08 Jurnal Sains Terapan Vol. 11 (1) : 88 – 103 (2021)
DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232

MANAJEMEN TRAFIK MENGGUNAKAN HTB UNTUK MENINGKATKAN KUALITAS LAYANAN IP NETWORK

(Traffic Management Using HTB to Improve IP Network Service Quality)

Bayu Widodo[1]
[1]Sekolah Vokasi, Institut Pertanian Bogor, Jl. Kumbang No. 14 Bogor, Jawa Barat
E-mail **: bayuwi@aps.ipb.ac.id**

Diterima 2 Juli 2021/Disetujui 29 Agustus 2021

ABSTRACT

In IP-based networks with many users, a bandwidth usage regulation mechanism is needed to ensure service quality. Mikrotik RouterOS has a feature to run QoS known as queue. RouterOS uses the Hierarchichal Token Bucket (HTB) method as the main queue method. The results of the tests carried out show that HTB is able to provide traffic limitation facilities for each classification and guarantee or ensure the amount of bandwidth is less or equal to the predetermined amount. If there is unused bandwidth it can be used by a lower classification (borrowing mechanism).

Key Words: Hierarchichal Token Bucket, Queue Tree, Firewall Mangle

PENDAHULUAN

Saat ini, *Internet, World Wide Web*, dan *Information Super Highway* adalah istilah yang tidak asing lagi bagi jutaan orang di seluruh dunia. Transmission Control Protocol/ Internet Protocol (TCP/IP) adalah rangkaian protokol yang dikembangkan untuk Internet. TCP/IP dikembangkan dengan tidak tergantung dengan sistem operasi atau perangkat keras tertentu dan menjadi standar de facto jaringan komputer global. Terlepas dari aplikasi atau layanan non *real-time* (tradisional) seperti WWW, e-mail, File Transfer Protocol (FTP), maupun layanan yang bersifat *real-time* seperti layanan VoIP, TCP/IP telah mengubah paradigma hidup (Rodriguez et al. n.d.) (Nayak, Rai, and Mall 2016) (Hunt 1992).

Perkembangan berbagai jenis layanan ini tidak akan berhasil tanpa dukungan teknologi berbasis protokol Internet (Internet Protocol/IP) dengan kapasitas dan data *rate* (*bandwidth*) yang mampu menangani trafik dalam jumlah yang sangat besar. Interkoneksi pada jaringan IP dibedakan interkoneksi pada *managed network* dan *interkoneksi* pada *unmanaged network*. Interkoneksi pada *unmanaged network* atau disebut juga sebagai best-effort merupakan *interkoneksi* antara jaringan dengan layanan aplikasi (konten) tanpa ada pengendalian trafik dan kualitas layanan.

Pada perkembangannya untuk aplikasi/ layanan tertentu interkoneksi yang bersifat *Best-Effort* tidaklah cukup. Untuk memenuhi kebutuhan-kebutuhan layanan yang berbeda, yang menggunakan infrastruktur yang sama perlu ada mekanisme lain yang disebut sebagai *Quality of Service* (QoS). QoS merupakan

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232

suatu tantangan yang besar dalam jaringan berbasis IP dan internet secara keseluruhan. Fitur yang dimiliki QoS menjadikan *bandwidth, jitter, latency* dapat diprediksi dan dicocokkan dengan kebutuhan aplikasi yang digunakan di dalam jaringan (Rodriguez et al. n.d.) (Hunt 1992).

Salah satu metode implementasi QoS pada jaringan adalah dengan pengendalian trafik jaringan atau dengan melakukan manajemen *bandwidth*. Manajemen bandwidth dimaksudkan agar *bandwidth* sebagai sumber daya terbatas dapat digunakan oleh pengguna secara adil dan tepat guna. RouterOS sebagai sistem operasi berbasis Linux yang diperuntukkan sebagai network *router* telah dilengkapi dengan berbagai fitur queue yang dapat melakukan pengaturan alokasi *bandwidth* bagi setiap user (Balan and Potorac 2009) (Hidayat 2018).

Fokus penelitian ini adalah membangun adaptive manajemen *bandwidth* HTB yang diimplementasikan pada Mikrotik RouterOS yang berperan sebagai *Internet* gateway diantara jaringan lokal dan jaringan Internet (publik). *Router* akan mengatur lalu lintas Internet baik yang masuk maupun keluar jaringan lokal dan sekaligus mengalokasikan *bandwidth* untuk setiap user atau kelompok user yang ada di jaringan lokal, memberikan level layanan sesuai kebutuhan dan prioritas sesuai kebutuhan. Pengalokasian *bandwidth* yang tepat dapat menjadi salah satu metode dalam memberikan jaminan kualitas suatu layanan jaringan (Balan and Potorac 2009) (Manual n.d.) (Lee and Kim 2013) .

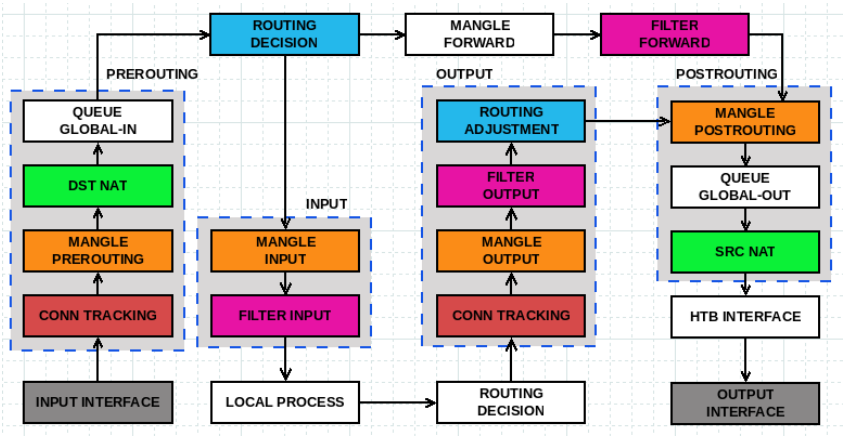
Hasil dari penelitian ini adalah suatu mekanisme pengaturan *bandwidth* dengan tujuan mencegah terjadinya monopoli penggunaan *bandwidth* atau terjadi pemakaian bandwidth secara berlebihan oleh satu atau beberapa user dan memberikan jaminan alokasi *bandwidth* minimum kepada setiap user di dalam jaringan.

