PERANGKAT LUNAK *TRAFFIC CONFIGURATOR* DAN *TRAFFIC MONITOR* UNTUK PENGATURAN TRAFIK JARINGAN BERBASIS PROTOKOL TCP/IP DAN *LIBRARY PACKET CAPTURE*

Royyana M. Ijtihadie, Febriliyan Samopa, Hindrawan Aris
Jurusan Teknik Informatika,
Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Telp. + 62 31 5939214, Fax. + 62 31 5913804
Email : roy@its-sby.edu, iyan@its-sby.edu

ABSTRAK

Dengan semakin banyaknya orang yang mengakses internet untuk mendapatkan informasi dan tidak adanya pengaturan pada trafik data maka akan mengakibatkan jaringan menjadi lambat. Pengaturan trafik jaringan yang ada di Linux selama ini masih menggunakan scripts yang relatif lebih sulit untuk digunakan dan dimengerti oleh sebagian orang sehingga diperlukan alat bantu untuk mengkonfigurasikannya.

Paper ini menjelaskan tentang pembuatan Traffic Management Configurator untuk mempermudah pengaturan traffic dalam gateway yang menghubungkan jaringan yang berbeda

Aplikasi Network Monitoring digunakan untuk menghitung utilitas dan statistik jaringan. Kedua aplikasi tersebut menggunakan web sebagai interface kepada pengguna.

***Kata Kunci** : Network Monitoring, PCAP, Traffic Control, HTB, HTB GUI.*

1. PENDAHULUAN

Penggunaan akses internet secara massal mengakibatkan turunnya performansi jaringan seiring dengan peningkatan jumlah pengguna. Apalagi jika bandwidth yang ada tidak dikelola sebaik mungkin.

Traffic Control (TC) memegang peranan yang sangat penting dalam hal ini. Linux sebagai suatu sistem operasi yang bersifat open dan free, telah menawarkan berbagai teknik TC untuk memfasilitasi proses manajemen bandwidth pada suatu jaringan. Salah satunya adalah dengan menggunakan teknik TC Hierarchical Token Bucket (HTB), yang menjamin para pengguna jaringan mendapatkan bandwidth yang sesuai dengan yang telah didefinisikan, dan juga terdapat fungsi pembagian bandwidth yang adil di antara para pengguna jaringan sehingga performansi jaringan tetap dapat terjaga.

Salah satu kendala dari TC Hierarchical Token Bucket (HTB) di Linux adalah masih menggunakan perintah console yang relatif lebih sulit untuk digunakan dan dimengerti oleh sebagian orang. Sehingga dibutuhkan waktu yang lebih lama dan mungkin biaya yang lebih besar bagi seorang administrator jaringan untuk bisa menggunakannya dengan benar.

2. TCP/IP (TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL)

TCP/IP adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data pada Wide Area Network (WAN). TCP/IP terdiri dari sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Berkat adanya pembagian tanggung jawab, tugas masing-masing protokol menjadi jelas dan sederhana. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain, selama ia masih bisa saling mengirim dan menerima data.

Sekumpulan protokol TCP/IP pada umumnya dimodelkan dengan empat lapis TCP/IP, sebagaimana terlihat pada Gambar 1.

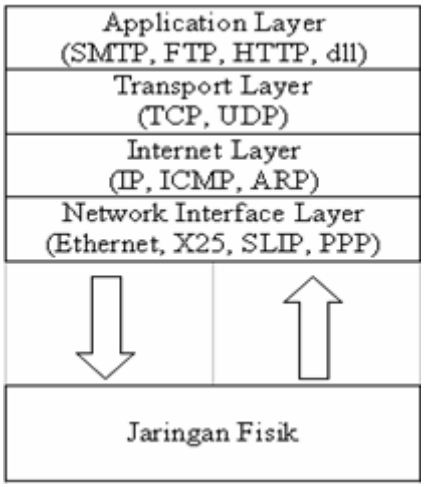
Internet Protocol (IP) merupakan inti dari TCP/IP. IP menyediakan jasa pengiriman paket yang merupakan dasar pembentuk jaringan TCP/IP. Semua protokol, dalam lapis di atas dan bawah IP, menggunakan IP untuk mengantarkan data ke tujuan. Semua data TCP/IP baik yang masuk maupun keluar mengalir lewat IP, tanpa memperhatikan tujuan akhirnya.

