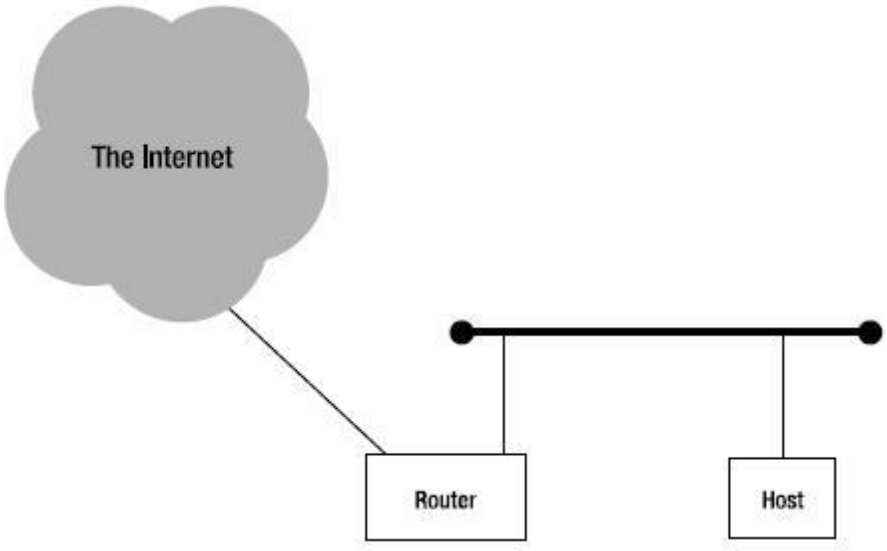
BAB IV – ROUTING

Pada paket IPv6 untuk mencapai tujuan terpencil, umumnya harus melewati beberapa router yang terdapat pada IPv6. Bab ini akan dijelaskan bagaimana cara menyiapkan routing pada Ipv6, baik dalam user dengan lingkup yang sederhana dan ISP atau lingkup perusahaan, di mana satu atau lebih protokol routing dikerahkan. Gambar 4-1 menunjukkan tata letak contoh jaringan sederhana ,dimana akan dibahas di sisa bab ini. Sebuah router terletak diantara host dan Internet. Ethernet merupakan hubungan antara router dan host yang digambarkan agak kuno karena memiliki beberapa panjang kabel Ethernet koaksial dengan terminator di kedua ujungnya.



Gambar 4-1. Contoh jaringan sederhana menggunakan router dan host

Routing IPv6

Ada perbedaan antara routing dan forwarding protokol. Routing adalah proses mempertahankan tabel routing, biasanya (tidak harus) dibantu oleh protokol routing. Dengan tabel routing tersebut, setiap paket yang datang yang tidak ditujukan kepada router dapat diteruskan ke tujuan yang telah ditetapkan, atau ke router lain yang lebih dekat ke tujuan tersebut. Tidak semua orang yang peduli untuk membuat perbedaan antara keduanya, sehingga “Routing” kadang-kadang digunakan untuk menggambarkan keduanya.

Dalam IPv6, ada lebih untuk menjadi router dari sekedar routing dan forwarding. Host yang ditunjukan pada Gambar 4-1 tidak hanya tergantung pada router untuk meneruskan paket ke dan dari internet, melainkan juga perlu router untuk memberikan prefiks alamat sehingga dapat autoconfigure alamat IPv6. Jadi router pada Gambar 4-1, seperti apa yang dilakukan semua router IPv6 yaitu mendengarkan alamat multicast allrouters (FF02 :: 2) untuk router paket router solicitation. Ketika menerima salah satu dari mereka, itu balasan dengan adverstiment router (ke alamat kelompok semua-host, FF02 :: 1) yang berisi informasi konfigurasi yang diinginkan. Router juga mengirimkan router advertisements

secara berkala untuk membiarkan host tahu bahwa informasi yang dipelajari sebelumnya masih berlaku.

Pada IPv4, router sering memberikan informasi konfigurasi untuk host dengan menggunakan Dynamic Host Configuration Protocol (DHCP), tetapi tidak ada persyaratan bahwa DHCP berjalan pada router: layanan DHCP dapat disediakan oleh server biasa, jika diinginkan. Autoconfiguration Stateless, di sisi lain, terkait erat dengan fungsi routing, dan karena itu menyediakan informasi yang klien perlu autoconfigure sendiri harus dilakukan pada router yang sebenarnya. Ada juga DHCP untuk IPv6 (DHCPv6), tapi itu tidak didukung secara luas belum, dan itu diragukan bahwa itu pernah akan sebagai mekanisme untuk host untuk mengkonfigurasi alamat IPv6 mereka. Lihat Bab 8 untuk informasi lebih lanjut tentang DHCPv6.

Routing pada Windows XP

Listing 4-1. Listing Interface IPv6

```
C:\>netsh interface ipv6 show interface
Querying active state...

Idx  Met   MTU    State         Name
---  ----  -----  ------------  -----
  7    1   1280  Connected     tun0
  6    2   1280  Disconnected  Teredo Tunneling Pseudo-Interface
  5    0   1500  Connected     Local Area Connection 3
  4    0   1500  Disconnected  Bluetooth Network
  3    1   1280  Connected     6to4 Pseudo-Interface
  2    1   1280  Connected     Automatic Tunneling Pseudo-Interface
  1    0   1500  Connected     Loopback Pseudo-Interface
```

Alamat yang terdaftar dengan perintah alamat yang dituju. Alamat untuk interface tun0 adalah 2001: db8: 31:1 :: 2. Mesin Windows akan bertindak sebagai router dan akan berbagi konektivitas itu dengan Internet IPv6 melalui tunnel IPv6 dengan host lain pada Ethernet lokal (lihat Gambar 4-1). Perintah yang diperlukan ditunjukkan pada Listing 4-2.

