap paket hanya dianalisa sekali di dalam *router* di mana paket tersebut masuk ke dalam jaringan untuk pertama kali, *router* tersebut berada di tepi dan dalam jaringan MPLS yang biasa disebut dengan *Label Swicthing Router* (LSR). Ide dasar teknik MPLS ini adalah mengurangi teknik pencarian rute dalam setiap *router* yang dilewati setiap paket, sehingga dapat dioperasikan dengan efisien dan jalannya pengiriman paket menjadi lebih cepat.

Sebuah MPLS label terdiri dari 32 bit dengan struktur seperti pada Gambar 2.4, pada 20 bit pertama adalah bit label. Bit 20 sampai 22 adalah tiga *experimental bit* (EXP). Bit-bit ini hanya digunakan untuk *Quality of Service* (QoS).



Gambar 2.4 Label MPLS [1]

Bit 23 merupakan bit *Bottom of Stack* (BoS) yang bernilai 0, kecuali jika label ini berada dalam *stack* maka BoS diset 1. *Stack* adalah sekumpulan label yang yang dipertemukan pada ujung paket. *Stack* terdiri dari satu label atau lebih. Jumlah label yang ditemui pada *stack* tak terbatas, meskipun jarang ditemui sebuah *stack* yang terdiri dari 4 atau lebih label.

Bit 24 sampai 31 adalah 8 bit yang digunakan sebagai *Time To Live* (TTL). TTL ini memiliki fungsi yang sama dengan TTL pada IP *header*. Fungsinya adalah menghindari agar paket tidak berhenti pada *routing loop*. Jika *routing loop* terjadi dan tidak ada TTL maka *looping* tidak akan berhenti. TTL ini berkurang 1 tiap hop dan jika TTL label mencapai 0, maka paket akan dibuang. [1]

2.2.2 Enkapsulasi Paket

Tidak seperti ATM yang memecah paket-paket IP, MPLS hanya melakukan enkapsulasi paket IP, dengan memasang *header* MPLS. *Header* MPLS terdiri atas 32 bit data, termasuk 20 bit label, 2 bit eksperiman, dan 1 bit identifikasi *stack*, serta 8 bit TTL. Label adalah bagian dari *header*, memiliki panjang yang bersifat tetap, dan merupakan satu-satunya tanda indentifikasi paket. Label digunakan untuk proses *forwarding*, termasuk proses *traffic engineering*.

Setiap LSR memiliki tabel yang disebut *label-switching table*. Tabel itu berisi pemetaan label masuk, label keluar, dan link ke LSR berikutnya. Saat LSR menerima paket, label pasti akan dibaca, kemudian diganti dengan label keluar, lalu paket dikirimkan ke LSR berikutnya.

Selain paket IP, paket MPLS juga bisa dienkapsulasi kembali dalam paket MPLS sehingga sebuah paket bisa memiliki beberapa *heade*r. Bit *stack* pada *header* menunjukkan apakah suatu *heade*r sudah terletak di dasar tumpukan *header* MPLS itu.[6]

2.2.3 Label Switch Router

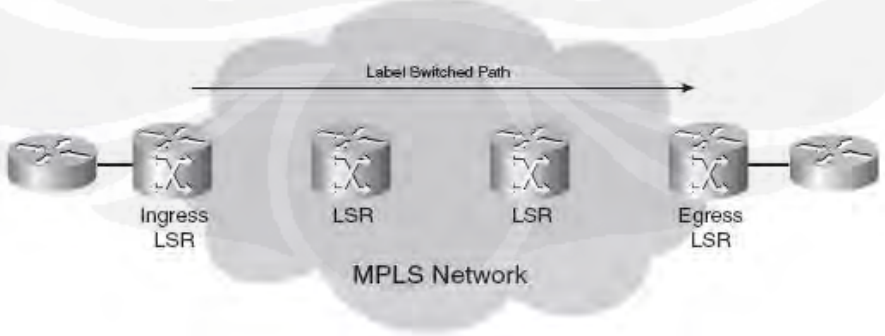
Label Switch Router (LSR) adalah *router* yang mendukung MPLS, mampu membaca label MPLS, menerima dan mengirimkan paket yang telah diberi label pada data link. Ada 3 LSR dalam jaringan MPLS

1. *Ingress LSR*, menerima paket yang belum diberi label, kemudian memberi label (melabeli) di depan paket dan mengirimnya ke data link.
2. *Intermediate* LSR, menerima paket yang datang, membaca label yang ada, menggantinya dengan label baru kemudian mengirimnya ke data link yang tepat sesuai label yang terbaca.
3. *Egress LSR*, menerima paket yang sudah diberi label, membuang labelnya kemudian mengirimnya ke data link. *Ingress* dan *Egress* LSR termasuk dalam *Edge LSR* (ELSR).[1]

LSR melakukan tiga operasi, yaitu POP, PUSH, dan SWAP. Jika paket yang datang adalah paket yang belum diberi label maka LSR akan melakukan operasi PUSH yang meletakkan label pada paket, hal ini disebut *emposing* LSR dan dilakukan oleh *Ingress* LSR. Sedangkan jika paket yang datang sudah diberi label, LSR akan melakukan operasi POP yang akan menghapus label pada paket (*disposing*), dan dilakukan oleh *Egress* LSR. Suatu LSR juga mampu melakukan SWAP, maksudnya ketika ada paket berlabel yang datang, top label akan diganti dengan label yang baru untuk kemudian dikirim ke data link.

2.2.4 Label Switch Path (LSP)

Suatu LSP merupakan rangkaian LSR yang menghubungkan paket yang sudah dilabeli melalui jaringan MPLS atau bagian dari jaringan MPLS. Pada dasarnya LSP adalah *path*/jalur yang dilalui jaringan MPLS atau bagian dari pengambilan paket. LSR pertama dari LSP adalah *ingress* LSR untuk LSP tersebut, sedangkan LSR terakhir dari LSP adalah *egress* LSR. Semua LSR yang terletak di antara *ingress* dan *egress* LSR adalah *intermediate* LSR.



Gambar 2.5 LSP pada Jaringan MPLS[1]

Pada Gambar 2.5 di atas, panah di atas menunjukkan arah LSP, karena LSP bersifat *unidirectional* (satu arah), di luar ELSR merupakan LSP yang lain.

2.2.5 Forwarding Equivalence Class (FEC)

Forwarding Equivalence Class (FEC) adalah sekumpulan aliran paket yang diteruskan dalam *path* yang sama. Semua paket pada FEC yang sama memiliki label yang sama, tetapi tidak semua paket yang mempunyai label yang sama berada dalam FEC yang sama, karena nilai EXP (*experimental bit*) bisa saja berbeda. *Router* yang akan menentukan paket mana berada pada FEC mana adalah *Ingress* LSR. sebab memang tugas *Ingress* LSR mengklasifikasi dan memberi label pada paket.[1]

2.2.6 Label Distribution

Label pertama yang diletakkan pada *ingress* LSR akan menjadi milik sebuah LSP. Jalur paket yang melalui jaringan MPLS dibatasi pada LSP tersebut. Satu-satunya yang berubah adalah label atas pada tumpukan label yang diganti/ditukar pada tiap-tiap hop. *Ingress* LSR mengisi satu atau lebih label pada sebuah paket, *Intermediate* LSR mengganti label atas (label yang datang) dari paket yang diterima dengan label lain dan kemudian mengirimkan ke *outgoing* link. *Egress* LSR dari LSP menghilangkan label ini dan meneruskan paket.

Contohnya adalah IPv4 pada MPLS yang merupakan contoh paling sederhana dari jaringan MPLS. IPv4 dalam MPLS terdiri dari LSR yang menjalankan IPv4 *Interior Gateway protocol* (IGP), misalnya OSPF,IS-IS, dan *Enhanced Interior Gateway Routing Protocol* (EIGRP). *Ingress* LSR akan mencari alamat tujuan dari IPv4, memberi label, dan meneruskan paket. LSR selanjutnya (*intermediate* LSR) menerima paket, menggantinya *incoming* label dengan *outgoing* label, dan meneruskan paket. *Egress* LSR mengambil label dan meneruskan paket IPv4 tanpa label pada *outgoing link*. Pada proses ini, setiap LSR harus sepakat mengenai label mana yang digunakan untuk tiap IGP *prefix*. Selain itu, masing-masing *Intermediate* LSR harus mampu menentukan dengan benar *outgoing* label mana yang menggantikan *incoming* label. Hal ini berarti

dibutuhkan mekanisme yang mengatakan pada *router* label mana yang harus digunakan ketika meneruskan paket.

Label hanya dikenal dengan sifat lokal oleh masing-masing *adjacent router* yang berpasangan. Label tidak bersifat global sepanjang jaringan. *Adjacent* r*outer* membutuhkan kesepakatan label mana yang akan digunakan untuk *prefix* yang mana, mereka membutuhkan komunikasi antara mereka, jika tidak, *router* tidak akan tahu *outgoing* label yang mana yang harus dicocokkan dengan *incoming* label. Untuk itulah suatu *Label Distribution Protocol* (LDP) dibutuhkan.

2.2.7 Distribusi Label dengan LDP

Label Distribution Protocol (LDP) merupakan suatu proses pemetaan dari setiap label masukan ke setiap label keluaran pada setiap LSR. Dalam arsitektur jaringan MPLS, sebuah LSR yang merupakan tujuan atau hop selanjutnya akan mengirimkan informasi tentang ikatan sebuah label ke LSR yang sebelumnya mengirimkan pesan untuk mengikat label tersebut bagi rute paketnya. Teknik ini biasa disebut *distribution label downstream on demand*.

Dalam melakukan pemetaan label, LDP melakukan operasi sebagai berikut:

1. *Discovery Message*, mengetahui keberadaan *adjacent* LSR dengan "*hello packet*".
2. *Adjacency Message*, untuk menjaga sesi LDP dengan *keepalive*.
3. *Label Advertisement Message*:
    a. *Label Mapping*, mengumumkan *mapping* FEC
    b. *Label Withdrawal*, mengumumkan pelepasan *mapping* FEC
    c. *Label Release*, pemberitahuan dari LSR yang menerima informasi label bahwa label sudah tidak digunakan.
4. *Notification Message*, pemberitahuan *error* dan *advisory*.[2]

Dari semua ikatan *remote* untuk satu *prefix*, LSR hanya mengambil satu ikatan remote untuk menentukan label *outgoing* untuk IP *prefix* tersebut. Tabel *routing* (biasa disebut *Routing Instace Base*/RIB) menentukan hop selanjutnya dari *prefix* IPv4. LSR memilih ikatan *remote* yang diterima dari *downstream* LSR, yang mana merupakan hop selanjutnya dari table *routing prefix* tersebut. Hal ini

digunakan sebagai informasi untuk mengeset label *information forwarding base* (LFIB) di mana label dari ikatan lokal berfungsi sebagai label *incoming* dan label dari ikatan remote dipilih melalui tabel *routing* dan berfungsi sebagai label *outgoing*. Oleh karena itu, ketika LSR menerima paket berlabel, LSR dapat mengganti label *incoming* dengan label *outgoing* yang diberikan oleh LSR *adjacent* di hop selanjutnya

LDP hanya memiliki *feature* dasar dalam melakukan *forwarding*. Untuk meningkatkan kemampuan mengelola QoS dan rekayasa trafik, beberapa protokol distribusi label lain telah dirancang dan dikembangkan juga. Yang paling banyak disarankan adalah *constrain-based routing* (CR-LDP) dan RSVP-TE (RSVP dengan ekstensi *Traffic Engineering*).

2.2.8 Label Forwarding Instance Base (LFIB)

LFIB adalah tabel yang digunakan untuk meneruskan paket berlabel. Di dalamnya berisi label *incoming* dan label *outgoing* untuk LSP. Semua *outgoing* yang berasal dari ikatan remote dapat ditemukan pada *Label Information Base* (LIB). LFIB akan memilih satu label *outgoing* dan memasangnya pada LFIB. Pemilihannya berdasarkan pada bagian mana yang merupakan jalur terbaik pada tabel *routing*.

Kunci pengambilan keputusan suatu paket oleh *router* ditentukan oleh semua sumber informasi yang dapat dikerjakan oleh sebuah label *switching* dengan melihat nilai suatu label yang panjangnya tertentu. Tabel ini biasa disebut *Label Forwarding Information Base* (LFIB). Sebuah label yang akan digunakan sebagai sebuah indeks suatu node yang akan digunakan untuk memutuskan tujuan selanjutnya, dengan pergantian label di dalam node tersebut. Label lama digantikan oleh label baru dan paket akan dikirimkan ke tujuan selanjutnya.

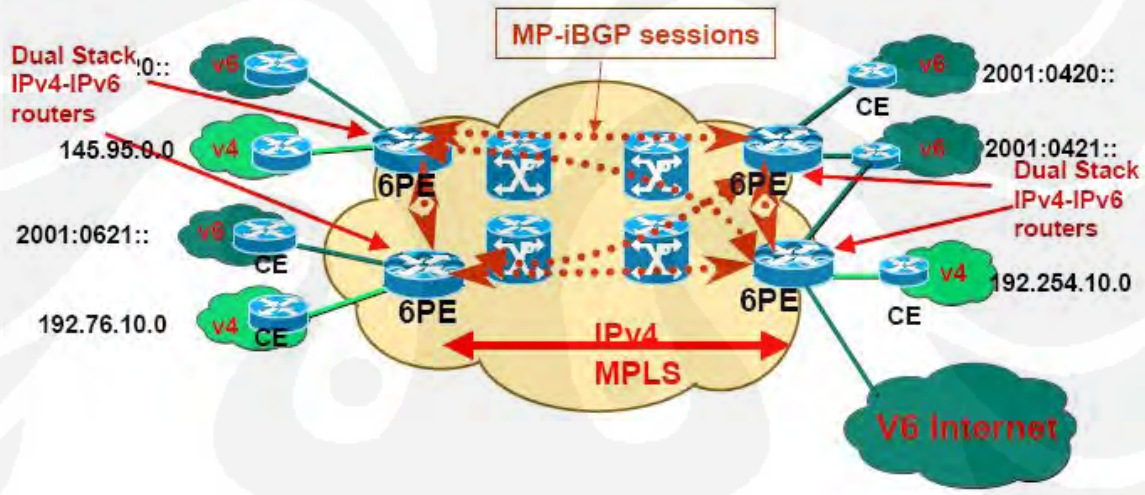
Sebuah *label switching* akan membuat pekerjaan *router* dan switch menjadi lebih mudah dalam menentukan pengiriman suatu paket. MPLS ini akan memperlakukan switch-switch sebagai suatu peer-peer, dan mengontrol *feature* yang secara normal hanya dapat berjalan di jaringan ATM. Dalam jaringan MPLS sekali suatu paket telah diberi label maka tidak perlu lagi terdapat analisa *header*

yang dilakukan oleh *router*, karena semua pengiriman paket telah dikendalikan oleh label yang ditambahkan tersebut.

