Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi
Vol. 1, No. 2, Juli 2005 ISSN 0216 - 0544

PERANAN NETWORK ADDRESS TRANSLATION DALAM MANAJEMEN JARINGAN IP

Husni, S.Kom.

Jurusan Teknik Informatika
Universitas Trunojoyo

husni@trunojoyo.ac.id

ABSTRAK

Teknologi jaringan komputer semakin maju, termasuk di Indonesia. Jumlah pengguna Internet meningkat drastis meskipun penggunanya masih didominasi oleh kaum terdidik terutama kelas pendidikan tinggi. Peningkatan jumlah komputer yang terkoneksi ke Internet berarti peningkatan kebutuhan IP address publik yang dikenal di Internet dan ini merupakan masalah tersendiri. **Network Address Translation** (NAT) hadir sebagai solusi untuk penghematan IP address ini. Hanya komputer gateway yang memerlukan IP publik, host-host dibawahnya dapat menumpang menggunakan IP private yang tidak dikenal di Internet. Selain sebagai solusi penghematan IP address sekaligus penghematan cost, NAT juga dapat digunakan untuk membangun layanan jaringan yang termanage rapi. Fitur-fitur seperti load balancing, virtual server atau sekedar IP translation semakin diminati saat ini terutama pada perusahaan skala menengah ke atas. Berbagai teknik digunakan untuk mengimplementasikannya, pada berbagai sistem operasi dan device.

***Kata kunci** : NAT, IP address, virtual server, load balancing, routing*

1. PENDAHULUAN

Awalnya Network Address Translation (NAT) digunakan sebagai salah satu solusi penghematan IP address yang telah menipis persediaannya. Sekarang NAT terbukti tangguh dan telah digunakan secara lengkap pada berbagai bidang jaringan komputer berbasis IP dan mungkin berbagai bentuk aplikasi lain dari NAT akan muncul kemudian. Makalah ini mencoba menjelaskan peranan NAT saat ini dan mungkin pula di masa akan datang, karena NAT bukan hanya sebuah solusi jangka pendek, life cycle dari NAT diperkirakan masih panjang bahkan bersama dengan implementasi IP Next Generation (IPng atau IPv6). Eksperimen yang telah dilakukan menunjukkan bahwa protokol IPv6 sendiri tidak menyebabkan banyak masalah kompatibilitas sehingga migrasi NAT dapat berjalan dengan cepat walaupun beberapa aplikasi masih menggunakan standar IPv4 dan dapat memunculkan masalah saat migrasi – bukan karena NAT. Umur IPv4 sendiri masih cukup panjang, setidaknya untuk Local Area Network (LAN).

Secara umum NAT dapat dibagi ke dalam dua kategori. Pertama disebut NAT klasik, NAT orisinil sebagaimana ditemukan pada awal 1990-an dan tercatat pada RFC-1631. Tugas utama NAT ini adalah menghemat ruang IP address di Internet. Kedua adalah NAT yang telah berubah lebih fleksibel, lebih luas domainnya dan merujuk ke definisinya – bertugas mentranslasi IP address.

2. TEKNIK NAT KLASIK

Translasi IP address dapat dilakukan secara statis maupun dinamis. Pada NAT statis IP address orisinil selalu ditranslasi ke IP address NAT yang sama (tetap) – sepanjang waktu. Sedangkan pada NAT dinamis translasi ke IP NAT tergantung pada berbagai kondisi *runtime* dan mungkin berbeda-beda untuk setiap koneksi.

