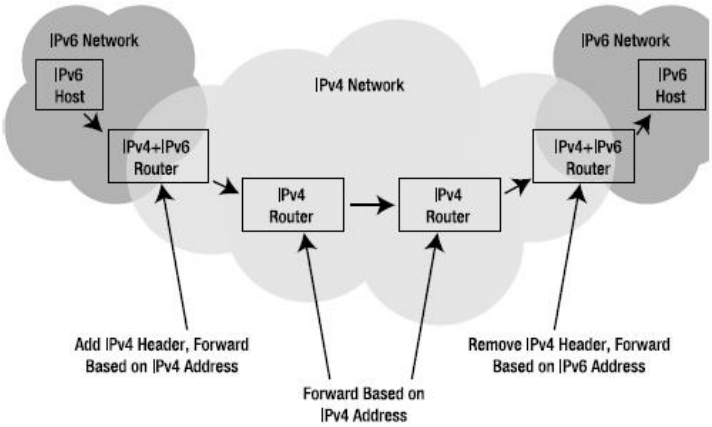
BAB III - Tunnels

Bab ini membahas 6to4 tunneling dan konfigurasi secara manual atau otomatis. Sebuah tunnels adalah mekanisme suatu protokol dirumuskan menjadi protokol lain yang digunakan pada jaringan yang tidak mendukung protokol asli. Tunneling IPv6 dalam IPv4 tunneling dilakukan dengan menambahkan header IPv4 sebelum paket IPv6 dengan mencantumkan alamat tujuan pada header IPv4

Gambar 3-1 menunjukkan apa yang terjadi pada paket tunnels IPv6 melalui jaringan IPv4.



Gambar 3-1. Tunneling paket IPv6 melalui jaringan IPv4

Saat ini setidaknya lima mekanisme yang berbeda didefinisikan yang memungkinkan otomatis tunneling IPv6 dalam IPv4:

- "Automatic Tunneling," menggunakan alamat IPv4-compatible (RFC 2893).
- 6over4: " Transmission of IPv6 over IPv4 Domains without Explicit Tunnels" (RFC 2529).
- ISATAP: "Intra-Site Automatic Tunnel Addressing Protocol" Pada saat penulisan ini, ISATAP belum diterbitkan sebagai RFC.
- 6to4: "Connection of IPv6 Domains via IPv4 Clouds" (RFC 3056).
- Teredo: "IPv6 Tunneling UDP melalui Nat." Teredo sebelumnya dikenal sebagai Shipworm. belum diterbitkan sebagai RFC.

1. Atomatic Tunnelling

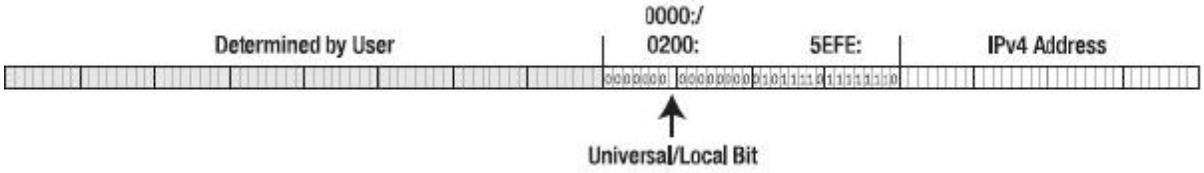
"Automatic tunneling" merupakan salah satu tunneling otomatis, "Automatic tunneling" sangat mirip dengan 6to4, alamat IPv6 disebut "IPv4-compatible" dan terdiri dari 96 bit nol (awalan :: / 96) diikuti dengan alamat IPv4 32-bit. Gambar 3-2 menunjukkan format alamat IPv4 yang kompatibel digunakan dalam "Automatic tunneling".

Gambar 3-2. Format alamat IPv4 yang kompatibel

2. 6over4 dan ISATAP

6over4 melakukan tunneling dengan memperlakukan jaringan IPv4 sebagai subnet IPv6, ini memungkinkan alamat reguler autoconfiguration. Sayangnya, dengan 6over4, IPv4 harus mendukungan multicast. Karena sebagian besar jaringan IPv4 tidak mendukung multicast routing, 6over4 tidak banyak digunakan.

ISATAP memperlakukan jaringan IPv4 sebagai Non-Broadcast, jaringan Multiple Access (NBMA). ISATAP mengkodekan alamat IPv4 ke interface identifier interface alamat IPv6. Gambar 3-3 menunjukkan hubungan antara alamat IPv4 dan IPv6 di ISATAP.



