Pada peremuan ini saya melampirkan sebuah video yang menjelaskan tentang proses Network Adress Translation. Proses ini berada pada di antara layer 3 dan layer 2 OSI.

Tugas: Berikan 1 contoh isu penelitian (Research Problem) yang bisa diangkat dari permasalahan Network Address Translation.

Tuliskan jawaban anda pada ms word, kemudian upload pada assignment ini.

Berikan 1 contoh isu penelitian (Research Problem) yang bisa diangkat dari permasalahan Network Address Translation:

Jawab :

Keterbatasan alamat IP lokal dapat menjadi masalah besar pada sebuah perusahaan yang berkembang. Dikarenakan jumlah alamat IP yang terbatas tidak sebanding dengan jumlah pengguna yang semakin bertambah.

Nama : Dhea Noranita Putri
Nim : 182420112

SOAL :

Berikan 1 contoh isu penelitian (Research Problem) yang bisa diangkat dari permasalahan Network Address Translation.

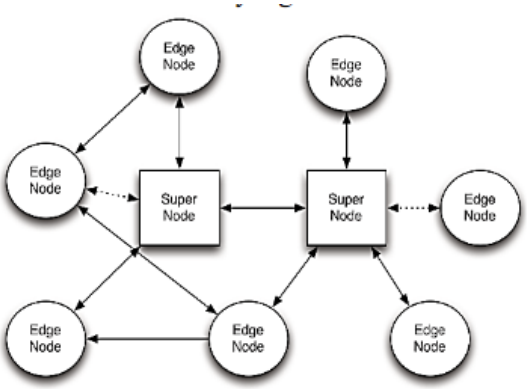
JAWAB :

Network Address Translation (NAT) adalah sebuah fungsi router yang memetakan alamat IP private (Lokal) ke alamat IP yang dikenal di Internet, sehingga jaringan private bisa mengakses internet. NAT merupakan salah satu metode yang memungkinkan host pada alamat private bisa berkomunkasi dengan jaringan di internet NAT jalan pada router yang menghubungkan antara private networks dan public Internet, dan menggantikan IP address dan Port pada sebuah paket dengan IP address dan Port yang lain pada sisi yang lain. Pada Jurnal yang saya baca yang berjudul "PEMANFAATAN P2P VPN UNTUK INTERKONEKSI KOMPUTER YANG BERADA DI BALIK NAT, STUDI KASUS SMS GATEWAY" ditulis oleh Haddad Sammi.

Penelitian pada jurnal ini ini membahas penggunaan P2P VPN yang lebih fleksibel dan dapat mengakomodir jumlah client dinamis yang berada di balik NAT dan alamat IP dinamis serta konfigurasi yang minim pada sisi server. Penelitian ini menjawab permasalahan bagaimana komputer / node yang berada di balik NAT dan memiliki IP dinamis dapat saling terbubung, serta bagaimana konfigurasi severdan clientpada P2P VPN agar komputer dapat mengakses SMS Gateway yang pada kondisi biasa tidak dapat diakses langsung karena berada pada jaringan komputer dan ISP yang terpisah dari yang digunakan oleh pengakses. Virtual Private Network (VPN) merupakan perpanjangan dari jaringan privat (private network) yang memanfaatkan jaringan internet sebagai penghubungnya (Knapp, 2000). VPN menggunakan enkripsi dan enkapsulasi untuk membentuk tunnel dari sebuah komputer ke komputer yang lain. Pengiriman data melalui tunnel, menjamin keamanan dan kerahasiaan data meskipun data dikirimkan melalui jaringan publik (internet)

N2N adalah jaringan privat layer 2 terenkripsi menggunakan protokol P2P. P2P merupakan metode yang memungkinkan pengguna menciptakan jaringan tertutup berbasis aplikasi di mana data dapat saling dipertukarkan tanpa dibatasi oleh firewall, alamat IP

dinamisdan NAT. N2N bekerja dengan cara memindahkan P2P dari level aplikasi ke level network(Deri & Andrews, 2008).



Gambar 1. Arsitektur N2N (Deri & Andrews, 2008)

Arsitektur N2N terdiri dari satu atau lebih supernodedan beberapa edge nodeyang terhubung kepadanya. Masing-masing edge nodememiliki daftar supernodetempat ia mendaftar pada saat startup, selanjutnya supernodesecara sementara akan menyimpan informasi dari setiap edge nodeyang mendaftar kepadanya dan secara periodik setiap edge nodeharus me-refresh informasi tersebut.

SMS gateway merupakan komunikasi menggunakan short message service (SMS) dengan memanfaatkan kode-kode yang telah disepakati untuk selanjutnya diproses sesuai dengan prosedur tertentu. Frontline SMS merupakan perangkat lunak yang berfungsi untuk mengubah komputer atau laptop yang terhubung modem SMS atau telepon seluler menjadi pusat perpesanan kelompok dua arah.

Hasilnya Jaringan private yang dirancang dalam penelitian ini terdiri dari dua komputer edge node yaitu komputer yang berada di balik NAT dan satu komputer supernodeyang memiliki ip publik yang dapat diakses oleh masing-masing edge. Setiap edge nodeberada pada jaringan di balik NAT. Penggunaa P2P VPN memberikan kemudahan pada sisi server di mana konfigurasi hampir tidak diperlukan lalu, Frontline SMS yang pada kondisi biasa tidak dapat diakses dari luar melalui internet, menjadi dapat diakses setelah berada di dalam VPN dengan menggunakan N2N dan P2P VPN memungkinkan komputer yang berada

di balik NAT untuk saling terhubung dan memungkinkan sebuah layanan yang di-hostingsecara lokal dapat diakses dari luar

Nama : Efrik Kartono Ahsa
NIM : 202420030
MK : Computer Network and Data Communication (Network Address Translation)

Network Address Translation Merupakan sebuah sistem untuk menggabungkan lebih dari satu komputer untuk dihubungkan ke dalam jaringan internet hanya dengan menggunakan sebuah alamat IP *(Internet Protocol)*. Sehingga setiap komputer di dalam NAT ketika berselancar di internet akan terlihat memiliki alamat IP yang sama jika dilacak. Dengan kata lain, sebuah alamat IP pada jaringan lokal akan terlebih dahulu ditranslasikan oleh NAT untuk dapat mengakses IP Publik dijaringan komputer. Sebelum proses translasi ini, maka pengguna tidak dapat terhubung ke internet.

Dari video tentang proses *Network Addres Translation* yang telah dijelaskan , dapat di jadikan sebuah *Research Problem "Network Address Translation Penghubung Ip Public Dan Ip Private"*

Dimana nantinya akan dibangun sebuah Topologi Local Network dengan IP Private untuk terhubung dengan IP Public. Saat menggunakan NAT, seorang klien dapat terhubung dengan internet melalui proses-proses berikut :

1. Pertama-tama, NAT menerima permintaan dari klien berupa paket data yang ditujukan untuk sebuah server remote di internet.

2. NAT kemudian mencatat alamat IP klien, lalu menyimpannya ke dalam tabel translasi alamat. Selanjutnya, alamat IP komputer klien tersebut diubah oleh NAT menjadi nomor IP NAT, lalu NAT lah yang akan melakukan permintaan kepada server.

3. Server kemudian merespon permintaan tersebut. Dari sudut pandang server, yang terlihat adalah alamat IP NAT, bukan alamat IP klien yang meminta data bersangkutan.

4. NAT menerima respon dari server, lalu melanjutkannya dengan mengirimkan ke alamat IP klien yang bersangkutan.

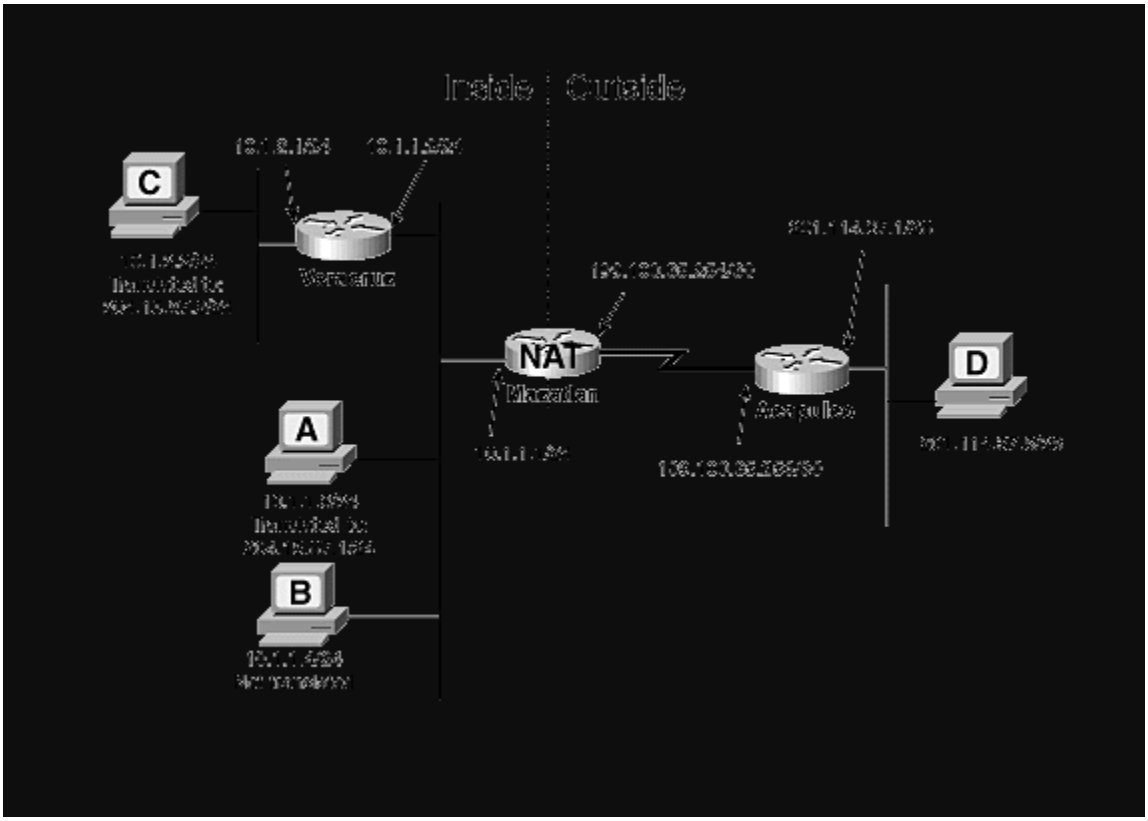
5. Keempat tahapan tersebut terjadi berulang-ulang, sehingga walaupun klien komputer tidak memiliki alamat IP publik, namun tetap dapat mengakses internet.

NAT sendiri dapat digunakan jika jumlah IP yang dimiliki sedikit sedangkan komputer yang akan disambungkan ke internet cukup banyak. Penggunaan mekanisme NAT dalam jaringan dapat menghemat biaya karena efisien dalam pemakaian IP Public dan penerapan NAT dalam jaringan dapat meningkatkan efisiensi manajemen LAN dalam internetworking.

Enggi Ardius

202420007

Studi Kasus: NAT Statis



Gambar 4.15

Pada Gambar 4-15 , jaringan bagian dalam ditangani dari ruang alamat 10.0.0.0. Dua dari perangkat tersebut, host A dan C, harus dapat berkomunikasi dengan dunia luar. Kedua perangkat tersebut diterjemahkan ke alamat publik 204.15.87.1/24 dan 204.15.87.2/24.

Contoh 4-4 menunjukkan konfigurasi untuk mengimplementasikan NAT di Mazatlan.

Gambar 4-15 Alamat Lokal Dalam Perangkat A dan C Secara Statis Diterjemahkan ke Dalam Alamat Global oleh Proses NAT di Router Mazatlan

Contoh 4-4 Menerapkan NAT Statis pada Router Mazatlan pada Gambar 4-15

```
antarmuka Ethernet0
 alamat ip 10.1.1.1 255.255.255.0
 ip nat di dalam
!
antarmuka Serial1
 tidak ada alamat ip
 frame-relay enkapsulasi
!
antarmuka Serial1.705 point-to-point
 alamat ip 199.100.35.254 255.255.255.252
 ip nat di luar
 antarmuka frame-relay-dlci 705
!
router ospf 100
 jaringan 10.1.1.1 0.0.0.0 area 0
 default-informasi berasal
!
ip nat di dalam sumber statis 10.1.2.2 204.15.87.2
ip nat di dalam sumber statis 10.1.1.3 204.15.87.1
!
rute ip 0.0.0.0 0.0.0.0 199.100.35.253
!
```

Antarmuka E0 router ditetapkan sebagai berada di dalam dengan perintah **ip nat inside** , dan subinterface Frame Relay S1.705 ditunjuk sebagai berada di luar dengan perintah **ip nat di luar** .

Selanjutnya, alamat lokal di dalam dipetakan ke dalam alamat global dengan **ip nat di dalam** perintah **statis sumber** . Ada dua dari perintah ini, satu untuk host C dan satu lagi untuk host A. Contoh 4-5 menunjukkan tabel NAT yang dihasilkan.

Contoh 4-5 Alamat IL dari Host C dan A Diterjemahkan Secara Statis ke Alamat IG

```
Mazatlan # tampilkan terjemahan ip nat
Pro Inside global Inside local Outside local Outside global
--- 204.15.87.2 10.1.2.2 --- ---
--- 204.15.87.1 10.1.1.3 --- ---
Mazatlan #
```

Ketika host A atau C mengirim paket ke luar, Mazatlan melihat alamat sumber dalam tabel NAT-nya dan membuat terjemahan yang sesuai. Router Acapulco memiliki rute (dalam hal ini, rute statis) ke jaringan 204.15.87.0 dan tidak memiliki pengetahuan tentang jaringan 10.0.0.0. Oleh karena itu, Acapulco dan host D dapat menanggapi paket dari host A dan C. Jika host B atau router Veracruz mengirimkan paket ke host D, paket tersebut akan diteruskan, tetapi tanpa terjemahan apa pun; ketika D merespons ke alamat IL yang tidak diterjemahkan, Acapulco tidak memiliki rute dan menjatuhkan paket, seperti yang ditunjukkan pada Contoh 4-6.

Contoh 4-6 Ketika Host D pada Gambar 4-15 Menanggapi Alamat IL yang Tidak Diterjemahkan dari Host B, Acapulco Tidak Memiliki Rute ke 10.0.0.0 dan Menurunkan Paket

```
Acapulco # debug ip icmp
Debug paket ICMP aktif
Acapulco #
1d00h: ICMP: dst (10.1.1.4) host tidak dapat dijangkau dikirim ke
201.114.37.5
1d00h: ICMP: dst (10.1.1.4) host tidak dapat dijangkau dikirim ke
201.114.37.5
1d00h: ICMP: dst (10.1.1.4) host tidak dapat dijangkau dikirim ke
201.114.37.5
1d00h: ICMP: dst (10.1.1.4) host tidak dapat dijangkau dikirim ke
201.114.37.5
1d00h: ICMP: dst (10.1.1.4) host tidak dapat dijangkau dikirim ke
201.114.37.5
```

Alamat global luar juga dapat diterjemahkan secara statis ke alamat lokal luar. Misalkan, sebagai contoh, administrator jaringan dalam pada Peraga 4-15 menginginkan host D "tampak" menjadi bagian dari jaringan dalam — katakanlah, dengan alamat 10.1.3.1. Contoh 4-7 menunjukkan konfigurasi NAT untuk Mazatlan.

Contoh 4-7 Mengonfigurasi Mazatlan untuk Secara Statis Menerjemahkan Alamat Global Luar ke Alamat Lokal Luar

```
ip nat di dalam sumber statis 10.1.1.3 204.15.87.1
ip nat di dalam sumber statis 10.1.2.2 204.15.87.2
ip nat di luar sumber statis 201.114.37.5 10.1.3.1
```

Konfigurasi NAT router tetap sama, kecuali untuk penambahan perintah **statik sumber luar ip nat** , yang dalam hal ini memetakan alamat OG 201.114.37.5 ke alamat OL 10.1.3.1. Contoh 4-8 menunjukkan tabel NAT yang dihasilkan.

Contoh 4-8 Pemetaan OG-ke-OL Ditambahkan ke Tabel NAT oleh Perintah Tambahan di Mazatlan

```
Mazatlan # tampilkan terjemahan ip nat
Pro Inside global Inside local Outside local Outside global
--- 204.15.87.2 10.1.2.2 --- ---
--- 204.15.87.1 10.1.1.3 --- ---
--- --- --- 10.1.3.1 201.114.37.5
Mazatlan #
```

Meskipun studi kasus ini hanya melibatkan pemetaan statis, beberapa pemetaan dinamis terjadi setelah lalu lintas melewati antara host A dan host D, dan antara host C dan host D, seperti yang diilustrasikan

Source : https://www.ciscopress.com/articles/article.asp?p=25273&seqNum=3

GATOT ARIFIANTO
NIM : 202420029
REGULER A MTI

COMPUTER NETWORK
AND COMMUNICATIONS

Pengertian dan Fungsi NAT (Network Address Translation) pada jaringan komputer

Apakah yang dimaksud dengan NAT (Network Address Translation) pada jaringan komputer?

NAT (Network Address Translation) adalah adalah sebuah proses pemetaan alamat IP dimana perangkat jaringan komputer akan memberikan alamat IP public ke perangkat jaringan local sehingga banyak IP private yang dapat mengakses IP public.

Dengan kata lain NAT akan mentranslasikan alamat IP sehingga IP address pada jaringan local dapat mengakses IP public pada jaringan WAN. NAT mentranslasikan alamat IP private untuk dapat mengakses alamat host diinternat dengan menggunakan alamat IP public pada jaringan tersebut. Tanpa hal tersebut(NAT) tidaka mungkin IP private pada jaringan local bisa mengakses internet.

Apa Fungsi dari NAT (Network Address Translation) pada jaringan komputer?

NAT (Network Address Translation) pada jaringan komputer berfungsi sebagai translasi alamat IP public ke alamat IP private atau sebaliknya sehingga dengan adanya NAT ini setiap komputer pada jaringan LAN dapat mengakses internet dengan mudah.

Kita tahu bahwa alamat IP Public didunia ini sudah semakin menipis sehingga penggunaan dati NAT ini dirasa sangatlah efisien dan efektif terutama dalam alokasi alamat IP.

Jenis - jenis dari NAT (Network Address Translation)

Pada jaringan komputer terdapat 2 jenis NAT, diantaranya:

- Dnat atau Destiantion Network Address Translation adalah sebuah NAT yang berfungsi untuk meneruskan paket dari IP public melalui firewall ke suatu host dalam jaringan. Dnat hanya bekerja pada tabel nat dan didalam tabel NAT berisi 3 bagian yang disebut dengan CHAIN, ketiga CHAIN tersebut meliputi prerouting, postrouting dan output.
- SNAT atau Source Network Address Translation yaitu sebuah NAT yang bertugas untuk merubah source address dari suatu paket data. SNAT hanya berlaku pada postrouting.

Kelebihan dan Kelemahan NAT (Network Address Translation)

Sebuah sistem tentunya akan memiliki kelebihan dan kelemahan, sehingga dengan memahami kelebihan dan kelemahan dan sistem tersebut kita bisa tahu kenapa kita harus menggunakan atau tidak menggunakannya, Berikut adalah kelebihan dan kelemahan menggunakan NAT pada jaringan:

Kelebihan dari NAT (Network Address Translation)

- Dengan adanya NAT dapat mengurangi adanya duplikasi IP address pada jaringan atau biasanya dikenal dengan conflict IP Address
- Dengan adanya NAT akan menghindari pengalamatan ulang pada saat jaringan tersebut berubah.
- Dapat menghemat IP Legal yang diberikan oleh ISP (Internet Service Provider)
- Dapat meningkatkan fleksibelitas untuk koneksi jaringan internet.

Kelemahan dari NAT (Network Address Translation)

- NAT dapat menyebabkan keterlambatan proses, ini disebabkan karena data yang dikirim harus melalui perangkat NAT terlebih dahulu.
- NAT dapat menyebabkan beberapa aplikasi yang tidak bisa berjalan dengan normal
- Dengan adanya NAT dapat menghilangkan kemampuan untuk melacak data karena data tersebut akan melewati firewall.

