

ANALISA UNJUK KERJA APLIKASI CBQ DAN HTB PADA JARINGAN KOMPUTER UNTUK PEMBATASAN BANDWIDTH BERBASIS IPv6

Akhmad Lukman Al-Hakim¹, M. Zen Samsono Hadi, ST. M.Sc², Ir. Nanang Syahroni, M.Kom³
Jurusan Telekomunikasi - Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember, Kampus ITS, Surabaya 60111
e-mail : aloenk@student.eepis-its.edu

Abstrak

Manajemen *bandwidth* menjadi hal yang sangat diperlukan dalam jaringan. Dimana semakin banyak aplikasi yang dapat dilayani oleh suatu jaringan berpengaruh pada penggunaan *link* dalam jaringan tersebut. *Link-link* yang ada harus mampu memenuhi kebutuhan user akan aplikasi tersebut, bahkan dalam kondisi apapun, harus ada suatu jaminan bahwa *link* yang ada tetap dapat bekerja dengan baik. Dalam hal ini, pengaturan *bandwidth* sangat diperlukan untuk mengatur jenis aplikasi dalam mengakses *link*, serta memberi jaminan terhadap aplikasi tersebut yang telah dibagi alokasi *bandwidth*-nya untuk terus mengirimkan data sesuai dengan kapasitas yang telah diberikan, meskipun terjadi kemacetan dalam jaringan. Ataupun dalam kondisi pada saat suatu layanan/aplikasi tidak digunakan, maka alokasi *bandwidth* yang sedang *idle* tersebut dapat dialihkan kepada kelas yang mengalami *backlog* (timbunan) antrian. Sehingga dapat mempercepat hilangnya *backlog*, serta mengoptimalkan penggunaan *bandwidth* yang ada.

Aplikasi CBQ (Class Based Queuing) dan HTB (Hierarchical Token Bucket) merupakan aplikasi untuk manajemen *bandwidth* yang dijalankan pada Sistem Operasi Linux yang akan dianalisa kelebihan dan kelemahannya berdasarkan beberapa parameter. Dalam penelitian ini, aplikasi CBQ dan HTB akan dijalankan berbasis sistem pengalamatan IPv6 (*Internet Protocol version 6*).

Hasil dari analisa aplikasi CBQ dan HTB diperoleh bahwa aplikasi HTB belum mendukung dengan sistem pengalamatan IPv6, sedangkan aplikasi CBQ masih mampu mendukung sistem IPv6 tersebut. Dan penggunaan CBQ sebagai aplikasi pembatasan *bandwidth* harus memperhatikan parameter *bounded* dan *isolated*, dimana penggunaan parameter *isolated* pada *parent leaf* CBQ akan mengakibatkan kegagalan proses pembatasan trafik jaringan.

Kata Kunci : CBQ, HTB, bandwidth, link, backlog, IPv6

I. PENDAHULUAN

Semakin berkembangnya jaringan komputer pada perusahaan/instansi, diikuti dengan semakin tinggi pula pemakaian koneksi internet ataupun koneksi antar jaringan dalam perusahaan tersebut. Hal ini menyebabkan kapasitas bandwidth pada jaringan menjadi tidak memadai, bahkan cenderung *overload*, sehingga sebagian besar perusahaan melakukan penambahan bandwidth pada jaringan mereka, dengan harapan agar koneksi antar komputer maupun koneksi internet bisa lebih cepat. Padahal, belum tentu semua komputer dalam jaringan tersebut terkoneksi secara nonstop, mengingat dalam setiap perusahaan pastinya memiliki jam kerja tersendiri yang diatur sesuai kebijaksanaan perusahaan tersebut. Sehingga, bila perusahaan melakukan penambahan bandwidth, tentunya menjadi tidak efisien dan cenderung merugikan ketika beberapa komputer dalam jaringan tidak aktif (misalnya, pada saat malam hari, atau ketika hari libur). Untuk itu diperlukan suatu

manajemen bandwidth agar penggunaannya lebih efisien dan bisa memaksimalkan bandwidth yang ada.

