

QUALITY OF SERVICE (QOS) PADA JARINGAN INTERNET DENGAN METODE HIERARCHY TOKEN BUCKET

Rasudin
rasudin@gmail.com

ABSTRAK

Permasalahan lambatnya akses Internet client terjadi ketika mengakses Internet secara bersamaan, akibat distribusi bandwidth yang tidak merata oleh pengguna yang melakukan aktivitas download dan upload. Pengertian Quality of Service (QoS) adalah kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan bandwidth. Tujuan saya menggunakan metode Hierarchy Token Bucket pada DD-WRT untuk mengatur kecepatan Upload dan download pada masing-masing alamat IP client secara sentralisasi menggunakan firmware dd-wrt. Dengan demikian jika ada client yang mengakses internet membutuhkan kapasitas bandwidth yang besar, maka client lain tidak akan terganggu, karena masing-masing sudah mempunyai kapasitas bandwidth yang dipakai untuk mengakses internet.

Kata Kunci : *Bandwidth, Quality of Service, Hierarchy Token Bucket*

PENDAHULUAN

Penggunaan jaringan *wireless* semakin banyak. Hal ini terlihat dengan terdeteksinya beberapa jaringan *wireless* dan *ssid* pada lokasi tertentu. Penggunaan jaringan *wireless* yang tidak disertai manajemen *bandwidth* yang benar dapat mengakibatkan beberapa masalah seperti terjadi penguasaan *bandwith* jaringan oleh segelintir user. Perangkat *wireless* merk tertentu sudah mendukung tertentu, mempunyai perangkat lunak dan mekanisme pengelolaan perangkat *wireless*. Program *wireless* yang dimaksudkan adalah dengan memanfaatkan teknologi *upgrade firmware DD-WRT*. Program meliputi mulai dari teknologi pemrograman sampai proses injeksi program pada perangkat *wireless* dengan memasukkan berbagai program eksekusi untuk menjadikan perangkat radio punya fitur yang lengkap.

Banyak sekali produk *access point* yang dapat mengimplementasikan *firmware* DD-WRT ini. Salah satu produk yang dapat digunakan adalah Linksys WRT54GL. Secara *default* Linksys WRT54GL dengan *firmware* aslinya hanya dapat berjalan dalam Mode AP saja, jadi dia tidak dapat berperan sebagai *station* atau *klient*. Melalui *upgrade firmware* dengan DD-WRT, maka kemampuan dari Linksys WRT54GL pun akan meningkat, fitur seperti *gain power yang dapat diatur, support WDS, VPN, Klient Bridge, WEP, WPA, WPA2, juga ada Bandwidth Monitoring, Firewall, MAC Address Cloning, Radius, Quality of Service.*

Jenis aplikasi di *internet* sangat beragam. Ada yang membutuhkan *delay* yang minimum seperti aplikasi *multimedia*, ada yang membutuhkan *bandwidth* sebesar-besarnya seperti aplikasi pada *ftp*, dan ada pula yang membutuhkan *bandwidth* sangat kecil seperti *icmp* dan *web statis*. Jika semua aplikasi tersebut tidak dikontrol, maka satu aplikasi dapat menguasai *bandwidth* yang tersedia dan aplikasi lainnya tidak mendapatkan jatah, sehingga kinerja jaringan terasa menurun. Jika hal itu terjadi pada jaringan dengan banyak *node*, maka turunnya kinerja jaringan akan sangat terasa (Alistair, 2000).

TCP/IP

Internet terbentuk dari jaringan-komputer yang tersebar di seluruh dunia, masing-masing jaringan-komputer terdiri dari tipe-tipe komputer dan jenis jaringan yang berbeda. Untuk mengintegrasikan seluruh jaringan komputer tersebut dibutuhkan sebuah protokol pengiriman data yang tak bergantung pada jenis komputer dan digunakan oleh semua komputer untuk saling bertukar data. Protokol pengiriman merupakan sebuah konvensi (kesepakatan) yang menetapkan dengan cara apa data dikirimkan dan bagaimana kesalahan yang terjadi dikenali serta dipecahkan. Secara sederhana proses pengiriman data terdiri atas dua langkah.

Pertama, data yang akan dikirimkan (misalnya sebuah file teks) dibagi ke dalam paket data berukuran data berukuran sama (paket), kemudian dikirimkan satu per satu. Di *Internet*, protokol ini disebut *IP (Internet Protocol)*. Kedua, harus dijamin setiap paket data sampai ke alamat yang benar dan semuanya benar diterima. Untuk itu diperlukan protokol lainnya, yaitu *Transmission Control Protocol (TCP)* mengaitkan sebuah blok data pada

paket data *IP*, yang antara lain mengandung informasi mengenai alamat, jumlah total paket data dan urutan setiap paket yang membentuk paket tersebut. Hanya secara bersamaan kedua *protokol* membentuk kesatuan yang berfungsi, karena itu biasanya disebut *TCP/IP*.

