

Manajemen Bandwidth Dengan Hierarchical Token Bucket (HTB) Pada Protokol TCP/IP Berbasis Linux

Eriek Orlando

Teknik Komputer, STMIK Jakarta STI&K
Jl. BRI No. 17 Radio Dalam, Kebayoran Baru, Jakarta Selatan
E-mail : eriek@jak-stik.ac.id

Abstrak

Manajemen bandwidth menjadi hal yang mutlak diperlukan bagi jaringan multi layanan, semakin banyak dan bervariasinya aplikasi yang dapat dilayani oleh suatu jaringan yang berpengaruh pada penggunaan link dalam jaringan tersebut. Link-link yang ada harus mampu menangani kebutuhan user akan aplikasi tersebut bahkan dalam keadaan congestion sekalipun. Harus ada suatu jaminan bahwa link tetap dapat berfungsi sebagaimana mestinya walaupun terjadi ledakan permintaan aplikasi. Pada penulisan ini akan dibahas mengenai Linux Traffic Control dengan mengimplementasikan manajemen bandwidth Hierarchical Token Bucket (HTB) pada PC router yang berbasis system operasi Linux. Manajemen bandwidth HTB ini bekerja pada lapisan network dalam jaringan TCP/IP. Software yang digunakan untuk implementasi ini adalah Cacti. Berdasarkan pengukuran yang dilakukan, penerapan manajemen bandwidth HTB pada jaringan TCP/IP menghasilkan nilai throughput yang mampu memenuhi ketentuan yang telah ditetapkan. Selain itu, implementasi manajemen bandwidth HTB dapat memanager semua aplikasi jaringan sehingga dapat menghasilkan performa jaringan yang lebih baik dan stabil.

Kata Kunci : HTB, TCP/IP, Bandwidth, Linux

Pendahuluan

Perkembangan komunikasi data telah mencapai kemajuan yang sangat pesat, ditandai oleh pemakaiannya yang begitu meluas sehingga tidak hanya dipakai oleh perusahaan-perusahaan berskala besar tetapi juga oleh individu-individu yang membutuhkan informasi terbaru dan cepat melalui jaringan komunikasi data terutama melalui Internet. Internet telah memberikan pengaruh yang sangat besar pada penyebaran informasi berupa data, dengan demikian akan semakin banyak individu yang mengakses data melalui internet. Hal ini akan berakibat pada meningkatnya trafik data yang dapat menyebabkan penurunan performansi jaringan terutama pada jaringan yang memiliki kapasitas bandwidth terbatas dan jaringan yang tidak berkeinginan untuk menambah kapasitas bandwidth. Untuk mengurangi penurunan performansi jaringan tanpa perlu melakukan penambahan biaya atau

kapasitas bandwidth, dapat dilakukan dengan salah satu cara yaitu dengan cara menerapkan manajemen bandwidth. Menurut survei, sekarang ini lebih banyak pengguna jaringan menggunakan PC router daripada router buatan pabrik. Hal ini dikarenakan untuk menekan biaya yang dikeluarkan dan kinerja PC router itu sendiri dalam mengatur performansi jaringan sama baiknya dengan router buatan pabrik. PC router yang digunakan sekarang ini banyak diimplementasikan pada PC router yang berbasis sistem operasi Linux.

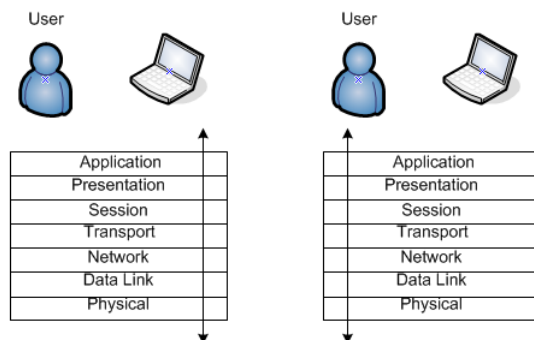
Untuk mengatur performansi jaringan dilakukan dengan cara menerapkan manajemen bandwidth. Manajemen bandwidth akan dilakukan pada jaringan TCP/IP yaitu pada lapisan network, dikarenakan manajemen bandwidth yang dilakukan berdasarkan IP address akan me-manage semua aplikasi dalam jaringan sehingga menghasilkan performansi jaringan paling baik. Dalam penulisan

ini akan diimplementasikan manajemen bandwidth pada PC router yang berbasis sistem operasi Linux. Di dalam sistem operasi Linux itu sendiri ada banyak cara untuk melakukan manajemen bandwidth. Algoritma manajemen bandwidth yang paling baik dalam sistem operasi Linux adalah algoritma Hierarchical Token Bucket (HTB). Dikarenakan algoritma ini melakukan manajemen bandwidth pada jaringan TCP/IP yaitu pada lapisan network.

Tinjauan Pustaka

Model Open System Interconnection (OSI)

Model OSI membagi kompleksitas komunikasi data dari asal (source) ke tujuan (destination) dengan melalui lapisan-lapisan (layer), dimana setiap lapisan memiliki fungsi layanan masing-masing dan hubungan antar lapisan. Dimana user sebagai pemakai yang menggunakan aplikasi berinteraksi dengan mengirim data ke tujuan. Data tersebut mengalir dari sumbernya melalui program yang dikelola oleh sistem operasi. Program ini dimulai dari lapisan application, ke lapisan transport, lapisan network, lapisan data link, sampai lapisan paling bawah yaitu lapisan physical. Lapisan physical mengubah data menjadi sinyal (dalam bentuk elektronik), kemudian sinyal ini akan diterima ditempat tujuan dan diubah kembali menjadi data yang akhirnya sampai ke penerima. Dalam arsitektur model OSI, protokol dapat dibagi menjadi tujuh lapisan.