STUDI LITERATUR

Aliran Paket Data

Jaringan komputer atau jaringan telekomunikasi biasanya didefinisikan sebagai interkoneksi dua komputer *autonomous* atau lebih untuk keperluan komunikasi data. Setiap komputer yang terhubung dengan jaringan disebut *node*. Jaringan lokal dan Internet sebagian besar menggunakan TCP/IP sebagai protokolnya. Protokol IP merupakan salah satu protokol kunci di dalam kumpulan protocol TCP/IP. TCP dan IP merupakan jantung protokol Internet. TCP bekerjasama dengan IP mengirimkan data antar komputer melintasi jaringan atau Internet (Rodriguez et al. n.d.) (Hunt 1992) (M 2001) (Cip et al. 2019).

Informasi yang dikirimkan dari komputer ke Internet dan sebaliknya akan diolah oleh protokol TCP/IP menjadi paket-paket yang lebih kecil. Paket akan beredar di jaringan lokal ataupun Internet melewati beberapa *router*. *Router* sebagai *intermediary device* akan menjalankan fungsi routing dan fungsi menentukan paket mana yang akan lebih dahulu dikirimkan dan paket mana yang harus berada di dalam antrian. Paket yang melewati *router* adalah paket paket yang dibangkitkan oleh komputer user (klien). Istilah *Packet flow* merupakan sebutan alur proses paket yang keluar dan masuk melewati router (Smansub et al. 2019) (Nayak, Rai, and Mall 2016) (Burgess 2011) .



Gambar 1 Paket Flow

Informasi-informasi yang dikirim dari komputer klien ke internet maupun sebaliknya akan diolah oleh protocol TCP/IP menjadi paket. Pada Gambar 1 memperlihatkan bagaimana paket-paket data masuk dari *interface input*, melewati, atau keluar dari *Mikrotik RouterOS* (*interface output*). Paket upload ke internet, trafik *upload* yang berasal dari LAN akan melewati router. Di dalam router, chain *Firewall* yang akan dilewati adalah *chain* prerouting, *chain* forward dan *chain* postrouting. *Chain prerouting* berguna untuk melakukan *marking* terhadap paket yang akan melintasi router ataupun paket yang akan menuju router. *Chain Forward* hanya akan melakukan marking pada paket yang melintasi paket dan bukan terhadap paket yang menuju router.

Chain Postrouting pada *Firewall Mangle* digunakan untuk melakukan *marking* terhadap paket yang melintasi *router* maupun paket yang berasal dari router. Jika dilihat pada Gambar 1 terlihat bahwa ada 2 anak panah yang masuk ke dalam *chain postrouting*. Jika *router* memiliki 2 (dua) *interface* di mana *interface* pertama (eth0) terhubung ke Internet dan *interface* kedua (eth1) terhubung ke jaringan lokal, maka hasil akhir yang didapat dari konfigurasi pada Gambar 1, adalah 2 (dua) jenis paket, masing-masing paket bisa diindentifikasi sebagai *client upload* dan *client download*. (Marcel 2018) (Nayak, Rai, and Mall 2016) (Burgess 2011)

Antrian (*Queue*)

Trafik jaringan berhubungan dengan paket data yang dibangkitkan oleh kartu ethernet/ *interface* perangkat jaringan. Paket paket dikirimkan dari asal ke tujuan melewati *link* (media) dengan kapasistas (*bandwidth*) tertentu. Untuk jaringan lokal masalah *bandwidth* (kapasitas link) tidak terlalu menjadi masalah, karena *bandwidth* lebih ditentukan oleh kapasitas perangkat router, switch dan kabel yang digunakan.

Masalah akan muncul saat jaringan terhubung ke jaringan publik (Internet), karena umumnya besar *bandwidth* yang berasal dari Internet sangat terbatas. Jika *bandwidth* yang tersedia tidak cukup untuk mengalirkan paket-paket data atau jumlah paket yang dikirim lebih banyak dibandingkan dengan kemampuan link untuk menampung atau menyalurkan paket maka akan timbul masalah *congesti*

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232

atau kejenuhan. Kondisi trafik semacam ini akan mengakibatkan perfomansi jaringan menjadi menurun, apalagi jika jumlah *node* yang ada di dalam jaringan sangat banyak.

Cara terbaik untuk mengatasi kongesti trafik adalah dengan meningkatkan kapasitas *bandwidth* jaringan. Namun, cara ini tidak praktis dalam hal biaya operasi dan pemeliharaan. Cara mengurangi penurunan performansi jaringan tanpa perlu melakukan penambahan biaya atau kapasitas bandwidth adalah dengan menerapkan manajemen *bandwidth* yakni memaksa paket-paket tersebut masuk ke dalam antrian (*queue*). Dengan kata lain, interface jaringan harus memiliki kemampuan untuk *ingress queue* (*inbond*) dan *engress queue* (*outbond*). *Ingress queue* akan menyimpan paket untuk sementara waktu sebelum paket tersebut diserahkan ke *kernel* untuk diproses dan selanjutnya diserahkan ke *interface* yang digunakan untuk mengirimkan paket tersebut. Sebelum paket dikeluarkan dari *interface* output, paket tersebut terlebih dahulu disimpan di dalam *egress queue*. (Kuperman, Mikityanskiy, and Efraim n.d.) (Iswadi, Adriman, and Munadi 2019) (Lee and Kim 2013)

Queue pada perangkat jaringan dibedakan antara *hardware queue* dan *software queue*. Hardware queue sangat bergantung dengan kapasitas *bandwidth interface*. *Hardware queue* dikenal sebagai *Transmit Queue(TxQ)* atau *TX-ring*. *Hardware queue* terjadi di sisi *output interface* sesaat sebelum paket dikirimkan ke jaringan. *Software queue* terjadi atau dibutuhkan jika hardware queue tidak mampu menangani kejenuhan jaringan (*congesti*). *Software queue* lebih rumit dibandingkan dengan *hardware queue*. *Hardware queue* menggunakan metode lebih sederhana dengan hanya menggunakan metode FIFO (*First IN First Out*), sedangkan software queue dapat diterapkan berbagai metode seperti FIFO,WRR, HTB, CBWFQ, RED (Balan and Potorac 2009) (Iswadi, Adriman, and Munadi 2019).