QUALITY OF SERVICE (QoS)

Quality of Service (QoS) adalah kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan *bandwidth*, mengatasi *jitter* dan *delay*. Parameter *QoS* adalah *latency*, *jitter*, *packet loss*, *throughput*, *MOS*, *echo cancellation* dan *PDD*. *QoS* sangat ditentukan oleh kualitas jaringan yang digunakan. Terdapat beberapa faktor yang dapat menurunkan nilai *QoS*, seperti : Redaman, *Distorsi*, dan *Noise*. *QoS* didesain untuk membantu *end user* (klien) menjadi lebih produktif dengan memastikan bahwa user mendapatkan performansi yang handal dari aplikasi-aplikasi berbasis jaringan. *QoS* mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda.

Kemampuan *QoS* mengacu padae tingkat kecepatan dan kehandalan penyampaian berbagai jenis beban data di dalam suatu komunikasi. Kemampuannya merupakan kumpulan dari beberapa parameter besaran teknis, yaitu :

- 1). *Throughput*, yaitu kecepatan (*rate*) transfer data efektif, yang diukur dalam bps. *Troughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada destination selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut.
- 2). *Delay* merupakan total waktu yang dilalui suatu paket dari pengirim ke penerima melalui jaringan. *Delay* dari pengirim ke penerima pada dasarnya tersusun atas hardware *latency*, *delay* akses, dan *delay* transmisi. *Delay* yang paling sering dialami oleh trafik yang lewat adalah *delay* transmisi.

Tabel 1. Delay/Latensi

KATEGORI LATENSI	BESAR DELAY
Sangat Memuaskan	< 150 ms
Memuaskan	150 s/d 300 ms

Kurang Memuaskan	300 s/d 450 ms
Tidak Memuaskan	> 450 ms

Untuk aplikasi-aplikasi suara dan video interaktif, kemunculan dari *delay* akan mengakibatkan sistem seperti tak merespon.

- 3). *Jitter* merupakan variasi dari *delay* end-to-end. Level-level yang tinggi pada *jitter* dalam aplikasi-aplikasi berbasis *UDP* merupakan situasi yang tidak dapat diterima dimana aplikasi-aplikasinya merupakan aplikasi-aplikasi real-time, seperti sinyal audio dan video. Pada kasus seperti itu, *Jitter* akan menyebabkan sinyal terdistorsi, yang dapat diperbaiki hanya dengan meningkatkan buffer di antrian.

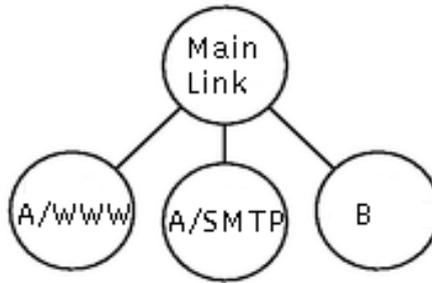
Tabel 2. Degradasi/Jitter

KATEGORI DEGRADASI	PEAK JITTER
Sangat memuaskan	0 ms
Memuaskan	0 s/d 75 ms
Kurang Memuaskan	76 s/d 125 ms
Tidak Memuaskan	125 s/d 225 ms

HIERARCHICAL TOKEN BUCKET

Hierarchical Token Bucket (HTB) merupakan teknik penjadwalan paket yang digunakan kebanyakan *router* berbasis Linux, dikembangkan pertama kali oleh Martin Devara (2002).

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



Gambar 1. Tampilan konsep Link Sharing

ROUTER

Router adalah perangkat untuk mengatur *Quality of Service (QoS)* sehingga diatur besarnya *bandwidth* yang dialokasikan. Pengguna lokal tentunya akan memperoleh *bandwidth* paling besar dibandingkan pengguna yang roaming, apalagi dibandingkan dengan pengguna tamu yang tidak diketahui identitasnya.

Fungsi rauter tidak terdapat dalam semua perangkat wireless (*access point*) secara maksimal. Atas dasar itu maka perlu dibahas *Quality of service (QoS)* yang memungkinkan user menggunakan fasilitas jaringan wireless dengan manajemen *bandwidth* yang lebih kompleks. Salah satunya adalah dengan mengganti *firmware* bawaan distributor (*default*) menjadi *firmware DD-WRT.QoS (Quality of Service)* pada *firmware DD-WRT* yang mampu memberikan prioritas berbeda untuk berbagai aplikasi, pengguna, aliran data, atau untuk menjamin tingkat kinerja tertentu ke aliran data. Caranya adalah dengan memanfaatkan fungsi *Mac & Port Priority* (Mulyanta, 2005).