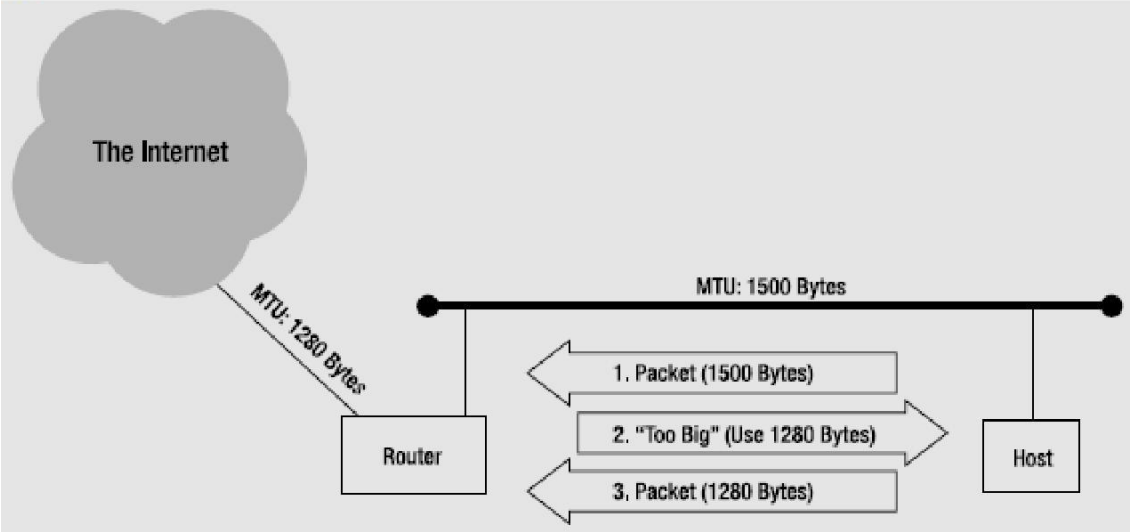
Listing 4-2. Konfigurasi Windows XP sebagai Router IPv6

```
C:\>netsh
netsh>interface ipv6
netsh interface ipv6>add address interface="local area connection 3" address=2001:➥
db8:31:2::1
Ok.

netsh interface ipv6>add route prefix=2001:db8:31:2::/64 interface=5 publish=yes
Ok.

netsh interface ipv6>set interface interface=5 forwarding=enabled advertise=enabled
Ok.

netsh interface ipv6>set interface interface=7 forwarding=enabled
Ok.
```

Pada langkah pertama, interface lokal Area Connection 3 (atau interface 5 untuk pendek, lihat Listing 4-1) mendapat alamat dalam subnet sebelumnya tidak terpakai. Baris kedua menambahkan subnet baru ke tabel routing, dan kata kunci mempublikasikan = ya memberitahu Windows bahwa alamat ini prefix adalah memenuhi syarat untuk dimasukkan dalam iklan router. Setelah itu, forwarding dan router iklan diaktifkan untuk interface Ethernet (5), dan forwarding diaktifkan pada interface tunnel (7). Pada Windows, memungkinkan forwarding pada sebuah interface memungkinkan paket yang diterima pada interface ini untuk diteruskan. Jadi, untuk membuat meneruskan paket Windows datang dari Internet melalui interface tunnel ke subnet Ethernet dan dari subnet Ethernet ke Internet, forwarding harus diaktifkan pada kedua interface. Karena tidak ada host yang membutuhkan informasi konfigurasi pada subnet tunnel, mengirim iklan router tidak sesuai pada interface ini (atau juga jendela memungkinkan). lihat Untuk informasi lebih lanjut tentang interface konteks ipv6 netsh perintah itu. Menjelajahi halaman ini mungkin tidak bekerja pada browser lain selain Internet Explorer.

Listing 4-3. Mengaktifkan traceroute dan Pesan ICMP PMTUD

```
netsh interface ipv6>firewall
netsh firewall>set icmpsetting type=11 mode=enable
Ok.

netsh firewall>set icmpsetting 2 enable
Ok.
```



Gambar 4-2. Jalur MTU

Terdapat perbedaan penting ketika PMTUD tidak diinginkan: dalam IPv4, PMTUD dapat dinonaktifkan dan paket dikirim dengan “fragmentasi” bit set ke nol, sehingga dapat terfragmentasi bila diperlukan. Tidak demikian di IPv6, sebagai router tidak bisa melakukan paket fragmen. Tapi tidak seperti IPv4, IPv6 memiliki minimal MTU wajar 1280 byte. Jadi host yang tidak siap untuk melakukan PMTUD dapat membatasi diri untuk mengirimkan paket 1280 byte atau kurang. Router, di sisi lain, tidak bisa memilih: mereka harus mengirim kembali “paket terlalu besar” pesan ICMP ketika mereka menghadapi sebuah paket yang

terlalu besar untuk link MTU. Jika tidak, komunikasi menjadi tidak mungkin, karena sumber host terus mengirimkan paket besar. Ini sudah masalah dengan IPv4, dan itu lebih buruk

dengan IPv6 karena tidak bisa diperbaiki seperti pada IPv4 dengan memecah-belah paket pula, meskipun nilai bit DF. Wajib minimum MTU 1280 byte dalam IPv6 berarti bahwa

"warisan" link dengan MTU yang lebih kecil (576 atau bahkan 296 dulunya umum untuk dial-up) harus ditingkatkan, atau langkah-langkah khusus harus diambil untuk membuat MTU fisik lebih kecil terlihat IPv6 . Lihat pembahasan IPv6 melalui IEEE 1394 dalam Bab 8.

