IMPLEMENTASI JARINGAN IPV6
PADA JARINGAN LOCAL AREA NETWORK (LAN)
UNIVERSITAS LAMPUNG DENGAN MEKANISME TUNNELING

ABSTRAK

Kelemahan dari mode pengalamatan *Internet Protocol* v4 (IPv4*)* adalah terbatasnya jumlah host yang dapat terhubung ke dalam jaringan, *Internet Protocol* v6 (IPv6*)* menawarkan fitur-fitur terbaru dalam teknologi *internet* seperti *real-time flows*, *provider selection*, *host mobility*, *end-to-end security*, dan *auto-reconfiguration.* Sebagai Universitas berbasis r*esearch*, Universitas Lampung memandang perlunya *backbone Local Area Network (LAN)* juga support IPv6 berjalan bersamaan dengan IPv4 yang sudah ada. Pada penelitian ini telah di rancang jaringan LAN kampus yang mengimplementasikan IPv6 pada *Border Gateway Protocol (BGP) router* melalui mekanisme tunneling, IPv6 juga diimplementasikan pada *web server, mail server, proxy server* dan aplikasi produksi lainnya.

Kata kunci : impelementasi IPv6, tunneling IPv6, IPv6 pada jaringan LAN

BAB I
PENDAHULUAN

1.1. Latar Belakang

Alasan utama untuk mulai beralih ke IPv6 adalah terbatasnya ruang pengalamatan *IPv4*. Saat ini bukan hanya komputer saja yang terhubung ke internet, namun peralatan sehari-hari seperti telepon seluler, *PDA*, *home appliances*, dan sebagainya juga terhubungkan ke internet, dapat dibayangkan berapa banyak alamat *IP* yang dibutuhkan untuk menghubungkan semua perangkat tersebut ke internet. Telah dikembangkan protokol jaringan baru, yaitu *IPv6* sebagai solusi dari masalah diatas. Protokol baru ini sudah banyak diimplementasikan pada jaringan-jaringan komputer besar dunia.

1.2. Tujuan

Tujuan dari penelitian ini adalah untuk mengimplementasikan protokol *IPv6* pada jaringan *LAN* Universitas Lampung termasuk aplikasi *client server* yang berjalan di dalamnya.

1.3. Permasalahan.

Dalam penelitian ini dirumuskan beberapa permasalahan yang dihadapi, yaitu :

1. Bagaimana merancang sebuah jaringan menggunakan protokol *IPv6* dengan menggunakan sumber daya jaringan yang sudah ada.
2. Bagaimana mengimplementasikan protokol *IPv6* ini dengan menggunakan *router base FreeBSD* dan protokol *routing BGP*.
3. Bagaimana mengimplementasikan *server* aplikasi *base on FreeBSD* yang sudah ada di Unit Pelayanan Teknis (UPT) Pusat Komputer Universitas Lampung, agar dapat berjalan dengan menggunakan protokol *IPv6*.

1.4. Metode Penelitian

1. Pengumpulan bahan-bahan referensi, meliputi referensi protokol IPv6, literature penelitian dari berbagai sumber.
2. Perancangan dan implementasi jaringan IPv6 di network UPT Puskom, selanjutnya men-*create* koneksi ke Hurricane Electric untuk mendapatkan koneksi IPv6 ke internet, mengaktifkan koneksi IPv6 ke internet dan membuat service yang ada di UPT Puskom mampu menggunakan protokol IPv6.
3. Penulisan laporan penelitian.

BAB II

LANDASAN TEORI

2.1. Dasar IPv6.

Standar IP yang saat ini digunakan pada kebanyakan jaringan adalah Internet Protocol Version 4 (IPv4). IPv4 dikembangkan pada awal tahun 1970 untuk memfasilitasi komunikasi dan pertukaran informasi antara peneliti di bidang pemerintahan dan bidang akademik.Pada waktu itu sistem yang ada terbatas, sehingga pengembang IPv4 tidak terlalu mementingkan variabel *security* dan *QoS*. Jumlah alamat yang tersedia pada saat itu juga dirasa sangat mencukupi yang mencapai 232 alamat.Saat ini jaringan komputer telah berkembang sangat pesat. Jumlah alamat pada IPv4 sudah tidak lagi mencukupi untuk memenuhi kebutuhan jaringanjaringan baru. Dukungan *security* dan *QoS* yang terintegrasi juga sangat dibutuhkan dalam kebanyakan konfigurasi jaringan dewasa ini. Untuk mengatasi kekurangan ini, IETF (*Internet Engineering Task Force*) pada tahun 1990 mulai mengembangkan Internet Protokol generasi baru yang dinamakan *Internet Protocol Version 6* (IPv6).

2.2. Terminologi IPv6.

Node

Peralatan yang mengimplementasikan IPv6.

Router

Node yang melewatkan paket IPv6.

Host

Node lainnya yang tidak merupakan *router*.

Upper-layer

Layer protocol yang secara langsung berada di atas IPv6. Sebagai contoh adalah protokol transport seperti TCP dan UDP, protokol control seperti ICMP, protokol *routing* seperti OSPF dan Internet atau protokol level bawah di*tunnel* melalui IPv6 seperti IPX, Appletalk, dan IPv6 sendiri (IPX over IPv6, Appletalk over IPv6 dan IPv6 over IPv6).

Link

Fasilitas komunikasi atau medium, yaitu *node* dapat berkomunikasi pada *layerlink*. *Layerlink* ini yang secara langsung dibawah *layer* IPv6. Sebagai contoh dari *link* adalah Ethernet (secara sederhana maupun menggunakan bridge); *link* PPP; X.25, Frame Relay, atau jaringan ATM, dan *layer* Internet *tunnel* seperti *tunnel* melalui IPv4 atau IPv6 sendiri.

Neighbors

Node lain yang dihubungkan dalam *link* yang sama

Interface

Media penghubung dari *node* (berada pada *node*) ke jaringan.

Address

Identifikasi pada *layer* IPv6 untuk *interface* atau sekumpulan *interface*.

Packet

Header IPv6 dan payload-nya (isi).

Link MTU

Maximum transmission unit. Ukuran maksimum paket dalam ukuran byte yang dapat disampaikan melalui *link*.

Path MTU

Link MTU yang paling kecil dari semua *link* dalam *pathnode* asal sampai *node* tujuan.

2.3. Format Header IPv6.

Format *header* alamat IPv6 menyederhanakan format *header* pada alamat IPv4. Perbandingan antara format *header* IPv6 (Gambar 2.1) dan IPv4 (Gambar 2.2).

Ver.	*header*	TOS	Total length	
Identification			flag	*Fragment* offset
TTL		Protokol	Checksum	
32 bit *Source Address*				
32 it *Destination Address*				

Gambar 2.1., Format Header IPv4

Ver.	TrafficClass	Flow Label	
Payload Length		*Next Header*	Hop Limit
128 bit *Source Address*			
128 bit *Destination Address*			

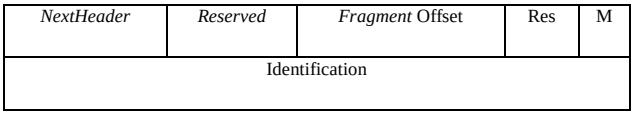
Gambar 2.2. Format header IPv6

Keterangan

Version	4-bit nomor versi *Internet Protocol* = 6.
Traffic Class	8-bit *field traffic class*.
Flow Label	20-bit *flow label*
Payload Length	16-bit *unsigned* integer. Panjang dari payload IPv6, sebagai contoh, keseluruhan paket tersebut mengikuti *header* IPv6 ini, dalam oktet. (Perlu diperhatikan bahwa *header* ekstensi manapun yang ada merupakan bagian dari payload, termasuk dalam jumlah panjangnya)
Next Header	8-bit selector. Mengidentifikasi tipe *header* yang langsung mengikuti *header* IPv6. Menggunakan nilai yang sama seperti *field* protokol IPv4.
Hop Limit	8-bit *unsigned* integer. Dikurangi dengan 1 oleh setiap *node* yang meneruskan paket.
SourceAddress	128-bit alamat asal dari paket.
DestinationAddress	128-bit alamat penerima yang dituju dari paket (bisa jadi bukan penerima terakhir, jika terdapat *headerrouting*)

2.4. Fragment Header.

HeaderFragment digunakan oleh suatu *source* IPv6 untuk mengirim suatu paket yang lebih besar dari yang akan berada pada *path* MTU ke *Destination*-nya. (Catatan: tidak seperti IPv4, *Fragment*asi dalam IPv6 hanya dilakukan oleh *node source*, bukan oleh *router* sepanjang delivery *path* dari suatu paket) *HeaderFragment* diidentifikasi dengan nilai 44 *NextHeader* pada tepat sebelum *header*, dan memiliki format sebagai berikut :



Gambar 2.3.Format FragmentHeader

Keterangan gambar:

Next Header	Selektor 8-bit. Mengidentifikasi inisial tipe *headerFragment*able Part dari paket asli/awal (yang didefinisikan sebelumnya). Menggunakan nilai yang sama dengan *field* IPv4 Protocol.
Reserved	*Fieldreserved* 8-bit. Diinisialisasi dengan nol untuk transmisi; diabaikan pada penerimaan.
Fragment Offset	13-bit *unsigned* integer. Offset, dalam satuan 8-oktet, dari data yang mengikuti/setelah *header* ini, relatif pada awal *Fragment*able Part dari paket mula-mula.
Res	*Fieldreserved* 2-bit. Diinisialisasi dengan nol untuk transmisi; diabaikan pada penerimaan.
M flag	1 = more *Fragment*s; 0 = last *Fragment*.
Identification	32 bit. Lihat deskripsi di atas.

Untuk mengirim suatu paket yang terlalu besar yang sesuai dengan *path* MTU ke *Destination*-nya, suatu *nodesource* boleh membagi paket tersebut menjadi *Fragment-Fragment* dan mengirim masing-masing *Fragment* sebagai paket terpisah, kemudian disatukan kembali pada penerima, Untuk setiap paket yang akan dipecah-pecah, *node* asal harus membangkitkan nilai yang digunakan untuk mengidentifikasi paket yang dipecah tersebut. Dalam satu paket yang dipecah-pecah tersebut harus mempunyai nilai identifikasi yang berbeda. Jika *headerrouting* terdapat pada paket, tujuan alamat dikonsentrasikan pada tujuan terakhir.

2.5. Ars*ite*ktur Pengalamatan IPv6.

Tipe-tipe pengalamatan pada IPv6 :

a. ***Unicast***

Pengidentifikasi untuk *interface* tunggal. Paket yang dikirimkan ke alamat unicast adalah paket yang dikirimkan ke sebuah *interface* yang diidentifikasi oleh alamat tersebut.

b. ***Anycast***

Pengidentifikasi untuk sekumpulan *interface* (umumnya milik *node* yang berbeda). Paket yang dikirimkan ke alamat *anycast* adalah paket yang dikirimkan ke salah satu dari sekumpulan *interface* yang diidentifikasi oleh alamat tersebut (alamat yang paling dekat, mengacu pada pengukuran jarak dari protokol *routing*).

c. ***Multicast***

Pengidentifikasi untuk sekumpulan *interface* (umumnya milik *node* yang berbeda). Paket yang dikirimkan ke alamat *multicast* adalah paket yang dikirimkan ke semua *interface* yang diidentifikasi oleh alamat tersebut.Tidak ada alamat broadcast dalam IPv6, fungsi alamat broadcast digantikan oleh alamat *multicast*.