Agar lebih mudah dipahami, dua variabel *m,n* didefinisikan sebagai berikut:

m: jumlah IP address yang perlu ditranslasi (IP orisinil)

n: jumlah IP address yang tersedia sebagai tujuan translasi (IP NAT)

2.1 Translasi Address Statis

Translasi - m:n, m,n>=1 dan *m=n* (*m,n* dalam *N*)

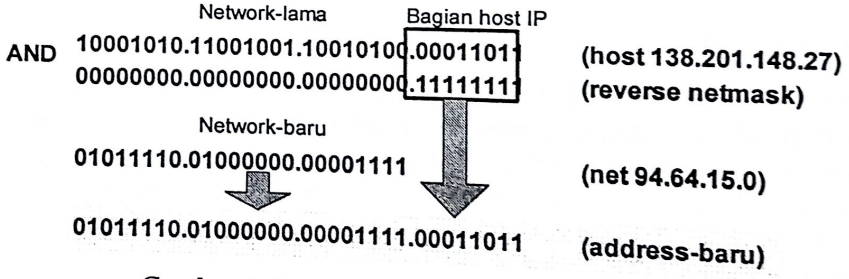
Dengan *static address translation* kita dapat melakukan translasi antar jaringan IP yang berisi mempunyai ukuran sama (berisi jumlah IP sama). Kasus khusus adalah saat kedua jaringan hanya satu IP, misalnya diperlihatkan oleh bentuk netmask 255.255.255.255. Strategi NAT ini mudah diimplementasi karena proses translasi secara keseluruhan dapat ditulis sebagai satu baris berisi transformasi logika sederhana :

```
Address-baru = network-baru OR (address-lama AND (NOT netmask))
```

Tidak ada informasi tentang status koneksi yang sedang ditranslasi yang perlu dicatat, cukup melihat setiap paket IP secara individu. Koneksi dari jaringan luar (*outside*) ke host-host di dalam jaringan (*inside*) tidak mengalami masalah, host-host tersebut hanya mempunyai IP berbeda dengan yang dipakai pada *inside* sehingga NAT statis dapat dikatakan hampir transparan.

Contoh:

- Aturan NAT : translasikan semua IP dalam network 138.201.148 ke IP dalam network 94.64.15, netmask untuk kedua network adalah 255.255.255.0
- Sekarang 138.201.148.1 ditranslasi ke 94.64.15.1, 138.201.148.27 ditranslasi ke 94.64.15.27, dan seterusnya



Gambar 1. Proses translasi IP pada NAT statis

2.2 Translasi Address Dinamis

Translasi - m:n, m>=1 dan *m>=n* (*m,n* dalam *N*)

Dynamic address translation diperlukan saat jumlah IP yang akan ditranslasi tidak sama dengan jumlah IP tujuan translasi, atau jumlahnya sama tetapi karena suatu alasan tidak akan menerapkan pemetaan statis. Jumlah host yang berkomunikasi secara umum (pada NAT statis) dibatasi oleh jumlah IP NAT yang tersedia. Saat semua IP NAT telah digunakan maka tidak ada lagi koneksi yang dapat ditranslasi dan karena itu paket akan di tolak oleh router-NAT, misalnya dengan mengembalikan pesan **'host unreachable'**. NAT dinamis lebih kompleks dibanding NAT statis karena sistem harus mencatat *track* komunikasi setiap host dan mungkin pula harus melihat informasi TCP dalam paket-paket data.

NAT dinamis juga bermanfaat saat tersedia cukup IP NAT - seperti telah disebutkan, misalnya saat m = n. Sebagian orang menggunakan ini sebagai sebuah ukuran keamanan (*security measure*), adalah tidak mungkin bagi seseorang di luar jaringan (*outside*) dapat terkoneksi ke host-host di

