MATICS : Jurnal Ilmu Komputer dan Teknologi Informasi ISSN : 1978-161X(p); 2477-2550(e)
Volume 8, No. 2 (2016), pp 67-72 DOI : 10.18860/mat.v8i2.3558
Received : July 29th 2016; **Accepted** : September 15th 2016 ; **Avalaible Online** : November 20th 2016

Implementasi Manajemen *Bandwidth* Dengan Disiplin Antrian *Hierarchical Token Bucket* (HTB) Pada Sistem Operasi Linux

Muhammad Nugraha, Shoffin Nahwa Utama

***Abstract*—Bandwidth is data transfer rate between client and server on the networking which is usually expressed in bits per second (bps). The most important problem on the Internet is domination resource and bandwidth by some user, meanwhile other user did not get resource and bandwidth properly. To solve that problem we need to implement traffic control and bandwidth management system in router. In this research the author will implement Hierarchical Token Bucket algorithm as a queue discipline (qdsic) to produce bandwidth management properly. The final result on this research is management bandwidth cheaply and efficiently by using Hierarchical Token Bucket qdisc on router with linux operating system.**

Index Terms—*traffic control, Hierarchical Token Bucket, bandwidth management, linux.*

Abstrak—Bandwidth adalah nilai hitung atau perhitungan konsumsi transfer data telekomunikasi yang dihitung dalam satuan bit per detik atau yang biasa disingkat bps yang terjadi antara komputer server dan komputer client dalam waktu tertentu dalam sebuah jaringan komputer. Salah satu masalah terpenting dari jaringan internet adalah pendominasian sumberdaya jaringan dan bandwidth oleh sejumlah pengguna tertentu sementara pengguna yang lain tidak dapat memperoleh bandwidth yang semestinya mereka dapatkan. Untuk mengatasi hal tersebut perlu diterapkannya traffic control dan sistem pengaturan bandwidth pada router. Pada penelitian ini penulis ingin menerapkan algoritma Hierarchical Token Bucket sebagai disiplin antriannya untuk mendapatkan pengaturan bandwidth yang akurat, sehinga pengguna layanan dapat mendapatkan bandwidth sebagaimana mestinya. Hasil akhir dari penelitian ini adalah terbentuknya suatu manajemen bandwidth yang murah dan efisien dengan menggunakan disiplin antrian Hierarchical Token Bucket pada Sistem Operasi Linux yang mampu mengatur penggunaan bandwidth sesuai dengan diinginkan.

Manuscript received Juli 29, 2016. This work was supported in part by Informatics Engineering Department of Darusalam Gontor University Ponorogo, Jawa Timur.

Muhammad Nugraha is with the Infornatics Engineering Departement of Darusalam Gontor Univeristy , Ponorogo, Jawa Timur; email mnugraha@unida.gontor.ac..id)

Shoffin Nahwa Utama, was with the Infornatics Engineering Departement of Darusalam Gontor Univeristy , Ponorogo, Jawa Timur; email shoffin@unida.gontor.ac..id)

Kata Kunci—*traffic control, Hierarchical Token Bucket, manajemen bandwidth, linux.*

I. PENDAHULUAN

Layanan komunikasi data telah menjadi sangat penting dalam kehidupan sehari-hari. Hampir disetiap bidang kehidupan telah menggunakan layanan ini. Layanan inipun tidak hanya digunakan secara individual tetapi juga digunakan oleh banyak pengguna. Banyak sekali organisasi atau lembaga yang menggunakan akses Internetnya dengan banyak pengguna seperti lembaga pendidikan, perkantoran, usaha Internet, dll. Penggunaan akses Internet dengan banyak pengguna akan mengakibatkan turunnya *performance* jaringan dimana sebagian pengguna akan mendominasi pemakaian *bandwidth*sedangkan sebagian yang lain sulit untuk mendapatkan*bandwidth*yang semestinya diperoleh. Pengaturan *bandwidth*perlu diterapkan sedemikian rupa sehingga setiap pengguna bisa mendapatkan *bandwidth*dengan semestinya sesuai dengan rancangan yang dikehendaki. Hal ini akan lebih penting lagi jika di lingkungan usaha atau komersial, dimana penyedia layanan harus memastikan bahwa pelanggan mendapatkan layanan sesuai dengan konvensasi yang diberikan.