Datagram adalah format paket yang didefinisikan oleh IP. Sebuah datagram IP berisi header IP dan data, dan dikelilingi oleh *Media Access Control*

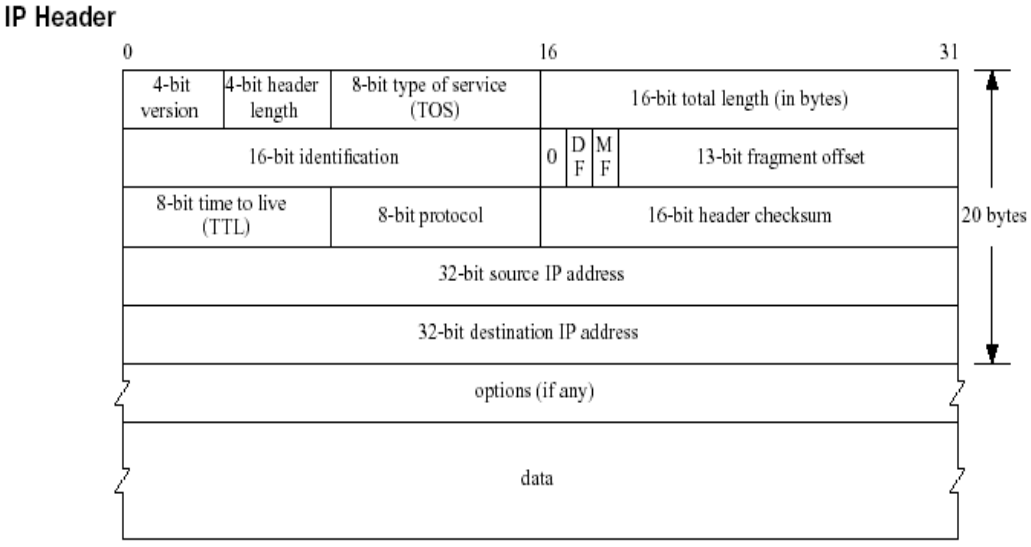
(MAC) header dan MAC trailer. Satu pesan dapat dikirim dalam urutan datagram-datagram yang disusun kembali menjadi pesan asli pada sisi penerima. Format dari datagram IP dapat di lihat pada Gambar 2.

Pada TCP/IP yang sederhana, hanya ada dua tipe dari peralatan jaringan –gateway dan host. Gateway mengirimkan/meneruskan paket di antara jaringan, dan host tidak. Bagaimanapun juga, jika host dihubungkan ke lebih dari satu jaringan, host bisa berfungsi sebagai gateway yaitu meneruskan paket antar jaringan.

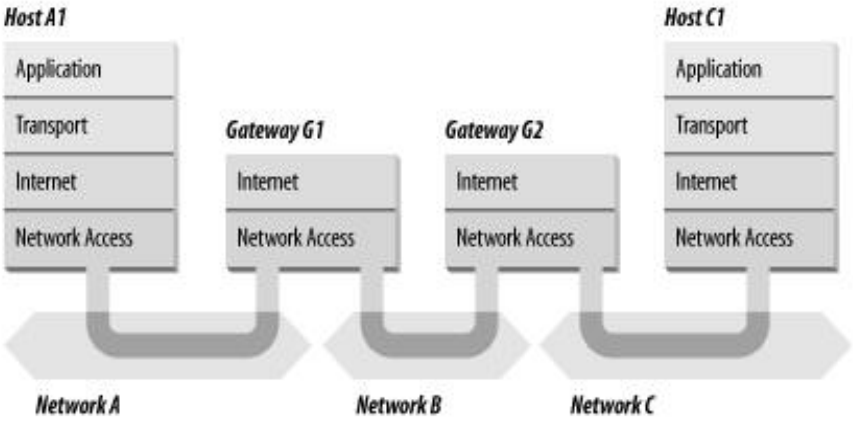
Gambar 3 menunjukan penggunaan gateway untuk meneruskan paket. Host akan melewatkan dan memproses paket pada keempat lapis protokol, sedangkan gateway memproses paket hanya sampai ke Internet Layer dimana keputusan routing (jalur paket) dibuat.