Gambar 3-3. Format alamat ISATAP

3. TEREDO

Ide dibalik Teredo adalah untuk memungkinkan host balik Network Address Translators ke IPv6 tunnels di IPv4. Meskipun beberapa implementasi Teredo tersedia, protokol ini masih sedikit dari keadaan fluks, dan server dan infrastruktur yang diperlukan estafet belum tersedia.

4. 6to4

6to4 mirip dengan ISATAP (atau lebih tepatnya, sebaliknya, 6to4 memungkinkan untuk IPv6 tunnel lebih dari IPv4 antara situs.Gambar 3-4 menunjukkan alamat 6to4

Gambar 3-4. Format alamat 6to4

Ketika sistem 6to4 ingin mengirim paket ke sistem 6to4 lain, suatu paket IPv6 dalam paket IPv4 dan alamat paket ini ke alamat IPv4 dikodekan dalam alamat tujuan 6to4. Setelah resepsi, IPv4 host tujuan menghilangkan header IPv4 dan terus memproses paket IPv6. Komunikasi antara dunia 6to4 dan Internet IPv6 biasa difasilitasi oleh relay. Dengan cara ini, paket otomatis menemukan jalan ke salah satu dari relay tanpa perlu untuk mengkonfigurasi setiap relay. Relay merangkum paket dalam IPv4 dan mengirimkan ke alamat IPv4 dikodekan dalam alamat 6to4.

4.1. 6to4 Pada Windows

Tidak perlu untuk secara khusus mengaktifkan 6to4 pada Windows XP. Ketika IPv6 diinstal,sistem secara otomatis membuat sebuah interface semu 6to4 jika (non-RFC 1918) alamat IPv4 publik tersedia. Jadi paket ke alamat 6to4 secara langsung ditunnel ke tujuan mereka dalam IPv4, asalkan sistem memiliki alamat IPv4 biasa. Selain itu, jika tidak ada konektivitas IPv6 lainnya, Windows menginstal satu atau lebih route IPv6 default yang mengarah ke relay 6to4. Listing 3-1 menunjukkan output netsh relevan dengan awalan 6to4 berdasarkan alamat IPv4 223.224.225.226.

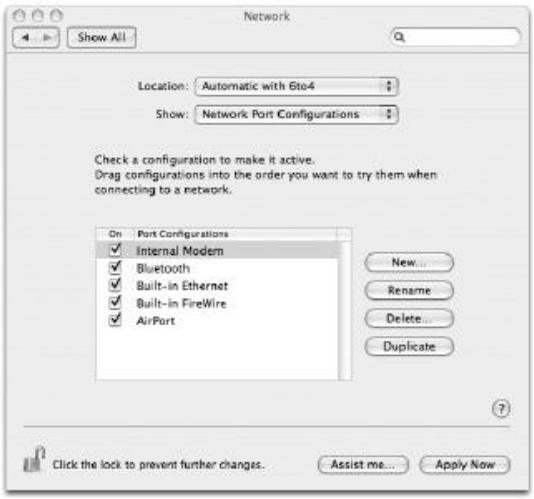
Listing 3-1. *Listing Addresses and Routes Using the* netsh *Command*

```
C:\>netsh interface ipv6 show address
Interface 3: 6to4 Tunneling Pseudo-Interface
Addr Type DAD State Valid Life Pref. Life Address
-------- --------- ----------- ----------- ----------------------
Other Preferred infinite infinite 2002:dfe0:e1e2::dfe0:e1e2
C:\>netsh interface ipv6 show routes
Querying active state...
Publish Type Met Prefix Idx Gateway/Interface Name
------- ------- ---- -------- --- --------------------
yes Manual 1191 ::/0 3 2002:836b:213c:1:e0:8f08:f020:8
yes Manual 1041 ::/0 3 2002:c058:6301::c058:6301
yes Manual 1001 2002::/16 3 6to4 Tunneling Pseudo-Interface
```

Perintah netsh juga dapat digunakan untuk mengubah nama dan pengaturan untuk relay 6to4

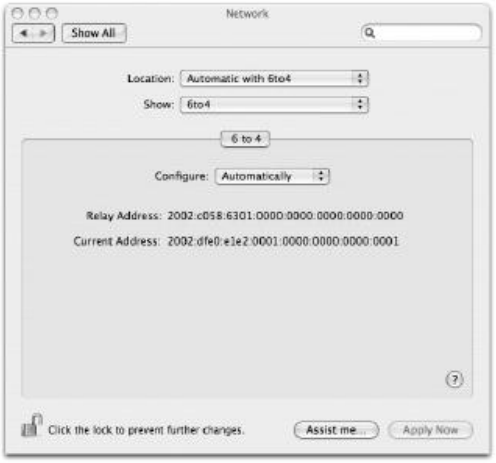
4.2. 6to4 Pada MacOS X

Tidak seperti Windows XP, MacOS X tidak secara otomatis mengaktifkan 6to4. Jika Anda ingin menggunakan 6to4 tunneling, Anda harus membuat port jaringan baru. Melakukannya dengan memilih Network Port Configurations di Network panel dari System Preferences, seperti yang ditunjukkan pada Gambar 3-5, mengklik New button, mengetik nama port baru, dan memilih Port: 6to4.