Seperti yang kita tahu banyak sekali perangkat khusus pengatur *traffic*atau *bandwidth* komersial yang beredar di pasaran seperti Cisco dan Mikrotik. Kemampuan perangkat khusus tersebut dari segi *performance* sudah teruji dan berjalan baik, akantetapi perangkat khusus tersebut dari segi harga cukup mahal dan hal ini akan cukup memberatkan bagi lembaga yang tidak memiliki dana yang cukup besar.

Quality of Service (QoS) memegang peranan yang sangat penting dalam hal ini. Linux sebagai suatu sistem operasi yang bersifat *open* dan *free*, telah menawarkan berbagai teknik QoS untuk memfasilitasi proses manajemen *bandwidth* pada suatu jaringan. Salah satunya adalah dengan menggunakan disiplin antrian *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 diantara para pengguna jaringan sehingga *performance*jaringan tetap dapat terjaga [4] [1].

Pada saat ini sudah ada beberapa penelitian yang sudah dilakukan mengenai manajemen *bandwidth*, diantaranya adalah penelitian yang dilakukan oleh [7] dengan judul penelitian"*Perencanaan dan Implementasi Manajemen Proses Jaringan pada Warnet*", yang pada penelitiannya itu perancangannya menekankan pada LAN warnet, sehingga *traffic* yang dimonitoring ataupun dianalisa adalah *traffic* yang terjadi pada *workstation* atau yang melewati gateway dari jaringan warnet. Dan dari hasil penelitiannya itu didapat gambaran bahwa saat terjadi dominasi penggunaan *bandwidth* oleh workstation tertentu dalam suatu jaringan, hal ini akan mempengaruhi kecepatan dan kelancaran akses Internet dari *workstation* lain. Sehingga diharapkan nantinya manajemen proses jaringan pada warnet dapat lebih dikembangkan pada tataran yang lebih yaitu pada proses *network remote monitoring*. Penelitian lain yang sudah dilakukan adalah "*Implementasi Differentiated Service (Diffserv) DiJaringan Testbed Menggunakan Disiplin Antrian Priority Queuing (PQ) dan Hierarchy Token Bucket (HTB)*" oleh [6], yang mengimplementasikan *Differentiated Service* (DiffServ) pada jaringan *testbed*, mengukur dan menganalisis parameter-parameter kualitas layanan jaringan berbasis *DiffServ*, serta mencari disiplin antrian yang mampu memberikan layanan paling sesuai dengan kebutuhan kelas *trafficExpedited Forwarding* (EF) dan *Assured Forwarding* (AF). Hasil pengujiannya memperlihatkan bahwa *DiffServ* mampu mengklasifikasikan *traffic*ke dalam kelas-kelas yang sesuai dan memberikan pelayanan yang berbeda diantara kelas-kelas *traffic*tersebut. Hasil pengujian juga menunjukkan bahwa pemberian prioritas pertama pada *traffic*menggunakan disiplin antrian PQ menghasilkan kinerja layanan yang mampu memenuhi kebutuhan kelas *traffic*EF, yaitu waktu tunda, jitter, dan paket hilang yang kecil. Sedangkan penerapan disiplin antrian HTB memberikan kinerja layanan yang paling sesuai untuk memenuhi kebutuhan kelas *traffic*AF, yaitu pembagian dan penjaminan kapasitas jaringan untuk masing-masing kelas.

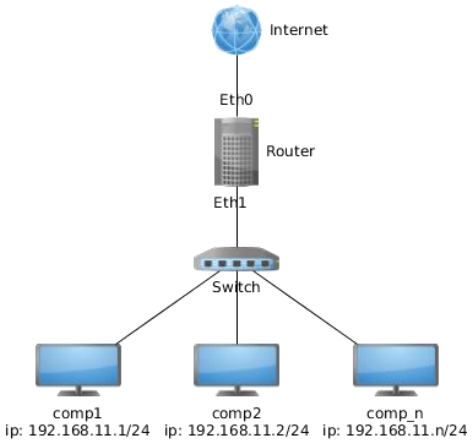
Berdasarkan dari beberapa penelitian sebelumnya mengenai manajemen *bandwidth* maka pada penelitian ini akan dibangun sistem yang lebih mudah tetapi memberikan hasil yang optimal. Sistem yang akan dibangun berupa sebuah manajemen *bandwidth* dengan disiplin antrian menggunakan *Hierarchical Token Buket* (HTB) dengan sistem operasi Linux dan tool konfigurasinya menggunakan *iproute2 suite* dan *netfilter iptables*. Titik beratnya penelitian ini mengarah kepada pengaturan kapasitas *bandwidth* bagi komputer *client*, dengan kriteria-kriteria *filter* dan prioritas berdasarkan alamat IP asal, alamat IP tujuan, *port* asal dan *port* tujuan (*port* TCP / *port* UDP).