2.6. Model Pengalamatan IPv6

Alamat-alamat IPv6 dari semua tipe diberikan pada *interface*, tidak pada *node*. Alamat *unicast* IPv6 mengacu pada *interface* tunggal. Karena setiap *interface* milik *node* tunggal, alamat *unicast* yang diberikan pada *node* tersebit juga digunakan untuk mengidentifikasi *node* tersebut.Semua *interface* diharuskan untuk mempunyai setidaknya satu alamat *unicastlink-local*. Satu buah *interface* dapat diberikan atau dialokasikan alamat IPv6 lebih dari satu dengan berbagai macam tipe alamat atau *scope.* Alamat *unicast* dengan *scope* lebih besar dari *link-scope* tidak diperlukan untuk *interface* yang tidak digunakan sebagai alamat asal atau tujuan dari paket IPv6. Hal ini

kadang-kadang tepat untuk *interface* point-to-point, atau dalam bentuk *link* point-to-point, tidak perlu adanya pemberian alamat *unicast* pada kedua *interface* tersebut. Ada satu pengecualian pada model pengalamatan ini, yaitu alamat *unicast* atau sekumpulan ala,at *unicast* mungkin diberikan ke *interface* fisik yang banyak jika implementasi tersebut menganggap *interface* yang banyak tersebut sebagai satu kesatuan *interface* ketika dihadapkan pada *layer* internet. Hal ini sangat berguna untuk load-sharing melalui *interface* fisik yang banyak.Saat ini IPv6 melanjutkan model IPv4 dimana prefix subnet diasosiasikan dengan satu *link* (*link* tunggal). Prefix subnet yang mungkin diberikan pada *link* yang sama dapat lebih dari satu.

2.7. Representasi Teks dari Alamat.

Ada tiga jenis bentuk konversional untuk merepresentasikan alamat IPv6 sebagai string teks :

1. Bentuk umum adalah x:x:x:x:x:x:x:x, x adalah nilai heksadesimal dari 8 satuan yang mana setiap satuan terdiri atas 16 bit

   Contoh :

   FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

   1080:0:0:0:8:800:200C:417A

   Catatan : Tidak perlu menulis permulaan nilai nol dalam setiap kolom (dipisahkan dengan tanda ":"), misalkan 0008 cukup dapat ditulis 8 saja. Namun, setidaknya harus ada satu dalam setiap kolom jika semuanya berupa 0.

2. Ada beberapa metode dalam pengalokasian gaya tertentu dari alamat IPv6, hal ini khususnya untuk alamat yang berisi string nol bit yang panjang. Dalam rangka untuk membuat mudah penulisan alamat yang berisi bit nol, special sintaks tersedia untuk memadatkan kumpulan dari tiap-tiap nilai nol sepanjang 16 bit yng berurutan. Tanda "::" hanya dapat tampil sekali dalam sebuah alamat. Tanda "::" juga dapat digunakan untuk memadatkan kumpulan nilai 16 bit yang terdapat pada awal alamat.

   Contoh :

   1080:0:0:0:8:800:200C:417A alamat *unicast*

   FF01:0:0:0:0:0:0:101 alamat *multicast*

   0:0:0:0:0:0:0:1 alamat *loopback*

0:0:0:0:0:0:0:0	alamat tak terdefinisi mungkin direpresentasikan menjadi:
1080::8:800:200C:417A	alamat *unicast*
FF01::101	alamat *multicast*
::1	alamat *loopback*
::	alamat tak terdefinisi

3. Bentuk alternative yang kadang-kadang lebih tepat ketika dihadapkan dengan lingkungan gabungan dari IPv4 dan IPv6 adalah x:x:x:x:x:x:d.d.d.d dimana x menandakan nilai heksadesimal dari enam satuan yang masing-masing terdiri atas 16 bit, dan d adalah nilai decimal dari empat satuan yang masing-masing terdiri dari 7 bit (standar representasi IPv4).

   Contoh :

   0:0:0:0:0:0:202.154.63.9

   0:0:0:0:0:FFFF:10.122.1.77

   atau dalam bentuk dipadatkan :

   ::202.154.63.9

   ::FFFF:10.122.1.77

2.8. Representasi Tipe Alamat

Tipe spesifik dari alamat IPv6 diindikasikan dengan bit-bit awal yang berada dalam alamat. Pangjang bit-bit awal yang bervariabel ini disebut format prefix (FP). Inisialisasi alokasi dari prefix-prefix ini ada sebagai berikut:

Alokasi	Prefix (biner)	Fraction of *Address* Space
Reserved	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP Allocation	0000 001	1/128
Reserved for IPX Allocation	0000 010	1/128
Unassigned	0000 011	1/128
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Aggregatable global *UnicastAddress*	001	1/8
Unassigned	010	1/8
Unassigned	011	1/8
Unassigned	100	1/8
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link-local *UnicastAddress*	1111 1110 10	1/1024
Site-local *UnicastAddress*	1111 1110 11	1/1024
MulticastAddress	1111 1111	1/256

Gambar 2.4. Inisialisasi Alamat IPv6

Catatan:

1. "unspecified *Address*" di atas adalah alamat *loopback* dan alamat IPv6 yang digabungkan dengan alamat IPv4. Alamat-alamat tersebut diberikan alokasi dengan spasi format prefix 0000 0000.
2. Format prefix-prefix dari 001 sampai 111, kecuali alamat *multicast* (111 1111), semuanya distaratkan harus mempunyai pengudentifikasi *interface* 64 bit dalam format EUI-64 (panjang prefix maksumum adalah 64 bit dan 64 bit selanjutnya adalah pengidentifikasi *interface*).

Alamat *unicast* dibedakan dari alamat *multicast* dengan nilai octet berorder tinggi dalam alamat. Nilai FF (1111 1111) mengidentifikasikan alamat sebagai alamat *multicast*, nilai lain dari alamat adalah alamat *unicast*. Alamat *anycast* diambil dari spasi alamat *unicast*, dan secara sintaks tidak berbeda dari alamat *unicast*.

2.9. Alamat *Unicast*

Alamat *Unicast* merupakan alamat yang dapat diagregasi dengan masking pada bit-bit seperti pada alamat IPv4 di bawah CIDR (Class-less InterDomain *Routing*). Ada beberapa bentuk pemberian alamat *unicast* dakam IPv6, termasuk alamat *unicast* global teragregatisasi, alamat NSAP, alamat hierarki IPX, alamat *site*-local, alamat *link*-local, dan alamat dengan gabungan IPv4. Tipe alamat tambahan dapat didefinisikan pada masa depan. *Node* IPv6 boleh mempunyai sedikit pengetahuan dari struktur internal dari alamat IPv6, tergantung pada peran tersebut dilaksanakan (misal di *host* atau *router*). Pada minimumnya, *node* boleh menganggap bahwa alamat *unicast* tidak mempunyai internal struktur:

2.9.1. Pengidentifikasi *Interface*

Pengidentifikasi *interface* dalam alamat *unicast* IPv6 digunakan untuk mengidentifikasi *interface* pada *link*. Mereka diharuskan untuk unik pada *link* tersebut. Mereka mungkin juga unik pada cope yang besar. Pada banyak kasus pengidentifikasi *interface* akan sama dengan alamat *interface* pada *layerlink* atau *layer* 2. Pengidentifikasi *interface* yang sama mungkin digunakan pada *interface-interface* pada *node* tunggal.Dalam sejumlah format prefix, *interface* id disyaratkan mempunyai panjang 64 bit dan disusun dalam format IEEE EUI-64. Pnegidentifikasi *interface* berbasis EUI-64 boleh mempunyai scope global ketika terdapat token global (misalnya MAC, Media Access Control, IEEE 48 bit atau biasa disebut alamat MAC) atau boleh mempunyai scope local ketika tidak terdapat token global (seperti serial *linktunnel* end-point).

2.9.2. Alamat Tak Terspesifikasi

Alamat 0:0:0:0:0:0:0:0 disebut alamat yang tak terspesifikasi. Alamat tersebut tidak boleh diberikan pada setiap *node*. Alamat tersebut mengidentifikasikan kehadiran alamat. Salah satu contoh penggunaannya adalah di dalam kolom alamat asal dari setiap paket IPv6 yang dikirim oleh *host* penginisialisasi sebelum *host* tersebut telah mempelajari alamatnya sendiri.Alamat yang tak terspesifikasikan tidak boleh digunakan sebagai alamat tujuan dari paket IPv6 atau dalam *headerrouting*.

2.9.3. Alamat *Loopback*

Alamat *unicast* 0:0:0:0:0:0:0:1 disebut alamat *loopback*. Alamat tersebut mungkin digunakan oleh *node* untuk mengirimkan paket IPv6 ke dirinya sendiri. Alamat tersebut tidak pernah diberikan pada setiap *interface* fisik. Hal ini mungkin dihubungkan dengan intergace virtual (misalkan *interfaceloopback*).Alamat *loopback* tidak boleh digunakan sebagai alamat asal dalam paket IPv6 yang dikirimkan keluar *host*. Paket IPv6 dengan alamat tujuan *loopback* tidak boleh dikirim keluar dari *host* dan tidak dapat diforwardkan/dilewatkan oleh *router* IPv6.

2.10. Alamat *Anycast*

Alamat *Anycast* IPv6 adalah alamat yang diberikan kepada lebih dari satu *interface* (biasanya milik *node* yang berbeda), paket yang dikirim ke alamat *anycast* dirutekan ke *interface* terdekat yang mempunyai alamat tersebut, mengacu pada pengukuran jarak dari protokol *routing*.Alamat *anycast* dialokasikan dari spasi alamat *unicast*, menggunakan berbagai macam definisi format alamat berbeda dengan alamat *unicast*. Ketika alamat *unicast* berubah ke alamat *anycast*, *node* yang diberikan alamat tersebut harus secara eksplisit tahu bahwa alamat tersebut adalah alamat *anycast.* Satu yang diharapkan dari penggunaan alamat-alamat *anycast* adalah untuk mengidentifikasi sekumpulan *router* yang menjadi milik organisasi yang menyediakan layanan Internet. Alamat tersebut dapat digunakan sebagai alamat pertengahan dalam *headerrouting* IPv6, untuk menyebabkan paket dikirim melalui agregasi khusus atau urutan untuk mengidentifikasi sekumpulan *router* yang terkoneksi dalam subnet tertentu atau mengidentifikasi sekumpulan *router* yang menyediakan entri ke dalam domain *routing* tertentu.

Ada beberapa larangan yang diberikan pada alamat *anycast*:

a. Alamat *anycast* harus tidak digunakan sebagai alamat asal dari paket IPv6.
b. Alamat *anycast* harus tidak diberikan kepada *host* IPv6 dengan demikian mungkin diberikan ke *router* IPv6 saja

2.11. Alamat *Multicast*

Alamat *multicast* IPv6 adalah pengidentifikasi untuk sekumpulan *node*. *Node* dapat menhadi anggora sejumlah group *multicast*. Alamat *multicast* mempunyai format :

11111111 pada awal alamat tersebut menunjukkan jika alamat tersebut adalah alamat multicst. Flags adalah sekumpulan dari 4 bit flags yang mempunyai format berikut

Tiga urutan terdepan dicadangkan, dan harus diberi nilai nol.

T=0 mengidentifikasikan alamat*multicast* yang diberikan secara permanen (“well-known”). Pemberian alamat ini dilakukan ileh badan Internet global yang berwenang pada masalah penomoran

T=1 menandakan alamat *multicast* yang diberikan secara non-permanen(transien)

Scope adalah nilai jangkuan*multicast* sepanjang 4 bit digunakan untuk membatasi jangkauan group *multicast*.

Nilai ini adalah:

1. 0 dicadangkan
2. 1 jangkaunan *node*-lokal
3. 2. Jangkuan *link* local
4. 3 belum diberikan
5. 4 belum diberikan
6. 5 Jangkauan *Site* local
7. 6 belum diberikan
8. 7 belum diberikan
9. 8 Jangkuan organizaziton-local
10. 9 Belum diberikan
11. A belum diberikan
12. B belum diberikan
13. C Belum diberikan
14. D Belum diberikan
15. E Jangkuan global
16. F dicadangkan

Group ID mengidentifikasi group *multicast*, apakah diberikan secara permanen atau transien, dalam jangkuan yang diberikan.

Maksud dari alamat *multicast* yang diberikan secara permanen adalah nilai jangkuan yang independen. Sebagai contoh, jika group server NTP diberikan alamat *multicast* permanen dengan ID group 101 (dalam heksa), maka:

FF01:0:0:0:0:0:0:101 berarti semua server NTP pada *node* yang sama sebagai pengirim.
FF02:0:0:0:0:0:0:101 berarti semua server NTP pada *link* yang sama sebagai pengirim.
FF05:0:0:0:0:0:0:101 berarti semua server NTP pada *site* yang sama sebagai pengirim.
FF0E:0:0:0:0:0:0:101berarti semua server NTP dalam internet.