FreeBSD

FreeBSD sangat mudah untuk mengkonfigurasi sebagai router IPv6. Semua yang dibutuhkan,misalnya konektivitas IPv6 melalui sebuah tunnel seperti ditunjukkan pada Listing 3-18 dalam Bab 3, adalah beberapa baris tambahan di / etc / rc.conf, seperti ditunjukkan pada Listing 4-4.

Listing 4-4. Mengaktifkan IPv6 Routing Dalam FreeBSD ipv6_ifconfig_xl0="2001:db8:31:2:: eui64 prefixlen 64" ipv6_gateway_enable="YES"

rtadvd_enable="YES"
rtadvd_interfaces="xl0"

Baris pertama mengkonfigurasi alamat IPv6 untuk interface xl0 Ethernet, dengan kata kunci eui64 memerintahkan sistem untuk menggunakan alamat MAC yang diturunkan dimodifikasi EUI-64 untuk lebih rendah 64 bit alamat. Jika interface sendiri tidak memiliki alamat MAC, satu dipinjam dari interface lain. Baris kedua memungkinkan IPv6 forwarding pada semua interface. Dua baris terakhir mengaktifkan dan mengkonfigurasi rtadvd, router Advertisement daemon.

LINUX

Seperti FreeBSD, menyiapkan Red Hat Linux untuk routing IPv6 adalah cukup sederhana bila menggunakan skrip sistem startup. Dengan asumsi konektivitas ke Internet IPv6 melalui interface sit1 tunnel, sesuai Daftar 3-23 dan 3-24 dalam Bab 3, semua yang dibutuhkan adalah beberapa perubahan / etc / sysconfig / network dan / etc / sysconfig / network-scripts / ifcfg -eth0. Listing 4-7 daftar yang pertama: Daftar 4-8 yang terakhir.

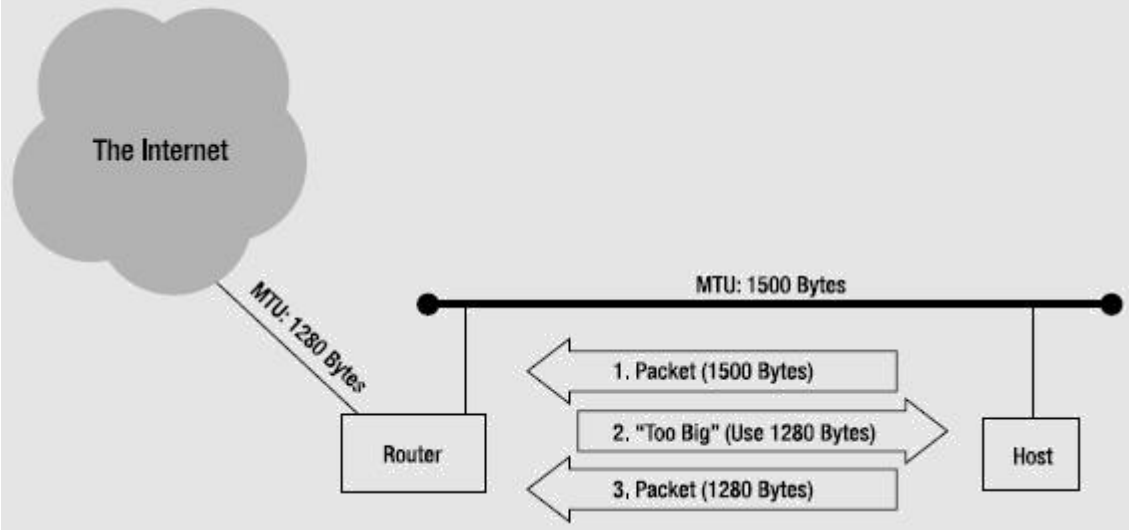
Listing 4-7. *Enabling IPv6 Routing in* /etc/sysconfig/network
NETWORKING_IPV6=yes
IPV6FORWARDING=yes
Listing 4-8.*Manual Configuration of the* eth0 *Interface*
IPV6INIT=yes
IPV6ADDR=2001:db8:31:2::1/64

Meskipun semua variasi Red Hat datang dengan semua skrip yang diperlukan, satu hal yang kurang dari distribusi workstation: router iklan daemon. Sebuah daemon radvd (sebagai lawan rtadvd bawah FreeBSD) dapat diinstal dengan menggunakan berbagai paket.

STATIC ROUTES

Jika jaringan IPv6 lebih kompleks, Anda mungkin perlu mengatur route statis untuk mendapatkan paket dari satu router ke yang berikutnya. Misalnya, host dalam subnet 2001:

db8: 31: C03 :: / 64 di Gambar 4-3 mungkin dicapai melalui Router 2, yang memiliki, misalnya, alamat 2001: db8: 31:2 :: abf.



Gambar 4-3. Host diantara dua Router

Listing 4-10. Sebuah Route Static Pada Windows

```
C:\>netsh
netsh>interface ipv6
netsh interface ipv6>add route prefix=2001:db8:31:c03::/64 interface=5 ➥
nexthop=2001:db8:31:2::abf
Ok.

netsh interface ipv6>show routes
Querying active state...

Publish  Type      Met  Prefix                    Idx  Gateway/Interface Name
-------  --------  ---  ------------------------  ---  ------------------
no       Manual      0  2001:db8:31:c03::/64        5  2001:db8:31:2::abf

netsh interface ipv6>delete route prefix=2001:db8:31:c03::/64 interface=5 ➥
nexthop=2001:db8:31:2::abf

Ok.
```

Ini juga mungkin untuk mengatur route statis di /etc /rc.conf di FreeBSD. Lihat contoh di bawah “ipv6_static_routes” di /etc /default /rc.conf.