JARINGAN NIRKABEL/WIRELESS

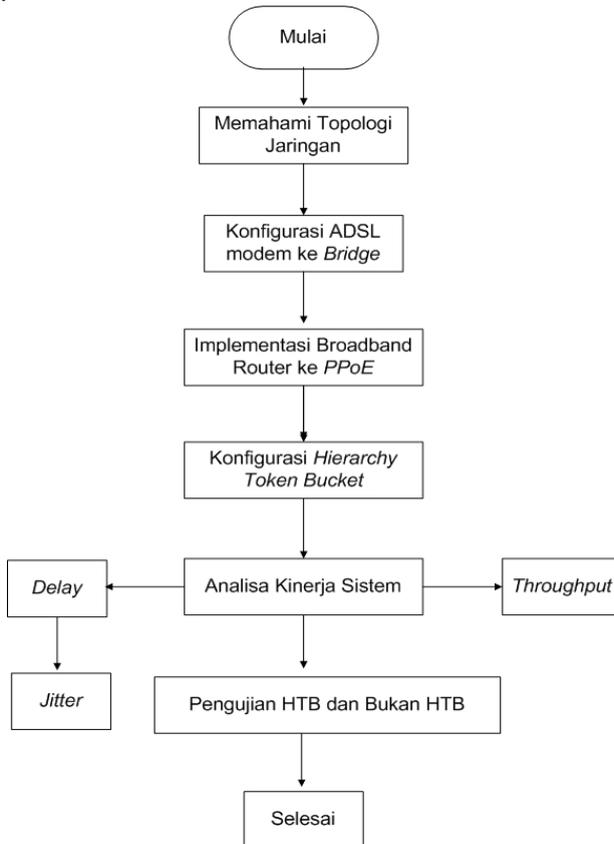
Wireless LAN bekerja dengan menggunakan gelombang radio. Sinyal radio menjalar dari pengirim ke penerima melalui *free space*, pantulan, difraksi, *Line of Sight* dan *Obstructed (LOS)*. Ini berarti sinyal radio tiba dipenerima melalui banyak jalur *Multipath*, dimana tiap sinyal (pada jalur yang berbeda-beda) memiliki level kekuatan, *delay* dan fase yang berbeda-beda.

Konfigurasi wireless LAN pada umumnya, membutuhkan alat *transmitter/receiver (transceiver)* yang biasa disebut *access point (AP)*, terhubung

pada wired *network* dari lokasi yang tetap menggunakan pengkabelan yang standar. Acces point menerima, *buffer*, dan mentransmisikan data antara *wireless LAN* dan *wired LAN*.

I. METODOLOGI PENELITIAN

Metode kerja yang digunakan dalam pembuatan *Quality of Service* Jaringan internet adalah Dengan Metode *Hierarchy Token Bucket* menggunakan *firmware DD-WRT*. Dengan adanya perancangan yang baik maka akan didapatkan suatu hasil yang baik. Ada beberapa tahapan yang digunakan dalam proses pembuatan *Hierarchy Token Bucket*, yaitu:



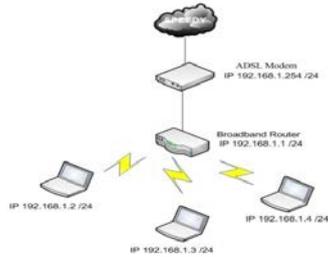
Gambar 2. Tampilan cara kerja HTB

- a. Memahami *topologi* jaringan dengan menggambarkan struktur dari suatu jaringan yang akan dibuat.
- b. Konfigurasi ADSL Modem sebagai *Bridge Mode* dan implementasi *Broadband Router* sebagai *PPoE* dan *DHCP Server*. Modem berfungsi sebagai menjembatani koneksi yang dilakukan oleh PC ke server *Internet Service Provider*. Pada *Broadband Router* bukan hanya berfungsi sebagai *Bandwidth* Manajemen namun berbagai fitur lain dapat difungsikan untuk berbagai layanan *Public* yaitu melakukan *remote router* melalui *Telnet*, *SSH*, *Putty* dan *Web Besh*, *Web Server*, *FTP Server*, dan memberikan layanan yang dapat menyewakan alamat *IP* dan informasi *TCP/IP* lainnya kepada semua klien yang memintanya.
- c. Konfigurasi *Hierarchy Token Bucket* yaitu dengan menentukan berapa *downlink* dan *uplink*, *Traffic Classification*, *DL Rate*, *DL Ceil*, *UL Rate*, *UL Ceil*, dan *Priority*.
- d. Analisis Kinerja Sistem yaitu suatu proses untuk menentukan hubungan antara 3 konsep utama, yaitu sumber daya (*resources*), penundaan (*delay*) dan daya-kerja (*throughput*) dan (*jitter*). Obyektif analisa kinerja mencakup analisa sumber daya dan analisa daya kerja. Nilai keduanya ini kemudian digabung untuk dapat menentukan kinerja yang masih dapat ditangani oleh sistem.
- e. Pengujian dilakukan terhadap *bandwidth* ketika sebelum menerapkan *hierarchy token bucket* dan sesudah menerapkan *hierarchy token bucket*. Pengujian ini dilakukan untuk mengetahui apakah metode *hierarchy token Bucket* dalam manajemen *bandwidth* dapat berjalan dengan baik dan lancar.