Gambar 1: Model OSI

Seperti yang telah dijelaskan diatas, dalam

model OSI protokol dibagi menjadi tujuh lapisan. Ketujuh lapisan tersebut adalah:

1. Lapisan Application Protokol pada lapisan ini berfungsi untuk menyediakan akses ke pengguna.
2. Lapisan Presentation Protokol pada lapisan ini berfungsi untuk memberikan kebebasan bagi lapisan application untuk memproses data dalam berbagai format.
3. Lapisan Session Protokol pada lapisan ini berfungsi untuk menangani fungsi control dalam komunikasi antar aplikasi, seperti memulai koneksi, manajemen koneksi, dan mengakhiri koneksi.
4. Lapisan Transport Protokol pada lapisan ini berfungsi untuk melakukan transfer data antar komputer, menangani kesalahan yang terjadi (error recovery), dan menjamin data diterima seluruhnya dengan urutan yang benar (flow control).
5. Lapisan Network Protokol pada lapisan ini berfungsi untuk memulai koneksi, manajemen koneksi, dan mengakhiri koneksi antar komputer.
6. Lapisan Data Link Protokol pada lapisan ini berfungsi untuk melakukan transmisi data melalui koneksi fisik, melakukan pembungkahan, mendeteksi kesalahan (error detection), dan menyesuaikan kecepatan transfer data (flow control).
7. Lapisan Physical Protokol pada lapisan ini berfungsi untuk hubungan dengan spesifikasi antarmuka yang digunakan (mekanik, elektrik, fungsional, dan procedural).

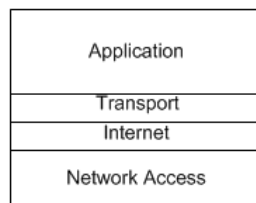
Model TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) adalah protokol yang akan dikembangkan dalam sebuah eksperimen jaringan packet switched pada tahun 1969 yang didanai oleh Defence Advanced Research Project Agency (DARPA) dengan nama ARPANET. Protokol ini ditujukan untuk menghubungkan perangkat keras dan perangkat lunak yang berbeda.

Protokol TCP/IP terkenal karena sifatnya yang sangat terbuka, yang berarti

bahwa protokol ini dapat menghubungkan segala jenis komputer dengan arsitektur yang berbeda. Interoperabilitas inilah yang menyebabkan TCP/IP banyak digunakan di seluruh dunia, untuk menghubungkan berbagai sistem yang berbeda tanpa harus mengkhawatirkan masalah ketidakkompatibilitas antar sistem.

Model TCP/IP

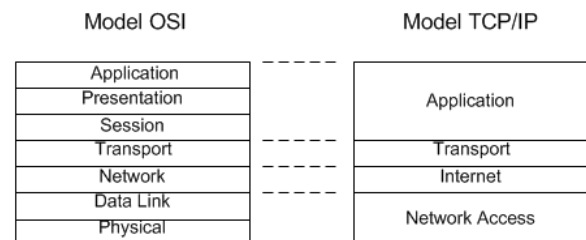


Gambar 2: Model TCP/IP

TCP/IP dibagi menjadi empat lapisan protokol yang bertingkat. Keempat lapisan tersebut adalah :

1. Lapisan Network Access Protokol pada lapisan ini berfungsi untuk mengatur pertukaran data antara end- system media fisik, seperti kabel.
2. Lapisan Internet Protokol pada lapisan ini berfungsi untuk melakukan proses pengiriman paket data ke alamat yang tepat. Pada lapisan ini TCP/IP mendukung Internet Protocol (IP) dan didukung oleh protokol lain yaitu:
 - (a) Address Resolution Protocol (ARP) ARP digunakan untuk menyesuaikan alamat IP dengan MAC Address.
 - (b) RARP (Reverse Address Resolution Protocol (RARP) RARP memungkinkan host menemukan alamat IP nya jika dia sudah tahu MAC Address-nya. Ini berlaku pada saat host baru terkoneksi ke jaringan.
 - (c) Internet Control Message Protocol (ICMP) ICMP adalah suatu mekanisme yang digunakan oleh sejumlah host dan gateway untuk mengirim notifikasi datagram yang mengalami masalah kepada host pengirim.
3. Lapisan Transport Protokol pada lapisan ini berfungsi untuk mengadakan komunikasi antara dua buah end-system yang terlibat dalam komunikasi. Pada lapisan ini terdapat dua protokol, yaitu Transmission Control Protocol (TCP) dan User Datagram Protocol (UDP).
4. Lapisan Application Protokol ini adalah kombinasi lapisan session, presentation dan application pada model OSI. Semua aplikasi yang menggunakan protokol TCP/IP terdapat pada lapisan ini.

Perbandingan antara model OSI dengan TCP/IP akan terdapat berbagai macam persamaan dan perbedaan



Gambar 3: Pemetaan Lapisan OSI Terhadap Lapisan TCP/IP

Internet Protocol (IP)

Protokol IP merupakan inti dari protokol TCP/IP. Seluruh data yang berasal dari protokol pada lapisan diatas IP harus dilewatkan, diolah oleh protokol IP, dan dipancarkan sebagai paket IP, agar sampai ke tujuan. IP memiliki sifat unreliable, connectionless, dan datagram delivery service dalam pengiriman paket data.

