BAB 7 - TRANSISI

Sejak awal, IETF menyadari bahwa tidak mungkin untuk seluruh Internet untuk beralih dari IPv4 ke IPv6 pada yang telah ditentukan "hari bendera." Kesadaran ini menyebabkan perkembangan beberapa mekanisme transisi (lihat RFC 2893), yang secara kasar dapat dikelompokkan menjadi tiga kategori:

1. Dual stack (juga disebut “dual layer” or “Dual Stack Transition Mechanism,” DSTM)
2. Tunnel
3. Translasi dan Proxying

Sejauh ini, kami telah diasumsikan bahwa kemampuan IPv6 akan menjadi tambahan kemampuan IPv4. Jelas, di beberapa titik, IPv4 harus dimatikan, seperti menjalankan dual stack tanpa batas tidak memecahkan masalah. Sebaliknya, itu membuat segalanya (sedikit) sulit, karena itu masih perlu untuk melakukan segala sesuatu yang harus dilakukan untuk mendukung IPv4 seperti sebelumnya dan menambahkan IPv6 di atas itu. Di sisi lain, itu sangat menarik untuk menjalankan dual stack, karena hal ini hanya menambah kemampuan baru tanpa mengambil yang sudah ada jauh.

Planning the Transition

Deplesi Alamat IPv4 dan Rasio HD

Pada titik tertentu, itu hanya lebih mudah untuk membuang tabung dan membuka yang baru. Ruang alamat adalah persis seperti itu: di beberapa titik, upaya yang diperlukan untuk mengelola ruang alamat semakin banyak digunakan-up menjadi terlalu besar, sehingga lebih mudah untuk memperluas itu. Dalam RFC 3194, Alain Durand dan Christian Huitema datang dengan nomor yang mengungkapkan penggunaan address dengan cara yang memungkinkan untuk menarik kesimpulan pada pengalaman masa lalu.

Jadi, jika suatu organisasi memiliki jaringan kelas B tua dengan 65.536 alamat di dalamnya, dan 4096 dari alamat tersebut sedang digunakan, rasio HD akan log (4096) / log (65536) = 12/16 = 0,75 atau 75%. Dalam contoh ini, dasar untuk logaritma adalah dua, tapi dasar apapun dapat digunakan karena rasio antara logaritma yang penting, bukan nilai absolut. Setelah melihat berbagai jenis alamat seperti Perancis dan nomor telepon Amerika Utara dan jaringan DECnet seluruh dunia, Durand dan Huitema menyimpulkan bahwa rasio HD dari 80% atau disamakan lebih rendah ke tingkat yang nyaman, dan pada 87% atau lebih tinggi, ruang alamat menjadi begitu sulit untuk mengelola bahwa panjang alamat diperluas, atau teknik untuk mengurangi penggunaan alamat dikerahkan.

$$HD = \frac{\log(\text{addresses used})}{\log(\text{total addresses})}$$

IPv6 vs.Network Address Translation

NAT sudah digunakan secara luas saat ini, dan bekerja sangat baik untuk aplikasi client / server, seperti web dan email, di mana klien berada di belakang NAT. Tapi NAT tidak bekerja dengan baik untuk aplikasi peer-to-peer seperti Voice over IP dan aplikasi client / server, di mana lebih dari satu server di belakang perangkat

NAT, dan itu sangat membatasi aplikasi baru mungkin, seperti otomatisasi rumah atau komputasi di mana-mana. Untuk membuat NAT benar-benar berguna sebagai teknik konservasi alamat, itu akan memungkinkan beberapa server / rekan-rekan untuk berbagi satu alamat IP. Karena jelas dua server yang berbeda atau aplikasi peer -to-peer pada alamat IP yang sama tidak dapat berbagi sejumlah port tunggal, ini berarti akhir dari "well-known port" konsep. Pada dasarnya, TCP atau UDP nomor port akan menjadi bagian dari alamat. Perubahan yang diperlukan untuk membuat karya ini mungkin serupa dalam skala yang dibutuhkan untuk membuat pekerjaan IPv6.

Aplikasi Skenario Transisi

Ketika mempertimbangkan pelaksanaan mekanisme transisi seperti terjemahan dan proxy, penting untuk diingat model komunikasi yang berbeda yang digunakan oleh berbagai aplikasi. Misalnya, klien email hanya berkomunikasi dengan server tertentu, dan berkomunikasi server antara satu sama lain, seperti yang digambarkan dalam Gambar 7-1. Mari kita sebut ini "model email."