Alamat *multicast* yang diberikan secara nonpermanen berarti hanya dalam jangkauan yang diberikan. Sebagai contoh, group yang diidentifikasikan oleh alamat *multicastsite*-local nonpermanen FF15:0:0:0:0:0:0:101 pada suatu *site* tidak berhubungan dengan group yang menggunakan alamat yang sama pada *site* yang berbeda, atau juga tidak berhubungan terhadap group non permanen yang berada pada jangkauan yang berbeda, atau juga tidak berhubungan dengan group permanen dengan group ID yang sama.Alamat *multicast* harus tidak digunakan sebagai alamat asal dalam paket IPv6 atau terdapat dalam semua *headerrouting*.

Berikut ini adalah alamat *multicast* well-konwn yang sudah didefinisikan :

*ReservedMulticastAddress*es:

FF00:0:0:0:0:0:0:0
FF01:0:0:0:0:0:0:0
FF02:0:0:0:0:0:0:0
FF03:0:0:0:0:0:0:0
FF04:0:0:0:0:0:0:0
FF05:0:0:0:0:0:0:0
FF06:0:0:0:0:0:0:0
FF07:0:0:0:0:0:0:0
FF08:0:0:0:0:0:0:0
FF09:0:0:0:0:0:0:0
FF0A:0:0:0:0:0:0:0
FF0B:0:0:0:0:0:0:0
FF0C:0:0:0:0:0:0:0
FF0D:0:0:0:0:0:0:0
FF0E:0:0:0:0:0:0:0
FF0F:0:0:0:0:0:0:0

Alamat *multicast* ini dicadangkan dan tidak boleh diberikan pada semua group *multicast*

All *Node*s *Address*es: **FF01:0:0:0:0:0:0:1**

FF02:0:0:0:0:0:0:1

Alamat *multicast* ini mengidentifikasi group dari semua *node* IPv6, dalam scope 1 (*node*-local) atau 2 (*link*-local).

All *Router*s *Address*es: **FF01:0:0:0:0:0:0:2**

FF02:0:0:0:0:0:0:2

Alamat *multicast* ini mengidentifikasi group dari semua *router*, dalam scope 1 (*node*-local), atau 2 (*link*-local) atau 5(*site*-local)

DHCP Server/Relay-Agent: **FF02:0:0:0:0:0:0:C**

Alamat *multicast* ini mengidentifikasi groups dari semua IPv6 DHCP server dan Relay Agent, dalam scope 2 (*link*-local)

Solicited-NodeAddress: **FF02:0:0:0:0:1:FFXX:XXXX**

Solicited-nodeAddress adalah alamat yang diperhitungkan sebagai fungsi alamat *unicast* dan *anycastnode*. Alamat ini dibentuk dengan mengambil 24 bit terakhir dari alamat (*unicast* atau *anycast*) dan menambahkan 24 bit terakhir yang diambil tersebut pada prefix **FF02:0:0:0:0:1:FF00::/104**.

Prefix **FF02:0:0:0:0:1:FF00::/104** ini menghasilkan alamat *multicast* dalam jangkauan **FF02:0:0:0:0:1:FF00:0000** sampai **FF02:0:0:0:0:1:FFFF:FFFF**

Sebagai contoh, alamat *multicast* model ini yang berhubungan dengan alamat IPv6 **2001:200:830::200E:8C6C** adalah **FF02::1:FF0E:8C6C**

2.11.1. Alamat yang Diperlukan *Node*

Host diharuskan untuk mengenali alamat-alamat berikut sebagai perngidentifikasi untuk diri sendiri:

a. Alamat *link*-local untuk setiap *interface*
b. Alamat *unicast* yang diberikan
c. Alamat *loopback*
d. Alamat *multicastsolicited-node* untuk setiap alamat *unicast* dan *anycast* yang diberikan
e. Alamat *multicast* dari semua group tempat *host* menjadi anggota

Router diharuskan untuk mengenali semua alamat seperti yang diperlukan pada *host*, ditambah alamat-alamat berikut sebagai pengidentifikasi untuk diri sendiri:

a. Alamat subnet *anycastrouter* untuk *interface-interface* yang dikonfigurasi untuk bertindak sebagai *router*.
b. Semua alamat *anycast* lain yang sudah dikonfigurasi pada *router*.
c. Alamat *multicast* semua *router*
d. Alamat *multicast* dari semua group lain tempat *router* menjadi salah satu anggotanya

Alamat prefix-prefix yang harus didefinisikan sebelumnya pada implementasi IPv6 adalah:

a. Unspecified *Address*
b. Alamat *loopback*
c. Prefix *Multicast* (FF)
d. Prefix local-use (*link*-local dan *site*-local)
e. Pre-defined *multicastAddress*es
f. IPv4-Compatible prefix

Implementasi harus mengasumsikan semua alamat sebagai *unicast* kecuali ada konfigurasi yang sudah dispesifikasikan.

2.12. *Neighbor Discovery* (ND) untuk *IPv6*.

Node (*host* dan *router*) menggunakan ND untuk mencari alamat-alamat *layerlink* (misalkan alamat MAC pada ethernet) guna mengetahui *node-node* tetangga yang berada pada *link* yang sama dan secara cepat menghapus alamat-alamat tersebut pada cache jika alamat tersebut sudah tidak berlaku lagi. *Host-host* juga menggunakan ND untuk mencari *router-router* yang menjadi tetangganya yang bersedia melewatkan paket-paketnya untuk kepentingan *host* itu sendiri. Dan juga, *node-node* menggunakan protokol ini untuk mendeteksi perubahan pada alamat *layerlink* (*layer* 2). Ketika *router* atau jalur ke *router* gagal/rudak, *host* akan secara aktif mencari alternatif yang aktif.Alamat *multicast* untuk semua *node* adalah FF02::1, yang merupakan jangkauan alamat *link-link* untuk dapat mencapai semua *node*. Sedangkan alamat *multicast* untuk semua *router* adalah FF02::2, yang merupakan jangkauan alamat *link*-local untuk mencapai semua *router*.

2.12.1. *Overview Protocol.*

Protokol ini memecahkan sejumlah masalah sebelumnya yang berhubungan dengan interaksi antara *node-node* yang terhubung dalam *link* yang sama. Protokol ini mendefinisikan mekanisme untuk memecahkan setiap masalah berikut:

1. *Router Discovery*, Bagaimana *host-host* mencari *router* yang berkoneksi pada *link*
2. Prefix *Discovery*, Bagaimana *host-host* menemukan alamat prefix yang merupakan alamat atau pengidentifikasi *link* tempat *host-host* tersut saling terinterkoneksi (*node-node* menggunakan prefix untuk membedakan apakah *node* yang akan tersbut beradapada *link* yang sama dengan *node* asal atau *node* yang akan dituju tersebut yang hanya dapat dijangkau melalui *router*).
3. Parameter *Discovery*, Bagaimana *node* mempelajari parameter-parameter pada *link* seperti *link* MTU atau parameter-parameter internet seperti jumlah batasan hop yang akan ditempatkan pada peket yang akan dikirim.
4. *Address* Autoconfiguration, Bagaimana *node-node* secara otomatis mengkonfigurasi alamat IPv6 untuk *interface*nya.
2. *Address* Resolution, Bagaimana *node-node* mencari alamat *linklayer* dari *node* yang akan dituju yang masih berada pada *link* yang sama (misalnya *node* tetangga) hanya dengan diberikan alamat IP *node* tujugannya saja.
3. *Next*-Hop determination, Algoritma untuk memerakan alamat IPv6 dari *node* tujuan ke dalam alamat IPv6 *node* tetangga. Trafik untuk *node* tujuan tersebut akan dikirimkan ke *node* tetangga tersbut. *Next*-Hop ini dapat berupa *router* atau *host* tujuan itu sendiri (bergantung pada ke mana trafik itu akan dikirim, jika ke tujuan yang masih dalam satu *link* yang sama maka *next* hop adalah *node* tujuan itu sendiri, jika tujuannya sudah berbeda *link*/prefix maka *next* hop tersbut adalah *router*).
4. Neighbor Unreachability Detection, Bagaimana *node* mempelajari bahwa salah saru tetangga sudah tidak aktif lagi. Untuk *node* tetangga yang digunakan sebagai *router, node* tersebut dapat mencoba rute default alternatif. Untuk kedua *router* dan *host*, *Address* resolution dapat dilakukan kembali.
5. Duplicate *Address* Detection, Bagaimana *node* mempelajari bahwa alamat yang ingin digunakan sedang tidak digunakan oleh *node* lain.

6. Redirect, Bagaimana *router* memberitahu *host* tentang *node* pertama mana yang baik sebagai *next* hop untuk mencapai tujuan tertentu.

Neighbor *Discovery* mendefinisikan lima tipe paket ICMP yang berbeda, yaitu sepasang yang terdiri atas *router* solocotation dan *router* advertisement messages, sepasang yang terdiri atas Neighbor Solicitation dan Neighbor Advertisement Messages dan terakhir pesan Redirect. Pesan-pesan tersebut melayani beberapa tujuan berikut:

a. ***Router* Solocitation**: ketika sebuah *interface* pada *host* menjadi diaktifkan, *host-host* boleh mengirimkan keluar *router* solicitation yang meminta *router* untuk mengirimkan *router* advertisement sesegera mungkin ke *host* tersbut.
b. ***Router*Advertisement** : *router* mengumumkan keberadaan mereka dengan berbagai macam *link* dan parameter internet secara periodik atau dalam merespon pesan *router* solicitation. *Router* advertisement ini dikirim kepada *host-host* yang berada di sekitar *router*. *Router* advertisement berisi prefix-prefix yang digunakan untuk pengenalan/pencarian *link* dan/atau untuk konfigurasi alamat, nilai perkiraan batasan hop dan sebagainya.
c. **Neighbor Solicitation :** dikirim oleh *node* untuk mencari alamat *layerlinknode* tetangga-tetangganya, atau untuk memeriksa apakah tetangga tersbut masih dapat dijangkau dengan alamat *layerlink* yang berada dalam memori cache-nya atau tidak Neighbor Solicitation juga digunakan untuk mendeteksi alamat yang ganda.
d. **Neighbor Advertisement** adalah sebuah respon kepada pesan neighbor solicitation. *Node* boleh juga mengirimkan neighbor advertisemnt tanpa didahului oleh neighbor solicitation yang digunakan untuk mengumumkan perubahan alamat *layerlink* pada tetangga tersbut.
e. **Redirect** Digunakan oelh *router-router* untuk menginformasikan *host-host* tentang hop pertama yang paling baik untuk sebuah tujuan tertentu.

Dengan kemampuan *multicast* pada *link*, setiap *router* secara periodik mengirimkan secara *multicast* paket *router* advertisement yang mengumumkan keberadaanya. *Host* menerima *router* advertisement dari semua *router*, dan kemudian membuat daftar *router* default. *Router* mengirimkan pesan *router* advertisement ini cukup sering sehingga *host* dapat mempelajari keberadaannya mereka dalam beberapa menut namun juga tidak cukup sering untuk mendeteksi adanya kesalahan pada *router* dengan menggunakan tidak adanya *router* advertisement yang

diterima. Algoritma deteksi ketidakterjangkauannya tetangga ini dibuat terpisah dan algoritma ini menyediakan deteksi kesalahan.

Router advertisement berisi daftar dari prefix-prefix yang digunakan untuk penentuan alamat *link* yang digunakan dan/atau konfigurasi alamat pada *host* secara mandiri. *Host* menggunakan prefix-prefix yang diumumkan oleh *router* untuk membuat dan memelihara daftar yang akan digunakan untuk memutuskan apakah paket tujuan ini masih dalam *link* yang sama atau harus dijangkau dengan *router*. Perlu diingat bahwa tujuan dapat juga benda dalam *link* yang sama meskipun tudak diikutsertakan dalam pengumuman prefix-prefix yang berada pada saru *link* yang sama. Pada kasus ini *router* dapat mengirimkan pesan redirect yang menginformasikan pengirim bahwa tujuan redirect yang menginformasikan pengirim bahwa tujuan tersebut adalah masih metupakan tetangga.