Paket Marking

Mangle merupakan salah satu fitur pada firewall Mikrotik RouterOS yang digunakan untuk menandai paket. Pemberian label atau nama kepada setiap paket akan memudahkan *router* untuk mengelola paket tersebut. *Mangle* mengenal 3 (tiga) jenis *marking*, yaitu *Connection Mark*, *Packet Mark* dan *Route Mark*. Dalam jaringan komputer, bisa saja terjadi komputer klien membuat koneksi secara bersama-sama, sehingga *router* harus menandai paket-paket yang datang. Tahap pertama *router* melakukan marking bagi semua jenis koneksi yang dibuat oleh semua komputer klien, kemudian *router* akan melakukan marking terhadap keseluruhan trafik *upload* maupun trafik *download*. Pada saat melakukan konfigurasi packet mark pada trafik upload, konfigurasi harus berisi parameter *connection-mark=client-A_CONN* dan parameter *in-interface=ether* yang disesuaikan dengan interface mana semua trafik upload akan masuk. Sedangkan untuk melakukan marking terhadap paket *download* parameter *connection-mark* yang harus digunakan adalah *all-clinet_CONN* dan *in- interface=ether* dimana semua trafik download dari Internet akan masuk melalui *interface* tersebut (Manual n.d.) (Burgess 2011).

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232

Queue Pada Mikrotik RouterOS

Manajemen *bandwidth* merupakan suatu istilah yang ditujukan pada suatu subsistem antrian paket pada suatu jaringan atau *network devices*. RouterOS sebagai sistem operasi *router* sudah dilengkapi dengan fitur-fitur untuk menjalankan QoS atau lebih dikenal dengan manajemen *bandwidth*. Ada 2 cara (metode) RouterOS melakukan fungsi manajemen *bandwidth* terhadap paket yang diterimanya yaitu: *Queue Simple* dan *queue tree*. Kedua metode tersebut sama sama memanfaatkan memory RAM di *router* sebagai *bufer* penampuangan antrian paket. Bufer memiliki batas kapasitas, jika antrian paket telah memenuhi buffer maka paket yang tidak tertampung akan di *Drop*. Jika aplikasi menggunakan protokol TCP, maka paket yang telah didrop akan dikrim ulang (Hardiman 2018) (Ivancic, Hadjina, and Basch 2005) (M 2001).

Queue Simple merupakan cara termudah untuk melakukan pengaturan *bandwidth* seperti membatasi pemakaian *bandwidth upload* dan *download* setiap user. *Simple queue* dapat melimit *upload*, *download* atau total (*upload*+*download*) sekaligus dalam 1 *rule* (Manual n.d.).

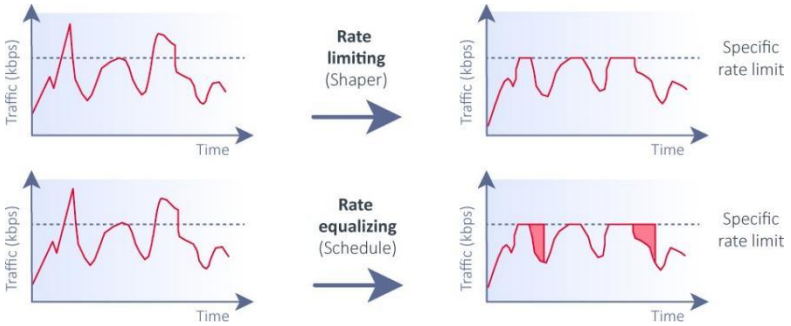
Queue tree merupakan penerapan queue yang lebih rumit dibandingkan dengan *queue simple*. *Queue tre*e dapat melakukan pembatasan *bandwidth* berdasarkan grup atau kelas tertentu, melakukan skala prioritas bahkan dapat melakukan pembatasan *bandwidth* secara hirarki. Untuk menggunakan metode antrian *queue tree* ini, harus melakukan *classification packet* dan *marking packet* dengan menggunakan fitur *Mangle* pada *Firewall*. *Queue tree* tidak dapat berjalan tanpa *Mangle*. Batas maksimum dan minimum alokasi *bandwidth* yang diterima oleh user diatur oleh parameter CIR dan MIR.

Metoda metoda *queue* pada RouterOS baik *simple queue* maupun *queue tree* pada intinya bekerja untuk mengendalikan atau membatasi *rate* dari *packet flow* yang diterima ataupun paket yang akan dikirim melalui *interface router*.

Mikrotk routerOS mempunyai dua cara untuk membatasi *rate* dari aliran paket, yaitu *rate limiting* (*dropper/ shaper*) dan *rate equalizing* (*scheduller*). R*ate limiting* atau *shaper* adalah prinsip pembatasan atau pengaturan *rate* dengan cara membuang paket paket jika paket yang akan diproses lebih besar dari rate yang telah ditentukan. Sedangkan *rate equalizing* atau *scheduler* akan menyimpan paket paket yang melebihi *rate* yang telah ditentukan untuk sementara waktu di dalam *queue* dan akan dikirim jika kondisi memungkinkan.

Rate scheduller terdiri dari metode *queue* FIFO (Bytes FIFO, Packets FIFO, Multi Queue Packets FIFO), RED dan SFQ. Sedangkan *Rate Shaper* teridiri dari metode queue PCQ dan HTB. Gambar 2 berikut memperlihatkan bagaimana *bandwidth shaper* dan *Bandwidth scheduler* bekerja untuk membatasi kelebihan paket dari *rate* yang sudah ditentukan.

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232



Gambar 2 Rate Scheduller dan Rate Shaper
Sumber : www.help.Mikrotik.com

Terlihat pada Gambar 2, dalam kasus pertama, semua trafik melebihi *rate* yang telah ditentukan dan dihentikan. Dalam kasus lain, trafik melebihi *rate* tertentu dan tertunda dalam *queue* dan akan dikirimkan kemudian jika memungkinkan, tetapi paket hanya dapat ditunda jika antrian tidak penuh. Jika tidak ada lagi ruang dalam buffer antrian, paket-paket akan dijatuhkan (*drop*).