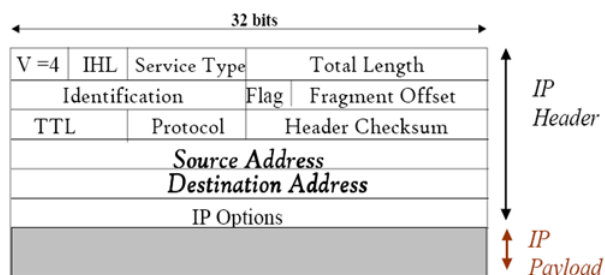
Unreliable atau ketidakhandalan berarti bahwa protokol IP tidak memberikan jaminan datagram yang dikirim akan sampai ke tujuan. Protokol IP hanya menjamin akan melakukan usaha yang sebaik-baiknya (best effort delivery service), agar paket yang dikirim tersebut sampai ke tujuan.

Connectionless berarti dalam pengiriman paket data dari tempat asal sampai tujuan, pihak pengirim dan penerima paket IP tidak perlu membuat suatu hubungan terlebih dahulu.

Datagram delivery service berarti setiap paket data yang dikirim adalah independen terhadap paket data yang lainnya. Akibatnya, jalur yang ditempuh oleh masing-masing paket data IP ke tujuannya bisa jadi berbeda satu dengan lainnya. Karena jalur yang ditempuh berbeda, maka paket yang sampai ke tujuan bisa jadi tidak berurutan (acak).

Setiap paket yang dikirim dari sumber ke tujuan melalui Internet disebut dengan IP datagram. Didalam IP datagram terdapat dua bagian yaitu:

1. Datagram Header: berisi alamat sumber dan tujuan paket data dan berukuran fixed.
2. Datagram Data Area: berisi besar payload yang dibawa oleh datagram dan berukuran hingga 64 KB.



Gambar 4: Format IP Datagram

Dalam pengiriman paket data dari sumber ke tujuan, IP meletakkan headernya sendiri pada paket data yang diterimanya dari TCP yang biasa disebut datagram header. Tanpa datagram header dari IP pada paket data tersebut, router perantara antara sumber dan tujuan tidak akan bisa menentukan bagaimana cara mengirimkan paket data tersebut. Format IP Datagram, menunjukkan setiap paket IP membawa datagram header yang terdiri atas :

1. Version (4 bit): menunjukkan versi dari protokol IP yang digunakan. Pada saat ini versi IP yang dipakai adalah IP versi 4.
2. Header Length (4 bit): menunjukkan panjang header paket IP.
3. Type of Service (8 bit): berisi kualitas service yang dapat mempengaruhi cara

penanganan paket IP. TOS ini terdiri dari precedence field (3 bit), TOS subfield (4 bit), dan satu bit yang terakhir tidak digunakan dan harus di-set menjadi 0. Pada TOS subfield (4 bit), dapat memberikan layanan klasifikasi kelas maksimum hingga 16 layanan.

4. Total Length (16 bit): menunjukkan panjang total IP datagram dalam ukuran byte. Maksimum panjang IP datagram adalah 65536 bytes.
5. Identification (16 bit): setiap paket yang dilewatkan mengalami fragmentasi (dipecah-pecah menjadi beberapa paket yang lebih kecil) sesuai dengan besar data maksimal yang bisa ditransmisikan melalui jalur tersebut, hal ini menunjukkan identitas datagram yang dikirim. Field ini berguna untuk penentuan urutan datagram pada saat disusun kembali pada bagian penerima.
6. Flags (3 bit): terdiri dari dua bagian yaitu Don't Fragment (DF) dan More Fragment (MF). DF berarti datagram tidak boleh difragmentasi. MF berarti fragmentasi terakhir dari datagram
7. Fragmentation Offset (13 bit): menunjukkan ukuran offset data pada fragmentasi paket dari data original (dalam ukuran bytes). Contoh: ukuran datagram 4000 bytes, ukuran fragmentasi 1500 bytes, ukuran offset: 0, 1500, 3000 bytes.
8. Time to live (8 bit): menunjukkan waktu propagansi maksimum atau hop count yang dapat dilewati oleh paket IP. Setiap kali paket IP melewati satu router, isi dari field ini dikurangi satu. Jika TTL telah habis dan paket tetap belum sampai ke tujuan, paket akan dibuang dan router terakhir akan mengirimkan paket ICMP time exceeded. Hal ini dilakukan untuk mencegah paket IP terus berada didalam jaringan.
9. Protocol (8 bit): mengandung angka yang mengidentifikasikan protokol atas pengguna isi data dari paket IP.
10. Header Checksum (16 bit): berisi nilai checksum yang dihitung dari seluruh field

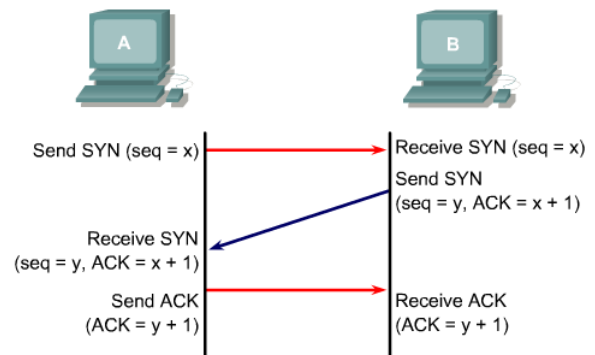
dari header paket IP. Sebelum dikirimkan, protokol IP terlebih dahulu menghitung checksum dari header paket IP tersebut untuk nantinya dihitung kembali di sisi penerima. Jika terjadi perbedaan ($\neq 0$), maka paket ini akan dianggap rusak dan dibuang.

11. Source Address (32 bit) dan Destination Address (32 bit): menunjukkan alamat pengirim paket dan penerima paket. Option: menunjukkan informasi tambahan yang dibawa datagram, misalkan untuk tingkat kerahasiaan datagram (security).

