BAB 8 - INTERNAL IPV6

Bab ini akan membahas mengenai internal IPv6. Banyak hal yang dapat dipelajari pada bab ini, seperti header IPv6 hingga penggunaan IPv6 sebagai pengganti dari IPv4. Namun pada bahasan ini , anda akan lebih banyak belajar tentang keunikan pengalamatan IPv6, yang sangat membantu dalam mengatasi masalah jaringan (pembahasan lebih lanjut di Bab 10). Pada beberapa kasus, seperti remunerasi dan permasalahan manajemen pengalamatan multicast, agar dapat memahami masalah dengan baik diperlukan untuk belajar lebih dalam tentang informasi latar belakang kemunculan hingga pengembangan IPv6. Oleh karena itu, beberapa mata kuliah untuk internal IPv6 sangatlah penting untuk dibahas.

A.Perbedaan Antara IPV4 dan IPV6

Semua pengetahuan tentang IPv6 dimulai dengan mempelajari format header IPv6 yang memiliki cara pengalamatan yang berbeda dengan format header IPv4. Meskipun ketika spesifikasi IPv6 ditulis, CPU 64-bit yang dipilih oleh desainer IPv6 untuk mengoptimalkan header IPv6 dalam pemrosesan 64-bit data. Oleh karena itu, perlu dipelajari "format header IPv6" seperti pada gambar 8-1, yang dapat digunakan sebagai acuan dalam pengalamatan jaringan menggunakan IPv6. Karena CPU 64-bit dapat membaca 64-bit kata-kata pada memori dalam satu waktu, hal ini sangat membantu dalam pembuatan jaringan yang menggunakan IPv6 dengan 64 bit data (atau kelipatan dari 64 bit) mulai dari data kecil hingga besar yang menggunakan analogi batas data adalah 64 bit. Karena setiap batas 64-bit juga merupakan batas 32-bit, hal ini tidak berlaku sebaliknya karena dasar dari IPv6 adalah IPv4 dan panjang bit data dari IPv4 merupakan bagian dari panjang bit data IPv6. Hal ini ditunjukkan dengan header IPv4 yang berjumlah 32 bit, sedangkan header IPv6 yang berjumlah 64 bit.