Listing 4-12. Sebuah Route Static Pada MacOS

```
% sudo route add -inet6 2001:db8:31:c03:: -prefixlen 64 2001:db8:31:2::abf
% netstat -rnf inet6
Routing tables

Internet6:
Destination           Gateway             Flags  Netif Expire
2001:db8:31:c03::/64  2001:db8:31:2::abf  UGSc   en0
% route -n get -inet6 2001:db8:31:c03:: -prefixlen 64
   route to: 2001:db8:31:c03::
destination: 2001:db8:31:c03::
       mask: ffff:ffff:ffff:ffff::
    gateway: 2001:db8:31:2::abf
  interface: en0
      flags: <UP,GATEWAY,DONE,STATIC,PRCLONING>
 recvpipe sendpipe ssthresh rtt,msec rttvar hopcount   mtu  expire
       0         0        0        0      0        0  1500       0
% sudo route delete -inet6 2001:db8:31:c03:: -prefixlen 64 2001:db8:31:2::abf
delete net 2001:db8:31:c03::: gateway 2001:db8:31:2::abf
```

Windows dan Linux hanya memungkinkan daftar seluruh tabel routing IPv6 (hanya route yang baru saja ditambahkan ditampilkan dalam daftar), sedangkan FreeBSD dan MacOS juga mendukung mencari individu dipilih. Sintaks MacOS agak berbeda dibandingkan dengan FreeBSD: panjang prefiks harus ditentukan dengan menggunakan opsi-prefixlen (puritan merasa bahwa-prefixlen harus digunakan pada FreeBSD juga), dan route mendapatkan perintah membutuhkan opsi-n untuk mengubah off DNS lookup, atau " route to " dan " destination " akan terdaftar sebagai " invalid " jika mereka tidak dalam DNS.

Perhatikan bahwa memanipulasi tabel routing (route add dan route delete) membutuhkan hak istimewa root, tetapi tidak hanya melihat di tabel.

DYNAMIC ROUTING

Dalam dua atau tiga router setup yang sederhana, menggunakan route statis tidak masalah. Namun, di beberapa titik, routing statis menjadi tidak terkendali. Saat ini empat protokol routing bekerja dengan IPv6:

[redacted]

[redacted]IS: OSI IS-IS routing protokol yang diperluas untuk [redacted]Pv6. -4 dengan Multiprotocol Extensions.

Routing Information Protocol (RIP) adalah sebuah routing protokol yang lama dan sangat sederhana untuk perusahaan kecil hingga jaringan menengah. Pada dasarnya menyiarkan isi dari tabel routing periodik dan menggabungkan siaran ini dari router lain dalam tabel sendiri.

Sebuah "hop count" sederhana memastikan bahwa route yang paling langsung. RIP menderita dua kerugian: tidak bekerja terlalu baik dalam jaringan yang besar karena semua siaran, dan dibutuhkan sangat lama (beberapa menit) untuk mendeteksi padam dan lalu lintas mengubah route sekitar kegagalan. Ini adalah masalah mendasar yang disebabkan oleh cara

kerja RIP, jadi mereka juga hadir dalam RIPng. OSPF mengirimkan “halo” paket untuk melihat apakah router tetangga masih terjangkau dan untuk menemukan tetangga baru. Selain

itu, hanya mengirimkan update ketika ada perubahan dalam jaringan. Dalam kasus ini, semua router menjalankan algoritma Shortest Path First, dan lalu lintas segera mulai mengambil jalan terbaik baru. Reaksi cepat untuk padam dan mekanisme mengandung informasi routing untuk subset dari jaringan membuat OSPF cocok untuk jaringan dari semua ukuran. Terlepas dari usang versi 1, saat ini ada dua versi OSPF: OSPFv2 (untuk IPv4) dan OSPFv3 (untuk IPv6). Mereka benar-benar protokol terpisah yang tidak berinteraksi ketika keduanya diaktifkan. Sebuah mesin Linux atau FreeBSD (atau bahkan MacOS satu) dapat berubah menjadi router IPv6 penuh dengan menginstal perangkat lunak yang tepat. Pada awalnya, ada Zebra, yang mengimplementasikan RIP, OSPF, dan BGP masing-masing untuk IPv4 dan IPv6. Pembuat Zebra kemudian mulai bekerja pada versi komersial, dijual oleh IP Infusion. The ZebOS komersial memiliki lebih banyak fitur, termasuk IS-IS, IPv4 dan IPv6 multicast routing, MPLS, dan VLAN switching. Zebra kemajuan melambat secara signifikan, dan akhirnya, kelompok lain mengambil keuntungan dari fakta bahwa Zebra dirilis di bawah GNU Public License dan mulai mengembangkan versi sendiri dengan nama Quagga. Quagga juga mendukung IS-IS.

INSTALASI ZEBRA

Zebra (atau Quagga) mungkin tersedia sebagai paket atau RPM untuk sistem Anda, tetapi kompilasi sumber sendiri ada masalah sama sekali (untuk protokol routing daemon, setidaknya). Itu sumber tersedia dari situs Zebra di http://www.zebra.org/. Listing 4-14 daftar perintah untuk membangun dan menginstal Zebra versi 0.94. Namun, Zebra 0.95 dirilis pada awal tahun 2005. Output dari perintah tersebut yang tersisa.