Transmission Control Protocol (TCP)

TCP merupakan protokol yang terletak di lapisan transport. Fungsi utama TCP adalah untuk menjamin keutuhan data yang dikirim antara host-host pada jaringan. Protokol ini memiliki sifat connection-oriented, reliable, dan byte stream service. Lapisan transport melakukan 2 fungsi utama yaitu:

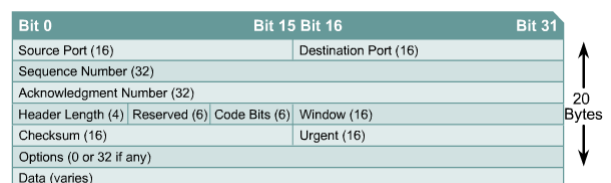
1. Flow Control, melalui Sliding Window Lapisan transport mengirim data segment, lapisan ini menjamin data tersebut tidak akan hilang. Data loss mungkin terjadi jika satu host tidak dapat memproses data dengan cepat begitu data tersebut datang. Kemudian host akan membuang paket data tersebut. Flow control menjamin paket dari host sumber tidak lebih dari besar dari buffer yang ada pada host tujuan. Dengan menyediakan flow control, TCP memperbolehkan host sumber dan host tujuan untuk berkomunikasi.
2. Transmisi yang reliable, melalui nomor urut (sequence number) dan Acknowledgment. Connection-oriented berarti sebelum melakukan pertukaran data atau komunikasi, kedua aplikasi pengguna TCP harus melakukan pembentukan hubungan terlebih dahulu. Hal ini berarti sebelum melakukan pengiriman data, mesin-mesin tersebut akan melalui 3 langkah cara berhubungan (3-way handshake). Pertama adalah initial request yaitu permintaan untuk mengirim, kedua adalah jawaban (reply) dari host yang jauh tersebut dan terakhir adalah Acknowledge atau jawaban kembali atas reply tersebut.



Gambar 5: Three-way handshake

Konsep reliable berarti TCP akan mendeteksi error pada paket yang akan dikirim dan bila itu terjadi, maka paket akan dikirim kembali. Sedangkan konsep byte stream service berarti paket-paket yang dikirimkan akan sampai ke tujuan secara berurutan. Hal ini dilakukan TCP pada sisi penerima dengan cara pengurutan kembali data yang diterima, dan memberikan data dengan urutan yang benar ke aplikasi penerimanya.

Untuk mengirimkan data ke tujuan, sebelumnya TCP membagi-bagi data ke dalam bagian-bagian kecil, yang disebut dengan datagram. Konsep datagram akan dilengkapi dengan header yang dibutuhkan untuk mengarahkan data tersebut ke tujuannya, serta merangkai kembali bagian-bagian data tersebut dengan benar jika sudah sampai pada tujuannya.



Gambar 6: Format TCP Header

Bandwidth

Seperti telah kita tahu, bandwidth paling banyak digunakan sebagai ukuran kecepatan aliran data. Tetapi apakah itu bandwidth sebenarnya? Bandwidth adalah suatu ukuran dari banyaknya informasi yang dapat mengalir dari suatu tempat ke tempat lain dalam suatu

waktu tertentu. Bandwidth dapat dipakaikan untuk mengukur baik aliran data analog mau pun aliran data digital. Sekarang telah menjadi umum jika kata bandwidth lebih banyak dipakaikan untuk mengukur aliran data digital. Satuan yang dipakai untuk bandwidth adalah bits per second atau sering disingkat sebagai bps. Seperti kita tahu bahwa bit atau binary digit adalah basis angka yang terdiri dari angka 0 dan 1. Satuan ini menggambarkan seberapa banyak bit (angka 0 dan 1) yang dapat mengalir dari satu tempat ke tempat yang lain dalam setiap detiknya melalui suatu media.

Bandwidth adalah konsep pengukuran yang sangat penting dalam jaringan, tetapi konsep ini memiliki kekurangan atau batasan, tidak peduli bagaimana cara Anda mengirimkan informasi mau pun media apa yang dipakai dalam penghantaran informasi. Hal ini karena adanya hukum fisika maupun batasan teknologi. Ini akan menyebabkan batasan terhadap panjang media yang dipakai, kecepatan maksimal yang dapat dipakai, mau pun perlakuan khusus terhadap media yang dipakai. Rumus dari bandwidth :

$$bandwidth = \frac{\sum bits}{n} \quad (1)$$

Throughput

Throughput adalah suatu bandwidth efektif selama proses transmisi data antara satu host dengan host lainnya dalam jaringan komputer. Ternyata konsep bandwidth tidak cukup untuk menjelaskan bandwidth yang sebenarnya pada suatu waktu real-time. Untuk itulah konsep throughput muncul. Sedangkan throughput walaupun memiliki satuan dan rumus yang sama dengan bandwidth, tetapi throughput lebih pada menggambarkan bandwidth yang sebenarnya (aktual) pada suatu waktu tertentu dan pada kondisi dan jaringan internet tertentu yang digunakan untuk mendownload suatu file dengan ukuran tertentu. Berikut adalah formula pembandingan throughput dengan bandwidth:

$$waktu\ download\ terbaik = \frac{ukuran\ file}{bandwidth} \quad (2)$$

$$waktu\ download\ typical = \frac{ukuran\ file}{throughput} \quad (3)$$

Metode Penelitian

Penelitian ini mengacu dari refensi yang didapatkan untuk membantu membuat penulisan penelitian ini bersumber dari :

1. Sumber acuan umum, berupa buku teks dan ensiklopedia.
2. Sumber acuan khusus, berupa: jurnal, bulletin penelitian dan sumber acuan lain yang memuat hasil-hasil penelitian yang relevan dengan judul penelitian yang penulis lakukan.