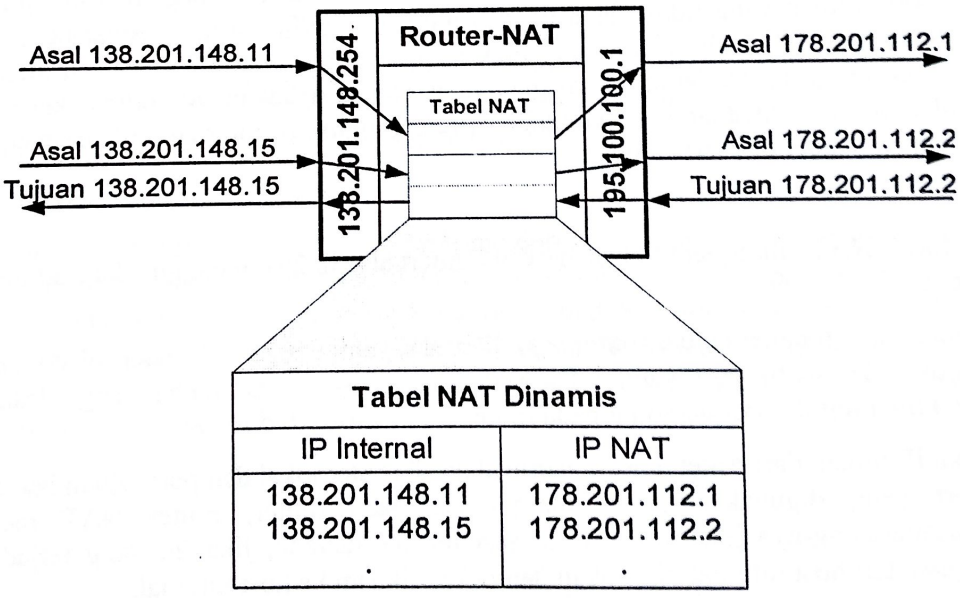
belakang router NAT yang melakukan translasi address dinamis - dengan melihat catatan koneksi host-host di bawahnya, karena bisa jadi pada koneksi berikutnya host yang hendak diakses telah menggunakan IP address yang berbeda. Dalam kasus khusus ini memiliki IP NAT lebih banyak daripada IP yang akan ditranslasi ($m<n$) dapat menjadi pertimbangan penting.

Koneksi dari luar hanya mungkin saat host yang hendak dicapai masih mempunyai IP NAT yang diberikan, misalnya masih mempunyai entri di dalam tabel NAT dimana router-NAT mencatat *track* pemetaan IP internal ke IP NAT. Sebagai contoh, sesi FTP non-passive, dimana server mencoba untuk membangun jalur data, tidak masalah karena saat server mengirim paket-paketnya ke client FTP telah tersedia suatu entri untuk client di dalam tabel NAT dan kemungkinan besar masih berisi pemetaan IP client ke IP NAT yang dibuat saat client memulai jalur control FTP, kecuali sesi FTP telah idle untuk waktu yang lama melebihi waktu *timeout* entri

Namun begitu, jika seorang outsider ingin membangun koneksi ke host tertentu yang ada di dalam pada waktu-waktu berbeda terdapat dua kemungkinan : host *inside* tidak mempunyai entri dalam tabel NAT dan karena itu koneksi tidak tercipta atau ada entri dalam tabel NAT tetapi IP NAT mana yang harus digunakan tidak diketahui, kecuali tentu saja IP tujuan koneksi diketahui oleh outside setelah host *inside* berkomunikasi dengan outside terlebih dahulu. Pada kasus terakhir ini, hanya IP NAT yang diketahui dan bukan IP host, pengetahuan ini berlaku hanya saat komunikasi host internal menempati tabel NAT dan belum *timeout*.

Contoh :

- Aturan NAT : secara dinamis translasikan semua IP dalam network 138.201 (class B) ke IP dalam network 178.201.112 (class C)
- Setiap koneksi baru dari inside mendapatkan suatu IP dari pool address C, selama ada address yang belum digunakan
- Jika pemetaan telah ada untuk host internal maka tidak diperlukan pemetaan baru
- Selama ada pemetaan, host internal dapat dicapai melalui IP yang secara temporer telah diberikan ke dirinya



Tabel NAT Dinamis	
IP Internal	IP NAT
138.201.148.11	178.201.112.1
138.201.148.15	178.201.112.2
.	.

Gambar 2. Proses translasi pada NAT dinamis

2.3 Masquerading (NAPT)

Transla i - m:n, m>=1 dan n=1 (m,n dalam N

Kasus lebih khusus dari NAT dinamis adalah translasi-**m:1**, disebut juga masquerading - menjadi begitu terkenal karena dapat diimplementasikan dengan mudah pada sistem Linux. Ini merupakan jenis teknik NAT yang paling banyak digunakan saat ini. Pada masquerading, banyak IP

address lokal disembunyikan di belakang satu IP publik. Dibandingkan dengan NAT dinamis orisinil, meskipun hanya satu IP NAT, semua host dibelakangnya dapat terkoneksi dan tidak perlu khawatir terjadi kesalahan transmisi data. Sejumlah koneksi yang diperbolehkan di-multipleks menggunakan informasi port TCP. Jumlah koneksi simultan terbatas hanya oleh jumlah port TCP yang tersedia pada router NAT.