Aplikasi CBQ (Class Based Queueing) dan HTB (Hierarchical Token Bucket) merupakan aplikasi yang bekerja pada platform Linux, dan digunakan untuk melakukan manajemen bandwidth pada jaringan. Dengan beberapa parameter, maka didapatkan suatu informasi yang akan menentukan QoS dari kinerja tiap aplikasi dalam hal manajemen *bandwidth* dan performansinya.

Pengalamatan berbasis IPv6 (*Internet Protocol version 6*) mampu memberikan alokasi alamat yang lebih luas dibandingkan IP versi sebelumnya (IPv4) yang sudah mulai habis. IPv6 juga dapat mengurangi kompleksitas proses routing dan routing table, sehingga jauh lebih efisien untuk diaplikasikan dalam jaringan.

II. TEORI PENUNJANG

2.1 CBQ (Class Based Queueing)

Teknik klasifikasi paket data yang paling terkenal adalah CBQ. Keunggulan dalam penggunaan CBQ adalah mudah dikonfigurasi, memungkinkan *sharing bandwidth* antar kelas (*class*) dan memiliki fasilitas *user interface*. CBQ adalah suatu algoritma yang didasarkan pada pembagian paket ke dalam kelas-kelas dan menjadwalkan paket di dalam antrian dengan suatu transmisi *rate* tertentu. Jika suatu antrian tidak digunakan, maka *bandwidth* disediakan untuk antrian lain. CBQ dapat mencapai penyekatan dan pembagian link *bandwidth* dengan struktur golongan. Tiap golongan mempunyai antrian tersendiri dan ditandai secara khusus, dimana juga dilakukan pembagian *bandwidth*. CBQ dapat mengatur penggunaan *bandwidth* dari suatu golongan. Golongan '*child*' dapat di konfigurasi untuk meminjam *bandwidth* dari golongan induknya selama kelebihan *bandwidth* masih tersedia. CBQ mengatur pemakaian *bandwidth* jaringan yang dialokasikan untuk tiap *user*, pemakaian *bandwidth* yang melebihi nilai set akan dipotong (*shaping*). CBQ juga dapat diatur untuk *sharing* dan meminjam *bandwidth* antar *class* jika diperlukan.

Parameter yang terdapat dalam aplikasi CBQ adalah :

- **avpkt**, jumlah paket rata – rata saat pengiriman
- **bandwidth**, lebar *bandwidth* kartu *ethernet* biasanya 10 – 100Mbit
- **rate**, kecepatan rata – rata paket data saat meninggalkan *qdisc*, ini parameter untuk men-set *bandwidth*.
- **cell**, peningkatan paket data yang dikeluarkan ke kartu *ethernet* berdasarkan jumlah *byte*, misalnya 800 ke 808 dengan nilai *cell* 8.
- **isolated / sharing**, parameter *isolated* mengatur agar *bandwidth* tidak bisa dipinjam oleh klas (*class*) lain yang sama tingkatannya / *sibling*. Parameter *sharing* menunjukkan *bandwidth* kelas (*class*) bisa dipinjam oleh kelas lain.
- **bounded / borrow**, parameter *borrow* berarti kelas (*class*) dapat meminjam *bandwidth* dari klas lain, sedangkan *bounded* berarti sebaliknya.

2.2 HTB (Hierarchical Token Bucket)

Hierarchical Token Bucket (HTB) merupakan jenis aplikasi yang dikembangkan oleh Martin Devera pada tahun 2001 yang digunakan untuk membatasi akses menuju ke port/IP tertentu tanpa mengganggu trafik *bandwidth* pengguna lain. Aplikasi ini berfungsi sebagai pengganti aplikasi yang masih sering digunakan, yaitu CBQ. HTB diklaim mampu melakukan pembagian trafik yang lebih akurat.

Teknik antrian HTB mirip dengan teknik pada CBQ. Hanya perbedaannya terletak pada opsi, dimana pada HTB opsi yang digunakan jauh lebih

sedikit dalam konfigurasinya, serta lebih presisi dalam penggunaannya. Teknik antrian HTB memberikan fasilitas pembatasan trafik pada setiap level ataupun klasifikasinya, sehingga *bandwidth* yang tidak terpakai dapat digunakan oleh klasifikasi lain yang lebih rendah.