II. METODE PENELITIAN

Pada penelitian ini akan dilakukan percobaan menggunakan 1 *PC*dengan sistem operasi linux sebagai router yang berfungsi untuk memanage *bandwidth*, memonitor dan mengcapture *traffic*, dan untuk memblock service-service yang tidak diinginkan. Selain itu untuk mendapatkan data yang diinginkandibutuhkan 30 *PC*yang nantinya akan di setting sedemikian rupa sehingga bisa mendapatkan hasil data yang akurat dan bisa dipertanggungjawabkan. Sistem yang akan di bangun ini tujuannya adalah untuk menjamin dan meastikan bahwa setiap *client yang menggunakan* sumber daya jaringan akan mendapatkan jumlah *traffic* baik yang lokal maupun dari Internet sebagai mana mestinya. Selain setiap *client* akan dijamin medapatkan *bandwidth* sesuai yang dialokasikan, sistem juga akan di setting bisa mendapatkan *bandwidth* lebih dari yang dialokasikan dengan syarat*bandwidth* yang tersedia dari sumber/pusat ada yang sedang tidak digunakan. Dengan kata lain *bandwidth* yang tidak di pakai tersebut di pinjam oleh *client* yang sedang membutuhkan *bandwidth* lebih besar dan *bandwidth* akan kembali seperti semula atau berkurang ketika ada*client* lain yang akan memakai *bandwidth* juga. Adapun untuk tools yang digunakan pada penelitian ini antara lain: sistem operasi Linux, disiplin antrian HTB, SFQ, dan FIFO, *iproute2 suite*, *netfilter iptables*, *cacti*, PHP-5, *MySQL*, *apache*, *rrdtool*, *snmp*, *iftop*.

Karena keterbatasan kemampuan infrastruktur, sistem yang akan dibangun ini harus dapat dijalankan dengan baik menggunakan perangkat keras maupun perangkat lunak yang relatif murah dibandingkan bila menggunakan perangkat khusus pengatur *traffic* komersial. Oleh karean hal tersebut, desain untuk penerapan sistem terdiri dari butir-butir berikut ini:

1. Sistem pengaturan dijalankan pada *router* yang dibangun dengan menggunakan perangkat keras komputer biasa, ditambah dengan sejumlah kartu jaringan (*network interface*), dan Sistem Operasi GNU/Linux. Disiplin antrian dan pengatur *traffic* utama adalah menggunakan *Hierarchical Token Bucket* (HTB), yang sudah termasuk di dalam kernel Linux.
2. Disiplin antrian untuk pengaturan dikaitkan pada *network interface* yang diperlukan, dimana paket keluar dari *interface* tersebut. Seperti rancangan sistem pada Gambar 1, disiplin antrian HTB yang bertugas untuk mengatur alokasi *upload* dari komputer *client* di jalankan di *interface eth0* dan untuk *downloadnya* dijalankan dari *interface eth1*.



Gambar 1. Rancangan sistem manajemen bandwidth

3. Pengaturan dibuat untuk memastikan dan menjamin bahwa komputer *client* mendapatkan

sesuai dengan kelasnya sendiri. Dengan disiplin antrian HTB, kelas penggunaan dibuat dengan nilai parameter *rate* = alokasi minimum dan *ceil* = alokasi maksimum. Untuk kelas penggunaan penuh, nilai *rate* = nilai *ceil*. Pada penelitian ini akan dilakukan pembagian untuk 30 *client* dengan pembagian seperti pada Tabel 1 di bawah ini dengan alokasi total *bandwidth* yang disediakan adalah 512 kbit/s:

Tabel 1. Rencana implementasi pembagian bandwidth

No	PC Client	Download (Kbps)		Upload (Kbps)	
		Rate	Ceil	Rate	Ceil
1	192.168.11.1	8	128	8	128
2	192.168.11.2	8	128	8	128
3	192.168.11.3	8	128	8	128
4	192.168.11.4	8	128	8	128
5	192.168.11.5	8	128	8	128
6	192.168.11.6	8	128	8	128
7	192.168.11.7	8	128	8	128
8	192.168.11.8	8	128	8	128
9	192.168.11.9	8	128	8	128
10	192.168.11.10	8	128	8	128
11	192.168.11.11	8	128	8	128
12	192.168.11.12	8	128	8	128
13	192.168.11.13	8	128	8	128
14	192.168.11.14	8	128	8	128
15	192.168.11.15	8	128	8	128
16	192.168.11.16	8	128	8	128
17	192.168.11.17	8	128	8	128
18	192.168.11.18	8	128	8	128
19	192.168.11.19	8	128	8	128
20	192.168.11.20	8	128	8	128
21	192.168.11.21	8	128	8	128
22	192.168.11.22	8	128	8	128
23	192.168.11.23	8	128	8	128
24	192.168.11.24	8	128	8	128
25	192.168.11.25	8	128	8	128
26	192.168.11.26	8	128	8	128
27	192.168.11.27	8	128	8	128
28	192.168.11.28	8	128	8	128
29	192.168.11.29	8	128	8	128
30	192.168.11.30	8	128	8	128

4. Membuat filter dengan kriteria berdasarkan alamat IP asal paket data, IP tujuan, *port* asal, *port* tujuan atau nilai lain dalam *header* paket *tcp/ip*. Sebagai contoh, dengan program *tc* di Linux, alamat IP asal dan alamat IP tujuan dibuat sebagai nilai untuk parameter *ip source* atau *ip destination* tergantung dari arah *traffic*, untuk kemudian diberikan kepada kelas HTB yang sesuai dengan alokasi nya.

Setelah tahap rancangan diimplementasikan maka diperlukan pengujian sistem untuk mengetahui apakah sistem berjalan dengan semestinya atau tidak. Metode pengujian sistem dilakukan dengan membandingkan antara implementasi rancangan dengan hasil aktualnya. Untuk itu perlu sistem monitoring terhadap jalannya manajemen *bandwidth* yang dengan monitoring ini akan diperoleh data-data dan informasi aktual atas jalannya sistem manajemen bandwidth.

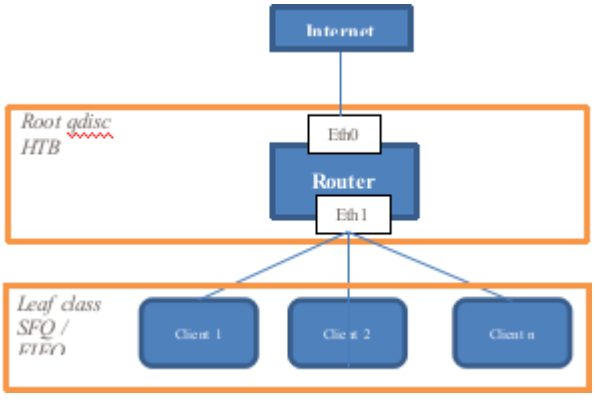
Program *tc* bisa digunakan untuk menampilkan berbagai statistik dari setiap disiplin antrian, kelas dan filter yang diterapkan, termasuk jumlah paket data yang masuk ke dalam satu kelas tertentu. Dengan membuat sebuah program/*script* sederhana, nilai jumlah paket data ini bisa diambil secara periodik dan kemudian diberikan pada sebuah aplikasi berbasis web yang bisa digunakan untuk menampilkan grafik pemakaian dalam jaringan seperti *Cacti* (*cacti.net*) atau *MRTG* (*oss.oetiker.ch/mrtg/*). Aplikasi-aplikasi ini memiliki kemampuan untuk menangani sejumlah sumber data (*data sources*) yang mengikuti format tertentu dan menampilkan datanya sebagai sebuah grafik dengan berbagai cara. Yang perlu dilakukan hanya menyiapkan sumber datanya dengan format yang dapat diproses oleh aplikasi tersebut. Untuk selanjutnya, implementasi dan pembahasan detil hanya sampai pada pengambilan data dari statistik HTB dan membuat hasilnya menjadi sumber data bagi aplikasi penampil grafik.

Oleh karena itu dari hasil monitoring yang dilakukan dapat dilihat dari data yang didapat untuk membandingkan antara rancangan dengan hasil monitoringnya, apakah rancangan yang dibangun sesuai dengan hasil pada aktualnya atau tidak. Ketika hasil monitoring sudah mencerminkan kesesuaian dengan rancangan implementasi maka dapat dikatakan bahwa sistem berjalan sesuai dengan semestinya.