Cara Kerja NAT (Network Address Translation) pada Jaringan Komputer

NAT mempunyai fungsi yaitu sebagai translasi sebuah IP address, sehingga dengan adanya NAT ini IP address private dapat dengan mudah mengakses alamat IP public. Berikut adalah cara kerja dari NAT:

- Didalam IP address terdapat sebuah bagian yang mana di dalam IP tersebut terdapat informasi-informasi berupa alamat asal, alamat tujuan, TTL, dll. Bagian ini disebut dengan **header**.
- Sebagai contoh adalah sebuah komputer client dengan IP 192.168.1.2 akan mengakses atau melakukan request ke alamat www.google.co.id dengan IP 216.239.61.104, maka proses yang akan terjadi adalah sebagai berikut :
- Pada header, informasi yang tersimpan antara lain alamat asal > 192.168.1.2
- Sehingga ketika paket telah sampai pada router (gateway dari client), maka isi dari header akan dirubah menjadi : alamat asal > 192.168.1.1
- Sebelum paket keluar (menuju internet), maka header tersebut akan kembali berubah menjadi, alamat asal > 200.100.50.2, demikian seterusnya.
- Proses di atas merupakan mekanisme dari SNAT (source NAT), dimana IP asal (komputer client) akan dirubah disesuaikan dengan IP ketika paket telah berpindah. Ketika server google melakukan response / balasan, maka akan terjadi DNAT (destination NAT), dimana IP tujuan akan berubah disesuaikan dengan tujuan paket (komputer client). Prosesnya adalah sebagai berikut :
- Pada header, ketika paket telah sampai pada Router, informasi IP tujuan >200.100.50.20

- Ketika paket berada pada gateway, IP tujuan >192.168.1.1
- Di sini header akan kembali mengalami perubahan, IP tujuan > 192.168.1.2
- Sehingga Paket dapat dikirim dan bisa sampai pada komputer client.

SELESAI

Nama: Hairun Anisyah

Kelas : MTI 23 Reguler A

NIM : 202420039

Tugas : Berikan 1 contoh isu penelitian (Research Problem) yang bisa diangkat dari permasalahan Network Address Translation. Tuliskan jawaban anda pada ms word, kemudian upload pada assignment ini.

Contoh : sebagai contoh saja, pada saat ini kita sedang bekerja di sebuah perusahaan yang memiliki 600 orang karyawan tetapi Internet Service Provider (ISP) kita hanya memberikan anda 60 ip publik. Sehingga itu berarti kita hanya dapat mengizinkan 60 host untuk mengakses ke internet pada saat yang bersamaan, masalah akan muncul bagaimana jika pada saat yang bersamaan lebih dari 60 karyawan harus mengakses internet secara bersamaan? disinilah Network Address Translation(NAT) dibutuhkan .

Satuhal yang harus kita ingat adalah tidak semua komputer karyawan menggunakan internet pada saat yang bersamaan. Misalnya 60 dari mereka menggunakan internet untuk membaca berita saat pagi, 60 lainnya menggunakan internet pada siang hari untuk melihat email. Dengan menggunakan NAT kita dapat secara dinamis memberikan 60 ip publik kepada siapa saja yang sangat membutuhkannya pada saat itu. Hal ini biasa dikenal dengan sebuatan dynamic NAT.

Namun dengan melakukan setingan dynamic NAT diatas tidaklah memecahkan masalah yang sedang kita hadapi secara keseluruhan, karena suatu hari bisa saja lebih dari 60 orang yang mengakses internet di pagi hari. Pada kasus dynamic NAT maka hanya 60 orang yang dapat melakukan akses internet, sedangkan yang lainnya harus menunggu giliran untuk mengakses.

Permasalahan lainnya, faktanya ISP yang kita gunakan hanya memberikan tidak lebih dari 60 bahkan kurang, karena ip publik sangat susah belakangan ini. Maka untuk mengatasi dua permasalahaan diatas kita akan menggunakan fitur lain dari NAT yaitu : NAT Overload atau biasa disebut dengan Port Address Translation (PAT).

PAT mengizinkan beberapa perangkat pada Local Area Network (LAN) untuk menggunakan satu ip publik dengan port number yang berbeda. Dengan kata lain, hal inilah yang disebut dengan port address translation (PAT). Ketika kita menggunakan PAT, router memantains unique source port number di inside global ip address untuk membedakan antar translasi. Pada contoh dibawah ini, setiap host diarahkan pada satu ip publik yang sama dengan alamat ip publik 125.1.1.1 tetapi dengan port number yang berbeda (dari port 1000 - 1002).

Outside host ip address dapat juga di ganti dengan menggunakan Network Address Translation(NAT). Alamat Outside global merepresentasikan outside host dengan ip publik yang dapat digunakan untuk routing ke internet.

Istilah alamat outside local, merupakan alamat ip privat dari sebuah perangkat ekternal yang mengarak kepada local network. kita dapat mengartikan alamat outside local sebagai alamat inside local dari perangkat eksternal yang berhubungan dengan internet.

Mungkin kita masih bertanya-tanya berapa banyak port yang dapat digunakan untuk setiap IP? Ok, karena port number memiliki ukuran 16 bit, PAT dapat mendukung hingga 2^16 port, dimana kira-kira lebih dari 64,00 koneksi dapat menggunakan 1 alamat ip public. Nah sekarang kita telah mempelajari semua fitur yang terdapat pada NAT, berikut kesimpulan dari teknologi NAT yang telah dijabarkan diatas tadi;

Terdapat 2 tipe dari NAT translation: dynamic dan static.

Static NAT : Rancangan yang mengizinkan one-to-one mapping antara alamat ip local dan alamat ip publik, konfigurasi ini mensaratkan kita harus memiliki satu IP publik untuk setiap host pada network kita.

Dynamic NAT : Desain yang memetakan ip lokal ke ip publik secara dinamis dengan menggunakan pool ip publik. Namun desain ini masih memiliki keter batasan karena untuk dapat mengakses internet secara bersamaan pada jaringan kita tetap membutuhkan jumlah ip publik sebanyak host yang ada.

PAT (NAT Overloading) : Merupakan bagian dari dynamic NAT yang memetakan beberapa alamat ip private ke satu alamat ip publik (many-to-one) dengan menggunakan port yang berbeda.

Static NAT dan Dynamic NAT keduanya merupakan one-to-one mapping dari inside local ke alamat inside global. Tetapi dengan menggunakan PAT, kita dapat memiliki ratusan user yang terkoneksi ke internet dengan menggunakan satu alamat ip publik. PAT merupakan teknologi yang dapat membantu kita untuk menghemat penggunaan IP publik pada internet, PAT merupakan jenis yang paling populer dari NAT.

JURNAL J – CLICK
Jurnal Sistem Informasi Dan Manajemen Informatika
E-ISSN : 2541 – 2469
P-ISSN : 2355 – 7958

PEMANFAATAN P2P VPN UNTUK INTERKONEKSI KOMPUTER YANG BERADA DI BALIK NAT, STUDI KASUS SMS GATEWAY

Haddad Sammir
Sistem Informasi, STMIK Jayanusa, Jl. Olo Ladang No. 1 Padang
email: h.sammir@gmail.com

Abstract

Internet service provider (ISP) generally serve behind Network Address Translator (NAT) internet connection for it's customer which cause computers on the network can not communicate directly or by peer-to-peer (P2P). This condition impact the service in that computer, such as SMS Gateway need a public ip to be globaly accessed. P2P VPN can connect two or more behind NAT computer. Utilization of this technology open interconnection opportunities among behind NAT computer so that a service in a computer can be directly accessed.

Keywords*: Nat, VPN, P2P, SMS Gateway.*

Abstrak

Bentuk layanan yang diberikan oleh penyedia layanan internet (ISP) pada umumnya adalah koneksi komputer di balik Network Address Translator (NAT) yang menyebabkan komputer-komputer tidak dapat saling terhubung secara langsung / peer-to-peer (P2P). Kondisi tersebut berimbas kepada layanan yang dibuat pada sebuah komputer, seperti SMS gateway membutuhkan alamat IP publik agar dapat diakses secara global. P2P VPN dapat menghubungkan dua komputer di balik NAT. Pemanfaatan teknologi ini membuka peluang interkoneksi antara beberapa komputer yang berada dibalik NAT sehingga layanan pada sebuah komputer dapat diakses langsung.

Kata kunci: NAT, VPN, P2P, SMS Gateway

PENDAHULUAN

Latar Belakang Penelitian

Internet pada awanya merupakan jaringan komputer dimana setiap *node* memiliki alamat unik dan dapat berkomunikasi secara langsung dengan *node* lainnya, namun belakangan tergantikan dengan arsitektur baru yang terdiri dari sebuah alamat global dan banyak alamat privat yang saling terhubung melalui *Network Address Traslator* (NAT) (Ford, Srisuresh, & Kegel, 2005). Penggunaan NAT dapat sangat berguna jika jumlah alamat ip yang tersedia terbatas karena satu almat ip dapat mengakomodir banyak *node* (Touch, 2002). Arsitektur seperti ini cocok diterapkan pada komunikasi *client - server*, dimana *client* berada pada alamat privat dan *server* berada pada alamat publik sehingga digunakan

oleh banyak penyedia jasa internet (ISP) (Ford et al., 2005).

Penggunaan NAT pada ISP mengakibatkan komputer pada jaringan privat tidak dapat terhubung secara langsung. Beberapa layanan yang tidak membutuhkan akses kepada perangkat keras khusus dapat mengunakan *web/file/application hosting* agar dapat diakses global, namun aplikasi yang membutuhkan perangakat keras khusus, contohnya SMS *Gateway* tidak dapat dengan mudah di-*hosting*. Kondisi tersebut membutuhkan pemanfaataan teknologi *Vitual Private Network* (VPN) agar *node* yang berada dibalik NAT dapat saling terhubung.

Penelitian penerapaan VPN umumnya membahas tentang bagaimana membuat *Wide Area Network*. VPN digunakan untuk menghubungkan komputer jarak jauh (Fauzi, 2008). Penelitian pada aspek bisnis membahas bagaimana menghubungkan kantor-kantor cabang perusahaan dengan biaya murah dengan memanfaatkan VPN (Suryani & Honey, 2007). Penelitian tersebut menekankan pada pemanfaatkan perangkat Cisco. Penelitian lainnya adalah dengan menggunakan VPN untuk menghubungkan aplikasi web (Moodle dan Drupal) dengan database yang berada pada jaringan yang berbeda (Putra & Paramartha, 2012).

Penelitian-penelitian sebelumnya memfokuskan penggunaan VPN konvesional. Konfigurasi VPN dominan berada pada *server* dan jumlah *client* yang terhubung relatif bersifat statis. Penelitian yang penulis lakukan ini membahas penggunaan P2P VPN yang lebih fleksibel dan dapat mengakomodir jumlah client dinamis yang berada di balik NAT dan alamat IP dinamis serta konfigurasi yang minim pada sisi *server*. Penelitian ini menjawab permasalahan bagaimana komputer / *node* yang berada di balik NAT dan memiliki IP dinamis dapat saling terbubung, serta bagaimana konfigurasi *sever* dan *client* pada P2P VPN agar komputer dapat mengakses SMS *Gateway* yang pada kondisi biasa tidak dapat diakses langsung karena berada pada jaringan komputer dan ISP yang terpisah dari yang digunakan oleh pengakses.

Network Address Translator (NAT)

NAT merupakan metoda yang digunakan untuk memetakan alamat IP dari sebuah *realm* ke *realm* yang lain untuk menyediakan perutean yang transparan ke *host* (Srisuresh & Holdrege, 1999). NAT membuat sekelompok komputer pada sebuah jaringan terlihat seolah sebagai sebuah komputer dengan satu alamat IP. Kondisi ini bermanfaat apabila stok alamat IP yang dimiliki tebatas (Touch, 2002). Penggunaan NAT, meksipun pada satu sisi terlihat menguntungkan, namun hanya sesuai dengan konsep komunikasi client server, dimana *server* berada pada alamat global dan *client* berada pada alamat privat (Ford et al., 2005) sehingga menghambat komunikasi langsung antara komputer ke komputer (P2P) yang terhubung ke jaringan.

Virtual Private Network (VPN)

Virtual Private Network (VPN) merupakan perpanjangan dari jaringan privat (*private network*) yang memanfaatkan jaringan internet sebagai penghubungnya (Knapp, 2000). VPN

JURNAL J – CLICK
Jurnal Sistem Informasi Dan Manajemen Informatika
E-ISSN : 2541 – 2469
P-ISSN : 2355 – 7958

menggunakan enkripsi dan enkapsulasi untuk membentuk tunnel dari sebuah komputer ke komputer yang lain. Pengiriman data melalui tunnel, menjamin keamanan dan kerahasiaan data meskipun data dikirimkan melalui jaringan publik (internet).

VPN scara umum meyediakan layanan (Fauzi, 2008) :

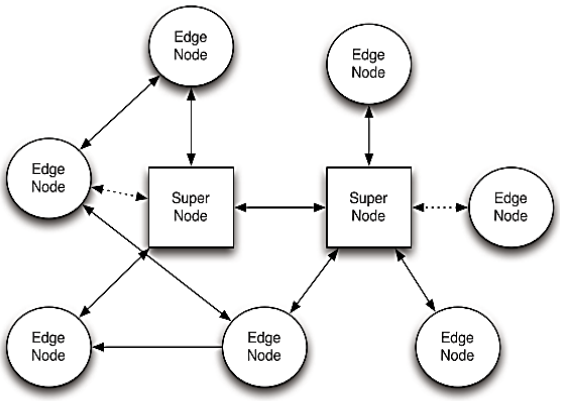
a) Keamanan.
b) Mengakomodasi perubahan pengguna yang dinamis.
c) Menyediakan kemampuan pertukaran informasi dalam beragam bentuk.
d) Mengakomodasi pengguna yang berbeda dengan berbagai macam browser, aplikasi dan sistem operasi.
e) Memungkinkan pengguna masuk ke dalam group.
f) Memelihara integritas sepanjang waktu, tanpa memperhatikan pergantian administrasi, perubahan teknologi, atau peningkatan kompleksitas sistem informasi perusahaan.

N2N Peer to Peer (P2P) VPN

N2N adalah jaringan privat layer 2 terenkripsi menggunakan protokol P2P. P2P merupakan metoda yang memungkinkan pengguna menciptakan jaringan tertutup berbasis aplikasi di mana data dapat saling dipertukarkan tanpa dibatasi oleh *firewall*, alamat IP dinamis dan NAT. N2N bekerja dengan cara memindahkan P2P dari level aplikasi ke level *network* (Deri & Andrews, 2008).

N2N dirancang dengan pertimbangan sebagai berikut (Deri & Andrews, 2008) :

a) N2N dirancang dengan menggunakan protokol P2P. Setiap node N2N memiliki nana dan *encryption key* bersama yang telah dibagikan dan diketahui bersama diantara pengguna yang tergabung.
b) Enkripsi dilakukan pada *edge node* dimana *private key* ditentukan sendiri oleh pengguna dan hanya pengguna bersangkutan yang dapat melakukan deenkripsi.
c) Masing-masing pengguna N2N dapat bergabung pada banyak komunitas (jaringan privat).
d) N2N memiliki satu atau lebih *supernode* yang berfungsi untuk memperkenalkan setiap *edge node* yang tergabung dan untuk menembus *firewall* simetris. *Supernode* tidak melakukan inspeksi terhadap paket yang dilewatkannya.
e) Komunitas (jaringan privat) N2N berdiri sendiri, namun dimungkinkan untuk mengarahkan paket menlintasi ke komunitas yang lain.



Gambar 1. Arsitektur N2N (Deri & Andrews, 2008)

Arsitektur N2N terdiri dari satu atau lebih *supernode* dan beberapa *edge node* yang terhubung kepadanya. Masing-masing *edge node* memiliki daftar *supernode* tempat ia mendaftar pada saat *startup*, selanjutnya *supernode* secara sementara

akan menyimpan informasi dari setiap *edge node* yang mendaftar kepadanya dan secara periodik setiap *edge node* harus me-refresh informasi tersebut.

SMS Gateway

SMS *gateway* merupakan komunikasi menggunakan short message service (SMS) dengan memanfaatkan kode-kode yang telah disepakati untuk selanjutnya diproses sesuai dengan prosedur tertentu. Prosedur tersebut dilakukan oleh aplikasi SMS *gateway*, sebuah perangkat lunak yang memanfaatkan teknologi seluler untuk mendistribusikan pesan-pesan yang dipadukan oleh sistem untuk didistribusikan kepada pengguna (Afrina & Ibrahim, 2015).

FrontlineSMS

FrontlineSMS merupakan perangkat lunak yang berfungsi untuk mengubah komputer atau laptop yang terhubung modem SMS atau telepon seluler menjadi pusat perpesanan kelompok dua arah (Lombardo, 2009). FrontlineSMS digunakan oleh petani, pekerja kesehatan atau reporter kemanusiaan untuk berkomunikasi dalam group SMS dan mendapatkan balasan kembali dalam bentuk SMS tanpa membutuhkan akses internet. FrontlineSMS telah mendapat reputasi dari pemanfaataannya dalam dunia nyata seperti pada saat terjadi bencana alam tsunami 2004 dan gempa Haiti (Bulkley, 2010).

METODE PENELITIAN

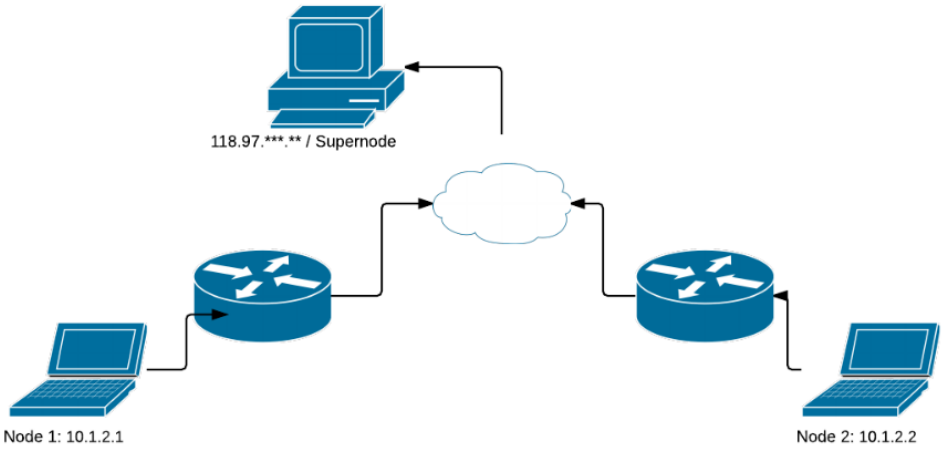
Metode yang penulis gunakan dalam penelitian ini adalah sebagai berikut:

1. **Kompilasi dan instalasi N2N layer 2 P2P VPN.**
   Kompilasi dan dilakukan pada komputer yang berperan sebagai supernode yang menggunakan sistem operasi Linux. Sedangkan pada komputer edge node hanya dilakukan istalasi N2N menggunakan distribusi *binary* pada sistem operasi windows beserta instalasi tap *driver*.
2. **Instalasi FrontlineSMS.**
   FrontlineSMS di-*install* pada salah satu *edge* dan terkoneksi ke sebuah ISP.
3. **Membuat konvensi komunitas.**
   N2N mengenkapsulasi jaringan privat dalam sebuah komunitas dengan nama, kelompok, alamat ip dan kunci enkripsi tertentu yang perlu disepakati sebelumnya.
4. **Aktivasi N2N.**
   N2N diaktifkan baik pada komputer supernode dan edge node.
5. **Pengujian jaringan**
   Mengakses FrontlineSMS yang berada pada salah satu edge node dari edge node yang lain pada jaringan komputer yang berada di balik NAT untuk menguji apakah FrontlineSMS dapat diakses.

HASIL

Jaringan prifat yang dirancang dalam penelitian ini terdiri dari dua komputer *edge node* yaitu komputer yang berada di balik NAT dan satu komputer *supernode* yang memiliki ip publik yang dapat diakses oleh masing-masing *edge*. Setiap *edge node* berada pada jaringan di balik NAT. Hasil perancangan VPN adalah seperti gambar berikut:

JURNAL J – CLICK
Jurnal Sistem Informasi Dan Manajemen Informatika
E-ISSN : 2541 – 2469
P-ISSN : 2355 – 7958



Gambar 2. Perancangan Virtual Private Network

Aktivasi Supernode

Supernode pada dasarnya hanya berfungsi sebagai *directory server* yang menyimpan informasi *node, community* dan *shared key*, sehingga tidak membutuhkan konfigurasi. *Supernode* dijalankan dengan perintah:

./supernode -l 1234 -v –f

Perintah di atas berarti menjalankan *supernode* pada port 1234 dan menampilkan *output* pada layar.

Aktivasi Edge Node

Aktivasi *edge node* membutuhkan beberapa konvensi. Setiap *edge node* perlu mendefinisikan community dan alamat ip, serta key (password) yang akan digunakan. Setiap *edge* yang memiliki *community* dan *key* yang sama maka berada pada jaringan yang sama dan dapat saling terhubung. Konvensi pada penelitian ini menggunakan "n2nhsammir" sebagai *community*, "password" sebagai *key*, alamat ip 10.1.2.1 untuk edge node 1 dan alamat 10.1.2.2 untuk edge node 2.