Listing 4-14. *Compiling Zebra*

```
# gunzip zebra-0.94.tar.gz
# tar xvf zebra-0.94.tar
# cd zebra-0.94
# ./configure
# make
# make install
```

Kode mengkompilasi di Linux, FreeBSD, dan MacOS. Di bawah MacOS, biasanya ide yang baik untuk menginstal perangkat lunak UNIX bawah awalan khusus, misalnya, dengan / configure -. Prefix = / sw. Dengan cara ini, binari akan dipasang di / sw / sbin dan file konfigurasi di / sw / etc bukan di / usr / local / sbin dan / usr / local / etc, masing-masing, sehingga mereka tidak mendapatkan di jalan dari sistem MacOS. Zebra terdiri dari kumpulan daemon yang berbeda:

- zebra, the daemon that ties it all together, on port 2601.
- ripd, the daemon that implements RIP for IPv4, on port 2602.
- ripngd, the daemon that implements RIPng for IPv6, on port 2603.
- ospfd, the daemon that implements OSPF for IPv4, on port 2604.
- bgpd, the daemon that implements BGP for both IPv4 and IPv6, on port 2605.
- ospf6d, the daemon that implements OSPF for IPv6, on port 2606.

Sebuah mesin Linux atau FreeBSD (atau bahkan MacOS satu) dapat berubah menjadi router IPv6 penuh dengan menginstal perangkat lunak yang tepat. Pada awalnya, ada Zebra, yang mengimplementasikan RIP, OSPF, dan BGP masing-masing untuk IPv4 dan IPv6. Pembuat Zebra kemudian mulai bekerja pada versi komersial, dijual oleh IP Infusion. The ZebOS komersial memiliki lebih banyak fitur, termasuk IS-IS, IPv4 dan IPv6 multicast routing, MPLS, dan VLAN switching. Zebra kemajuan melambat secara signifikan, dan akhirnya, kelompok lain mengambil keuntungan dari fakta bahwa Zebra dirilis di bawah GNU Public License dan mulai mengembangkan versi sendiri dengan nama Quagga.

INSTALASI ZEBRA

Zebra (atau Quagga) mungkin tersedia sebagai paket atau RPM untuk sistem Anda, tetapi kompilasi sumber sendiri ada masalah (untuk protokol routing daemon).

Listing 4-14. *Compiling Zebra*

```
# gunzip zebra-0.94.tar.gz
# tar xvf zebra-0.94.tar
# cd zebra-0.94
# ./configure
# make
# make install
```

• zebra, the daemon that ties it all together, on port 2601.

• ripd, the daemon that implements RIP for IPv4, on port 2602.

• ripngd, the daemon that implements RIPng for IPv6, on port 2603.

• ospfd, the daemon that implements OSPF for IPv4, on port 2604.

• bgpd, the daemon that implements BGP for both IPv4 and IPv6, on port 2605.

• ospf6d, the daemon that implements OSPF for IPv6, on port 2606.

Listing 4-15. Sebuah Konfigurasi zebra Dasar

```
!
hostname zebra
password easy-to-guess
enable password hard-to-guess
!
access-list zebra-access permit 127.0.0.1/32
!
ipv6 access-list zebra-access-ipv6 permit ::1/128
!
line vty
 access-class zebra-access
 ipv6 access-class zebra-access-ipv6
 exec-timeout 60
!
```

Listing 4-16. Menghubungkan ke Daemon zebra untuk Pertama Kalinya

```
# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is zebra (version 0.94).
Copyright 1996-2002 Kunihiro Ishiguro.


User Access Verification

Password:
zebra-t> enable
Password:
zebra-t# show running-config
Current configuration:
!
hostname zebra-t
password 8 U2uZd3cGSy89g
enable password 8 0qFtOGjdVxDwI
service password-encryption
!
interface lo
!
interface eth0
 ipv6 nd suppress-ra
!
interface sit1
 ipv6 nd suppress-ra
!
access-list zebra-access permit 127.0.0.1/32
!
!
line vty
 access-class zebra-access
!
end
zebra-t# configure terminal
zebra-t(config)# interface eth0
zebra-t(config-if)# description First Ethernet interface
zebra-t(config-if)# exit
zebra-t(config)# exit
zebra-t# show interface eth0
Interface eth0

Description: First Ethernet interface
index 3 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
HWaddr: 00:01:02:29:23:b6
inet 172.16.1.5/24 broadcast 255.255.255.255
inet6 fe80::201:2ff:fe29:23b6/64
inet6 2001:db8:31:2::1/64
  input packets 9624, bytes 1142979, dropped 0, multicast packets 0
  input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
  output packets 5549, bytes 1042517, dropped 0
  output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
  collisions 0
zebra-t# quit
Connection closed by foreign host.
```

OSPFv3

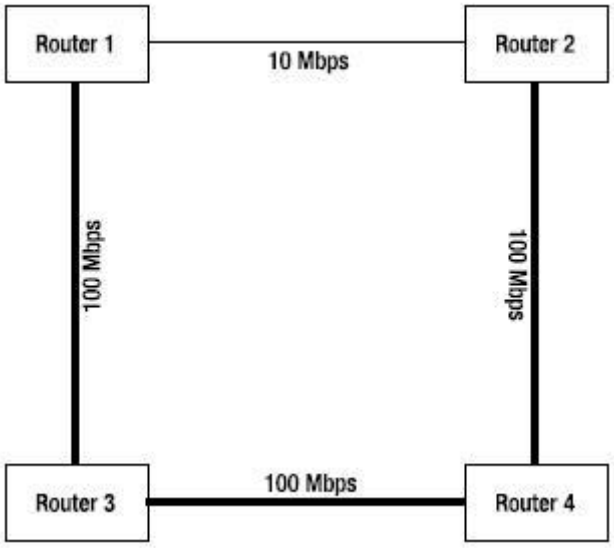
OSPF untuk IPv6 adalah protokol yang luas dan kompleks, dan tidak mungkin untuk melakukan keadilan dalam beberapa halaman.