Hierarchichal Token Bucket

Implementasi manajemen *bandwidth* di *Mikrotik RouterOS* banyak bergantung pada sistem HTB (*Hierarchical Token Bucket*) sebab *queue* yang dibuat di routerOS disusun berdasarkan prinsip HTB. HTB (*Hierarchical Token Bucket*) merupakan metode antrian *classfull* yang mampu menangani berbagai jenis trafik. Teknik antrian HTB akan menghasilkan struktur *queue* dengan bentuk hirarki (bertingkat) dan mengatur hubungan antar kelas-kelas hirarki dalam bentuk "*parent-child*" atau "*child-child*" sehingga metode ini sangat memudahkan dalam pengalokasisan *bandwidth* (Kuperman, Mikityanskiy, and Efraim n.d.) (Jia et al. 2018).

Inner-class adalah kelas yang memiliki *parent-child*. *Inner queue* minimal akan membawahi 1 (satu) *leaf* dan dapat juga *queue* yang memiliki *parent*. Dengan kata lain, queue yang memiliki parent dan child dapat dikategorikan sebagai *inner queue*. *Leaf queue* membuat konsumsi trafik aktual, *inner queue* hanya bertanggung jawab untuk distribusi trafik. Di RouterOS perlu menentukan opsi *parent* untuk menetapkan antrian sebagai *child* ke antrian lain.

Ketika trafik telah diklasifikasikan, trafik kemudian dijadwalkan (*scheduled*) and dibentuk (*shaped*) kemudian HTB akan melakukan tugas-tugas tersebut dengan menggunakan konsep *token*. Token dihasilkan oleh *token generator* dengan durasi (*interval*) yang tetap. Token-token kemudian dikumpulkan dalam bucket untuk mengontrol penggunaan *bandwidth* dalam *link*.

Pada Mikrotik RouterOS V5, queue dapat diterapkan pada 4 (empat) bagian yang akan dilewati paket, yaitu: *Global-IN* (*ingress queue*), *Global-Out* (*engress queue*), *Global-Total* dan *Outgoing interface* (*output interface*). Tiga *interface* yang pertama merupakan *interface* virtual sedangkan *Outgoing interface* merupakan interface fisik. Pemilihan *interface* mana yang akan dipilih untuk tempat melakukan queue apakah *interface* virtual ataupun *interface* fisik tergantung dari skenario jaringan yang dihadapi (Burgess 2011).

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232

Setiap antrian HTB memiliki *dua rate* (*dual limitation*), yaitu MIR (*max-limit*) dan CIR (*limit-at*). *Maximum Information Rate* (MIR) adalah batas maksimum *alokasi bandwidth* yang dapat diterima oleh user. Beberapa literatur menyebut parameter MIR sebagai *Peak Information Rate* (PIR). Sedangkan *alokasi bandwidt*h minimum yang masih dapat diterima oleh user ditunjukkan oleh *Committed Information Rate* (CIR). Seburuk apapun kondisi jaringan, komputer user tidak akan pernah mendapatkan *alokasi bandwidth* yang lebih rendah dari angka CIR (jaminan mendapatkan *bandwidth* sebesar *limit-at*) (Ivancic, Hadjina, and Basch 2005).

Parameter *max-limit* dan *limit-at* perlu dikonfigurasikan pada *queue* untuk memberikan nilai CIR dan MIR kepada user. Parameter *limit-at* hanya akan berfungsi jika konfigurasi *queue* dalam bentuk hirarki. Konfigurasi *queue* dalam bentuk hierarki yang dimaksud adalah konfigurasi *queue* yang bertindak sebagai *parent* (*inner-queue*) dan ada quieue yang bertindak sebagai *child* (*leaf-queue*) (Ivancic, Hadjina, and Basch 2005) (Lee and Kim 2013) (Balan and Potorac 2009) .

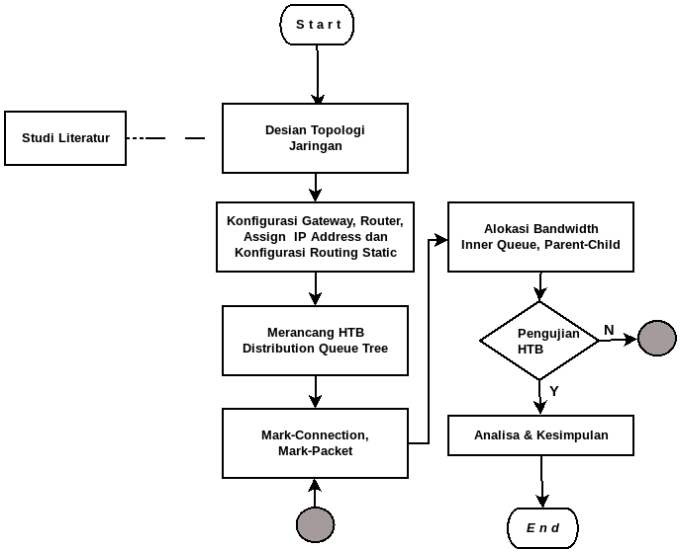
Fitur-fitur yang dimiliki HTB antara lain:

1. Hirarki
   - Hampir tidak ada limit hirarki.
2. Grup
   - Beberapa klien dapat dikelompokkan dalam satu parent
   - Klien tertentu dapat meminjam *bandwidth* dari klien yang lain dalam grup yang sama, jika dibutuhkan
3. Setiap *leaf* dapat memiliki pengaturan yang berbeda.

METODE PENELITIAN

Metode yang dilakukan dalam penelitian ini adalah metode eksperimen yaitu dengan mensimulasikan sistem operasi jaringan RouterOS, merencanakan dan merancang *alokasi bandwidth* klien sesuai dengan *rule queue* yang ditetapkan. Tahapan-tahapan yang dilakukan dalam penelitian ini dapat di lihat pada gambar 3. Tahapan tersebut digunakan sebagai sebagai pedoman dalam pelaksanaan penelitian.