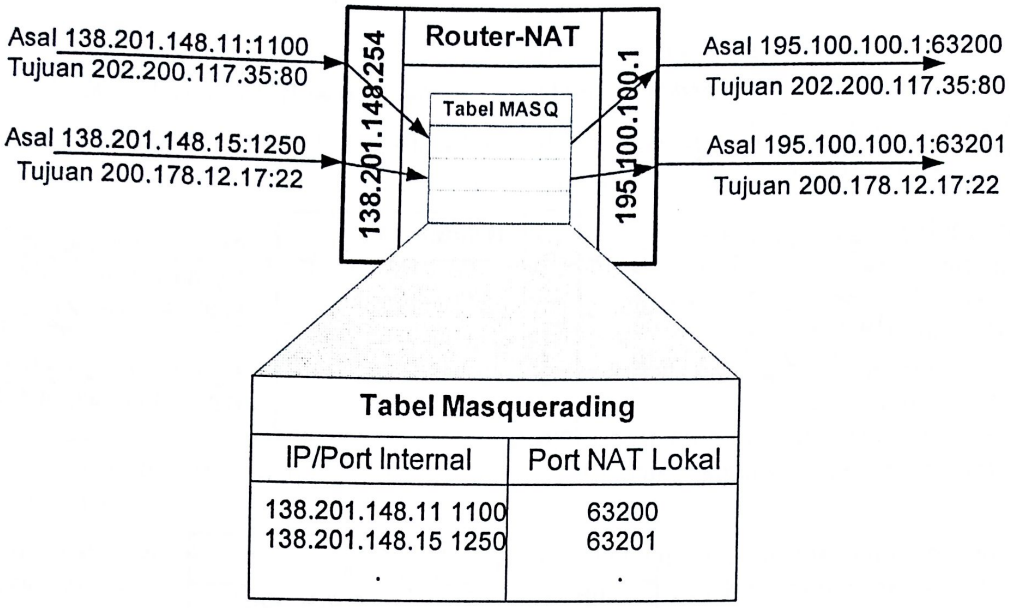
Masalah khusus dari masquerading adalah bahwa beberapa layanan (*service*) pada host tertentu hanya menerima koneksi yang datang dari port tertentu yang diijinkan (*privileged port*) dan memastikan bahwa koneksi tidak datang dari sembarang pengguna. Implementasi Linux menggunakan port-port non-privilege untuk masquerading untuk menghindari campuran dengan koneksi 'reguler' ke port-port ini. Masquerading biasanya menggunakan port-port dalam range atas, di Linux range ini dimulai dari port 61000 dan berakhir pada port 61000+4096, yang merupakan default dan dapat dengan mudah diubah melalui modifikasi konfigurasi network. Ini juga menunjukkan bahwa implementasi Linux secara default hanya mengijinkan 4095 koneksi pada satu waktu (*concurrent*). Untuk mengijinkan koneksi ter-masquerade pada port-port diluar range di atas, diperlukan perawatan dan manajemen informasi tentang status koneksi. Linux misalnya, memperlakukan semua paket dengan *IP tujuan = IP lokal* dan *port tujuan berada dalam range yang digunakan untuk masquerading*, sebagai paket yang harus di-demasquerade.

Koneksi masuk adalah tidak mungkin pada masquerade karena saat suatu host mempunyai entri dalam tabel masquerading pada peralatan NAT, entri ini hanya valid untuk koneksi aktif (sedang berjalan). Bahkan untuk ICMP-reply pun tidak dapat dilakukan secara langsung, semua harus di-filter dan di-relay oleh software router NAT. Tentu saja ada teknik untuk membuat koneksi dari luar dapat masuk ke host-host di belakarang masquerader tetapi tidak dengan teknik NAT. Misalnya men-setup peralatan NAT sehingga mesin ini me-relay semua koneksi yang masuk dari luar ke port telnet ke suatu host di dalam jaringan lokal.

