

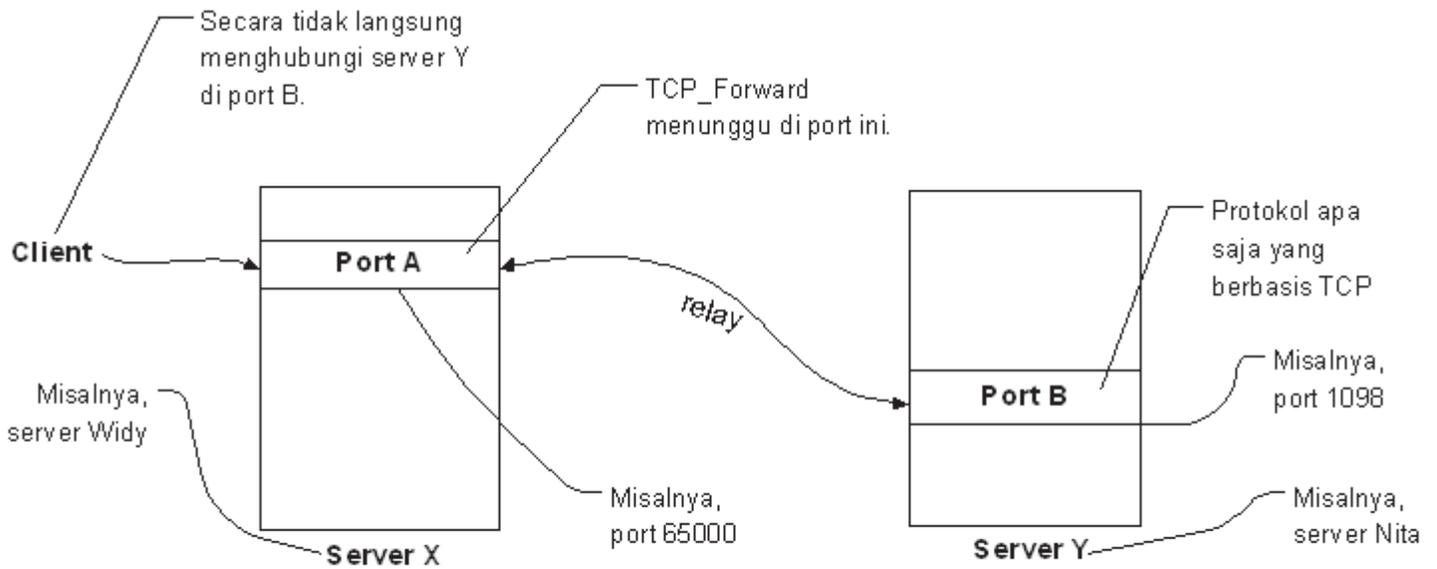
Menghubungkan dua komputer pada jaringan yang berbeda melalui suatu "gateway" tanpa merubah konfigurasi network dengan cara mem-forward data-data pada koneksi stream TCP.

Kata orang, masalah selalu merupakan awal dari suatu penemuan. Ternyata hal itu benar. Karena ada "masalah" (bagi para user, tentunya, bukan bagi sysadmins yang kadang memerintah dengan tangan besi... ;-)) di jaringan kampus, lalu saya merasa tertantang untuk memecahkannya.

Mungkin saya perlu cerita kilas baliknya dulu. Di lab komputer kampus pakai intranet berbasis TCP/IP, OS-nya Linux (berbagai distribs). Nah, untuk akses Internet harus lewat proxy, katakanlah nama server proxy-nya Widy (pakai huruf kapital biar gampang membacanya; tapi ini sebenarnya nama server lho... bukan nama orang). Biar bisa nge-web, kami musti telnet ke Widy, atau set proxy server-nya ke dia. Waktu itu awal semester, nggak tau apa karena Widy baru di-upgrade atau hanya ganti kernel; intinya proxy-nya Widy ngadat. Jadi kalau mau nge-web musti telnet ke Widy trus pakai Lynx. Berhubung Lynx cuma text-mode, saya pikir-pikir gimana pemecahannya, gimana biar bisa nge-browse pakai Netscape (dari terminal X-Windows di lab).

Setelah diusut-usut, ternyata dalam mengakses Internet, Widy memakai host lain, namanya Nita. Nampaknya host ini di-share untuk semua fasilitas Internet universitas (melayani berbagai fakultas). Tapi, sayangnya, Nita tidak bisa langsung diakses dari terminal lab, musti lewat Widy. Setelah merenung di bis ;-)) saya akhirnya mendapat pemecahannya.

Idenya sih sebenarnya sederhana saja. Saya buat suatu program server (yang notabene tidak harus jalan sebagai root) yang menerima request (koneksi TCP) dari terminal-terminal lab, trus nge-forward request-request tersebut ke Nita, namanya TCP_Forward. Servernya jalan di Widy, menunggu request di port yang nilainya besar, katakanlah 65000. Kalau ada client yang connect ke Widy:65000 (syntax host:portnum), TCP_Forward bakal connect ke Nita:1098 (proxy port). Lalu, setiap data yang dikirim oleh client bakal di-forward ke Nita, dan sebaliknya. Server Nita sendiri bakal tetap mengira bahwa yang connect ke dia adalah Widy, namun tidak menyadari bahwa sebenarnya adalah terminal dari lab.



Fungsi TCP_Forward

Mekanisme ini agak mirip dengan konsep IP Masquerading di Linux, hanya saja tidak sepenuhnya transparan, dan tidak perlu ada modifikasi pada kernel. Program dapat dijalankan dan di-kill oleh user biasa. Untuk jelasnya bisa lihat diagram pada [Gambar 1](#) yang mengilustrasikan kegunaan TCP_Forward.