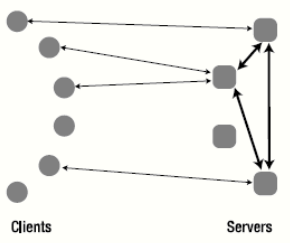

Figure 7-1. *The email communication model*

Dalam model email, itu sangat mudah bagi klien untuk bermigrasi dari IPv4 ke IPv6, asalkan server tertentu klien menggunakan dual stack, maka tidak akan ada masalah. Namun, karena setiap server berpotensi perlu berkomunikasi dengan server lain, server terakhir harus menjadi dual stack sebelum server pertama dapat pindah ke IPv6-only. odel komunikasi untuk Web, server Web tidak berkomunikasi satu sama lain, semua komunikasi antara klien dan server. Berbeda email, klien tidak berkomunikasi dengan server tertentu, tetapi dengan server apapun. Lihat Gambar 7-2 untuk "model Web."

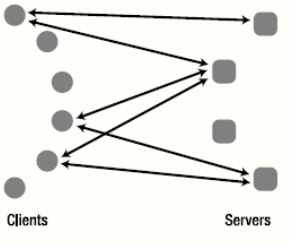

Figure 7-2. *The Web communication model*

Dalam model web , semua server yang merupakan bagian dari World Wide Web harus dual stack sebelum klien pertama dapat pindah ke IPv6 -only . Tentu saja , setumpuk klien ganda masih bisa berkomunikasi melalui IPv6 dengan tumpukan server yang ganda , dan jaringan pribadi dapat pindah ke IPv6 -only tanpa menunggu Web pada umumnya untuk melakukan hal yang sama . Model ketiga adalah rekan Model -untuk -peer . Ada banyak aplikasi peer-to -peer , dan mereka tidak semua berkomunikasi dengan cara yang sama . Namun dalam banyak kasus , ada server dari beberapa jenis yang berkomunikasi dengan server lain . Klien berkomunikasi dengan server tetapi juga dengan klien lain , maka moniker

peer-to -peer untuk kelompok aplikasi . Gambar 7-3 menunjukkan model komunikasi peer-to -peer umum .

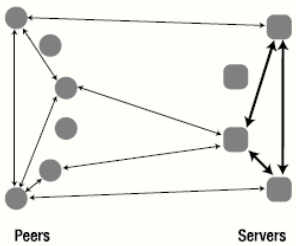

Figure 7-3. *Peer-to-peer communciation*

Pada model Web, semua server yang menjadi bagian dari World Wide Web harus dalam kondisi dual-stack sebelum client pertama dapat berpindah ke Ipv6 saja. Tentunya, client dual stack teta dapat berkomunikasi dengan Ipv6 melalui server dual stack, dan private network dapat berpindah ke Ipv6 saja tanpa harus menunggu Web melakukan hal yang sama.
Model ketiga adalah model peer-to-peer. Terdapat banyak aplikasi peer-to-peer, dan tidak semua aplikasi tersebut berkomunikasi dengan cara yang sama. Tetapi dalam kasus umumnya, terdapat server dari jenis tertentu yang berkomunikasi dengan server lainnya. Client tidak hanya berkomunikasi dengan server, tetapi juga dengan client lainnya. Gambar 7-3 menunjukkan suatu model umum komunikasi peer-to-peer.

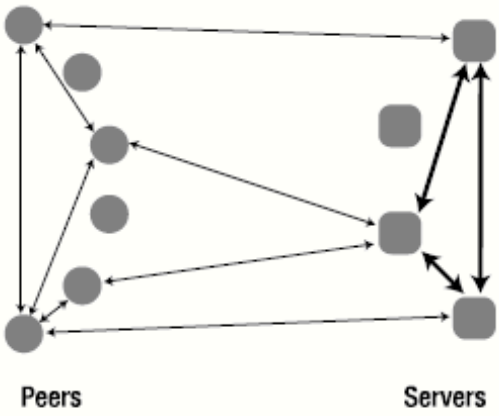

Figure 7-3. *Peer-to-peer communciation*

Meskipun aplikasi peer-to-peer yang berbeda menggunakan model komunikasi yang sama,
terdapat perbedaan yang sangat penting antara berbagai jenis aplikasi peer-to-peer. Salah satu jenis, yang meliputi Voice over IP (VoIP), membutuhkan client yang diberikan untuk
berkomunikasi dengan client lain yang spesifik. Ketika terkoneksi melalui Internet, kita hanya tertarik untuk berbicara dengan orang yang kita panggil, tidak ada pergantian pemain. Dengan aplikasi file sharing, seperti BitTorrent, di sisi lain, tidak ada persyaratan untuk dapat berkomunikasi dengan client tertentu. Perlu diketahui bahwa tracer (BitTorrent untuk server yang mengkoordinasikan komunikasi awal antar client) dan sejumlah client harus dual stack.

