

MODUL 4

MANAJEMEN BANDWIDTH DENGAN SQUID DAN HTB

TUJUAN PEMBELAJARAN:

Setelah melaksanakan praktikum ini, mahasiswa diharapkan :

1. Mengerti dan memahami konsep manajemen *bandwidth*.
2. Mampu melakukan instalasi dan konfigurasi *network traffic shapping*.

DASAR TEORI

Dalam lingkungan pekerjaan, pendidikan, dll saat penggunaan internet benar-benar dibutuhkan oleh setiap user. Hal ini tentu menjadi suatu masalah administrator jaringan, karena administrator harus mengelola pemakaian jaringan tersebut agar optimal. Sebagai contoh admin harus mengelola dan monitoring pemakaian jaringan tersebut agar benar-benar efisien dan terbagi oleh semua user, membatasi konten-konten apa saja untuk diakses, menentukan siapa saja user yang bisa menggunakan fasilitas internet, dll. Semua kendala tersebut harus diselesaikan oleh adminstrator agar jaringan bisa berjalan dengan baik

Bandwidth management dengan metode Squid Delay Pools adalah salah satu bentuk manajemen bandwidth yang sangat efisien karena setiap host/user akan mendapatkan bandwidth dengan ukuran yang sama tanpa mengganggu bandwidth dari host/user yang lain.

Pengertian Bandwidth Management

Bandwidth management adalah serangkaian mekanisme kontrol yang menilai data alokasi, penundaan variabilitas, tepat waktu pengiriman, dan kehandalan pengiriman dalam mengelola jalur internet agar kecepatannya menjadi efektif dan efisien. Dengan bandwidth management, kita dapat mengatur bandwidth sesuai dengan kebutuhan.

Pengertian Proxy

Proxy adalah sebuah komputer server yang digunakan sebagai perantara antara user dan internet. Untuk analoginya proxy sebagai perantara antara pihak pertama (user) dalam berhubungan dengan pihak kedua (internet), jadi pada saat user melakukan akses internet maka proxy sebagai perantara yang menyampaikan request dari user tersebut ke internet atau sebaliknya. Di sini user tidak langsung berhubungan dengan internet tetapi dengan menggunakan perantara proxy server user bisa terhubung dengan akses internet.

Proxy mempunyai banyak peran, tanpa proxy administrator akan sulit mengelola jaringan di sini, dan pasti permasalahan pasti akan banyak timbul, seperti load internet yang besar karena bandwidth yang tidak dibatasi, pembagian bandwidth yang tidak adil, banyaknya konten-konten tidak penting yang diakses user, dll. Di sini proxy mengatasi masalah tersebut, karena fungsi utama proxy yaitu melakukan proses **sharing**, **caching**, **filtering**, sehingga penggunaan internet dapat terkelola dengan baik.

▪ **Sharing**

Dimana semua user bisa bersama-sama saling terhubung ke proxy server dan dapat melakukan akses internet secara bersamaan melalui proxy server.

▪ **Caching**

Semua request yang diminta user dapat disimpan dalam jangka waktu yang cukup lama oleh proxy server dalam cache proxy, sehingga apabila user ingin mengakses situs atau konten yang sama, proxy tidak perlu lagi menghubungi alamat yang menyediakan konten tersebut, jadi user dapat mengakses konten tersebut dari cache yang disimpan proxy.