2.3 IPv6 pada Jaringan MPLS

Ada beberapa metode dalam pendekatan IPv6 pada jaringan MPLS:

1. IPv4 CE-to-CE Tunnel, pada metode ini *router* CE dikonfigur sebagai *dual stack* (IPv4 dan IPv6).
2. IPv6 *over* "*Circuit over* MPLS", di sini r*outer* PE harus diupgrade dengan AToM (Any Transport *over* MPLS), seperti ATM, Ethernet, dan lain-lain.
3. Native IPv6 MPLS, pada jaringan ini r*outing* IPv6 dikontrol oleh *router core* (*router* P)
4. IPv6 *Provider Edge Router* (6PE) *over* MPLS [9]



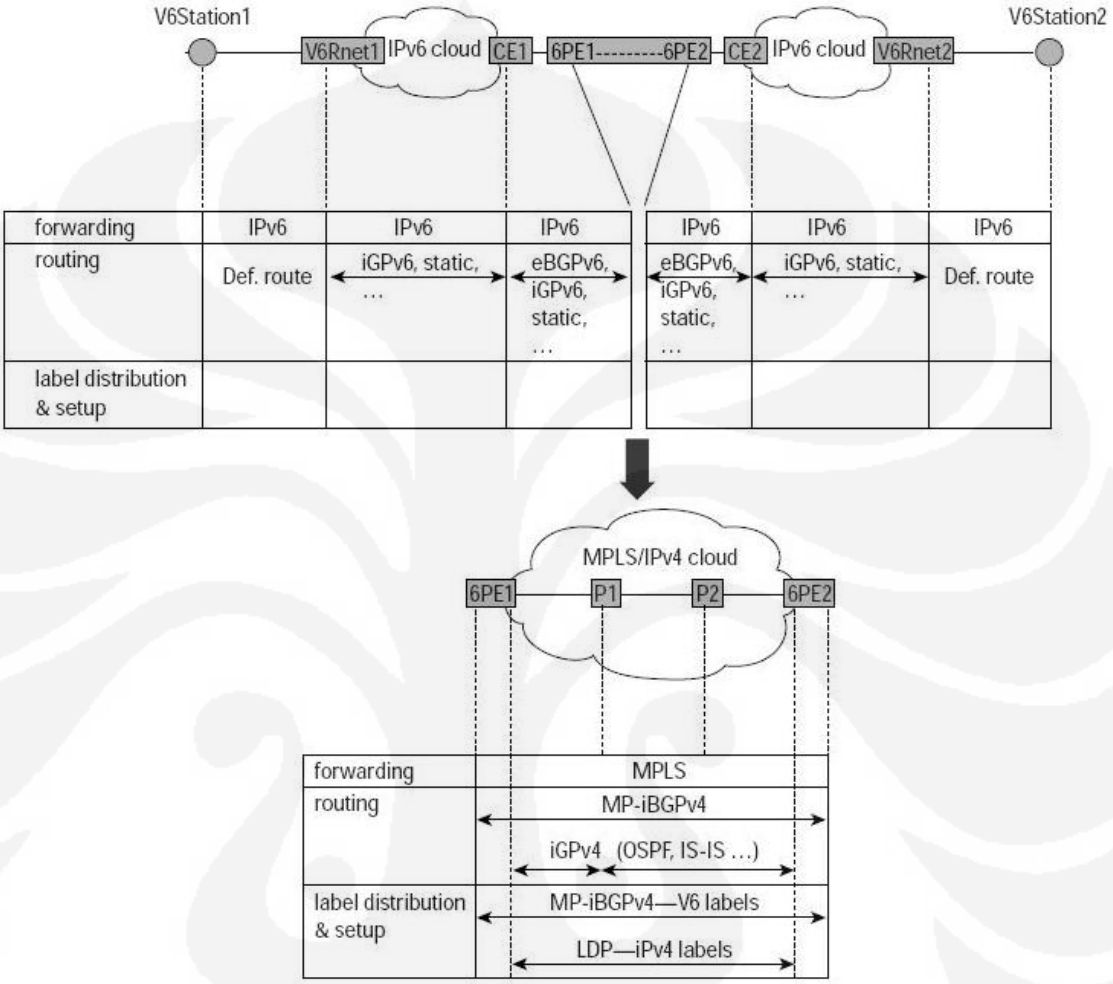
Gambar 2.6 IPv6 Provider Edge Router (6PE)

Gambar 2.6 di atas merupakan proses di dalam jaringan 6PE. MPLS IPv4 tidak mengetahui informasi IPv6, *routing* dilakukan oleh PE yang support *dual stack* (IPv4 dan IPv6). Pada skripsi ini akan dibahas mengenai 6PE.

2.3.1 Arsitektur 6PE

Pada Gambar 2.7 di bawah, MP-BGP *router* 6PE adalah *dual-stack* (IPv6 melalui *router* CE dan IPv4 melalui *core* MPLS). MP-iBGP digunakan antar 6PE untuk saling menukar informasi IPv6. Sebuah label diberikan pada masing-masing *prefix* IPv6 tujuan. Label ini merupakan kumpulan label IPv6 yang dialokasikan

oleh *egress* PE. Melalui MP-BGP *ingress router* 6PE akan mengetahui alamat IPv4 dari *egress* IPv6 lawan untuk mencapai *prefix* IPv6 yang dituju.



Gambar 2.7 Arsitektur 6PE [9]

Secara berulang, *ingress* 6PE mengekstrak alamat IPv4 yang terdapat pada IPv4 yang dipetakan pada alamat IPv6. Kemudian *ingress* 6PE memisahkan alamat IPv4 ini (menggunakan IPv4 *routing* table) sehingga mendapatkan label untuk LSP tujuan. Label IPv4 ini telah disimpan dalam tabel IPv4 melalui MPLS IPv4 menggunakan IPv4 IGP dan IPv4 LDP. Label IPv4 ini kemudian disimpan bersama dengan label BGP untuk IPv6 tujuan dalam IPv6 *forwarding* table pada *ingress router* PE.

2.3.2 Tugas 6PE

1. Berperan dalam IPv4 IGP untuk melakukan hubungan internal dalam *cloud* MPLS
2. Berperan dalam LDP atau TDP untuk melakukan pengikatan label IPv4
3. Menjalankan MP-iBGP untuk meng-*advertise* hubungan IPv6 dan mendistribusikan kumpulan label IPv6 antar mereka.
4. Menjalankan MP-eBGP, IPv6 IGP, atau *routing* statik dengan *router* CE untuk meng-*advertise* hubungan IPv6 dalam *cloud* MPLS.[9]

BAB III
PERANCANGAN SISTEM

3.1 Prinsip Kerja Sistem

Pada skripsi ini akan dirancang dan dijalankan 2 konfigurasi jaringan yaitu MPLS dengan IPv4 dan MPLS dengan IPv6 pada sisi PE (6PE), yang masing-masing terdiri dari 1 *router* P yang terhubung ke 2 *router* PE (PE1 dan PE2). Pada masing-masing jaringan dihubungkan dengan 2 buah laptop yang berfungsi sebagai CE dan bertindak sebagai *server* dan *client*. Pada masing-masing jaringan akan dilakukan pengujian aplikasi *File Transfer Protocol* (FTP) yang mana *client* melakukan *download* 5 buah file dengan ukuran yang berbeda-beda (16MB, 32MB, 64MB, 128MB, 256MB).

3.2 Topologi Jaringan

Pada kegiatan ini dilakukan *test bed* terhadap 3 macam jaringan, yaitu IPv4 tanpa MPLS, IPv4 dengan MPLS dan IPv6 dengan MPLS. Jaringan *test bed* yang digunakan merupakan simulasi jaringan yang terdiri dari 3 buah *router* dan 2 buah laptop dengan konfigurasi *router* yang berbeda-beda. Spesifikasi perangkat keras yang digunakan adalah sebagai berikut:

1. *Server*

   Processor : Intel Core 2 Duo
   RAM : 1Gbyte
   NIC : Ethernet 10/100 Mbps
   OS : Windows XP Service Pack 2

2. *Client*

   Processor : Intel Dual Core CPU 1.86
   RAM : 1Gbyte
   NIC : Ethernet 10/100 Mbps
   OS : Windows XP Service Pack 2

3. *Router* (3 buah)

   Cisco 2611

IOS 12.3 (24a)

4. Kabel Ethernet *cross* sepanjang 1-2 meter.

Sedangkan perangkat lunak (*software*) yang digunakan antara lain:

1. Wireshark,

   Wireshark merupakan perangkat lunak pengembangan dari Ethereal yang digunakan untuk mengamati paket-paket yang melalui suatu *interface*. Wireshark dapat dijalankan baik pada sistem operasi Windows maupun Linux.

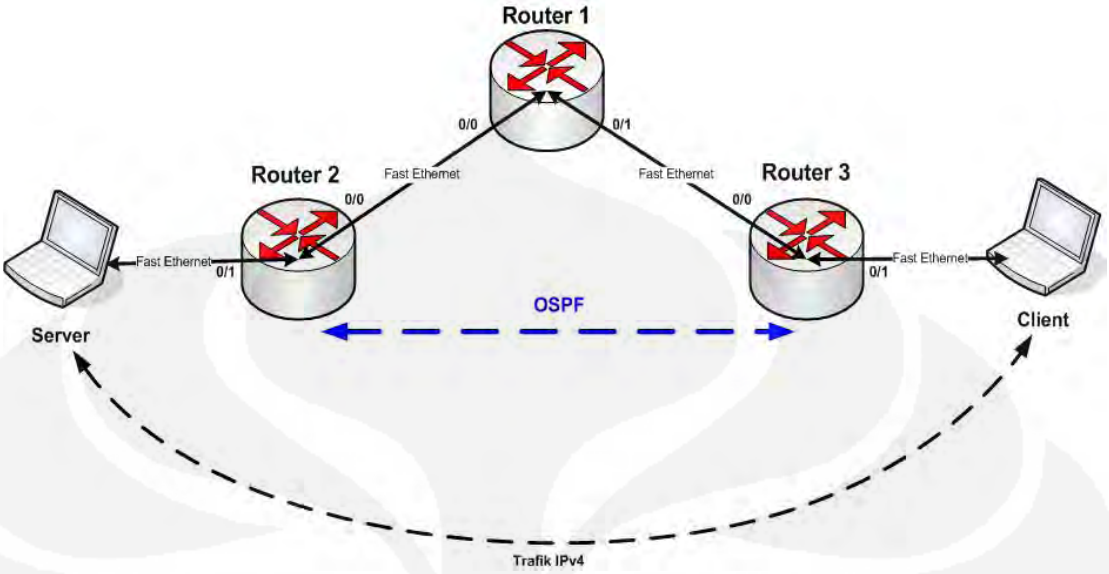
2. Xlight FTP

   Xlight merupakan perangkat lunak yang berfungsi sebagai *server* FTP. *Server* FTP digunakan untuk menyimpan file-file yang akan di*download* oleh FTP *client* maupun menampung file-file yang di*upload* oleh FTP *client*. Sebagai aplikasi yang digunakan untuk FTP *server*, dan Smart FTP *Client* sebagai FTP *client*. Kedua *software* FTP ini dapat digunakan untuk aplikasi FTP pada jaringan IPv4 maupun IPv6.

3. Smart FTP

   Smart FTP merupakan perangkat lunak yang berfungsi sebagai FTP *client*. FTP *client* digunakan oleh user untuk men*download* atau meng*upload* file-file dari dan ke FTP *server*. Smart FTP berjalan pada sistem operasi Windows dan telah mendukung IPv6.