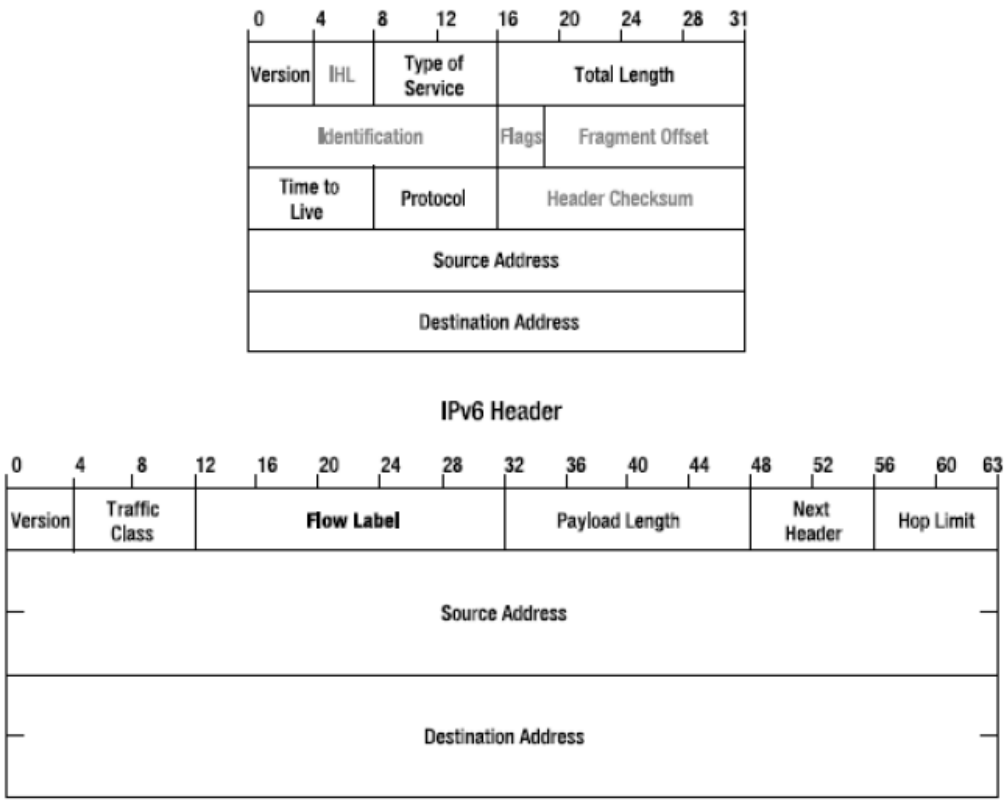

Figure 8-1. *The IPv4 and IPv6 headers*

Spesifikasi dari header pengalamatan IPv4 yang tidak ada pada spesifikasi IPv6 ditunjukkan dengan teks abu-abu, sedangkan spesifikasi dari header IPv6 yang tidak ada pada spesifikasi IPv4 akan ditunjukkan dengan teks cetak tebal. Perubahan pengalamatan IPv4 ke IPv6, yaitu:

- Versi dari aplikasi teknologi sekarang selalu menggunakan format IPv6 daripada IPv4.
- Implementasi *Internet Header Length* (IHL) yang menunjukkan panjang header IPv4 tidak lagi digunakan karena header IPv6 selalu memiliki panjang 40 byte.
- Jenis layanan aplikasi selalu memperhatikan masalah *traffic*. Format pengalamatan dari IPv4 untuk penerapan aplikasi telah digantikan berdasarkan aturan diffserv RFC 2474. Namun, pada IPv4, kedua interpretasi aplikasi tetap dapat digunakan (meskipun banyak router tidak dapat dikonfigurasi pada implementasi real IPv4). RFC IPv6 memiliki cara tertentu untuk mengatasi masalah *traffic* jaringan, tetapi umumnya aturan diffserv RFC 2474 tetap digunakan sebagai asumsi pada penggunaan IPv6.
- Ide awal *flow label* pada pengalamatan IPv6 adalah paket data pada pengiriman data yang sama dilakukan dengan membuat aliran transmisi paket yang mudah dikenali oleh *sender* dan *receiver* tanpa harus melihat header paket yang dikirimkan. Hal ini dapat dilakukan analisa performansi pengiriman data dengan melihat *Quality-of-Service* (QoS). Meskipun beberapa implementasi dilakukan dengan memperhatikan *flow label* untuk pengiriman data dengan paket TCP. Sebuah nilai 0 yang terdapat pada format header menunjukkan pengaturan *flow label* tiap sesi komunikasi yang tidak dikehendaki.
- Total panjang bit data yang dikirim sudah termasuk pada IPv4. Namun pada IPv6, panjang payload data tidak termasuk pada header IPv6 yang memiliki panjang 40-byte. Hal ini memudahkan host atau router menerima paket dengan cara memeriksa ukuran paket data apakah dapat ditampung pada header IP di frame pertama atau tidak, hal ini dilakukan untuk meningkatkan efisiensi transmisi data.
- *Identification*, *flags*, dan *fragment offset* suatu frame digunakan ketika paket IPv4 harus di fragmentasi. Proses fragmentasi di IPv6 memiliki cara kerja yang sangat berbeda (akan dijelaskan lebih rinci), sehingga proses ini dapat diturunkan ke header IPv6 sendiri.
- *Time-to-Live* sekarang disebut dengan Batas Hop. Bagian ini diinisialisasi dengan nilai yang sesuai dengan *source packet* yang akan dilewatkan pada setiap router pada jaringan. Ketika nilai tersebut mencapai nol pada implementasi jaringan, maka kondisi paket akan rusak. Proses paket ini akan terus mengalami *looping*. Pada RFC 791, waktu pengiriman paket pada IPv4 akan tetap bekerja dengan waktu yang semakin berkurang sesuai dengan jumlah detik dari paket yang buffer di router. Penerapan ini sangat sulit, sehingga tiap router harus mengurangi jumlah paket yang akan dikirimkan.

B. Checksums

Pada IPv4, header IP dilindungi oleh *header checksum*, dan protokol layer yang lebih tinggi juga memiliki *checksum*. Algoritma *checksum* untuk *header* IPv4, seperti : ICMP, ICMPv6, TCP, dan UDP (lihat sidebar). Kecuali pada IPv4, paket UDP dapat menggunakan *check summing* dan hanya memiliki *field checksum* dengan nilai nol. Pada IPv6, paket UDP harus memiliki *checksum* yang valid.