Gambar 1. Layer TCP/IP



Gambar 2. IP Header



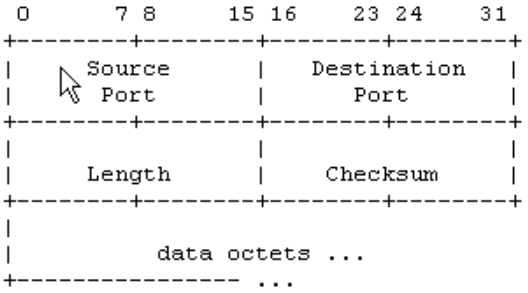
Gambar 3 Routing Packet TCP

Sistem hanya bisa mengirimkan paket ke peralatan lain yang berada pada jaringan fisik yang sama. Paket dari host A1 yang ditujukan kepada host C1 akan diteruskan melewati gateway G1 dan G2. Pertama, host A1 mengirimkan paket ke gateway G1, yang berada pada jaringan yang sama dengannya. Gateway G1 meneruskan paket ke G2 melalui jaringan B.Kemudia gateway G2 langsung mengirimkan paket ke host C1 yang berada satu jaringan dengannya. Host A1 tidak mengetahui apa-apa tentang gateway yang berada di luar G1. Demikian juga jika host C1 ingin mengirimkan paket ke host A1, dia harus melewatkan paketnya ke gateway G2.

2.1. UDP (USER DATAGRAM PROTOCOL)

User Data Protocol berfungsi untuk memungkinkan pengiriman datagram pada sistem komunikasi komputer dengan pertukaran paket dalam lingkungan jaringan komputer. UDP menggunakan IP sebagai protokol dibawahnya.

UDP menyediakan layanan bagi aplikasi untuk melakukan pengiriman pesan dengan mekanisme protokol yang minimum. Hal ini disebabkan karena UDP bersifat transaction oriented, sehingga tidak menjamin sampainya data serta tidak menjamin tidak adanya duplikasi. Gambar Header UDP dapat kita lihat pada gambar 4.



Gambar 4. UDP Header

2.2. LIBRARY PACKET CAPTURE (LIBPCAP)

Packet Capture Library merupakan library yang menyediakan antarmuka tingkat tinggi (high-level interface) dalam suatu sistem penangkapan paket (*packet capture system*). Pada Sistem Operasi, BPF (*Berkeley Packet Filter*) merupakan sistem packet capture. Library ini juga menyediakan subrutin untuk penguna (user level subroutine) yang merupakan antarmuka dengan BPF sehingga memungkinkan penguna untuk melakukan pembacaan terhadap lalu lintas pada jaringan. Dengan menggunakan Packet Capture Library, memungkinkan pengguna untuk menulis sendiri alat untuk monitoring jaringannya. Aplikasi yang menggunakan Packet Capture Library Subroutine harus dijalankan oleh root user. Tabel 1 menunjukkan struktur data pada LibpCap.

Tabel 1. Struktur Data PCAP

Struktur Data	Keterangan
struct pcap_file_*header*	Data struktur ini mendefinisikan pada *record* pertama dalam ***savefile*** yang berisi data dari paket yang sudah ditangkap disimpan
struct pcap_pkthdr	Data struktur ini mendefinisikan *header* dari paket yang ditambahkan diawal setiap paket yang disimpan di ***savefile***
struct pcap_stat	Ini merupakan data struktur yang dikembalikan oleh subrutin **pcap_stat,** berisi informasi yang berhubungan dengan statistik paket dari awal sesi penangkapan sampai saat subrutin **pcap_stat dipanggil**

2.3. TRAFFIC CONTROL (TC)

Traffic Control (TC) adalah istilah yang diberikan kepada satu kesatuan sistem dan mekanisme dalam penerimaan dan pengiriman paket pada router. Termasuk didalamnya pengambilan keputusan tentang paket mana yang harus diterima dengan kecepatan berapa pada masukan interface dan menentukan urutan pengiriman paket beserta kecepatannya pada keluaran dari interface.

TC merupakan satu set alat yang mengijinkan pengguna untuk bisa mengatur queue (urutan) dan mekanisme pengurutan paket dari peralatan jaringan. Kemampuan alat ini dalam mengurutkan kembali aliran traffic dan paket sangat luar biasa dan bisa menjadi rumit, tapi itu bukan apa bila dibandingkan dengan hasilnya dalam menyediakan bandwith yang memadai untuk semua. Istilah lain untuk TC adalah Quality of Service (QoS).