III. HASIL DAN PEMBAHASAN

3. 1 Implementasi Sistem

Sistem pengaturan *bandwidth* dengan HTB di router meliputi pembuatan displin antrian utama HTB, *class-class* HTB, disiplin antrian (SFQ) untuk setiap *class* HTB, dan *filter-filter* untuk dikaitkan dengan *class-class* tersebut.



Gambar 2 Rancangan implementasi disiplin antrian

3. 1. 1. Disiplin Antrian HTB (HTB qdisc)

Pengaturan *bandwidth* dengan HTB dibangun pertama-tama dengan membuat sebuah *root qdisc* menggunakan program *tc*, yang akan menjadi disiplin antrian utama. Dengan demikian, root *qdisc* di *router*dibuat untuk 2 buah *interface* jaringan (*eth0* dan *eth1*) dengan perintah-perintah seperti berikut:

```
tc qdisc add dev eth0 root handle 1: htb
default 30

tc qdisc add dev eth1 root handle 1: htb
default 30
```

Kedua perintah diatas menghasilkan 2 HTB *qdisc* yang dikaitkan pada *interface* eth0 dan eth1, dengan ID untuk class default adalah 30. Kedua *qdisc* tersebut akan menjadi dasar pengaturan untuk *trafficupload* (keluar dari *eth0*) dan *trafficdownload* (keluar dari *eth1*) seperti pada Gambar 2 di atas.

3. 1. 2. Kelas-kelas HTB (HTB class)

Class-class HTB dibangun dibawah *root qdisc* yang telah dibuat sebelumnya.Di bawah root *class* ini dibuat lagi sejumlah *class* yang lain sesuai dengan hirarki yang dikehendaki. pada masing-masing interface dibuat satu root class dengan nilai rate dan ceil yang besar, sehingga root class ini seakan-akan adalah perangkat keras interface jaringan itu sendiri. Kemudian untuk class-class yang ada di bawah root class ada 3 class, antara lain : class Internet, class local, class default. Di bawah class internet juga terdiri lagi leaf class yang terdiri dari n leaf class dimana n disini terdiri dari 30 dan ke 30 class ini merupakan class yang dimiliki oleh komputer-komputer client.

Setiap *leaf class* atau *class* ujung yang tidak mempunyai anak *class* dibawahnya harus mempunyai sebuah *classless qdisc* yang secara langsung akan menangani proses antrian paket-paket data. *Qdisc* yang digunakan disini adalah SFQ. Bila untuk sebuah *leaf class* tidak ditentukan *qdisc* secara spesifik seperti di atas, maka kernel akan memberikan FIFO sebagai *default qdisc*. SFQ dipilih karena mempunyai kinerja yang baik dan adil, dan cocok untuk digunakan pada *link-sharing* dengan *traffic* yang cenderung penuh.

3. 2. Monitoring Sistem

Monitoring terhadap sistem manajemen *bandwidth* bertujuan untuk mengetahui dan memastikan bahwa sistem ini berjalan sesuai dengan yang dikehendaki. Monitoring dilakukan dengan mengambil data-data kuantitatif dari statistik pengaturan HTB menggunakan program *tc*. Hasil yang diperoleh kemudian ditampikan dalam bentuk gambar grafik menggunakan program Cacti.

Dengan kemampuan *Cacti* untuk menjalankan sebuah perintah atau *script* sebagai metode pengambilan datanya, maka *script*yang diberikan pada *Cacti* kemudian dieksekusi dari dalam proses aplikasi *Cacti* tersebut. Untuk memudahkan dalam mengelola dan menampilkan informasi secara menyeluruh, data dari *router* diambil oleh *Cacti* dengan bantuan paket program *snmp* dan *snmp-server*.Proses *snmp-server*memiliki fasilitas *extend* dan *exec* yang dapat digunakan untuk mengeksekusi perintah/*script* tertentu dalam sistem, dan memasukkan outputnya ke dalam struktur *Management Information Base* dengan SNMP *ObjectID* yang bisa dipilih atau dikehendaki. Dengan demikian, data *monitoring* bisa diambil dengan melakukan *query* atas SNMP *ObjectID* yang sudah ditentukan sebelumnya. Dalam hal ini, SNMP Server akan diminta untuk mengeksekusi *script*yang kita tentukan, setiap kali ada SNMP *query* dari *Cacti* , atas SNMP ObjectiD yang berisi data statistik *class-class* HTB di atas.