Karena hanya ada satu IP yang visible di luar untuk pengaktifan koneksi masuk untuk layanan sama tetapi berbeda host pada *inside* maka sistem harus mendengar koneksi pada port-port berbeda pada mesin NAT, satu untuk setiap layanan dan IP internal. Karena sebagian besar aplikasi mendengar pada port well-known yang tidak dapat dengan mudah dikonfigurasi, ini benar-benar tidak tepat dan biasanya bukan pilihan, khususnya tidak untuk layanan publik. Solusinya adalah mempunyai banyak IP address eksternal untuk sejumlah layanan yang akan disediakan. IP addres eksternal masih dapat di-share oleh berbagai layanan dan kemudian di-remap untuk IP internal berbeda menggunakan NAT tetapi ini bukan bagian dari masquerade.

Contoh

- Aturan NAT: masquerade-kan network internal 138.201 menggunakan address router NAT sendiri
- Untuk setiap paket keluar (*outgoing*) IP asal diganti dengan IP router (eksternal) dan port asal ditukar ke suatu port yang tidak sedang digunakan dari range yang dicadangkan secara eksklusif untuk masquerading pada router
- Jika IP tujuan dari paket masuk merupakan IP router lokal dan port tujuan berada dalam range port yang digunakan untuk masquerade pada router, router NAT memeriksa tabel masquerading-nya jika paket milik sesi ter-masquerade; jika ini yang terjadi, IP dan port tujuan dari host internal disisipkan dan paket dikirim ke host internal.



Gambar 3. Proses translasi IP – port pada IP Masquerade

Keuntungan utama dari masquerading bagi banyak orang adalah bahwa hanya diperlukan satu IP address publik tetapi semua anggota jaringan internal dapat secara langsung mengakses Internet. Ini begitu penting karena IP address sudah begitu mahal. Selama terdapat gateway level aplikasi maka tidak diperlukan IP atau suatu jenis NAT dan cukup satu IP.

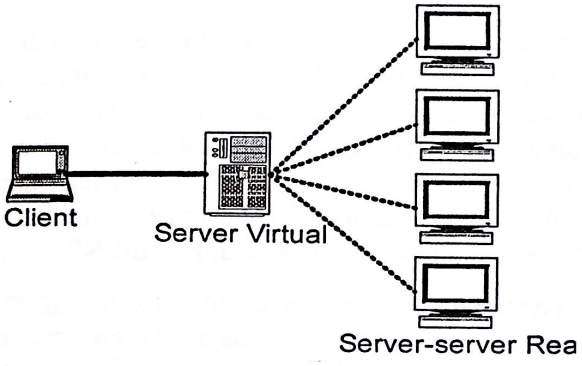
Masquerading pada beberapa pelaratana router sering disebut NATP atau PAT (*port address translation*).

3. TEKNIK – TEKNIK NAT LAINNYA

Penggunaan NAT terus bertambah dan bervariasi. NAT klasik melayani tujuan penghematan ruang IP address dan ini telah berlangsung lama - diukur berdasarkan usia Intenet. Sekarang berbagai bentuk implementasi teknololgi NAT ditemukan. Beberapa teknologi baru ini dapat diterapkan pada sistem Linux, Windows dan bahkan pada sistem embeded seperti Cisco Router.