IPv6 tidak lagi memiliki *header checksum* untuk melindungi header IP. Ini berarti ketika sebuah header paket rusak yang disebabkan oleh kesalahan transmisi, paket yang diterima akan memiliki isi yang beda dengan yang dikirim. Namun, protokol layer yang lebih

tinggi dapat mengatasi masalah ini. Pada lapisan bawah menggunakan algoritma CRC untuk mendeteksi kesalahan. Sebuah *Cyclic Redundancy Check* (CRC) mirip dengan *checksum*, tetapi menggunakan perhitungan yang lebih kompleks, sehingga memiliki proses deteksi kesalahan yang lebih baik. Ethernet, misalnya, menggunakan CRC 32-bit yang secara otomatis dihitung oleh hardware Ethernet.

C.Neighbor Discovery

Ketika sebuah sistem ingin mengirim paket IPv6 ke sistem lain yang terhubung ke subnet atau link yang sama, maka harus diketahui alamat MAC (*Media Access Control*) untuk mengatasi lalu lintas paket yang akan dikirimkan. Dalam hal ini bersifat *point-to-point interface*. Penemuan lain berupa sebuah sistem menemukan alamat MAC nya masing-masing, hampir sama dengan ARP pada Ethernet pada IPv4.

Sebuah sistem IPv6 akan melakukan "*request node*" pada kelompok multicast sesuai dengan pengalamatan masing-masing. Karena adanya *request node* tiap grup, yang terdiri dari prefix FF02:0:0:0:0:1:FF00:: / 104 yang diikuti dengan suffix 24 bit dari suatu, alamat prefix yang berbeda berdasarkan interface identifier (termasuk alamat *link local*) pada alamat yang sama akan dilakukan *request node*.

Ketika sistem membutuhkan alamat link untuk sistem lain yang berada pada link yang sama, maka akan dikirimkan *neighbor discovery* ke alamat node yang diminta bahwa alamat IPv6 dari sebuah sistem akan melakukan perpindahan *coverage area*. Untuk mngukur QoS jaringan, host harus mencantumkan MAC address agar sisi *neighbor* dapat mengirimkan balasan ke host yang dituju.

Karena pegalamatan IPv6 dapat memetakan alamat yang sama sesuai dengan yang di-*request* oleh sebuah sistem yang menerima data dari *neighbor* yang pertama, maka akan diperiksa apakah request tersebut terdaftar pada pengalamatan IPv6. Jika demikian, sistem akan mengirim kembali ke *neighbor discovery* sesuai dengan alamat link tersebut. Pada saat yang sama, sistem akan menyimpan kombinasi IPv6/MAC dari *request* yang terdapat pada tabel pemetaan *neighbor*, hal ini disebut dengan "*cache neighbor*".

Client dapat melihat dan memanipulasi daftar sistem neighbor IPv6 dengan perintah NDP dengan FreeBSD dan MacOSdan menampilkan daftar neighbor dengan program neighbor menggunakan Cisco IOS. Lihat Bab 10 untuk informasi lebih lanjut tentang penggunaan perintah NDP.

D.Neighbor Unreachability Detection

RFC 2461 menetapkan prosedur untuk deteksi *neighbor unreachability*. Host IPv6 dan router aktif akan mencari *neighbor* yang berada pada *coverage area*nya. Hal ini dilakukan dengan mengirimkan pesan secara periodik dari *neighbor* yang satu ke *neighbor* yang lainnya. Jika tidak terdapat respon dari neighbor berarti terdaat beberapa masalah dan sistem akan menghapus alamat MAC *neighbor* dan mencoba prosedur multicast. Hal ini memungkinkan sistem IPv6 mendeteksi *neighbor* dalam kondisi mati dan mengubah alamat MAC (*Media Access Control*) *neighbor*. Namun hal ini sangat bermanfaat untuk mendeteksi router yang tidak aktif. Subnet yang memiliki lebih dari satu router, host hanya dapat menginstall default antar router ketika router tidak dapat menjangkau *coverage area*.