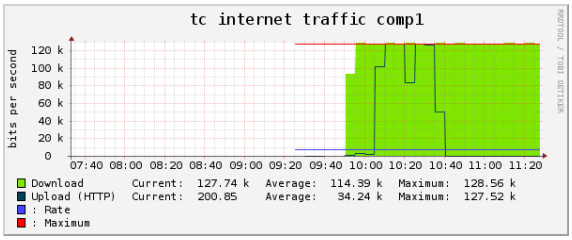
Cacti akan terus menjalankan background proses di system dan akan melakukan *query* untuk mengambil data secara periodik dengan defaultnya yaitu setiap 300 detik. Dengan demikian akan diperoleh grafik *monitoringtraffic* yang masuk ke dalam *class-class* pengaturan HTB yang sedang berjalan.

Dalam proses *monitoring* yang dilakukan untuk analisis disini digunakan sample 6 PC client yang masing-masing diberikan alokasi rate 8 kbit/s dan ceil 128 kbit/s untuk *traffic* internet. 6 PC ini mempunyai satu *class* induk yaitu *classinet* dengan alokasi bandwidth 512 kbit/s. Pengamatan dilakukan selama rentang waktu sekitar 2 jam dengan setiap PC client mendownload file yang cukup besar dari sumber yang koneksinya cukup bagus dan lancar. Skenario pengambilan datanya adalah pertama-tama penggunaan *traffic* hanya dilakukan oleh comp1 dan comp2 sekitar 30 menit, kemudian penggunaan *traffic* ditambah dengan comp3. Setelah itu, 15 menit kemudian ditambah lagi dengan comp4. Kemudian setelah sekitar 45 menit, pemakaian ditambah dengan comp5 dan comp6 secara bersamaan selama 30 menit.

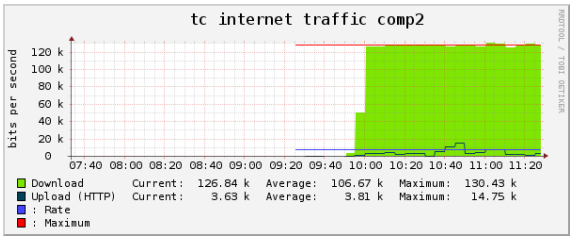
3. 3 Analisis Sistem

Dari hasil monitoring yang sudah dilakukan grafiknya bisa dilihat dari Gambar 3 berikut ini. Pada interval waktu 30 menit pertama comp1 dan comp2 masing-masing dapat menggunakan bandwidth sampai dengan 128 kbit/s meskipun alokasi rate untuk comp1 dan comp2 hanya 8 kbit/s. hal ini terjadi karena class induknya yang memiliki bandwidth 512kbit/s masih memiliki sisa bandwidth dan dapat dipinjamkan kepada comp1 dan comp2. Dan pada grafik traffic inet selama sekitar 30 menit pertama class ini menggunakan traffic 256 kbit/s yang mana traffic tersebut adalah agregat dari 2 class anaknya yaitu comp1 dan comp2.