Listing 4-25. *OSPFv3* pada *Cisco Router*

```
!
interface FastEthernet2/0
ipv6 ospf 230 area 0.0.0.0
!
```

Area dan Metrik

Perbedaan antara metrik OSPF dan hop RIP diilustrasikan pada Gambar 4-4. Ada dua cara untuk mendapatkan dari router 1 ke router 2: satu 10 Mbps atau tiga hop hop 100 Mbps. Untuk RIP ini adalah no-brainer (satu hop lebih baik dari tiga). Pada Zebra, mengubah default selalu merupakan ide yang baik, seperti Zebra tidak memiliki akses yang baik untuk menyebar informasi bandwidth interface (dengan asumsi itu mengganggu untuk melihat informasi ini sama sekali).



NEIGHBORS

Listing 4-28. Menampilkan OSPFv3 Neighbors Aktif

```
#show ipv6 ospf neighbor
Neighbor ID  Pri  State          Dead Time  Interface ID  Interface
192.0.2.91   128  FULL/BDR       00:00:38   3             FastEthernet2/0
192.0.2.17   128  FULL/DROTHER   00:00:35   2             FastEthernet2/0
192.0.2.19     1  FULL/DROTHER   00:00:30   8             FastEthernet2/0
```

Listing 4-29. Mengaktifkan OSPFv3 Dalam Zebra

```
!
interface xl0
 ipv6 ospf6 cost 10
!
router ospf6
 router-id 192.0.2.18
 redistribute static
 interface xl0 area 0.0.0.0
!
```

BGP

Border Gateway Protocol sangat berbeda dari semua protokol routing yang lancar lainnya, baik dalam tujuan dan dengan cara itu dikonfigurasi dan kemudian beroperasi. RIP, OSPF, dan IS-IS semua Protokol Gateway Interior (IGP). BGP, di sisi lain, adalah Gateway Protocol Eksternal (EGP), dan digunakan untuk berkomunikasi antara jaringan dari berbagai organisasi. BGP memungkinkan paket untuk menemukan jalan mereka dari satu Internet Service Provider ke yang berikutnya. Ini berarti bahwa semua ISP yang cukup besar untuk menghubungkan dua atau lebih ISP lain menjalankan BGP. Melakukan hal ini memungkinkan mereka untuk secara dinamis mengubah route yang ada selama sesi koneksi lain dan dengan demikian mengisolasi diri dari sebagian besar masalah yang mungkin terjadi dalam jaringan hulu ISP. Ini disebut “multihoming”.

Listing 4-30 menunjukkan kutipan dari IPv4 “tabel routing global,” yang adalah apa set lengkap routing BGP informasi bagi seluruh Internet disebut. (Semakin pendek alamat IPv4 membuat informasi lebih mudah untuk melihat dari informasi IPv6 setara).

Listing 4-30. Bagian dari BGP Routing Table global IPv4

```
   Network          Next Hop      Metric LocPrf     Weight Path
*  4.0.0.0          62.9.194.3        40             0 646 335 i
*                   80.31.82.129      50             0 645 335 i
*>                  23.248.72.89             105     0 129 335 i
*> 64.86.28.0/24    80.31.82.129      50             0 645 3047 i
*                   62.9.194.3        40             0 646 645 3047 i
*                   23.248.72.89      60             0 129 645 3047 i
*>i145.52.0.0       195.69.14.34       0    110      0 110 i
*                   62.9.194.3        40             0 646 110 i
*                   23.248.72.89      60             0 129 354 110 i
*                   80.31.82.129      50             0 645 354 110 i
```

ADDRESS FAMILIES

Listing 4-31. Pengaturan IPv4 BGP Sederhana

```
!
router bgp 65500
 no synchronization
 bgp log-neighbor-changes
 network 192.0.2.0
 neighbor 172.16.1.242 remote-as 65500
 neighbor 172.16.1.242 prefix-list outfilter out
 no auto-summary
!
ip route 192.0.2.0 255.255.255.0 Null0
!
ip prefix-list outfilter seq 5 permit 192.0.2.0/24
!
```

Listing 4-32. Pengaturan IPv6 BGP Sederhana

```
!
router bgp 65500
 bgp log-neighbor-changes
 neighbor 3ffe:9500:3C:74::10 remote-as 64900
 no neighbor 3ffe:9500:3C:74::10 activate
 !
 address-family ipv6
 neighbor 3ffe:9500:3C:74::10 activate
 neighbor 3ffe:9500:3C:74::10 prefix-list outfilter-ipv6 out
 network 2001:DB8:31::/48
 no synchronization
 exit-address-family
!
ipv6 prefix-list outfilter-ipv6 seq 5 permit 2001:DB8:31::/48
ipv6 route 2001:DB8:31::/48 Null0
!
```

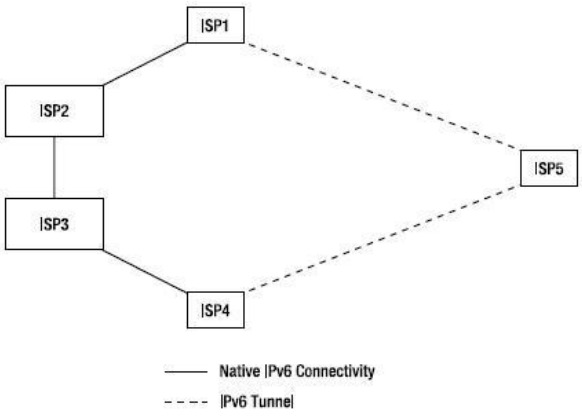
Listing 4-33. Konfigurasi Pengelompokan Address Family IPv4

```
Cisco#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Cisco(config)#router bgp 65500
Cisco(config-router)#address-family ipv4
Cisco(config-router-af)#^Z
Cisco#show running-configuration | begin router bgp
router bgp 65500
 bgp log-neighbor-changes
 neighbor 3ffe:9500:3C:74::10 remote-as 64900
 neighbor 172.16.1.242 remote-as 65500
 !
 address-family ipv4
 no neighbor 3ffe:9500:3C:74::10 activate
 neighbor 172.16.1.242 activate
 neighbor 172.16.1.242 prefix-list outfilter out
 no auto-summary
 no synchronization
 network 192.0.2.0
 exit-address-family
 !
```