3.1 Virtual Server

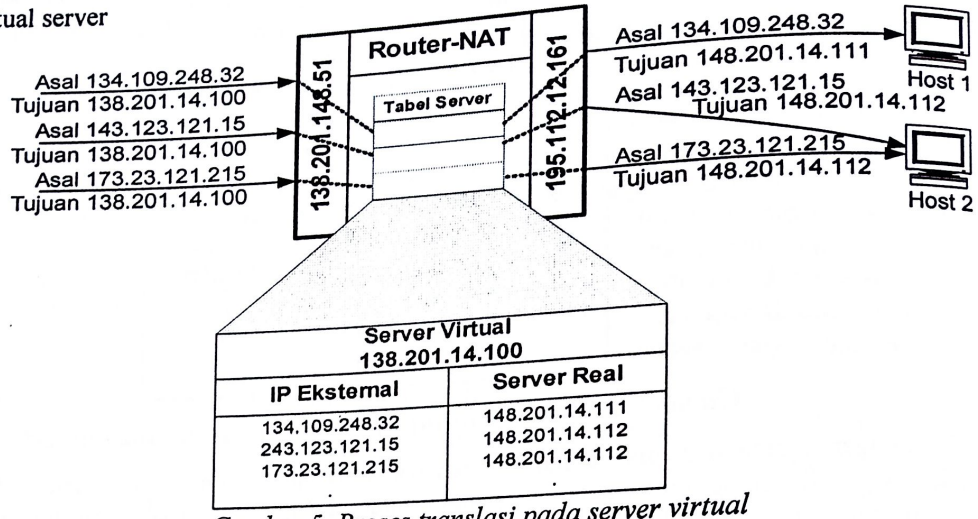
Secara singkat, terdapat sebuah IP yang merepresentasikan server virtual, tidak ada layanan server seperti web dan email yang disediakan di server ini, dan beberapa server lain (*real server*) menyediakan layanan server serta terhubung ke server virtual. Saat host terkoneksi ke IP virtual router, NAT mengganti address tujuan yang sebelumnya berisi IP virtaul ke IP salah satu server real. Tergantung pada algoritma yang digunakan untuk memilih IP real server, server virtual harus dapat melayani berbagai kebutuhan di bidang ini.



Gambar 4. Arsitektur sederhana Server Virtual

Contoh

- Aturan NAT : buat sebuah virtual server menggunakan IP 138.201.14.100
- Gunakan dua host dengan IP 138.201.14.111 dan 148.201.14.112, sebagai server real dari virtual server



Gambar 5. Proses translasi pada server virtual

- Sekarang koneksi dari outside ke server virtual dipetakan oleh router-NAT untuk menggunakan salah satu dari dua real server
- Server real mana yang akan dituju saat ada koneksi baru (belum ada di dalam tabel server virtual) tergantung pada algoritma yang digunakan

3.2 Load Balancing

Algoritma yang digunakan untuk menentukan IP real mana akan yang digunakan – misalnya dengan memeriksa beban (*load*) setiap server real (misalnya dengan menghitung paket/detik yang melewati router NAT ke server real) dan memilih IP server dengan beban paling ringan, dengan demikian secara relatif dicapai distribusi merata pada IP virtual terhadap server-server real. Jumlah algoritma yang dapat digunakan disini cukup banyak tetapi secara virtual semuanya dapat dikompromikan, karena prediksi beban tidak dapat didefinisikan dengan tepat. Kita dapat misalnya, menjalankan suatu daemon pada setiap server real dan menginformasikan router-NAT tentang beban pada mesin tersebut – beban mesin mungkin diperoleh walu tidak tepat 100% – dan me-remap koneksi baru ke sistem yang mempunyai nilai paling rendah. Ini memerlukan komunikasi antara server real dengan router-NAT, sehingga kita dapat merujuk untuk menggunakan data yang tersedia pada router, seperti jumlah koneksi yang saat ini sedang di-remap ke host, atau kita dapat menggunakan data yang tidak secara alami tersedia pada router tetapi dapat dengan mudah dikumpulkan, seperti rata-rata jumlah byte atau paket per-detik yang sedang ditangani oleh host.