Gambar 3-5. Menambahkan network port 6to4

Dengan port 6to4 dikonfigurasi, MacOS secara otomatis akan membuat alamat 6to4 dan menginstal default route menuju alamat estafet anycast secepat itu memiliki IPv4 reachability dengan publik, alamat non-RFC 1918. Alamat relay dapat diubah dalam konfigurasi port 6to4 dalam pengaturan jaringan, yang ditunjukkan pada Gambar 3-6. Alamat 6to4 ditugaskan ke port 6to4 tidak dapat diubah, hal ini selalu alamat 1 di subnet 1 dari awalan 6to4 berasal dari alamat IPv4 saat ini.



Gambar 3-6. Konfigurasi 6to4

Cara termudah untuk menggunakan utilitas baris perintah FreeBSD yang diturunkan yang perlu akses root bawah MacOS adalah dengan sudo, seperti ditunjukkan pada Listing 3-2. Namun, mengubah pengaturan yang biasanya dikendalikan oleh program Preferensi dapat menyebabkan hasil yang tak terduga.

Listing 3-2. *Using* sudo *to Execute Commands as Root*

```
% sudo ifconfig stf0 inet6 2002:dfe0:e1e2:1::1/16
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these two things:
#1) Respect the privacy of others.
#2) Think before you type.
Password:
```

saat digunakan pertama kalinya, sudo akan menampilkan peringatan. Kemudian meminta password pengguna dan kemudian menjalankan perintah yang diinginkan (ifconfig dalam kasus ini) sebagai user root. Penggunaan selanjutnya dari sudo dalam beberapa menit tidak memerlukan mengetik ulang password. Hanya pengguna dengan hak istimewa admin dapat menggunakan sudo.

4.3. 6to4 Pada FreeBSD

FreeBSD menggunakan stf pseudo device untuk 6to4 tunneling. Sayangnya, perangkat ini tidak termasuk dalam kernel generik di FreeBSD 4.9, jadi jika Anda ingin menjalankan 6to4 dengan FreeBSD, Anda harus membangun sebuah kernel kustom. Baris yang perlu ditambahkan pada file konfigurasi kernel

pseudo-device stf

Setelah menginstal kernel baru, interface jaringan baru bernama stf0 harus tersedia. Selanjutnya membuat interface stf dengan perintah ifconfig stf create. Interface stf diaktifkan dengan memberi alamat IPv6 yang alamat 6to4 valid yang sesuai dengan alamat host IPv4. Dengan menambahkan default route ke relay 6to4, semua lalu lintas IPv6 non-lokal akan dikirim melalui 6to4. Listing 3-3 set kedua alamat (berdasarkan alamat IPv4

223.224.225.226) dan default route menuju alamat anycast estafet. Perintah-perintah ini harus dijalankan sebagai root.

Listing 3-3. Mengizinkan 6to4

```
# ifconfig stf0 inet6 2002:dfe0:e1e2:1::1/16
# route add -inet6 default 2002:c058:6301::
```

Listing 3-4. Menghapus Address dan Default Route

```
# ifconfig stf0 inet6 -alias 2002:dfe0:e1e2:1::1
# route delete -inet6 default 2002:c058:6301::
```

Listing 3-5. Mengizinkan 6to4 di */etc/rc.conf*

```
stf_interface_ipv4addr="223.224.225.226"
stf_interface_ipv6_ifid="0:0:0:1"
stf_interface_ipv6_slaid="0"
ipv6_defaultrouter="2002:c058:6301::"
```

Jika **stf_interface_ipv6_ifid** tidak didefinisikan, identifier interface bagian dari alamat 6to4 lokal akan ditetapkan ke 1, tanpa menentukan stf_interface_ipv6_slaid, jumlah subnet akan menjadi 0. File / etc /default / rc.conf berisi opsi konfigurasi tambahan untuk penjelasan rc.conf dan singkat.