Windows XP, Linux, MacOS, dan FreeBSD memiliki cara kerja dimana sebuah router yang tidak memiliki alamat IPv6 dan tidak lagi menjalankan IPv6, maka akan beralih ke router yang lain. Namun, menon-aktifkan sebuah router memiliki dampak yang negatif, seperti download file yang tidak bisa dilakukan pada sementara waktu. Dalam hal ini tidak terdapat penjelasan untuk perbedaan kerja router lebih dalam.

E. Stateless Address Autoconfiguration

Host dan router selalu mengkonfigurasi alamat link-lokal pada setiap interface IPv6 diaktifkan. Alamat link-lokal hampir selalu berasal dari alamat interface MAC, tetapi untuk menjamin keunikan IPv6, dapat dilakukan Duplicate Address Detection (DAD), seperti yang dibahas dalam bab ini.

Setelah host memiliki alamat link-local, maka dapat diperoleh satu atau lebih alamat IPv6 global dengan menggunakan RFC 2462 *stateless address autoconfiguration*. Sebagaimana dibahas dalam Bab 2, router IPv6 akan mengirimkan paket *router advertisement* (RA) yaitu ICMPv6 tipe 134 secara berkala dan dalam menanggapi permintaan router. Informasi dalam RA meliputi :

- Sebuah 8 bit "*skr hop boundary*" akan menunjukkan host yang digunakan dalam *hop boundary* dari paket IPv6.
- Keberhasilan pengkonfigurasian alamat (M) akan ditunjukkan dengan sebuah flag. Ide dari sebuah flag ini dapat dimanfaatkan oleh host yang menggunakan mekanisme *statefull* (DHCPv6) untuk pengkonfigurasian alamat. Dan ketika flag tidak diatur, maka digunakan *stateless address autoconfiguration*.
- "*Other statefull configuration*" (O) flag memiliki kemiripan dengan M flag, namun menunjukkan bahwa host harus menggunakan mekanisme *statefull* untuk mencari *nonaddress configuration*.

F. IPv6 over Wi-Fi

Wi-Fi atau IEEE 802.11 adalah teknologi LAN nirkabel. Beroperasi pada frekuensi 2,4 GHz tanpa izin. Hal ini memiliki keuntungan bahwa IEEE 802.11 tidak memerlukan lisensi apapun, namun memungkinkan beberapa pengguna lain berada di pita frekuensi ini, seperti microwave, telepon nirkabel, bluetooth, dan komunikasi telepon-ke-perangkat mobile. Untuk dapat establish, standar IEEE 802.11 akan memancarkan sinyal melalui bagian yang sangat besar dari pita 2,4 GHz. Kenyataannya dari 11 saluran yang tersedia di AS, hanya saluran 1, 6, dan 11 yang dapat digunakan tanpa saling overlap. Sinyal wideband berisi banyak redundansi, sehingga dapat mengurangi interferensi dari sinyal yang lain.

Pada IEEE 802.11, komunikasi dapat terjadi dalam satu dari dua mode, yaitu dengan independen BSS (*Basic Service Set*) atau infrastruktur BSS. Independent BSS dikenal sebagai *ad-hoc* atau *peer-to-peer*, karena tidak ada jalur akses dalam mode ini. Infrastruktur BSS tidak memiliki nama yang umum digunakan, karena merupakan mode default yang menggunakan satu atau lebih titik akses. Dalam mode IBSS, pengiriman frame data langsung dari node sumber ke node tujuan, sedangkan dalam infrastruktur BSS, semua frame harus melalui jalur akses. Yang pertama adalah agak lebih efisien bila komunikasi yang terjadi langsung antara node nirkabel. Yang terakhir ini memiliki keuntungan bahwa setiap node dapat mencapai titik akses untuk bisa berkomunikasi dengan semua node lain dan jaringan kabel. Dalam mode *peer-to-peer*, mungkin selama tiga node, A, B, dan C, harus diposisikan sedemikian rupa sehingga A dan C dapat berkomunikasi dengan baik dengan B tetapi tidak saling berinterferensi.