3.2.1 Jaringan IPv4 tanpa MPLS



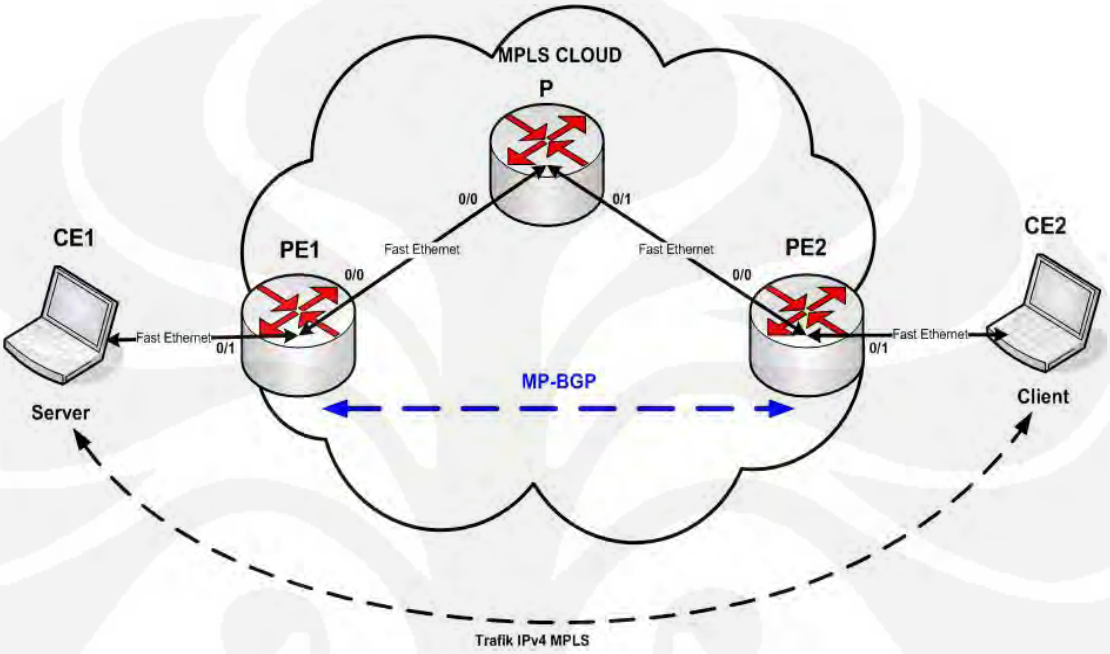
Gambar 3.1 Konfigurasi Jaringan OSPF

Pada Gambar 3.1 di atas merupakan jaringan tanpa MPLS yang menggunakan protokol OSPF. OSPF merupakan salah satu *routing* dinamik yang termasuk dalam kelompok *Interior Gateway Protocol* (IGP) jenis *link-state routing protocol*. Pada konfigurasi ini, keseluruhan jaringan menggunakan alamat IPv4 yang pemberian pada tiap *interface*nya dilakukan secara static atau manual. Alokasi alamat IPv4 yang diberikan adalah IP privat kelas C yaitu 172.16.1.0/24 dengan masing-masing *interface* diberikan IP point-to-point atau subnet /30 (255.255.255.252). IP loopback untuk masing-masing *router* adalah 10.0.1.x/32. Keseluruhan jaringan akan dibinding dalam *router* ospf.

Pemilihan protokol OSPF didasarkan pada alasan beberapa kelebihan OSPF yaitu bahwa protokol ini merupakan bagian dari IGP yang bersifat *link-state* yang memiliki kemampuan menyimpan seluruh topologi jaringan secara lengkap, kemudian masing-masing node akan mengkalkulasi hop terpendek yang akan dilalui untuk sampai ke tujuan. Berbeda dengan *distance vector routing* (seperti RIP, IGRP, EIGRP) yang membagi seluruh informasi *routing table*nya pada *router neighbor*nya. Pada protokol *link state* informasi yang dibagi hanya informasi untuk membentuk *map connectivity* antar *router* dalam suatu jaringan. Hal ini akan bermanfaat untuk jaringan yang luas dengan banyak node, selain itu

jika diterapkan dalam MPLS akan mempermudah ketika membentuk *Traffic Engineering* jaringan.

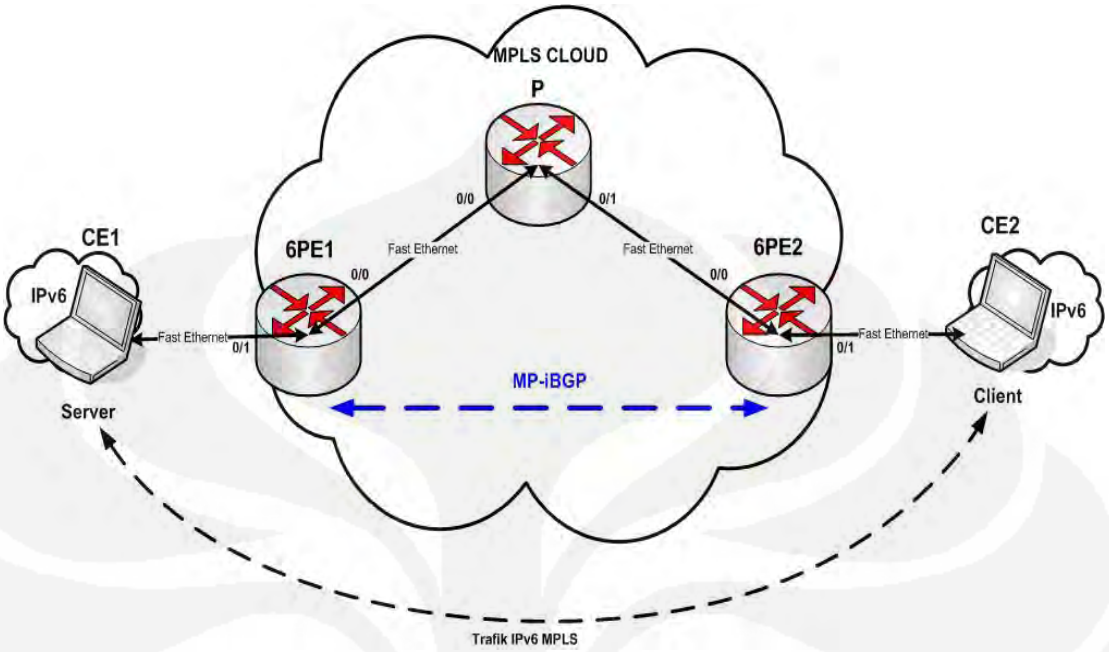
3.2.2 Jaringan MPLS IPv4



Gambar 3.2 Konfigurasi MPLS IPv4

Pada gambar 3.2 di atas merupakan konfigurasi jaringan MPLS IPv4 dengan 1 buah 2 *router* P dan 2 *router* PE. Konfigurasi *router* baik alokasi IPv4 maupun protokol koneksi IGPnya sama dengan jaringan IPv4 tanpa MPLS, hanya di sini pada masing-masing *interface router* ditambahkan komponen utama MPLS yaitu "*tag-switching ip*". Masing-masing *router* PE terhubung dengan laptop yang bertindak sebagai CE.

3.3.3 Jaringan MPLS IPv6



Gambar 3.3 Konfigurasi IPv6 over MPLS (6PE)

Pada gambar 3.3 di atas, jaringan pada *cloud* MPLS menggunakan IPv4 (*router* P dan *router* PE yang terhubung ke P), sedangkan pada *interface router* 6PE yang terhubung ke CE merupakan *dual stack* (IPv4 dan IPv6) yang dapat mengakomodir IPv4 dan IPv6. Pada kegiatan ini, pengetesan untuk IPv4 dan IPv6 dilakukan bergantian karena keterbatasan jumlah *interface router*. Di sini yang dujicobakan adalah koneksi PE dan CE yang menggunakan IPv6, untuk IPv4 sudah dujicobakan di kegiatan sebelumnya seperti Gambar 3.3. Konfigurasi jaringan pada *cloud* MPLS sama dengan konfigurasi pada MPLS IPv4, hanya di sini *interface router* 6PE yang terhubung ke CE diberikan alamat IPv6. *Prefix* IPv6 yang diberikan adalah 2001:DB8::/48, 2001:DB8:FFFF::/64 untuk 6PE1 dan 2001:DB8:DDDD::/64 untuk 6PE2. Pemberian alamat pada laptop sebagai CE bersifat *dynamic*, yaitu secara otomatis mendapatkan alokasi alamat IPv6 dari *router* yang terhubung pada *interface* ethernetnya. Koneksi antar *router* 6PE menggunakan protocol iBGP.

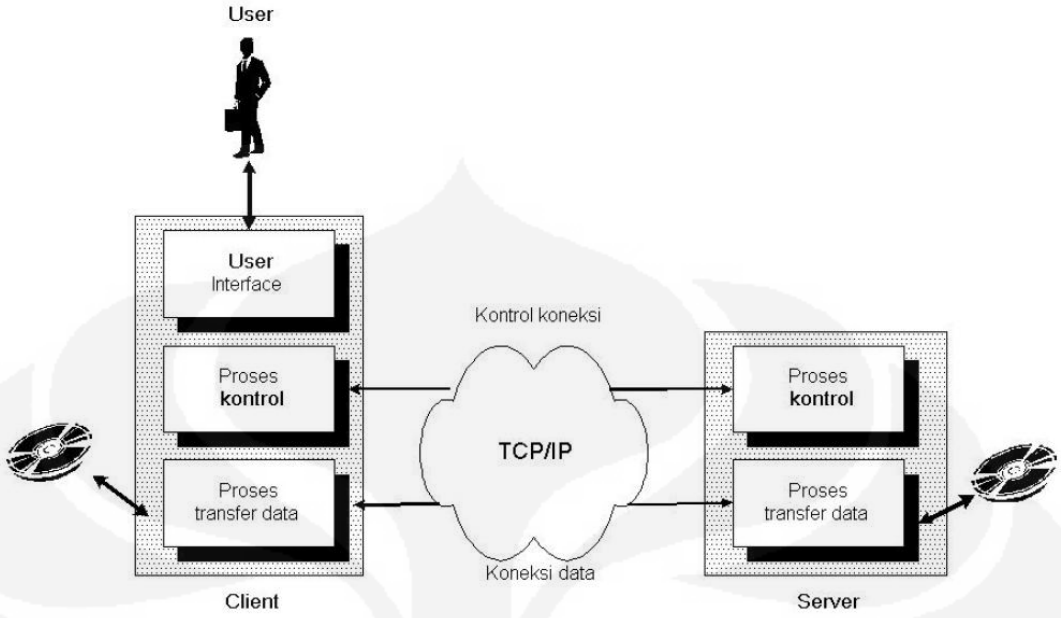
3.3 Metode Pengambilan Data

3.3.1 Pengujian Performa Jaringan untuk Aplikasi FTP

FTP (*File Transfer Protocol)* merupakan mekanisme standar yang dimiliki Protokol TCP/IP untuk keperluan penyalinan (*copying*) file dari satu *host* ke *host* yang lain. Operasi protocol FTP ini cukup sederhana. Dengan menggunakan *client* FTP, seorang pengguna dapat melihat isi direktori, memindahkan file dari dan ke *server* FTP serta membuat dan menghapus direktori di *server* tersebut, kemampuan *client* ini tergantung *permission* yang disetting pada *server*. Dalam melakukan operasi

yang berhubungan dengan pengiriman isi file, FTP menggunakan koneksi TCP tambahan yang khusus untuk mengirim file.

Dalam prosesnya sendiri, FTP memanfaatkan 2 port TCP/IP yaitu port 21 untuk *control connection* dan port 20 untuk *data connection*. *Control Connection* digunakan pada pola hubungan antara *client – server* normal. *Server* membuka diri secara pasif di sebuah port khusus selanjutnya *server* menunggu hubungan yang akan dilakukan oleh *client*. *Client* segera aktif membuka port tersebut untuk membangun *control connection*. *Control connection* ini akan dipertahankan sepanjang waktu selama *client* masih berkomunikasi dengan *server*. Hubungan ini digunakan oleh *client* untuk mengirim perintah-perintah ke *server*, dan *server* menggunakannya untuk memberi respon. Sedangkan d*ata connection* dibangun setiap kali file ditransfer antara *client server*. Hubungan ini bertujuan memaksimalkan ukuran data yang ditransfer (*throughput*), karena hubungan ini untuk transfer file.dibangun setiap kali sebuah file ditransfer. Pada Gambar 3.4 di bawah merupakan proses FTP.



Gambar 3.4 Proses FTP

Pada kegiatan uji coba aplikasi FTP ini, *software* yang digunakan adalah Xlight FTP (*server*) dan Smart FTP (*client*). *Client* akan melakukan *download* ke *server* suatu *file* yang berukuran 16MB, 32MB, 64MB,128 MB dan 256MB. Ukuran file yang bervariasi bertujuan untuk melihat ada atau tidaknya korelasi antara ukuran file dengan parameter-parameter yang diamati yaitu delay, throughput, dan transfer time. Selama FTP berlangsung, pada jaringan diberikan beban trafik sebesar 64000 MB yang diberikan dengan cara ping kontinyu dari *server* ke IP *client*. Trafik FTP akan ditangkap dengan *software* Wireshark yang diinstal pada laptop *client*. Percobaan dilakukan 10 kali untuk masing-masing file untuk *download*. Pengujian FTP dilakukan dengan bantuan perangkat lunak Xlight FTP pada sisi *server* dan Smart FTP pada sisi *client*.

3.3.2 Parameter yang Diamati

Pada perancangan ini, parameter yang akan diambil dan diamati untuk aplikasi FTP adalah:

1. *Delay* (Waktu tunda)

- *Delay* pada jaringan adalah waktu yang dibutuhkan oleh satu bit data mulai dikirim hingga sampai ke tujuan, biasa dinyatakan dalam ms. *Delay* dinyatakan

$$Delay(ms) = \frac{panjang\_paket(bit)}{kecepa\tan\_transmisi\_data(kbps)} \quad \text{….. (3.1)}$$

2. Throughput

Adalah kecepatan rata-rata dari data yang berhasil dikirimkan melalui media komunikasi dalam jangka waktu pengamatan tertentu, biasanya dalam *bit per second* (bps). *Throughput* dirumuskan dalam:

$$Throughput = \frac{jumlah\_bit(bit)}{waktu\_pengama\tan(s)} \quad \text{..... (3.2)}$$

3. *Transfer Time*

Adalah total waktu yang dibutuhkan oleh suatu data berjalan pada media untuk sampai dari sumber ke tujuan.