Aktivasi edge node 1:

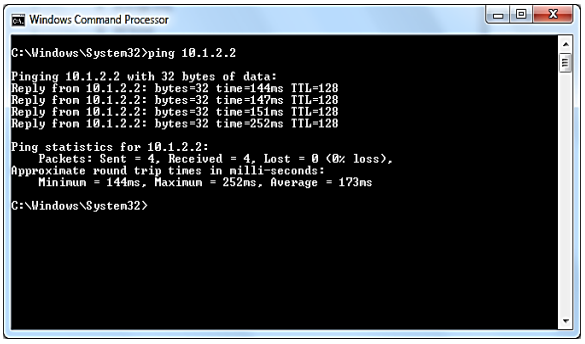
edge.exe -a 10.1.2.1 -c n2nhsammir -k password -l 118.97.***.**:1234 –f

Aktivasi edge node 2:

edge.exe -a 10.1.2.2 -c n2nhsammir -k password -l 118.97.***.**:1234 –f

Pengujian Ping

Pengujian ping dilakukan untuk melihat apakah koneksi telah terbangun pada dua buah *edge node*.

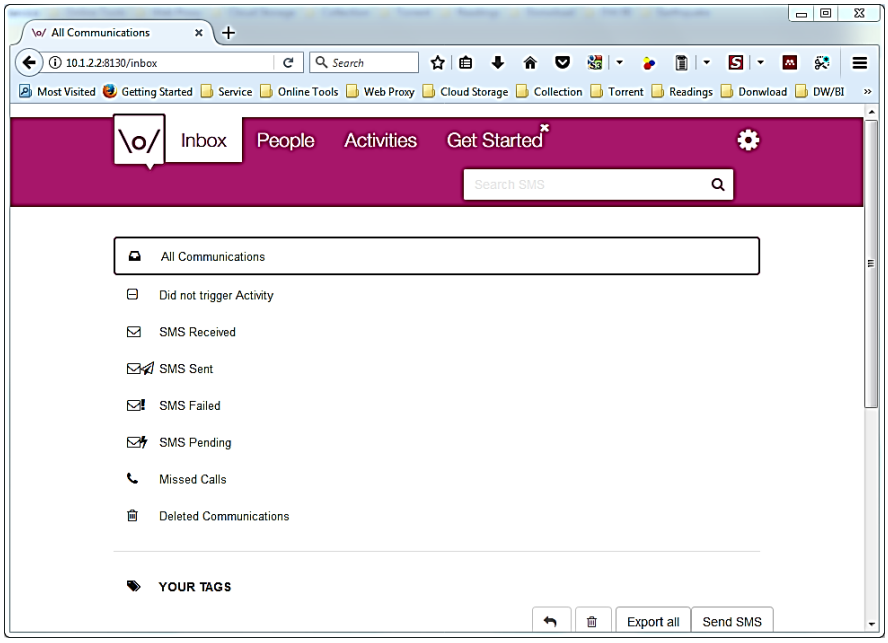


Gambar 3. Pengujian Ping

Pengujian Akses

Pengujian ini dilakukan untuk menguji fungsionalitas FrontlineSMS:

JURNAL J – CLICK
Jurnal Sistem Informasi Dan Manajemen Informatika
E-ISSN : 2541 – 2469
P-ISSN : 2355 – 7958



Gambar 4. Pengujian Akses

SIMPULAN

Penelitian ini menghasilkan beberapa simpulan yaitu:

1. Penggunaa P2P VPN memberikan kemudahan pada sisi *server* di mana konfigurasi hampir tidak diperlukan.
2. Setiap *edge node* yang hendak membentuk jaringan privat perlu mempersiapkan konvensi penamaan *community*, *key* dan pengalokasian alamat IP.
3. FrontlineSMS yang pada kondisi biasa tidak dapat diakses dari luar melalui internet, menjadi dapat diakses setelah berada di dalam VPN dengan menggunakan N2N.
4. P2P VPN memungkinkan komputer yang berada di balik NAT untuk saling terhubung dan memungkinkan sebuah layanan yang di-*hosting* secara lokal dapat diakses dari luar.

DAFTAR PUSTAKA

Afrina, M., & Ibrahim, A. (2015). Pengembangan Sistem Informasi SMS Gateway Dalam Meningkatkan Layanan Komunikasi Sekitar Akademika Fakultas Ilmu Komputer Unsri. *Jurnal Sistem Informasi (JSI)*, *7*(2), 852–864. Retrieved from http://ejournal.unsri.ac.id/index.php/jsi/index

Bulkley, K. (2010). Mobile technology takes centre stage in disaster relief | Activate | The Guardian. Retrieved September 9, 2017, from https://www.theguardian.com/activate/mobile-technology-disaster-relief

Deri, L., & Andrews, R. (2008). N2N: A layer two peer-to-peer VPN. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5127 LNCS*, 53–64. http://doi.org/10.1007/978-3-540-70587-1_5

Fauzi, A. (2008). Akses Jarak Jauh Layanan Intranet. *JURNAL TEKMAPRO*, *3*(2), 132–145.

Ford, B., Srisuresh, P., & Kegel, D. (2005). Peer-to-peer communication across network address translators. *Usenix.Org*, 1–31. http://doi.org/10.1109/CEC.2007.4424785

Knapp, L. (2000). Virtual Private Networking: An Overview. http://doi.org/10.1201/1079/43261.28.4.20001001/30357.3

Lombardo, J. (2009). Frontline SMS - PSFK. Retrieved September 11, 2017, from https://www.psfk.com/2009/10/frontline-sms.html

Putra, I. M. A. W., & Paramartha, C. R. A. (2012). PERANCANGAN IMPLEMENTASI KONSEP ROUTING DAN VIRTUAL PRIVATE NETWORK ANTARA WEBSERVER MOODLE DAN WEBSERVER DRUPAL. *JELIKU*, *1*(2), 99–103.

Srisuresh, P., & Holdrege, M. (1999). IP Network Address Translator (NAT) Terminology and Considerations. *IETF Informational*, *53*, 1689–1699. http://doi.org/10.1017/CBO9781107415324.004

Suryani, E., & Honey, S. N. R. (2007). Implementasi Virtual Private Network - Wan Dalam Dunia Bisnis. *Jurnal TI(JUTI)*, *6*(1), 31–38.

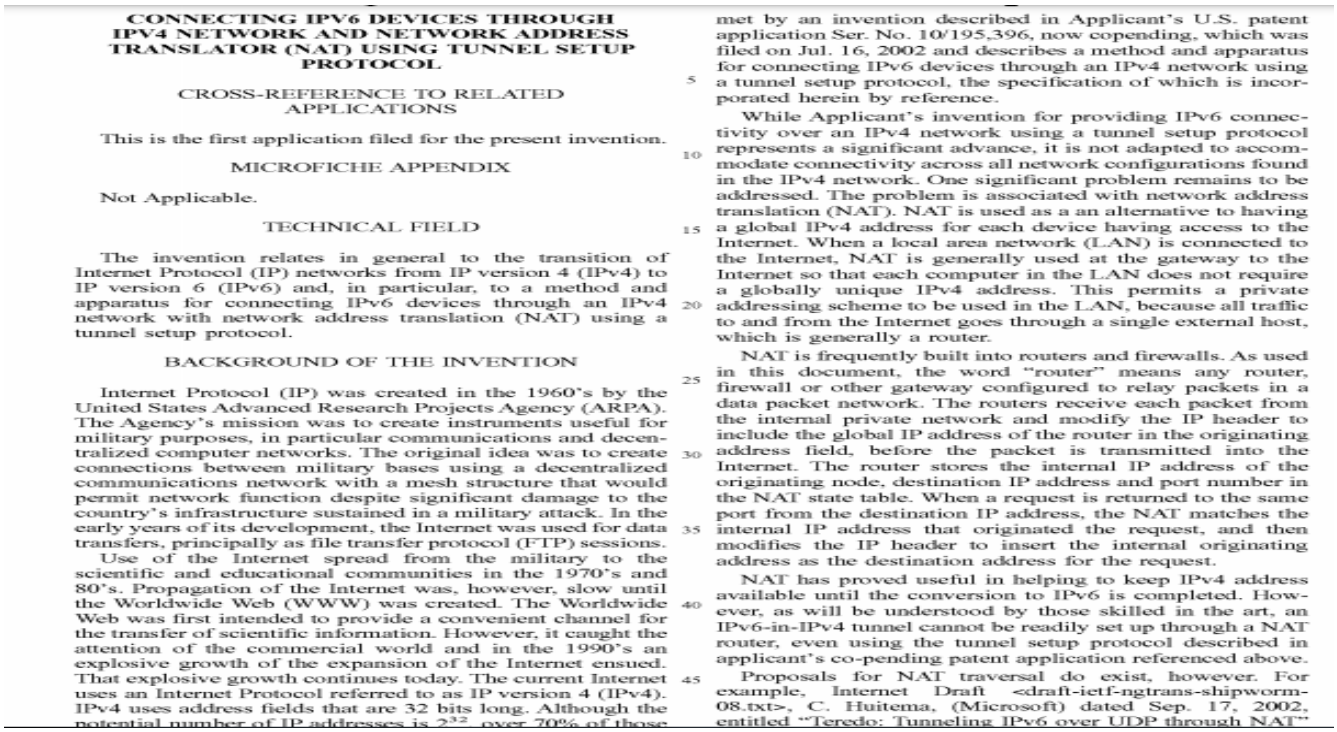
Touch, J. D. (2002). Those pesky NATs. *IEEE Internet Computing*, *6*(4), 96. http://doi.org/10.1109/MIC.2002.1020334

Judul Paper Jurnal: CONNECTING PV6 DEVICES THROUGH IPV4 NETWORK AND NETWORKADDRESS TRANSLATOR (NAT) USING TUNNEL SETUP PROTOCOL

Reviewer: Mirza Eka Putra

MTI23 REG B

ISI dan issue Paper: Berdasarkan garis besar penelitian ini yaitu, paper ini membahas mengenai isu bagaimana menghubungkan perangkat IPV6 dengan jaringan IPV4 dan menggunakan setup tunel protocol sebagai NAT dari kedua jaringan ini, adapun tujuan penemuan menyediakan a protokol pengaturan terowongan untuk mengotomatiskan pembentukan Tunnel IPv6-in-IPv4 melalui jaringan IPv4 dengan jaringan terjemahan alamat (NAT). agar tahu cara mengkoneksikan tuner server, router disediakan untuk membuat kontrol channel ke tunnel server, seperti dijelaskan di atas. Selanjutnya, pada langkah 202, klien dari tunel mengirimkan pesan penghubung ke server tunel untuk membangun saluran kontrol. Oleh karena itu, penemuan memungkinkan pembentukan otomatis tunnel IPv6-in-IPv4 melalui jaringan IPv4 denganNAT menggunakan saluran kontrol. Penggunaan saluran kontrolmemungkinkan negosiasi otomatis dari konfigurasi tertentu detail, seperti pilihan enkapsulasi, panjang prefiks IPv6,Delegasi DNS dan protokol peering router. Ini memfasilitasipenyebaran jaringan IPv6 dan memperbaiki tranposisi dari IPv4 ke IPv6. Penemuan ini sangat bergunadengan perangkat seluler, karena tunel IPv6-in-IPv4 baru bias dengan cepat dan otomatis dikonfigurasi



CONNECTING IPV6 DEVICES THROUGH IPV4 NETWORK AND NETWORK ADDRESS TRANSLATOR (NAT) USING TUNNEL SETUP PROTOCOL

CROSS-REFERENCE TO RELATED APPLICATIONS

This is the first application filed for the present invention.

MICROFICHE APPENDIX

Not Applicable.

TECHNICAL FIELD

The invention relates in general to the transition of Internet Protocol (IP) networks from IP version 4 (IPv4) to IP version 6 (IPv6) and, in particular, to a method and apparatus for connecting IPv6 devices through an IPv4 network with network address translation (NAT) using a tunnel setup protocol.

BACKGROUND OF THE INVENTION

Internet Protocol (IP) was created in the 1960's by the United States Advanced Research Projects Agency (ARPA). The Agency's mission was to create instruments useful for military purposes, in particular communications and decentralized computer networks. The original idea was to create connections between military bases using a decentralized communications network with a mesh structure that would permit network function despite significant damage to the country's infrastructure sustained in a military attack. In the early years of its development, the Internet was used for data transfers, principally as file transfer protocol (FTP) sessions.

Use of the Internet spread from the military to the scientific and educational communities in the 1970's and 80's. Propagation of the Internet was, however, slow until the Worldwide Web (WWW) was created. The Worldwide Web was first intended to provide a convenient channel for the transfer of scientific information. However, it caught the attention of the commercial world and in the 1990's an explosive growth of the expansion of the Internet ensued. That explosive growth continues today. The current Internet uses an Internet Protocol referred to as IP version 4 (IPv4). IPv4 uses address fields that are 32 bits long. Although the potential number of IP addresses is 2^{32}, over 70% of those

met by an invention described in Applicant's U.S. patent application Ser. No. 10/195,396, now copending, which was filed on Jul. 16, 2002 and describes a method and apparatus for connecting IPv6 devices through an IPv4 network using a tunnel setup protocol, the specification of which is incorporated herein by reference.

While Applicant's invention for providing IPv6 connectivity over an IPv4 network using a tunnel setup protocol represents a significant advance, it is not adapted to accommodate connectivity across all network configurations found in the IPv4 network. One significant problem remains to be addressed. The problem is associated with network address translation (NAT). NAT is used as a an alternative to having a global IPv4 address for each device having access to the Internet. When a local area network (LAN) is connected to the Internet, NAT is generally used at the gateway to the Internet so that each computer in the LAN does not require a globally unique IPv4 address. This permits a private addressing scheme to be used in the LAN, because all traffic to and from the Internet goes through a single external host, which is generally a router.

NAT is frequently built into routers and firewalls. As used in this document, the word "router" means any router, firewall or other gateway configured to relay packets in a data packet network. The routers receive each packet from the internal private network and modify the IP header to include the global IP address of the router in the originating address field, before the packet is transmitted into the Internet. The router stores the internal IP address of the originating node, destination IP address and port number in the NAT state table. When a request is returned to the same port from the destination IP address, the NAT matches the internal IP address that originated the request, and then modifies the IP header to insert the internal originating address as the destination address for the request.

NAT has proved useful in helping to keep IPv4 address available until the conversion to IPv6 is completed. However, as will be understood by those skilled in the art, an IPv6-in-IPv4 tunnel cannot be readily set up through a NAT router, even using the tunnel setup protocol described in applicant's co-pending patent application referenced above.

Proposals for NAT traversal do exist, however. For example, Internet Draft <draft-ietf-ngtrans-shipworm-08.txt>, C. Huitema, (Microsoft) dated Sep. 17, 2002, entitled "Teredo: Tunneling IPv6 over UDP through NAT"

US007305481B2

(12) **United States Patent**
Blanchet et al.

(10) **Patent No.: US 7,305,481 B2**
(45) **Date of Patent: Dec. 4, 2007**

(54) **CONNECTING IPV6 DEVICES THROUGH IPV4 NETWORK AND NETWORK ADDRESS TRANSLATOR (NAT) USING TUNNEL SETUP PROTOCOL**

(75) Inventors: **Marc Blanchet**, St-Augustin (CA); **Florent Parent**, Cap-Rouge (CA); **Jean-Francois Boudreault**, Ste-Foy (CA)

(73) Assignee: **Hexago Inc.**, Montreal (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 912 days.

(21) Appl. No.: **10/337,428**

(22) Filed: **Jan. 7, 2003**

(65) **Prior Publication Data**

US 2004/0133692 A1 Jul. 8, 2004

(51) **Int. Cl.**
G06F 15/16 (2006.01)
H04L 12/28 (2006.01)

(52) **U.S. Cl.** **709/230**; 709/227; 709/203; 370/389

(58) **Field of Classification Search** 709/248–253, 709/200–203, 217–229, 230, 231, 232, 238; 370/389; 711/202, 206; 719/310, 311, 313, 719/317, 318
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,580,717	B1 *	6/2003	Higuchi et al.	370/401
6,981,278	B1 *	12/2005	Minnig et al.	726/12
7,028,335	B1 *	4/2006	Borella et al.	726/11
7,032,242	B1 *	4/2006	Grabelsky et al.	726/11
7,036,143	B1 *	4/2006	Leung et al.	726/15
7,079,520	B2 *	7/2006	Feige et al.	370/338
2001/0040895	A1 *	11/2001	Templin	370/466
2003/0088702	A1	5/2003	Iwata et al.	709/245
2003/0225911	A1	12/2003	Lee et al.	709/245
2004/0013130	A1 *	1/2004	Blanchet et al.	370/466
2004/0052257	A1 *	3/2004	Abdo et al.	370/392
2004/0093434	A1 *	5/2004	Hovell et al.	709/249
2004/0100953	A1 *	5/2004	Chen et al.	370/389
2005/0223095	A1 *	10/2005	Volz et al.	709/225

FOREIGN PATENT DOCUMENTS

WO	WO 03/041365	5/2003
WO	WO 03/084184	10/2003
WO	WO 03/084185	10/2003

OTHER PUBLICATIONS

"IPv6 over IPv4 profile for Tunnel Setup Protocol (TSP)", M. Blanchet, et. al., pp. 1-13, Jul. 13, 2001.*
"An overview of the introduction of IPv6 in the Internet", W. Biemolt, et al., pp. 1-28, Feb. 2002.*
"Tunnel Setup Protocol" www.chone.net/ngtrans/ietf-51-london/tsp.ppt, Aug. 10, 2001, Marc Blanchet et al.*
"Tunnel Setup Protocol" www.ietf.org/procedings/99nov/ngtrans-blancket-tunnel-setup/tsldool.htm, Marc Blanchet, Nov. 1999.*

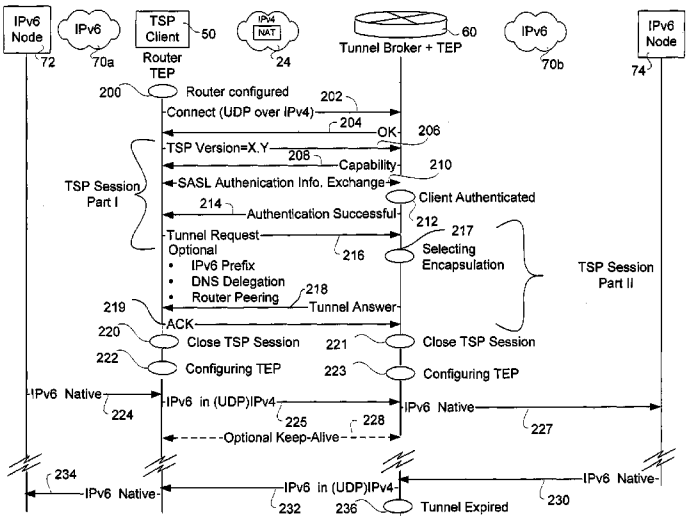
* cited by examiner

Primary Examiner—Haresh Patel
(74) *Attorney, Agent, or Firm*—Kent Daniels; Ogilvy Renault LLP

(57) **ABSTRACT**

A tunnel setup protocol enables tunnel clients to set up IPv6-in-IPv4 networks to permit IPv6 nodes to communicate across the IPv4 network using IPv6 native packets, even if the IPv4 network contains a Network Address Translation function. The tunnel setup protocol uses a control channel to negotiate tunnel configuration parameters and exchange tunnel configuration data between a tunnel client and a tunnel broker server. The tunnel setup is automatic, and migration to IPv6 is ameliorated.