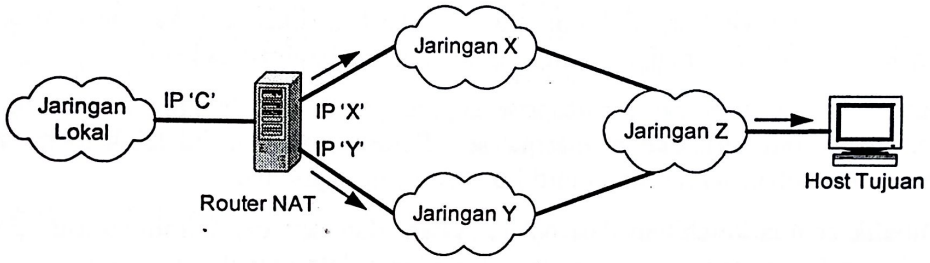
Terdapat sejumlah pendekatan terhadap load balancing, sebagian besar berada pada level atas (pengguna). Salah satunya disebutkan dalam RFC 1794, yaitu *DNS Support for Load Balancing*. Server DNS mengontrol beban mesin server real dengan memberikan IP dari mesin yang kurang sibuk saat terdapat queri. Karena queri DNS akan di-cache di server DNS berikutnya, kontrol menjadi sangat terbatas, tetapi ini akan bekerja cukup baik jika terdapat banyak queri dan ketika terdapat permintaan dari banyak client. Namun pendekatan ini tidak akan membantu saat salah satu server gagal atau tidak tersedia – informasi server real tujuan masih dipegang oleh DNS.

Contoh lain adalah program cache terkenal squid yang menggunakan algoritma kompleks untuk menemukan dari mana obyek diambil. Ini bukanlah solusi general tetapi terbatas untuk program tertentu. NAT - pada sisi lain, dapat melakukan distribusi beban uuntuk berbagai layanan yang lebih besar selama layanan tersebut berbasis IP.

Kita dapat menggunakan NAT untuk mendistribusikan beban ke beberapa host dan dicapai ketersediaan lebih tinggi dari layanan berbasis host. Kita dapat menggunakan NAT untuk mengerjakan hal sama untuk jaringan? Virtual host yang merepresentasikan bebeberapa host real, juga dapat membuat suatu koneksi jaringan virtual yang terdiri dari bebepa kabel real yang memberikan keuntungan dan kekurangan sama seperti teknik server virtual.

Bagaimana kita dapat melakukan ini dengan NAT? Bayangkan kita mempunyai dua penyedia jasa Internet (*Internet Service Provider*, ISP). Ini dilakukan untuk menjamin koneksi internet selalu berjalan – saat satu jalur dis-koneksi, jalur lain menggantikannya, atau kedua-duanya berjalan sehingga saling berbagi beban. Setiap host yang memerlukan akses Internet memerlukan IP publik unik sehingga harus dibelikan IP address untuk setiap host dari setiap ISP. Saat suatu host ingin menggunakan provider 1 maka host tersebut menggunakan IP address yang diberikan oleh ISP 1 sebagai IP lokal; saat ingin menggunakan provider 2 maka host itu menggunakan IP yang diberikan oleh ISP 2 sebagai IP lokal. Setiap host mempunyai 2 IP dan sekarang dapat terkoneksi Internet menggunakan 2 jalur.

Kita dapat melakukan ditsribusi beban secara manual dengan mengatur beberapa host menggunakan provider 1 dan lainnya menggunakan provider 2 dan kita mendapatkan ketersediaan koneksi Internet yang lebih tinggi. Host-host dapat terkoneksi ke tujuan sama tetapi menggunakan jalur berbeda. Kondisi ini, dapat memunculkan masalah misalnya host 1 dan host 2 mengakses www.saffanah.com tetapi tingkat kesuksesannya berbeda. Di saat kritis, siang hari misalnya, jalur koneksi manakah yang sebaiknya digunakan oleh suatu host?



Gambar 6. Load balancing pada koneksi Internet 2 ISP

Menggunakan NAT, komputer lokal hanya memerlukan satu IP karena host-host sudah tidak perlu memutuskan provider mana yang akan digunakan. Jika mempunyai provider favorit, kita dapat menggunakan IP provider ini untuk host-host tetapi jika host hanya memerlukan akses keluar dan tidak ikut menyediakan layanan Internet maka lebih baik menggunakan IP internal yang bersifat private. Sekarang, saat host internal ingin membangun suatu koneksi baru dengan suatu tujuan di Internet, host tersebut hanya mengirim paket ke router defaultnya – dalam hal ini adalah router-NAT, dan IP asal adalah IP lokal (internal) host. Router-NAT, karena server mengetahui dan mencatat semua koneksi, menentukan provider mana yang dipilih dan mengirimnya ke router provider terpilih. Karena alamat asal adalah IP address dari provider terpilih, jawaban juga akan datang dengan cara seperti itu. Host dari mana paket berasal tidak pernah dapat mengetahui provider mana yang dipilih oleh router-NAT, jadi proses ini berlangsung secara transparan.