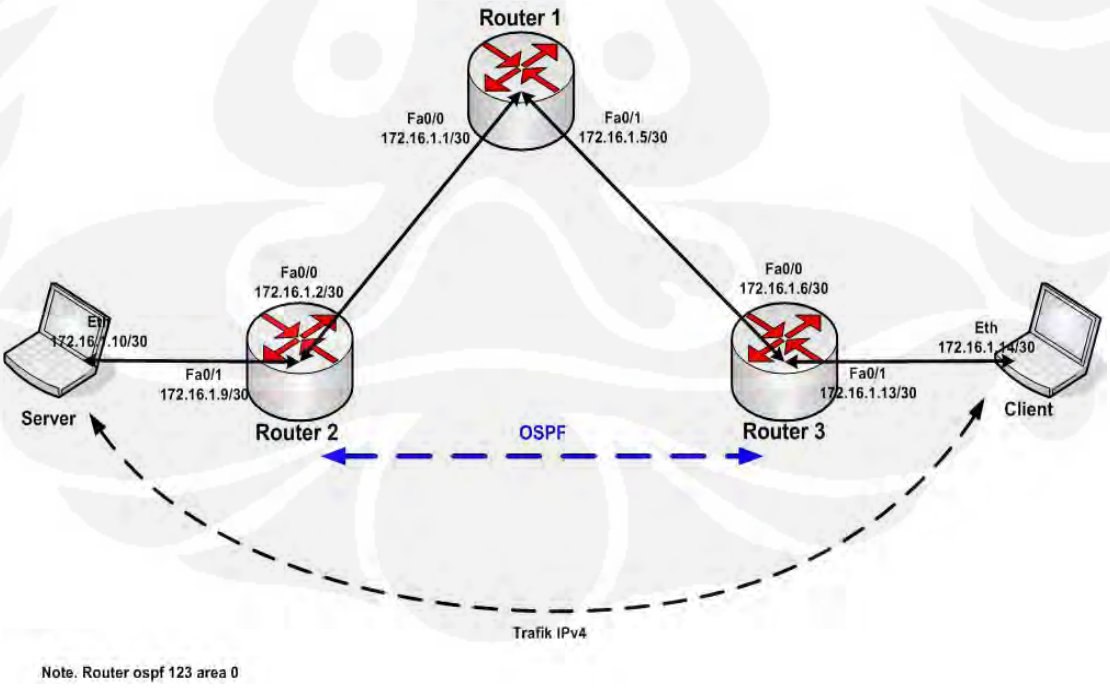
BAB IV
PENGAMBILAN DATA DAN ANALISA

Untuk menganalisa performansi pada jaringan MPLS (baik IPv4 dan IPv6), dalam skripsi ini dilakukan dengan jaringan MPLS sederhana yang terdiri dari 1 *router* P, 2 *router* PE dan 2 laptop yang berfungsi sebagai CE. Jaringan tersebut dibandingkan dengan jaringan IPv4 tanpa MPLS, dalam skripsi ini menggunakan OSPF. Kesuksesan pengiriman data dilihat dari 2 faktor yaitu *delay* dan *throughput*.

4.1 Konfigurasi Jaringan

4.1.1 Jaringan IPv4 tanpa MPLS

Parameter yang digunakan sebagai pembeda antara jaringan MPLS dengan tanpa MPLS pada skripsi ini adalah pada jaringan MPLS ini digunakan protokol *routing* (MPLS mampu menggabungkan 2 layer yaitu *switching* dan *routing*). Protokol *routing* yang digunakan adalah OSPF. Konfigurasi router selengkapnya ada pada Lampiran 1.



Gambar 4.1 Konfigurasi Jaringan IPv4 dengan OSPF

Dari Gambar 4.1 di atas didapatkan *routing table* sebagai berikut, *routing table* ditangkap dari *Router*1

172.16.0.0/30 is subnetted, 4 subnets

O 172.16.1.12 [110/2] via 172.16.1.6, 00:01:50, FastEthernet0/1

O 172.16.1.8 [110/2] via 172.16.1.2, 00:01:50, FastEthernet0/0

C 172.16.1.4 is directly connected, FastEthernet0/1

C 172.16.1.0 is directly connected, FastEthernet0/0

10.0.0.0/32 is subnetted, 3 subnets

O 10.0.1.3 [110/2] via 172.16.1.6, 00:01:50, FastEthernet0/1

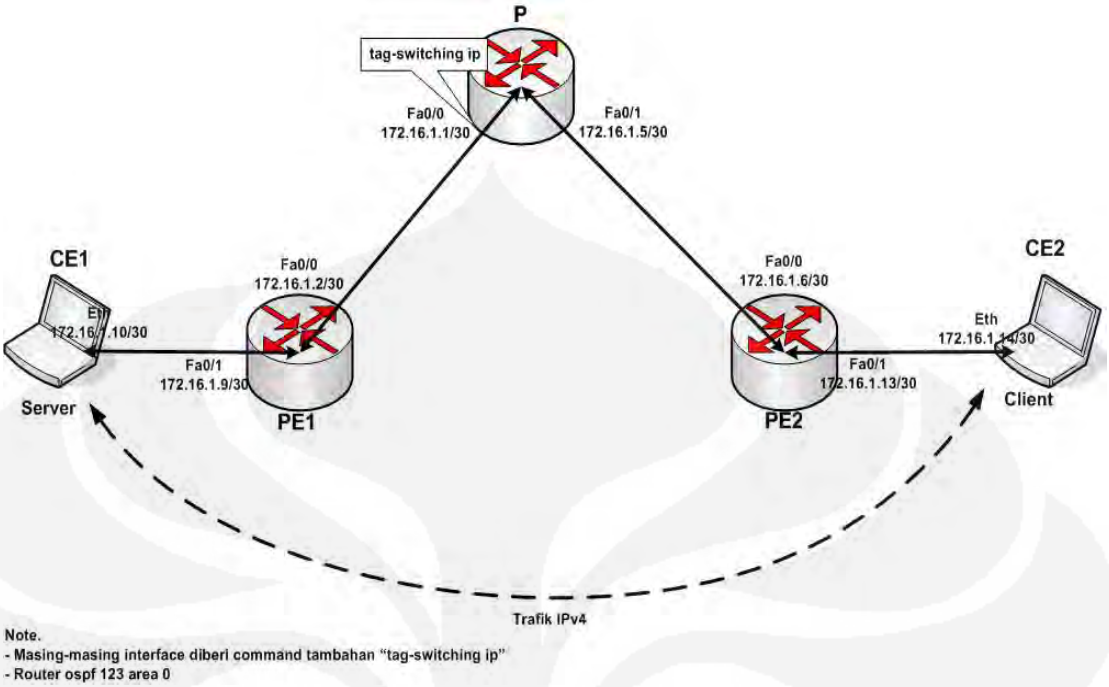
O 10.0.1.2 [110/2] via 172.16.1.2, 00:01:50, FastEthernet0/0

C 10.0.1.1 is directly connected, Loopback0

Dari *routing table* tersebut dapat dilihat masing-masing network IP *router neighbor* terdistribut sebagai OSPF (ditunjukkan dengan huruf O, C menandakan sebagai *direct connected*).

4.1.2 Jaringan MPLS IPv4

Konfigurasi jaringan MPLS dengan IPv4 ini sudah umum digunakan di berbagai *provider* telekomunikasi. Pada skripsi ini disusun jaringan MPLS sederhana yang terdiri dari 1 *router* P, 2 *router* PE dan 2 buah laptop yang berfungsi sebagai CE. *Router* yang digunakan adalah Cisco 2611, sedangkan protokol IGP yang digunakan adalah *Open Short Path First Protocol* (OSPF). Perbedaan antara jaringan MPLS dengan OSPF adalah jika pada MPLS pengiriman paket dengan cara pemberian label pada paket tersebut dan *router* hanya melakukan pembacaan label, sedang pada jaringan OSPF masing-masing *router* yang dilalui membaca dan memeriksa keseluruhan isi paket untuk mengetahui alamat tujuan dan menentukan rute yang akan dilalui paket.



Gambar 4.2 Konfigurasi Jaringan MPLS IPv4

Secara umum konfigurasi jaringan IPv4 dengan dan tanpa MPLS hampir sama, hanya di sini pada tiap-tiap *interface* di *router* ditambahkan perintah "tag-*switching* ip" yang merupakan komponen utama jaringan MPLS. Konfigurasi router selengkapnya ada pada Lampiran 2. Dengan konfigurasi dan alokasi IP seperti pada Gambar 4.2 di atas didapatkan parameter MPLS seperti di bawah. *Capture* dilakukan dari *router* P yang merupakan *core* MPLS.

C2600-1#sh mpls forwarding-table

Local tag	*Outgoing tag or VC*	*Prefix or Tunnel Id*	*Bytes tag switched*	*Outgoing interface*	*Next Hop*
16	*Pop tag*	*172.16.1.12/30*	*0*	*Fa0/1*	*172.16.1.6*
17	*Pop tag*	*172.16.1.8/30*	*0*	*Fa0/0*	*172.16.1.2*
18	*Pop tag*	*10.0.1.3/32*	*0*	*Fa0/1*	*172.16.1.6*
19	*Pop tag*	*10.0.1.2/32*	*0*	*Fa0/0*	*172.16.1.2*

C2600-1#sh mpls forwarding-table detail

Local tag	*Outgoing tag or VC*	*Prefix or Tunnel Id*	*Bytes tag switched*	*Outgoing interface*	*Next Hop*

16 Pop tag 172.16.1.12/30 0 Fa0/1 172.16.1.6

MAC/Encaps=14/14, MRU=1504, Tag Stack{}

CC0204B00000CC0004B000018847

No output feature configured

Per-packet load-sharing

17 Pop tag 172.16.1.8/30 0 Fa0/0 172.16.1.2

MAC/Encaps=14/14, MRU=1504, Tag Stack{}

CC0104B00000CC0004B000008847

No output feature configured

Per-packet load-sharing

18 Pop tag 10.0.1.3/32 0 Fa0/1 172.16.1.6

MAC/Encaps=14/14, MRU=1504, Tag Stack{}

CC0204B00000CC0004B000018847

No output feature configured

Per-packet load-sharing

19 Pop tag 10.0.1.2/32 0 Fa0/0 172.16.1.2

MAC/Encaps=14/14, MRU=1504, Tag Stack{}

CC0104B00000CC0004B000008847

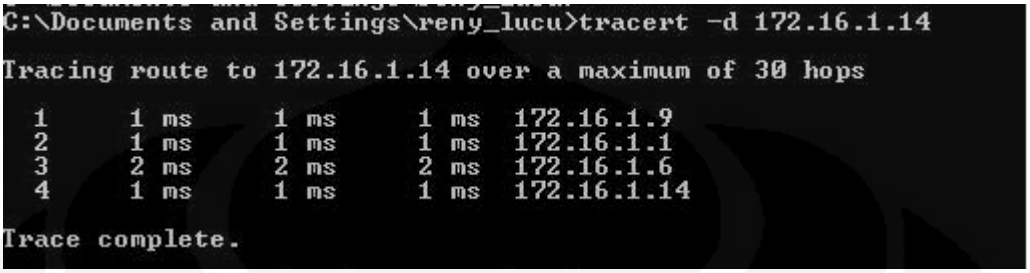
No output feature configured

Per-packet load-sharing

Dari hasil tersebut dapat diketahui label-label pada LFIB yang diberikan untuk masing-masing paket IP network *router neighbor* untuk mencapai *Router*1. Hasil *capture* di atas, *local tag* adalah label yang diberikan oleh LSR yang bersangkutan dan informasinya disebar ke LSR yang lain, sedangkan *outgoing tag* adalah label yang akan menggantikan label yang ada pada paket yang masuk untuk didistribusikan ke LSR selanjutnya. Pada hasil LFIB di atas, dapat dilihat bahwa *outgoing tag* adalah *pop tag*, maksudnya adalah ketika LSR menerima sebuah paket dan membacanya sebagai label sebagaimana label yang tertera pada *local tag*, LSR kemudian akan mengambil satu label paling atas kemudian mengirimnya ke LSR selanjutnya sebagai paket berlabel atau sebuah paket IP.

MPLS cocok digunakan pada jaringan yang luas dan memiliki banyak node yang terhubung satu sama lain. Karena keterbatasan perangkat maka di sini

hanya dilakukan simulasi untuk 5 node yang terhubung dengan topologi bus. Berikut pada Gambar 4.3 adalah hasil *traceroute* dari *server* yang beralamat 172.16.1.10 ke *client* yang beralamat 172.16.1.14



Gambar 4.3 Traceroute dari *Server* ke *Client* untuk Jaringan MPLS IPv4

4.1.3 Jaringan MPLS IPv6 (6PE)

Pada dasarnya tidak ada perbedaan mencolok pada konfigurasi antara MPLS murni IPv4 dengan MPLS untuk IPv6. Yang diperlukan pada *router* 6PE adalah kemampuan *dual stack* yaitu kemampuan mengakomodir baik IPv4 maupun IPv6. Pada Cisco sendiri, tidak semua tipe memenuhi syarat tersebut, hanya beberapa tipe dengan IOS tertentu.