Router Advertisement mengizinkan *router-router* untk menginformasikan *host-host* di bawahnya bagaimana melaukan konfigurasi alamat otomatis. Sebagai contoh *router* dapat menspesifikasikan apakah *host-host* tersbut sebaiknya menggunakan konfigurasi alamat secara stateful (DHCPv6) dan/atau secara autonomous/sendiri (stateless). Pesan *router* advertisement juga berisi parameter-parameter internet seperti batas hop yang *host-host* sebaiknya gunakan dalam paket yang akan ditransmisikan dan secara opsional dapat berisi pula parameter seperti *link* MTU. Fasilitas ini memusatkan administrasi dari parameter-parameter yang penting yang dapat diset di *router-router* dan secara otomatis dipropagasikan ke semua *host-host* yang terkoneksi di bawah *router*.

Node melaksanakan resolusi/pencarian alamat dengan mengirimkan balik alamat *layerlink* nya. Pesan neighbor solicitation adalah pesan yang di*multicast*kan (dikirim ke group/kelompok *node* tertentu) ke alamat *multicast* tertentu tempat *node* yang merupakan target tersebut menjadi salah satu anggotanya. *Node* yang nerupakan target mengirimkan alamat *layerlink* nya ke *node* yang meminta dalam bentuk pesan neighbor advertisement yang *unicast*. Satu pasang tanya-jawab dari paket ini sudah cukup untuk kedua *node* yang melakukan tanya-jawab untuk saling mengetahui alamat *linklayer*. Hal ini karena *node* yang merupakan inisiator juga mengirimkan alamat *layerlink* dalam pesan neighbor solicitaion. Pesan neighbor solicitation dapat juga digunakan untuk mempelajari jika lebih dari satu *node* dialokasikan alamat *unicast* yang sama.

Neighbor Unreachability Detection mendeteksi adanya masalah pada *node* tetangga atau masalah pada jalur pelewatan paket ke tetangga. Metode ini memerlukan konfirmasi positif yang menyatakan bahwa paket yang dikirimkan ke tetangga telah benar-benar sampai dan telah diproses dengan baik oleh *layer* IP. Neighbor Unreachability Detection menggunakan konfirmasi dari dua sumber. Ketika memungkinkan, protokol pada *layer* yang lebih tinggi menyediakan konfirmasi positif yang menyatakan bahwa koneksi telah mencapai kemajuan yang berarti data yang dikirimkan sebelumnya telah sampai dengan benar. Ketika knfirmasi positif tidak menunjukkan tanda-tanda kedatangannya , *node* mengirimkan pesan *unicast* neighbor solicitation yang meminta neighbor advertisement sebagai konfirmasi keterjangkauan *node* yang dituju pada *node* yang menhadi hop berikutnya. Untuk mengurangi trafik jaringan yang tidak perlu, pesan untuk penyelidikan hanya dikirim ke tetangga tempat *node* secara aktif mengirimkan paket-paket.

Sebagai tambahan terhadap masalah umum di atas, Neighbor *Discovery* juga menangani situasi berikut ini:

Perubahan alamat - *link-layernode* yang tahu alamat *link-layer*nya berubah dapat me*multicast*kan pesan dengan mengirimkan beberapa paket Neighbor Advertisement ke semua *node* untuk secara cepat mengupdate alamat *link-layer* dalam memori cache yang sidah tidak berlaku lagi.

Inbound load balancing – *node* yang mempunyai *interface* ganda ataupun lebih dari satu mungkin ingin melakukan load balancing penerimaan paket-paket yang dapat melalui *interface* jaringan yang lebih dari satu pada *link* yang sama. Seperti *node* yang mempunyai alamat *link-layer* yang lebih dari satu yang diberikan pada *interface* yang sama. Sebagai contoh dari driver jaringan yang tunggal (software) dapat merepresentasikan card *interface* jaringan yang lebih dari satu sebagai *interface* logikal tunggal yang mempunyai alamat *layerlink* yang lebih dari satu. Load balancing ditangani oleh *router* dengan menghilangkan alamat *link-layer* asal dari paket *Router* Advertisement, dengan demikian memaksa tetangga untuk menggunkan pesan Neighbor Solicitation untuk mempelajari alamat *layerlinkrouter*. Pesan neighbor advertisement yang kembali dapat berisi alamat *layerlink* yang berbeda.

Alamat *anycast* – alamat *anycast* mengidentifikasi satu dari sekumpulan nide yang menyediakan layanan yang sama, dan *node-node* pada *link* yang sama dapat dikonfigurasi untuk mengenal

alamat *anycast* dengan menerima Neighbor Advertisement lebih dari satu untuk target yang sama. Semua advertisement yang diterima untuk alamat *anycast* ditandai sebagai advertisement yang tidak dapat ditimpa. Dengan mengguakan peraturan-peraturan yang spesifik maka akan dipelajari advertisement yang mana yang akan digunakan.

2.13. Transmisi Paket *IPv6* melalui *Ethernet*

2.13.1. *MTU(maximum Transmission Unit)*.

Ukuran MTU secara default untuk paket IPv6 pada ethernet adalah 1500 oktet. Ukuran ini mungkin dikurangi oleh *router* advertisement yang berisi pilihan MTU yang menspesifikasikan MTU yang lebih kecil, atau oleh konfigurasi manual setiap *node* nya. Jika *router* advertisement diterima pada *interface* yang mempunyai pilihan MTU yang menspesifikasikan MTU lebih besar dari 1500, atau lebih besar dari nilai yang dikonfigurasi secara manual, pilihan MTU tersebut akan dicatat pada manajemen sistem dan harus diabaikan

2.13.2. *FormatFrame.*

Paket IPv6 ditransmisikan dalam frame standar ethernet. *Header* ethernet berisi alamat ethernet tujuan dan asal serta berisi kode tipe ethernet, dimana harus berisi alamat hexadesimal dengan nilai 86DD. Kolom data berisi *header* IPv6 diikuti secara langsung oleh payload dan oktet padding untuk memenuhi ukuran frame minimum pada *link* ehternet.

2.13.3. *Stateless Auto configuration.*

Node yang pertama kali tersambung ke jaringan akan secara otomatis mengkonfigurasi alamat IPv6 *site*-local dan global tanpa memerlukan manual konfigurasi atau bantuan dari server seperti server *DHCP(Dynamic Host Configuration Protocol).* Dengan IPv6, *router* akan mengirimkan pesan *router* advertisement yang berisi prefix global dan *site*-local.Pesan *router* advertisement ini akan dikirim secara periodik atau dapat dikirm sewaktu-waktu untuk merespon pesan *router* solicitation yang dikirim oleh *host* pada waktu startup sistem. Alamat global IPv6 dibentuk oleh prefix yang diberikan oleh pesan *router* advertisement dan pengidentifikasi unterface (EUI-64). Alamat prefix IPv6 digunakan untuk stateless autoconfiguration dari *interface* erhernet harus mempunyai panjang 64 bit.Pengidentifikasi *interface* untuk *interface* ethernet berbasis pada pengidentifikasi EUI-64 yang diperoleh dari alamat *interface* IEEE 802 dengan pangjang 48 bit yang sudah ada pada tiap *interface* etheternet. Deskripsi EUI-64 sebagai berikut:

Tiga oktet pertama dari lamat ehternet menjadi perusahaan ID dari EUI-64 (tiga oktet pertama). Oktet keempat dan kelima dari EUI diset tetap dengan nilai FFFE dalam hexadesimal. Tiga oktet terakhir dari alamat ethernet menjadi tiga alamat terakhir dari EUI-64.

Pengidentifikasi *interface* kemudia dibentuk dari EUI-64 dengan mengkomplemenkan bit "Universal/Local" (U/L), yang mana merupakan bit order terendah dari oktet pertama EUI-64. Mengkomplemenkan hal ini akan berubah dari nilai 0 menjadi 1, karena alamat yang sudah terdapat pada *interface* diharapkan dari alamat yang dialokasikan secara universal dan mempunyai nilai yang unik. Alamat IEEE 802 yang diaokasikan secara universal atau EUI-64 ditandai dengan 0 pada posisi U/L bitnya, sedangkan pengidentifikasi *interface* IPV6 secra global ditandai dengan nilai 1 pada posisi yang sama.

Sebagai contoh, pengidentifikasi *interface* untuk *interface* ethernet yang mempunyai alamat yang sudah berada didalamnya, dalam hexadesimal:
34-56-78-9A-BC-DE akan menjadi **36-56-78-FF-FE-9A-BC-DE**.

2.13.4. Alamat *link*-local

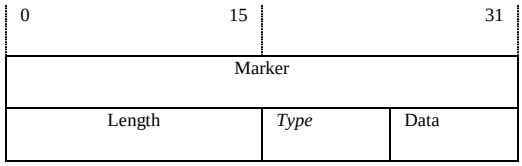
Alamat *link* local IPv6 untuk *interface* ethernet dibentuk oleh dengan menambahkan pengidentifikasi *interface* dengan prefix FE80::/64.

2.14. *Protokol Routing* BGP4

Border Gateway Protokol adalah *routing* protokol yang memakai system autonomous.Fungsi utama dari BGP adalah untuk saling tukar-menukar informasi konektivitas jaringan antar BGP sistem. Informasi konektifitas ini antara lain adalah daftar dari Autonomous System (AS). Informasi ini digunakan untuk membuat daftar *routing* sehingga terjadi suatu koneksi.BGP4 mampu melakukan suatu advertaisement dan IP-prefix serta menghilangkan keterbatasan tentang network class.BGP memakai pola *Hop-by-Hop* yang artinya hanya meggunakan jalur yang berikutnya yang terdaftar dalam Autonomous System.BGP menggunakan TCP sebagai media transport. BGP menggunakan port 179 untuk koneksi BGP.BGP mendukung CIDR.

BGP mampu mempelajari jalur internet malalui internal atau eksternal BGP dan dapat memilih jalur terbaik dan memasukkannya dalam ip forwarding.

BGP dapat digunakan pada dual maupun multi-homed, dengan syarat memiliki nilai AS. BGP tidak dapat digunakan pada single-homed.



Gambar 2.5. Format BGP Header

Type dari BGP:

Open, tipe pesan yang diterima sewaktu koneksi antar BGP tersambungkan.

Update, tipe pesan yang dikirimkan untuk mengirimkan informasi *routing* antar BGP.

Keepalive, tipe pesan yang dikirimkan untuk mengetahui apakah pasangan BGP masih hidup

Notification, tipe pesan yang dikirimkan apabila terjadi error.

Atribut yang dimiliki oleh BGP:

AS_*path*, adalah jalur yang dilalui dan dicatat dalam data BGP route, dan dapat mendeteksi loop.

*Next*_Hop, adalah jalur berikutnya yang akan dilalui dalam *routing* BGP, biasanya adalah local network dalam eBGP. Selain itu bisa didapat dari iBGP.

Local Preference, penanda untuk AS BGP local

Multi-Exit Discriminator (MED), bersifat non-transitif digunakan apabila memiliki eBGP yang lebih dari 1.

Community, adalah sekumpulan BGP yang berada dalam satu AS.

Perbandingan BGP-4 antara yang digunakan untuk IPv4 dan IPv6 adalah kemampuan dari BGP yang dapat mengenali *scope* dari IPv6, yaitu *global, site-local, link-local*. Apabila IPv6 masih menggunakan IPv4 sebagai transport maka alamat *peer* pada BGP yang lainnya harus diikutkan pada konfigurasi.

2.15. Interoperabilitas IPv4 dan IPv6.

Sebelum *IPv4* sepenuhnya digantikan dengan *IPv6* dibutuhkan suatumekanisme transisi yang mempermudah interoperabilitas antara *IPv4* dan *IPv6*.Mekasisme transisi sangat dibutuhkan karena tidak mungkin untuk menggantikanseluruh alamat IPv4 dalam waktu singkat.Ada banyak metode transisi yang dapatdigunakan dalam masa transisi yaitu dual stack, tunneling, dan translation.

1. Dual stack

Dual stack memungkinkan satu antar muka menggunakan IPv6 dan IPv4 secara bersama-sama. Jika komunikasi menggunakan IPv4 maka antar muka akan bertindak antar muka IPv4 murni, dan jika komunikasi menggunakan IPv6 maka antar muka akan bertindak sebagai antar muka IPv6 murni.