Proxying

Meskipun banyak aplikasi yang menawarkan penggunaan proxy, tetapi proxy sebagian besar
digunakan untuk HTTP dan FTP. Untuk sebagian besar aplikasi lain atau protokol, ketersediaan penggunaan proxy dalam aplikasi dan IPv6, software proxy adalah sedemikian rupa sehingga proxy adalah tidak mungkin atau tidak layak kesulitan. Web dan FTP proxy yang
paling popular adalah Squid, tapi sayangnya, Squid tidak memiliki built-in yang support pada penggunaan IPv6. Alternatif yang ditawarkan adalah Apache, yang sudah dijelaskan pada bab sebelumnya. Sebuah fitur yang sangat bagus dari kumpulan HTTP / HTTPS / FTP proxy adalah bahwa hal tersebut memungkinkan client IPv6 untuk menyambung ke Web IPv4 dan juga client IPv4 dapat terhubung ke IPv6.

Apache sebagai Proxy

Untuk mendapatkan proxy, diperlukan pengkompilasian Apache 2 dengan dukungan beberapa modul tambahan. Listing 7-2, ketika di-execute dalam direktori sumber Apache, membangun Apache dengan dukungan SSL dan beberapa pilihan proxy. Dukungan SSL tidak
diperlukan untuk proxy.

Listing 7-2. *Building Apache with Proxy Support*

```
make clean
./configure --enable-so --enable-ssl --enable-mods-shared="proxy proxy_http proxy_➥
ftp proxy_connect auth_digest"
make
make install
```

List program diatas digunakan untuk menghapus file yang tersisa dari konfigurasi yang sebelumnya telah dibangun. Perhatikan bahwa baris konfigurasi diatas memungkinkan sharing objek yang support. Yang membedakan hanya shared modul proxy secara umum: HTTP, FTP, HTTPS, proxy dan autentifikasi. Jika Apache anda tidak support dengan SSL, maka anda harus keluar dari fungsi enable-options. Jika Apache sudah berjalan, hentikan dengan apachectl stop, sebelum menghentikan fungsi Apache, buat installcand, kemudian mulai Apache new built-in dengan fungsi apachectl start. Listing 7-3 menunjukkan syntax yang harus ditambahkan ke file konfigurasi Apache untuk dapat mengaktifkan proxy.

Listing 7-3. *Configuring Apache to Be a Proxy*

```
LoadModule  proxy_module          modules/mod_proxy.so
LoadModule  proxy_http_module     modules/mod_proxy_http.so
LoadModule  proxy_ftp_module      modules/mod_proxy_ftp.so
LoadModule  proxy_connect_module  modules/mod_proxy_connect.so
LoadModule  auth_digest_module    modules/mod_auth_digest.so

ProxyRequests On

<Proxy *>
  Order allow,deny
  Allow from 2001:db8::/32 192.0.2.0/24 example.com
</Proxy>
```

Pada baris ProxyRequests On, memungkinkan permintaan penggunaan proxy. Tetapi sebelum hal tersebut dilakukan, shared object yang diperlukan dalam konfigurasi harus dimuat terlebih dahulu. Selain itu, tidak disarankan ketika pada konfigurasi dijalankan proxy tanpa penggunaan jenis kontrol akses. Spam open-proxy memungkinkan mereka untuk spam ke seluruh koneksi internet, hal ini dikarenakan proxy menyembunyikan alamat mereka. Untuk alasan yang sama, anti-spam group tidak mendukung adanya open-proxy, sehingga mereka menempatkan open-proxy kedalam blacklist. Banyak orang menggunakan blacklist untuk memblokir email dari host yang menjalankan/support open-proxy, sehingga server anda tidak akan termasuk kedalam golongan blacklist.