28 Claims, 8 Drawing Sheets



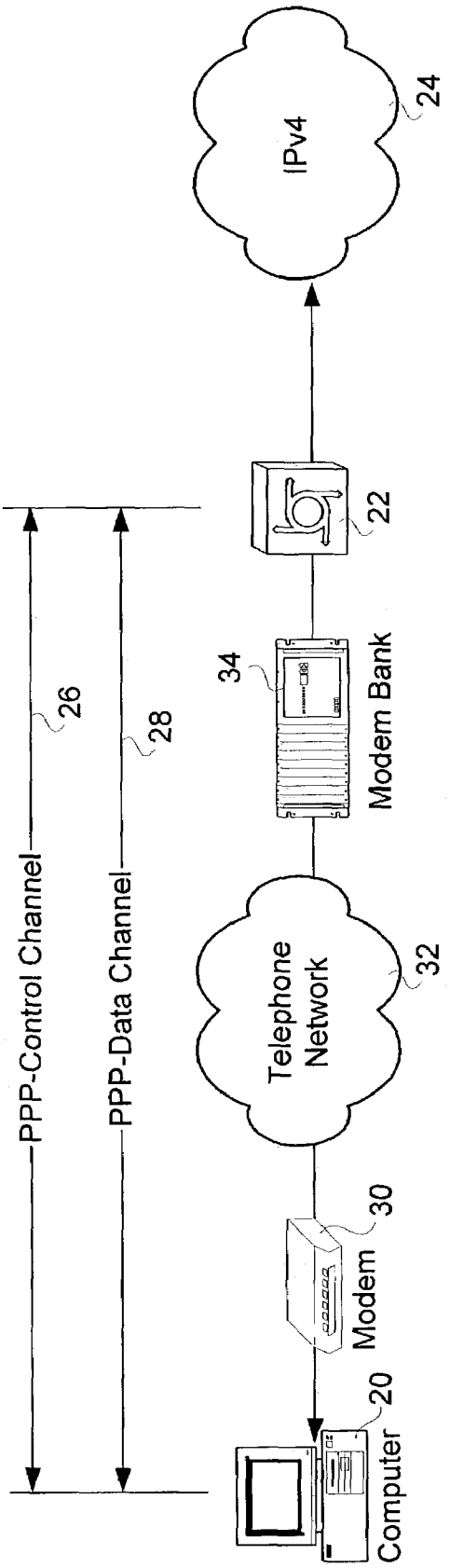


FIG. 1
Prior Art

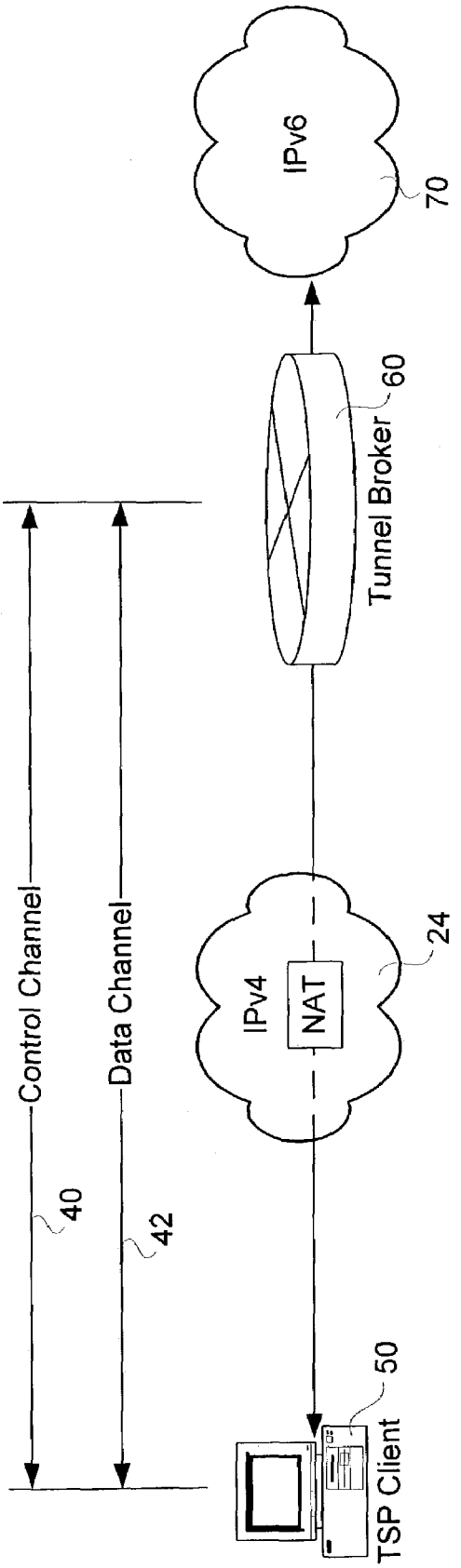


FIG. 2

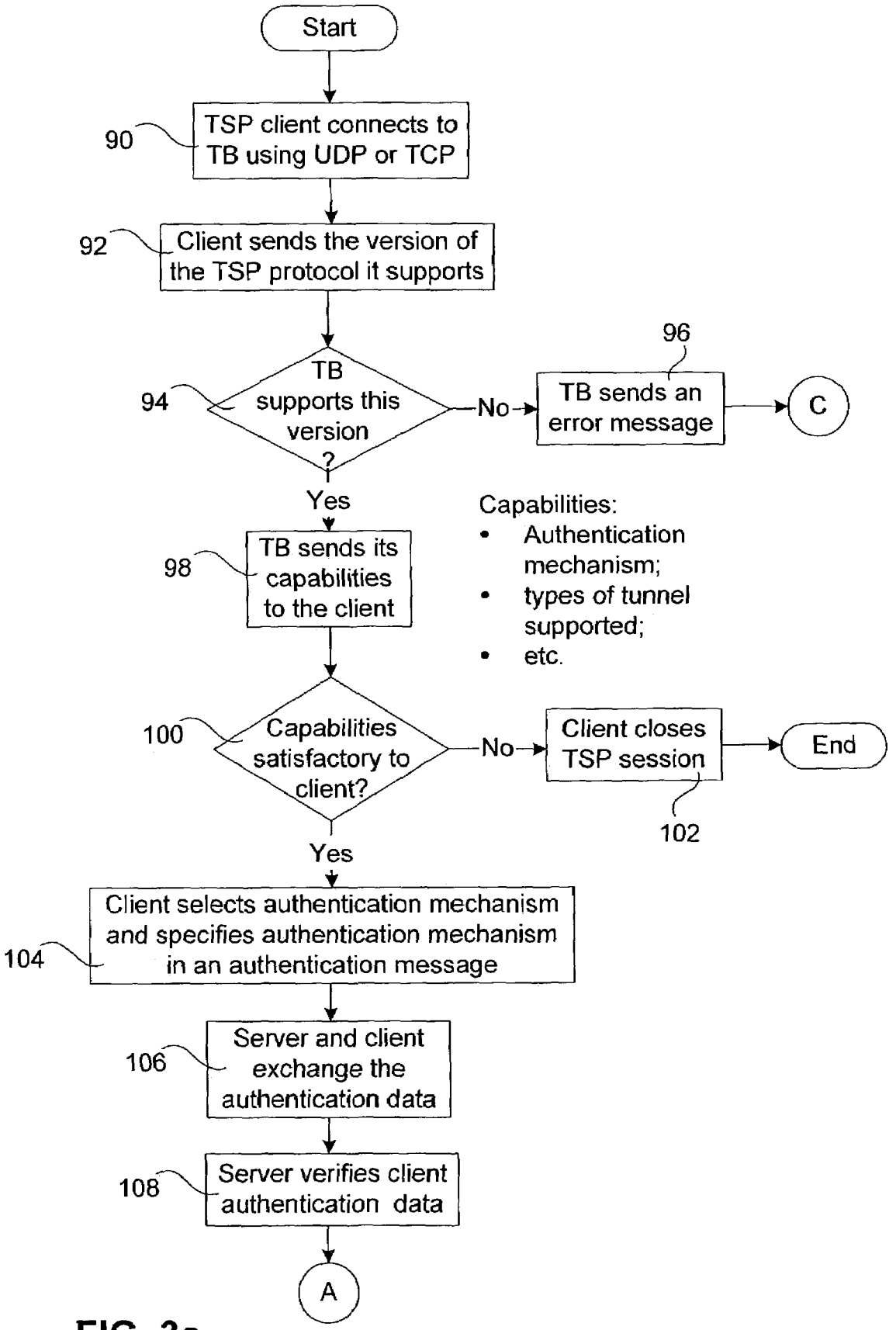

Start
90
TSP client connects to TB using UDP or TCP
92
Client sends the version of the TSP protocol it supports
94
TB supports this version ?
No
96
TB sends an error message
C
Yes
Capabilities:
• Authentication mechanism;
• types of tunnel supported;
• etc.
98
TB sends its capabilities to the client
100
Capabilities satisfactory to client?
No
Client closes TSP session
End
102
Yes
104
Client selects authentication mechanism and specifies authentication mechanism in an authentication message
106
Server and client exchange the authentication data
108
Server verifies client authentication data
A

FIG. 3a

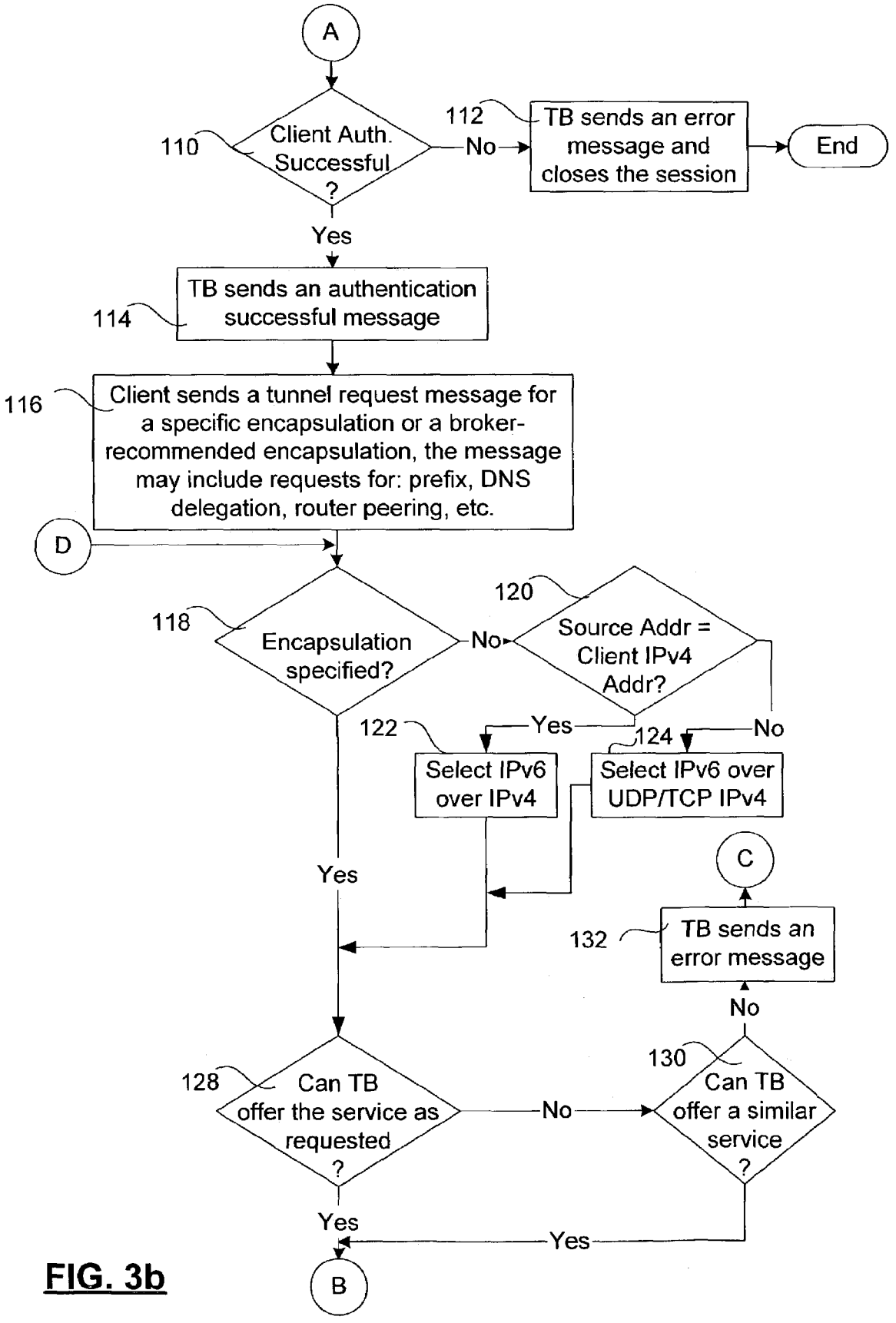

A
110
Client Auth. Successful ?
No
112
TB sends an error message and closes the session
End
Yes
114
TB sends an authentication successful message
116
Client sends a tunnel request message for a specific encapsulation or a broker-recommended encapsulation, the message may include requests for: prefix, DNS delegation, router peering, etc.
D
118
Encapsulation specified?
No
120
Source Addr = Client IPv4 Addr?
Yes
122
Select IPv6 over IPv4
No
124
Select IPv6 over UDP/TCP IPv4
Yes
C
132
TB sends an error message
No
128
Can TB offer the service as requested ?
No
130
Can TB offer a similar service ?
Yes
Yes
B

FIG. 3b

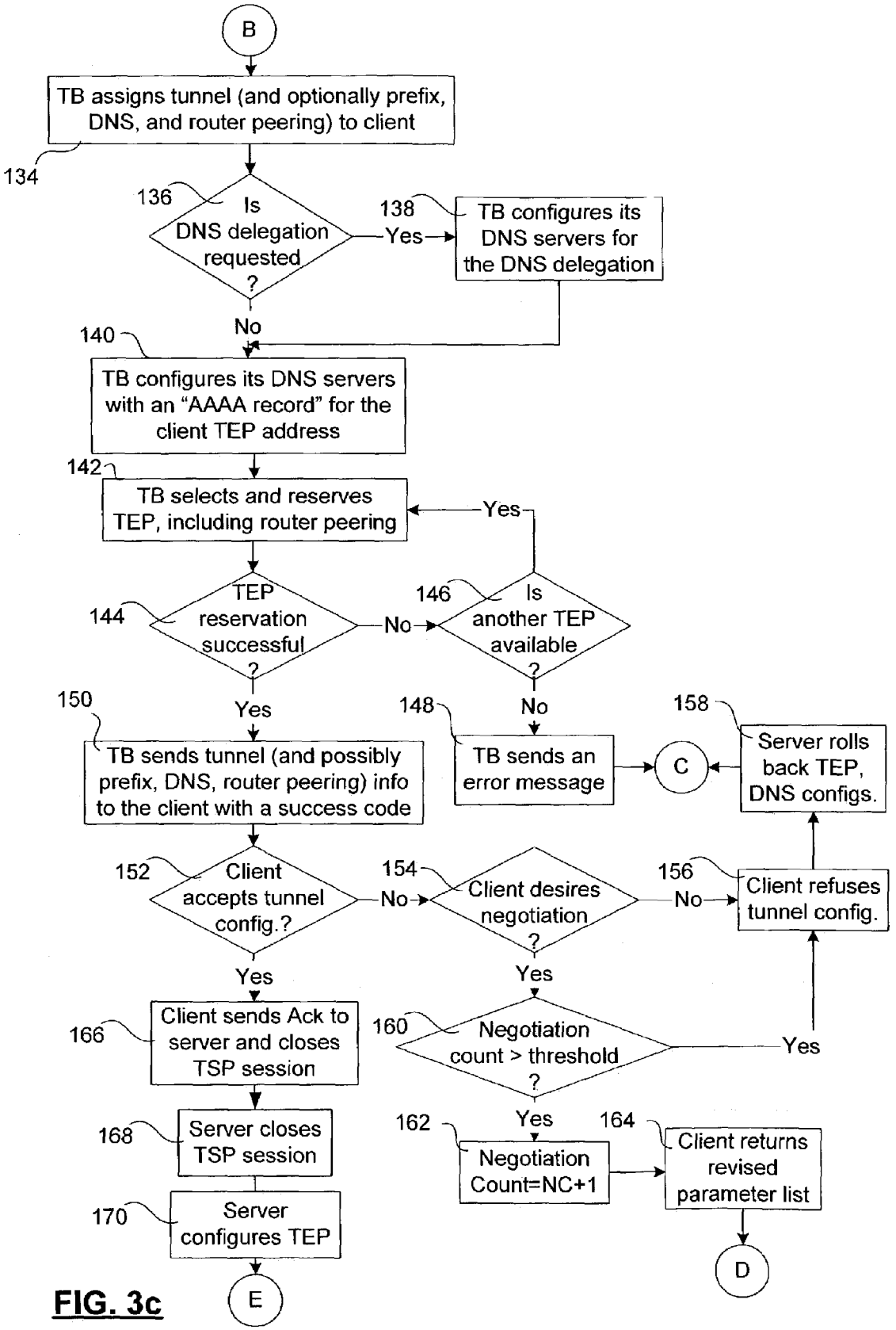

B
TB assigns tunnel (and optionally prefix, DNS, and router peering) to client
134
136
Is DNS delegation requested ?
Yes
138
TB configures its DNS servers for the DNS delegation
No
140
TB configures its DNS servers with an "AAAA record" for the client TEP address
142
TB selects and reserves TEP, including router peering
Yes
144
TEP reservation successful ?
No
146
Is another TEP available ?
Yes
No
150
TB sends tunnel (and possibly prefix, DNS, router peering) info to the client with a success code
148
TB sends an error message
C
158
Server rolls back TEP, DNS configs.
152
Client accepts tunnel config.?
No
154
Client desires negotiation ?
No
156
Client refuses tunnel config.
Yes
Yes
166
Client sends Ack to server and closes TSP session
160
Negotiation count > threshold ?
Yes
168
Server closes TSP session
162
Yes
Negotiation Count=NC+1
164
Client returns revised parameter list
170
Server configures TEP
D
E

FIG. 3c

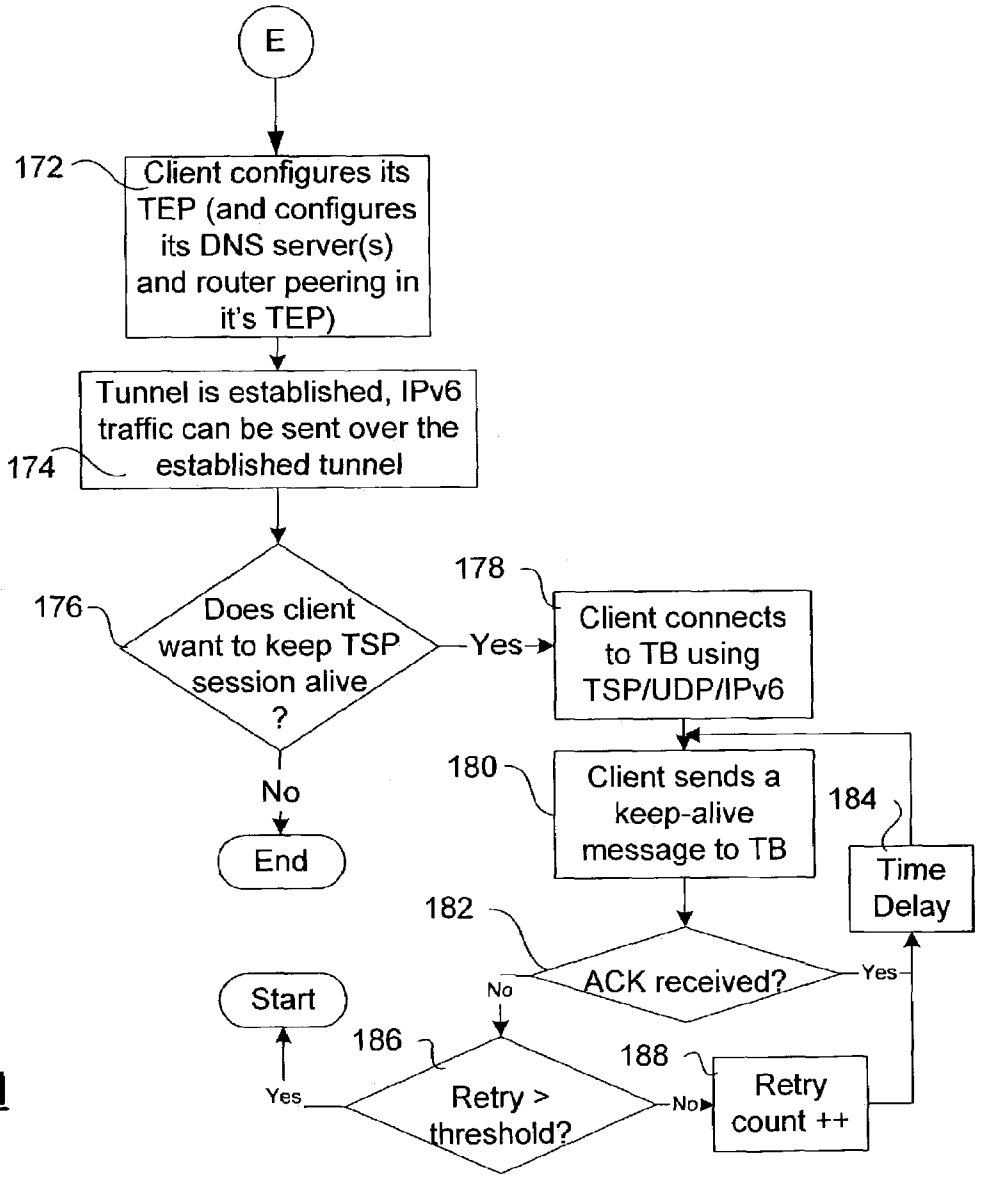

E
172
Client configures its TEP (and configures its DNS server(s) and router peering in it's TEP)
Tunnel is established, IPv6 traffic can be sent over the established tunnel
174
176
Does client want to keep TSP session alive ?
Yes
No
End
178
Client connects to TB using TSP/UDP/IPv6
180
Client sends a keep-alive message to TB
184
Time Delay
182
ACK received?
Yes
No
Start
186
Retry > threshold?
Yes
No
188
Retry count ++