HASIL DAN PEMBAHASAN

Topologi Jaringan

Topologi Jaringan yang digunakan dalam implementasi *Hierarchy Token Bucket* seperti pada gambar berikut:

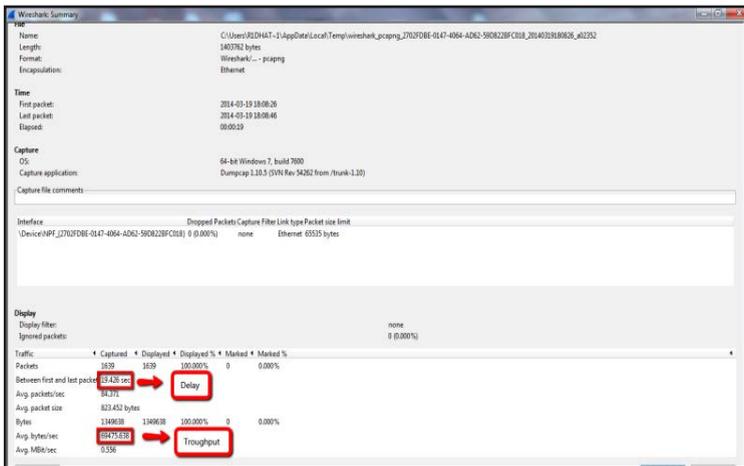


Gambar 3. Topologi Jaringan

ANALISA KINERJA SISTEM THROUGHPUT, DELAY DAN JITTER SEBELUM MENGGUNAKAN HIERARCHY TOKEN BUCKET.

Sebelum *hierarchy token bucket* diterapkan pada jaringan, dilakukan suatu pengukuran terhadap kinerja jaringan. Hal ini dimaksudkan agar terlihat bagaimana efek yang akan terjadi setelah diterapkannya *hierarchical token bucket* pada jaringan, terutama pada masalah *throughput*, *delay* dan *jitter* dari jaringan.

Pada gambar 4., ketika melakukan *download* ukuran file 1.16 MB yang terdiri *bandwidth* 585 kbps, Traffic Packet yang diambil adalah 1639. *delay* antar packet pertama dan terakhir ialah 19.426 sec dan *throughput* rata-rata bytes/sec ialah 69475.638 jika di ukur dengan kilobit ialah 556 kb.



Gambar 4. Tampilan halaman *delay* dan *throughput download*

Pada gambar 5 di sisi *jitter* server dibawah ini jumlah transfer 716 KBytes dengan *bandwidth* 584 Kbit/sec dalam waktu 0.0-10.0 sec dan total datagrams 499 mengalami rata-rata *jitter* 0.811ms.

```
C:\Users\rldhatech\Desktop\iperf>iperf -s -u -i 1
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
-----
[ 3] local 192.168.1.145 port 5001 connected with 192.168.1.119 port 63122
[ ID] Interval      Transfer      Bandwidth      Jitter      Lost/total Datagrams
[ 3] 0.0- 1.0 sec   71.8 KBytes   588 kbts/sec   0.699 ms    0/ 50 (0%)
[ 3] 1.0- 2.0 sec   71.8 KBytes   588 kbts/sec   0.683 ms    0/ 50 (0%)
[ 3] 2.0- 3.0 sec   70.3 KBytes   576 kbts/sec   1.258 ms    0/ 49 (0%)
[ 3] 3.0- 4.0 sec   71.8 KBytes   588 kbts/sec   1.709 ms    0/ 50 (0%)
[ 3] 4.0- 5.0 sec   71.8 KBytes   588 kbts/sec   1.273 ms    0/ 50 (0%)
[ 3] 5.0- 6.0 sec   70.3 KBytes   576 kbts/sec   1.281 ms    0/ 49 (0%)
[ 3] 6.0- 7.0 sec   71.8 KBytes   588 kbts/sec   2.357 ms    0/ 50 (0%)
[ 3] 7.0- 8.0 sec   71.8 KBytes   588 kbts/sec   1.509 ms    0/ 50 (0%)
[ 3] 8.0- 9.0 sec   71.8 KBytes   588 kbts/sec   1.202 ms    0/ 50 (0%)
[ 3] 9.0-10.0 sec   71.8 KBytes   588 kbts/sec   0.865 ms    0/ 50 (0%)
[ 3] 0.0-10.0 sec   716 KBytes   585 kbts/sec   0.811 ms    0/ 499 (0%)
```

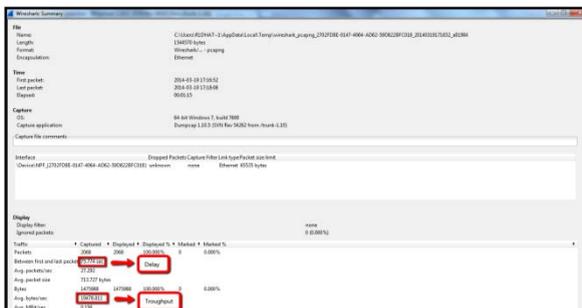
Gambar 5. Tampilan halaman *jitter* server *download*