Transport Protocol Translation (TRT)

Kelemahan proxy adalah memerlukan banyak pengetahuan tentang protocol, layer aplikasi seperti HTTP dan FTP. kemampuan ini memungkinkan untuk fitur tambahan seperti caching, jadi bentuk yang lebih sederhana dari terjemahan antara IPv6 dan IPv4 dapat digunakan :

Transport Relay Terjemahan (TRT, RFC3142).

Sebuah implementasi TRT mendengarkan session masuk TCP (dan kadang-kadang UDP) di sisi IPv6. Ketika sesi masuk, perangkat TRT set-up koneksi TCP ke alamat IPv4 dikodekan dalam 32 bit dari alamat tujuan IPv6 dan kemudian melanjutkan untuk relay data antara dua sesi TCP. host IPv6 yang berasal sesi pertama hanya berisi IPv6, host tujuan IPv4 hanya berisi IPv4, terjadi komunikasi copy data antara kedua sesi TCP.

DNS ALG: Trick-or-Treat Daemon

Dalam instalasi kecil dimana hanya beberapa alamat IPv4 spesifik harus dibuat melalui IPv6 yang tersedia, cara itu paling mudah untuk menambahkan alamat IPv6 yang terhubung ke IPv4 tujuan melalui TRT ke zona DNS. Misalnya, seluruh jaringan IPv6 kecuali untuk perangkat yang harus dikelola dengan SSH. Perangkat ini memiliki alamat 192.0.2.25, dan awalan 2001: db8:31:6464:: /96. Alamat TRT untuk perangkat tersebut kemudian akan menjadi 2001: db8: 31:6464 :: 192.0.2.25 atau 2001: db8:31:6464 ::C000:219, dan alamat ini dapat dimasukkan ke dalam DNS di bawah domain yang tepat. Namun, hal ini tidak bekerja dengan baik sebagai mekanisme umum di mana IPv6 menggunakan TRT untuk berkomunikasi dengan IPv4. Disinilah layer aplikasi gerbang masuk DNS-DNS ALG, penyadapan DNS lookup oleh IPv6. Ketika permintaan untuk note AAAA tidak dapat dipenuhi karena tidak cukup note AAAA, DNS ALG me-record dan membuat note AAAA palsu dengan menggabungkan record dengan awalan TRT.

Faith on FreeBSD

Mungkin implementasi pertama dari TRT adalah *faith on IPv6 Kame stack*, yang merupakan dasar dari implementasi IPv6 di FreeBSD dan anggota lain dari keluarga BSD. Faith TRT terdiri dari dua bagian : faith network interface, dan daemon faithd. FreeBSD biasanya memiliki antarmuka faith0, dan interface faith yang baru dapat dibuat dengan faith ifconfig. antarmuka untuk melakukan pekerjaan yang berguna, itu perlu untuk mengaktifkannya dengan menggunakan pengaturan *sysctl*, dan awalan TRT harus diteruskan ke antarmuka.

Karena FreeBSD tidak memiliki cara yang langsung untuk rute awalan menuju

antarmuka, itu paling mudah untuk mengkonfigurasi alamat di awalan TRT pada interface faith.

Network Address Translation–Protocol Translation

Mekanisme yang lebih umum daripada transportasi translasi layer Stateless IP/ICMP Translation (SIIT) sebagaimana ditentukan dalam RFC 2765. Seperti namanya menunjukkan, SIIT tidak memerlukan implementasi untuk menjaga informasi apa jenis komunikasi yang terjadi. Namun tidak perlu mengetahui alamat IPv4 yang akan diterjemahkan ke alamat IPv6 dan sebaliknya. Ini berarti bahwa SIIT sendiri berguna dalam jumlah yang sangat terbatas sehingga itu umumnya digunakan sebagai PT bagian dalam NAT-PT: Network Address Translation-Protokol Translation. NAT-PT menambahkan perilaku IPv4 NAT di atas SIIT sehingga host IPv6 hanya dapat menghubungkan ke alamat IPv6 sintetik yang dihasilkan oleh DNS ALG. Bagian NAT
kemudian akan melacak alamat mana dan port kombinasi bersama-sama sehingga PT atau SIIT bagian dapat melakukan terjemahan yang sebenarnya. Karena NAT-PT bekerja pada lapisan IP, itu setidaknya secara teoritis untuk menjaga IPsec ESP utuh saat menerjemahkan. Selain itu, ia menyediakan cukup banyak layanan yang sama seperti TRT.