Perintah kunci yang perlu ditambahkan pada *router* 6PE jaringan MPLS IPv4 yang telah terbentuk agar dapat mengakomodir IPv6 ada 2, yaitu:

1. ***neighbor <ip-address> send-label***, perintah ini digunakan untuk mengaktivkan kemampuan *router* untuk mengirim label MPLS dengan BGP. Perintah ini diberikan pada *address family* ipv6 pada BGP. Pada jaringan yang digunakan pada skripsi ini yaitu:

   router bgp 12345
   no bgp default ipv4-unicast
   bgp log-neighbor-changes
   neighbor 10.0.1.3 remote-as 12345
   neighbor 10.0.1.3 update-source Loopback0
   !
   address-family ipv6
   neighbor 10.0.1.3 activate
   neighbor 10.0.1.3 send-label
   network 2001:DB8:FFFF::/48

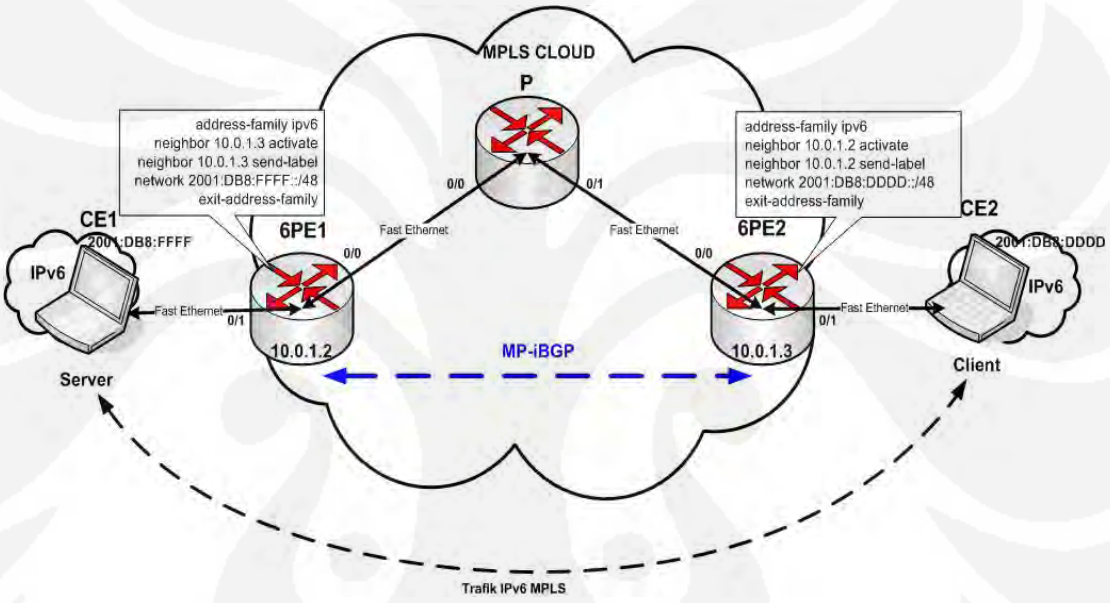
exit-address-family

!

2. ***mpls ipv6 source-interface <interface>,*** perintah ini diberikan pada konfigurasi global *router* dan berfungsi untuk menspesifikasi *interface* yang digunakan sebagai alamat sumber untuk mengenerate paket. Contoh pada jaringan ini:

   mpls ipv6 source-interface Loopback0

Konfugurasi router selengkapnya ada pada Lampiran 3.



Gambar 4.4 Konfigurasi Jaringan 6PE

Pada konfigurasi ini *router* P hanya berisi alamat IPv4 dan tidak mengetahui informasi IPv6 pada PE dan CE, tugasnya hanya melewatkan paket yang dibawa 6PE. Paket IPv6 dilewatkan melalui LSP jaringan MPLS IPv4. Jika dilakukan perintah *show ip route* dari *router* P, jaringan IPv6 di *router neighbor*nya tidak akan muncul. Proses pada Gambar 4.4 di atas dapat dijelaskan sebagai berikut:

a. *Router* 6PE1 menerima paket IPv6 dari *Server* yaitu *prefix* 2001:DB8:FFFF::
b. Paket yang diterima dilakukan pembacaan dan pelabelan.

c. *Router* 6PE1 memberi informasi pada *router-router neighbor*nya bahwa *prefix* 2001:DB8:FFFF:: dapat dicapai melalui 10.0.1.2, demikian juga *router* 6PE2 memberi informasi bahwa *prefix* 2001:DB8:DDDD:: dapat dicapai melalui 10.0.1.3.
d. Label diikat dalam satu LDP ke alamat IPv4 yang merupakan alamat BGP *next hop* (dalam hal ini adalah 10.0.1.3)
e. *Router* P menerima paket MPLS IPv4, membaca label dan meneruskan ke *router* selanjutnya
f. *Router* 6PE2 menerima paket MPLS, mengambil label kemudian melakukan pembacaan paket IPv6 dan meneruskan ke alamat tujuan.
g. Jika terdapat banyak *router* 6PE pada jaringan tersebut, maka alamat IPv4 semua *router neighbor* tersebut harus didapftarkan dalam BGP dan *address-family ipv6*.

Berikut ada *routing table* yang terbaca pada *router* 6PE1:

C2600-PE1#sh ipv6 route
IPv6 Routing Table - 6 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
U - Per-user Static route
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
B 2001:DB8:DDDD::/48 [200/0]
via ::FFFF:10.0.1.3, IPv6-mpls
S 2001:DB8:FFFF::/48 [1/0]
via ::, Ethernet1/1
C 2001:DB8:FFFF::/64 [0/0]
via ::, Ethernet1/1
L 2001:DB8:FFFF::1/128 [0/0]
via ::, Ethernet1/1
L FE80::/10 [0/0]
via ::, Null0
L FF00::/8 [0/0]

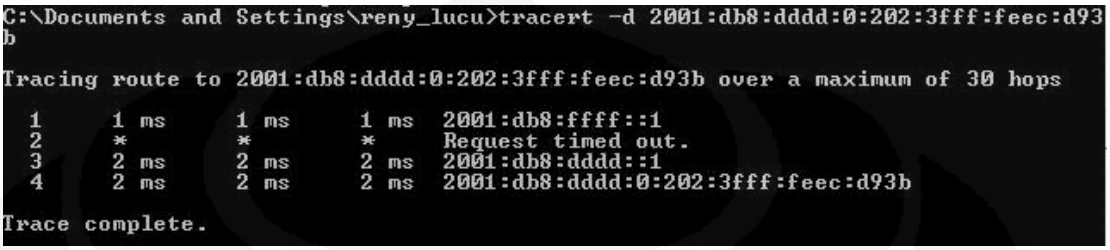
via ::, Null0

Label pada LFIBnya sebagai berikut:

```
C2600-PE1#sh mpls forwarding-table  detail
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     Pop tag     172.16.1.4/30    0          Fa0/0      172.16.1.1
       MAC/Encaps=14/14, MRU=1504, Tag Stack{}
       CC000BAC0000CC010BAC00008847
       No output feature configured
   Per-packet load-sharing
17     Pop tag     10.0.1.1/32      0          Fa0/0      172.16.1.1
       MAC/Encaps=14/14, MRU=1504, Tag Stack{}
       CC000BAC0000CC010BAC00008847
       No output feature configured
   Per-packet load-sharing
18     18          10.0.1.3/32      0          Fa0/0      172.16.1.1
       MAC/Encaps=14/18, MRU=1500, Tag Stack{18}
       CC000BAC0000CC010BAC00008847 00012000
       No output feature configured
   Per-packet load-sharing
19     Aggregate   2001:DB8:FFFF::/48  \
                                0
       MAC/Encaps=0/0, MRU=0, Tag Stack{}
       No output feature configured
   Per-packet load-sharing
```

Dari hasil di atas, *outgoing tag* untuk *prefix* IPv6 adalah *aggregate*. Hal ini disebabkan LSR membaca suatu *range prefix* dan tidak dapat mengirim paket yang masuk dengan metode *label-swapping* biasa tetapi LSR harus menghilangkan semua label pada paket tersebut dan mengirimnya sebagai paket IP dan harus dilakukan IP *lookup* untuk menentukan *prefix* yang lebih spesifik untuk mengirim paket IP tersebut. Untuk itu pada MPLS IPv6 ini paket IPv6 "ditumpangkan" pada paket IPv4 untuk melewati LSP MPLS IPv4 yang ada.

Sebagaimana *router* P yang tidak mengetahui informasi IPv6 pada *router* 6PE dan CE, maka pada sisi CE yang menjalankan *prefix* IPv6 juga tidak mengetahui informasi IPv4 yang dilaluinya. Berikut pada Gambar 4.5 adalah hasil *traceroute* dari *server* ke *client* untuk jaringan MPLS IPv6 (6PE).



Gambar 4.5 *Traceroute Server* ke *Client* untuk Jaringan MPLS IPv6 (6PE)

Pengalamatan pada laptop *server* dan *client* diberikan secara *dyamic* atau otomatis dari *router* yang terhubung. Hal ini dimaksudkan untuk mencegah atau mengurangi terjadinya duplikat IP ketika terhubung ke jaringan yang luas. Dari hasil *traceroute* di atas, dapat dilihat pada hop kedua, *server* tidak mengetahui informasi IPv4 yang terdapat pada *router* P.

4.2 Performa Aplikasi FTP pada Jaringan

Pengujian yang dilakukan di sini adalah aplikasi *File Transfer Protocol* (FTP) yang merupakan salah satu protokol yang memanfaatkan protokol TCP/IP. Pada TCP (*Transmission Control Protocol*) memiliki kemampuan untuk menjamin transfer dan kontrol data hingga sampai ke tujuan dengan adanya proses *acknowledgement* (ACK) yang dimiliki, dengan demikian dapat mengurangi bahkan meniadakan adanya *packet loss*. Pada prosesnya, suatu koneksi FTP memiliki bentuk koneksi *client-server*. FTP *server* menyimpan file-file yang di*download* oleh FTP *client*. Proses diawali dengan *three way handshaking*, yaitu *client* mengirimkan SYN ke *server*, *server* membalas dengan SYN-ACK, barulah *client* mengirimkan ACK ke *server* dan terbentuk sebuah hubungan antara *client-server*. Pada pengujian ini *user* disetting sebagai a*nonymous*, sehingga setiap FTP *client* dapat mengakses FTP *server* tanpa perlu terdaftar. File yang di*download* dari *server* memiliki ukuran bervariasi yaitu 16MB, 32MB, 64MB, 128MB, dan 256 MB. Masing-masing file akan

di*download* sebanyak 10 kali untuk kemudian didapatkan rata-rata dari hasil pengujian. File-file tersebut adalah:

16m.rar , berukuran 17,125,640 byte

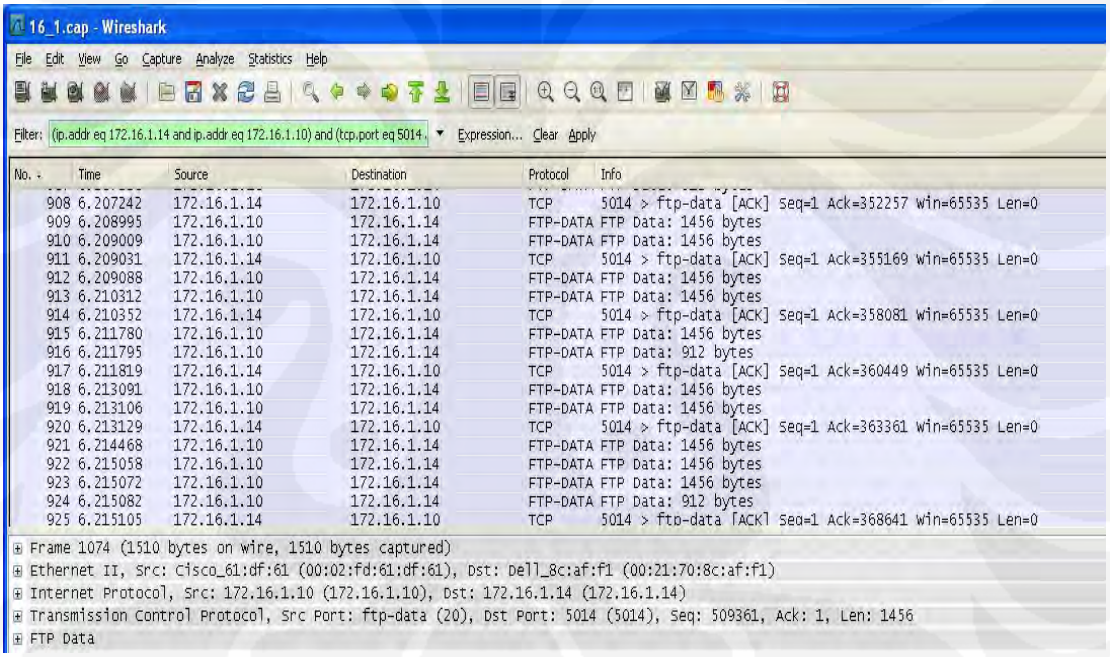
32m.rar, berukuran 32,974,684 byte

64m.rar, berukuran 65,354,045 byte

128m.rar, berukuran 131,707,992 byte

256m.rar, berukuran 263,752,385 byte

Paket-paket TCP ini akan ditangkap dengan *software* wireshark untuk didapatkan nilai dari paramater-parameter yang akan diamati. Pada Gambar 4.6 di bawah merupakan contoh hasil *capture* yang dilakukan oleh Wireshark.



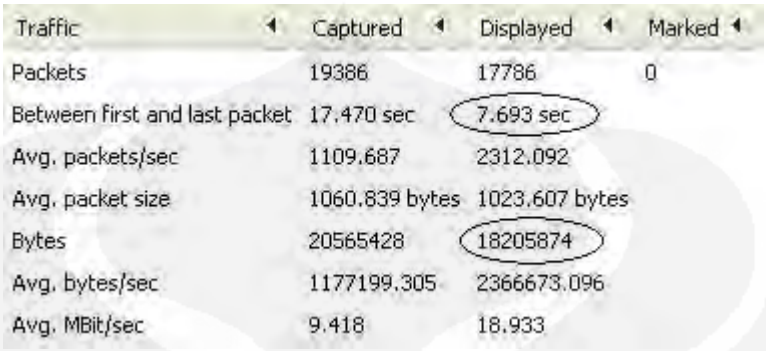
Gambar 4.6 Contoh hasil *capture* paket data oleh Wireshark

Paramater yang diamati pada pengujian ini adalah *delay*, *throughput*, serta *transfer time*. Parameter-parameter tersebut merupakan besaran yang dirasa mampu menunjukkan performa aplikasi FTP pada suatu jaringan.