MENGHINDARI TUNNELS

Menjalankan IPv6 berarti menghubungkan ke 6bone atas tunnel, setiap orang diberikan akses ke orang lain melalui tunnel tersebut. Hari-hari ini, jaringan yang lebih dan lebih mendukung IPv6 asli, dan orang-orang yang umumnya tidak memiliki tunneling mereka selaras cukup erat dengan infrastruktur IPv4, sehingga aturan, tunnel jarak antara organisasi yang berbeda tidak diperlukan lagi. Namun, beberapa tunnel ini tetap beroperasi, dan beberapa jaringan memungkinkan lalu lintas antara dua jaringan remote untuk melewati jaringan mereka.

Dalam prakteknya, ini berarti bahwa kadang-kadang dua ISP (sebagian besar pengguna akhir tidak bisa melakukan BGP) memiliki “lebih pendek” tunnel jalur antara mereka, seperti yang terlihat oleh BGP. Dalam Gambar 4-5, jalur asli dari ISP1 ke ISP4 memiliki dua hop tambahan, sedangkan jalur tunnel selama ISP5 terpencil hanya memiliki satu hop menengah.



Gambar 4-5. Asli vs konektivitas yang ditunnel di IPv6

Listing 4-40. Menurunkan Preferensi untuk Destinasi tunnel

```
!
router bgp 65500
 neighbor 2001:7f8:1::a506:3000:1 remote-as 64900
 neighbor 3ffe:9500:3C:74::10 remote-as 65200
!
 address-family ipv6
 neighbor 2001:7f8:1::a506:3000:1 activate
 neighbor 2001:7f8:1::a506:3000:1 route-map punish-tun in
 neighbor 3ffe:9500:3C:74::10 activate
 neighbor 3ffe:9500:3C:74::10 route-map punish-tun in
 neighbor 3ffe:9500:3C:74::10 route-map prepend2 out
!
ip as-path access-list 66 permit _64512_
ip as-path access-list 66 permit _64999_
!
route-map punish-tun permit 10
 match as-path 66
 set local-preference 66
!
route-map punish-tun permit 20
!
route-map prepend2 permit 10
 set as-path prepend 65500 65500
!
```

Konfigurasi dalam contoh memiliki dua neigbour. Keduanya telah alamat keluarga IPv6 diaktifkan, dan untuk kedua, route peta menghukum-tun diterapkan untuk update BGP masuk. Peta route ini, tercantum di dekat bagian akhir contoh, pertama melewati semua update melalui jalur SEBAGAI saringan 66, yang cocok dengan route BGP dengan AS 64.512 atau 64.999 di AS jalan Karena semua route lainnya memiliki nilai local preference kosong, yang sama dengan 100, route-route ini sekarang hanya digunakan jika tidak ada alternatif.

Melihat Gambar 4-5, kita dapat membayangkan bahwa ISP5 akan terpengaruh oleh route ini peta sehingga route dari ISP1 ke ISP4 lebih ISP5 akan memiliki preferensi lokal 66, sedangkan route melalui ISP 2 dan 3 memiliki preferensi lokal standar . Karena lebih tinggi mengalahkan preferensi lokal lebih pendek AS jalan, paket-paket dari ISP1 sekarang mengalir ke ISP4 melalui ISP2 dan ISP3.

ISP4 harus menerapkan kebijakan serupa untuk paket untuk menghindari tunnel di belakang perjalanan. Hanya dalam kasus ISP4 tidak melakukan hal ini, Listing 4-40 juga menerapkan peta route prepend2 pada update keluar menuju AS 65200, yang diduga sebuah tunnel-senang AS, sehingga lebih kecil kemungkinannya untuk “menarik” lalu lintas dengan mengorbankan konektivitas asli.

OSPFV3 DAN BGP UNTUK IPV6 DI JUNIPER

Sejauh ini, satu-satunya jenis router (kecuali satu yang Anda membangun sendiri berdasarkan UNIX) yang telah kita bahas adalah Cisco. Cisco memiliki banyak keuntungan : itu membuat banyak produk baik, mulai dari sangat kecil hingga sangat besar. Dan meskipun laporan konfigurasi aneh pada suatu saat, mereka tidak terlalu sulit untuk diikuti. Semua ini sangat berbeda dengan Juniper : model terkecil adalah jarak menengah-tinggi, dan meskipun mekanisme konfigurasinya memiliki banyak terjadi untuk itu, sulit dimengerti pada awalnya. Di sisi lain, Juniper menerapkan IPv6 sangat baik, dan, yang lebih penting, cepat. The Application Specific Integrated Circuit (ASIC) yang router Juniper kekuasaan sepenuhnya mendukung IPv6, sehingga IPv6 bukan warga negara kelas dua di wilayah Juniper.

CATATAN : Karena kompleksitas tambahan, itu tidak mungkin untuk bahkan menggores permukaan sintaks konfigurasi Juniper. Anda harus dapat mengkonfigurasi router Juniper untuk IPv4 untuk memahami listing berikut.

Listing 4-41 menunjukkan bagian pertama dari konfigurasi Juniper mana karakteristik interface didefinisikan.