Pada gambar 6 di sisi *jitter* klien dibawah ini laporan server dari waktu 0.0 sampai dengan 10.0 sec transfer 716 Kbytes dengan kecepatan *bandwidth* 584 Kbit/sec mengalami *jitter* 0.811 ms data total datagram terkirim semuanya.

```
Client connecting to 192.168.1.145, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
-----
[ 3] local 192.168.1.119 port 63122 connected with 192.168.1.145 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec   716 KBytes   584 kbts/sec
[ 3] Sent 499 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec   716 KBytes   585 kbts/sec   0.811 ms    0/ 499 (0%)
```

Gambar 6. Tampilan halaman *jitter* klien *download*

Pada gambar 7 ketika melakukan *upload* ukuran file 1.16 MB yang terdiri *bandwidth* 585 kbps, Traffic Packet yang diambil adalah 2068. *delay* antar packet pertama dan terakhir ialah 75.744 sec dan *throughput* rata-rata bytes/sec ialah 194778.811 jika di ukur dengan kilobit ialah 556 kb.



Gambar 7. Tampilan halaman *delay* dan *throughput* *upload*

Pada gambar 8 di sisi *jitter* server dibawah ini jumlah transfer 716 KBytes dengan *bandwidth* 584 Kbit/sec dalam waktu 0.0-10.0 sec dan total datagrams 499 mengalami rata-rata *jitter* 1.777ms.

```
C:\Users\r\idhatech\Desktop\iperf>iperf -s -u -i 1
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
-----
[ 3] local 192.168.1.145 port 5001 connected with 192.168.1.119 port 63123
[ ID] Interval      Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 3] 0.0- 1.0 sec  71.8 KBytes  588 Kbits/sec  0.978 ms    0/ 50 (0%)
[ 3] 1.0- 2.0 sec  71.8 KBytes  588 Kbits/sec  2.193 ms    0/ 50 (0%)
[ 3] 2.0- 3.0 sec  70.3 KBytes  576 Kbits/sec  1.465 ms    0/ 49 (0%)
[ 3] 3.0- 4.0 sec  71.8 KBytes  588 Kbits/sec  2.310 ms    0/ 50 (0%)
[ 3] 4.0- 5.0 sec  71.8 KBytes  588 Kbits/sec  1.594 ms    0/ 50 (0%)
[ 3] 5.0- 6.0 sec  70.3 KBytes  576 Kbits/sec  1.760 ms    0/ 49 (0%)
[ 3] 6.0- 7.0 sec  71.8 KBytes  588 Kbits/sec  1.582 ms    0/ 50 (0%)
[ 3] 7.0- 8.0 sec  71.8 KBytes  588 Kbits/sec  0.992 ms    0/ 50 (0%)
[ 3] 8.0- 9.0 sec  71.8 KBytes  588 Kbits/sec  1.308 ms    0/ 50 (0%)
[ 3] 9.0-10.0 sec  71.8 KBytes  588 Kbits/sec  1.899 ms    0/ 50 (0%)
[ 3] 0.0-10.0 sec  716 KBytes  583 Kbits/sec  1.777 ms    0/ 499 (0%)
```

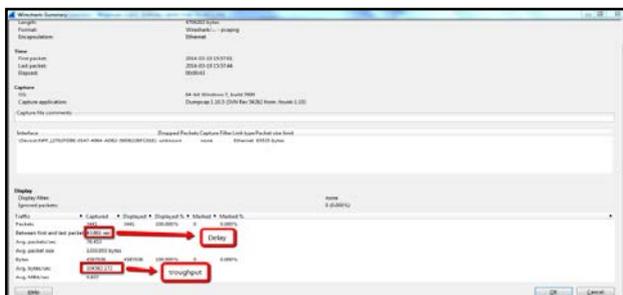
Gambar 8. Tampilan halaman *jitter* di server *upload*

Pada gambar 9 di sisi *jitter* klien dibawah ini laporan server dari waktu 0.0 sampai dengan 10.0 sec transfer 716 Kbytes dengan kecepatan *bandwidth* 584 Kbit/sec mengalami *jitter* 0.811 ms data total datagram terkirim semuanya.

```
C:\Users\r\idha\Desktop\iperf-2.0.5-2-win32\IPERF>iperf -c 192.168.1.145 -u -b 5k
-----
Client connecting to 192.168.1.145, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
-----
[ 3] local 192.168.1.119 port 63123 connected with 192.168.1.145 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec  716 kbytes    584 kbits/sec
[ 3] Sent 499 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec  716 KBytes  585 Kbits/sec  1.777 ms    0/ 499 (0%)
```

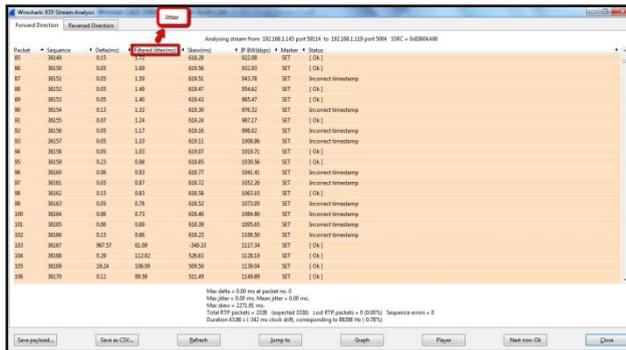
Gambar 9. Tampilan halaman *jitter* klien *upload*

Pada gambar 10 ketika melakukan *streaming* video ukuran file 63.5 MB yang terdiri *bandwidth* 585 kbps, Traffic Packet yang diambil adalah 3441. *delay* antar packet pertama dan terakhir ialah 43.861 sec dan *throughput* rata-rata bytes/sec ialah 104582.172 jika di ukur dengan kilobit ialah 556 kb.