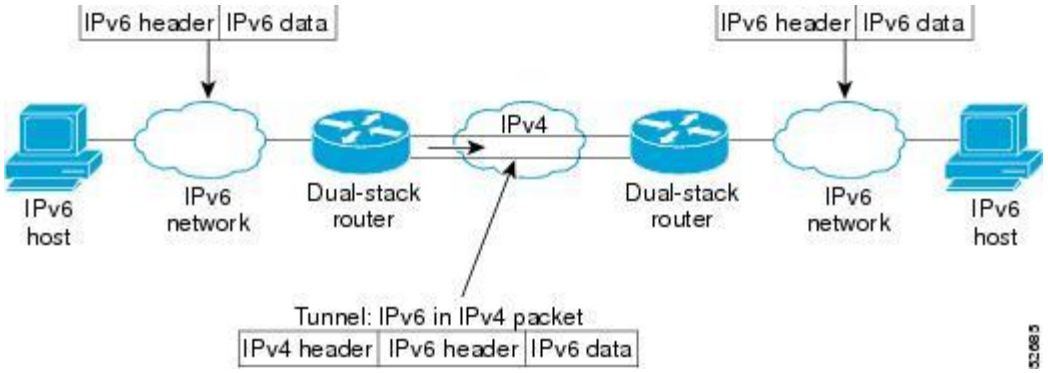
2. Tunneling

Tunneling digunakan untuk membangun jaringan IPv6 dengan memanfaatkan infrastruktur jaringan IPv4 yang sudah ada.Tunneling sering juga disebut enkapsulasi. Dengan metode ini protokol IPv6 akan dienkapsusi pada protokol IPv4. Paket yang terenkapsulasi ini kemudian di teruskan melalui jaringan IPv4 melalui infrastruktur jaringan IPv4.

Proses enkapsulasi terdiri dari beberapa bagian yaitu:

- Enkapsulasi pada tunnel entry point
- Dekapsulasi pada tunnel exit point
- Manajemen tunnel

Overlay Tunnels IPv6



Gambar 2.6. Overlay Tunnel

Pada gambar diatas dijelaskan bahwa Paket Data IPv6 dienkapsulasi diatas jaringan IPv4

3. Translation

Metode translation didefinisikan pada RFC 2765 dan 2766.Metode translasi dapat digunakan untuk menerjemahkan paket dari jaringan IPv6 sehingga dapat diterima pada jaringan IPv4 dansebaliknya.

BAB III
HASIL PENELITIAN

5.1. Kesiapan Infrastruktur IPv6 Universitas Lampung.

Jaringan yang ada di Universitas Lampung saat ini masih menggunakan protokol IPv4. Infrastruktur Jaringan yang dimiliki Unila antara lain:

1. *Rapier Allied Telesis*versi 8xxx dan 9xxx dengan modul Fiber Optik dan Ethernet yang tersebar di beberapa jurusan di Unila.
2. Switch 3Com Gigabit tersebar di beberapa jurusan di Unila.
3. *FreeBSD* server yang memberikan layanan berupa dns server, mail server, web server, proxy server, dan ftp server.
4. *Linux* server dengan fungsi layanan aplikasi SIAKAD.

Backbone Jaringan LAN berada di gedung MISTC Rektorat Lantai 3 dan menggunakan layanan ISP dari PT. MORATELINDO via *Fiber Optic* sebesar 155Mbps yang diterminasi di BBS Unilanet Gedung Puskom baru lantai 2. LAN Unila dibedakan menjadi beberapa *VLAN* untuk mempermudah mengelola jaringan yang ada. Secara umum seluruh infratruktur sudah full support *IPv6*.

5.2. Koneksi ke IPv6 Global.

Spesifikasi perangkat keras dan perangkat lunak pada gateway terluar Unila;

- Server menggunakan IBM X Series 3450.
- Sistem Operasi router adalah FreeBSD versi 8.1 release.
- Aplikasi routing menggunakan Quagga Suite Application.

Pada penelitian ini penulis memanfaatkan layanan gratis BGP Tunneling dari Hurricane Electric untuk membuat jalur tunnel ke IPv6 Global, dengan terlebih dahulu meng-*assign* blok prefix IPv6 Unila di APNIC. Langkah pertama yang dilakukan adalah dengan membuat account baru di http://tunnelbroker.net, selanjutnya mengcreate menu BGP Regular tunnel, memasukkan parameter prefix IPv6, dan membuat surat permohonan resmi dari Institusi untuk pengaktifan BGP Peer ke Hurricane Electric. Setelah proses aktifasi BGP Tunnel selesai, selanjutnya aktifkan Tunnel kearah HE pada server gateway utama dengan parameter sebagai berikut;.

```
ifconfig gif0 create
 ifconfig gif0 tunnel 103.3.46.254 216.218.221.2
 ifconfig gif0 inet6 2001:470:17:9::2 2001:470:17:9::1 prefixlen 128
 route -n add -inet6 default 2001:470:17:9::1
 ifconfig gif0 up
```

Gambar 3.1. Parameter Tunneling dari Tunnel Broker

Agar parameter ini dapat secara permanen tersimpan pada Gateway utama, modifikasi file **/etc/rc.conf**menjadi seperti berikut;

```
##Options IPv6
 ipv6_enable="YES"
 ipv6_network_interfaces="auto"
 ipv6_defaultrouter="2001:470:17:9::1
 ipv6_router_enable="YES"
 gif_interfaces="gif0"
 gifconfig_gif0="103.3.46.254 216.218.221.2"
 ipv6_ifconfig_gif0="2001:470:17:9::2 2001:470:17:9::1 prefixlen 128"
```

Gambar 3.2. Parameter Permanent Start-up IPv6 Tunnel

Konfigurasi BGP Peering dengan Hurricane Electric

```
!
 hostname INTL-global-gw-POP1-unila
 !
 router bgp 56237
 bgp router-id 27.50.31.178
 network 103.3.46.0/24
 neighbor 27.50.31.177 remote-as 23947
  neighbor 2001:470:17:9::1 remote-as 6939
  neighbor 2001:470:17:9::1 description UNILA-HE
  neighbor 2001:470:17:9::1 update-source 2001:470:17:9::2
  neighbor 2001:470:17:9::1 remove-private-AS
 !
  address-family ipv6
  neighbor 2001:470:17:9::1 activate
 exit-address-family
 !
 line vty
```

Gambar 3.3. Parameter BGP Peer ke HE

Pada konfigurasi diatas Unila melakukan BGP Peering dengan jaringan Hurricane Electric menggunakan metode AS Peering. AS atau Autonomous System adalah sebuah string unik sepanjang 2^{16} bit sebagai pengenal atau identifikasi jaringan dari sebuah organisasi. Nomor AS bisa didapatkan dari Regional Internet Registry (RIR), dalam hal ini untuk kawasan Asia Pacific

ditangani oleh APNIC, Universitas Lampung saat ini terdaftar dengan AS Number 56237, record detail AS Unila pada APNIC adalah sebagai berikut.

```
aut-num:        AS56237
as-name:        UNILA-AS-ID
descr:          Universitas Lampung
descr:          University / Direct Member IDNIC
descr:          Bandar Lampung, Lampung 35145
country:        ID
admin-c:        MK670-AP
tech-c:         MK670-AP
import:         from AS23947 action pref=100; accept ANY
export:         to AS23947 announce AS56237
notify:         hostmaster@idnic.net
mnt-by:         MNT-APJII-ID
mnt-irt:        IRT-UNILA-ID
mnt-routes:     MAINT-ID-UNILA
remarks:        Send Spam and Abuse report to : abuse@unila.ac.id
changed:        hostmaster@idnic.net 20110519
source:         APNIC

role:           APNIC Hostmaster
address:        6 Cordelia Street
address:        South Brisbane
address:        QLD 4101
country:        AU
phone:          +61 7 3858 3100
fax-no:         +61 7 3858 3199
e-mail:         helpdesk@apnic.net
admin-c:        AMS11-AP
tech-c:         AH256-AP
nic-hdl:        HM20-AP
remarks:        Administrator for APNIC
notify:         noc@apnic.net
mnt-by:         MAINT-APNIC-AP
changed:        hm-changed@apnic.net 19981111
changed:        dbmon@apnic.net 19990702
changed:        hm-changed@apnic.net 20020211
changed:        hm-changed@apnic.net 20070612
changed:        hm-changed@apnic.net 20100217
changed:        hm-changed@apnic.net 20101217
changed:        hm-changed@apnic.net 20110815
source:         APNIC

person:         Muhammad  Komarudin
address:        Kampus UNILA
address:        Jl.Prof.SumantriB roionegoroN o.1,Gedong Meneng
address:        Bandarlampung, Lampung 35145
country:        ID
phone:          +62-721-701609
fax-no:         +62-721-702767
e-mail:         hostmaster@unila.ac.id
nic-hdl:        MK670-AP
mnt-by:         MNT-APJII-ID
changed:        hostmaster@idnic.net 20110325
```

```
INTL-global-gw-POP1-unila# show bgp neighbors 2001:470:17:9::1
 BGP neighbor is 2001:470:17:9::1, remote AS 6939, local AS 56237, ex
 Description: UNILA-HE
 BGP version 4, remote router ID 72.52.92.170
 BGP state = Established, up for 00:33:12
 Last read 17:31:02, hold time is 180, keepalive interval is 60 secon
 Neighbor capabilities:
 4 Byte AS: advertised
 Route refresh: advertised and received(old & new)
 Address family IPv4 Unicast: advertised
 Address family IPv6 Unicast: advertised and receivedUpdate source is
 Private AS number removed from updates to this neighbor
 Community attribute sent to this neighbor(both)
 0 accepted prefixes
 For address family: IPv6 Unicast
 Community attribute sent to this neighbor(both)
 6051 accepted prefixes
 Connections established 1; dropped 0
 Last reset never
 Local host: 2001:470:17:9::2, Local port: 53517
 Foreign host: 2001:470:17:9::1, Foreign port: 179
 INTL-global-gw-POP1-unila#
```

Gambar 3.4. Show BGP Neighbor

Gambar diatas adalah parameter pengecekan apakah Peer BGP neighbour sudah aktif ataukah belum, dari hasil pengetesan dengan perintah show bgp neighbor terlihat bahwa peer kearah HE sudah established yang artinya link IPv6 sudah berfungsi dan siap dipergunakan.

```
INTL-global-gw-POP1-unila# show bgp ipv6 unicast statistics
 BGP IPv6 Unicast RIB statisticsTotal Advertisements          :
 Total Prefixes                :         6052
 Average prefix length         :        38.18
 Unaggregateable prefixes      :         4091
 Maximum aggregateable prefixes:         1961
 BGP Aggregate advertisements  :          279
 Address space advertised      :  15023908850
 %% announced : 1502390886400.00
 /8 equivalent :       895.49
 /24 equivalent :  58687144.00
 Advertisements with paths     :         6052
 Longest AS-Path (hops)        :           13
 Average AS-Path length (hops) :         3.03
 Largest AS-Path (bytes)       :           54
 Average AS-Path size (bytes)  :        14.14
 Highest public ASN            :       393246
 INTL-global-gw-POP1-unila#
```

Gambar 3.5. Show BGP Statistic prefix

Gambar diatas menunjukkan advertise IPv6 unicast yang didapat dari Announce Peer HE dengan total Prefix IPv6 sebanyak 6052 prefix.