Kita dapat menggunakan algoritma yang sama seperti di server virtual, sehingga kita dapat melakukan load balancing dan mendapatkan fitur *high availability*. Perbedaan utama dengan implementasi server virtual adalah bahwa kita harus turut campur dalam proses routing. Pada contoh ini kita sebenarnya menggunakan dua router default, misalnya.

4. MASALAH DENGAN PROTOKOL ROUTING

NAT tidak selalu berbentuk proses yang transparan. Apapun akan berjalan baik jika IP hanya berupa protokol yang membawa informasi alamat IP. Terdapat berbagai protokol yang mengirim IP sebagai bagian dari data yang ditransmisi, dan jika merupakan IP terjemahan yang dikirim ke penerima di belakang router-NAT maka host penerima mendapatkan masalah. Cara untuk menyelesaikan masalah ini adalah mencari data yang ditransmisi dengan protokol tertentu yang

diketahui untuk menyertakan informasi IP, dan tentu saja menambah pekerjaan dan semakin kompleks.

Protokol-protokol tersebut di antaranya adalah FTP, ICMP, DNS, dan BOOTP. Masalah dalam NAT juga dapat disebabkan oleh pemakaian berbagai jenis protokol routing berbeda seperti RIP, IGRP, EIGRP, OSPF, dan BGP. Protokol routing dibuat oleh beberapa vendor dan biasanya mempunyai spesifikasi berbeda, ada yang open – dapat diterapkan di berbagai device, namun ada juga yang khusus untuk device tertentu. Secara umum, ada 3 hal yang perlu dipertimbangkan menyangkut protokol routing, yaitu :

- Jangan menggunakan hanya routing statis seperti RIP versi 1, sehingga peluang terjalinnya kerjasama dengan sistem lain yang dinamis lebih terbuka.
- Gunakan gateway level aplikasi - ini tentu terkait dengan kemudahan penanganan operasinya
- Tulis ulang paket

5. KESIMPULAN

NAT sudah banyak digunakan sekarang. Hampir semua device router mengimplementasikan NAT maupun variannya. Dua sistem operasi yang tangguh di bidang ini adalah Cisco Internetwork Operating System (IOS) dan Linux. Berbagai bentuk implementasi telah disediakan oleh diterapkan namun penggunaan utama masih seputar masquerading terutama pada perusahaan-perusahaan kecil dan warung internet (WarNet) di Indonesia, ini terpaksa dilakukan karena ketersediaan IP address publik semakin menipis dan mahal – tentu masalah utama bagi usaha kecil.

Pada perusahaan besar – enterprise seperti perbankan, hosting server, ISP dan layanan e-commerce, NAT lebih dari sekedar masquerade. Perusahaan skala ini lebih menfokuskan NAT pada jaminan distribusi beban server dan multi koneksi ke jaringan luar.

Dibalik semua kelebihan dari NAT, system dan network administrator perlu bekerja keras untuk menjamin keamanan jaringan serta ketersediaan data saat diperlukan akses dari luar, terutama yang berkaitan dengan koneksi multi-routing protocol – Internet adalah gabungan ribuan router dengan berbagai kelebihan dan kekurangannya.

6. REFERENSI

1. Bill Parkhurst, Routing first-step, Cisco Press, 2004.
2. Cisco Systems, Introduction to Cisco Internetwork Technology - Student Guide, Cisco Press, 2003.
3. Husni, Implementasi Jaringan Komputer dengan Linux Redhat, Andi, 2004.
4. Linas Vepstas, Linux Network Address Translation, http://linas.org/linux, 2002.
5. Michael Hasenstein, NAT and Network, http://www.suse.de/~mha/linux-ip-nat/diplom/node1.html, 2003.