Cara kerjanya sederhana saja, dan mirip dengan cara kerja server Unix pada umumnya. Program menunggu pada port tertentu (diambil dari command line), dan setiap ada client X yang connect, program akan meng-clone dirinya dengan `fork()`. Child process yang terbentuk akan membuat connection ke server Y:port_Y (juga dari command line). Setiap data dari X diteruskan ke Y, dan sebaliknya. Event diagram a'la James Martin pada [Gambar 2](#) akan menjelaskan garis besar cara kerja TCP_Forward.

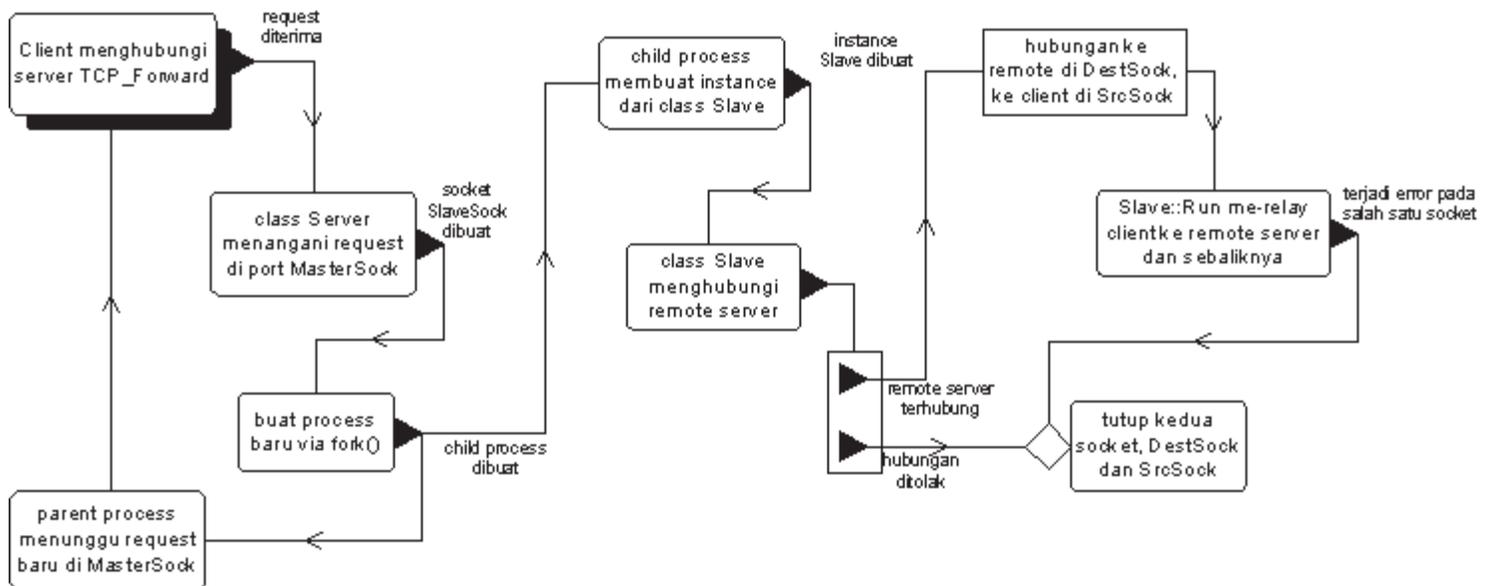


Diagram Kerja TCP_Forward

Program menggunakan fungsi `select()` untuk mencapai semi-asynchronous I/O antara sumber (client) dengan target host yang dituju. Disebut "semi" karena hanya operasi "read" yang menggunakan `select()`. Dengan itu, `TCP_Forward` dapat merespon lebih cepat dari pihak (X atau Y) yang telah siap dengan data. Karena `TCP_Forward` tidak dirancang untuk protokol tertentu (tetapi harus TCP), jadi tidak diketahui pada saat "kapan" pihak yang "mana" telah siap dengan data.

Setelah dicoba-coba lagi, ternyata `TCP_Forward` tidak hanya manjur untuk protokol HTTP Proxy, tapi juga bisa untuk HTTP (biasa), dan Telnet. Hasilnya cukup memuaskan. Sayangnya, implementasi yang saya buat masih boros CPU (kadang sampai 20% CPU usage, waktu saya cek pakai 'top' di Widy, walau saya kurang tahu CPU-nya apa) dan terbatas pada protokol TCP (connection-oriented), bukan pada UDP yang connectionless.

Anyway, sekarang proxy server-nya Widy sudah aktif lagi, jadi `TCP_Forward` jarang dipakai. Cuma kadang-kadang aja proxy-nya ngadat, jadi `TCP_Forward` nggak benar-benar mubazir, masih kadang dipakai. Mungkin para pembaca bisa menyarankan kegunaan lain untuk `TCP_Forward` (anonymous browsing, mungkin?).

Para pembaca yang tertarik dapat memodifikasi `TCP_Forward` agar bisa lebih efisien dalam penggunaan CPU, mungkin juga menambah kemampuannya agar bisa mem-forward protokol UDP. Mungkin dapat digunakan fasilitas multithreading (*lightweight process*) yang tersedia di Linux sehingga program dapat lebih efisien dalam penggunaan memory; daripada menggunakan fungsi `fork()` yang meng-clone seluruh address-space program (yang notabene boros banget). Tokh programnya berlisensi GNU General Public Licence -- open-source. Okay? ;-)

Cheers,
Sasmito Adibowo.
adibs@yahoo.com

Pustaka:

Martin, James. *Principles of Object Oriented Analysis and Design*. Prentice Hall. 1993.

Comer, Douglas E. David L. Stevens. *Internetworking with TCP/IP volume III*. Prentice Hall. 1993.