Hasil dan Pembahasan

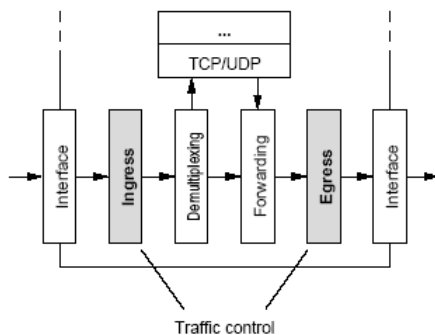
Proses paket pada kernel linux

Layanan Internet sekarang ini bekerja berdasarkan protocol IP dan mendukung layanan best-effort. Dengan pertumbuhan Internet yang semakin pesat, jaringan TCP/IP tidak hanya harus mendukung layanan seperti File Transfer Protocol (FTP) dan e-mail, tetapi juga harus mendukung layanan real-time dan aplikasi video streaming. Karakteristik jaringan yang digunakan untuk aplikasi tersebut adalah bandwidth dan delay. Kedua karakteristik ini dapat ditangani dengan layanan QoS dengan mekanisme Differentiated Services (DiffServ).

Kemudian, masalah terbesar yang akan timbul pada suatu jaringan yang saling berhubungan adalah bagaimana cara mengalokasikan sumber jaringan, seperti buffer dan link bandwidth untuk user yang berbeda. Untuk jumlah sumber jaringan yang terbatas ini maka diperlukan trafik control yang baik agar performansi jaringan tetap terjaga dengan baik. Traffic Control atau tc merupakan salah satu mekanisme yang digunakan untuk mengetahui kondisi trafik jaringan dan aplikasinya. Biasanya cara yang digunakan adalah dengan memberikan beberapa trafik prioritas yang lebih tinggi dibanding dengan yang lainnya, hal ini berguna untuk membatasi trafik yang dikirim dan untuk mencegah trafik yang tidak diinginkan.

Proses paket dalam kernel Linux memakai layanan Differentiated Services dalam QoS. Traffic control di linux terjadi antara proses ingress node dan egress node. Ingress node merupakan satu kumpulan fungsi-fungsi yang diperlukan untuk menangani aliran trafik yang

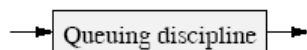
masuk ke dalam satu DS-domain, fungsi-fungsi tersebut antara lain seperti membuang paket-paket yang tidak diinginkan dan fungsi untuk proses klasifikasi paket-paket. Egress node merupakan satu kumpulan fungsi-fungsi yang diperlukan untuk menangani aliran trafik yang keluar dari satu DS-domain, termasuk fungsi kontrol trafik untuk antrian paket-paket.



Gambar 7: Proses paket pada kernel Linux

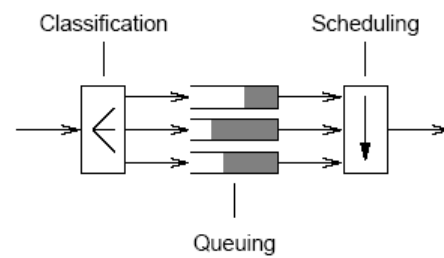
Linux Traffic Control

Linux traffic control mengkombinasikan antrian dan penjadwalan paket ke dalam prosesnya sehingga disebut "queuing disciplines" atau qdisc. Setiap network device memiliki satu antrian disiplin didalamnya, dimana antrian ini mengontrol paket-paket yang masuk ke dalam device tersebut.



Gambar 8: Suatu antrian disiplin yang sederhana tanpa adanya class

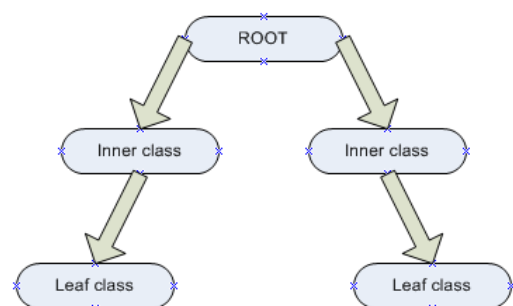
Faktor utama yang termasuk dalam traffic control pada kernel linux adalah classification, scheduling, dan queuing. Pada elemen classification mengandung isi paket atau informasi lain yang terhubung dengan paket dan atribut-atribut classes. Kemudian paket-paket akan ditaruh ke dalam antrian atau queuing. Selanjutnya penjadwalan untuk proses pengiriman. Kelas paket menggambarkan antrian paket-paket yang akan diproses dan dijadwalkan atau scheduling.



Gambar 9: Struktur Traffic Control

Algoritma queuing disciplines atau disiplin antrian terbagi 2 macam:

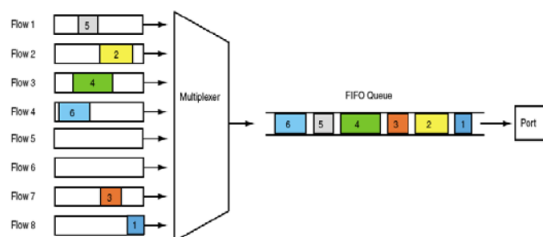
1. Classless Queuing Disciplines Setiap algoritma yang termasuk dalam classless qdisc biasanya digunakan sebagai primer qdisc untuk interface. Selain itu, biasanya digunakan dalam suatu leaf class yang termasuk dalam classful qdisc. Algoritma ini merupakan dasar penjadwalan yang digunakan dalam kernel linux.
2. Classfull Queuing Disciplines Fleksibilitas dan kontrol dalam linux traffic control merupakan bagian dari classful qdisc. Algoritma ini dapat mem-filter paket yang datang, dan melewatkannya ke dalam kelas dan sub-antrian. Ada beberapa contoh umum yang dapat menggambarkan kelas yang langsung di-attached ke root qdisc dan terminal class. Kelas yang di-attached ke root qdisc biasanya disebut root class dan dibawahnya atau didalamnya terdapat kelas inner class. Semua kelas terminal merupakan bagian antrian disiplin yang disebut sebagai leaf class dengan analogi kelas seperti struktur pohon.