▪ **Filtering**

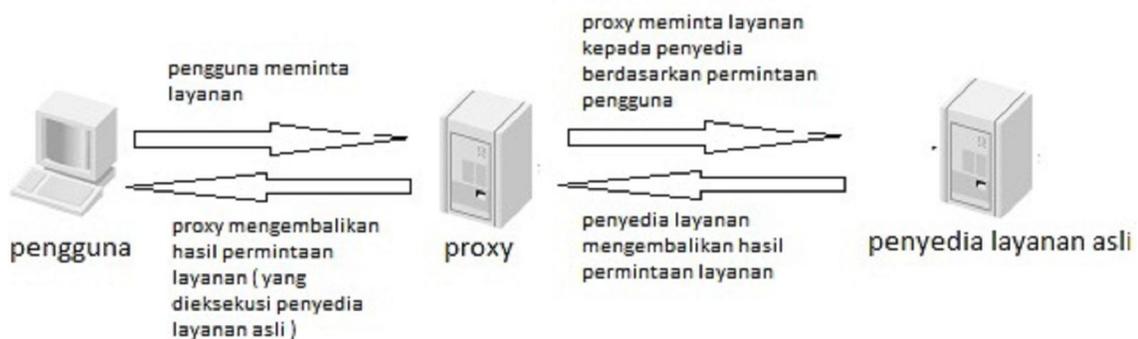
Proxy dapat diatur agar dapat melakukan penyaringan terhadap konten-konten yang tidak diinginkan seperti porno, judi, sara, phishing, konten-konten yang memakai bandwidth besar, dll. Sehingga semua user tidak dapat mengakses konten-konten tersebut.

Fungsi lain proxy server yaitu dapat menentukan user mana saja yang bisa mengakses internet, membagi dan membatasi bandwidth para user, membatasi download, melakukan pengaturan untuk akses-akses situs tertentu pada jam waktu yang diinginkan.

Hal ini mempermudah pekerjaan administrator jaringan, karena di sini peran administrator jaringan hanya terpusat di proxy server. Sehingga administrator hanya perlu melakukan monitoring penggunaan proxy dan melakukan troubleshoot apabila service proxy down atau bermasalah.

Cara Kerja Proxy

Dari sisi pengguna, proxy sama seperti penyedia layanan asli. Pengguna hanya perlu mengirimkan permintaan layanan, dan proxy akan melayani permintaan tersebut. Namun dalam proses eksekusi layanan tersebut, alih-alih mengeksekusinya sendiri, proxy melakukan permintaan layanan ke penyedia layanan asli. Setelah penyedia layanan asli memberikan hasil, kemudian proxy baru akan mengembalikan hasil eksekusi permintaan layanan ke pengguna. Sehingga dari sisi penyedia layanan asli, proxy sama seperti pengguna layanan.



Squid

Squid adalah high-performance proxy caching server untuk web klien, yang sudah mendukung FTP, gopher, dan HTTP data object. Berbeda dengan software caching yang lama, Squid menangani semua permintaan tunggal (single), non-blocking, I/O-driven proses. Squid menyimpan meta data yang di simpan di RAM, menyimpan DNS lookups, mendukung non-blocking DNS lookups, dan implementasi negative-caching jika permintaan gagal. Squid merupakan software proxy yang dapat diperoleh secara gratis. Squid juga dapat digunakan untuk mengendalikan pemakaian bandwidth berdasarkan ekstensi file-file tertentu, menyaring situs-situs yang boleh diakses.

Delay Pools

Delay pools merupakan salah satu fasilitas squid untuk membatasi bandwidth yang dikonsumsi client, delay pools juga adalah opsi untuk menspesifikasi berapa jumlah pool yang digunakan untuk membatasi jumlah bandwidth dari ACL tertentu. ACL (Access Control List), sederhananya digunakan untuk mengizinkan atau tidak paket host menuju ke tujuan tertentu. ACL terdiri atas aturan-aturan dan kondisi yang menentukan trafik jaringan dan menentukan proses nantinya akan dilewatkan atau tidak. Sebelum mulai mengkonfigurasi delay pools, harus dipersiapkan terlebih dahulu aplikasi squid yang sudah dikompilasi dengan support delay pools. Beberapa distro besar seperti RedHat/Mandrake biasanya sudah di atur support delay pools. Bila tidak, dapat dikompilasi sendiri/manual. Biasanya delay pools akan dirangkaikan bersama opsi-opsi yang lain, yaitu:

- Delay class, opsi ini menspesifikasikan dari masing- masing pool yang telah didefinisikan pada opsi delay pools. Ada 3 class yang didukung squid, antara lain:
 - a. Class 1, Akses dibatasi dengan single bucket, artinya hanya bisa mendefinisikan overall bandwidth untuk suatu ACL saja, tidak bisa mendefinisikan bandwidth dengan lebih mendetail.
 - b. Class 2, Semua akses dibatasi dengan single agregate dengan dua parameter bandwidth. Parameter pertama mendefinisikan berapa bandwidth maksimal yang didapatkan ACL, parameter kedua mendefinisikan berapa bandwidth overall untuk ACL yang spesifik yang ada pada network tersebut.

c. Class 3, Kelompok yang definisi bandwidth-nya paling mendetail. Parameter pertama mendefinisikan berapa bandwidth maksimal yang didapatkan ACL, parameter kedua mendefinisikan berapa bandwidth normal yang didapatkan ACL secara umum, dan parameter yang ketiga adalah mendefinisikan bandwidth yang didapatkan ACL jika mengakses ACL-ACL tertentu yang spesifik, misalnya file mp3.

Class-class ini dispesifikasi berdasarkan IP address dari ACL.

- Delay Parameter, Opsi ini menspesifikasikan berapa jumlah transfer rate atau lebih sering disebut bandwidth untuk suatu pool. Bandwidth dispesifikasi dalam transfer rate rata-rata dan transfer rate maksimum yang dapat dicapai suatu pool.
- Delay Access, Opsi ini mendefinisikan siapa-siapa ACL yang akan dimasukkan ke pool tertentu untuk mendapatkan “perlambatan” bandwidth.

HTB

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.

Pada antrian HTB mempunyai parameter yang menyusunnya dalam antrian yaitu :

1. Rate

Parameter *rate* menentukan *bandwidth* maksimum yang bisa digunakan oleh setiap *class*, jika *bandwidth* melebihi nilai “*rate*”, maka paket data akan dipotong atau dijatuhkan (*drop*).

2. Ceil

Parameter *ceil* di-set untuk menentukan peminjaman *bandwidth* antar *class* (kelas), peminjaman *bandwidth* dilakukan kelas paling bawah ke kelas di atasnya. Teknik ini disebut *link sharing*.

3. Random Early Detection (RED)

Random Early Detection atau bisa disebut *Random Early Drop* biasanya digunakan untuk *gateway/router backbone* dengan tingkat trafik yang sangat tinggi. RED mengendalikan trafik jaringan sehingga terhindar dari kemacetan pada saat trafik tinggi berdasarkan pemantauan perubahan nilai antrian minimum dan maksimum. Jika isi antrian dibawah nilai minimum, maka mode ‘*drop*’ tidak berlaku, saat antrian mulai terisi hingga melebihi nilai maksimum, maka RED akan membuang (*drop*) paket data secara acak sehingga kemacetan pada jaringan dapat dihindari. Pada antrian RED juga mempunyai parameter yang menyusunnya, yaitu :

a. Min

Yaitu nilai rata-rata minimum antrian (*queue*).

b. Max

Nilai rata-rata maksimum antrian, biasanya dua kali nilai minimum atau dengan rumus :

$$\text{Max} = \text{bandwidth (bps)} * \text{latency (s)}$$

c. Probability

Jumlah maksimum probabilitas penandaan paket data. Nilainya berkisar antara 0.0 sampai dengan 1.0.

d. Limit

Batas paling atas antrian secara riil, jumlah paket data yang melewati *limit* pasti dibuang. Nilai limit harus lebih besar daripada ‘*max*’ dan dinyatakan dengan persamaan :

$$\text{limit} = \text{max} + \text{burst}$$

e. Burst

Digunakan untuk menentukan kecepatan perhitungan nilai antrian mempengaruhi antrian riil (*limit*). Bisa dihitung dengan persamaan :

$$\text{Burst} = (\text{min} + \text{min} + \text{max}) / 3 * \text{avpkt}$$

f. Avpkt

Nilai rata – rata paket data/grafik yang melintasi *gateway* RED, sebaiknya diisi 1000.

g. Bandwidth

Yaitu lebar *bandwidth* kartu Ethernet.

h. Ecn (*Explicit Congestion Notification*)

Parameter ini memberikan fasilitas *gateway* RED untuk memberitahukan kepada *client* jika terjadi kemacetan.