FIG. 3d

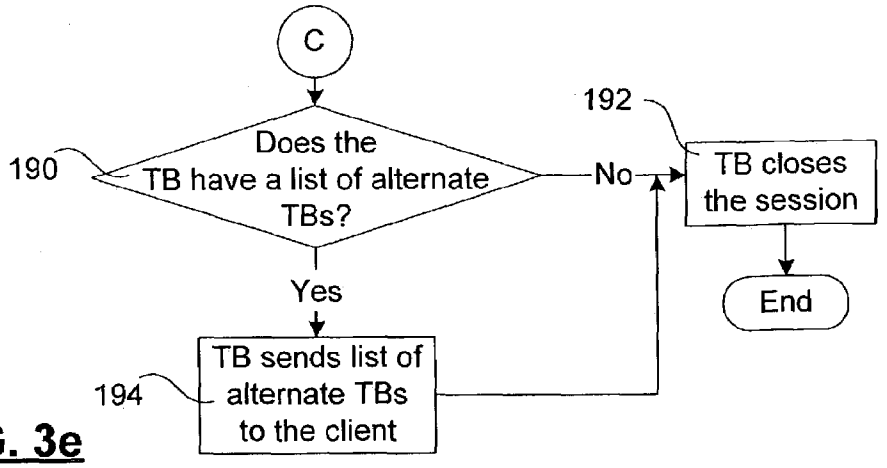

C
190
Does the TB have a list of alternate TBs?
No
Yes
192
TB closes the session
End
194
TB sends list of alternate TBs to the client

FIG. 3e

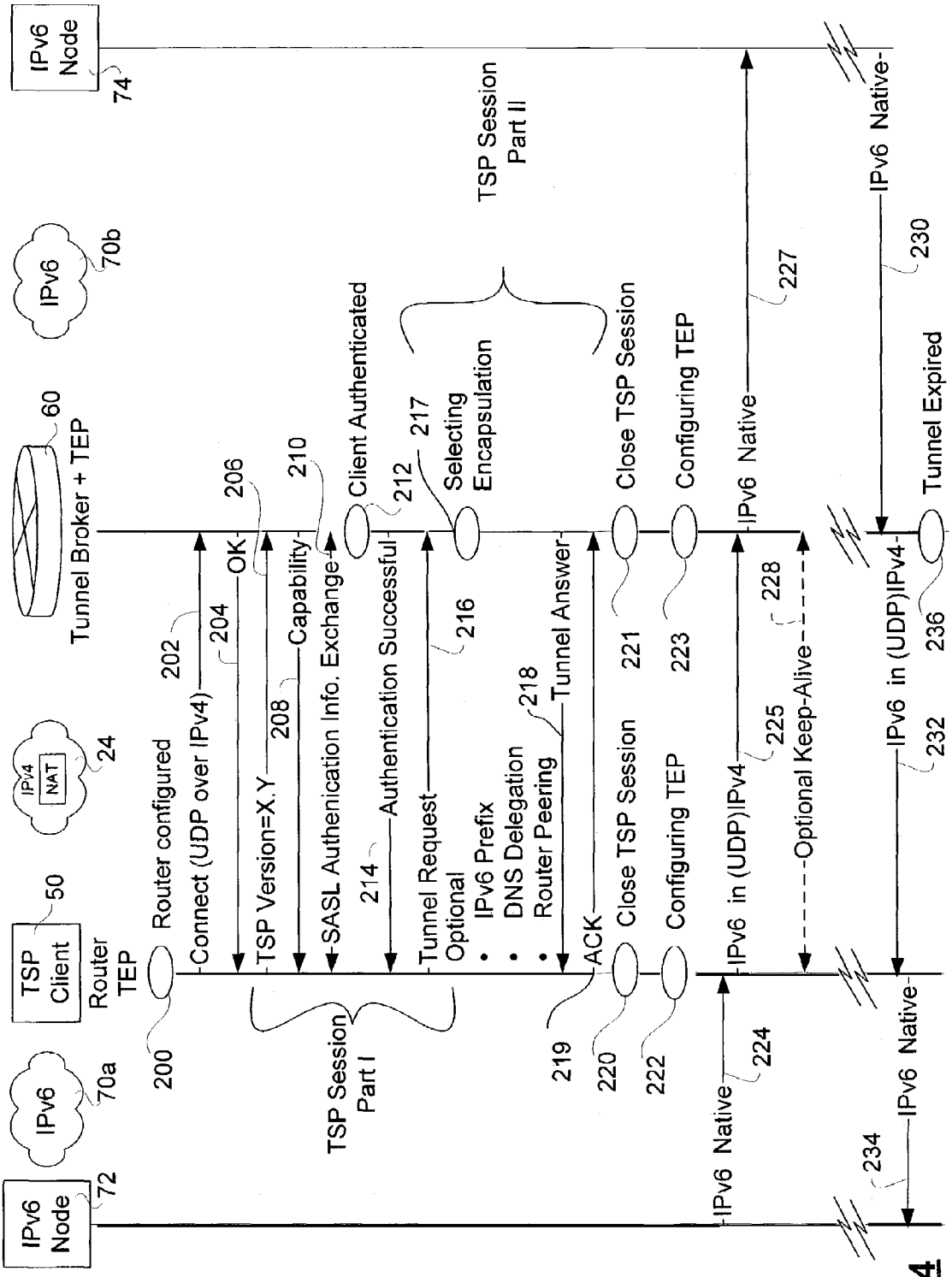

IPv6 Node
74
IPv6
70b
Tunnel Broker + TEP
60
IPv4 NAT
24
TSP Client
50
Router TEP
IPv6
70a
IPv6 Node
72
200
Router configured
202
Connect (UDP over IPv4)
204
OK
206
TSP Version=X.Y
208
Capability
210
SASL Authenication Info. Exchange
Client Authenticated
212
214
Authentication Successful
217
Selecting Encapsulation
TSP Session Part II
TSP Session Part I
Tunnel Request
Optional
• IPv6 Prefix
• DNS Delegation
• Router Peering
216
218
Tunnel Answer
219
ACK
220
Close TSP Session
221
Close TSP Session
222
Configuring TEP
223
Configuring TEP
IPv6 Native
224
IPv6 Native
227
IPv6 in (UDP)IPv4
225
Optional Keep-Alive
228
IPv6 Native
230
IPv6 in (UDP)IPv4
232
Tunnel Expired
236
IPv6 Native
234

FIG. 4

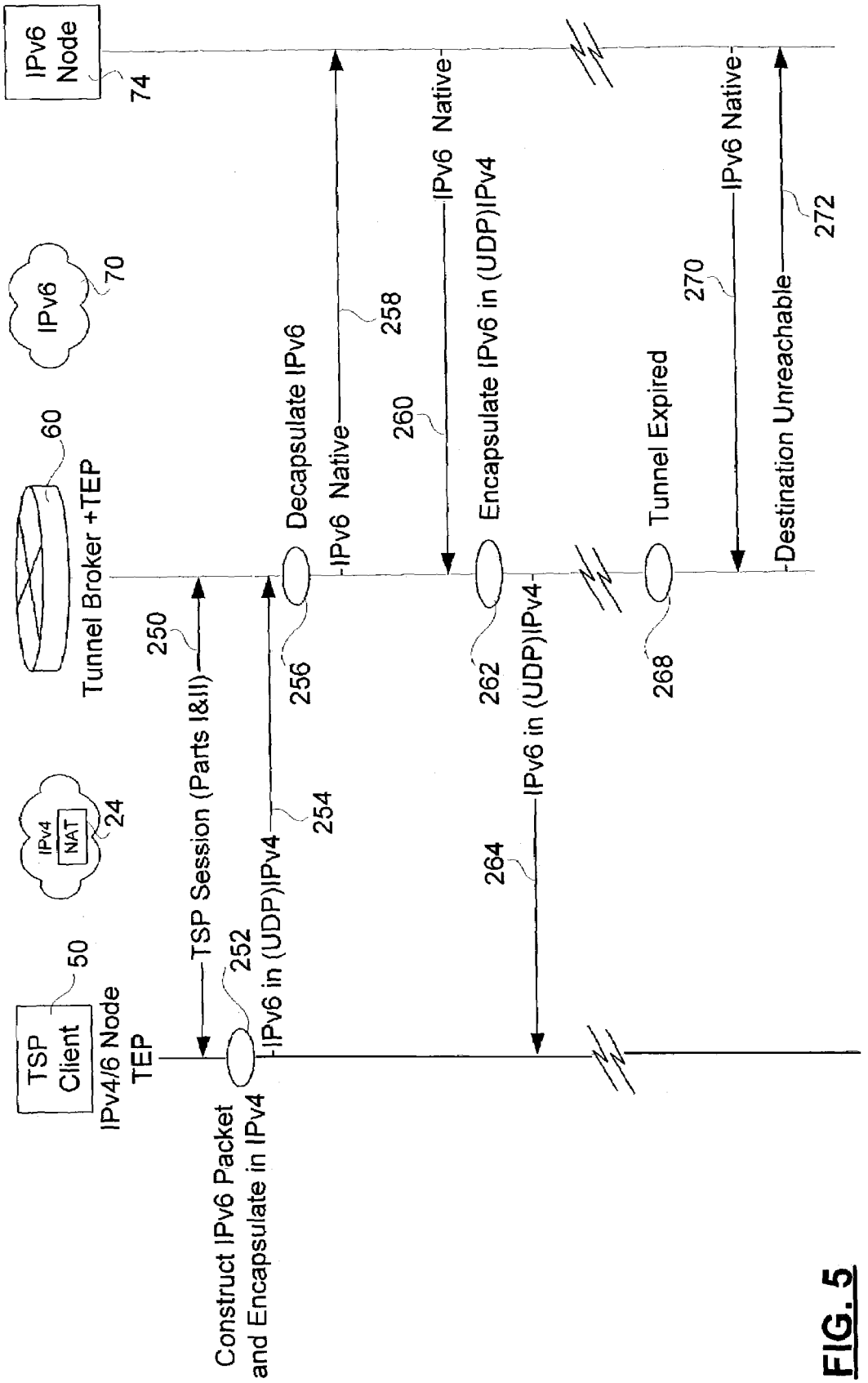

IPv6 Node
74
IPv6
70
60
Tunnel Broker +TEP
IPv4
NAT
24
TSP Client
50
IPv4/6 Node
TEP
250
TSP Session (Parts I&II)
252
Construct IPv6 Packet
and Encapsulate in IPv4
254
IPv6 in (UDP)IPv4
256
Decapsulate IPv6
IPv6 Native
258
260
IPv6 Native
Encapsulate IPv6 in (UDP)IPv4
262
264
IPv6 in (UDP)IPv4
268
Tunnel Expired
270
IPv6 Native
Destination Unreachable
272

FIG. 5

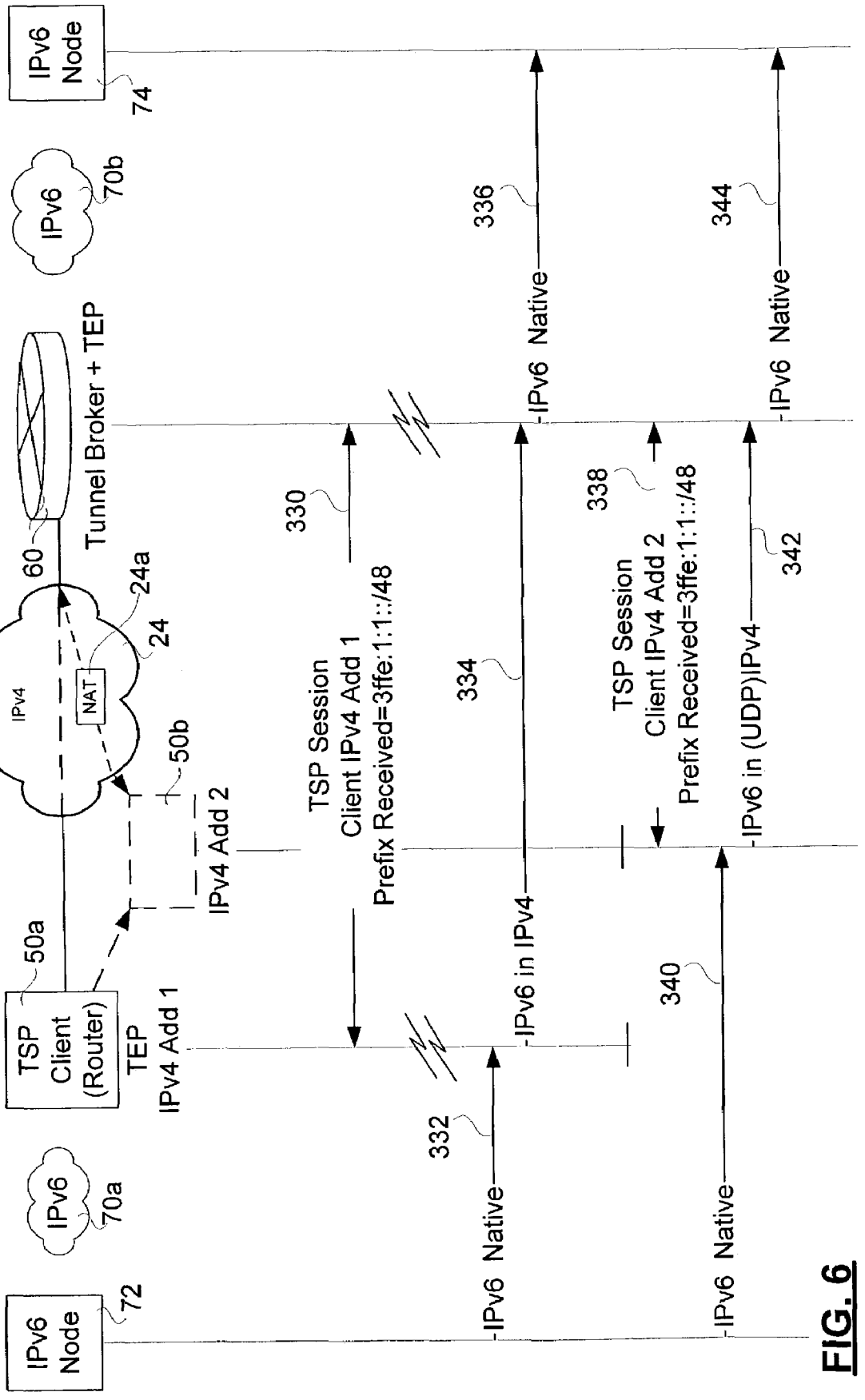

IPv6 Node
72
IPv6
70a
TSP Client (Router)
50a
TEP
IPv4 Add 1
IPv4 Add 2
50b
IPv4
NAT
24
24a
60
Tunnel Broker + TEP
IPv6
70b
IPv6 Node
74
330
TSP Session
Client IPv4 Add 1
Prefix Received=3ffe:1:1::/48
332
IPv6 Native
334
IPv6 in IPv4
336
IPv6 Native
338
TSP Session
Client IPv4 Add 2
Prefix Received=3ffe:1:1::/48
340
IPv6 Native
342
IPv6 in (UDP)IPv4
344
IPv6 Native

FIG. 6

CONNECTING IPV6 DEVICES THROUGH IPV4 NETWORK AND NETWORK ADDRESS TRANSLATOR (NAT) USING TUNNEL SETUP PROTOCOL

CROSS-REFERENCE TO RELATED APPLICATIONS

This is the first application filed for the present invention.

MICROFICHE APPENDIX

Not Applicable.

TECHNICAL FIELD

The invention relates in general to the transition of Internet Protocol (IP) networks from IP version 4 (IPv4) to IP version 6 (IPv6) and, in particular, to a method and apparatus for connecting IPv6 devices through an IPv4 network with network address translation (NAT) using a tunnel setup protocol.

BACKGROUND OF THE INVENTION

Internet Protocol (IP) was created in the 1960's by the United States Advanced Research Projects Agency (ARPA). The Agency's mission was to create instruments useful for military purposes, in particular communications and decentralized computer networks. The original idea was to create connections between military bases using a decentralized communications network with a mesh structure that would permit network function despite significant damage to the country's infrastructure sustained in a military attack. In the early years of its development, the Internet was used for data transfers, principally as file transfer protocol (FTP) sessions.

Use of the Internet spread from the military to the scientific and educational communities in the 1970's and 80's. Propagation of the Internet was, however, slow until the Worldwide Web (WWW) was created. The Worldwide Web was first intended to provide a convenient channel for the transfer of scientific information. However, it caught the attention of the commercial world and in the 1990's an explosive growth of the expansion of the Internet ensued. That explosive growth continues today. The current Internet uses an Internet Protocol referred to as IP version 4 (IPv4). IPv4 uses address fields that are 32 bits long. Although the potential number of IP addresses is 2^{32}, over 70% of those addresses have already been assigned and, if as expected the explosive growth of the Internet continues at its current pace, total exhaustion of IPv4 addresses will occur by 2006. Consequently, the Internet Engineering Task Force (IETF) has developed a new Internet standard referred to as IPv6 which uses 128-bit addressing. The address space in IPv6 is intended to accommodate connection of substantially any intelligent electronic device to the IP network. This includes mobile devices.

It is well known that IPv4 and IPv6 are not compatible because of the differences in address space. Consequently, IPv4 and IPv6 networks can only be interconnected by gateway nodes provisioned with both IPv4 and IPv6 network stacks. However, because of the current lack of available IPv4 address space, IPv6 networks are being deployed and connected to the IPv4 network. A need has therefore arisen for equipment and methods to permit IPv6 devices to communicate across the IPv4 network in order to enhance IPv6 device interconnectivity. This need has been partially

met by an invention described in Applicant's U.S. patent application Ser. No. 10/195,396, now copending, which was filed on Jul. 16, 2002 and describes a method and apparatus for connecting IPv6 devices through an IPv4 network using a tunnel setup protocol, the specification of which is incorporated herein by reference.

While Applicant's invention for providing IPv6 connectivity over an IPv4 network using a tunnel setup protocol represents a significant advance, it is not adapted to accommodate connectivity across all network configurations found in the IPv4 network. One significant problem remains to be addressed. The problem is associated with network address translation (NAT). NAT is used as a an alternative to having a global IPv4 address for each device having access to the Internet. When a local area network (LAN) is connected to the Internet, NAT is generally used at the gateway to the Internet so that each computer in the LAN does not require a globally unique IPv4 address. This permits a private addressing scheme to be used in the LAN, because all traffic to and from the Internet goes through a single external host, which is generally a router.

NAT is frequently built into routers and firewalls. As used in this document, the word "router" means any router, firewall or other gateway configured to relay packets in a data packet network. The routers receive each packet from the internal private network and modify the IP header to include the global IP address of the router in the originating address field, before the packet is transmitted into the Internet. The router stores the internal IP address of the originating node, destination IP address and port number in the NAT state table. When a request is returned to the same port from the destination IP address, the NAT matches the internal IP address that originated the request, and then modifies the IP header to insert the internal originating address as the destination address for the request.

NAT has proved useful in helping to keep IPv4 address available until the conversion to IPv6 is completed. However, as will be understood by those skilled in the art, an IPv6-in-IPv4 tunnel cannot be readily set up through a NAT router, even using the tunnel setup protocol described in applicant's co-pending patent application referenced above.

Proposals for NAT traversal do exist, however. For example, Internet Draft <draft-ietf-ngtrans-shipworm-08.txt>, C. Huitema, (Microsoft) dated Sep. 17, 2002, entitled "Teredo: Tunneling IPv6 over UDP through NAT" proposes a service that enables nodes located behind one, or several, IPv4 NAT(s) to obtain IPv6 connectivity by tunneling packets over User Datagram Protocol (UDP). The service is called the "Teredo" service. Running the service requires the assistance of "Teredo servers" and "Teredo relays". The Teredo servers are stateless, and only have to manage a small fraction of the traffic between Teredo clients. The Teredo relays act as IPv6 routers between the Teredo service and the "native" IPv6 Internet. This represents the first attempt to have NAT traversal for IPv6. However, Teredo does not accomodate any negotiation of parameters (IPv6 prefixes, domain name system (DNS), router peering, etc.), does not handle the optimization of encapsulation, and uses open relays that expose users to important security issues.

Consequently, there exists a need for a method and apparatus for automating and simplifying the establishment of IPv6-in-IPv4 tunnels to enable tunnel setup through a NAT router until the conversion to IPv6 is completed.

SUMMARY OF THE INVENTION

It is therefore an object of the invention to provide a tunnel setup protocol for automating the establishment of IPv6-in-IPv4 tunnels through an IPv4 network with network address translation (NAT).