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232



Gambar 3 Metode Penelitian

Strategi awal yang dilakukan adalah dengan melakukan studi pustaka dengan cara mempelajari jurnal terkait dan sumber buku yang berkaitan untuk dikutip sebagai acuan teori dalam mendukung penulisan penelitian ini. Tahapan selanjutnya adalah merancang topologi jaringan dan mensimulasikannya di mesin virtual. Pada tahap selanjutnya melakukan konfigurasi dan test perubahan konfigurasi jaringan terhadap beberapa parameter HTB untuk melakukan klasifikasi, pembatasan trafik dan penentuan prioritas kelas yang akan diutamakan oleh HTB. (Marcel 2018)

Topologi Simulasi Jaringan yang digunakan dalam implementasi *Hierarchy Token Bucket* tampak seperti pada Gambar 4 berikut :

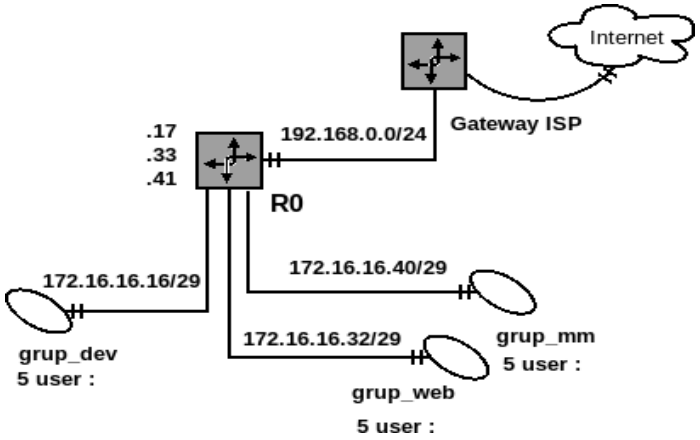
a. Simulasi yang dirancang terdiri dari 1 (satu) buah *router* Mikrotik sebagai *gateway* yang ditempatkan diantara jaringan lokal (LAN) dan jaringan Internet. Jaringan lokal terdiri dari 3 (tiga) buah *subnet* dan mendapatkan akses Internet dari *router* Mikrotik. *Router* Mikrotik selain berperan sebagai *gateway* juga berperan mengatur trafik Internet baik yang masuk maupun keluar dari *subnet* tersebut. *router* Mikrotik akan membagi koneksi Internet (*bandwidth*) yang didapatkan dari ISP yang kemudian dibagi ke beberapa komputer *user* yang ada di jaringan lokal.
b. Simulasi jaringan diarahkan untuk menjawab beberapa hal yang menjadi latar belakang dan tujuan yang telah disebutkan sebelumnya. Ketiga *subnet* masing- masing mewakili grup developer, gru web dan grup mutimedia. Grup developer (grup_dev) merupakan grup dengan subnet /16 dalam simulasi penelitian ini disebut grup16. Grup web merupakan grup dengan subnet /32 disingkat dengan gerup_web dan simulasi penelitian disebut grup32. Grup multimedia yang merupakan grup dengan subnet /40 disingkat dengan gruo_mm dan dalam penelitian disebut grup40.

Pengelompokkan user ke dalam grup yang berbeda didasarkan pada kebutuhan *bandwidth* yang berbeda pada setiap aplikasi dan transaksi bisnis yang dilakukannya. Secara umum transaksi bisnis setiap grup melalui Internet dengan

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232

layanan http (web), layanan email (smtp dan pop3), ftp dan layanan multimedia.

Gambar 4 berikut menunjukkan pembagian grup, pembagian *subnet* dan jumlah *user* tiap *subnet*.



Gambar 4 Topologi Jaringan

1. *Subnet* 172.16.16.16/29 adalah kelompok user dari grup_dev, *subnet* 172.16.16.32/29 adalah kelompok user grup_web, dan *subnet* 172.16.16.40/28 merupakan *subnet* dari *user* yang tergabung dalam kelompok grup_mm.
2. R0 merupakan satu satunya jalur menuju jaringan publik (Internet) dengan 4 (empat) buah *interface*. R0 berperan sebagai default gateway dengan menggunakan mode NAT. *Interface* eth0 mendapatkan IP Address secara dinamik dari ISP sedangkan *interface* lainnya menghadap ke *router* yang melayani *subnet* jaringan lokal.
3. Perlakuan manajemen *bandwidth* hanya membedakan trafik *download* (*downstream*) untuk setiap grup sesuai dengan konfigurasi *queue tree*.

Berikut ini adalah tabel detail konfigurasi Mode Network VBox dari skema jaringan sebagaimana terlihat pada Gambar 4.

Tabel 1 Distribusi IP Address

Router	**Deskripsi**	***eth0***	***eth1***	***eth2***	***eth3***
R0	Mode	Bridged	Int net	Int net	Int net
	Nama	-	net16	net32	0
	Network	192.168.0.0	172.16.16.16	172.16.16.32	172.16.16.40
	IP Addr	192.168.0.18	172.16.16.17	172.16.16.33	172.16.16.41
	IP	Internet	*subnet 16*	*subnet* 32	*subnet* 40
	Gateway	192.168.0.1	172.16.16.17	172.16.16.33	172.16.16.42

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232

HASIL DAN PEMBAHASAN

Router Mikrotik (R0) secara default menjalankan *hardware queue* dengan metode *pfifo*. Kemampuan *hardware queue* di setiap *interface* adalah 50 paket. Artinya selama kongesti terjadi, *hardware queue* hanya dapat menampung 50 paket. Jika di dalam *router* sudah ada 50 paket, maka paket yang baru masuk akan didrop. *Hardware queue* ditempatkan pada *output interface* sesaat sebelum paket masuk ke dalam media jaringan. HTB dibutuhkan saat *hardware queue* sudah penuh dan tidak bisa menangani kongesti.

Pada *router* Mikrotik (R0) dilakukan manajemen *bandwidth* terhadap *subnet* 16/29, *subnet* 32/29 dan *subnet* 40/29. Pada skenario ini, semua user mendapatkan *sharing* internet dari R0. Artinya R0 menjalankan *Network Address Translation* (NAT) yang berfungsi mengganti IP address pada setiap paket data yang keluar dari komputer *user* menjadi IP address yang ada di eth0 (*interface* yang menuju Internet).

DNS server yang digunakan adalah DNS server ISP dengan IP Address 192.168.0.1 dengan pilihan *allow remote request=yes* pada konfigurasinya agar R0 bertindak sebagai DNS server juga. Sehingga konfigurasi DNS pada komputer user cukup diarahkan ke *router* Mikrotik.

Alokasi *bandwidth* dari ISP sebesar 3.9 Mbps *upload* dan 24.0 Mbps *download*. *Bandwidth* 24.0 Mbps akan *dialokasikan* ke grup_dev (grup16), grup_web (grup32), dan grup_mm (grup40). Setiap grup terdiri dari 5 (lima) user sesuai dengan jumlah IP address *valid* yang ada di dalam setiap *subnet*.