Gambar 10. Tampilan halaman *delay* dan *throughput* stream video

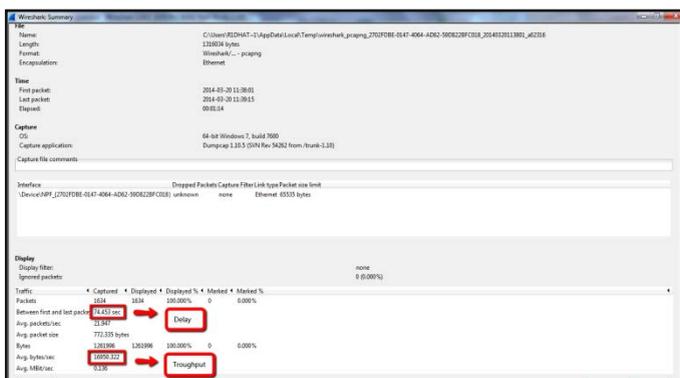
Pada gambar 11 ketika melakukan *streaming* video ukuran file 63.5 MB yang terdiri *bandwidth* 585 kbps, Total RTP packets adalah 3338. Lost RTP packets 0% dan duration 43.86 s (-0.78%) dan packet Incorrect timestamp terdiri dari 10 packet.



Gambar 11. Tampilan halaman *jitter* stream video

ANALISA KINERJA SISTEM THROUGHPUT, DELAY DAN JITTER MENGGUNAKAN HIERARCHY TOKEN BUCKET.

Ketika melakukan *download* ukuran file 1.16 MB yang terdiri *bandwidth* 585 kbps, *delay* antar packet pertama dan terakhir ialah 74.453 sec dan *throughput* rata-rata bytes/sec ialah 16950.332 jika di ukur dengan kilobit ialah 136 kb.



Gambar 12. Tampilan halaman *delay* dan *throughput* download

Pada gambar di bawah ini jumlah transfer 716 KBytes dengan *bandwidth* 584 Kbit/sec dalam waktu 0.0-10.0 sec dan total datagrams 499

mengalami rata-rata *jitter* 0.811ms. itu berarti *jitter* nya mengalami dengan rata-rata *jitter* 0.379 ms.

```
C:\Users\ridhatech\Desktop\iperf>iperf -s -u -i 1
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 64.0 Kbyte (default)
-----
[ 3] local 192.168.1.145 port 5001 connected with 192.168.1.119 port 57463
-----


| ID | Interval     | Transfer    | Bandwidth     | Jitter   | Lost/Total | Datagrams |
|----|--------------|-------------|---------------|----------|------------|-----------|
| 3] | 0.0- 1.0 sec | 70.3 Kbytes | 576 Kbits/sec | 1.065 ms | 0/         | 49 (0%)   |
| 3] | 1.0- 2.0 sec | 71.8 Kbytes | 588 Kbits/sec | 1.305 ms | 0/         | 50 (0%)   |
| 3] | 2.0- 3.0 sec | 70.3 Kbytes | 576 Kbits/sec | 0.999 ms | 0/         | 49 (0%)   |
| 3] | 3.0- 4.0 sec | 71.8 Kbytes | 588 Kbits/sec | 0.761 ms | 0/         | 50 (0%)   |
| 3] | 4.0- 5.0 sec | 71.8 Kbytes | 588 Kbits/sec | 0.816 ms | 0/         | 50 (0%)   |
| 3] | 5.0- 6.0 sec | 71.8 Kbytes | 588 Kbits/sec | 0.651 ms | 0/         | 50 (0%)   |
| 3] | 6.0- 7.0 sec | 71.8 Kbytes | 588 Kbits/sec | 0.777 ms | 0/         | 50 (0%)   |
| 3] | 7.0- 8.0 sec | 71.8 Kbytes | 588 Kbits/sec | 0.449 ms | 0/         | 50 (0%)   |
| 3] | 8.0- 9.0 sec | 70.3 Kbytes | 576 Kbits/sec | 0.203 ms | 0/         | 49 (0%)   |
| 3] | 9.0-10.0 sec | 71.8 Kbytes | 588 Kbits/sec | 0.431 ms | 0/         | 50 (0%)   |
| 3] | 0.0-10.0 sec | 716 Kbytes  | 584 Kbits/sec | 0.379 ms | 0/         | 499 (0%)  |


```

Gambar 13. Tampilan halaman *jitter* server *download*

Pada gambar 14 di sisi *jitter* klien dibawah ini laporan server dari waktu 0.0 sampai dengan 10.0 sec transfer 716 Kbytes dengan kecepatan *bandwidth* 584 Kbit/sec mengalami *jitter* 0.378 ms data total datagram terkirim semuanya.