4.2.1 Delay

Delay adalah waktu yang dibutuhkan oleh satu bit data mulai dikirim hingga sampai ke tujuan. Pada pengujian ini, nilai *delay* didapatkan dari nilai

transfer time dan bytes yang terbaca dari wireshark. Contohnya dapat dilihat pada Gambar 4.7 di bawah:



Traffic	Captured	Displayed	Marked
Packets	19386	17786	0
Between first and last packet	17.470 sec	7.693 sec	
Avg. packets/sec	1109.687	2312.092	
Avg. packet size	1060.839 bytes	1023.607 bytes	
Bytes	20565428	18205874	
Avg. bytes/sec	1177199.305	2366673.096	
Avg. MBit/sec	9.418	18.933	

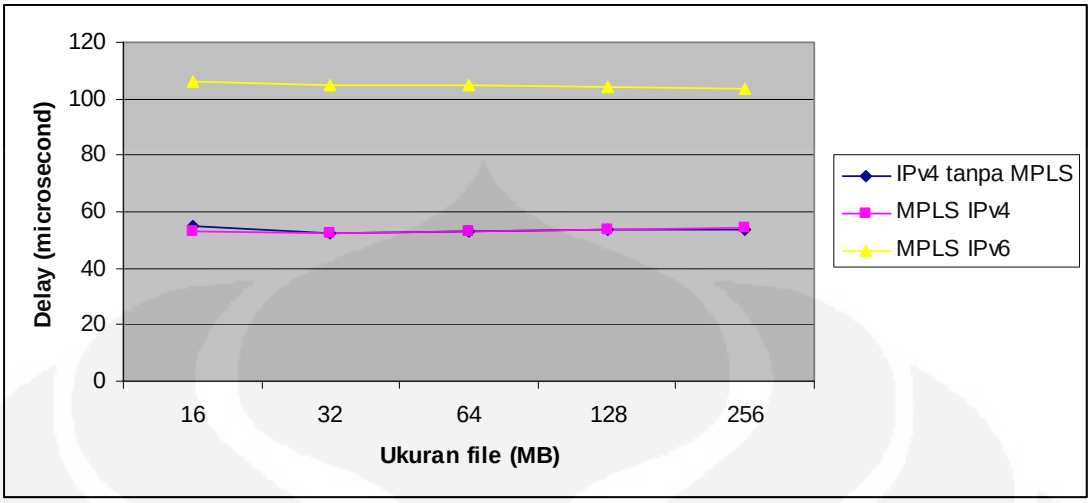
Gambar 4.7 Contoh hasil *summary* paket yang ditangkap oleh Wireshark

Dari data di atas, untuk mendapatkan *delay* adalah nilai *transfer time* dibagi dengan nilai byte yang telah diubah menjadi bit. Hasil selengkapnya untuk masing-masing file dan konfigurasi dapat dilihat pada Lampiran 4,5 dan 6. Tabel 4.1 berikut adalah nilai rata-rata dari 10 kali percobaan yang dilakukan.

Tabel 4.1 Data rata-rata nilai *delay*

Ukuran File (MB)	Delay (µs)		
	IPv4 tanpa MPLS	MPLS IPv4	MPLS IPv6
16	55.2	52.9	106.34
32	52.4	52.3	104.56
64	53	52.8	104.92
128	53.8	53.6	104.39
256	53.9	54.2	103.7

Dari Tabel 4.1 di atas dapat dilihat bahwa semakin besar ukuran file maka *delay* juga akan semakin besar, namun tidak drastis, kenaikannya hanya berkisar antara 0.9%-1.2%. *Delay* menunjukkan waktu yang dibutuhkan suatu jaringan untuk menghantarkan paket data, semakin kecil *delay* berarti performa jaringan tersebut akan makin baik.



Gambar 4.8 Grafik perbandingan delay dengan ukuran file untuk tiap konfigurasi

Dari grafik yang ditunjukkan pada Gambar 4.8 di atas dapat dilihat tidak ada perbedaan yang signifikan antara jaringan IPv4 tanpa MPLS dengan jaringan IPv4 dengan MPLS. Hal ini disebabkan karena dalam pengujian ini hanya sedikit node yang digunakan, MPLS akan dapat dibandingkan dengan forwarding ip biasa/konvensional jika node yang terbentuk banyak, mencapai puluhan atau ratusan. Dari grafik tersebut nilai jaringan IPv4 (dengan dan tanpa MPLS) memiliki nilai rata-rata *delay* 52.3µs – 55.2 µs tergantung dari besar ukuran file. Data tabel dan grafik dapat dilihat juga nilai *delay* untuk jaringan MPLS IPv6 (6PE) berkisar antara 103.7 µs – 106.34 µs. Jaringan MPLS IPv4 memiliki *delay* yang lebih kecil dibanding jaringan MPLS IPv6 (6PE), hal ini berarti jaringan MPLS IPv4 lebih baik 92.65% - 98.3%.

Delay IPv6 yang lebih besar disebabkan karena jumlah bit IPv6 yang lebih panjang yaitu 128 bit dari pada bit IPv4 yang hanya 32 bit. Selain itu panjang *header* IPv6 yang 2 kali lebih panjang dari IPv4 juga merupakan faktor yang menyebabkan *delay* pengiriman paket lebih besar.

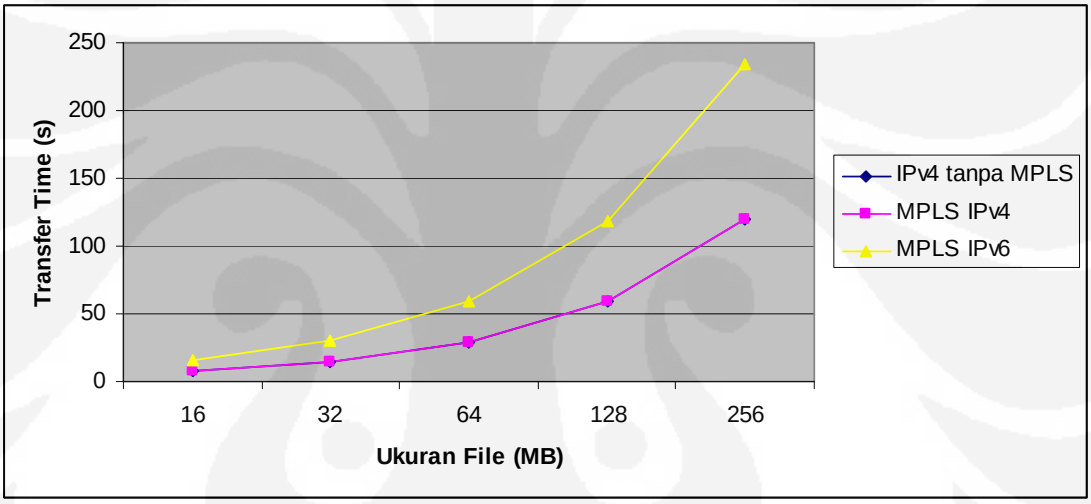
4.2.2 Transfer Time

Transfer time adalah jumlah total waktu yang dibutuhkan oleh suatu jaringan untuk mentransfer data dari FTP *server* ke FTP *client*, atau sebaliknya. Dari hasil pengujian data yang dilakukan 10 kali untuk tiap ukuran file dan konfigurasi, didapatkan rata-rata hasil *transfer time* sebagai berikut:

Tabel 4.2 Data nilai rata-rata *transfer time*

Ukuran File (MB)	Transfer Time (s)		
	IPv4 tanpa MPLS	MPLS IPv4	MPLS IPv6
16	8.0439	7.7043	15.805
32	14.734	14.6976	29.7136
64	29.4556	29.3182	59.4813
128	59.6652	59.4652	118.0515
256	119.3367	119.9367	234.1883

Dari Tabel 4.2 di atas dapat dilihat bahwa semakin besar ukuran file yang dikirim maka *transfer time* juga akan makin besar. Dengan demikian performa jaringan akan semakin baik jika *transfer time* makin kecil.



Gambar 4.9 Grafik *transfer time* terhadap ukuran file untuk setiap konfigurasi

Berdasarkan grafik pada Gambar 4.9 di atas, MPLS IPv4 memiliki nilai *transfer time* yang lebih kecil. Nilai *transfer time* jaringan MPLS IPv4 berkisar antara 7.7043s - 119.9367s, sedangkan untuk jaringan MPLS IPv6 (6PE) berkisar antara 15.805s - 234.1883s, tergantung dari ukuran data yang dikirimkan. Hal ini berarti jaringan MPLS IPv4 95.26% - 105.15%.

Gambar pada Lampiran 7 akan menunjukkan grafik *transfer time* yang didapatkan untuk 10 kali pengambilan data dari setiap filenya. Hal ini untuk mengetahui tingkat kestabilan jaringan. Dari grafik tersebut dapat dilihat bahwa jaringan MPLS IPv4 cenderung lebih stabil dalam pengiriman data dibanding jaringan MPLS IPv6 (6PE).

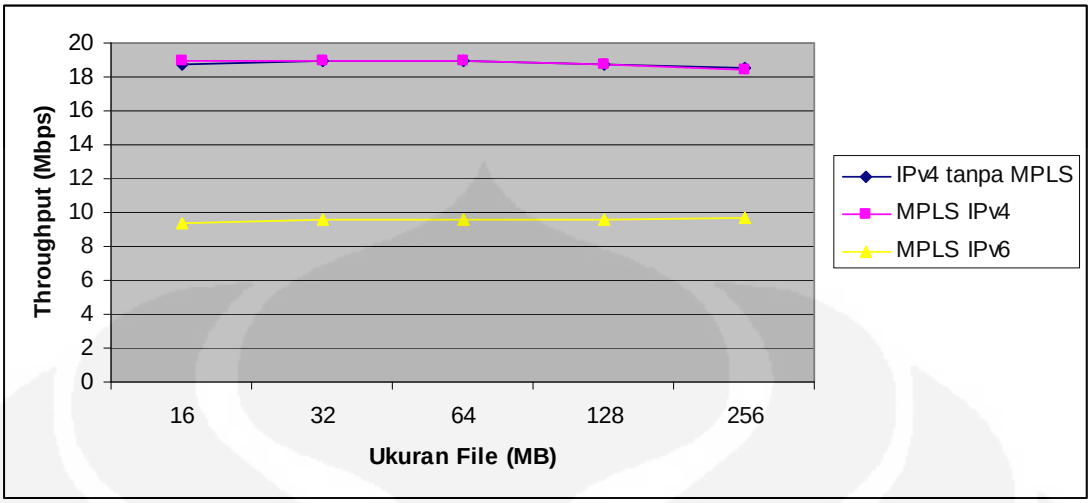
4.2.3 Throughput

Throughput adalah kecepatan rata-rata dari data yang berhasil dikirimkan melalui suatu media komunikasi dalam jangka waktu pengamatan tertentu, dinyatakan dalam *bit per second*. Nilai *throughput* sangat dipengaruhi oleh nilai *transfer time*, semakin besar *transfer time* maka *throughput* akan semakin kecil sehingga performa jaringan semakin buruk. Sebaliknya jika *transfer time* semakin kecil maka nilai *throughput* makin besar dan performa jaringan makin baik. Dari hasil pengujian untuk 10 kali pengambilan data seperti pada Lampiran 4,5 dan 6, di bawah ini adalah tabel hasil rata-rata *throughput* untuk masing-masing konfigurasi jaringan.

Tabel 4.3 Data nilai rata-rata *throughput*

Ukuran File (MB)	Throughput (Mbps)		
	IPv4 tanpa MPLS	MPLS IPv4	MPLS IPv6
16	18.746	18.907	9.4045
32	18.9112	18.9532	9.5642
64	18.9253	18.9491	9.553
128	18.7175	18.7074	9.5918
256	18.5159	18.4659	9.6471

Pada Tabel 4.3 di atas dapat dilihat bahwa nilai rata-rata *throughput* akan naik seiring dengan kenaikan ukuran file yang ditransfer. Karena *throughput* menunjukkan kecepatan transfer data suatu jaringan, maka semakin besar nilai *throughput* akan semakin baik performa jaringan tersebut.



Gambar 4.10 Grafik *throughput* terhadap ukuran file untuk setiap konfigurasi

Berdasarkan grafik pada Gambar 4.11 di atas, *throughput* jaringan IPv4 yang menggunakan MPLS dan tanpa MPLS tidak menunjukkan perbedaan yang signifikan namun perbedaannya dengan jaringan MPLS IPv6 (6PE) cukup besar. *Throughput* jaringan MPLS IPv4 berkisar antara 18.4659 Mbps – 18.9253 Mbps, sedangkan *throughput* jaringan MPLS IPv6 (6PE) berkisar antara 9.4045 Mbps – 9.6471 Mbps. Hal ini berarti jaringan MPLS IPv4 lebih baik 96. 17% - 96.35%. Pada gambar 4.12 di bawah merupakah contoh perbandingan nilai *throughput* antara jaringan MPLS IPv4 dengan MPLS IPv6 (6PE) yang terbaca dari Wireshark.

Traffic	Captured	Displayed	Marked
Packets	282700	271635	0
Between first and last packet	124.277 sec	117.731 sec	
Avg. packets/sec	2274.756	2307.256	
Avg. packet size	1036.863 bytes	1018.794 bytes	
Bytes	293121194	276740188	
Avg. bytes/sec	2358610.831	2350619.067	
Avg. MBit/sec	18.869	18.805	

Traffic	Captured	Displayed	Marked
Packets	296079	273105	0
Between first and last packet	253.198 sec	233.822 sec	
Avg. packets/sec	1169.358	1168.003	
Avg. packet size	1067.861 bytes	1033.601 bytes	
Bytes	316171104	282281642	
Avg. bytes/sec	1248711.734	1207249.360	
Avg. MBit/sec	9.990	9.658	

Gambar 4.12 *Throughput download* file 256MB IPv4 (kiri) dan IPv6 (kanan)

4.3 Analisa Keseluruhan

Setelah melakukan perbandingan dari hasil pengujian untuk ketiga konfigurasi, dapat disimpulkan bahwa jaringan IPv4 dengan MPLS dan tanpa MPLS tidak menunjukkan perbedaan yang signifikan jika diaplikasikan pada jaringan dengan skala kecil.