Dalam penelitian ini, konfigurasi queue yang dibahas hanya untuk alokasi bandwidth download dan *interface* HTB yang akan digunakan sebagai *parent* pada konfigurasi adalah *interface* eth1, *interface* eth2, *interface* eth3. *Queue* akan dilakukan baik di *interface* eth1, *interface* eth2 maupun di *interface* eth3 atau dengan kata lain *queue* akan dijalankan di *router*.

Konfigurasi inner queue dilakukan di ketiga interface, dimana dalam kasus ini memiliki 3 (tiga) *inner queue* dan masing-masing *inner queue* memiliki 3 (tiga) *child* (*leaf*), yakni dedicated, K512, dan K256. Distribusi atau *alokasi bandwidth* masing-masing kelas dijelaskan sebagai berikut:

1. *Inner queue* grup16, grup32 dan grup40 membawahi 3 (tiga) konfigurasi *leaf queue*. Parameter *priority* 6 diberikan untuk konfigurasi *queue dedicated* baik grup16, grup32 maupun grup40.
2. *Maximum Information Rate* (MIR) merupakan parameter HTB yang merupakan nilai *maksimum bandwidth* yang bisa digunakan oleh setiap class, jika *bandwidth* melebihi *rate* maka paket data akan dipotong atau di jatuhkan (drop). *Committed Information Rate* (CIR) merupakan parameter bandwidth yang menentukan peminjaman *bandiwdth* antar *class*. Peminjaman *bandwidth* dilakukan *class* paling bawah ke kelas di atasnya. CIR merupakan nilai *bandwidth* minimum yang akan diperoleh user jika jaringan sedang sibuk.
3. Jika semua klien menggunakan *alokasi bandwidthnya*, maka masing-masing klien akan mendapatkan bandwidth seperti pada tabel berikut :

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232

Tabel 2 Strategi Alokasi Bandwidth,Mbps

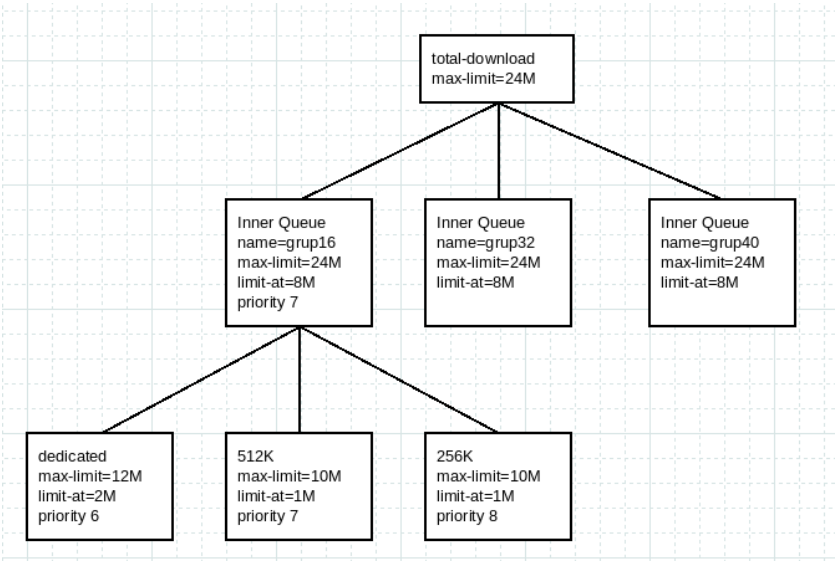
Nama Queue	*priority*	CIR	MIR
grup16	7	8,0	24,0
grup32	8	8,0	24,0
grup40	8	8,0	24,0
dedicated	6	6,0	12,0
K512	7	1,0	10,0
K256	8	1,0	10,0

Struktur *Queue Tree* HTB

Berangkat dari strategi *alokasi bandwidth* sebagaimana tabel 2 di atas, diagram hirarki *Queue* HTB untuk *downstream* digambarkan pada Gambar 5 berikut.

Strategi alokasi *bandiwdth* grup16 diberikan prioritas sedikit lebih tinggi dari *queue* grup yang lain. Konfigurasi *queue* untuk grup16 menggunakan parameter priority=7, sedangkan grup32 dan grup40 memiliki priority=8. Konfigurasi dengan priority=7 akan mendapatkan prioritas yang lebih tinggi dan akan diutamakan.

Class dedicated memiliki prioritas yang lebih tinggi jika dibandingkan dengan class 512K dan 256K. *Class dedicated* akan lebih diutamakan dibanding *queue* untuk *class* yang lain. *Class dedicated* akan mendapatkan prioritasnya setelah *class* yang lain mendapatkan alokasi CIR-nya. Skenario alokasi bandiwdth ini tidak akan mengganggu nilai CIR yang sudah dialokasikan untuk masing-masing class.



Gambar 5 Strategi Alokasi Bandwidth

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232

Marking Paket

Konfigurasi *marking paket* menggunakan parameter *src-address-list* terhadap IP Address yang telah didaftarkan pada *address list*. Berikut adalah contoh *address list* dan klien dalam kategori *address list network* 172.16.16.16/29:

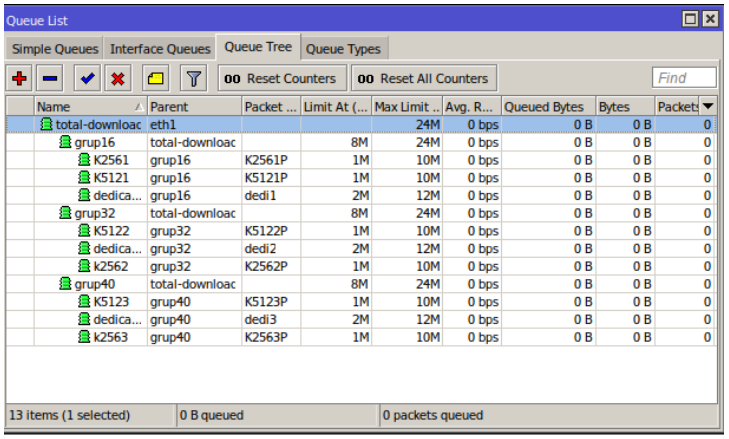
```
Flags: X - disabled, D – dynamic
#      LIST             ADDRESS
0      dedicated-1      172.16.16.18

 3  K-512-1    172.16.16.19
 9  K-256-1    172.16.16.21
```