2.3 Internet Protocol version 6 (IPv6)

2.3.1 Pendahuluan

IPv6 (*Internet Protocol version 6*) adalah sebuah jenis pengalamatan jaringan yang digunakan dalam protokol jaringan TCP/IP menggunakan protokol IP versi 6 dengan panjang totalnya 128 bit. IPv6 merupakan kelanjutan dari IPv4 yang telah digunakan sampai sekarang. Tujuan dari Ipv6 adalah untuk meningkatkan jumlah *IP address*. Dimana IPv4 menggunakan 32 bit, artinya jumlah *IP address* adalah 2^{32} (sekitar 4 miliar), dan sekarang sudah mulai habis. Secara teoritis, pengalamatan dengan IPv6 mencapai 128 bit, dengan perincian $2^{128} = 3,4 \times 10^{38}$ *host* di seluruh dunia. IPv6 juga membentuk infrastruktur *routing* yang disusun secara *hierarchies*, untuk mengurangi kompleksitas proses *routing* dan *routing table*.

Perubahan dari IPv4 ke IPv6 pada dasarnya terjadi karena beberapa hal yang dikelompokkan dalam kategori berikut :

1. Kapasitas perluasan alamat
IPv6 meningkatkan ukuran dan jumlah alamat yang mampu didukung oleh IPv4 dari 32 bit menjadi 128 bit. Peningkatan kapasitas alamat ini digunakan untuk mendukung peningkatan hirarki atau kelompok pengalamatan, peningkatan jumlah atau kapasitas alamat yang dapat dialokasikan dan diberikan pada node dan mempermudah konfigurasi alamat pada node sehingga dapat dilakukan secara otomatis. Peningkatan skalabilitas juga dilakukan pada *routing multicast* dengan meningkatkan cakupan dan jumlah pada alamat *multicast*. IPv6 ini selain meningkatkan jumlah kapasitas alamat yang dapat dialokasikan pada *node* juga mengenalkan jenis atau tipe alamat baru, yaitu alamat *anycast*. Tipe alamat *anycast* ini didefinisikan dan digunakan untuk mengirimkan paket ke salah satu dari kumpulan *node*.
2. Penyederhanaan format header
Beberapa kolom pada *header* IPv4 telah dihilangkan atau dapat dibuat sebagai *header* pilihan. Hal ini digunakan untuk mengurangi biaya pemrosesan hal-hal yang umum pada penanganan paket IPv6 dan membatasi biaya *bandwidth* pada *header* IPv6. Dengan demikian, pemrosesan *header* pada paket IPv6 dapat dilakukan secara efisien.

3. Peningkatan dukungan untuk header pilihan dan header tambahan (*Options and extention header*)
Perubahan yang terjadi pada *header-header* IP yaitu dengan adanya pengkodean *s* (pilihan) pada IP dimasukkan agar lebih efisien dalam penerusan paket (*packet forwarding*), agar tidak terlalu ketat dalam pembatasan panjang *header* pilihan yang terdapat dalam paket IPv6 dan sangat fleksibel/dimungkinkan untuk mengenalkan *header* ipilihan baru pada masa akan datang.
4. Kemampuan pelabelan aliran paket
Kemampuan atau fitur baru ditambahkan pada IPv6 ini adalah memungkinkan pelabelan paket atau pengklasifikasian paket yang meminta penanganan khusus, seperti kualitas mutu layanan tertentu (*QoS*) atau *real-time*.
5. Autentifikasi dan kemampuan privasi
Kemampuan tambahan untuk mendukung autentifikasi, integritas data dan data penting juga dispesifikasikan dalam alamat IPv6.