PERALATAN :

- Sebuah komputer sebagai *server*, dengan 1 NIC Card
- Sebuah komputer sebagai *client*, dengan 1 NIC Card
- Sebuah komputer sebagai router, dengan 2 NIC Card
- *Hub/switch* sebagai penghubung jaringan
- Kabel jaringan secukupnya

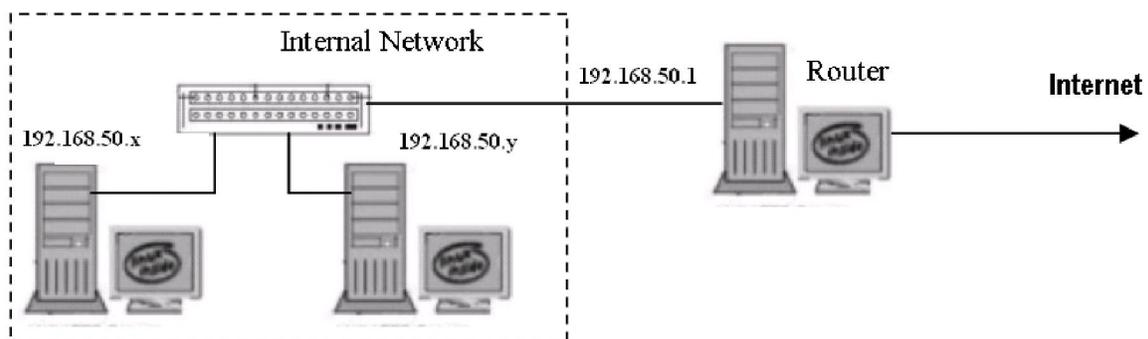
TUGAS PENDAHULUAN

1. Apa tujuan dengan adanya bandwidth management ?
2. Jelaskan perbedaan yang paling mendasar antara SQUID dan HTB untuk bandwidth management ?

PERCOBAAN

A. Traffic shaping dengan SQUID

Bangun jaringan sederhana seperti pada gambar berikut :



Gambar 1 Jaringan Percobaan

NB:

Gunakan dhclient di masing-masing PC untuk mendapatkan IP dari router.

192.168.50.x & y : IP dari router

Pilih 192.168.50.50 sebagai PC Server dengan Squid

Pilih 192.168.50.60 sebagai PC Client

1. Instalasi squid

```
# apt-get install squid
```

2. Edit file squid.conf

```
# vim /etc/squid/squid.conf
```

3. Cari bagian ini dan tambahkan rule:

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
```

```
acl kelompok1 src 192.168.50.0/255.255.255.0
```

=> no IP PC Client

```
acl admin src 192.168.50.60/255.255.255.255
http_access allow kelompok1
http_access allow admin
```

```
delay_pools 2
delay_class 1 1
delay_parameters 1 -1/-1
delay_access 1 allow admin
delay_access 1 deny kelompok1
```

=> terdapat 1 aturan
=> diberi class type 1
=> diberi full bandwidth
=> diberlakukan pada admin
=> kelompok1 ditolak

```
delay_class 2 1
delay_parameters 2 5000/8000
delay_access 2 allow kelompok1
```

=> maks 5Kb jika download file diatas 8Kb

4. Simpan dan kemudian restart squid


```
# /etc/init.d/squid restart
```
5. Ujicoba dengan menggunakan wget utk komputer selain punya 'admin'