Hasil pengecekan link dan path menuju ke Global IPv6

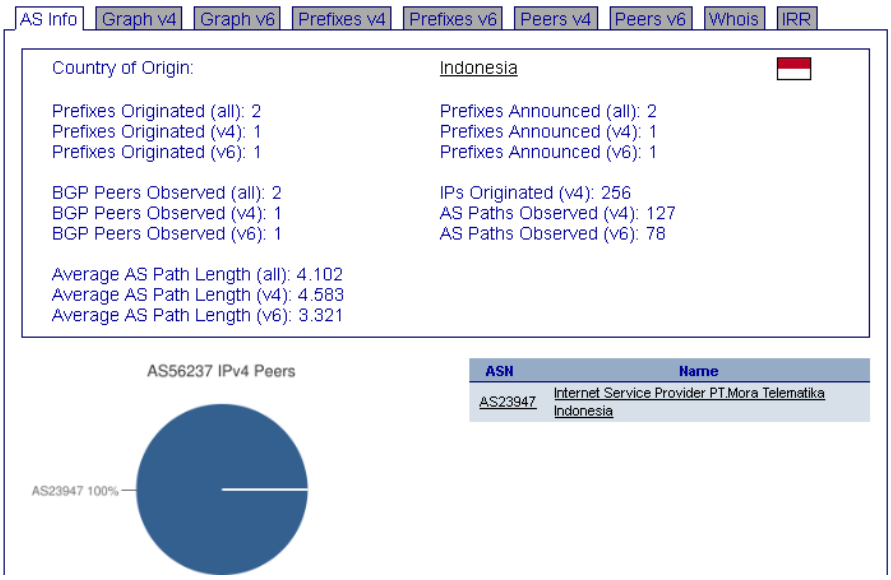
```
INTL-global-gw-POP1-unila-Console# ping6 ipv6.google.com
PING6(56=40+8+8 bytes) 2001:470:17:9::2 --> 2404:6800:800b::68
 16 bytes from 2404:6800:800b::68, icmp_seq=0 hlim=56 time=184.737 ms
 16 bytes from 2404:6800:800b::68, icmp_seq=1 hlim=56 time=176.618 ms
 16 bytes from 2404:6800:800b::68, icmp_seq=2 hlim=56 time=185.147 ms
 16 bytes from 2404:6800:800b::68, icmp_seq=3 hlim=56 time=191.974 ms
 ^C
 --- ipv6.l.google.com ping6 statistics ---
 4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 176.618/184.619/191.974/5.441 ms

INTL-global-gw-POP1-unila-Console# traceroute6 ipv6.google.com
traceroute6 to ipv6.l.google.com (2404:6800:800b::68) from
2001:470:17:9::2, 64 hops max, 12 byte packets
 1  donovanp-2.tunnel.tserv19.hkg1.ipv6.he.net  160.901 ms  167.632 ms
172.882 ms
 2  tserv19.hkg1.ipv6.he.net  176.392 ms  158.802 ms  157.613 ms
 3  google3-10G.hkix.net  151.873 ms  148.500 ms  144.507 ms
 4  2001:4860::1:0:16  146.547 ms
 2001:4860::1:0:1063  148.803 ms  163.786 ms
 5  2001:4860::1:0:3c0  243.801 ms  182.216 ms  177.280 ms
 6  2001:4860::2:0:3c6  180.306 ms  186.157 ms  186.558 ms
 7  2001:4860:0:1::257  184.410 ms  190.957 ms  201.653 ms
 8  2404:6800:800b::68  199.736 ms  198.316 ms  191.021 ms
INTL-global-gw-POP1-unila-Console#
```

Gambar 3.6. Show ping6 statistic ke google

Dari gambar diatas terlihat bahwa dari Gateway Utama telah terhubung dengan jaringan global IPv6.

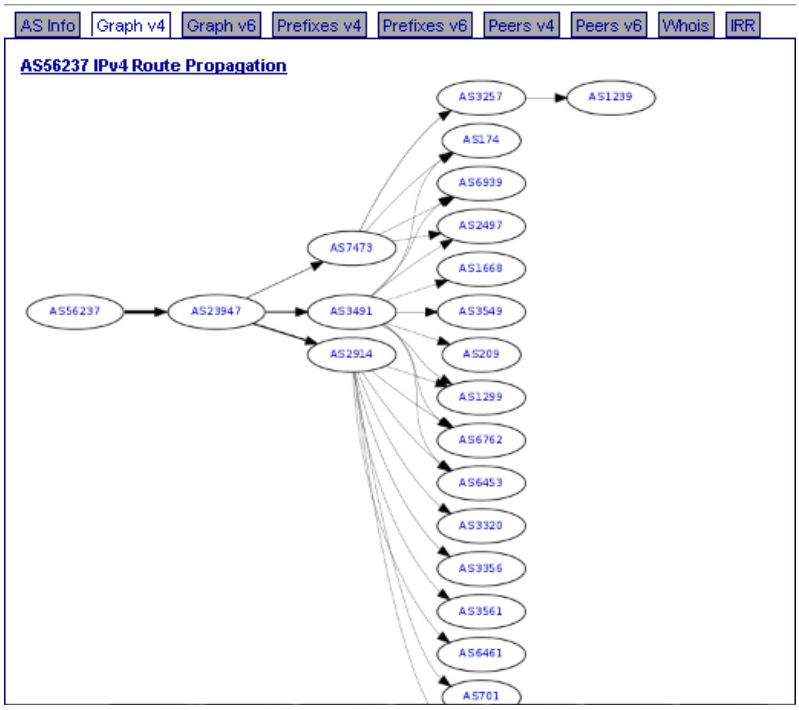
5.2.1 HasilBGP Check Online Tool dari Hurricane Electric

ASN Info



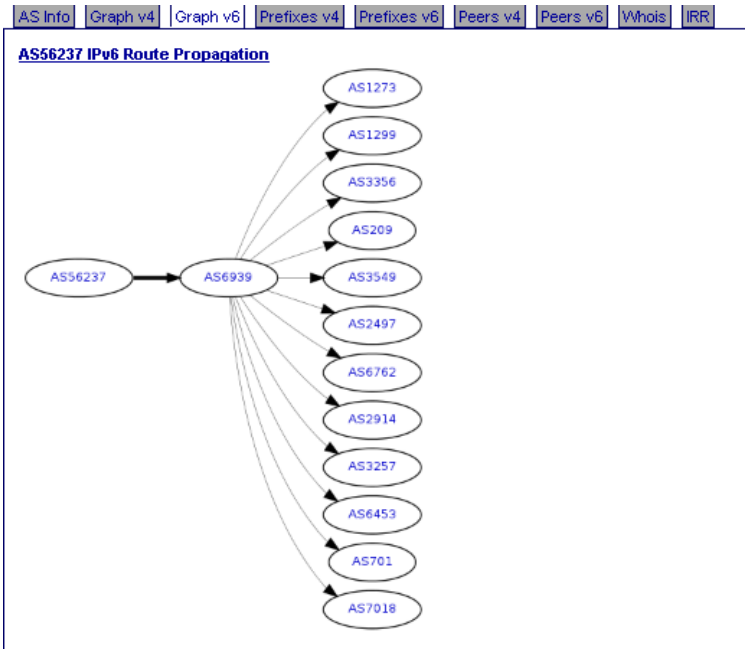
Gambar 3.7 ASN Info

Graph IPv4



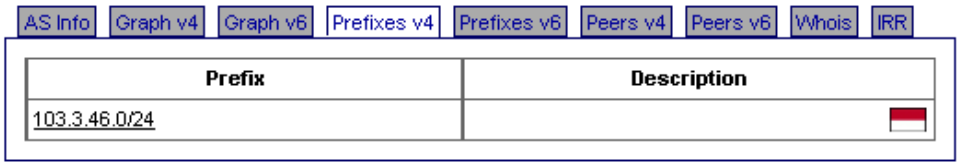
Gambar 3.8 IPv4 Route Propagation

Graph IPv6



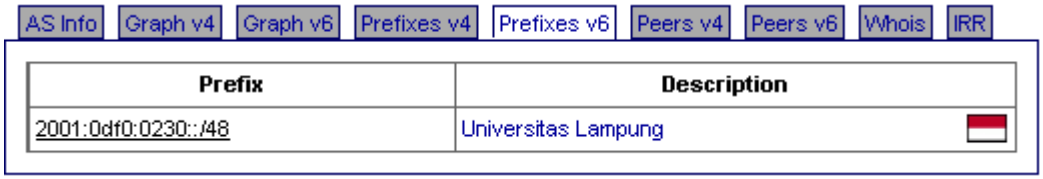
Gambar 3.9 IPv6 Route Propagation

Graph Prefix IPv4



Gambar 3.10 Prefix IPv4

Graph Prefix IPv6



Gambar 3.11 Prefix IPv6

5.3. Migrasi ke IPv6

***Router* Advertisement**

Untuk menyebarkan IPv6 yang didapat, UPT Puskom menggunakan model *router* advertisement. *Router* Advertisement yang digunakan adalah RAdvD dan Zebra. Konfigurasi yang perlu dilakukan adalah, menambahkan konfigurasi pada file */usr/local/etc/radvd*.conf untuk *RadvD* dan */usr/local/etc/zebra/zebra.conf* untuk menggunakan Zebra. ***/usr/local/etc/radvd*.conf**

```
interface eth0
{
 AdvSendAdvert on;
Prefix 2001:470:17:9::/64


 {
 AdvOnLink on;
 AdvAutonomous on;
 };
```

Gambar 3.12. //usr/local/etc/radvd.conf

/usr/local/etc/quagga/zebra.conf

```
interface eth0
 no ipv6 nd suppress-ra
ipv6 nd prefix-advertisement 2001:470:17:9::/64
2592000 604800 onlink autoconfig
```

Gambar 3.13. /usr/local/etc/quagga/zebra.conf

5.4. Domain Name Server (DNS).

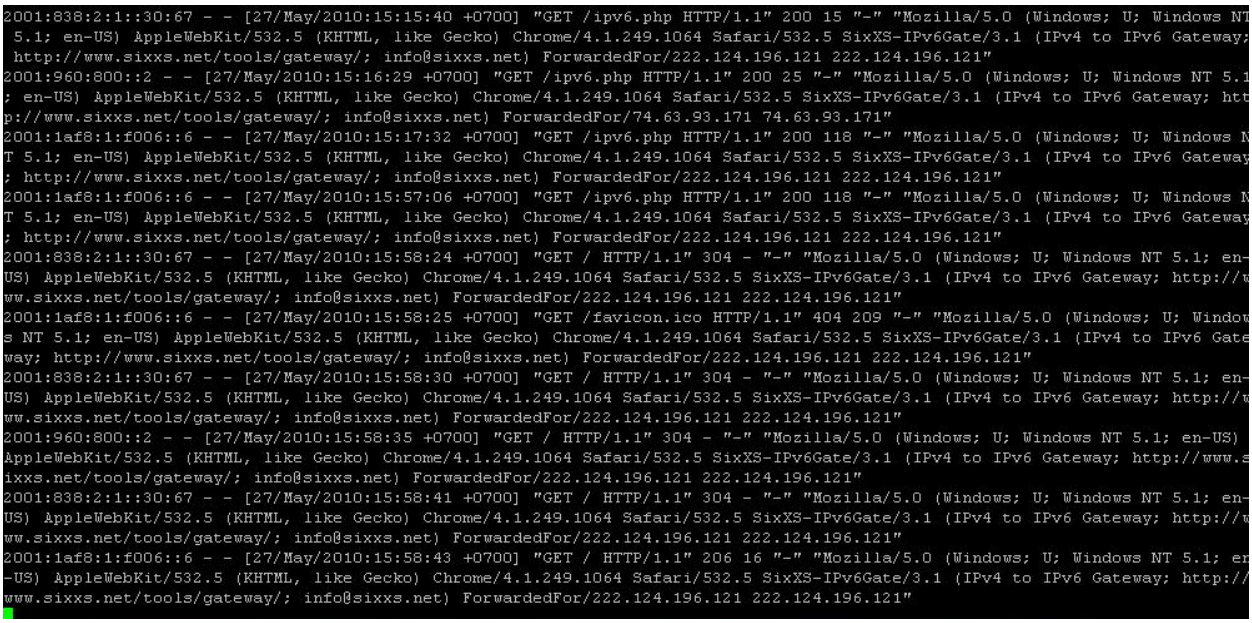
Universitas Lampung saat ini memiliki 3 name server public yaitu ns1.unila.ac.id (103.3.46.2) ns2.unila.ac.id (103.3.46.3) , ns3.unila.ac.id (103.3.46.4), dengan layanan utama menghandle zone unila.ac.id berikut reverse Pointer nya, ditambah lagi dengan Name Server secondary yang di hosting di server ns1.he.net.

```
$TTL    3600

          @      IN      SOA     ns1.unila.ac.id. admin.unila.ac.id.  (

                        2011080590 ; serial
                        3600       ; Refresh
                        900        ; Retry
                        3600000    ; Expire
                        3600 )     ; Minimum
     IN      NS                      ns1.unila.ac.id.
     IN      NS                      ns2.unila.ac.id.
     IN      NS                      ns1.he.net..
     IN      MX 10                   barracuda.unila.ac.id.
     IN      A                       103.3.46.1
     IN      MX 20                   zimbra.unila.ac.id.
     IN      AAAA                    2001:470:17:9::3
     ns1     IN      A               103.3.46.2
             IN      MX 10           barracuda.unila.ac.id.
           IN      AAAA              2001:470:17:9::4

     ns2     IN      A               103.3.46.2
             IN      MX 300          barracuda.unila.ac.id.

     ns3     IN      A               103.3.46.4
           IN      MX 300            barracuda.unila.ac.id.
www  IN   AAAA                       2001:470:17:9::3
barracuda  IN   AAAA                2001:470:17:9::5
```

Gambar 3.14. zone.unila.ac.id

5.5. Web Server

Apache adalah server web handal dan paling banyak digunakan oleh pada administrator yang menggunakan system operasi unix. Walaupun banyak digunakan pada system operasi unix, Apache ini juga dapat digunakan pada system operasi Windows NT/9x, 2000, Netware 5.x dan OS/2. Selain handal, Apache adalahserver web fleksibel dan mengimplementasikan protokol-protokol web terbaru seperti HTTP/1.1 (RFC 2616).Salah satu sebab kenapa Apache banyak digunakan karena sifat dari software Apache sendiri yang open *source* dan tidak menggunakan lisensi dalam pemakaian software tersebut.Secara Default Apache Web server pada server www.unila.ac.id telah enable IPv6, pengecekan dilakukan dengan membaca access.log pada server



Gambar 3.15. Apache Log

Pada gambar diatas terlihat bahwa web server telah melayani request www dari source dengan alamat IPv6.