Namun jika jaringan MPLS IPv4 dibandingan dengan jaringan MPLS IPv6 (6PE), dari hasil pengujian dapat disimpulkan bahwa jaringan MPLS IPv4 memiliki keunggulan di setiap parameter pengujian dibanding jaringan MPLS IPv6 (6PE). MPLS IPv6 (6PE) bisa dikatakan sebagai salah satu solusi untuk mengurangi biaya jika diperlukan adanya migrasi dari IPv4 ke IPv6 pada jaringan MPLS, namun beberapa kelemahan seperti masalah *delay* tinggi di atas perlu dipertimbangkan atau dicarikan upaya untuk mangatasai kekurangan tersebut.

BAB V
KESIMPULAN

Dari hasil pengujian didapatkan hasil sebagai berikut:

1. Delay jaringan MPLS IPv4 memiliki performa lebih baik sebesar 92.65% - 98.3% dibanding jaringan MPLS IPv6 (6PE).
2. Transfer time jaringan MPLS IPv4 memiliki kecepatan lebih tinggi 95.26% - 105.15% dibanding jaringan MPLS IPv6 (6PE).
3. Throughput jaringan MPLS IPv4 lebih besar 96.17% - 96.35% dibanding jaringan MPLS IPv6 (6PE).
4. Performa IPv4 pada jaringan MPLS dan non MPLS tidak menunjukkan perbedaan yang signifikan jika diaplikasikan pada jaringan yang berskala kecil.

DAFTAR REFERENSI

1 De Ghein, Luc (2007). MPLS Fundamental. Indianapolis: Cisco Press

2 Gallagher, Rick (2003). MPLS Training Guide: Building Multi Protocol Label Swicthing Networks. Rockland, MA: Syngress Publishing, Inc.

3 Lobo, Lanci (2005). MPLS Configuration on Cisco IOS Software. Indianapolis: Cisco Press.

4 Nasrun, Irvan (2005). Mengenal IP Versi 6.

5 Popoviciu, Ciprian & Abegnoli-Levy, Eric & Grossetete, Patrick (2006). Deploying IPv6 Networks. Indianapolis: Cisco Press.

6 Wastuwibowo, Kuncoro (2003). Jaringan MPLS. Whitepaper.

7 Cisco System, Inc (2008). Cisco IOS IPv6 Configuration Guide. San Jose: Cisco Americas Headquarter. Didownload dari Cisco.com pada 18 Maret 2009 http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/12_4/ipv6_12_4_book.html

8 Cisco System, “MPLS”. Diakses pada 18 Maret 2009 dari Cisco.com http://www.cisco.com/en/US/tech/tk436/tk428/tsd_technology_support_protocol_home.html

9 Cisco System, ”Configuring Basic MPLS Using OSPF”. Diakses pada 18 Maret 2009 dari Cisco.com http://www.cisco.com/en/US/tech/tk436/tk428/tsd_technology_support_protocol_home.html

10 Cisco System, Cisco IOS IPv6 Provider Edge Router (6PE) over MPLS. Didownload tanggal 30 April 2009. http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_data_sheet09186a008052edd3.html

11 Hogg, Scott (2007). Microsoft IPv6 Commands. Global Technology Resources, Inc. Didownload pada 5 April 2009. http://www.rmwtug.org/Talks/IPv6_2007-09/Microsoft_IPv6_Commands-2007-09-18.doc

12 Wikipedia, “IPv6”. Diakses tanggal 18 Maret 2009 http://en.wikipedia.org/wiki/IPv6

13 Wikipedia “Network Delay”. Diakses 20 Mei 2009, dari Wikipedia http://en.wikipedia.org/wiki/Network_delay

14 Wikipedia, “File Transfer Protocol”. Diakses tanggal 7 April 2009. http://en.wikipedia.org/wiki/File_transfer_protocol

15 Wikipedia,”Throughput”. Diakses tanggal 7 April 2009 http://en.wikipedia.org/wiki/Throughput

16 Wikipedia, “Transmission Control Protocol”. Diakses tanggal 28 April 2009 http://en.wikipedia.org/wiki/Transmission_Control_Protocol

LAMPIRAN

Lampiran 1
Konfigurasi jaringan IPv4 tanpa MPLS (OSPF)

```
Router#1
interface Loopback0
 ip address 10.0.1.1 255.255.255.255
!
interface FastEthernet0/0
 description To C2600-2
 ip address 172.16.1.1 255.255.255.252
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description to C2600-3
 ip address 172.16.1.5 255.255.255.252
 duplex auto
 speed auto
!
router ospf 123
 router-id 10.0.1.1
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 172.16.1.0 0.0.0.255 area 0

Router#2
interface Loopback0
 ip address 10.0.1.2 255.255.255.255
!
interface FastEthernet0/0
 description To C2600-1
 ip address 172.16.1.2 255.255.255.252
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description To CE-1
 ip address 172.16.1.9 255.255.255.252
 duplex auto
 speed auto
!
router ospf 123
 router-id 10.0.1.2
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 172.16.1.0 0.0.0.255 area 0
```

(lanjutan)

```
Router#3
interface Loopback0
 ip address 10.0.1.3 255.255.255.255
!
interface FastEthernet0/0
 description To C2600-1
 ip address 172.16.1.6 255.255.255.252
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description To CE-2
 ip address 172.16.1.13 255.255.255.252
 duplex auto
 speed auto
!
router ospf 123
 router-id 10.0.1.3
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 172.16.1.0 0.0.0.255 area 0
```

Laptop Server
IP address: 172.16.1.10
Subnet mask: 255.255.255.252
IP gateway: 172.16.1.9

Laptop Client
IP address: 172.16.1.14
Subnet mask: 255.255.255.252
IP gateway: 172.16.1.13

Lampiran 2
Konfigurasi MPLS IPv4

```
Router P
interface Loopback0
 ip address 10.0.1.1 255.255.255.255
!
interface FastEthernet0/0
 description To C3600-PE1 Fa0/0
```

(lanjutan)

```
 ip address 172.16.1.1 255.255.255.252
 duplex auto
 speed auto
 tag-switching ip
!
interface FastEthernet0/1
 description To C3600-PE2 Fa0/0
 ip address 172.16.1.5 255.255.255.252
 duplex auto
 speed auto
 tag-switching ip
!
router ospf 123
 router-id 10.0.1.1
 log-adjacency-changes
 redistribute static
 network 10.0.1.0 0.0.0.255 area 0
 network 172.16.1.0 0.0.0.255 area 0
!
router bgp 12345
 no synchronization
 bgp router-id 10.0.1.1
 bgp log-neighbor-changes
 neighbor 10.0.1.2 remote-as 12345
 neighbor 10.0.1.2 update-source Loopback0
 neighbor 10.0.1.3 remote-as 12345
 neighbor 10.0.1.3 update-source Loopback0
 no auto-summary

Router PE1
interface Loopback0
 ip address 10.0.1.2 255.255.255.255
!
interface FastEthernet0/0
 description To C3600-P Fa0/0
 ip address 172.16.1.2 255.255.255.252
 duplex auto
 speed auto
 tag-switching ip
!
interface FastEthernet0/1
 description To C3600-CE1
 ip address 172.16.1.9 255.255.255.252
 tag-switching ip
!
router ospf 123
 router-id 10.0.1.2
```

(lanjutan)

```
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 172.16.1.0 0.0.0.255 area 0
!
router bgp 12345
 bgp log-neighbor-changes
 neighbor 10.0.1.3 remote-as 12345
 neighbor 10.0.1.3 update-source Loopback0
 neighbor 10.0.1.1 remote-as 12345
 neighbor 10.0.1.1 update-source Loopback0
```

Router PE2

```
interface Loopback0
 ip address 10.0.1.3 255.255.255.255
!
interface FastEthernet0/0
 description To C3600-P Fa0/1
 ip address 172.16.1.6 255.255.255.252
 duplex auto
 speed auto
 tag-switching ip
!
interface FastEthernet0/1
 description To C2600_CE1
 ip address 172.16.1.13 255.255.255.252
 tag-switching ip
!
router ospf 123
 router-id 10.0.1.3
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 172.16.1.0 0.0.0.255 area 0
!
router bgp 12345
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.0.1.2 remote-as 12345
 neighbor 10.0.1.2 update-source Loopback0
 neighbor 10.0.1.3 remote-as 12345
 neighbor 10.0.1.3 update-remote Loopback0
!
```

Laptop Server

IP address: 172.16.1.10
Subnet mask: 255.255.255.252
IP gateway: 172.16.1.9

Laptop Client
IP address: 172.16.1.14
Subnet mask: 255.255.255.252
IP gateway: 172.16.1.13

Lampiran 3
Konfigurasi Jaringan MPLS IPv6 (6PE)

```
Router P
ip cef
tag-switching tdp router-id Loopback0
!
interface Loopback0
 ip address 10.0.1.1 255.255.255.255
!
interface FastEthernet0/0
 description To C3600-PE1 Fa0/0
 ip address 172.16.1.1 255.255.255.252
 duplex auto
 speed auto
 tag-switching ip
!
interface FastEthernet0/1
 description To C3600-PE2 Fa0/0
 ip address 172.16.1.5 255.255.255.252
 duplex auto
 speed auto
 tag-switching ip
!
router ospf 123
 router-id 10.0.1.1
 log-adjacency-changes
 redistribute static
 network 10.0.1.0 0.0.0.255 area 0
 network 172.16.1.0 0.0.0.255 area 0
!
router bgp 12345
 no synchronization
bgp router-id 10.0.1.1
 bgp log-neighbor-changes
 neighbor 10.0.1.2 remote-as 12345
 neighbor 10.0.1.2 update-source Loopback0
 neighbor 10.0.1.3 remote-as 12345
 neighbor 10.0.1.3 update-source Loopback0
 no auto-summary
!
```

Router 6PE1

```
ip cef
ipv6 unicast-routing
ipv6 cef
mpls ipv6 source-interface Loopback0
tag-switching tdp router-id Loopback0
!
interface Loopback0
 ip address 10.0.1.2 255.255.255.255
!
interface FastEthernet0/0
 description To C3600-P Fa0/0
 ip address 172.16.1.2 255.255.255.252
 duplex auto
 speed auto
 tag-switching ip
!
interface FastEthernet0/1
 description To C3600-CE1
 ip address 192.168.1.1 255.255.255.252
 speed auto
 full-duplex
 ipv6 address 2001:DB8:FFFF::1/64
 ipv6 enable
 tag-switching ip
!
router ospf 123
 router-id 10.0.1.2
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 172.16.1.0 0.0.0.255 area 0
!
router bgp 12345
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
neighbor 10.0.1.3 remote-as 12345
 neighbor 10.0.1.3 update-source Loopback0
 !
 address-family ipv6
 neighbor 10.0.1.3 activate
 neighbor 10.0.1.3 send-label
 network 2001:DB8:FFFF::/48
 exit-address-family
!
ip http server
ip classless
!
!
```

(lanjutan)

```
ipv6 route 2001:DB8:FFFF::/48 FastEthernet0/1

Router 6PE2
cef
ipv6 unicast-routing
ipv6 cef
mpls ipv6 source-interface Loopback0
tag-switching tdp router-id Loopback0
!
interface Loopback0
 ip address 10.0.1.3 255.255.255.255
!
interface FastEthernet0/0
 description To C3600-P Fa0/1
 ip address 172.16.1.6 255.255.255.252
 duplex auto
 speed auto
 tag-switching ip
!
interface FastEthernet0/1
 description To C2600_CE1
 ip address 192.168.0.1 255.255.255.252
 speed auto
 half-duplex
 ipv6 address 2001:DB8:DDDD::1/64
 ipv6 enable
 tag-switching ip
!
router ospf 123
 router-id 10.0.1.3
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 172.16.1.0 0.0.0.255 area 0
!
router bgp 12345
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.0.1.2 remote-as 12345
 neighbor 10.0.1.2 update-source Loopback0
 !
 address-family ipv6
 neighbor 10.0.1.2 activate
 neighbor 10.0.1.2 send-label
 network 2001:DB8:DDDD::/48
 exit-address-family
!
no ip http server
```

(lanjutan)

```
ip classless
!
!
ipv6 route 2001:DB8:DDDD::/48 FastEthernet0/1
```

Lampiran 4
Data pengujian konfigurasi IPv4 tanpa MPLS

Delay (µs)										
	Pengambilan ke									
Ukuran File (MB)	1	2	3	4	5	6	7	8	9	10
16	55.1	54.5	54.17	54.39	57.12	55.58	55.8	55.07	55.43	55.1
32	52.83	52.89	52.85	53.11	47.87	52.9	52.84	53.04	52.87	52.82
64	53.2	52.98	53.06	52.95	52.92	52.8	52.05	53.04	53.08	53.1
128	54.42	53.73	54.18	53.39	54.12	53.78	53.7	53.31	54.03	53.52
256	53.18	53.13	53.38	53.23	54.88	54.48	55.23	53.5	53.4	53.71

Transfer Time (s)										
	Pengambilan ke									
Ukuran File (MB)	1	2	3	4	5	6	7	8	9	10
16	8.025	7.937	7.892	7.92	8.322	8.095	8.128	8.022	8.073	8.025
32	14.717	14.732	14.72	14.793	14.711	14.736	14.718	14.775	14.727	14.711
64	29.554	29.428	29.483	29.419	29.398	29.34	29.474	29.467	29.492	29.501
128	60.334	59.573	60.055	59.186	60.02	59.618	59.533	59.106	59.894	59.333
256	117.731	117.627	118.174	117.85	121.516	120.618	122.27	118.448	118.217	118.916