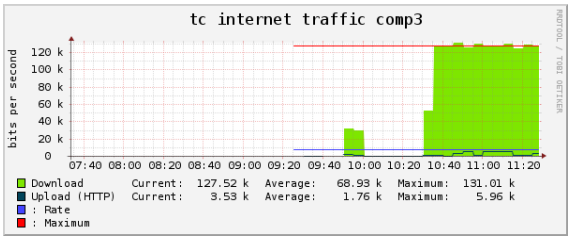
comp1 berjalan pada jam 09.50



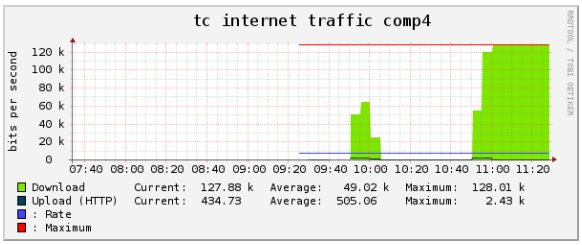
comp2 berjalan pada jam 09.50



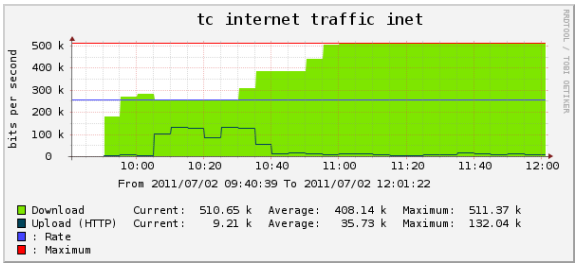
comp3 berjalan pada jam 10.20



comp4 berjalan pada jam 10.50



Trafick ke Internet

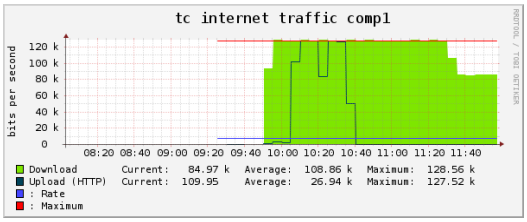


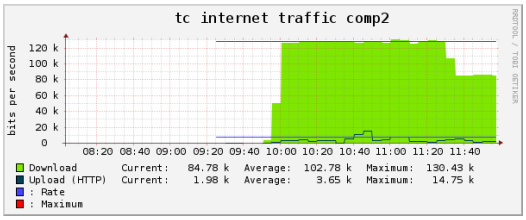
Gambar 3 *Monitoring* grafik *comp1*, comp2, comp3, comp4, dan inet awal

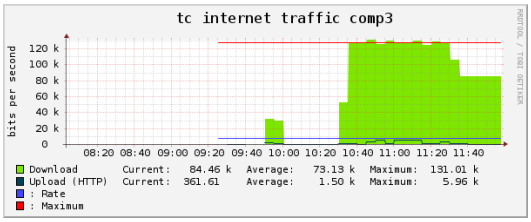
Kemudian pada interval waktu berikutnya comp3 mulai menggunakan traffic dan comp3 ini bisa menggunakan bandwidth samapai dengan ceil classnya yaitu 128 kbit/s karena class induknya yaitu class inet masih memiliki alokasi bandwidth sampai dengan 512 kbit/s. Dan bisa dilihat juga pada grafik inet saat interval waktu setelah comp3 dinyalakan bahwa class inet mengalami kenaikan penggunaan bandwidth menggunakan bandwidth sekitar 384 kbit/s yang mana traffic tersebut adalah agregat dari 3 class anaknya yaitu comp1, comp2, dan comp3.

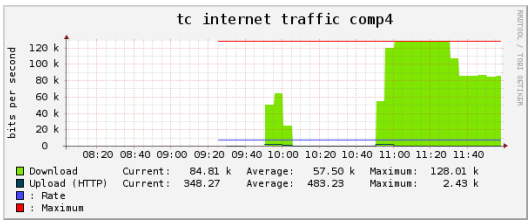
Pada interval waktu berikutnya comp4 mulai menggunakan traffic dan comp4 pun masih bisa memperoleh bandwidth sampai dengan ceilnya yaitu 128 kbit/s. Seperti pada interval sebelumnya, comp1, comp2, comp3, dan comp4 masih bisa menikmati bandwidth sampai dengan ceilnya 128 kbit/s karena class induknya yaitu class inet memiliki alokasi sampai dengan 512 kbit/s.

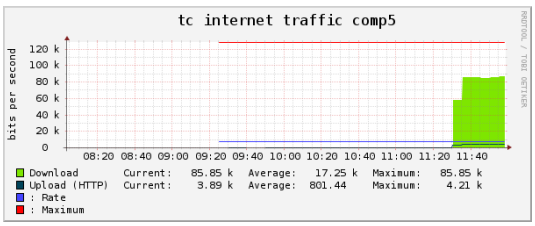
Pada interval waktu yang terakhir ini comp5 dan comp6 mulai menggunakan traffic internet sehingga ada 6 komputer yang menggunakan traffic. Pada interval ini semua komputer mengalami penurunan pemakaian bandwidth menjadi sekitar 85 kbit/s. Dapat dilihat pada Gambar 4 bahwa comp5 dan comp6 mendapatkan bandwidth dikisaran 85 kbit/s selain itu terjadi penurunan bandwidth juga pada comp1, comp2, comp3, dan comp4 yang dari awalnya mendapatkan bandwidth sekitar 128 kbit/s menjadi 85 kbit/s. hal ini terjadi karena alokasi bandwidth class induknya yaitu class inet yang sebelumnya digunakan oleh comp1, comp2, comp3, dan comp4 harus dibagikan sebagian ke comp5 dan comp6 karena sebenarnya comp1, comp2, comp3, dan comp4 sudah memperoleh bandwidth sesuai dengan ratenya yaitu 8 kbit/s dan kelebihannya adalah pinjaman dari class induknya yang sedang tidak terpakai.