The invention provides a tunnel setup protocol that facilitates a transition from IPv4 to IPv6 by permitting IPv6 devices to communicate across the IPv4 network, even if the communications are subject to network address translation (NAT). As used in this document, the word "NAT" means one or successive NAT devices in a network path. In accordance with the invention, a control channel is established between a tunnel client and a tunnel broker server. The control channel established between the tunnel client and the tunnel broker server is used to exchange tunnel configuration information and, optionally, to negotiate configuration parameters for the IPv6-in-IPv4 tunnel. After the tunnel configuration parameters have been established, the tunnel broker server configures a tunnel broker server endpoint. If NAT is present in the tunnel path, the tunnel broker client and the tunnel broker server use the tunnel broker endpoint.

The tunnel client also configures a tunnel endpoint, referred to as the tunnel client endpoint for the IPv6-in-IPv4 tunnel. If NAT is present in the tunnel path, the tunnel client endpoint is configured on the tunnel client.

The invention therefore permits the automated establishment of IPv6-in-IPv4 tunnels through an IPv4 network with NAT using a control channel. The use of the control channel enables the automated negotiation of specific configuration details, such as encapsulation selection, IPv6 prefix length, DNS delegation and router peering protocol. This facilitates the deployment of IPv6 networks and ameliorates the transition from IPv4 to IPv6. The invention is particularly useful with mobile devices, since new IPv6-in-IPv4 tunnels can be rapidly and automatically configured to permit true, unencumbered mobility of those devices even if they are behind an IPv4 NAT router, thus enhancing the attraction of deploying IPv6.

BRIEF DESCRIPTION OF THE DRAWINGS

Further features and advantages of the present invention will become apparent from the following detailed description, taken in combination with the appended drawings, in which:

FIG. **1** is a schematic diagram of a point-to-point (PPP) data connection over a dial-up link between a computer and a network access server;

FIG. **2** is a schematic diagram of a connection between an IPv4/IPv6 node and an IPv6 network implemented in accordance with the invention;

FIGS. **3***a*-**3***e* are a flow chart of a method for connecting IPv6 devices through an IPv4 network using UDP and a tunnel setup protocol in accordance with the invention;

FIG. **4** is a connection progress diagram of the establishment and maintenance of an IPv6-in-IPv4 tunnel between a tunnel client and a tunnel broker server using UDP and a tunnel setup protocol, and subsequent use of the tunnel by IPv6 nodes connected to respective IPv6 networks;

FIG. **5** is a connection progress diagram of another implementation of the invention in which a tunnel client connects to a tunnel broker server and establishes an IPv6-in-IPv4 tunnel using UDP for the purposes of communicating with an IPv6 node in an IPv6 network;

FIG. **6** is a connection progress diagram illustrating the establishment of IPv6-in-IPv4 tunnels by a mobile tunnel client that uses an IPv6-in-IPv4 tunnel in a first location and an IPv6-in-UDP/IPv4 tunnel in a second location.

It will be noted that throughout the appended drawings, like features are identified by like reference numerals.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The invention provides a method and apparatus for connecting IPv6 devices through an IPv4 network with network address translation (NAT) using a tunnel setup protocol (TSP) and User Datagram Protocol (UDP) or Transport Control Protocol (TCP), as described in Applicant's Internet-Draft, which bears a date of Jun. 24, 2002 and is entitled "TSP-TEREDO: Stateful IPv6 over IPv4 Tunnels with NAT using TSP and TEREDO, draft-parent-blanchet-ngtrans-tsp-teredo-00.txt", which is incorporated herein by reference.

In accordance with the invention, a control channel is established between a tunnel client and a tunnel broker server using either User Datagram Protocol (UDP) or Transport Control Protocol (TCP), if NAT is performed anywhere in the network between the tunnel client and the tunnel broker. Both the tunnel client and the tunnel broker server must be connected to the IPv4 network. The control channel established between the tunnel client and the tunnel broker server is used to negotiate configuration parameters other than the transport protocol for the IPv6-in-IPv4 tunnel. After the configuration parameters are established, the tunnel broker server configures a tunnel broker server endpoint and the tunnel client configures a tunnel client endpoint for the IPv6-in-IPv4 tunnel. The respective tunnel endpoints are configured on the respective tunnel client and tunnel broker server. The invention therefore permits the automated establishment of IPv6-in-IPv4 tunnels through IPv4 networks with NAT, which facilitates the deployment of IPv6 networks and ameliorates the transition from IPv4 to IPv6.

FIG. **1** is a schematic diagram of a point-to-point (PPP) dial-up connection between a client computer **20** and a network access server **22** to provide access to an IPv4 network **24** in a manner well known in the art. As is well understood, a PPP-control channel **26** is established over the dial-up connection between the client computer **20** and the network access server **22**. The dial-up connection passes through a modem **30**, a switched telephone network **32** and a modem bank **34** in a manner well known in the art. The PPP control channel **26** shares the dial-up connection with a PPP data channel **28**, which is used to send IPv4 data packets from the client computer **20** to one or more selected hosts in the IPv4 network **24**.

FIG. **2** is a schematic diagram illustrating one implementation of a system provisioned with a tunnel setup protocol through an IPv4 network with NAT in accordance with the invention. A control channel **40** is established through the IPv4 network **24** between a tunnel client **50** and a tunnel broker server **60** using User Datagram Protocol (UDP) or Transport Control Protocol (TCP) messaging, although for reasons of efficiency, UDP is preferred. The control channel **40** is used to negotiate parameters for establishing an IPv6-in-IPv4 tunnel through the IPv4 network **24**. The tunnel is used to establish a data channel **42** that extends between first and second tunnel endpoints, the tunnel client **50** and the tunnel broker server **60**. The data channel is used to transfer IPv6 data packets through the IPv4 network using UDP or TCP. The IPv6 data packets are encapsulated at the

opposite endpoints of the IPv6-in-IPv4 tunnel, as will be explained below in more detail.

FIGS. **3***a*-**3***e* are flow diagrams illustrating the tunnel setup protocol in accordance with the invention. The process begins in step **90** when a tunnel setup protocol (TSP) client, hereinafter referred to as a tunnel client **50** (FIG. **2**) connects to a tunnel broker server (TB) **60** using UDP or TCP, as explained above. If the tunnel client **50** is aware that it is behind a NAT in the IPv4 network, the tunnel client **50** preferably uses UDP messaging to establish the control channel **40**, since the protocol used to establish the control channel will also be used to establish the IPv4 tunnel. After the control channel **40** is established, the tunnel client sends the version of the TSP that it supports using the control channel **40** to the tunnel broker server **60** (step **92**). On receipt of the TSP protocol version, the tunnel broker server **60** determines whether it supports the same version of the tunnel setup protocol (step **94**). If it is not provisioned to support the tunnel client's version of the tunnel setup protocol, the tunnel broker server **60** returns an error message via the control channel **40** (step **96**) and branches to connector C (see FIG. **3***e*), where the tunnel broker server **60** determines whether it has an alternate list of tunnel broker servers that it can provide to the tunnel client (as will be explained below in more detail). If the tunnel broker server **60** does support the tunnel client's version of the tunnel setup protocol, the tunnel broker server **60** returns a list of its capabilities (step **98**) to the tunnel client **50** over the control channel **40**. The capabilities of the tunnel broker server **60** include, for example, authentication mechanisms, types of tunnel supported, lengths of IPv6 prefixes that can be assigned, as well as Domain Name Service (DNS) delegation supported, and router peering protocols supported, etc.

In step **100**, the tunnel client **50** determines whether the capabilities of the tunnel broker server **60** are satisfactory for the purposes it requires. If not, the tunnel client **50** closes the tunnel setup protocol session (step **102**) and the process ends. Otherwise, the tunnel client **50** selects an authentication mechanism from the list supported by the tunnel broker server **60** and specifies the authentication mechanism in an authentication message sent via the control channel **40** to the tunnel broker server **60** (step **104**). Subsequently, the tunnel broker server **60** and the tunnel client **50** exchange authentication data (step **106**) via the control channel **40**. In step **108**, the tunnel broker server **60** verifies the tunnel client authentication data.

As shown in FIG. **3***b*, after verifying the tunnel client authentication data, the tunnel broker server **60** determines whether the tunnel client **50** is authorized to establish the tunnel (step **110**). If the tunnel client **50** is not authorized to establish the tunnel, the tunnel broker server **60** returns an error message via the control channel **40** and closes the session (step **112**). If the tunnel client **50** is authorized to establish the tunnel, the tunnel broker server **60** returns an authentication successful message (step **114**) to the tunnel client **50**. The tunnel client **50** then sends a tunnel request message via the control channel **40** (step **116**) to the tunnel broker server **60**. The tunnel request message may include requests for a specific encapsulation or a broker-recommended encapsulation, an IPv6 prefix, a DNS delegation, router peering, etc., as will be explained below in more detail. On receipt of the tunnel request message, the tunnel broker server **60** determines whether the encapsulation protocol has been specified by the tunnel client **50** (step **118**). If an encapsulation protocol has not been specified, the tunnel broker **60** examines the tunnel request message to determine whether an IPv4 source address of the tunnel request message matches an IPv4 client address in the tunnel request message. If there is a match, an IPv6 in IPv4 tunnel can be established in the IPv4 network between the tunnel client **50** and the tunnel broker **60**. Consequently, the tunnel broker **50** returns a recommendation that an IPv6-in-IPv4 tunnel be established (step **122**), which is the most efficient and reliable protocol. If the two addresses do not match, the tunnel broker **60** recommends that an IPv6-in-(UDP/TCP) IPv4 tunnel be established (step **124**). The selection of UDP or TCP depends, as explained above, on which protocol was used to establish the control channel. In either case, the tunnel broker **60** then examines the balance of the tunnel request message to determine if it is provisioned to offer the service as requested (step **128**). If not, the tunnel broker server **60** determines (step **130**) whether it is provisioned to offer a similar service. If not, the tunnel broker server **60** returns an error message via the control channel **40** and branches to connector C, where it determines in step **190** (see FIG. **3***e*) if it is provisioned with a list of alternate tunnel broker servers. If not, it closes the session (step **192**). If so, it returns the list (step **194**) via the control channel **40** to the tunnel client **50** to permit the tunnel client **50** to attempt the establishment of an IPv6-in-IPv4 tunnel using another tunnel broker.

If the tunnel broker is provisioned to provide the requested service or a similar service as determined in steps **128**, **130**, the tunnel broker server **60** assigns an IPv6-in-IPv4 or an IPv6-in-(UDP/TCP)IPv4 tunnel, as determined in steps **120**-**124**, to the tunnel client **50**. The tunnel broker may also assign an IPv6 prefix in a manner well known in the art, provide domain name service (DNS) delegation, as will be explained below in more detail, and router peering to the tunnel client **50**, as appropriate (step **134**, FIG. **3***c*).

In step **136**, the tunnel broker server **60** determines whether DNS delegation has been requested. If so, the tunnel broker server **60** configures its DNS servers for the DNS delegation by registering the tunnel client's DNS server addresses for name space associated with the assigned IPv6 prefix (step **138**) to DNS servers associated with the tunnel broker server **60**. The tunnel broker server **60** also configures its DNS servers with an "AAAA record" (step **140**) for the client tunnel endpoint address, in a manner known in the art. In step **142** (FIG. **3***c*), the tunnel broker server **60** selects and reserves a tunnel endpoint for the tunnel it assigned in step **134**. The configuration of the tunnel endpoint includes configuring router peering. The tunnel broker then awaits confirmation that the tunnel endpoint reservation was successful (step **144**). If the reservation was not successful, the tunnel broker server **60** determines in step **146** whether another tunnel endpoint is available by, for example, consulting a table of tunnel endpoints stored in the tunnel broker server memory (step **146**). If another tunnel endpoint is not available, or all tunnel endpoints are at capacity, the tunnel broker server **60** sends an error message over the control channel (step **148**) to the tunnel client **50** and branches to steps **190**-**194**, as explained above.

If the tunnel endpoint configuration is determined to be successful in step **144**, the tunnel broker server **60** sends the tunnel configuration parameters along with any required IPv6 prefix, DNS information, router peering information, etc. to the tunnel client **50** using the control channel **40**, along with a success code (step **150**). On receipt of this information, the tunnel client determines whether it will accept the tunnel configuration (step **152**). If it does not find the tunnel configuration acceptable, the tunnel client deter-

mines (step **154**) whether it will negotiate a different configuration. It should be noted that the tunnel client may be implemented with or without the capacity for parameter negotiation. If it is not equipped for negotiation or decides to terminate negotiation, the process moves to step **156**, in which the client refuses the tunnel configuration and advises the tunnel broker **60** by sending a refusal message over the control channel **40** (step **156**). On receipt of the refusal message, the tunnel broker server **60** rolls back the configuration of the tunnel endpoint, the DNS configurations, etc. (step **158**) and branches to steps **190-194**, as explained above.

If the client determines in step **154** that it will negotiate the tunnel configuration, it may, for example, assess whether negotiation should proceed by comparing a negotiation count with a predetermined threshold (step **160**). If the negotiation count is greater than the threshold, the process branches to steps **156**, **158** and **190-194**, as explained above. Otherwise, the negotiation counter is incremented (step **162**) and the tunnel client **50** returns via the control channel **40** a revised parameter list to the tunnel broker server **60** and the process branches back to step **118**.

If the tunnel client **50** accepts the tunnel configuration in step **152**, the tunnel client **50** sends an acknowledgement message (ACK) to the tunnel broker server **60** and closes the TSP session (step **166**). On receipt of the ACK message, the tunnel broker server **60** also closes the TSP session (step **168**). The respective TSP sessions are closed because the same channel is used for data traffic. In step **170**, the tunnel broker server then configures the tunnel end point (TEP).

As shown in FIG. **3***d*, the tunnel client **50** configures its tunnel endpoint and, if required, configures its DNS server (s) as explained above, and router peering in its tunnel endpoint, if required (step **172**). The tunnel is thus established and IPv6 traffic can be sent over the established tunnel (step **174**). The tunnel client **50** then determines whether it wants to keep the tunnel setup protocol session alive (step **176**). If so, the tunnel client **50** connects to the tunnel broker server **60** using Tunnel Session Protocol-over-User Datagram Protocol-over-IPv6 (TPS/UDP/IPv6) (step **178**). After the connection is established, as will be explained below in more detail with reference to FIG. **4**, the tunnel client **50** sends a keep-alive message to the tunnel broker server **60** via the control channel **40** (step **180**) and waits for an acknowledgement (ACK) of the keep-alive message (step **182**). If the ACK is received, as determined in step **182**, the tunnel client **50** waits for a predetermined period of time before sending another keep-alive message (step **180**) and the loop (steps **180-184**) is repeated. If an ACK is not received in step **182**, the tunnel client verifies (step **186**) that a retry count threshold has not been exceeded (step **186**). If the retry count is less than a predetermined threshold, the retry count is incremented (step **188**) and after the predetermined time delay (step **184**) the tunnel client **50** repeats steps **180**, **182**. Most NAT devices close the translation table entry when no traffic occurs for some predetermined period of time, however, the keep-alive messages keep the NAT state open for the established tunnel. If the tunnel client **50** determines in step **186** that the retry count exceeds the threshold, the NAT state might have been closed and the tunnel setup process should restart from the beginning (FIG. **3***a*).

FIG. **4** is a connection progression diagram illustrating an exemplary implementation of the tunnel setup protocol in accordance with the invention. In this example, an IPv6-in-IPv4 tunnel is established between a tunnel client **50** and a tunnel broker server **60**, which respectively serve as endpoints for the tunnel. The tunnel client **50** is a router that is connected to an IPv6 network **70***a* and the IPv4 network **24**. Consequently, the tunnel client **50** is provisioned with an IPv4 stack as well as an IPv6 stack and is further provisioned to encapsulate IPv6 packets in IPv4 packets, as well as to decapsulate IPv6 packets encapsulated in IPv4 packets, to permit IPv6 traffic to pass through the tunnel. The tunnel broker server **60** is likewise connected to both the IPv4 network **24** and the IPv6 network **70***b* and provisioned with the same stacks and data encapsulation/decapsulation capability.

As shown in the diagram, in step **200**, the router is configured as a tunnel client **50**. Once configured as a tunnel client **50** so that it knows how to contact the tunnel broker server **60**, the router is provisioned to establish a control channel **40** to the tunnel broker server **60**, as explained above. Subsequently, in step **202**, the tunnel client **50** sends a connect message to the tunnel broker server **60** to establish the control channel **40**. The tunnel client **50** may be prompted to establish the control channel for any number of reasons. For example, the tunnel client **50** is prompted to establish the control channel when the IPv6 node **72** generates IPv6 traffic addressed to an IPv6 node in a different IPv6 network, on reboot, on re-establishing IPv4 re-connectivity, etc. On receipt of the connect message, the tunnel broker server **60** returns an acknowledgement message (step **204**) and the control channel **40** is established. The tunnel client **50** then sends the version of the tunnel setup protocol it supports to the tunnel broker server **60** (step **206**) via the control channel **40**. The tunnel broker server **60** returns, via the control channel **40**, a list of the tunnel setup functions it supports (step **208**). The tunnel client **50** selects an authentication mechanism and authentication information is exchanged (step **210**). In step **212**, the tunnel broker server **60** determines that the tunnel client **50** is authorized for the service and returns an authorization successful message (step **214**). On receipt of the message, the tunnel client **50** formulates a tunnel request message which it sends to the tunnel broker server **60** in step **216**. The request, as explained above, optionally includes a request for an IPv6 prefix, DNS delegation, and a router peering.

On receipt of the request, the tunnel broker server **60** examines the contents of the request to determine if the source address equals the IPv4 address, as explained above with reference to steps **120-124** of FIG. **3***b*. If they do not match, the tunnel broker determines that there is NAT in the control channel path, and selects (step **217**) IPv6 over UDP IPv4, or IPv6 over TCP IPv4, depending on the protocol used to establish the control channel, as explained above. In this example, the tunnel broker selects IPv6 over UDP IPv4. The tunnel broker server **60** then returns a tunnel answer message (step **218**), which includes tunnel configuration parameters, including IPv4 and IPv6 addresses for both the tunnel broker server and the tunnel client endpoints as well as the encapsulation protocol and any other information requested by the tunnel client **50** in step **216**. On receipt of the tunnel answer message, the tunnel client **50** returns an acknowledgement message (ACK) (step **219**) and ends the TSP session (step **220**). On receipt of the ACK sent by the tunnel client in step **219**, the tunnel broker server **60** also ends the TSP session (step **221**).

Meanwhile, the tunnel client **50** configures its tunnel endpoint (step **222**), and the tunnel broker server likewise configures its tunnel endpoint (step **223**). Thereafter, the tunnel client begins to send data traffic through the configured tunnel, as the traffic is received from IPv6 nodes that it services. In step **224**, the tunnel client **50** receives one or more data packets in a native IPv6 protocol from an IPv6

node **72**. The data packets are encapsulated in UDP/IPv4 by the tunnel client **50**, and sent through the tunnel in step **225**. On receipt of the UDP/IPv4 datagrams, the tunnel broker server decapsulates the datagrams and forwards them in the native IPv6 protocol to the addressee (an IPv6 node **74**) (step **227**).

The tunnel client **50** may optionally send keep-alive messages (step **228**) to keep the NAT state open. Keep alive messages use the TSP protocol over UDP/IPv6.

After the tunnel expires (step **236**), tunnel broker server **60** deconstructs the tunnel endpoint, DNS delegation and router peering so that traffic can no longer pass through the tunnel, as will be explained below with reference to FIG. **5**.

FIG. **5** is a connection progression diagram that further explains the process in accordance with the invention. In this example, the tunnel setup protocol client **50** is an IPv4/6 node that serves as a tunnel endpoint. In step **250**, the tunnel protocol session parts I and II are performed as described above with reference to FIG. **4**. In step **252**, the tunnel client **50** starts an IP session by constructing an IPv6 packet and encapsulating the IPv6 packet in a UDP/IPv4 packet in a manner known in the art. The IPv6 packet encapsulated in the UDP/IPv4 packet is sent in step **254** through the tunnel to the tunnel broker server **60**. The tunnel broker server **60** decapsulates the IPv6 packet (step **256**) and forwards it in IPv6 native format to the IPv6 node **74** (step **258**). The IPv6 node **74** returns an IPv6 packet in IPv6 native format (step **260**). The packet is encapsulated in a UDP/IPv4 packet by the tunnel broker server **60** (step **262**) and forwarded through the tunnel in step **264**. In step **268**, the tunnel lifetime expires and the tunnel endpoint is deconstructed, as explained above. Thereafter, when the IPv6 node **74** sends an IPv6 packet in native format (step **270**), the tunnel broker returns a destination unreachable packet (step **272**) in a manner known in the art.