2.3.2 Arsitektur pengalaman IP versi 6

Alamat IPv6 adalah pengidentifikasi sepanjang 128 bit untuk *interface* dan sekumpulan *interface*. Ada tiga tipe dari alamat IPv6 :

1. *Unicast* : Pengidentifikasi untuk *interface* tunggal. Paket yang dikirimkan ke alamat *unicast* adalah paket yang dikirimkan ke sebuah *interface* yang diidentifikasi oleh alamat tersebut.
2. *Anycast* : Pengidentifikasi untuk sekumpulan *interface* (umumnya milik *node* yang berbeda). Paket yang dikirimkan ke alamat *anycast* adalah paket yang dikirimkan ke salah satu dari sekumpulan *interface* yang diidentifikasi oleh alamat tersebut (alamat yang paling dekat, mengacu pada pengukuran jarak dari protokol *routing*).
3. *Multicast* : Pengidentifikasi untuk sekumpulan *interface* (umunya milik *node* yang berbeda). Paket yang dikirimkan ke alamat *multicast* adalah paket yang dikirimkan ke semua *interface* yang diidentifikasi oleh alamat tersebut.

Tabel 2-1 berikut akan menjabarkan perbedaan IPv4 dengan IPv6 secara keseluruhan.

Tabel 2-1 Perbedaan IPv4 dengan IPv6

IPv4	IPv6
Panjang alamat 32 bit (4 bytes)	Panjang alamat 128 bit (16 bytes)
Dikonfigurasi secara manual atau DHCP IPv4	Tidak harus dikonfigurasi secara manual, bias menggunakan <i>address autoconfiguration</i> .
Dukungan terhadap IPSec opsional	Dukungan terhadap IPSec dibutuhkan
Fragmentasi dilakukan oleh pengirim dan pada router, menurunkan kinerja router.	Fragmentasi dilakukan hanya oleh pengirim.
Tidak mensyaratkan ukuran paket pada link layer dan harus bisa menyusun kembali paket berukuran 576 byte.	Paket link-layer harus mendukung ukuran paket 1280 byte dan harus bisa menyusun kembali paket berukuran 1500 byte
Checksum termasuk pada <i>header</i> .	Checksum tidak masuk dalam <i>header</i> .
Header mengandung <i>option</i> .	Data opsional dimasukkan seluruhnya ke dalam <i>extensions header</i> .
Menggunakan ARP Request secara broadcast untuk menterjemahkan alamat IPv4 ke alamat link-layer.	ARP Request telah digantikan oleh Neighbor Sollicitation secara multicast.
Untuk mengelola keanggotaan grup pada subnet lokal digunakan Internet Group Management Protocol (IGMP).	IGMP telah digantikan fungsinya oleh Multicast Listener Discovery (MLD).

Saat ini IPv6 melanjutkan model IPv4 dimana *prefix subnet* diasosiasikan dengan satu *link* (*link* tunggal). *Prefix subnet* yang mungkin diberikan pada *link* yang sama dapat lebih dari satu.

2.4 Audio dan Video Streaming

Audio Streaming merupakan suatu layanan yang memungkinkan suatu server untuk mem-*broadcast* suatu audio yang bisa diakses oleh clientnya. Layanan *audio streaming* memungkinkan pengguna untuk mengakses audionya secara *real time* ataupun sudah direkam sebelumnya.

Video Streaming merupakan suatu layanan yang memungkinkan suatu server untuk mem-*broadcast* suatu video yang bisa diakses oleh clientnya. Layanan *video streaming* memungkinkan pengguna untuk mengakses videonya secara *real time* ataupun sudah direkam sebelumnya. Isi dari video ini dapat dikirimkan dengan tiga cara dibawah ini :

- ❖ Live Video → Server dilengkapi dengan web camera yang memungkinkan untuk memperlihatkan suatu kejadian secara

langsung. Walaupun hal ini dikaitkan dengan *broadcast* video, tetapi video ini ditransmisikan menggunakan protokol IP *multicast*.

- ❖ Scheduled Video → Video yang sudah direkam sebelumnya dikirimkan dari suatu server pada waktu yang sudah ditentukan. *Scheduled Video* ini juga menggunakan protocol IP *multicast*.
- ❖ Video-on-demand → Pengguna yang sudah di-authorisasi bisa mengakses video yang sudah direkam sebelumnya dari server kapan saja mereka mau melihatnya.