Throughput (Mbps)										
	Pengambilan ke									
Ukuran File (MB)	1	2	3	4	5	6	7	8	9	10
16	18.184	18.163	18.278	18.094	18.153	18.264	18.114	18.186	18.098	18.126
32	18.926	18.907	18.846	18.869	18.896	19.058	18.925	18.879	18.873	18.933
64	18.797	19.026	19.041	18.798	18.983	18.936	18.849	18.982	19.009	18.832
128	18.376	18.61	18.872	18.731	18.784	18.595	18.941	18.757	18.824	18.685
256	18.805	18.821	17.827	18.758	18.219	17.984	18.718	18.69	18.727	18.61

Lampiran 5
Data pengujian konfigurasi IPv4 dengan MPLS

Delay (µs)										
	Pengambilan ke									
Ukuran File (MB)	1	2	3	4	5	6	7	8	9	10
16	52.82	52.92	52.81	54.39	53.01	52.42	52.81	52.56	52.8	52.42
32	52.84	52.89	52.72	52.72	47.7	52.47	52.84	53.04	52.7	52.81
64	53.2	52.56	52.52	52.53	52.68	52.1	53.05	52.68	52.6	53.1
128	54.42	53.74	53.27	53.39	53.23	53.78	53.7	53.32	54.02	53.52
256	53.18	53.13	56.1	53.23	54.89	57.2	53.42	53.5	53.4	53.72

Transfer Time (s)										
	Pengambilan ke									
Ukuran File (MB)	1	2	3	4	5	6	7	8	9	10
16	7.693	7.707	7.692	7.92	7.722	7.636	7.693	7.656	7.689	7.635
32	14.717	14.732	14.685	14.683	14.66	14.616	14.718	14.775	14.679	14.711
64	29.554	29.195	29.177	29.185	29.265	29.34	29.474	29.267	29.224	29.501
128	60.334	59.573	59.055	59.186	59.02	59.618	59.533	59.106	59.894	59.333
256	117.731	117.627	124.174	117.85	121.516	126.618	118.27	118.448	118.217	118.916

Throughput (Mbps)										
	Pengambilan ke									
Ukuran File (MB)	1	2	3	4	5	6	7	8	9	10
16	18.933	18.898	18.936	18.386	18.865	19.075	18.936	19.026	18.94	19.075
32	18.926	18.907	18.966	18.969	18.996	19.058	18.925	18.879	18.973	18.933
64	18.797	19.026	19.041	19.036	18.983	18.936	18.849	18.982	19.009	18.832
128	18.376	18.61	18.771	18.731	18.784	18.595	18.941	18.757	18.824	18.685
256	18.805	18.821	17.827	18.758	18.219	17.484	18.718	18.69	18.727	18.61

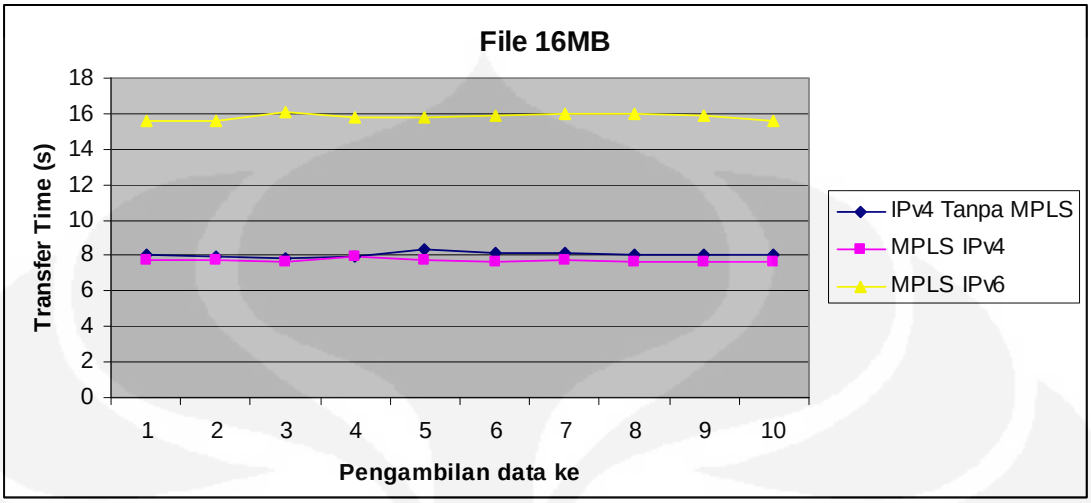
Lampiran 6
Data pengujian konfigurasi IPv6 dengan MPLS

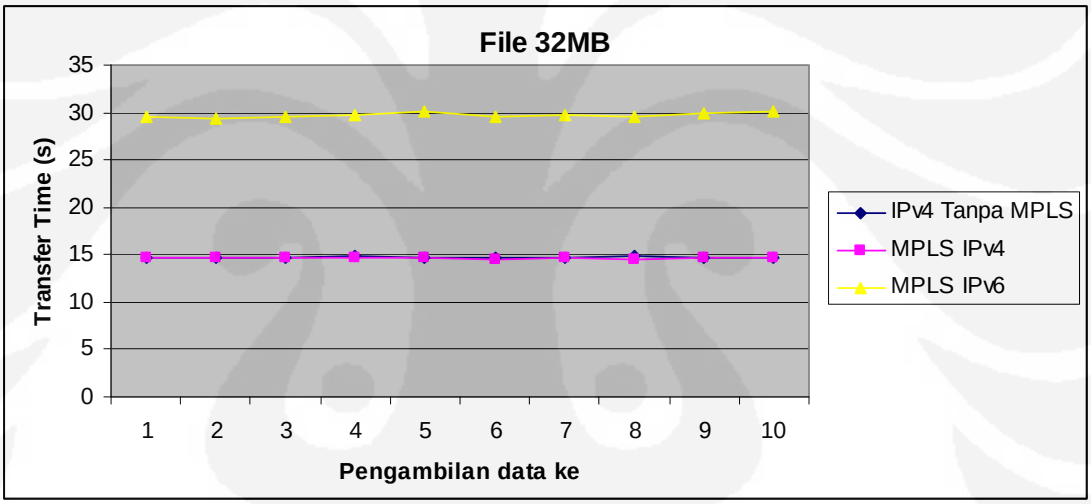
Delay (µs)										
	Pengambilan ke									
Ukuran File (MB)	1	2	3	4	5	6	7	8	9	10
16	105.01	104.82	107.92	105.97	106.06	107.22	107.27	107.34	106.86	104.94
32	103.93	103.48	104	104.7	105.76	103.97	104.57	104.1	104.95	106.15
64	105.32	104.3	104.5	105.15	105.29	104.94	104.97	105.2	104.41	105.12
128	104.48	104.56	104.35	104.48	104.49	104.05	103.83	104.49	104.51	104.65
256	103.54	103.85	103.7	103.7	103.39	103.76	103.78	103.75	103.78	103.75

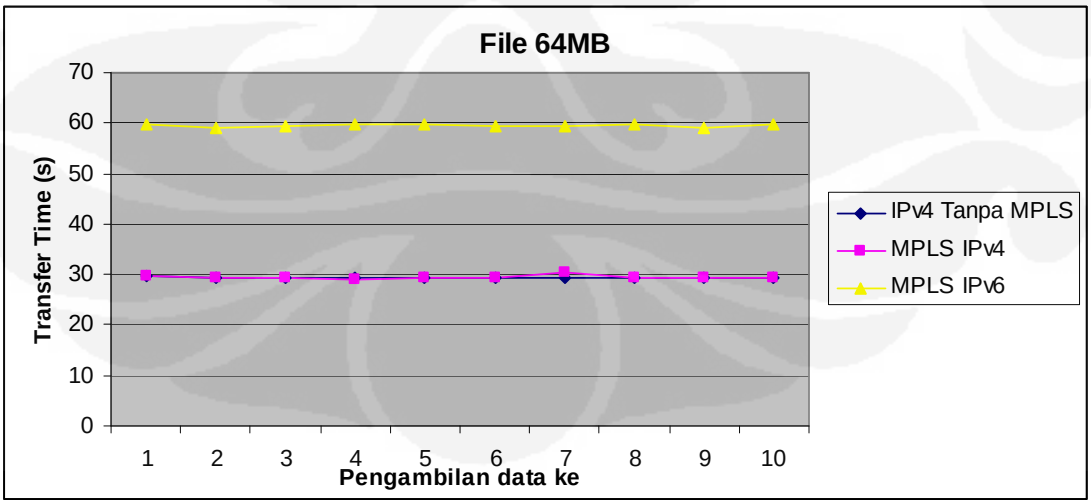
Transfer Time (s)										
	Pengambilan ke									
Ukuran File (MB)	1	2	3	4	5	6	7	8	9	10
16	15.608	15.581	16.043	15.749	15.763	15.936	15.944	15.95	15.879	15.597
32	29.538	29.406	29.555	29.75	30.051	29.544	29.714	29.585	29.828	30.165
64	59.708	59.137	59.249	59.614	59.694	59.482	59.508	59.639	59.189	59.593
128	118.147	118.243	118.001	118.17	118.183	117.665	117.405	118.174	118.211	118.316
256	233.822	234.55	234.205	234.191	233.489	234.349	234.351	234.264	234.363	234.299

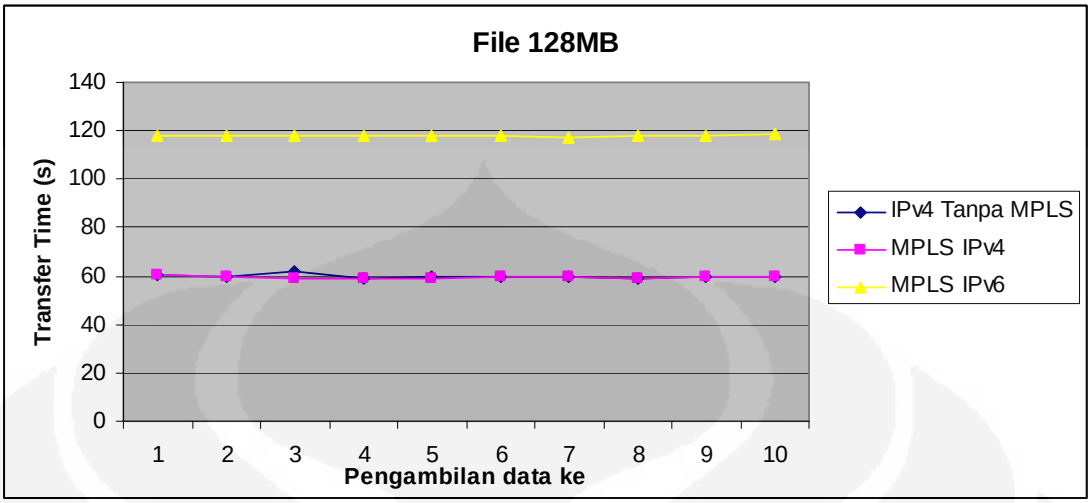
Throughput (Mbps)										
	Pengambilan ke									
Ukuran File (MB)	1	2	3	4	5	6	7	8	9	10
16	9.523	9.54	9.265	9.437	9.428	9.327	9.322	9.316	9.358	9.529
32	9.621	9.664	9.615	9.551	9.455	9.618	9.563	9.606	9.528	9.421
64	9.494	9.587	9.569	9.51	9.497	9.952	9.526	9.558	9.299	9.538
128	9.571	9.564	9.583	9.571	9.57	9.611	9.631	9.592	9.637	9.588
256	9.658	9.629	9.643	9.63	9.672	9.637	9.636	9.653	9.684	9.629

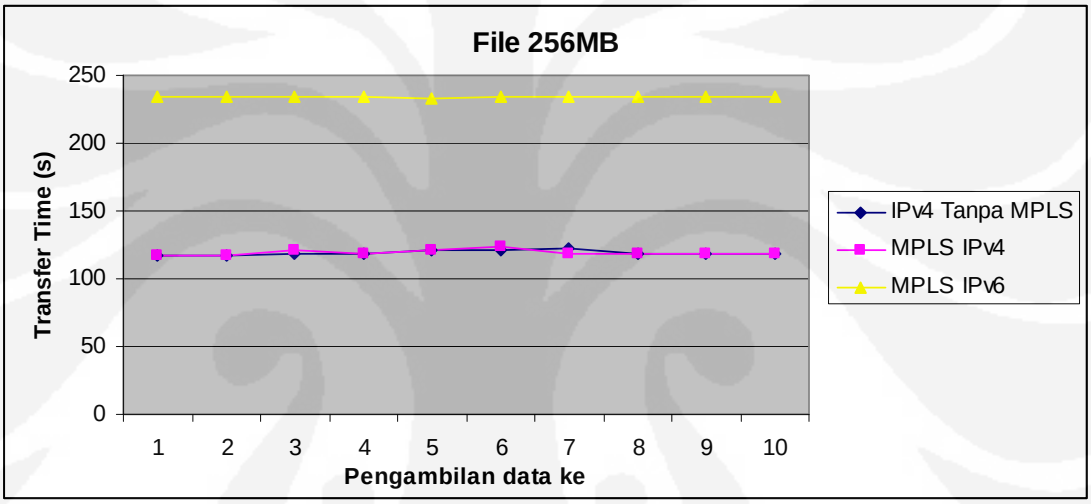
Lampiran 7
Grafik Perbandingan Transfer Time








File 128MB
Transfer Time (s)
140
120
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9
10
Pengambilan data ke
IPv4 Tanpa MPLS
MPLS IPv4
MPLS IPv6


File 256MB
Transfer Time (s)
250
200
150
100
50
0
1
2
3
4
5
6
7
8
9
10
Pengambilan data ke
IPv4 Tanpa MPLS
MPLS IPv4
MPLS IPv6