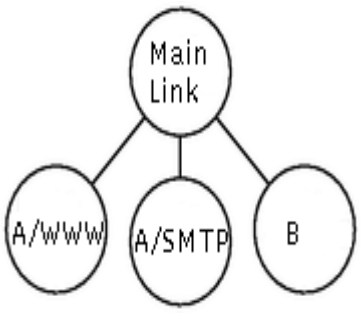
Salah satu keuntungan yang mungkin diperoleh jika menerapkan TC secara benar adalah bisa lebih memprediksi penggunaan sumber daya jaringan dan bisa mencegah terbuangnya sumber daya yang ada. Dengan ini bisa mengatur agar bandwidth tidak hanya akan dipakai untuk lalu lintas bulk download saja, tetapi juga untuk lalu lintas paket yang lain. Bahkan bisa dilakukan alokasi bandwith untuk pengiriman data dengan prioritas rendah seperti email tanpa mempengaruhi lalu lintas yang lain.

Ada banyak alasan kenapa menggunakan TC dan banyak skenario dalam penggunaan TC, dibawah ini beberapa contoh masalah yang mungkin bisa dipecahkan dengan TC:

- Membatasi bandwidth total ke kecepatan tertentu.
- Membatasi bandwidth dari pengguna, service atau komputer tertentu.
- Memaksimalkan keluaran TCP pada hubungan asymetri.
- Menyimpan bandwidth untuk aplikasi atau pengguna tertentu.
- Mencegah keterlambatan dalam lalu lintas data.

2.4. HTB (HIERARCHICAL TOKEN BUCKET)

HTB merupakan salah satu disiplin antrian yang memiliki tujuan untuk menerapkan link sharing secara presisi dan adil. Dalam konsep link sharing, jika suatu kelas meminta kurang dari jumlah service yang telah ditetapkan untuknya, sisa bandwidth akan didistribusikan ke kelas-kelas yang lain yang meminta service.



Gambar 5. Link Sharing

HTB menggunakan TBF sebagai estimator yang sangat mudah diimplementasikan. TBF sangat mudah diset karena banyak dari administrator jaringan yang memiliki ilmu tentangnya. Estimator ini hanya menggunakan parameter rate, sebagai akibatnya seseorang hanya perlu mengeset kecepatan (rate) yang akan diberikan ke suatu kelas. Oleh karena itu HTB lebih mudah dan intuitif dibandingkan CBQ.

Pada HTB terdapat parameter ceil sehingga kelas akan selalu mendapatkan bandwidth di antara base rate dan nilai ceil rate-nya. Parameter ini dapat dianggap sebagai estimator kedua, sehingga setiap kelas dapat meminjam bandwidth selama bandwidth total yang diperoleh memiliki nilai di bawah nilai ceil. Hal ini mudah diimplementasikan dengan cara tidak mengijinkan proses peminjaman bandwidth pada saat kelas telah melampaui rate ini (keduanya leaves dan interior dapat memiliki ceil). Sebagai catatan, apabila nilai ceil sama dengan nilai base rate, maka akan memiliki fungsi yang sama seperti parameter bounded pada CBQ, di mana kelas-kelas tidak diijinkan untuk meminjam bandwidth. Sedangkan jika nilai ceil diset tak terbatas atau dengan nilai yang lebih tinggi seperti kecepatan link yang dimiliki, maka akan didapat fungsi yang sama seperti kelas non-bounded. Sebagai contoh, seseorang dapat menjamin bandwidth 1 Mbit untuk suatu kelas, dan mengijinkan penggunaan bandwidth sampai dengan 2 Mbit pada kelas tersebut apabila link dalam keadaan idle. Parameter ceil ini sangatlah berguna untuk ISP karena para ISP kemungkinan besar akan memakainya untuk membatasi jumlah servis yang akan diterima oleh suatu pelanggan walaupun pelanggan lain tidak melakukan permintaan servis, dengan kata lain kebanyakan ISP menginginkan pelanggan untuk membayar sejumlah uang lagi untuk memperoleh pelayanan yang lebih baik.

2.5. TOKEN BUCKET FILTER (TBF)

Token bucket merupakan suatu definisi formal dari suatu transfer rate. Antrian ini memiliki tiga buah komponen: ukuran burst, mean rate, dan interval waktu (Tc).