2.5 QoS (Quality of Service)

Quality of Service (QoS), sebagaimana dijelaskan dalam rekomendasi CCITT E.800 adalah :

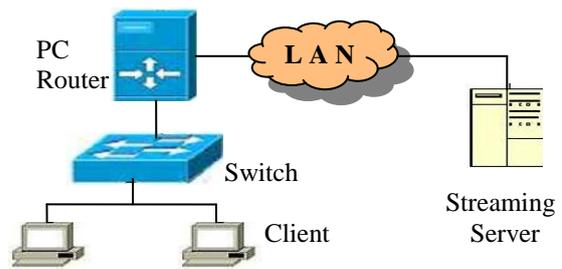
“Efek kolektif dari kinerja layanan yang menentukan derajat kepuasan seorang pengguna terhadap suatu layanan”.

Jika dilihat dari ketersediaan suatu jaringan, terdapat karakteristik kuantitatif yang dapat dikontrol untuk menyediakan suatu layanan dengan kualitas tertentu. Kinerja jaringan *VoIP – softswitch* dievaluasi berdasarkan parameter – parameter kualitas layanan *VoIP*, yaitu : *delay*, *jitter*, *packet loss* dan *throughput*. Berikut ini adalah definisi singkat dari keempat parameter layanan tersebut.

1. Jitter
Merupakan variasi *delay* yang terjadi akibat adanya selisih waktu atau interval antar kedatangan paket di penerima.
2. Delay
 - a. Waktu yang dibutuhkan untuk mengirim data dari sumber (pengirim) ke tujuan (penerima).
 - b. *Delay* maksimum yang direkomendasikan oleh ITU untuk aplikasi suara adalah **150 ms**, dan yang masih bisa diterima pengguna adalah **250 ms**.
3. Packet Loss
Kehilangan paket ketika terjadi *peak load* dan *congestion* (kemacetan transmisi paket akibat padatnya *traffic* yang harus dilayani) dalam batas waktu tertentu.
4. Throughput
Aspek utama *throughput* yaitu berkisar pada ketersediaan *bandwidth* yang cukup untuk suatu aplikasi. Hal ini menentukan besarnya trafik yang dapat diperoleh aplikasi saat melewati jaringan. Aspek penting lainnya adalah *error* (pada umumnya berhubungan dengan *error rate*) dan *losses* (pada umumnya berhubungan dengan kapasitas *buffer*).

III. PERANCANGAN SISTEM

Arsitektur perancangan sistem dari pembatasan bandwidth pada jaringan IPv6 dapat ditunjukkan seperti gambar 3.1 berikut :



Gambar 3.1 Topologi Jaringan

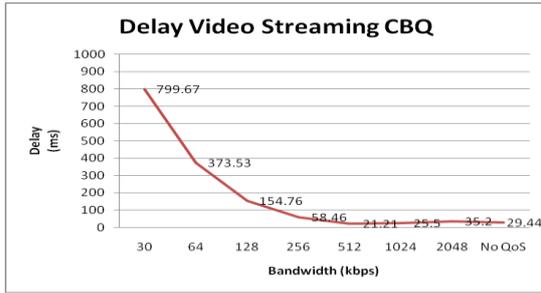
Cara kerja pada perancangan sistem diatas adalah diawali dari instalasi aplikasi pembatasan bandwidth (CBQ dan HTB) beserta aplikasi penunjang yang digunakan untuk melakukan pengamatan parameter unjuk kerja dari jaringan ketika dilakukan pembatasan *bandwidth*, diantaranya pengamatan *throughput*, *jitter*, paket *loss* serta performansi jaringan. Kemudian, dilakukan perancangan sistem pengalamatan jaringan yang berbasis IPv6, dan diimplementasikan pada jaringan yang selanjutnya dilakukan pengujian pembatasan *bandwidth* pada jaringan menggunakan aplikasi CBQ dan HTB. Dari pengujian tersebut, dilakukan pengamatan hasil pengujian pembatasan *bandwidth* berdasarkan parameter yang telah disebutkan, dan digunakan sebagai bahan analisis berdasarkan hasil pengujian pada masing-masing aplikasi, sehingga dapat diketahui perbandingan kinerja pembatasan bandwidth dari aplikasi CBQ dan HTB ketika menggunakan pengalaman jaringan yang berbasis IPv6.