4.4. 6to4 Pada LINUX

Di bawah Red Hat Linux 9, Anda dapat mengaktifkan 6to4 selama sistem startup dengan memastikan bahwa baris berikut yang hadir dalam file / etc / sysconfig / network:

```
NETWORKING_IPV6="yes"
IPV6_DEFAULTDEV="tun6to4"
```

Selain itu, isi dari file yang menentukan perilaku interface yang memasok alamat IPv4 (seperti **/etc/sysconfig/network-scripts/ifcfg-eth0** dalam kasus interface eth0) harus memiliki baris ini di dalamnya:

```
IPV6INIT=yes
IPV6TO4INIT=yes
```

Anda harus menjadi root untuk mengubah file-file ini. Dengan pengaturan ini, sistem akan menggunakan alamat IPv4 yang telah dikonfigurasi atau ditemukan dengan menggunakan DHCP untuk interface tersebut untuk membangun sebuah awalan 6to4 dan menginstal default semi-route di atas alamat estafet 6to4 anycast. Banyak pengaturan tambahan yang mungkin, dan mereka terdaftar dalam sysconfig.txt berkas yang ada di direktori **/usr/share/doc/initscripts-7.14/** atau satu berakhir di nomor versi yang sedikit berbeda.

Listing 3-6. Output netstat Command

```
# netstat -rn --inet6
Kernel IPv6 routing table
Destination                    Next Hop           Flags Metric Ref    Use Iface
::1/128                        ::                 U     0      0        0 lo
::/96                          ::                 U     256    0        0 tun6to4
2002:dfe0:e1e2:1::1/128        ::                 U     0      0        0 lo
2002::/16                      ::                 UA    256    0        0 tun6to4
::/0                           ::                 UDA   256    0        0 eth0
2000::/3                       ::192.88.99.1      UG    1      0        0 tun6to4
```

Pada pandangan pertama, route default tampaknya menunjuk pada interface eth0 (yang adalah karena "asumsi on-link" dibahas kemudian bab ini). Namun, ada juga route untuk 2.000 :: / 3, yang merupakan global ruang alamat unicast IPv6 keseluruhan sebagai saat ini didefinisikan oleh IANA: RIR blok awal (2001 :: / 16), blok 6bone (3ffe :: / 16), dan blok 6to4 (2002 :: / 16) adalah semua bagian dari blok yang lebih besar. 2000 :: / 3 route mengarah ke alamat anycast estafet 6to4. Karena 2000 :: / 3 route yang lebih spesifik dari :: / 0 default, yang satu ini diutamakan.

Menggunakan 2000 :: / 3 daripada :: / 0 ketika route default disebut adalah sedikit dari tradisi di Linux. Tampaknya bahkan ada beberapa versi Linux yang tidak akan menerima :: / 0 route dan memerlukan 2000 :: / 3 route sebaliknya ketika sistem dikonfigurasi sebagai router IPv6. Namun, hard-coding fakta bahwa Internet Assigned Numbers Authority sejauh ini hanya dialokasikan 2000 :: / 3 untuk digunakan sebagai ruang alamat unicast ke dalam sistem adalah bukan ide yang baik, sebagai bagian lain dari ruang alamat IPv6 dapat digunakan untuk unicast di masa depan juga. Setelah daftar blok alamat multicast dan tujuan khusus, RFC 3513 menyatakan:

> *"Future specifications may redefine one or more sub-ranges of the global unicast space for other purposes, but unless and until that happens, implementations must treat all addresses that do not start with any of the above-listed prefixes as global unicast addresses."*

Jika Anda ingin membuat Red Hat Linux sistem Anda RFC 3513-kompatibel, Anda dapat melakukan ini dengan mengubah ketiga kejadian 2000 :: / 3 :: / 0 di file / etc / sysconfig / network- scripts / network-fungsi -ipv6.

Manual manipulasi fungsi 6to4 paling baik dilakukan dengan menggunakan paket iproute yang datang dengan Red Hat. Kebanyakan distribusi Linux memiliki paket ini di papan, tetapi beberapa distribusi tidak atau bahkan kekurangan dukungan untuk netlink soket di kernel, yang digunakan oleh iproute (dan, opsional, oleh Zebra). Jika Anda adalah tipe petualang, Anda bisa menginstal iproute diri sendiri dan mengkompilasi sebuah kernel dengan dukungan netlink. Listing 3-7 membuat interface tunneling 6to4 dan default route menuju alamat estafet 6to4 anycast. Perintah-perintah ini harus dijalankan sebagai root.

Listing 3-7.Manual Konfigurasi 6to4

```
# ip tunnel add tun6to4 mode sit ttl 64 remote any local 223.224.225.226
# ip link set dev tun6to4 up
# ip -6 address add 2002:dfe0:e1e2:1::1/16 dev tun6to4
# ip -6 route delete ::/0
```

```
# ip -6 route add ::/0 via ::192.88.99.1
```

FreeBSD dan Cisco IOS membuat paket mengalir ke relay router 6to4 dengan menunjuk route ke alamat estafet IPv6, tetapi dengan Linux, route ke relay 6to4 harus ditentukan dalam bentuk IPv4- kompatibel, maka :: 192.88.99.1. Menghapus route default di kedua baris terakhir ini diperlukan untuk menghapus route default yang diinstal pada saat boot untuk menghasilkan pesan "destination unreachable".