Gambar 10: Struktur pohon

First In First Out (FIFO)

Algoritma FIFO merupakan teknik yang paling sederhana dalam disiplin antrian untuk membatasi bandwidth. Konsep algoritma ini adalah paket yang pertama kali masuk atau datang akan menjadi paket yang pertama kali dilayani terlebih dahulu. Ketika antrian paket sedang penuh dan apabila datang sebuah paket baru, paket tersebut akan dibuang. Gambar 11 merupakan contoh dari algoritma FIFO, dimana paket-paket dari aliran yang berbeda tiba di router kemudian paket-paket tersebut akan di-multiplex kedalam algoritma FIFO (1,2,3,4,5,6).



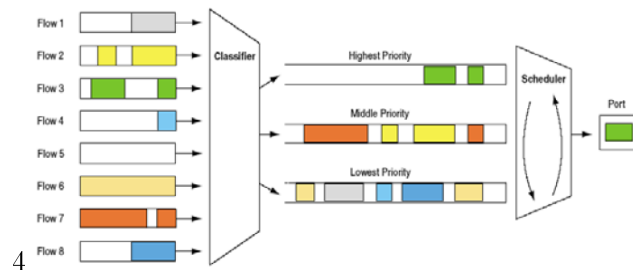
Gambar 11: Algoritma FIFO

Priority Queuing (PQ)

Algoritma PQ merupakan suatu metode yang mempermudah untuk sebuah layanan Differentiated Service (DiffServ). Nama lain algoritma Priority Queuing (PQ) adalah prio. Teknik ini menggunakan multiple antrian, yang berarti antrian akan dilayani berdasarkan perbedaan tingkat prioritas dan antrian dengan prioritas paling tinggi akan dilayani pertama kali atau terlebih dahulu. Paket-paket akan ditempatkan dalam antrian berdasarkan klasifikasi. Penjadwalan paket-paket dimulai dari antrian yang memiliki prioritas paling tinggi sampai prioritas paling rendah atau sampai habis. Penjadwalan paket menggunakan algoritma FIFO dalam setiap prioritas antrian. Apabila terjadi kemacetan, paket akan dibuang dari antrian yang memiliki prioritas paling rendah.

Dalam gambar tersebut digunakan tiga prioritas dalam antrian paket-paket. Aliran ketiga merupakan kelas yang memiliki prioritas paling tinggi, lalu aliran kedua dan ketujuh merupakan kelas yang memiliki prioritas menengah dan aliran yang lainnya merupakan kelas

yang memiliki antrian prioritas paling rendah. Pertama-tama, paket dari antrian yang memiliki prioritas paling tinggi akan dilayani sesuai algoritma FIFO. Hanya sekali antrian yang memiliki prioritas paling tinggi boleh kosong, selanjutnya antrian akan dilayani dengan cara yang sama.



Gambar 12: Algoritma PQ

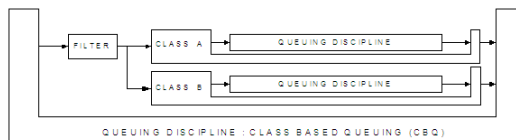
Class Based Queuing (CBQ)

Ide utama dibalik algoritma CBQ adalah penjadwalan dalam antrian paket data dan garansi untuk kecepatan transmisi. Jika tidak ada paket data dalam satu antrian, maka bandwidth akan dibuat sedemikian rupa hingga dapat tersedia dalam antrian-antrian paket yang lain. Kekuatan dari metode ini adalah mampu melewati atau menanggulangi arus dengan berbagai macam perbedaan bandwidth minimum. Algoritma CBQ ini menghindari masalah bandwidth mati yang terjadi pada algoritma PQ, yaitu paling sedikit satu paket yang akan dilayani dari setiap antrian selama setiap layanan. Algoritma CBQ dan WFQ hampir mirip antar keduanya dan kedua algoritma ini mencoba untuk mendapatkan keuntungan dari algoritma PQ dan FQ. Satu perbedaan antara algoritma CBQ dan WFQ adalah penjadwalan paket dengan cara algoritma CBQ lebih teratur dan tidak mengambil info atau laporan tentang panjang paket.

Konsep algoritma ini adalah pertama-tama, paket-paket akan diklasifikasikan kedalam suatu layanan kelas berdasarkan beberapa kriteria (pengalamatan, protokol, user, dan lain-lain) lalu disimpan dalam antrian. Setiap antrian akan dilayani dengan cara algoritma round-robin. Perbedaan jumlah bandwidth untuk setiap antrian dibedakan menjadi dua cara, yaitu mengizinkan satu antrian untuk mengirimkan lebih dari satu paket untuk setiap

layanan dan mengijinkan satu antrian untuk mengirimkan paket hanya boleh satu untuk setiap layanan tetapi pada antrian yang sama diijinkan untuk bekerja sama dengan antrian yang lain untuk setiap layanan yang sama.

Algoritma CBQ merupakan implementasi sistem trafik kontrol klasik. Di dalam algoritma ini berisi pengiriman elemen sama baiknya dengan prioritas kapabilitas. Pengiriman menggunakan jalur kalkulasi waktu idle berdasarkan waktu dalam antrian dan jalur bandwidth. Algoritma ini juga merupakan kombinasi antara partisi dan pembagian jalur bandwidth dengan struktur kelas yang hirarki. Setiap kelas memiliki antrian sendiri dan ditandai untuk pembagian bandwidth. Setiap kelas child dapat dikonfigurasi untuk meminjam bandwidth dari kelas parent selama bandwidth tersebut tersedia.