FIG. **6** is a connection progression diagram that illustrates yet another implementation of the system in accordance with the invention. In this example, the tunnel client **50** is a mobile device, such as a cellular telephone, a personal data assistant (PDA) or a laptop computer, which serves as a router in an IPv6 subnetwork. As illustrated, the mobile device in a first location functions as a tunnel client **50***a* having an IPv4 address (Add 1). In the first location, the mobile tunnel client **50***a* commences and performs a tunnel setup protocol session with the tunnel broker (step **330**) and in the course of the tunnel setup protocol session receives an IPv6 prefix from the tunnel broker server **60**. In this example, the prefix received is "3ffe:1:1::/48. As is well known in the art, this prefix is known as a "/48" prefix which permits the tunnel client router to assign session addresses to IPv6 devices in the domain it controls, in a manner well known in the art. After the tunnel is established in step **330**, the IPv6 node **72** is enabled to communicate with the IPv6 node **74** (steps **332-336**) by sending and receiving IPv6 packets in native format.

Subsequently, the mobile tunnel client **50** moves to location **50***b* and its service provider in the IPv4 network assigns a new IPv4 address (Add 2). Consequently, a new tunnel must be established. However, after the move, a NAT **24***a* is introduced into the tunnel path between the tunnel client **50** in location **50***b* and the tunnel broker server **60**. The new tunnel must therefore be setup using UDP (or TCP) over IPv4, as explained above with reference to FIG. **4**. The tunnel client **50***b* and the tunnel broker server **60** therefore initiate and performs the tunnel setup protocol session (step **338**), and the UDP/IPv4 tunnel is established, as explained above.

After the tunnel parameters are negotiated, the tunnel client **50***b* receives the same IPv6 prefix "3ffe:1:1::/48", because the broker recognizes the same client and is provisioned to provide the same prefix, because for the sake of efficiency the client wishes to keep the same prefix. Consequently, a new tunnel is established between the mobile tunnel client **50***b* and the tunnel broker server **60** that permits the IPv6 node **72** to again send IPv6 packets in native format to the IPv6 node **74** (steps **340-344**). By receiving the same IPv6 prefix, the IPv6 node **72** keeps its same IPv6 address. Consequently, in the IPv6 realm the mobility of the IPv4 tunnel end point is not perceived. As will be understood by persons skilled in the art, packets transferred via the tunnel through the IPv4 network (step **334**) are encapsulated with UDP over IPv4 headers because of the NAT **24***a* in the path through IPv4 network **24** established from the new location **50***b* of the tunnel client **50**.

The methods and apparatus in accordance with the invention therefore permit mobile devices to automatically establish IPv6-in-IPv4 tunnels through the IPv4 network, with or without the existence of NAT in the path, to permit IPv6 nodes to communicate with other IPv6 nodes in other IPv6 subnetworks. This is of critical importance to the exponentially expanding use of wireless devices and mobile devices in general, and permits seamless networking of such devices.

The embodiment(s) of the invention described above is(are) intended to be exemplary only. The scope of the invention is therefore intended to be limited solely by the scope of the appended claims.

We claim:

1. A method for connecting an IPv6 devices in a first IPv6 network through an IPv4 network with network address translation (NAT) to an IPv6 node in a second IPv6 network, comprising steps of:

sending a message from a tunnel client to a tunnel broker server to establish a control channel through the IPv4 network between the tunnel client the tunnel broker server, the tunnel client being connected to the IPv4 network and the first IPv6 network, and the tunnel broker server being connected to the IPv4 network and the second IPv6 network;

sending to the tunnel broker server, via the control channel, a request message to establish an IPv6-in-IPv4 tunnel through the IPv4 network, the request including tunnel configuration parameters desired by the tunnel client;

determining at the tunnel broker server whether network address translation (NAT) occurs between the tunnel client and the tunnel broker server;

when the NAT occurs between the tunnel client and the tunnel broker, setting up the IPv6-in-IPv4 tunnel through the NAT using a tunnel setup protocol (TSP) session, between the tunnel client and the tunnel broker server, and subsequently maintaining a NAT state of the NAT open to preserve the IPv6-in-IPv4 tunnel for at least a duration of a communications session between the IPv6 node and the IPv6 device;

receiving at the tunnel broker server, from the tunnel client, a version of a tunnel session protocol installed at the tunnel client;

determining whether the version of the tunnel session protocol is supported by the tunnel broker server; and

when the version of the tunnel session protocol is not supported by the tunnel broker server, returning an error message to the tunnel client.

2. The method as claimed in claim 1 wherein the step of determining whether there is NAT between the tunnel client and the tunnel broker server comprises comparing a source address extracted from an IPv4 message encapsulating the request message with an IPv4 tunnel client endpoint address specified in the request.

3. The method as claimed in claim 1 wherein after the step of returning an error message, the method further comprises a step of returning, from the tunnel broker server to the tunnel client, a list of alternate tunnel broker servers to permit the tunnel client to attempt to obtain service from another tunnel broker server.

4. The method as claimed in claim 1 further comprising, when the tunnel broker server supports the version of the tunnel session protocol, a step of returning, service capabilities of the tunnel broker server to the tunnel client.

5. The method as claimed in claim 4 wherein the service capabilities comprise a specification of authentication types, and the method further comprises steps of selecting by the tunnel client an authentication type, and sending authentication information to the tunnel broker server to permit the tunnel broker server to verify that the client is authenticated for the service.

6. The method as claimed in claim 1 wherein the step of sending the message through the IPv4 network comprises a step of formulating either one of a Transfer Control Protocol (TCP) and an User Datagram Protocol (UDP) message that is sent to the tunnel broker server to establish the control channel.

7. The method as claimed in claim 1 wherein the step of sending the request message comprises a step of formulating tunnel configuration parameters, comprising a tunnel action type, a tunnel type, and an IPv4 tunnel endpoint address.

8. The method as claimed in claim 7 wherein the tunnel configuration parameters further comprise a request for an IPv6 prefix of a specified length, and a domain name service (DNS) delegation and router peering.

9. The method as claimed in claim 1 wherein the step of receiving the acceptance from the tunnel broker server comprises a step of receiving information specifying a tunnel lifetime, a tunnel client endpoint IPv4 address, a tunnel client endpoint IPv6 address, a tunnel broker server endpoint IPv4 address, and a tunnel broker server endpoint IPv6 address, and an indication that the control channel is subject to NAT and the control channel must be used as a data channel.

10. The method as claimed in claim 9 wherein subsequent to receiving the information, the tunnel client further performs a step of closing the TSP session and configuring the tunnel client endpoint using the tunnel client endpoint IPv4 address and the tunnel client endpoint IPv6 address.

11. The method as claimed in claim 1 wherein prior to sending an acceptance of the request with a specification of information respecting the tunnel configuration parameters desired by the tunnel client, the tunnel broker server performs a step of reserving a tunnel broker server endpoint.

12. The method as claimed in claim 11 further comprising, when the step of reserving the tunnel broker server endpoint is unsuccessful, a step of returning an error message to the tunnel client, along with a refusal to establish the tunnel.

13. The method as claimed in claim 12 wherein the tunnel broker server further returns a list of alternate tunnel broker servers to the tunnel client, to permit the tunnel client to attempt to establish a tunnel using another tunnel broker server.

14. The method as claimed in claim 1 wherein after the tunnel client receives an acceptance of the request with a specification of information respecting the tunnel configuration parameters desired by the tunnel client, the tunnel client periodically sends a keep-alive message to the tunnel broker server to maintain the NAT state opened to preserve the tunnel.

15. An Apparatus for connecting an IPv6 device in a first IPv6 network through an IPv4 network with network address translation (NAT) to an IPv6 node in a second IPv6 network, comprising:

a tunnel broker server connected to the IPv4 network and the second IPv6 network, the tunnel broker server being programmed to:

respond to a message from a tunnel client establishing a control channel through the IPv4 network between the tunnel client and the tunnel broker server, the tunnel client being, connected to the iPv4 network and the first IPv6 network;

authenticate the tunnel client to establish an IPv6-in-IPv4 tunnel through the IPv4 network;

accept desired parameters for a configuration of the IPv6-in-IPv4 tunnel from the tunnel client;

determine whether or not network address translation (NAT) occurs between the tunnel client and the tunnel broker; and

when the NAT occurs between the tunnel client and the tunnel broker, setting up the IPv6-in-IPv4 tunnel through the NAT using a tunnel setup protocol (TSP) session, between the tunnel client and the tunnel broker server, and subsequently maintaining a NAT state of the NAT open to preserve the IPv6-in-IPv4 tunnel for at least a duration of a communications session between the IPv6 node and the IPv6 device;

receiving at the tunnel broker server, from the tunnel client, a version of a tunnel session protocol installed at the tunnel client;

determining whether the version of the tunnel session protocol is supported by the tunnel broker server; and

when the version of the tunnel session protocol is not supported by the tunnel broker server, returning an error message to the tunnel client.

16. The Apparatus as claimed in claim 15 wherein the tunnel broker server is further programmed to return to the tunnel client an encapsulation specification and parameters for a configuration of the IPv6-in-IPv4 tunnel after accepting the desired parameters from the tunnel client.

17. The Apparatus as claimed in claim 15 wherein the tunnel broker server is programmed to return a list of other tunnel broker servers which may be used by the tunnel client, when the tunnel broker server is not able to provide service in accordance with desired parameters for a configuration of the IPv6-in-IPv4 tunnel from the tunnel client.

18. The Apparatus as claimed in claim 16 wherein the tunnel broker server is programmed to configure a tunnel end point after returning parameters for a configuration of the IPv6-in-IPv4 tunnel to the tunnel client and closing the TSP session.

19. The Apparatus as claimed in claim 15 wherein the tunnel client is programmed to:

establish a control channel with the tunnel broker server;

provide authentication information to the tunnel broker server to permit the tunnel broker server to authenticate the tunnel client;

provide to the tunnel broker desired parameters for a configuration of the tunnel;

accept from the tunnel broker an encapsulation specification and parameters for the configuration of the tunnel;

configure a tunnel endpoint given the parameters for the configuration of the tunnel after acknowledging acceptance of the configuration of the tunnel and closing the TSP session; and

encapsulate packets to be sent through the tunnel using the encapsulation specification provided by the tunnel broker.

20. The Apparatus as claimed in claim **19** wherein the tunnel client is a router and is further programmed to request an IPv6 prefix of a specified length when providing the tunnel broker server with the desired parameters for a configuration of the tunnel.

21. The Apparatus as claimed in claim **19** wherein the tunnel client is programmed to configure itself as the tunnel endpoint.

22. The Apparatus as claimed in claim **19** wherein the tunnel client is programmed to maintain the NAT state open by periodically sending keep-alive messages to the tunnel broker server.

23. The Apparatus as claimed in claim **22** wherein the tunnel client encapsulates the keep-alive messages using tunnel setup protocol (TSP) over User Datagram Protocol (UDP) over IPv6 over UDP over IPv4.

24. A system for connecting an IPv6 device in a first IPv6 network through an IPv4 network with network address translation (NAT) to an IPv6 node in a second IPv6 network using a tunnel setup protocol (TSP) session, comprising:

a tunnel client connected to the IPv4 network and the first IPv6 network;

a tunnel broker server connected to the IPv4 network and the second IPv6 network, the tunnel broker server being programmed to respond to a message sent from the tunnel client to establish a control channel between the tunnel client and the tunnel broker server, use the control channel to authenticate the tunnel client attempting to establish an IPv6-in-IPv4 tunnel through the IPv4 network, and accept parameters for a configuration of the IPv6-in-IPv4 tunnel sent by the tunnel client via the control channel;

the tunnel broker server and the tunnel client being respectively programmed to configure a tunnel endpoint for the IPv6-in-IPv4 tunnel, to determine at the tunnel broker server whether network address translation (NAT) occurs between the tunnel client and the tunnel broker server, and when the NAT occurs to set up the tunnel through the NAT using a tunnel setup protocol (TSP) session, between the tunnel client and the tunnel broker server, and subsequently maintaining a NAT state of the NAT open to preserve the IPv6-in-IPv4 tunnel for at least a duration of a communications session between the IPv6 node and the IPv6 device;

the tunnel broker server further programmed to:

receive from the tunnel client, a version of a tunnel session protocol installed at the tunnel client;

determine whether the version of the tunnel session protocol is supported by the tunnel broker server; and

when the version of the tunnel session protocol is not supported by the tunnel broker server, returning an error message to the tunnel client.

25. The system as claimed in claim **24** wherein the tunnel client is a host in the IPv4 network.

26. The system as claimed in claim **24** wherein the tunnel client is a router having an IPv4 stack and an IPv6 stack, and at least one link to each of the IPv4 and IPv6 networks.

27. The system as claimed in claim **24** wherein the tunnel broker server is programed to assign an IPv6 prefix to be used by the tunnel endpoint for a duration of the IPv6-in-IPv4 tunnel.

28. The system as claimed in claim **27** wherein the tunnel client is programmed to request the IPv6 prefix from the tunnel broker client.

* * * * *

Berikan 1 contoh isu penelitian (Research Problem) yang bisa diangkat dari permasalahan Network Address Translation:

Jawab :

Keterbatasan alamat IP lokal dapat menjadi masalah besar pada sebuah perusahaan yang berkembang. Dikarenakan jumlah alamat IP yang terbatas tidak sebanding dengan jumlah pengguna yang semakin bertambah.

Nama : Nurul Amalina Setyorini
NIM : 202420005
Jurusan : Magister Teknik Informatika
Kelas : Regular B

Pada pertemuan ini saya melampirkan sebuah video yang menjelaskan tentang proses Network Adress Translation. Proses ini berada pada di antara layer 3 dan layer 2 OSI.

Tugas: Berikan 1 contoh isu penelitian (Research Problem) yang bisa diangkat dari permasalahan Network Address Translation.

Pada video tersebut di jelaskan bagaimana proses *Network Address Translation* dengan IPv4 dan IP address 128.243.35.9 serta menjelaskan bagaimana kerja NAT (*Network Address Translation*). Di dalam NAT terdapat IP, TCP dan Data sebagai source portnya.

Isu penelitian yang bisa diangkat dari permasalahan tersebut adalah bagaimana membandingkan kerja *Network Address Translation* dengan system *Proxy*. Hampir mirip dengan NAT, suatu jaringan kecil dengan *proxy* bisa menempatkan beberapa mesin untuk mengakses *web* dibelakang sebuah mesin yang memiliki IP *address* valid. Ini juga merupakan langkah penghematan biaya dibanding harus menyewa beberapa account dari ISP dan memasang modem & sambungan telepon pada tiap mesin.

Namun demikian, *proxy* server ini tidak sesuai untuk jaringan yang lebih besar. Bagaimanapun, menambah *hard disk* dan RAM pada server *proxy* supaya *proxy* berjalan efisien tidak selalu dapat dilakukan (karena *constraint* biaya). Lagi pula, persentase *web page* yang bisa dilayani oleh *cache proxy* akan makin menurun sejalan dengan semakin menipisnya ruang kosong di *hard disk*, sehingga penggunaan *cache proxy* menjadi tidak lebih baik dari pada sambungan langsung. Tambahan lagi, tiap koneksi bersamaan akan meng-*generate* proses tambahan dalam *proxy*. Tiap proses ini harus menggunakan *disk I/O channel* yang sama, dan saat *disk I/O channel* jenuh, maka terjadilah *bottle neck*.

NAT menawarkan solusi yang lebih fleksibel dan *scalable*. NAT menghilangkan keharusan mengkonfigurasi *proxy*/*sock* dalam tiap *client*. NAT lebih cepat dan mampu menangani trafik *network* untuk beribu-ribu *user* secara simultan.

Selain itu, translasi alamat yang diterapkan dalam NAT, membuat para *cracker* di Internet tidak mungkin menyerang langsung sistem-sistem di dalam jaringan internal. *Intruder* harus menyerang dan memperoleh akses ke mesin NAT dulu sebelum menyiapkan serangan ke mesin-mesin di jaringan internal. Penting di ketahui bahwa, sementara dengan NAT jaringan internal terproteksi, namun untuk masalah *security*, tetap saja diperlukan paket *filtering* dan metoda pengamanan lainnya dalam mesin NAT.

Nama : Rachmad Iqbal
Nim : 202420002

Contoh Kasus *Network Address Translation (NAT)*

Sebuah perusahaan kecil memiliki sejumlah komputer dan sambungan ke Internet. Komputer-komputer itu saat ini telah membentuk suatu LAN. Sambungan Internet-nya diasumskan berupa *dedicated T1 link*

Langkah-langkah yang harus dilakukan

1. **Installasi FreeBSD**

   Sediakan satu komputer untuk dijadikan *Gateway*. Penulis menyarankan penggunaan **FreeBSD RELEASE 2.2.6** (Natd hanya jalan di FreeBSD 2.2.1 ke atas), karena selain gratis juga *requirement hardware*-nya tidak terlalu boros. PC 486 dengan 16 MB *memory* dan HD 850 MB juga sudah cukup mewah.

   Untuk mengetahui proses installasi FreeBSD, silahkan baca kembali tulisan-tulisan di Infokomputer sebelumnya dan manual FreeBSD sendiri.

2. **Installasi *Gateway***

   Pasang 2 *network interface* agar mesin ini menjadi *gateway*. *Network Card* (misal NE2000 atau 3COM) satu dihubungkan ke jaringan internal dan satu lagi untuk koneksi ke ISP. Misalnya dua-duanya NE2000 *Compatible*. maka *nick* untuk *card* yang menghadap ke dalam adalah ed0 dan untuk card yang menghadap keluar adalah ed1.

   Pastikan juga option gateway = ”YES” tertulis dengan benar dalam *file* rc.conf. Atau bisa juga dengan mengetik perintah: sysctl -w net.inet.ip.forwarding=1

3. **Installasi *Firewall***

   Pasang IP *firewall* di mesin FreeBSD ini. Caranya adalah :

   a. Edit *kernel source* di /usr/src/sys/i386/conf
      Tambahkan *option-option* berikut ini pada file *kernel*.

options	IPFIREWALL
options	IPFIREWALL_VERBOSE
options	“IPFIREWALL_VERBOSE_LIMIT=100”
options	IPDIVERT

   b. Compile *kernel* tersebut
   c. Aktifkan *firewall* di rc.conf dengan menambahkan

firewall="YES"
firewall_type="OPEN"

3. Installasi Nat

Langkah-langkahnya adalah sbb:

a. *Download source* nya di ftp://ftp.suutari.iki.fi/pub/natd
b. *Unzip* dan *untar archive* tersebut dengan perintah
   gzip -dc natd_1.12.tar.gz | tar -xvf -
c. Lakukan *make* dan *make install* di direktori yang dihasilkan. Ketikkan perintah berikut:
   cd natd_1.12
   make
   make install
d. Edit *startup file* supaya Natd berjalan secara otomatis
   Buat file natd.sh di /usr/local/etc/rc.d. Isi file tersebut adalah

```
#!/bin/sh
    /sbin/ipfw -f flush
    /sbin/ipfw add divert 13494 ip from any to any via ed0
    /sbin/ipfw add pass all from 127.0.0.1 to 127.0.0.1
    /sbin/ipfw add pass ip from any to any
    /usr/local/sbin/natd -port 13494  -interface ed0
```

Arti dari file ini adalah:

- Hapuskan semua rule *firewall*
- Tambahkan feature *divert* di *port* 13494 (Anda bisa mengganti ini dengan *port* yang Anda inginkan) untuk mendiversi paket dari dan ke *gateway* lewat *interface* ed0
- Bolehkan semua paket lewat di atas local host
- Bolehkan semua paket IP lewat semua *interface*
- Jalankan Natd dengan menjadi *daemon* yang menunggu di *port* 13494 via *interface* ed0.

e. Reboot mesin FreeBSD-nya supaya setting bisa diaktifkan.

4. Konfigurasikan TCP/IP *Client*.

Jadikan nomor IP *card* ed0 di FreeBSD sebagai *gateway* dari tiap *workstation*, IP tiap-tiap *work station* harus berada dalam *network* yang sama dengan *card* ed0 yang ada di mesin *gateway*. Misal ed0 di-beri nomor IP 192.168.1.1 dan ed1 167.205.19.5, maka *workstation* diberi nomor IP 192.168.1.2 s/d 192.168.1.14 jika digunakan *mask* 16 atau 255.255.255.240. ed1 adalah *interface* yang memiliki IP *address* valid

Setelah semuanya langkah-langkah di atas dijalankan dengan baik maka, applikasi Internet di *client* siap dijalankan via NAT.

Untuk kasus lain misalnya sambungan ke Internet-nya menggunakan modem, maka mekanismenya sama saja, tinggal diganti *interface* di *gateway* yang menghadap keluar dengan *interface* modem (tun0) dan jalankan program ppp untuk men-*dial* ISP-nya. Khusus untuk *dial-out*, ppp sebenarnya memiliki mekanisme sendiri untuk kasus ini yaitu dengan option -alias. Jadi jika kita menjalankan ppp dengan option -alias maka kita tidak perlu menjalankan Natd, karena option ini menyediakan fasilitas yang sama dengan Natd khusus untuk dial-out.