Tahapan berikutnya adalah melakukan *marking* pada semua jenis koneksi dari kelas yang telah dibuat, tanpa melihat jenis koneksi Internet yang dilakukan menggunakan paramater *src-address-lis*t. Hasil marking koneksi untuk int*erface route*r adalah sebagai berikut (tidak dituliskan seluruhnya) :

```
Flags: X - disabled, I - invalid, D - dynamic
0      chain=prerouting action=mark-connection \
       new-connection-mark=dedi-1-con passthrough=yes \
       src-address-list=dedicated-1 in-interface=eth1
1      chain=prerouting action=mark-packet new-packet-
       mark=dedi1 \ passthrough=no connection-mark=dedi-
       1-con \
6      chain=prerouting action=mark-connection \ new-
       connection-mark=K5121 passthrough=yes \ src-
       address-list=K-512-1 in-interface=eth1
7      chain=prerouting action=mark-packet new-packet-
       mark=K5121P \ passthrough=no connection-mark=K5121
12     chain=prerouting action=mark-connection \ new-
       connection-mark=K2561 passthrough=yes src-
       address-list=K-256-1 in-interface=eth1
13     chain=prerouting action=mark-packet new-packet-
       mark=K2561P \ passthrough=no connection-mark=K2561
```

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232

Konfigurasi Queue Tree HTB



Gambar 6 Pengujian HBT Tree

Gambar 6 memperlihatkan diagram manajemen *bandwidth* untuk paket *download* untuk grup16 dengan 1 (satu) *inner queue* dan 3 (tiga) buah *leaf*. Diagram yang sama bisa digambarkan untuk grup32 dan grup40.

Dengan memperhatikan alokasi CIR pada setiap *leaf* dari ketiga *class* tersebut dapat ditentukan bahwa *leaf dedicated* dalam kondisi terburuk tetap akan mendapatkan *alokasi bandwidth* sebesar 2.0M. Dengan prioritas yang lebih tinggi dibandingkan *leaf* dalam *parent* yang sama, *leaf dedicated* berhak mengambil keseluruhan sisa *alokasi bandwidth* yang ada dari *parentnya*. Dari konfigurasi *subnet*, *leaf* K521 dan *leaf* K256 maksimal klien yang dapat dimasukkan dalam kategori ini adalah 2 (dua) klien dan setiap klien dijamin akan mendapatkan CIR 1.0 Mbps dan MIR sampai dengan 10.0 Mbps.

Pengujian

Tahap pengujian dilakukan untuk mencocokan konfigurasi limitasi *bandwidth* apakah telah bekerja dengan baik, dalam arti ada perbedaan besar *bandwidth* sesaat sebelum *queue* dijalankan di *router* dan setelah queue dijalankan di *router*. Uji konfigurasi dilakukan dengan mengukur besar *bandwidth upload* dan *bandwidth download* sebelum (pre-HTB) menggunakan *queue* dan setelah (post-HTB) menggunakan *queue*. Hasil pengukuran menggunakan aplikasi *speedtest-cli* di setiap klien pre-HTB dan post-HTB, seperti pada Tabel 3.

Tabel 3 Hasil Pengukuran Pre-HTB dan Post-HTB

No	IP Addr	Pre-HTB, Mbps (d/u)	Post-HTB, Mbps (d/u)
	Data ISP		
	Latency		0
1	Download		23,43 Mbps
	Upload		3,88 Mbps
	Packet Loss		1,4%
Test 1 Subnet 16/29			
1	172.16.16.18/16	23,56 / 3,17	10,13 / 1,67

DOI : 10.29244/jst.11.1.88 - 102
P-ISSN : 2088-8732|E-ISSN : 2722-5232

2	172.16.16.20/16	20,95 / 3,44	7,97 / 1,41
3	172.16.16.22/16	22,47 / 3,15	5,45 / 2,81
Test 2 Dedicated ketiga subnet			
1	172.16.16.19/16	22,42 / 3,4	9,93 / 2,22
2	172.16.16.35/32	21,67 / 3,76	9,95 / 3,79
3	172.16.16.42/40	22,95 / 3,34	9,23 / 1,91

Tabel 3 menunjukkan besar *bandwidth upload* dan *download* sesungguhnya (*throughput*) dari ISP. *Bandwidth download* sebesar 23,43 Mbps dan *upload* sebesar 3,88 Mbps. Dalam penelitian ini, alokasi *bandwidth upload* tidak diperhitungkan dengan asumsi *bandwidth upload* yang diberikan ISP lebih besar dari pemakaian bandwidth upload dari user yang tergabung di dalam setiap *subnet*.

Dengan memperhatikan *throughput* tabel 3, saat *router* Mikrotik tidak menjalankan HTB *queue throughput* setiap *leaf* berkisar antara 20 Mbps sampai dengan 23 Mbps. Beberapa faktor yang mempengaruhi perbedaan angka ini antara lain kecepatan modem dan mesin komputer yang digunakan, keadaan jaringan dan banyaknya pengguna yang mnegakses secara bersamaan. Dalam hal ini pengguna yang ada di dalam *network* 192.168.0.1.

Pada kolom berikutnya pada tabel yang sama, sesuai dengan strategi rancangan yang dikonfigurasi, untuk kelas 16/29 *leaf dedicated* dialokasikan MIR sebesar 12 Mbps dan CIR sebesar 2M dengan *priority* 6. Terlihat pada tabel post-HTB *leaf dedicated* angka *download* sebesar 9,93 Mbps yang lebih rendah dari MIRnya. Demikian pula dengan leaf 512 besar *download* sebesar 7,97 Mbps dan *leaf* 256 besar *bandwidth* sebesar 5,45 Mbps. Kedua angka terakhir ini ternyata masih lebih rendah dari angka MIR-nya. Jika melihat *priority leaf dedicated* sebesar 6 dan lebih tinggi dari leaf yang lain pada kelas yang sama, *alokasi bandwidth* yang diterima pada kelas yang sama ternyata lebih besar.