```
# export http_proxy=http://192.168.50.50:3128
# wget http://galatea.eepis-its.edu/files/fonts.tar
```
6. Lihat hasilnya dari pembatasan bandwidth tersebut dan isilah tabel berikut:

Bandwidth	Waktu (s)	Throughput (Kbps)
64Kbps		
128Kbps		
256Kbps		
no QoS		

NB: untuk file, jangan gunakan file yang sama ketika didownload karena sudah tersimpan di cache PC Client.

B. Traffic shaping dengan HTB

1. Installasi aplikasi HTB

a. Download paket HTB

```
# mkdir /usr/src/htb
# cd /usr/src/htb
# wget http://galatea.eepis-its.edu/files/HTB-tools-0.3.0a-i486-1.tgz
```

b. Ekstrak file tersebut

```
# tar -zxvf HTB-tools-0.3.0a-i486-1.tgz
```

c. Pindahkan semua hasil ekstrak ke folder aslinya

```
# mv sbin/* /sbin/
# mv usr/* /usr/
# mv etc/htb /etc/
# mv etc/rc.d/* /etc/init.d/rc.htb
# chmod +x /etc/init.d/rc.htb
```

d. Copykan file konfigurasi berikut ini, karena yang dipakai hanya eth0, sehingga eth1 dihapus.

```
# cd /etc/htb/
# mv eth0-qos.cfg.new eth0-qos.cfg
# rm eth1-qos.cfg.new
```

e. Konfigurasi HTB Untuk proses download: eth0-qos.cfg

```
# vim /etc/htb/eth0-qos.cfg
```

```

class class_1 {
    bandwidth 10000;
    limit 20000;
    burst 2;
    priority 1;

    client client_1 {
        bandwidth 64;
        limit 128;
        burst 2;
        priority 1;
        dst {
            192.168.50.60/32;
        };
    };
};

class default { bandwidth 8; };

```

NB: Jika ingin maksimal di client dapat 64kbps :
 bandwidth 64;
 limit 64;

- f. Untuk menjalankan HTB, masuk ke /etc/htb, jalankan perintah berikut :
- ```

/etc/init.d/rc.htb stop_eth0
/etc/init.d/rc.htb start_eth0
/etc/init.d/rc.htb stats => untuk melihat statistik

```

2. Lakukan pengetesan seperti langkah A.6 dan bandingkan hasilnya.

| Bandwidth | Waktu (s) | Throughput (Kbps) |
|-----------|-----------|-------------------|
| 64Kbps    |           |                   |
| 128Kbps   |           |                   |
| 256Kbps   |           |                   |
| no QoS    |           |                   |

NB:

untuk memonitoring traffic bisa digunakan iptraf.

Untuk menguji kecepatan akses bisa menggunakan <http://www.eepis-its.edu/speedtest/>

**Tambahan:**

**Wondershaper**

1. Instalasi paket wondershaper  
**apt-get install wondershaper**
2. Lalu selanjutnya untuk membatasi, Syntax :  
**wondershaper [network interface] [down speed kbit/s] [up speed kbit/s]**

Contoh penggunaan :

Misalkan hendak memberikan kecepatan unduh (downspeed) maksimum 8 KiloBytes/Second dan kecepatan unggah (upload) 8 KiloBytes/Second di antarmuka jaringan eth0.

Pertama kita konversi dulu dari KiloBytes ke kilobits.

1 Kilobyte = 8 kilobits, maka kecepatan unduh =  $8 \times 8 = 64$  kilobits/second, kecepatan unggah =  $8 \times 8 = 64$  kilobits/second.

contoh : `wondershaper eth0 64 64`

3. Jika ingin menghapus antrian wondershaper (menonaktifkan traffic shaping):

**wondershaper clear [network interface]**

contoh : `wondershaper clear eth0`

## LAPORAN RESMI

### Daftar Pertanyaan

1. Berikan kesimpulan hasil praktikum yang anda lakukan.
2. Tugas akan diberikan pada waktu praktikum.