IV. PENGUJIAN DAN ANALISA

Dalam penelitian ini, semua jaringan dibebankan trafik dengan menggunakan aplikasi manajemen bandwidth yaitu CBQ dan HTB tools secara bergantian. Untuk video yang akan distreamingkan adalah dengan format avi, video codec yang digunakan adalah standar MPEG4 dengan bitrate 1024 kbps. Dalam pengambilan data, menggunakan durasi full (sampai habis) untuk setiap sample bandwidth pada video streaming.

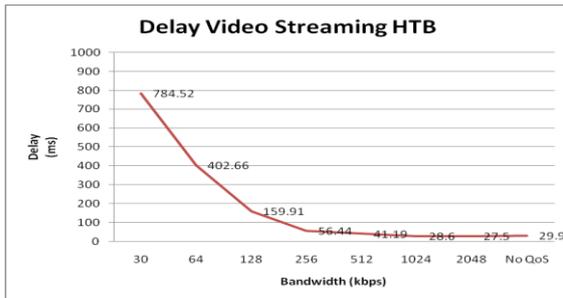
4.1 Delay dan Jitter

Jitter merupakan variasi delay antar paket yang terjadi pada jaringan IP. Besarnya nilai jitter akan sangat dipengaruhi oleh variasi beban trafik dan besarnya tumpukan antar paket (*congestion*) pada jaringan. Semakin besar beban trafik, semakin besar pula peluang terjadinya *congestion* dengan demikian nilai jitter-nya akan semakin besar. Untuk pengukuran delay dan jitter menggunakan software wireshark dengan durasi peng-capturan sampai video selesai di-stream untuk setiap sample bandwidth.



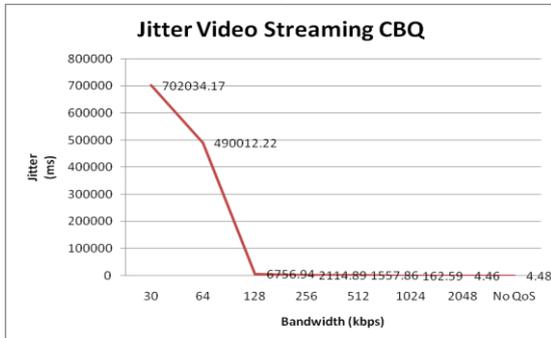
Gambar 4.1 Grafik delay video streaming CBQ

Nilai delay pada video streaming dengan CBQ semakin menurun bila nilai bandwidth-nya ditingkatkan. Sehingga nilai QoS semakin rendah pula. Berturut-turut nilai delay ini adalah 799.67 ms pada bandwidth 30 kbps, 373.53 ms pada bandwidth 64 kbps, 154.76 ms pada bandwidth 128 kbps, dan seterusnya mencapai dibawah 50 ms untuk bandwidth diatas 256 kbps. Sedangkan pada HTB, semakin besar bandwidth, maka delay yang terjadi semakin menurun. Sehingga buffer yang terjadi semakin berkurang, dan streaming dapat dilakukan dengan lancar pada bandwidth diatas 512 kbps.

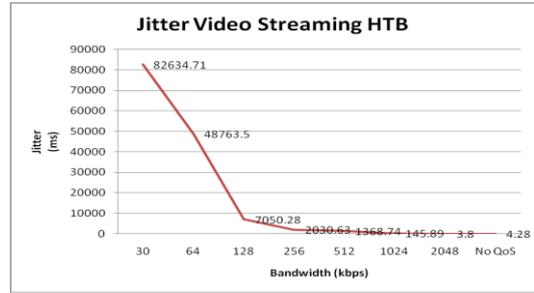


Gambar 4.2 Grafik delay video streaming HTB

Jitter yang didapatkan pada pengukuran ini sangat kecil. Hal ini tergantung dari banyaknya buffer yang menumpuk di jaringan. Semakin banyak buffer yang menumpuk, semakin besar pula jitter yang dihasilkan. Namun, secara keseluruhan, hal tersebut tidak berpengaruh pada kualitas streaming, karena dari hasil yang didapatkan, masih adanya nilai paket loss, delay, dan throughput yang wajar, serta nilai bandwidth yang diberikan.