5.6. Mail Server

Mail Server Unila menggunakan software MTA Postfix, Postfix adalah mail transfer agent yang dikembangkan oleh Wietse Venema. Terdiri beberapa program kecil yang sangat handal. Postfix dijalankan dengan proteksi bertingkat, oleh program-program kecil yang saling tidak percaya. Masing-masing program dijalankan oleh user khusus (bukan setuid). Multiple Transport. Postfix dapat mengirim surat dengan modus SMTP (Simple Mail Transfer Protocol) dan UUCP (Unix to

Unix Copy Protocol) sekaligus.Mendukung format Maildir, secara default Postfix terbaru sudah support IPv6 terlihat dari Full Header email berikut

Return-Path: < ipv6@he.net>
X-Original-To: gigih@groups.unila.ac.id
Delivered-To: gigih@groups.unila.ac.id
Received: from ns1.unila.ac.id (unknown [192.168.1.8])
by groups.unila.ac.id (Postfix) with ESMTP id E34902281A
for< gigih@groups.unila.ac.id>; Tue, 1 Jun 2011 10:47:58 +0700 (WIT)
Received: from maiser.unila.ac.id (unknown [192.168.1.2])
by ns1.unila.ac.id (Postfix) with ESMTP id 0F02933DC5
for< gigih@groups.unila.ac.id>; Tue, 1 Jun 2011 11:19:40 +0700 (WIT)
Received: from ns1.unila.ac.id (unknown [192.168.1.8])
by maiser.unila.ac.id (Postfix) with ESMTP id 3F37439836
for< gigih@unila.ac.id>; Tue, 1 Jun 2011 10:41:22 +0700 (WIT)
Received: from ipv6.he.net (ipv6.he.net [IPv6:2001:470:0:64::2])
by ns1.unila.ac.id (Postfix) with ESMTP id 7407333C42
for< gigih@unila.ac.id>; Tue, 1 Jun 2011 11:19:37 +0700 (WIT)
To:
From: ipv6@he.net
Message-ID: < 4c0488be37b2b.1275365566@ipv6.he.net>
Subject: IPv6 Certification Mail Test
Date: Mon, 31 May 2011 21:12:46 -0700

Gambar 3.16. Mail Header IPv6 Ready

5.7. Proxy Server

Http *proxy server* yang digunakan adalah Squid (http://www.squid-cache.org/). Squid adalah server proxy cache dengan performa tinggi untuk web client, mendukung akan FTP, Gopher, dan data object HTTP. Meng-cache internet object adalah menyimpan data hasil request ke internet bisa dalam bentuk FTP, HTTP, HTTPS di system yang lebih dekat dari yang meminta request daripada ke sumber. Kemudian web browser dapat menggunakan squid dalam local area network sebagai server proxy HTTP, dan dapat menghemat waktu dan *bandwidth*

Instalasi.

```
unila-inherent-gtw# cd /usr/ports/www/squid31/
 unila-inherent-gtw# make config
 ┌──────────────────────────────────────────────┐
 │                Options for squid 3.1.14                │
 │ ┌──────────────────────────────────────────┐ │
 │ │[X] SQUID_KERB_AUTH      Install Kerberos authentication helpers
 │ │[X] SQUID_LDAP_AUTH      Install LDAP authentication helpers
 │ │[X] SQUID_NIS_AUTH       Install NIS/YP authentication helpers
 │ │[ ]  SQUID_SASL_AUTH      Install SASL authentication helpers
 │ │[X] SQUID_IPV6           Enable IPv6 support
 │ │[X] SQUID_DELAY_POOLS    Enable delay pools
 │ │[X] SQUID_SNMP           Enable SNMP support
 │ │[ ] SQUID_SSL            Enable SSL support for reverse proxies
 │ │[ ] SQUID_PINGER         Install the icmp helper
```

Gambar 3.17. Squid config

```
unila-inherent-gtw# make install
 ===>  Vulnerability check disabled, database not found
 ===>  License GPLv2 accepted by the user
 ===>  Found saved configuration for squid-3.1.14
 ===>  Extracting for squid-3.1.14
 => SHA256 Checksum OK for squid3.1/squid-3.1.14.tar.bz2.
 ===>   squid-3.1.14 depends on file: /usr/local/bin/perl5.8.8 - found
 ^C
```

Gambar 3.18. Squid Install

```
# Squid normally listens  port defaultnya adalah 3128
        http_port 3128

# Definiskan host IPv4 yang boleh menggunakan layan proxy ini

      acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
      acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
      acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
     http_access allow localnet

  # Buat acl baru yang membolehkan tujuan ke IPv6

       acl to_ipv6 dst ipv6

# Definisikan TCP Outgoing jika proxy server ingin digunakan Dual IPv4/IP
        tcp_outgoing_address 2001:470:18:aa7::2  to_ipv6
        tcp_outgoing_address 103.3.46.254  !to_ipv6
```

Gambar 3.19. Squid.conf

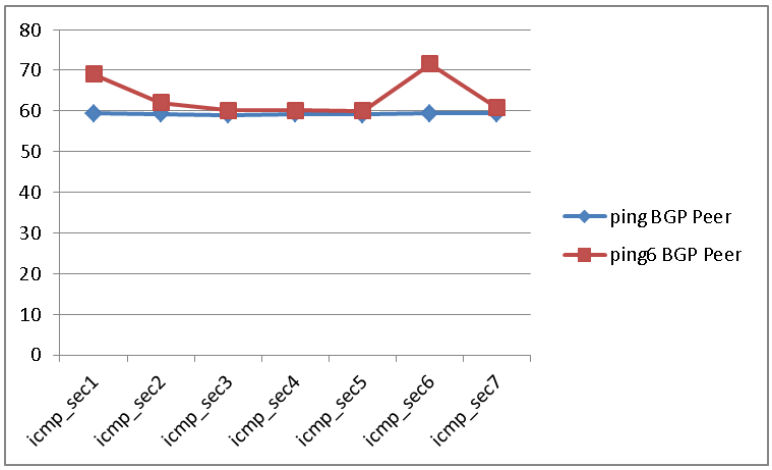
Paramater diatas digunakan untuk mengaktifkan Proxy server agar dapat berjalan untuk traffic IPv6.

5.8. Pengujian jaringan IPv6

5.8.1. Perbandingan Query ping dan ping6 ke beberapa Host

5.8.1.1. Menuju domain Peer BGP HE di Hongkong

Hasil Pengujian yang dilakukan dari Main gateway Utama IPv6 Unila menuju ke arah Peer BGP Hurricane Electric di Hongkong dengan fix IPv4 216.218.221.2 dan IPv6 2001:470:17:9::1

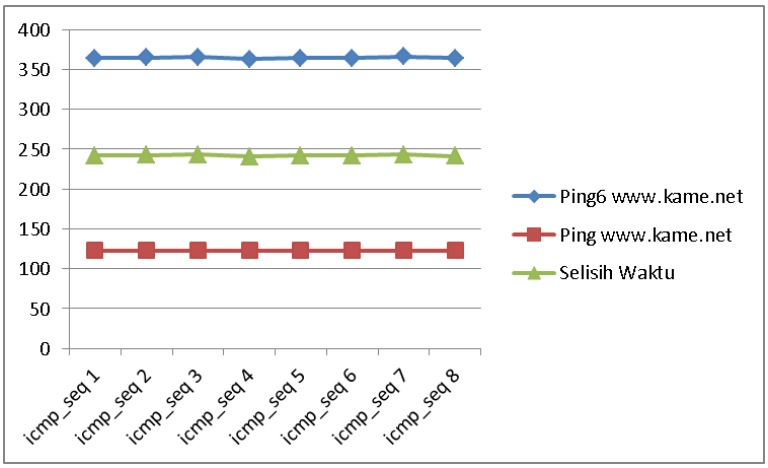


Gambar 3.20. Koneksi ke BGP Peer HE.NET

Dari data hasil pengujian terlihat bahwa rata rata TTL ke IPv4 Peer BGP :**59.263** ms , dan rata-rata TTL ke IPv6 Peer adalah **63.379** ms , Hal ini menunjukkan bahwa metode Tunneling 6to4 dengan proses enkapsulasi IPv6 over IPv4 terjadi didalamnya berdampak pada jumlah bit data yang lebih besar sehingga waktu transmit data pun lebih besar pula.

Hasil pengujian yang dilakukan dari Main Gateway Utama IPv6 Unila menuju kearah domain www.kame.net dengan fix IPv4 203.178.141.194 dan IPv6 address 2001:200:dff:fff1:216:3eff:feb1:44d7

5.8.1.2. Menuju domain www.he.net

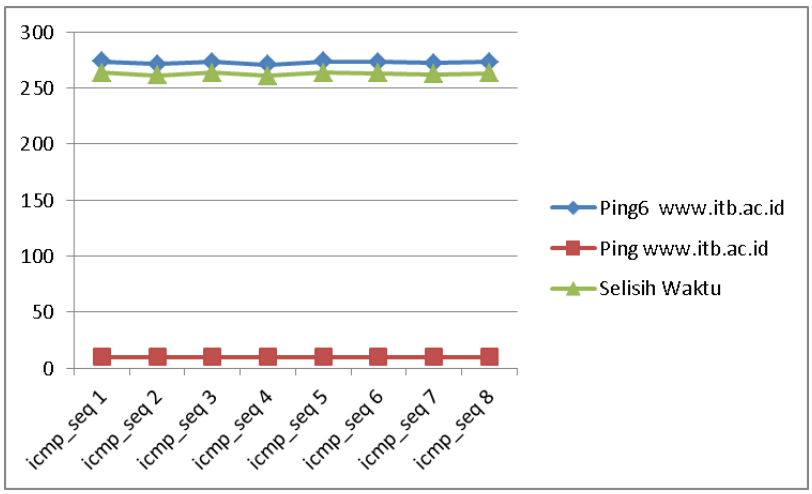


Gambar 3.21. Statistic kearah www.kame.net

Data hasil pengujian didapatkan angka rata rata ping ke alamat IPv4 sebesar **122.708** msDan rata-rata ping ke alamat IPv6 **364.723** ms , dan selisih waktu adalah **242.0334** ms. Dari data ini terlihat bahwa performance menuju domain www.kame.net menggunakan protocol IPv4 lebih baik daripada menggunakan jalur IPv6, selisih waktu sebesar **242.0334** ms diakibatkan delay time yang dibutuhkan IPv6 Tunnel Gateway Unila untuk mencapai IPv6 Tunnel server di Hongkong, serta adanya proses enkapsulasi IPv6 over IPv4.