Untuk informasi lebih lanjut tentang paket iproute dan perintah ip, kita lihat di file ip-cref.ps yang harus tersedia di / usr/share/doc/iproute-2.4.7/directory, atau satu dengan nama berdasarkan pada nomor versi yang sedikit berbeda. Ini adalah file PostScript, sehingga Anda perlu printer PostScript untuk mencetaknya atau perangkat lunak yang dapat menampilkan PostScript, seperti Ghostscript.

MASALAH KEAMANAN 6TO4

Secara teori, host harus mampu menangani semua paket yang salah dan paket yang berbahaya sehingga dibutuhkan filter atau firewall untuk menyaring paket yang tidak diinginkan.

Internet Service Provider memastikan hanya mengirim paket dari pelanggan ke seluruh internet jika paket tersebut memiliki alamat sumber milik pelanggan tersebut, ini disebut "anti-spoofing". Dengan filter anti-spoofing pelanggan masih dapat menyerang host lain di Internet , tapi paket yang terlibat dalam serangan tersebut mudah untuk ditelusuri dan disaring. 6to4 memungkinkan orang untuk membuat paket dengan alamat IPv6 palsu dan menyembunyikan dalam paket IPv4 yang sah, sehingga melewati filter anti-spoofing yang berlaku.

Untuk menghilangkan serangan yang menggunakan 6to4, kebanyakan sistem menyaring beberapa rentang alamat 6to4 yang tidak valid (lihat Daftar 3-9 dan 3-10). Selain itu, Di host 6to4 masa depan, router dan / atau relay akan menolak paket 6to4 yang memiliki header IPv4 yang tidak sesuai.

Monitoring 6to4

Listing 3-9 menunjukkan bagaimana ifconfig dan perintah netstat dapat digunakan untuk memantau perilaku FreeBSD atau MacOS X interface stf0 dan tabel routing IPv6.

Listing 3-9.Monitoring 6to4 bawah FreeBSD atau MacOS

```
# ifconfig stf0
stf0: flags=1<UP> mtu 1280
        inet6 2002:dfe0:e1e2:1::1 prefixlen 16
# netstat -rnf inet6
Destination        Gateway               Flags      Netif Expire
::/96              ::1                   UGRSc       lo0 =>
default            2002:c058:6301::      UGSc       stf0
::1                ::1                   UH         lo0
::ffff:0.0.0.0/96  ::1                   UGRSc      lo0
2002::/24          ::1                   UGRSc      lo0 =>
2002::/16          2002:dfe0:e1e2:1::1   Uc        stf0
2002:7f00::/24     ::1                   UGRSc      lo0
2002:dfe0:e1e2::1  link#7                UHL        lo0
2002:e000::/20     ::1                   UGRSc      lo0
2002:ff00::/24     ::1                   UGRSc      lo0
```

Flag netstat-r untuk membuat daftar tabel routing -n untuk mencari nama jaringan di DNS, dan-f memungkinkan kita untuk menentukan golongan alamat, di mana “inet6” berarti IPv6. Netstat output agak luas, bahkan tanpa informasi lokal link yang telah ditinggalkan dalam contoh. :: / 96 route memungkinkan penanganan khusus alamat IPv4 yang kompatibel untuk tujuan tunneling otomatis, dan alamat localhost (:: 1) route poin ke interface loopback. The :: ffff: 0.0.0.0/96 route ini terkait dengan alamat khusus yang memungkinkan program menggunakan API soket IPv6 untuk berkomunikasi melalui IPv4 (lihat Bab 6). 2002 :: / 24, 2002:7 F00 :: / 24, 2002: E000 :: / 20, dan 2002: ff00 :: / 24 route sesuai dengan 0.0.0.0 / 8, 127.0.0.0 / 8, 224.0.0.0 / 4, dan 255.0.0.0 / 8 prefiks, masing-masing. Ini adalah IPv4 blok alamat yang bukan sumber atau tujuan dari paket 6to4 yang valid.

2002: dfe0: e1e2 :: 1 router sesuai dengan alamat 6to4 lokal. 2002 :: / 16 router memastikan bahwa semua paket dengan tujuan 6to4 ditangani oleh interface stf0 untuk tunnel mereka langsung ke tujuan mereka atas IPv4. Yang terakhir, route default mengarahkan semua paket yang tersisa ke 6to4 anycast alamat estafet 2002: c058: 6301 :: untuk merelay.

Listing 3-10 daftar tabel routing IPv6 di Linux dengan perintah ip. Ini memberikan lebih banyak informasi dari perintah netstat.