Listing 4-41. The “Interfaces” Bagian dari Konfigurasi Juniper

```
interfaces {
    ge-0/0/0 {
        vlan-tagging;
        unit 288 {
            vlan-id 288;
            family inet6 {
                address 2001:db8:31:288::/64 {
                    eui-64;
                }
            }
        }
    }
}
```

Listing 4-42. Bagian “Routing-option” dari Konfigurasi Juniper

```
routing-options {
    rib inet6.0 {
        static {
            route 2001:db8:31::/48 {
                discard;
                install;
                readvertise;
            }
            route 2001:db8:31:3000::/52 {
                next-hop 2001:db8:31:3::2;
                install;
            }
        }
    }
    router-id 192.0.2.7;
    autonomous-system 65500;
}
```

Tulang rusuk inet6.0 pada baris kedua adalah standar IPv6 Routing Information Base, jadi ini adalah rute IPv6 statis biasanya pergi. Dalam kasus ini, ada rute statis untuk 2001: db8: 31 :: / 48 yang membuang semua paket yang menangkap dan readvertised ke protokol routing. Rute statis kedua adalah untuk potongan besar dari / 48 (a / 52) dan menunjuk ke alamat yang cocok paket harus diteruskan. Listing 4-43 menunjukkan "protokol" bagian dari konfigurasi.

Listing 4-43. Bagian "Protocols" dari Konfigurasi Juniper

```
protocols {
    bgp {
        group ibgp {
            type internal;
            local-address 192.0.2.7;
            family inet {
                unicast;
            }
            family inet6 {
                unicast;
            }
            peer-as 65500;
            neighbor 192.0.2.18;
        }
```

```
        group bgp-v6 {
            type external;
            import bgp-v6-in;
            family inet6 {
                unicast;
            }
            export bgp-v6-out;
            neighbor 2001:7f8:1::a506:3000:1 {
                authentication-key "$9$5Fdsikekasi/97dj"; ## SECRET-DATA
                peer-as 64900;
            }
        }
    ospf3 {
        export redist-ospf3;
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/2/0.288;
        }
    }
}
```

Listing 4-43. Bagian "*Policy-options*" dari Konfigurasi Juniper

```
policy-options {
    policy-statement import-v6 {
        term 1 {
            from {
                route-filter 2001:16F8::/32 orlonger;
            }
            then accept;
        }
        then reject;
    }
```

```
    policy-statement bgp-v6-in {
        term 1 {
            from policy import-v6;
            then reject;
        }
        then {
            local-preference 300;
            accept;
        }
    }
    policy-statement bgp-v6-out {
        term 1 {
            from {
                route-filter 2001:db8:31:/48 exact;
            }
            then accept;
        }
        then reject;
    }
    policy-statement redist-ospf3 {
        term connected {
            from protocol direct;
            then accept;
        }
        term static {
            from protocol static;
            then accept;
        }
    }
}
```

Filter pertama, impor-v6, memungkinkan awalan lokal 2001: db8: 31 :: / 48 atau lebih prefiks dalam hal ini / 48. Namun, filter ini tidak digunakan secara langsung. Sebaliknya, itu disebut oleh yang berikutnya, BGP-v6-in. Filter ini menolak semua paket yang cocok 2001: db8: 31 :: / 48 atau lebih melalui impor-v6 dan kemudian melanjutkan untuk menerima semua rute lain untuk diproses lebih lanjut, menugaskan mereka nilai local preference dari 300 dalam proses.

Filter BGP-v6-out, di sisi lain, tidak diatur untuk skalabilitas masa depan dan cocok awalan lokal secara langsung. Tidak seperti filter yang masuk, yang menolak awalan kita sendiri dan setiap awalan lagi jatuh di dalamnya, yang satu ini cocok hanya / 48 dan tidak ada lagi untuk menghindari bocor lebih spesifik.

SITE-LOCAL ADDRESSES

Pada IPv4, itu umum untuk menggunakan RFC 1918 rentang alamat pribadi (10.0.0.0 / 8, 172.16.0.0/12, dan 192.168.0.0/16) untuk komunikasi internal. Idenya adalah untuk mengambil alamat pribadi ke tingkat berikutnya dalam IPv6 dengan memperkenalkan "scope" mekanisme. Kami sudah mendiskusikan alamat dengan scope link-lokal. Karena mereka hanya berlaku pada subnet individu, mereka dapat digunakan kembali pada subnet lain tanpa masalah nyata. Alamat situs-lokal harus bekerja dalam nada yang sama: mereka hanya digunakan dalam sebuah "site" individu sehingga situs lain dapat menggunakan kembali kisaran alamat yang sama.

Gambar 4-6 menunjukkan format alamat ini. Awalan adalah fc00 :: / 7. Bit local menunjukkan apakah ID global secara acak (L = 1) atau terdaftar melalui registry (L = 0), yang dimungkinkan di masa depan. The 40-bit global yang ID cukup besar untuk membuat kecelakaan tabrakan, tetapi mereka masih mungkin terjadi pada kesempatan. Sebuah tabrakan adalah situasi di mana dua organisasi memilih awalan lokal unik yang sama.

Gambar 4-6. Site local unik IPv6 format alamat unicast

Pada IPv4, itu praktek umum bahwa semua host memiliki alamat pribadi yang diterjemahkan ke dalam alamat global pada perbatasan jaringan. Karena kurangnya NAT, konfigurasi ini sulit untuk mengimplementasikan IPv6 di: alternatif adalah dengan menggunakan proxy, tapi ini tidak tersedia untuk semua protokol. Sebuah alternatif adalah untuk memberikan semua host baik swasta / lokal dan masyarakat / dunia alamat. Untuk menghindari terbalik, di mana tuan rumah mencoba untuk menghubungi server jauh berdasarkan alamat lokal, dianjurkan untuk menjaga alamat ini keluar dari DNS (mereka harus tidak muncul baik AAAA atau catatan PTR). Karena, jelas, menggunakan alamat berarti memiliki mereka dalam DNS, ini secara efektif berarti IETF memberi mandat praktek "two-faced DNS", di mana server DNS memberikan jawaban yang berbeda berdasarkan alamat host melakukan query.