Gambar 13: Linux Trafik Control dengan CBQ

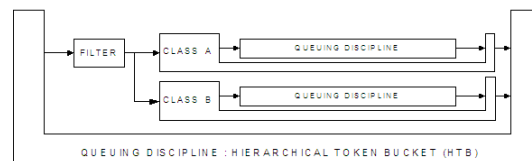
Hierarchical Token Bucket (HTB)

HTB merupakan termasuk classful queuing discipline yang dibuat oleh Martin Devera dengan metode mendistribusikan jumlah bandwidth menjadi beberapa kelas yang tersusun secara hirarki. HTB adalah teknik yang digunakan dalam Linux Traffic Control untuk menyediakan layanan QoS. Bentuk trafik HTB yang terjadi berdasarkan algoritma TBF, dimana tidak tergantung dengan karakteristik interface dan tidak perlu tahu berapa jumlah bandwidth yang keluar dari interface. Baik CBQ atau HTB dapat digunakan untuk mengontrol jalan dan jumlah bandwidth. Algoritma HTB menggunakan lapisan network OSI untuk manajemen bandwidth sama dengan algoritma CBQ.

HTB menggunakan konsep token dan bucket dengan sistem class-based dan filter untuk melewati trafik kontrol yang kompleks. Dengan model peminjaman yang kompleks, HTB dapat menunjukkan teknik trafik kontrol yang begitu bagus, khususnya dalam manajemen bandwidth. Satu jalan yang memudahkan

dalam penggunaan algoritma HTB adalah trafik shaping. Trafik shaping digunakan untuk mengontrol aliran trafik yang masuk ke dalam jaringan dan menspesifikasikan rate untuk paket-paket yang akan ditransmisikan. Teknik antrian disiplin ini menggunakan konsep token bucket filter untuk menangani masalah pembagian bandwidth.

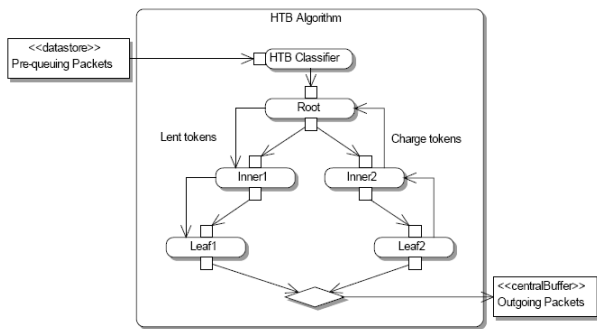
Diagram blok HTB untuk Traffic Control pada Linux sama persis dengan CBQ dengan komponen-komponen yang serupa pula



Gambar 14: Linux Trafik Control dengan HTB

Root qdisc ini bisa berisi satu kelas atau beberapa kelas tergantung dari topologi jaringan. Satu kelas HTB umumnya berisi dengan 2 parameter, yaitu RATE dan CEIL. Kelas parent terletak pada kelas paling atas dari suatu hirarki dan harus menggambarkan jumlah total dari jumlah bandwidth kelas-kelas dibawahnya. Dalam HTB, RATE berarti besar bandwidth yang akan digaransi (bandwidth minimum) yang akan diberikan kelas tersebut, dan CEIL berarti besar bandwidth burstable (bandwidth maksimum) yang bisa didapat kelas untuk dipakai. Semua bandwidth dapat digunakan antara rate dan ceil yang dipinjamkan dari kelas parent, dengan skenario parameter RATE dan CEIL ini sama dengan jumlah bandwidth kelas paling atas atau kelas parent.. Jumlah kelas child yang dibuat terletak didalam kelas parent, setiap kelas child akan diloalokasikan jumlah bandwidth (tergantung pemakaian) dari jumlah bandwidth yang tersedia dari kelas parent tersebut. Dalam kelas child, nilai parameter RATE dibuat berdasarkan kebutuhan pemakai (bandwidth minimum), sedangkan nilai parameter CEIL harus sesuai dengan jumlah bandwidth dari kelas parent tersebut. HTB merupakan implementasi dari mekanisme classfull queuing discipline untuk sistem linux trafik control dan menyediakan parameter RATE dan CEIL untuk memudahkan pemakai mengontrol jumlah total bandwidth untuk kelas-

kelas dibawahnya dengan menggunakan ratio dalam distribusi bandwidth ketika terjadi bandwidth ekstra atau bandwidth yang tidak digunakan dalam kelas parent tersebut.



Gambar 15: Algoritma HTB

Shaping HTB

Salah satu cara dalam HTB untuk manajemen trafik adalah trafik shaping yang men-transmisikan paket-paket menggunakan spesifik rate. Semua proses trafik shaping terjadi dalam kelas-kelas leaf. Algoritma HTB tidak akan melakukan delay paket dalam kelas-kelas inner. Tidak ada trafik shaping yang terjadi dalam kelas inner atau root, melainkan pada kelas inner atau root menunjukkan bagaimana mekanisme borrowing bekerja seharusnya mendistribusikan token yang tersedia. Trafik shaping memiliki beberapa kriteria, antara lain:

1. Menggunakan limit bandwidth dengan menunda antrian paket-paket yang melebihi batas atau limit dan mengirimkan paket-paket tersebut nanti ketika permintaan untuk bandwidth lebih rendah.
2. Membuang paket-paket jika buffer paket telah penuh. Paket yang akan dibuang atau di-drop harus dipilih dari aliran paket yang memiliki trafik paling padat.
3. Memprioritaskan trafik sesuai dengan pilihan administrator. Jika trafik yang memiliki prioritas paling tinggi bertambah maka jalur komunikasi menjadi penuh, trafik yang memiliki prioritas paling rendah harus untuk sementara memberi tempat untuk trafik yang memiliki prioritas paling tinggi.