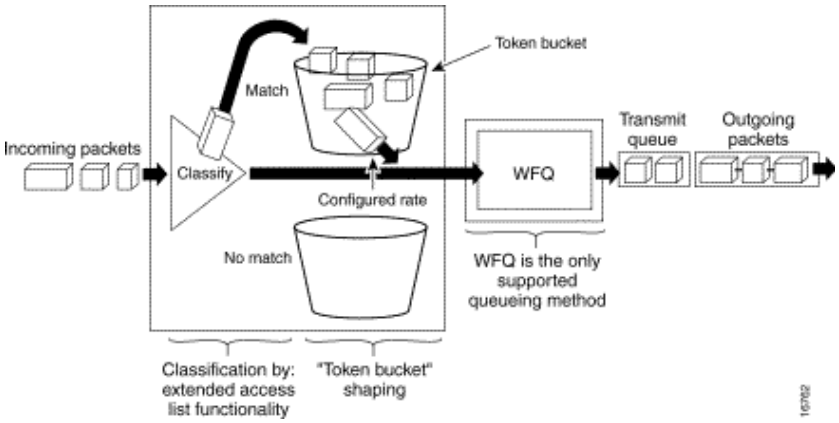
Implementasi TBF terdiri dari sebuah buffer (bucket), yang secara konstan diisi oleh beberapa informasi virtual yang dinamakan token, pada rate yang spesifik (token rate). Parameter paling penting dari bucket adalah ukurannya, yaitu banyaknya token yang dapat disimpan.

Setiap token yang masuk mengumpulkan satu paket yang datang dari antrian data dan kemudian dihapus dari bucket. Token bucket filter bisa dilihat pada gambar 6.

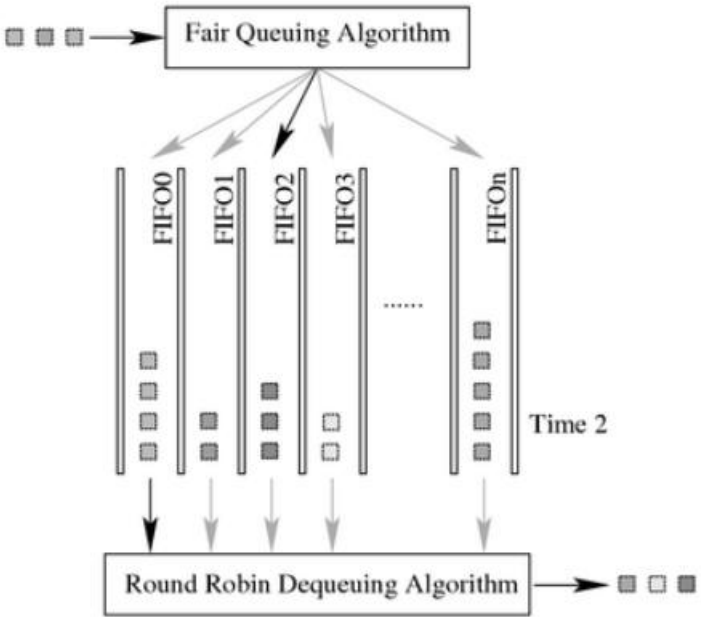
2.6. STOCHASTIC FAIRNESS QUEUEING (SFQ)

Stochastic Fairness Queueing (SFQ) merupakan suatu implementasi sederhana dari keluarga algoritma fair queueing. Perhatikan gambar 7. Ia kurang akurat

apabila dibandingkan dengan yang lain, tetapi ia juga membutuhkan perhitungan yang lebih sedikit dibandingkan dengan yang lain. Kata kunci dari SFQ adalah conversation (atau aliran), yang sangat berhubungan dengan sebuah sesi TCP atau sebuah aliran UDP. Traffik akan dibagi ke dalam beberapa jumlah besar antrian FIFO, satu untuk setiap conversation. Traffik kemudian dikirim secara round-robin, dengan memberikan kesempatan kepada tiap sesi untuk mengirimkan data pada gilirannya. SFQ disebut stochastic karena ia sebenarnya tidak menyediakan sebuah antrian untuk setiap sesi, ia memiliki suatu algoritma yang membagi traffik melalui sejumlah antrian yang terbatas dengan menggunakan algoritma hashing. Karena hash inilah, sesi yang banyak dapat berakhir di bucket yang sama, yang akan membagi dua tiap sesi kesempatan untuk mengirimkan paket, sehingga membagi dua kecepatan efektif yang tersedia. Untuk menghindari situasi ini menjadi terlihat, SFQ mengubah algoritma hashing yang ia miliki secara sering sehingga dua sesi yang bertabrakan akan terjadi dalam waktu yang singkat. SFQ hanya berguna jika interface outgoing yang dimiliki benar-benar penuh. Jika tidak, maka tidak akan ada queue pada mesin linux sehingga tidak akan ada efeknya.