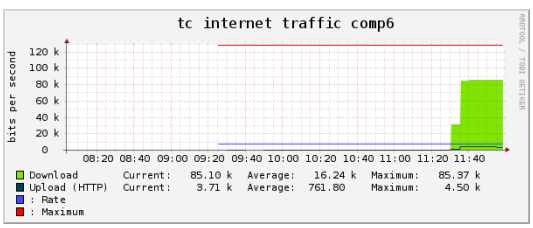


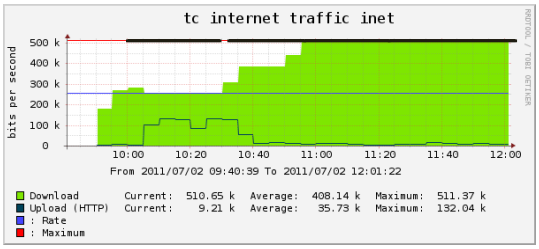












Gambar 4 Monitoring grafik 6 komputer client menggunakan bandwidth

Keadaan seperti di atas tersebut bandwidth yang tersedia dipinjamkan untuk 6 komputer. Setelah semua class anaknya (komputer 1 sampai 6) mendapatkan alokasi rate masing-masing (8 kbit/s) baru kemudian sisa alokasi bandwidthnya class induk dapat dipinjamkan / dibagikan kepada class-class anaknya. Dengan demikian, dapat dilihat bahwa setiap class pengaturan HTB untuk comp1, comp2, comp3, comp4, comp5, dan comp6 akan mendapatkan bandwidth sebesar alokasi rate yang ditentukan yaitu 8 kbit/s dan bisa memperoleh bandwidth lebih sampai dengan batas ceilnya yang ditentukan apabila class induknya masih memiliki sisa alokasi setelah semua rate class anaknya terpenuhi.

IV. KESIMPULAN

Implementasi manajemen bandwidth dengan disiplin antrian Hierarchical Token Bucket pada sistem operasi Linux dapat terealisasi sesuai dengan yang dirancang. Hasil monitoring traffic bandwith dapat membuktikan bahwa pengaturan bandwidth yang dibuat sudah sesuai dengan rancangan, yaitu pembatasan rate dan ceilnya sudah sesuai dengan pengaturan yang diatur oleh admin. Sehingga dengan terealisasinya hal tersebut tidak ada user atau client yang mendominasi dalam penggunaan bandwidth.

REFERENCES

[1] Devera, Martin; Cohen, Don., 2002. Hierarchical Token Bucket – Linux Queueing Discipline, http://luxik.cdi.cz/~devik/qos/htb/ , di akses pada tanggal 23 maret 2015.

[2] Figgins, S., Siever, E., Weber, A. 2003. Linux in a Nutshell: O'Reilly & Associates, Inc.

[3] Floyd, S., Jacobson, V. 1995. Link-sharing and Resource Management Models for Packet Networks. IEEE/ACM Transactions on Networking, Vol. 3 No. 4.

[4] Gabriel, D. & Potorac, D.A. 2009. Linux HTB queuing discipline implementations. First International Conference on Networked Digital Technologies, pp. 122 - 126 .

[5] Ivancic, D., Hadjina, N., Basch, D. 2005. Analysis of precision of the HTB packet scheduler. Applied Electromagnetics and Communications, 2005.

[6] Pramudita D.A., 2008. Implementasi Differentiated Service (Diffserv) Di Jaringan Testbed Menggunakan Disiplin Antrian Priority Queuing (PQ) dan Hierarchy Token Bucket (HTB). Skripsi Program Studi Teknik Elektro Institut Teknologi Bandung.

[7] Purwaka, I.A., 2002. Perencanaan dan Implementasi Manajemen Proses Jaringan pada Warnet. Skripsi Program Studi Teknik Elektro Institut Teknologi Bandung.