Natd hanyalah salah satu cara untuk menghemat persediaan IP *address* yang semakin menipis. Dengan adanya fakta bahwa untuk bergabung ke Internet, *host* pencari informasi (*Client*) sebenarnya tidak perlu memiliki IP *address* legal, maka IP *address* legal tersebut bisa dicadangkan untuk *host-host* penyedia informasi (*Server*). Penelitian untuk terus memperbaiki performansi Internet ini masih terus dikembangkan. Sekarang ini juga sedang dikembangkan model IP versi baru yaitu IP versi 6 (IPv6), yang bisa menampung lebih banyak lagi komputer-komputer di Internet. Namun demikian untuk kondisi sekarang, Natd masih merupakan solusi ampuh sebelum IPv6 diterapkan.

Nama : Trada Ayang Pratiwi

NIM : 2015210046

Network Address Translation Merupakan sebuah sistem untuk menggabungkan lebih dari satu komputer untuk dihubungkan ke dalam jaringan internet hanya dengan menggunakan sebuah alamat IP *(Internet Protocol)*. Sehingga setiap komputer di dalam NAT ketika berselancar di internet akan terlihat memiliki alamat IP yang sama jika dilacak. Dengan kata lain, sebuah alamat IP pada jaringan lokal akan terlebih dahulu ditranslasikan oleh NAT untuk dapat mengakses IP Publik dijaringan komputer. Sebelum proses translasi ini, maka pengguna tidak dapat terhubung ke internet.

Dari video tentang proses *Network Addres Translation* yang telah dijelaskan , dapat di jadikan sebuah *Research Problem "Network Address Translation Penghubung Ip Public Dan Ip Private"*

Dimana nantinya akan dibangun sebuah Topologi Local Network dengan IP Private untuk terhubung dengan IP Public. Saat menggunakan NAT, seorang klien dapat terhubung dengan internet melalui proses-proses berikut :

1. Pertama-tama, NAT menerima permintaan dari klien berupa paket data yang ditujukan untuk sebuah server remote di internet.

2. NAT kemudian mencatat alamat IP klien, lalu menyimpannya ke dalam tabel translasi alamat. Selanjutnya, alamat IP komputer klien tersebut diubah oleh NAT menjadi nomor IP NAT, lalu NAT lah yang akan melakukan permintaan kepada server.

3. Server kemudian merespon permintaan tersebut. Dari sudut pandang server, yang terlihat adalah alamat IP NAT, bukan alamat IP klien yang meminta data bersangkutan.

4. NAT menerima respon dari server, lalu melanjutkannya dengan mengirimkan ke alamat IP klien yang bersangkutan.

5. Keempat tahapan tersebut terjadi berulang-ulang, sehingga walaupun klien komputer tidak memiliki alamat IP publik, namun tetap dapat mengakses internet.

NAT sendiri dapat digunakan jika jumlah IP yang dimiliki sedikit sedangkan komputer yang akan disambungkan ke internet cukup banyak. Penggunaan mekanisme NAT dalam jaringan dapat menghemat biaya karena efisien dalam pemakaian IP Public dan penerapan NAT dalam jaringan dapat meningkatkan efisiensi manajemen LAN dalam internetworking.

Nama : Yudistira Sira Permana
NIM : 182420104
Kelas : MTI 23 A

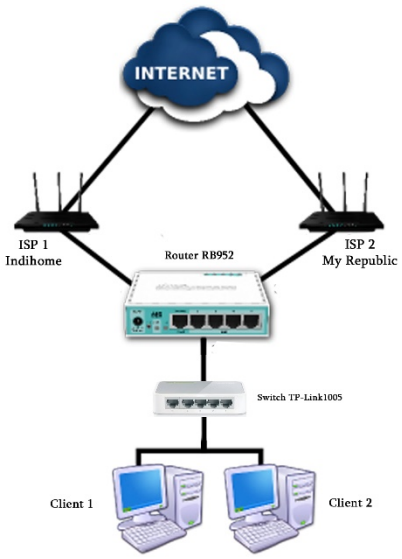
Configurasi NAT pada Load Balancing

Stasiun LRT Punti Kayu Palembang.

Stasiun LRT Punti Kayu Palembang adalah salah satu stasiun yang melayani angkutan LRT Palembang. Stasiun ini berada dekat dengan Taman Wisata Punti Kayu dan Gramedia World Palembang. Stasiun LRT yang dibuka pada tanggal 24 September 2018 ini mengalami kendala tidak stabilnya *bandwitd* dengan dua ISP yang berbeda yaitu Indihome dan MyRepublik. Maka dari itu deperlukannya *Load Balancing* kedua ISP tersebut menggunakan MikroTik. *Load Balancing* adalah aspek kunci untuk menghindari situasi di mana beberapa node kelebihan beban sementara yang lain menganggur atau memiliki sedikit pekerjaan yang harus dilakukan (Jafarnejad Ghomi et al., 2017).

Load balance mempunyai beberapa metode diantaranya seperti *Per Connection Classifier* (PCC), Nth (bilangan ke-n), *Equal Cost Multi Path* (ECMP), dan *Static Route* dengan *Address List*. Dari kelebihan dan kekurangan tersebut penulis membandingkan Metode Nth dan *Per Connection Classifier* (PCC) dikarenakan Metode terbaik jika berdasarkan data *Throughput* dan rata-rata *Delay* dengan lebar *bandwidth* yang sama pada kedua jalurnya adalah Nth karena mampu membagi data menjadi paket-paket yang dikirim melalui jalur yang disediakan (Khusaini Dwi Cahyono, Mimin F Rohmah, 2018).

Perancangan fisik merupakan perancangan sebuah struktur jaringan yang berhubungan dengan peralatan yang digunakan dan pembentukan sebuah topologi jaringan (Utomo, 2011). Adapun perancangan topologi jaringan sebagai berikut :



Perancangan Logic bertujuan untuk membagi fungsi dari masing-masing perangkat sesuai strata dan beban fungsionalitas jaringannya (Pamungkas & Prayitno, 2018). Berikut adalah tabel IP address dari desain topologi jaringan yang telah pada **Perancangan Fisik**.

Perangkat	Interface	IP Address	Gateway
Mikrotik RB952	ISP-1 (eth2) ISP-2 (eth3) Lokal (eth4)	DHCP DHCP 192.168.10.1	- - -
Switch Hub	Ethernet	-	-
PC Client 1	Ethernet	192.168.10.4	192.168.10.1
PC Client 2	Ethernet	192.168.20.6	192.168.20.1

Konfigurasi *load balancing* memerlukan beberapa tahapan, yang pertama adalah melakukan konfigurasi dasar. Pada tahap ini, yang pertama dilakukan adalah melakukan konfigurasi *interface* yang digunakan sebagai jalur keluar masuk internet lewat router mikrotik. Dan setelah melalui pemeriksaan awal, kemudian menetapkan koneksi dengan ISP dan melakukan permintaan alamat IP (IP-DHCP). Selanjutnya melakukan konfigurasi IP *address* pada masing masing *Ethernet* dan DNS yang akan digunakan.

Setelah melakukan konfigurasi IP dan DNS, selanjutnya harus menambahkan konfigurasi NAT (*Network Address Translation*). NAT berguna agar *client* dapat terhubung dengan internet. NAT akan mengubah alamat sumber paket yaitu alamat *client* yang memiliki IP *address private* agar dapat dikenali oleh internet yaitu dengan cara mentranslasikannya menjadi IP *address public*. Pengaturan NAT ini menggunakan metode *Masquerading* NAT. Karena provider yang digunakan hanya memberkan satu IP *public*, jadi semua IP *address* dari *client* akan dipetakan kepada satu IP *public.*

Chain	Out Interface	Action
Scrnat	ISP1	Masquerade
Scrnat	ISP2	Masquerade

Jika konfigurasi NAT sudah selesai maka PC client sudah dapat terkoneksi dengan jaringan internet internet.

Nama : Yusria Lenitasari
NIM : 202420003
Jurusan : Magister Teknologi Informatika
Tugas 01 : *Computer Network and Communication*

---

1. Berikan 1 contoh isu penelitian *(Research Problem)* yang bisa diangkat dari permasalahan *Network Address Translation.* Tuliskan jawaban anda pada ms word, kemudian upload pada assignment ini.

   Jawab :
   Isu penelitian :
   Dengan terkoneksinya jaringan internet dan intranet pada satu host. Bagaimana cara menghemat alamat IP pada kasus tersebut ? dan apakah keamanan data intranet dapat terancam jika host tersebut terkoneksi pada internet ?

   Hasil Penelitian :
   Konsep NAT merupakan solusi keterbatasan alamat IP. IP *masquerading* merupakan salah satu kemampuan tambahan pada system operasi LINUX yang menerapkan konsep NAT. Implementasi IP *masquerading* pada intranet dengan menggunakan konsep NAT menyebabkan identitas setiap host dalam internet tersamarkan. Layanan IP *masquerading* bersifat statefull karena paket balasan dari internet yang diperbolehkan masuk adalah paket balasan dari koneksi yang dibuat dari dalam intranet sehingga dapat meningkatkan keamanan data intranet. Untuk membangun layanan IP *masquerading* hal yang dibutuhkan adalah satu buah koneksi (*account*) di internet, satu line telpon dan satu buah modem untuk menghubungkan semua host intranetnya untuk melakukan koneksi ke internet.

Aldo Fajarino
202420004
MTI Reg B

Berikan 1 contoh isu penelitian (Research Problem) yang bisa diangkat dari permasalahan Network Address Translation

ANALISA INTERKONEKSI INTERNET PROTOCOL SECURITY PADA JARINGAN KOMPUTER BERBASIS NETWORK ADDRESS TRANSLATION
Oleh Helmi Kurniawan , Iwan Fitrianto Rahmad

Seiring dengan meningkatnya pengguna jaringan internet, penggunaan alamat IP (Internet Protocol) yang terdaftar di jaringan internet juga meningkat. Untuk mengatasi permasalahan tersebut, diperlukan suatu metode yang dapat mengefisienkan penggunaan alamat IP, metode tersebut yaitu Network Address Translation (NAT). Metode ini telah banyak diimplementasikan pada Internet Service Provider (ISP), Small Office Home Office (SOHO) dan perusahaan-perusahan menengah ke atas, yang memungkinkan jaringan pribadi dengan alamat IP yang tidak terdaftar di jaringan internet dapat berkomunikasi dengan jaringan internet melalui satu atau lebih alamat IP yang terdaftar di jaringan internet. Internet Protocol Security (IPsec) merupakan suatu set ekstensi protokol yang dikembangkan oleh Internet Engineering Task Force (IETF) sebagai standar mekanisme sistem keamanan pada layer IP.

Di antara NAT dan IPsec terdapat perbedaan mekanisme mendasar yang membuat perangkat IPsec di jaringan internet tidak dapat berkomunikasi dengan perangkat IPsec yang berada dibelakang perangkat NAT, hal ini dapat dilihat dari tujuan fundamental IPsec, yaitu untuk menjaga kerahasiaan data dan keutuhan data pada layer IP, sedangkan mekanisme dari NAT justru melakukan modifikasi pada IP agar jaringan pribadi yang berada di belakangnya dapat berkomunikasi dengan jaringan publik atau internet dan begitu pula sebaliknya. Inkompatibilitas antara mekanisme kerja IPsec dan NAT telah menjadi suatu halangan dalam pengembangan implementasi IPsec sebagai standar mekanisme keamanan di layer IP, berangkat dari permasalahan tersebut, penulis tertarik untuk menganalisa mekanisme kerja dari IPsec sehingga dapat berinteraksi dengan jaringan komputer yang mengimplementasikan NAT.

IPsec merupakan suatu set ektensi protokol dari Internet Protocol (IP) yang dikeluarkan oleh Internet Engineering Task Force (IETF). Istilah dari IPsec mengacu pada suatu set dari mekanisme yang didesain untuk mengamankan trafik pada level IP atau pada network layer. Teknologi dari IPsec ini didasari oleh teknologi modern dari kriptografi, dimana layanan keamanan yang disediakan antara lain yaitu:

Confidentiality : Untuk mejamin kerahasiaan dimana sulit bagi pihak yang tidak berwenang untuk dapat melihat atau mengerti kecuali oleh penerima yang sah bahwa data telah dikirimkan. Integrity : Untuk menjamin bahwa data tidak berubah dalam perjalanan menuju tujuan.

Authenticity : Untuk menjamin bahwa data yang dikirimkan memang berasal dari pengirim yang benar.

Anti Reply : Untuk menjamin bahwa transaksi hanya dilakukan sekali, kecuali yang berwenang telah mengijinkan untuk mengulang transaksi. Terdapat dua protokol yang berjalan di belakang Ipsec yaitu: Authentication Header (AH), menyediakan layanan authentication, integrity, replay protection pengamanan pada header IP, namun tidak menyediakan layanan confidentiality

Network Address Translation merupakan suatu metode dimana IP address dipetakan dari satu grup ke grup lainnya secara transparan bagi sisi penerima. Network Address Port Translation (NAPT) merupakan suatu metode dimana alamat-alamat jaringan beserta port TCP/UDP-nya ditranslasikan menjadi satu alamat jaringan beserta port TCP/UDP-nya. Secara bersamaan kedua operasi di atas mengacu kepada Traditional NAT, yang menyediakan mekanisme komunikasi antara jaringan lokal (alamat IP tidak terdaftar di internet) dengan jaringan global (alamat IP terdaftar di internet)

1. Dua Tipe NAT Dua tipe NAT adalah Statik dan Dinamik yang keduanya dapat digunakan secara terpisah maupun bersamaan

a. Statik

Translasi Statik terjadi ketika sebuah alamat lokal (inside) di petakan ke sebuah alamat global/internet (outside). Alamat lokal dan global dipetakan satu lawan satu secara statik.

b. Dinamik

NAT dengan Pool (kelompok) Translasi Dinamik terjadi ketika router NAT diset untuk memahami alamat lokal yang harus ditranslasikan, dan kelompok (pool) alamat global yang akan digunakan untuk terhubung ke internet. Proses NAT Dinamik ini dapat memetakan bebarapa kelompok alamat lokal ke beberapa kelompok alamat global.

Dari penelitian yang dilakukan diambil beberapa kesimpulan yaitu

NAT Overload Sejumlah IP lokal/internal dapat ditranslasikan ke satu alamat IP global/internet. Hal ini sangat menghemat penggunakan alokasi IP global dari ISP. Pemakaian bersama satu alamat IP ini menggunakan metoda port multiplexing, atau perubahan port ke outbound packet yang disebut juga dengan metode Network Address Port Translation (NAPT)

IPsec dapat diimplementasikan pada jaringan berbasis NAT, dengan catatan, tidak diimplementasikannya protokol AH dan hanya mengimplementasikan protocol ESP. Karena mekanisme kerja AH mengikutsertakan IP asal dan IP tujuan dalam, NAT maupun NAT reverse bekerja dengan mengubah field address pada IP header yang akan membuat pengecekan terhadap integritas data oleh AH gagal karena data dianggap sudah tidak valid lagi. Karena protokol ESP tidak mengikutsertakan IP asal dan IP tujuan dalam integritas datanya, maka inkompatibilitas ini tidak terjadi terhadap ESP.

TCP dan UDP checksums memiliki ketergantungan terhadap IP asal dan IP tujuan melalui penambahan dari pseudo header dalam perhitungannya. Sehingga ketika checksums dikalkulasi dan di cek pada sisi pengirim, hasilnya tidak sama ketika melewati perangkat NAT atau NAT reverse. IPsec ESP hanya dapat melalui NAT jika tidak melibatkan protokol TCP/UDP (IPsec pada mode tunnel atau IPsec dienkapsulasi oleh GRE), atau tidak ada

pengecekan checksum, hal ini dapat dilakukan oleh UDP IPv4, karena pengecekan checksum pada UDP IPv4 bersifat opsional, sementara pada TCP IPv4, checksum diperlukan.

DAFTAR RUJUKAN

Aboba, B., dan Dixon, W. IPsec-Network Address Translation (NAT) Compatibility Requirements. RFC 3715. Maret 2004.

Holdrege, M., dan Srisuresh, P. Protocol Complications with the IP Network Address Translator. RFC 3027, Januari 2001.

Halsall, Fred. Data Communication Computer Networks and Open Systems Fourth Edition. Addisnon-Wesley. 1996.

Kent, S., dan Atkinson, R. Security Architecture for the Internet Protocol. RFC 2401. November 1998. Kent, S., dan Atkinson, R. IP Authentication Header. RFC 2402, November 1998.

Kent, S., dan Atkinson, R. IP Encapsulating Security Payload (ESP). RFC 2406. November 1998. Luthfi, Muhammad. Pengamanan Komunikasi Data Pada Jaringan Internet Menggunakan protokol IP Security. Tugas Akhir. Teknik Elektro Telekomunikasi STT Telkom. 2001.

Perdana, Indrajaya Pitra. Analisa dan Implementasi Mekanisme Interkoneksi Internet Protocol Security (IPsec) dengan Network Address Translation (NAT) pada Jaringan Komputer. Tugas Akhir. Teknik Elektro Telekomunikasi STT Telkom. 2005.

Postel, J. User Datagram Protocol (UDP). STD 6. RFC 768. Agustus 1980.

Postel, J. Transmission Control Protocol (TCP) Specification. STD 7. RFC 793. September 1981. Postel, J., dan Reynolds, J. File Transfer Protocol (FTP). STD 9. RFC 959. Oktober 1985.

Rosenberg, J. STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs). RFC 3489. Maret 2003.

Srisuresh, P., dan Holdrege, M. IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663. Agustus 1999.

Srisuresh, P. Security Model with Tunnel-mode IPsec for NAT Domains. RFC 2709. Oktober 1999. Srisuresh, P., dan Egevang, K. Traditional IP Network Address Translator (Traditional NAT). RFC 3022. Januari 2001.

TUGAS
NETWORK ACCESS
KELAS MTI 23 A

Di Buat Oleh :

Ari Hardiyantoro Susanto

NIM : 202420015

Dosen Pengasuh :

Dr. Edi Surya Negara,S.Kom., M.Kom.

Program Pasca Sarjana

Universitas Bina Darma Palembang

2020/2021

Soal : Berikan 1 contoh isu penelitian (Research Problem) yang bisa diangkat dari permasalahan Network Address Translation

Jawaban :

Network Address Translation Merupakan sebuah sistem untuk menggabungkan lebih dari satu komputer untuk dihubungkan ke dalam jaringan internet hanya dengan menggunakan sebuah alamat IP *(Internet Protocol)*. Sehingga setiap komputer di dalam NAT ketika berselancar di internet akan terlihat memiliki alamat IP yang sama jika dilacak. Dengan kata lain, sebuah alamat IP pada jaringan lokal akan terlebih dahulu ditranslasikan oleh NAT untuk dapat mengakses IP Publik dijaringan komputer. Sebelum proses translasi ini, maka pengguna tidak dapat terhubung ke internet.

Dari video tentang proses *Network Addres Translation* yang telah dijelaskan , dapat di jadikan sebuah *Research Problem "Network Address Translation Penghubung Ip Public Dan Ip Private"*

Dimana nantinya akan dibangun sebuah Topologi Local Network dengan IP Private untuk terhubung dengan IP Public. Saat menggunakan NAT, seorang klien dapat terhubung dengan internet melalui proses-proses berikut :

1. Pertama-tama, NAT menerima permintaan dari klien berupa paket data yang ditujukan untuk sebuah server remote di internet.

2. NAT kemudian mencatat alamat IP klien, lalu menyimpannya ke dalam tabel translasi alamat. Selanjutnya,alamat IP komputer klien tersebut diubah oleh NAT menjadi nomorIP NAT, lalu NAT lah yang akan melakukan permintaan kepada server.

3. Server kemudian meresponpermintaan tersebut. Dari sudutpandang server, yang terlihat adalah alamat IP NAT, bukanalamat IP klien yang meminta databersangkutan.

4. NAT menerima respon dari server, lalu melanjutkannya dengan mengirimkan ke alamat IP klien yang bersangkutan.

5. Keempat tahapan tersebut terjadi berulang-ulang, sehingga walaupun klien komputer tidak memiliki alamat IP publik, namun tetap dapat mengakses internet.

NATsendiri dapat digunakan jika jumlah IP yang dimiliki sedikit sedangkan komputer yang akan disambungkan ke internet cukup banyak. Penggunaan mekanisme NAT dalam jaringan dapat menghemat biaya karena efisien dalam pemakaian IP Public dan penerapan NAT dalam jaringan dapat meningkatkan efisiensi manajemen LAN dalam internet working.