Gambar 6. Token Bucket Filter



Gambar 7. Stochastic Fairness Queueing

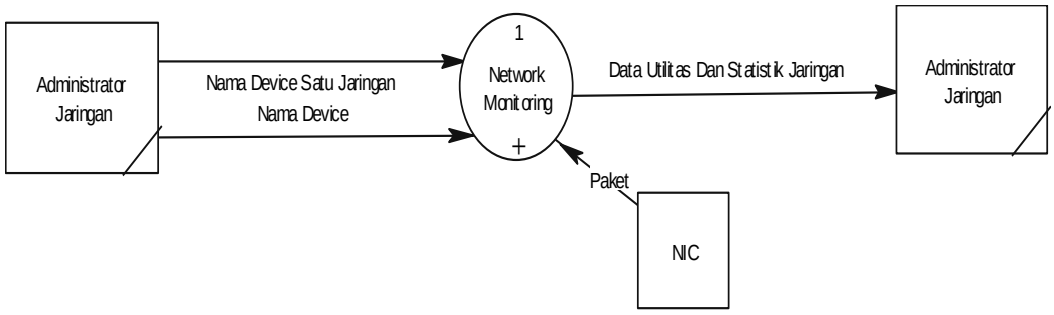
3. DESKRIPSI SISTEM

Aplikasi ini akan melakukan penangkapan terhadap paket yang masuk dan keluar dari interface jaringan. Tetapi tidak semua trafik paket yang melalui interface ditangkap, hanya paket yang berada pada protokol TCP/IP saja yang akan ditangkap dan diolah. Dari paket yang ditangkap tadi akan dihasilkan data session TCP-UDP-ICMP, data utilitas jaringan saat ini, data statistik utilitas jaringan per-jam dan data statistik utilitas jaringan per-hari. Masing-masing data akan disimpan dengan format XML dan disimpan dengan nama file khusus serta pada direktori tertentu untuk memudahkan pembacaan.

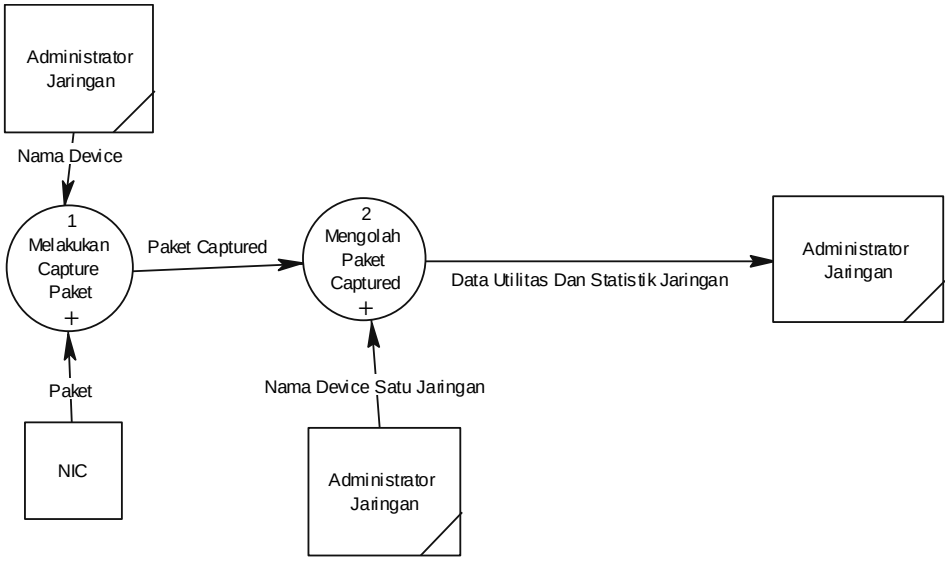
Aplikasi Server Network Monitoring ini memiliki kemampuan untuk melakukan hal-hal sebagai berikut:

- Melakukan penangkapan paket yang masuk ataupun keluar pada interface jaringan yang sudah ditentukan. Paket yang ditangkap tidak akan diubah, hanya akan dilakukan pengecekan pada header TCP/IP untuk pemrosesan selanjutnya. Selain itu hanya paket yang berada pada protokol TCP/IP saja yang akan ditangkap.
- Menghasilkan data session TCP, UDP, dan ICMP yang sedang terjadi saat ini. Selain itu juga bisa menghasilkan data utilitas saat ini dan statistik jaringan yang dikelompokkan berdasarkan jam dan hari. Termasuk didalam utilitas dan statistik jaringan yaitu jumlah byte dan paket yang keluar dan masuk pada interface, jumlah byte dan paket yang keluar dan masuk pada IP yang berada dalam satu jaringan serta jumlah byte dan paket yang keluar dan masuk yang dikelompokkan berdasarkan port server.

Proses Network Monitoring diturunkan menjadi dua sub-proses pada level 1, yaitu Melakukan Capture Paket dan Mengolah Paket Captured. Untuk lebih jelasnya dapat dilihat pada gambar 9.



Gambar 8. DAD Level 0: Network Monitoring



Gambar 9. DAD Level 1: Network Monitoring

4. UJI COBA

Uji coba dilakukan dengan memakai tiga skenario utama, yaitu:

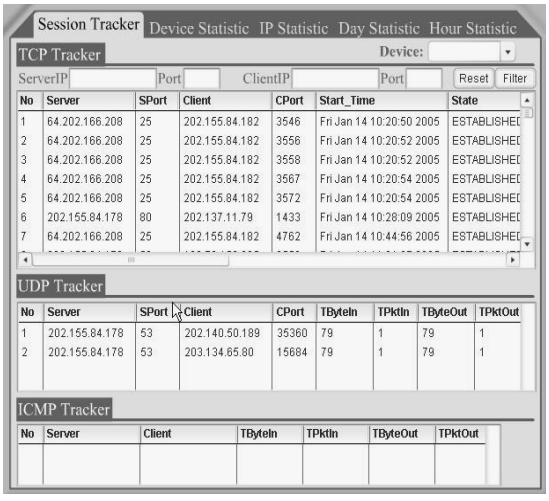
- Uji coba aplikasi server Network Monitoring
- Uji coba aplikasi client Network Monitoring
- Uji coba aplikasi HTB GUI

Dari uji coba aplikasi server Network Monitoring dihasilkan file xml untuk session, utilitas dan statistik jaringan. Pada gambar 10 merupakan file xml session TCP.

```
<?XML version="1.0"?>
<TCP_PACKETS>
<PACKET NUMBER="1" SERVER="64.202.166.208" SERVER_PORT="25"
CLIENT="202.155.84.182" CLIENT_PORT="3546" START_TIME="Fri Jan
14 10:20:50 2005" STATE="ESTABLISHED" IDLE="286078"
TOTALBYTES_IN="60" TOTALPACKETS_IN="1" TOTALBYTES_OUT="452"
TOTALPACKETS_OUT="8" CURRENTBYTES_IN="0" CURRENTPACKETS_IN="0"
CURRENTBYTES_OUT="0" CURRENTPACKETS_OUT="0"
AVERAGEBYTES_IN="0.000000" AVERAGEPACKETS_IN="0.000000"
AVERAGEBYTES_OUT="0.000000" AVERAGEPACKETS_OUT="0.000000" />
. . . . . → sampai dengan session TCP terakhir
<PACKET NUMBER="904" SERVER="202.155.84.188" SERVER_PORT="445"
CLIENT="202.155.148.217" CLIENT_PORT="1688" START_TIME="Mon
Jan 17 18:00:34 2005" STATE="SYN_SENT" IDLE="1"
TOTALBYTES_IN="66" TOTALPACKETS_IN="1" TOTALBYTES_OUT="0"
TOTALPACKETS_OUT="0" CURRENTBYTES_IN="66"
CURRENTPACKETS_IN="1" CURRENTBYTES_OUT="0"
CURRENTPACKETS_OUT="0" AVERAGEBYTES_IN="66.000000"
AVERAGEPACKETS_IN="1.000000" AVERAGEBYTES_OUT="0.000000"
AVERAGEPACKETS_OUT="0.000000" />
</TCP_PACKETS>
```