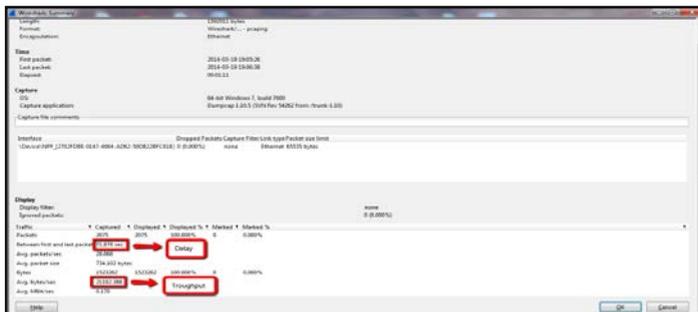
```
C:\Users\ridha\Desktop\iperf-2.0.5-2-win32\IPERF>iperf -c 192.168.1.145 -u -b 5k
-----
Client connecting to 192.168.1.145, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
-----
[ 3] local 192.168.1.119 port 57463 connected with 192.168.1.145 port 5001
-----


| ID | Interval           | Transfer   | Bandwidth     | Jitter   | Lost/Total | Datagrams |
|----|--------------------|------------|---------------|----------|------------|-----------|
| 3] | 0.0-10.0 sec       | 716 Kbytes | 584 Kbits/sec |          |            |           |
| 3] | Sent 499 datagrams |            |               |          |            |           |
| 3] | Server Report:     |            |               |          |            |           |
| 3] | 0.0-10.0 sec       | 716 Kbytes | 584 Kbits/sec | 0.378 ms | 0/         | 499 (0%)  |


```

Gambar 14. Tampilan halaman *jitter* klien *download*

Pada gambar dibawah ini ketika melakukan *upload* ukuran file 1.16 MB yang terdiri *bandwidth* 585 kbps, Traffic Packet yang diambil adalah 2075. *delay* antar packet pertama dan terakhir ialah 71.878 sec dan *throughput* rata-rata bytes/sec ialah 21192.388 jika di ukur dengan kilobit ialah 170 kb.



Gambar 15. Tampilan halaman *delay* dan *throughput upload*

Pada gambar di sisi *jitter* server dibawah ini jumlah transfer 716 KBytes dengan *bandwidth* 584 Kbit/sec dalam waktu 0.0-10.0 sec dan total datagrams 499 mengalami rata-rata *jitter* 1.777ms.

```

C:\Users\rldhatech\Desktop\iperf>iperf -s -u -i 1
-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
-----
[ 3] local 192.168.1.145 port 5001 connected with 192.168.1.119 port 64904
[ ID] Interval      Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 3] 0.0- 1.0 sec  70.3 KBytes  576 Kbits/sec  1.017 ms    0/ 49 (0%)
[ 3] 1.0- 2.0 sec  71.8 KBytes  588 Kbits/sec  0.591 ms    0/ 50 (0%)
[ 3] 2.0- 3.0 sec  70.3 KBytes  576 Kbits/sec  0.718 ms    0/ 49 (0%)
[ 3] 3.0- 4.0 sec  71.8 KBytes  588 Kbits/sec  0.921 ms    0/ 50 (0%)
[ 3] 4.0- 5.0 sec  71.8 KBytes  588 Kbits/sec  1.132 ms    0/ 50 (0%)
[ 3] 5.0- 6.0 sec  70.3 KBytes  576 Kbits/sec  0.924 ms    0/ 49 (0%)
[ 3] 6.0- 7.0 sec  73.2 KBytes  600 Kbits/sec  0.684 ms    0/ 51 (0%)
[ 3] 7.0- 8.0 sec  71.8 KBytes  588 Kbits/sec  0.550 ms    0/ 50 (0%)
[ 3] 8.0- 9.0 sec  70.3 KBytes  576 Kbits/sec  1.056 ms    0/ 49 (0%)
[ 3] 9.0-10.0 sec  71.8 KBytes  588 Kbits/sec  1.326 ms    0/ 50 (0%)
[ 3] 0.0-10.0 sec  716 KBytes  584 Kbits/sec  1.576 ms    0/ 499 (0%)
    
```