4. Menyediakan garansi bandwidth.

Borrowing HTB

Salah satu bagian dasar dari HTB qdisc adalah mekanisme borrowing. Kelas-kelas Child meminjam token dari kelas-kelas parent mereka yang dinamakan RATE. Setiap kelas-kelas child akan terus meminjam sampai batas paling tinggi dari kelas-kelas parent yang disebut CEIL. Setelah sampai ceil, akan dimulai antrian paket untuk ditransmisikan hingga token yang kosong tersedia. Hanya ada dua tipe borrowing yang dapat dibuat dengan HTB. Tabel 1 adalah beberapa kemungkinan yang memungkinkan terjadinya mekanisme borrowing.

Penutup

Implementasi manajemen bandwidth HTB pada jaringan TCP/IP memiliki beberapa kesimpulan antara lain adalah:

1. Pada implementasi manajemen bandwidth HTB pada jaringan TCP/IP dengan menggunakan PC router, ternyata throughput yang dihasilkan dengan menggunakan manajemen bandwidth HTB secara umum mampu memenuhi ketentuan atau rule yang telah ditetapkan. Akan tetapi, karena fluktuasi trafik yang datang setiap saat berbeda-beda maka kadang-kadang throughput yang telah dibatasi atau di-manage mengalami kebocoran. Selain itu, selisih nilai throughput antara perhitungan dengan real-time hanya sedikit atau tidak terlalu berbeda jauh.
2. Pada implementasi manajemen bandwidth HTB pada jaringan lokal, throughput yang dihasilkan lebih stabil dibandingkan pengukuran pada jaringan Internet. Hal ini dikarenakan, pada jaringan Internet bandwidth yang diperoleh tergantung dengan koneksi Internet yang didapat dari ISP.
3. Berdasarkan dari hasil pengukuran, performa jaringan yang tidak menggunakan manajemen bandwidth mengalami penurunan dan dapat menyebabkan timbulnya monopoli bandwidth oleh salah satu atau beberapa client sehingga menyebabkan timbunan antrian paket atau backlog

pada client-client yang lain. Sedangkan performa jaringan yang menggunakan manajemen bandwidth HTB dengan atau tanpa meminjam bandwidth idle mengalami kestabilan.

4. Berdasarkan hasil pengukuran, ternyata implementasi manajemen bandwidth

HTB dapat me-managed semua aplikasi jaringan sehingga dapat menghasilkan performa jaringan yang begitu baik. Hal ini disebabkan manajemen bandwidth HTB bekerja pada layer network dan memiliki kemampuan untuk menjadwalkan antrian paket berdasarkan kelas-kelas yang tersusun secara hirarki.

Tabel 1: HTB class states and potential actions taken

Tipe kelas	Keadaan kelas	Keadaan internal HTB	Tindakan
LEAF	< RATE	HTB_CAN_SEND	Kelas leaf tidak akan meneruskan antrian bytes hingga token yang ada tersedia (tidak lebih dari paket burst)
LEAF	> RATE, < CEIL	HTB_MAY_BORROW	Kelas leaf mungkin akan meminjam token/ctoken dari kelas parent. Jika token kosong tersedia, maka akan terus meminjam dengan menambah nilai quantum (=1500 bytes) dan kelas leaf tidak akan meneruskan antrian paket sampai cburst byte.
LEAF	> CEIL	HTB_CANT_SEND	Tidak ada paket-paket yang akan diantrikan, yang menyebabkan delay paket dan nilai latensi akan bertambah hingga mencapai kecepatan yang diinginkan
inner, root	< RATE	HTB_CAN_SEND	Kelas inner akan meminjam token-token untuk kelas di bawahnya
inner, root	> RATE, < CEIL	HTB_MAY_BORROW	Kelas inner akan mencoba untuk meminjam token/ctoken dari kelas parent, hal ini digunakan untuk bersaing dengan kelas child pada penambahan nilai quantum setiap permintaan.
inner, root	> CEIL	HTB_CANT_SEND	Kelas inner tidak akan mencoba untuk meminjam token dari kelas parent dan tidak akan meminjam token/ctoken untuk kelas child

Daftar Pustaka

- [1] Laohanpengsang, O 2009. "factor influencing internet shopping behaviour: a survey of consumers in Thailand." journal of fashion marketing and manangement vol.13 No.4,2009 pp.501-513.
- [2] Anhar, 2010, Php & Mysql Secara Otodidak, Mediakita, Jakarta.
- [3] Betha Sidik, 2014, Pemrograman Web dengan PHP, Penerbit Informatika, Bandung.
- [4] Andi Sunyoto, 2007, Ajax Membangun Web dengan Teknologi Asynchrone JavaScript & XML, Penerbit Andi, Yogyakarta.
- [5] Royhan Nahado, 2010, Bumbu-Bumbu Membuat Website: Cara Membuat Website Cantik Dan Mudah, <http://bangroyhan.pas.in/>.
- [6] Fani Ariasari, 2010, Panduan Praktis Bikin Blog dengan Wordpress, Mediakita, Jakarta.

- [7] Janner Simarmata, 2010, Rekayasa Web, Penerbit Andi, Yogyakarta.
- [8] E-Media Solusindo, 2012, Membangun Komunitas Online Secara Praktis dan Gratis, PT. Elex Media Komputindo, Jakarta.
- [9] <https://bambangdp.files.wordpress.com/2010/09/model-tradisional.jpeg?w=594>
- [10] <https://bambangdp.files.wordpress.com/2010/09/sikus-request-response.jpeg>