Listing 3-10.Monitoring 6to4 bawah Red Hat Linux 9

```
# ip -6 route
::/96 via :: dev tun6to4  metric 256  mtu 1480 advmss 1420
unreachable ::/96 dev lo  metric 1024  error -101 mtu 16436 advmss 16376
unreachable ::ffff:0.0.0.0/96 dev lo  metric 1024  error -101 mtu 16436 advmss 16376
unreachable 2002:a00::/24 dev lo  metric 1024  error -101 mtu 16436 advmss 16376
unreachable 2002:7f00::/24 dev lo  metric 1024  error -101 mtu 16436 advmss 16376
unreachable 2002:a9fe::/32 dev lo  metric 1024  error -101 mtu 16436 advmss 16376
unreachable 2002:ac10::/28 dev lo  metric 1024  error -101 mtu 16436 advmss 16376
unreachable 2002:c0a8::/32 dev lo  metric 1024  error -101 mtu 16436 advmss 16376
unreachable 2002:e000::/19 dev lo  metric 1024  error -101 mtu 16436 advmss 16376
2002::/16 dev tun6to4  proto kernel  metric 256  mtu 1480 advmss 1420
unreachable 3ffe:ffff::/32 dev lo  metric 1024  error -101 mtu 16436 advmss 16376
default via ::192.88.99.1 dev tun6to4  metric 1024  mtu 1480 advmss 1420
```

Perbedaan lain dari FreeBSD adalah bahwa Linux mengimplementasi filter lebih valid prefiks IPv4: 10.0.0.0 / 8, 169.254.0.0/16, 172.16.0.0/12, 192.168.0.0/16, dan 224.0.0.0 / 3. Perhatikan bahwa route-route / filter tidak muncul baik dalam output netstat-r atau aturan firewall iptables. Rupanya, para pengembang Linux juga merasa perlu untuk menyaring 3ffe: ffff :: / 32, yang terakhir / 32 blok di awalan 6bone, yang sering digunakan dalam contoh. Namun, Red Hat ES4 tidak menginstal salah satu route yang tidak terjangkau.

KONFIGURASI TUNNELS MANUAL

Tunnels dikonfigurasi secara manual atau tunnels *point-to-point* dapat mengangkut paket *multicast*. Selain memungkinkan penggunaan aplikasi multicast, *tunnels* juga memungkinkan untuk mekanisme konfigurasi manual IPv6 reguler dan protokol routing.

WINDOWS

Listing 3-11. Konfigurasi Tunnel manual

```
C:\>netsh
netsh>interface ipv6
netsh interface ipv6>add v6v4tunnel interface=tun0 localaddress=192.0.2.1
remoteaddress=223.224.225.226
netsh interface ipv6>add address interface=tun0 address=2001:db8:31:1::2
Ok.

netsh interface ipv6>add route prefix=2001:db8:31:1::/64 interface=tun0
Ok.

netsh interface ipv6>add route prefix=::/0 interface=tun0 nexthop=2001:db8:31:1::1
Ok.

netsh interface ipv6>quit
```

Dari konfigurasi diatas, *netsh* dimulai tanpa argument, sehingga menunggu perintah. Perintah yang tersedia dalam beberapa konteks *interface* ipv6. Pilihan *interface* dengan mengambil nama *tunnels* sebagai argumen, dalam hal ini, *tun0*. *Local address* dan *remote address* menentukan alamat lokal dan remote yang akan digunakan untuk tunnels ini (endpoint).

Perintah selanjutnya yaitu membentuk sebuah alamat interface baru. Tidak seperti sistem lain, Windows tidak secara otomatis terhubung langsung ke route sistem di subnet yang sama, sehingga route harus dikonfigurasi ke dirinya sendiri. Hal ini menjelaskan bahwa perintah alamat *add* tidak mengambil argumen yang menentukan panjang prefiks. Setelah membuat *route subnet* yang tidak termasuk panjang prefiks subnet, default route menuju router di ujung tunnels. Meskipun opsi nexthop pada alamat IPv6 dari router sebagai argumen adalah opsional, tanpa ini route default tidak akan bekerja. Tergantung pada paket layanan yang terinstal, sistem sekarang dapat merespon saat sistem lain mencoba untuk ping6 dengan Service Pack 1, Windows XP tidak membalas ping IPv6 atau Traceroute. Dengan Service Pack 2 dapat di-*ping* tetapi tidak untuk *Traceroute.*

Listing 3-12 Menghapus route, alamat, dan interface diciptakan pada Listing 3-11.