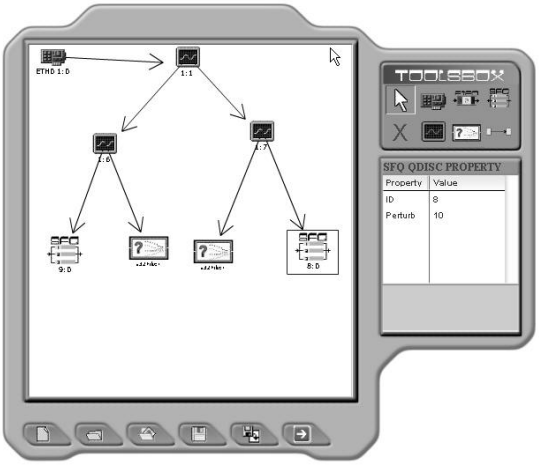
Gambar 10. Data xml session TCP

Dari uji coba aplikasi Client Network Monitoring dilakukan dengan menampilkan data xml yang dihasilkan oleh aplikasi server melalui antarmuka yang sudah dibuat. Gambar 11 merupakan antarmuka untuk menampilkan data session TCP.



Gambar 11. Antarmuka session TCP

Uji Coba Aplikasi HTB GUI dilakukan dengan membuat model HTB melalui aplikasi ini. Gambar 12 merupakan model HTB yang sudah dibuat.



Gambar 12. Rancangan Model HTB

Setelah dilakukan eksekusi konfigurasi model HTB tersebut pada Linux, diperoleh hasil yaitu konfigurasi *Traffic Control* HTB di Linux berubah sesuai dengan konfigurasi yang dirancang melalui aplikasi ini. Gambar 13 dan 14 menunjukkan konfigurasi baru hasil eksekusi perintah tc oleh aplikasi.

```
[root@stallion Files]# /sbin/tc qdisc show  dev eth0
qdisc sfq 3: limit 128p quantum 1514b perturb 15sec
qdisc sfq 2: limit 128p quantum 1514b perturb 5sec
qdisc htb 1: r2q 10 default 3 direct_packets_stat 0
```

Gambar 13. Hasil eksekusi qdisc di Linux

```
[root@stallion Files]# /sbin/tc class show  dev eth0
class htb 1:1 root rate 128Kbit ceil 128Kbit burst 1762b
cburst 1762b
class htb 1:2 parent 1:1 leaf 2: prio 0 rate 96Kbit ceil
128Kbit burst 1721b cburst 1762b
class htb 1:3 parent 1:1 leaf 3: prio 0 rate 32Kbit ceil
128Kbit burst 1639b cburst 1762b
```

Gambar 14. Hasil eksekusi class di Linux

5. KESIMPULAN

Dari hasil uji coba dan evaluasi perangkat lunak yang telah dilakukan, didapatkan beberapa kesimpulan sebagai berikut:

- Aplikasi HTB GUI berhasil membuat dan menjalankan script perintah TC pada mesin Linux dari model HTB yang dibuat oleh pengguna.
- Aplikasi server Network Monitoring berhasil menghasilkan data utilitas dan statistik jaringan dari trafik paket. Data yang dihasilkan berupa file XML.

- Aplikasi client Network Monitoring berhasil menampilkan data yang dihasilkan dari aplikasi server Network Monitoring melalui antarmuka web.

6. DAFTAR PUSTAKA

1. Bert Hubert, "Linux Advance Routing & Traffic Control HOWTO", Netherland BV, 2003.
2. Martin A. Brown, "Traffic Control HOW TO", SecurePipe Inc., 2003.
3. Martin A. Brown, "Traffic Control using tcng and HTB HOWTO", SecurePipe Inc., 2003.
4. RFC:793, "Transmission Control Protocol Darpa Internet Program Protocol Specification", Information Sciences Institute University of Southern California, September 1981.
5. TCPDUMP, "Programming with pcap", www.tcpdump.org, 2004.
6. O'Reilly, "Action Script Cookbook", O'Reilly Inc., 2003.
7. Wiley & Sons, "Macromedia Flash MX 2004 Action Script Bible", Wiley & Sons, 2004.