Keadaan berikutnya mencoba melihat perbedaan pre-HTB dan post-HTB pada leaf pada *inner queue* yang berbeda atau *leaf* pada *subnet* 16/29, 32/29, dan 40/29. Diperoleh bahwa besar *bandwidth* di ketiga *leaf* pada pre-HTB masing masing sebesar 21,42 Mpbs; 21,67 Mpbs; dan 22,95 Mbps. Dengan rancangan MIR dan CIR di setiap *leafnya* sebagaimana tampak pada Gambar 6 diperoleh *bandwidth* aktual masing-masing sebesar 9,93 Mbps (MIR 12M), 9,95 (MIR 12 Mbps) Mbps dan 9,23 (12 Mbps) Mbps. *Bandwidth* yang diperoleh lebih rendah dibandingkan dengan *max-limitnya*. Teknik HTB mampu mengatur *alokasi bandwidth* secara hirarki, dan *leaf queues* adalah pengguna *bandwidth* sesungguhnya. Sedangkan *inner queues* untuk perhitungan dan distribusi *bandwidth*.

SIMPULAN

Teknik antrian HTB memberikan fasilitas pembatasan trafik pada setiap level maupun klasifikasi namun jika terdapat *bandwidth* yang tidak terpakai bisa digunakan oleh klasifikasi yang lebih rendah. Teknik antrian HTB merupakan solusi yang efektif ketika penggunaan *bandwidth* perlu dikontrol dan dikendalikan.

Metoda HTB mampu memberikan *fleksibilitas* dalam manajemen *bandwidth*. HTB memungkinkan membuat *queue* menjadi lebih terstruktur, dengan melakukan pengelompokan-pengelompokan bertingkat. HTB memastikan jumlah layanan yang disediakan untuk tiap kelas kurang atau sama dengan jumlah yang telah ditetapkan. Ketika sebuah kelas memiliki permintaan yang kurang dari jumlah yang ditetapkan, sisa *bandwidth* akan didistribusikan ke kelas yang lain.

Parameter prioritas pada *queue tree* berfungsi memberikan prioritas yang berbeda diantara *queue* yang ada dan hanya bekerja di *leaf queue*. Jika parameter *priority* tidak ditetapkan *leaf queue* akan menggunakan *priority default* (nilai 8, *priority* terendah). Parameter *max-limit parent* haruslah lebih besar dari jumlah seluruh *limit-at childnya*.

Nilai *max-limit parent* harus lebih besar atau sama dengan jumlah dari keseluruhan *max-limit child*-nya. Jika *max-limit child* lebih besar dari *max-limit parentnya*, maka klien tidak akan pernah mencapai trafik sesuai dengan *max-limitnya*. Parameter *priority* hanya berperan sebagai pembagi *bandwidth* yang tersisa (konsep peminjaman *bandwidth* dan *priority* hanya bisa digunakan pada *leaf queue*.

DAFTAR PUSTAKA

Burgess, Dennis. 2011. Learn Routeros - Second Edition.

Cip, Cybertraining et al. 2019. Lab 20 : Classifying TCP Traffic Using Hierarchical Token Bucket (HTB).

D. G. Balan and D. A. Potorac, "Linux HTB queuing discipline implementations," 2009 First International Conference on Networked Digital Technologies, 2009, pp. 122-126, doi: 10.1109/NDT.2009.5272182..

Hardiman, et al. 2018. “Analisis Perbandingan QoS (Quality Of Service) Pada Manajemen Bandwidth Dengan Metode PCQ (Per Connection Queue) Dan HTB (Hierarchical Token Bucket).” semanTIK 4(1): 121–28.

Hidayat, Arif. 2018. “Comparative Analysis of Mikrotik Site Filter Using Address List Techniques, Layer7 Protocols, Web Proxy, Mangle and DNS Static.” International Journal of Engineering and Technology(UAE) 7(3.4 Special Issue 4): 272–75.

Hunt, Craig. 1992. TCP/IP Network Administration. EDITION, T. ed. Mike Loukides and Debra Cameron. O’Reilly Media, Inc.

Iswadi, Defri, Ramzi Adriman, and Rizal Munadi. 2019. “Adaptive Switching PCQ-HTB Algorithms for Bandwidth Management in RouterOS.” Proceedings: CYBERNETICSCOM 2019 - 2019 IEEE International Conference on Cybernetics and Computational Intelligence: Towards a Smart and Human-Centered Cyber World: 61–65.

Ivancic, Dorian, Nikola Hadjina, and Danko Basch. 2005. “Analysis of Precision of the HTB Packet Scheduler.” 2005 18th International Conference on Applied Electromagnetics and Communications, ICECom 2005: 1–4.

Jia, Chengjun et al. 2018. “Multi-Core HTB for Bandwidth Sharing.” ANCS 2018 - Proceedings of the 2018 Symposium on Architectures for Networking and Communications Systems (July): 169–71.

Konstantinos S.2001.”Quality of service for IP-based networks”.Tesis.Tidak Diterbitkan.Master of Science in Computer Science and Master of Science in Information Technology Management.Naval Postgraduate School: Monterey, CA

Kuperman, Yossi, Maxim Mikityanskiy, and Rony Efraim. “Hierarchical QoS Hardware Offload (HTB).”

Lee, Chang Hwan, and Young Tak Kim. 2013. “QoS-Aware Hierarchical Token Bucket (QHTB) Queuing Disciplines for QoS-Guaranteed Diffserv

M, Embedded Systems. 2001. “Quality of Service for Packet Networks.” (M).

Marcel. 2018. “Performance Evaluation of MikroTik-Based Virtual Machine for Small-Scale Network Virtualization on VMware Platform.” Proceedings - 2018 International Conference on Control, Electronics, Renewable Energy and Communications, ICCEREC 2018: 154–58.

Mikrotikls SIA. 2007.”MikroTik RouterOS™” v3.0 Reference Manual. Riga, Latvia: Mikrotik.

Nayak, Ajit Kumar, Satyananda Champati Rai, and Rajib Mall. 2016. Computer Network Simulation Using NS2 Introduction to NS2.

Provisioning with Optimized Bandwidth Utilization and Priority-Based Preemption.” International Conference on Information Networking: 351–58.

Rodriguez, Adolfo, John Gatrell, John Karas, and Roland Peschke. “TCP / IP Tutorial and Technical Overview.”

Smansub, C., B. Purahong, P. Sithiyopasakul, and C. Benjangkaprasert. 2019. “A Study of Network Bandwidth Management by Using Queue Tree with per Connection Queue.” Journal of Physics: Conference Series 1195(1).