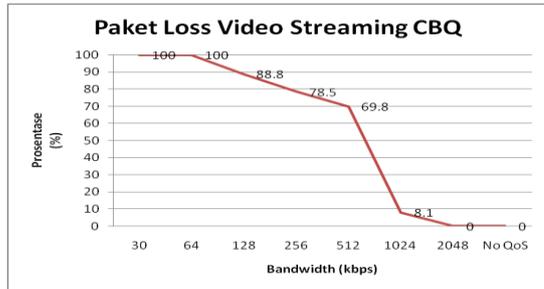
Gambar 4.3 Grafik jitter video Streaming CBQ



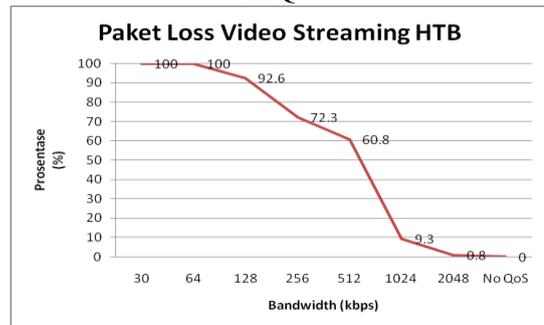
Gambar 4.4 Grafik jitter video Streaming HTB

4.2 Paket Loss

Pada gambar terlihat bahwa yang memiliki nilai paket loss terbesar yaitu pada saat pengukuran dengan bandwidth 30 kbps dengan nilai mencapai 100%, sedangkan untuk pengukuran dengan bandwidth 512 kbps, 1Mbps, 2 Mbps dan tanpa pembatasan bandwidth, nilai paket loss dapat ditekan hingga mencapai 0%. Menurut ITU-T G.1010 paket loss yang diperbolehkan harus tidak lebih dari 1% loss, jadi untuk sistem manajemen bandwidth CBQ untuk HTB



Gambar 4.5 Grafik paket loss Video Streaming CBQ

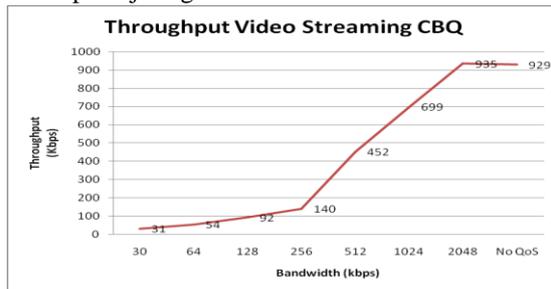


Gambar 4.6 Grafik paket loss video Streaming HTB

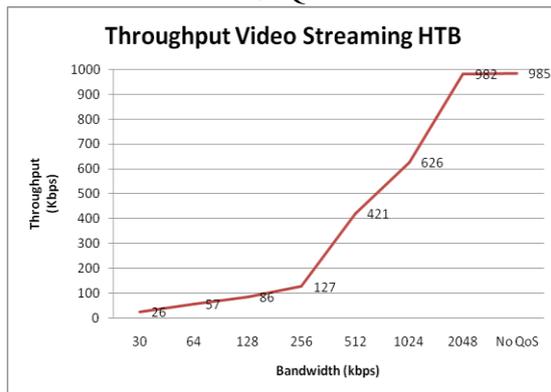
4.3 Throughput

Throughput adalah kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data. Biasanya throughput dikaitkan dengan bandwidth, dikarenakan throughput bisa disebut dengan bandwidth dalam kondisi yang sebenarnya. Sementara throughput sifatnya adalah dinamis, tergantung trafik yang sedang terjadi. Semakin besar bitrate maka akan semakin besar throughputnya. Semakin besar nilai

throughputnya menunjukkan semakin bagus pula kemampuan jaringan dalam mentransmisikan file.