Listing 3-12. *Removing a Manual Tunnel*

```
C:\>netsh
netsh>interface ipv6
netsh interface ipv6>delete route prefix=::/0 interface=tun0 nexthop=2001:db8: ➥
31:1::1
Ok.

netsh interface ipv6>delete route prefix=2001:db8:31:1::/64 interface=tun0
Ok.

netsh interface ipv6>delete address interface=tun0 address=2001:db8:31:1::2
Ok.

netsh interface ipv6>delete interface interface=tun0
Ok.
```

Meskipun ada beberapa cara untuk mendirikan sebuah tunnels pengguna di Linux, namun hanya satu yang menghasilkan hasil yang diinginkan yaitu perintah ip. Listing 3-19 menciptakan dan mengkonfigurasi *new tunnel device tun0* dengan ip.

Listing 3-19. Membuat *Manual Tunnel* Pada *Linux*

```
# ip tunnel add name tun0 mode sit local 192.0.2.1 remote 223.224.225.226 ttl 64
# ip link set dev tun0 up
# ip address add 2001:db8:31:1::2/64 dev tun0
```

Jenis tunnels Simple Internet Transition (SIT) digunakan untuk semua tunnels IPv6-in-IPv4, termasuk 6to4. Perbedaan antara sebuah tunnels 6to4 dan tunnels pada alamat remote yaitu setiap tunnels 6to4 diatur ke nilai tertentu untuk tunnels dikonfigurasi secara manual.

Listing 3-20. Menambahkan Default Route dan Menampilkan Tabel Routing dengan ip

```
# ip route add default via 2001:db8:31:1::1 metric 15
# ip -6 route show
2001:db8:31:1::/64 via :: dev tun0  proto kernel  metric 256  mtu 1480 advmss 1420
fe80::/64 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
fe80::/64 via :: dev tun0  proto kernel  metric 256  mtu 1480 advmss 1420
default via 2001:db8:31:1::1 dev tun0  metric 15  mtu 1480 advmss 1420
default dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
unreachable default dev lo  metric -1  error -101
```

Menambah dan menampilkan route juga dapat dilakukan dengan menggunakan route dan perintah netstat, seperti ditunjukkan pada Listing 3-21.

Listing 3-21. Menambahkan Default Route dan Menampilkan Tabel Routing dengan route netstat

```
# route -A inet6 add default gw 2001:db8:31:1::1
# netstat -rnA inet6
Kernel IPv6 routing table
Destination                    Next Hop           Flags Metric Ref Use Ifac
::1/128                        ::                 U     0      0     0 lo
2001:db8:31:1::2/128           ::                 U     0      0     0 lo
2001:db8:31:1::/64             ::                 UA    256    1     0 tun0
fe80::/64                      ::                 UA    256    0     0 eth0
fe80::/64                      ::                 UA    256    0     0 tun0
::/0                           2001:db8:31:1::1   UG    1      0     0 tun0
::/0                           ::                 UDA   256    0     0 eth0
```

Kedua route dan perintah netstat mengambil *-A inet6 flag* yang menentukan IPv6 "domain address". Argumen *gw* menunjukkan bahwa nilai berikutnya adalah alamat Hop berikutnya. Route default memastikan bahwa semua paket dikirim ke router dengan alamat ini. Pada route, tidak ada untuk menentukan metriks, seperti route yang dibuat dengan perintah ini menerima metric 1 secara default. Dua *flag* tambahan ditentukan untuk *netstat -r* dan *-n*. Yang pertama memberitahu netstat untuk menampilkan tabel routing (default untuk menunjukkan saat TCP, UDP, dan soket UNIX), dan yang terakhir menonaktifkan *DNS lookup* untuk entri tabel *routing*.

Perbedaan antara *ip route* dan *output netstat-r* sebagian besar konsekuensi kecil (tidak terdokumentasi), kecuali netstat yang menunjukkan *route host* yang diciptakan untuk alamat sistem sendiri perlu untuk melacak.

Listing 3-22. Menghapus route, alamat IPv6, dan tunnels interface.

Listing 3-22. menghapus route, alamat IPv6, dan tunnels interface.

```
ip route delete default via
2001:db8:31:1::1 ip address del
2001:db8:31:1::2/64 dev tun0
ip tunnel del name tun0
```

Listing 3-23. Isi dari /etc/sysconfig/network

```
NETWORKING_IPV6="yes" IPV6_DEFAULTDEV=sit1
IPV6_DEFAULTGW=2001:db8:31:1::1
```

Listing 3-24. Isi dari /etc/sysconfig/network-scripts/ifcfg-sit1

```
DEVICE=sit1
BOOTPROTO
=none
ONBOOT=yes
IPV6INIT=yes
IPV6TUNNELIPV4=192.0.2.1
IPV6TUNNELIPV4LOCAL=223.224.225.226
IPV6ADDR=2001:db8:31:1::2
```

Perhatikan nama tunnel interface harus SIT dan digit, namun tidak boleh *sit0*, dan IPv6TUNNELIPv4 menentukan remote tunnel alamat endpoint.