Karena 802.11 bekerja secara internal dari layer yang lebih tinggi, berdasarkan "IPv6 melalui IEEE 802.11" RFC: IPv6, IEEE 802.11 akan terlihat seperti Ethernet biasa. Namun, ada beberapa perbedaan, khususnya yang berkaitan dengan *multicast*. Transmisi melalui udara memiliki *noise* yang relatif tinggi, sehingga semua *frame unicast* IEEE 802.11akan dipancarkan pada *datalink layer*. Namun, tidak ada cara bagi *receiver* untuk penerimaan *multicast* atau siaran. Hal ini menunjukkan bahwa jalur akses transmisi *multicast* dikatakan "aman" dengan performansi tinggi. Transmisi multicast pada kecepatan rendah diperlukan untuk pengimplementasian 802.11 yang dibatasi sampai 2 Mbps. Hal ini meningkatkan paket *multicast* dapat diterima ke *receiver*. Namun, kurang dari megabit per

senilai kedua *multicast* dapat menimbulkan titik saturasi pada kanal nirkabel. Dan kemungkinan besar *multicast* akan hilang. Dalam teori, hal ini berarti IPv6 melalui IEEE 802.11 kurang dapat diandalkan dibandingkan dengan IPv4. Ini dikarenakan IPv6 bergantung pada *multicast*. Namun, dalam prakteknya, LAN nirkabel yang dapat digunakan oleh IPv4, juga dapat digunakan oleh IPv6.

G. IPv6 over IEEE 1394

IEEE 1394 (atau *Firewire*) adalah teknologi link yang sangat menarik. Secara umum, saham aspek USB dan Ethernet. Seperti USB, dapat digunakan untuk menghubungkan berbagai periferal ke komputer. Tapi tidak seperti USB, dapat digunakan untuk menghubungkan beberapa komputer bersama-sama. Secara fisik, IEEE 1394 muncul baik sebagai konektor kecil empat lead, konektor enam-lead agak lebih besar, atau konektor sembilan memimpin. Empat-dan enam-lead konektor mendukung 100, 200, dan 400 Mbps (IEEE 1394a), dan konektor sembilan-lead juga mendukung 800 Mbps (IEEE 1394b). Kabel memiliki dua pasang yang membawa data dalam arah yang berlawanan. Versi enam menambahkan dua daya searah sehingga perangkat kecil dapat didukung melalui *bus Firewire*, dan versi sembilan menambahkan sinyal carrier tambahan. Perangkat dengan konektor yang berbeda dapat saling berhubungan menggunakan kabel khusus atau adaptor. Kabel *firewire* memiliki panjang maksimum 4,5 meter (hanya di bawah 15feet).

IEEE 1394 mendukung tiga mode komunikasi, yaitu:

1. Asynchronous block writes
2. Asynchronous stream packets
3. Isochronous stream packets

Asynchronous block writes adalah mode komunikasi di mana transmisi suatu node berdasarkan IEEE 1394 akan dilakukan dengan menulis data ke node lain, dan node penerima mengirimkan kembali sebuah *acknowledgement* (ACK) jika komunikasi berjalan dengan baik. Paket Streaming mirip dengan "datagram tidak dapat diandalkan" dimana layanan IP tidak mendapatkan ACK. Komunikasi *asynchronous* dapat terjadi kapan saja ketika kanal tidak ditempati, tetapi komunikasi *isochronous* memungkinkan untuk alokasi bandwidth, sehingga sangat cocok untuk real-time audio dan aplikasi video. RFC 2734 dan 3146 menentukan penggunaan baik *asynchronous block writes* atau *isochronous stream packets* untuk komunikasi IP *unicast* dan *asynchronous stream packets* untuk siaran dan *multicast*.

Ukuran paket maksimum berdasarkan IEEE 1394 tidak tetap seperti menggunakan Ethernet. Ukuran paket maksimum tergantung pada kecepatan komunikasi. Untuk node 100 Mbps, ukuran paket maksimum adalah 512 byte, dan setiap dua kali lipat dari kecepatan link, ukuran paket akan mempengaruhi. Jadi pada 400 Mbps, kemungkinan paket berukuran 2048 byte dan pada 800 Mbps berukuran 4096 byte. Karena 512 bukan ukuran paket dapat digunakan untuk IPv4 dan IPv6 membutuhkan minimal MTU 1280 byte, IPv4 dan IPv6 berdasarkan IEEE 1394 RFC terjadi fragmentasi dan *reassembly*. Penggunaan IP sebuah host berdasarkan IEEE 1394 dilakukan dengan memecah paket IP ke dalam beberapa IEEE 1394 paket, dan IP yang diterima dari node IEEE 1394 dilakukan dengan memasang kembali fragmen sebelum meneruskan paket ke layer IPv4 atau IPv6. Jadi untuk layer IP, MTU selalu terlihat cukup besar, meskipun fisik MTU dapat dikurangi ketika beroperasi pada 100 atau 200 Mbps. RFC 2734 dan 3146 menentukan MTU default 1500, tetapi Apple mengimplementasikan MTU yang hanya sedikit lebih kecil dari hardware maksimum 2048 atau 4096 byte. Meskipun MTU *mismatch* ketika *Firewire* 400 dan *Firewire* 800 Mac tersambung, *Firewire* 400 Mac mampu menerima paket lebih dari 4000 byte melalui link *Firewire*. Karena menggunakan link-layer fragmentasi, kinerja akan kurang optimal.