Gambar 16. Tampilan halaman *jitter* upload

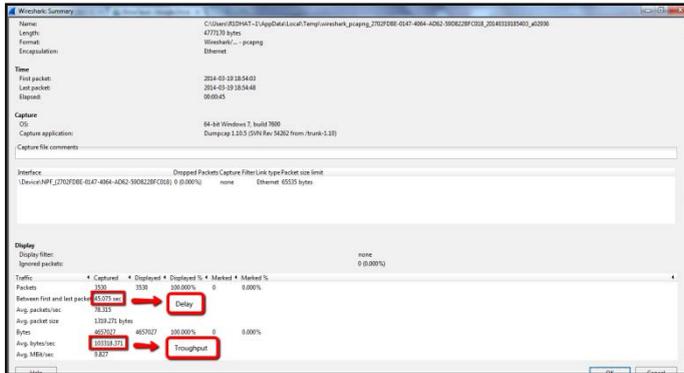
Pada gambar di sisi *jitter* klient dibawah ini laporan server dari waktu 0.0 sampai dengan 10.0 sec transfer 716 Kbytes dengan kecepatan *bandwidth* 584 Kbit/sec mengalami *jitter* 1.575 ms data total datagram terkirim semuanya.

```

Administrator: Command Prompt
C:\Users\rldha\Desktop\iperf-2.0.5-2-win32\IPERF>iperf -c 192.168.1.145 -u -b 58
5k
-----
Client connecting to 192.168.1.145, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
-----
[ 3] local 192.168.1.119 port 64904 connected with 192.168.1.145 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec  716 KBytes  584 Kbits/sec
[ 3] Sent 499 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec  716 KBytes  584 Kbits/sec  1.575 ms    0/ 499 (0%)
    
```

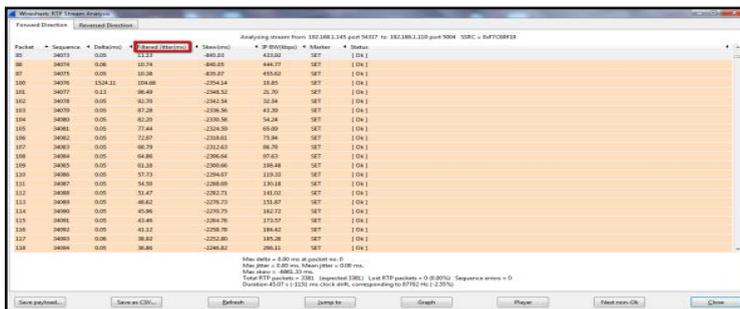
Gambar 17. Tampilan halaman *jitter* klien upload

Pada gambar dibawah ini Ketika melakukan *streaming* video ukuran file 63.5 MB yang terdiri *bandwidth* 585 kbps, Traffic Packet yang diambil adalah 3530. *delay* antar packet pertama dan terakhir ialah 45.075 sec dan *throughput* rata-rata bytes/sec ialah 103318.371 jika di ukur dengan kilobit ialah 827.



Gambar 18. Tampilan halaman *delay* dan *throughput* stream video

Pada gambar dibawah ini ketika melakukan *streaming* video ukuran file 63.5 MB yang terdiri *bandwidth* 585 kbps, Total RTP packets adalah 3381. Lost RTP packets 0% dan *duration* 45.07 (-2.55%) dan packet *Incorrect timestamp* terdiri dari 10 packet.



Gambar 19. Tampilan halaman *jitter* stream video

KESIMPULAN

Dari hasil pembahasan maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Jaringan yang tidak diterapkan manajemen *bandwidth* menggunakan *hierarchy token bucket* akan berakibat pada *throughput delay* dan *jitter* yang tidak terkontrol. Hal ini dapat dilihat perbedaan antara ketika melakukan *download*, *upload* ukuran file 1.16 MB yang terdiri *bandwidth* 585 kbps, terjadi perebutan *bandwidth* dan klien 1 sedang melakukan *download*,

klient 2 browsing terganggu dan menyebabkan *delay* dan *jitter* bertambah.

2. Konfigurasi sebelum menggunakan HTB pada saat *download* mengalami 0.811 dan menggunakan HTB *jitter* mengalami 0.378. ketika melakukan *upload* sebelum menggunakan HTB mengalami *jitter* mengalami 1.777 ms dan menggunakan HTB 1.575 ms.
3. Ketika melakukan *streaming* video dengan ukuran file 63.5 MB yang terdiri dari *bandwidth* 585 kbps tidak menggunakan HTB mengalami nilai *delta*(ms) 0.05 dan 0.06 berstatus *incorrect timestamp*. Sedangkan *jitter* yang bernilai *delta*(ms) 0.05 dan 0.06 berstatus ok. *Delta* (ms) ialah data yang pengiriman data sampai ketempat tujuan.

REFERENSI

- Croll, Alistair, 2000, *Managing Bandwidth Deploying QoS in Enterprise Networks*, Prentice Hall, New Jersey.
- Ferguson, P. & Huston, G., 1998, *Quality of Service*, John Wiley & Sons Inc
- Mulyanta, Edi S. 2005. *Pengenalan Protokol Jaringan Wireless Komputer*. Andi. Yogyakarta
- Prabawati, Th Ari. 2011. *Tips Jitu Optimasi Jaringan Wi-Fi*. Wahana Komputer Semarang.
- Purbo, Onno W. 2006. *Buku Pegangan Internet Wireless dan Hotspot*. Elex Media Komputindo. Jakarta
- Purbo, Onno W. dan Wiharjito, Tony. 2000. *Keamanan Jaringan Internet*. Elex Media Komputindo. Jakarta