5.8.1.3. Menuju domain www.itb.ac.id

Hasil pengujian yang dilakukan dari Main Gateway Utama IPv6 Unila menuju kearah domain www.itb.ac.iddengan fix IPv4 167.205.1.46dan IPv6 address 2403:8000:1:32::46

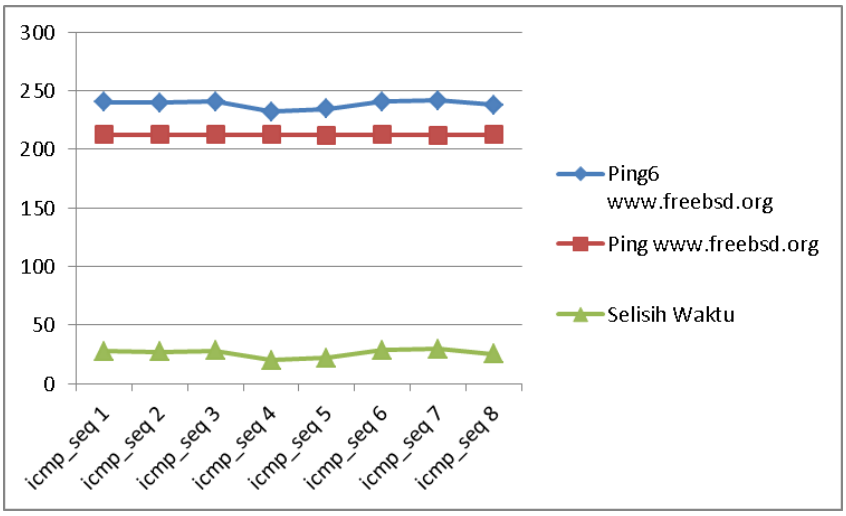


Gambar 3.22. Statistic kearah www.itb.ac.id

Ping IPv4 avg :**9.99 ms** , Ping6 average:**238.44 ms** , Selisih waktu average : **228.43 ms**

5.8.1.4. Menuju domain www.freebsd.org

Hasil pengujian yang dilakukan dari Main Gateway Utama IPv6 Unila menuju kearah domain www.freebsd.orgdengan fix IPv4 69.147.83.34dan IPv6 address 2001:4f8:fff6::22

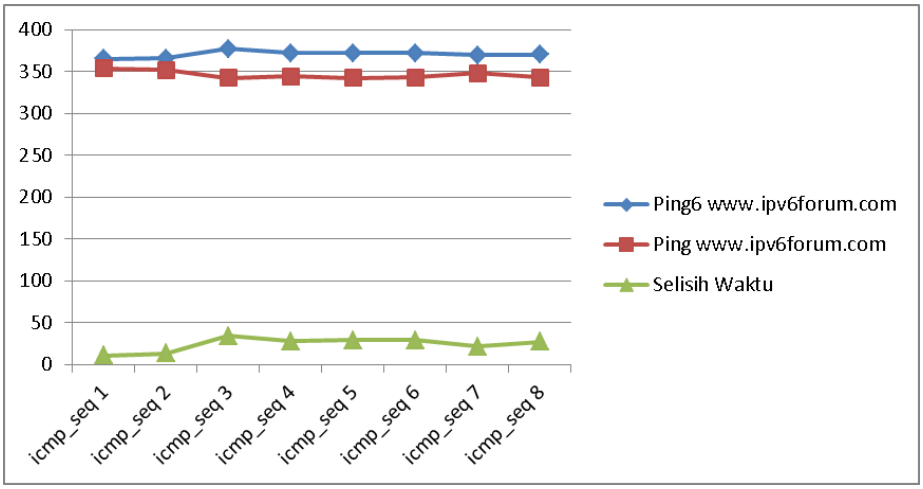


Gambar 3.23. Statistic kearah www.freebsd.org

Ping IPv4 avg :**212.27ms** , Ping6 average:**238.4397ms** , Selisih waktu average : **26.17ms**

5.8.1.5. Menuju domain www.ipv6forum.com

Hasil pengujian yang dilakukan dari Main Gateway Utama IPv6 Unila menuju kearah domain www.ipv6forum.comdengan fix IPv4 158.64.50.42dan IPv6 address 2001:a18:1:20::42

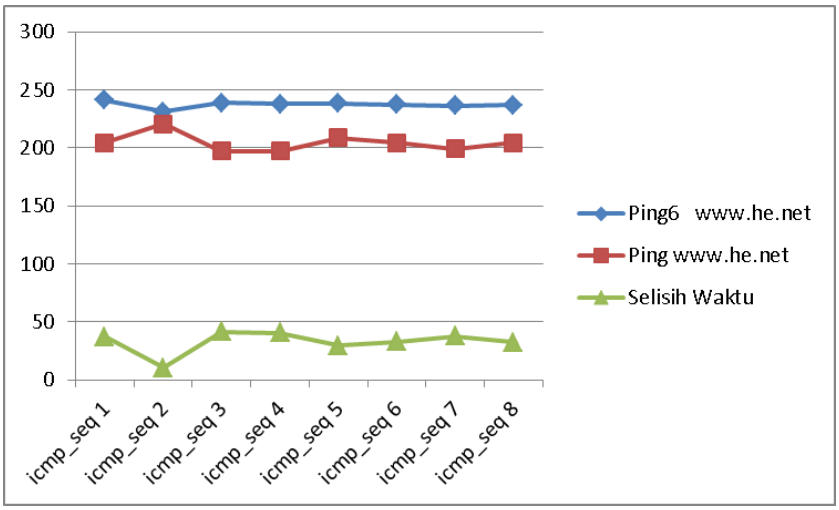


Gambar 3.24. Statistic kearah www.ipv6forum.com

Ping IPv4 avg :**346.57ms** , Ping6 average:**370.529ms** , Selisih waktu average : **23.94ms**

5.8.1.6. Menuju domain www.ipv6forum.com

Hasil pengujian yang dilakukan dari Main Gateway Utama IPv6 Unila menuju kearah domain www.he.netdengan fix IPv4 216.218.186.2dan IPv6 address 2001:470:0:76::2

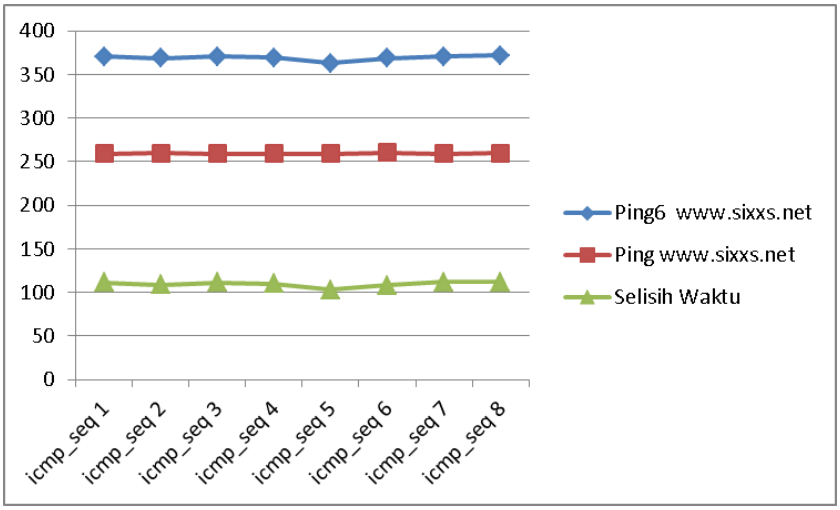


Gambar 3.25. Statistic kearah www.he.net

Ping IPv4 avg :**204.47ms** , Ping6 average:**237.31ms** , Selisih waktu average : **32.842ms**

5.8.1.7. Menuju domain www.sixxs.net

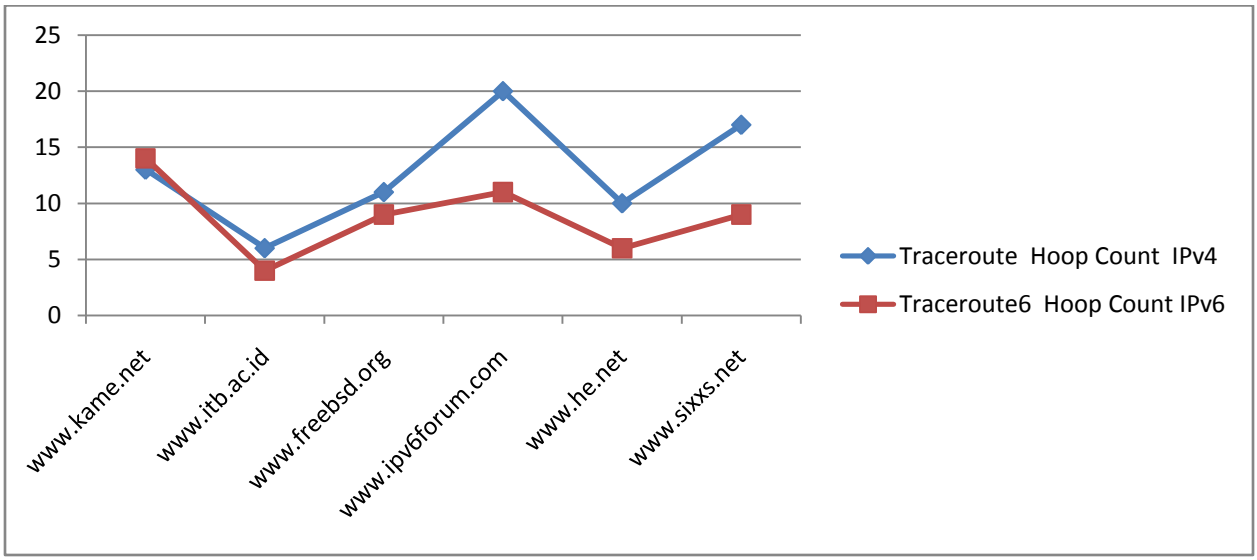
Hasil pengujian yang dilakukan dari Main Gateway Utama IPv6 Unila menuju kearah domain www.sixxs.netdengan fix IPv4 38.229.76.3dan IPv6 address 2001:838:2:1::30:67



Gambar 3.26. Statistic kearah www.sixxs.net

Ping IPv4 avg :**259.30ms** , Ping6 average:**368.69ms** , Selisih waktu average : **109.3ms**

5.8.2. Perbandingan Query Traceroute dan Traceroute6 ke beberapa Host



Gambar 3.27. Statistic Hoop Path Count ke beberapa domain

Dari data terlihat bahwa meskipun TTL kebeberapa domain lebih lama untuk tujuan ke IPv6 daripada IPv4, namun terlihat bahwa Hoop Path Count menuju ke address IPv6 adalah lebih pendek jika dibandingkan dengan Hoop Path Count menuju ke domain dengan address IPv4, hal ini menunjukkan bahwa rute tempuh yang dilewati jalur IPv6 dengan metode BGP Tunnel ini lebih baik dibandingkan dengan rute tempuh IPv4 melewati jalur provider Moratelindo.

BAB IV
KESIMPULAN

1. Dari hasil penelitian dapat disimpulkan bahwa dengan resource Hardware dan Sofware yang dimiliki Universitas Lampung saat ini dapat menjalankan Protokol IPv4 dan IPv6 secara bersamaan dengan menggunakan metode tunneling.
2. Server Produksi yang dimiliki UPT Puskom saat ini telah sukses menjalankan layanan support IPv6, sebagian diantaranya menggunakan servis dari aplikasi open source seperti BIND, Apache, Squid, Proftd, Postfix, Courier-IMAP, SSH, Packet Filter, Quagga.
3. Penggunaan metode Peer BGP Tunneling ke provider Hurricane Electric cukup efektif dalam rangka menyambungkan Jaringan Public Universitas Lampung ke Global IPv6.
4. Dari hasil penelitian dapat disimpulkan bahwa latency pengujian link ke beberapa host yang sudah support IPv4/IPv6, untuk ke alamat IPv4 latency lebih baik dibandingkan dengan latency ke IPv6, hal ini dikarenakan route IPv6 dipaksa menuju ke Peer BGP neighbor server di Hongkong dengan rata-rata latency dari Unila ke Hongkong adalah **63.379** ms.
5. Metode Peer BGP Tunneling merupakan metode paling efektif untuk mengadvertise prefix alokasi IPv6 ketika jalur IPv6 ke Provider tidak tersedia.