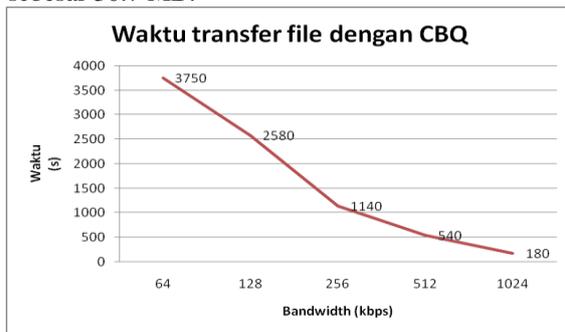
Gambar 4.7 Grafik throughput video streaming CBQ



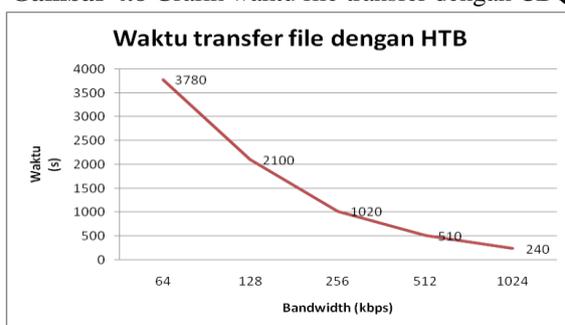
Gambar 4.8 Grafik throughput video streaming HTB

4.4 File Transfer

Dalam pengamatan ini, dilakukan transfer file menggunakan protocol ftp. File yang ditransfer dalam percobaan ini adalah file video dengan format *.flv, sedangkan ukuran file yang ditransfer adalah sebesar 30.7 MB.



Gambar 4.8 Grafik waktu file transfer dengan CBQ



Gambar 4.8 Grafik waktu file transfer dengan HTB

V. KESIMPULAN

Setelah dilakukan pengujian dan analisa pada sistem jaringan yang berbasis IPv6, maka dapat disimpulkan sebagai berikut :

1. Secara keseluruhan, performa jaringan yang berbasis IPv6 jauh lebih baik dan lebih stabil dibandingkan dengan jaringan yang berbasis IPv4. Dimana pada IPv6 mendukung penyusunan address secara terstruktur dan menyediakan kemampuan routing baru yang tidak terdapat pada sistem pengalaman IPv4, serta dapat dilakukan pemilihan route secara efisien.
2. Dari sample data video streaming yang diambil, yaitu sebesar 256 kbps, 512 kbps dan 1Mbps, yang paling baik nilai QoS-nya adalah pada bandwidth diatas 512 kbps dengan pengiriman video streaming yang nilai bitratennya adalah 1024 kbps.
3. Aplikasi bandwidth management HTB tidak bisa support dengan sistem pengalaman IPv6. Sehingga banyak konfigurasi yang tidak bisa dijalankan ketika HTB bekerja pada sistem pengalaman IPv6.
4. Adanya buffer yang besar mempengaruhi nilai jitter yang dihasilkan. Semakin besar buffer yang terjadi, semakin besar pula nilai jitter-nya.

VI. DAFTAR PUSTAKA

- [1] Abas Ali P. "Analisis Perbandingan HTB (Hierarchical Token Bucket) dan CBQ (Class Based Queuing) Untuk Mengatur Bandwidth Menggunakan Linux", STMIK AMIKOM Yogyakarta.
- [2] Nur Hayati. "Analisa Kualitas Multimedia Pada Jaringan Mobile IP versi 6", PENS-ITS Surabaya. 2010.
- [3] Hagino. "IPv6 Network Programming", Elsevier Digital Press United Kingdom. 2005.
- [4] M. Levinson. "Bandwidth Trailblazers", CIO Magazine, March 1, 2002.
- [5] Andi Rusiawan. "Dasar IPv6 : Fitur IPv6", Januari 2007.
- [6] M. Zen Samsono H. "Performance & Monitoring Network", PENS-ITS Surabaya.