Tunnels dan NAT Dikonfigurasi secara manual

Tunnels dikonfigurasi secara manual dapat menangani *Network Address Translation* (NAT) tanpa masalah. Jika konfigurasi pada kedua ujung diubah untuk mencerminkan gagasan setiap *end* memiliki alamat sendiri dan alamat remote akhir . Sebagai contoh, mari diasumsikan sebuah tunnel antara 192.0.2.1 dan 223.224.225.226. Sekarang host yang digunakan untuk memiliki alamat 223.224.225.226 dipindahkan di belakang NAT dan diberikan baru (privat) yaitu alamat 10.0.0.203. Alamat lama diberikan ke kotak NAT. Untuk menjaga kinerja tunnel, konfigurasi pada host dibelakang NAT harus mencerminkan bahwa ia memiliki *private address*, sehingga sumber tunnel menjadi 10.0.0.203. Tetapi konfigurasi pada host lain tetap sama.

Namun, kebanyakan implementasi NAT dapat bekerja hanya dengan TCP dan UDP dan gagal untuk menangani paket IP dengan protokol 41 payload. Sebuah kelas yang lebih besar dari NAT dapat menangani forward protokol 41 paket menuju alamat internal tetap, yang tentunya menjadi host titik akhir tunnel. Item konfigurasi ini sering disebut "*default host*" atau "DMZ." Sebuah NAT dapat menangani IPv6-in-IPv4 tunneling tanpa konfigurasi apapun. Namun, tidak ada cara yang baik untuk mengetahui apa jenis NAT diimplementasikan dalam perangkat tertentu selain hanya melihat apa yang terjadi.

MENDAPATKAN SEBUAH TUNNEL

Setelah semua cara membuat tunnel dikonfigurasi secara manual, hanya satu bahan yang hilang untuk setup tunnel sukses disisi lain. Dalam hal ini, ada dampak baik dan dampak buruk. Dampak baiknya adalah bahwa banyak tempat di seluruh tawaran koneksi tunnel Bersih. Kabar buruknya adalah bahwa banyak dari mereka yang sulit untuk menemukan. Hal ini terutama berlaku untuk ISP yang menawarkan tunnel, seperti IPv6 bukan bisnis tepatnya besar (belum) untuk ISP. Namun, sebuah tunnel dari ISP memiliki keuntungan yang cukup. Pertama, karena lalu lintas mengalir melalui jaringan mereka pula, tidak ada jalan memutar untuk lalu lintas IPv6. Kedua, mereka tahu

Anda, sehingga biasanya mungkin untuk mendapatkan sebuah tunnel tetap tanpa perlu untuk sistem otentikasi. Ketiga, mendapatkan lalu lintas IPv6 dari ISP Anda tidak membiayai mereka setiap bandwidth ekstra. Mendapatkannya dari pihak ketiga berarti bahwa semua IPv6 yang berlalu lintas ke dan dari Anda perlu mengalir melalui jaringan mereka, dan memberikan ini secara gratis bukanlah sebuah model bisnis yang berkelanjutan dalam jangka panjang.

Namun, jika ISP Anda tidak mendukung IPv6, Anda harus mendapatkan sebuah tunnel di tempat lain. 6bone sedang dihapus, sehingga saran lama menghubungkan ke 6bone dengan tunnel yang masih mengapung di sekitar Net tidak lagi sangat berguna. Kebanyakan, jika tidak semua, broker tunnel memberikan tunnel IPv6 untuk masyarakat umum secara gratis. Namun, mereka semua mendekati tugas ini secara berbeda. Sebagian besar membutuhkan beberapa jenis pendaftaran, dan beberapa menggunakan perangkat lunak klien khusus untuk mengatur tunnel. Jumlah ruang alamat mereka memberikan variasi, dari satu alamat ke penuh / 48. Tabel 3-1 mencantumkan beberapa broker tunnel terkenal. Itu bernilai baik waktu untuk menemukan satu yang sesuai dengan kebutuhan Anda daripada untuk memilih hanya satu acak.

Tabel 3-1. Broker Tunnel

Name	URL	Location
Hurricane Electric IPv6 Tunnel Broker	http://www.tunnelbroker.net/	Fremont, California
Hexago Freenet6	http://www.hexago.com/	Quebec, Canada
Consulintel Tunnel Broker	http://tb.consulintel.euro6ix.org/	Madrid, Spain
SixXS Tunnel Broker	http://www.sixxs.net/	Various locations in Europe