H. IPv6 over PPP

Point-to-Point Protocol (PPP) adalah protokol yang dapat digunakan pada semua jenis *link point-to-point*. Penggunaan PPP sangat sederhana: pada dasarnya, hanya menyediakan jenis *field* untuk menjaga protokol yang berbeda (seperti IPv4 dan IPv6) secara terpisah. Hal ini juga dilakukan *acknowledgement* (ACK) di awal dan akhir frame dan mengimplementasikan CRC, jika hardware tidak beroperasi. Karena menurut definisi, hanya dua sistem yang terhubung ke *subnet point-to-point*, tidak ada kebutuhan untuk alamat *link-layer*. Namun, PPP mengandung cara kerja yang bagus sebelum paket data dikirimkan. Sebelum hal lain terjadi, *Control Protocol Link* (LCP) melakukan sinkronisasi dengan *Maximum Receive Unit* (MRU), parameter otentikasi (jika ada), dan beberapa rincian lainnya. Ketika LCP dilakukan, *Network Control Protocol* (NCP) untuk protokol layer jaringan yang berbeda akan dilakukan proses sinkronisasi. IPCP, IP *Control Protocol*, biasanya mensinkronisasi alamat IP untuk satu sisi link, bersama dengan alamat server DNS.

Hal ini berbeda dengan IPv6 *Control Protocol* (IPV6CP). RFC 2023, yang mengatur IPv6 melalui PPP, hanya menentukan sinkronisasi berdasarkan 32-bit yang dapat digunakan sebagai *interface identifier*. Sepertinya, sistem IPv6 yang tidak memiliki alamat IPv6 yang menggunakan konfigurasi otomatis *stateless* untuk mendapatkan alamat IPv6 pada *interface*nya. Secara praktek hal ini tidak bekerja dengan baik. Listing 8-13 menunjukkan baris perintah untuk memulai sesi PPP pada sistem MacOS. FreeBSD dan Linux menggunakan implementasi PPP yang sama, tetapi menggunakan perangkat tty yang berbeda.

Listing 8-13. *Starting a PPP Session*

```
sudo /usr/sbin/pppd /dev/tty.USA19H3b1P1.1 38400 noauth local passive persist ➥
silent ipv6 ::2005
```

Keterangan :

Listing Program	Keterangan
/ dev/tty.USA19H3b1P1.1	Nama perangkat untuk serial interface. Di Linux, port PC COM 1-4 diberi nomor ttyS0-ttyS3.
38400	Kecepatan serial interface.
noauth	Jangan meminta client untuk mengotentikasi sendiri (diperlukan root untuk mengeksekusi pppd).
local	Jangan gunakan garis kontrol modem (yaitu koneksi melalui null modem).
passive	Tunggu sisi lain untuk memulai sesi LCP.
persist	Cobalah untuk membuka kembali sesi PPP setelah gagal untuk membukanya.
silent	Tunggu pengiriman paket LCP sampai menjadi aktif.
ipv6 :: 2005	Gunakan 0x00002005 sebagai identifikasi untuk akhir lokal dan indentifikasi untuk remote akhir.

Dengan menggunakan sintaks ini, perangkat PPP akan menggunakan pengalamatan IPv6 dan membuat alamat *link-local*. Namun, meskipun Cisco mengirimkan *Router Advertisement* (RA), *interface* akan mengkonfigurasi alamat IPv6 global. Jika ingin menjalankan IPv6 melalui link PPP, maka digunakan script untuk mengatur alamat IPv6 secara manual, atau dapat juga digunakan DHCPv6 sebagai *prefix*.