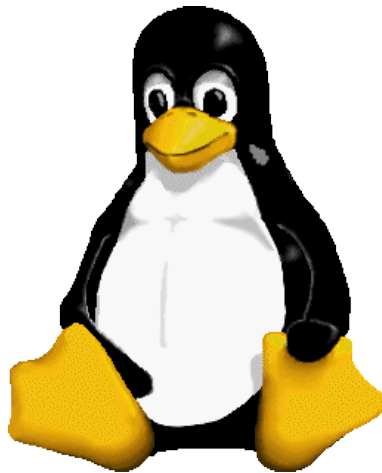


MANAJEMEN BANDWIDTH INTERNET DAN INTRANET
linux.multimedia@gmail.com

MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

DI SUSUN OLEH:

Budi Santosa,ST
linux.multimedia@gmail.com



Quality of Service

Quality of Service atau QoS digunakan untuk mengukur tingkat kualitas koneksi jaringan TCP/IP internet atau intranet. Ada beberapa metode untuk mengukur kualitas koneksi seperti konsumsi bandwidth oleh user, ketersediaan koneksi, latency, losses dll. Sekarang kita bahas istilah – istilah dalam Quality of Service

Bandwidth

Bandwidth adalah kapasitas atau daya tampung kabel ethernet agar dapat dilewati trafik paket data dalam jumlah tertentu. Bandwidth juga bisa berarti jumlah konsumsi paket data per satuan waktu dinyatakan dengan satuan bit per second [bps]. Bandwidth internet di sediakan oleh provider internet dengan jumlah tertentu tergantung sewa pelanggan. Dengan QoS kita dapat mengatur agar user tidak menghabiskan bandwidth yang di sediakan oleh provider.

Latency

Jika kita mengirimkan data sebesar 3 Mbyte pada saat jaringan sepi waktunya 5 menit tetapi pada saat ramai 15 menit, hal ini di sebut latency. Latency pada saat jaringan sibuk berkisar 50 – 70 msec.

Losses

Losses adalah jumlah paket yang hilang saat pengiriman paket data ke tujuan, kualitas terbaik dari jaringan LAN / WAN memiliki jumlah losses paling kecil.

Availability

Availability berarti ketersediaan suatu layanan web, smtp, pop3 dan aplikasi pada saat jaringan LAN / WAN sibuk maupun tidak.

Linux Traffic Control

Pengendalian trafik jaringan

Trafik jaringan berhubungan dengan paket data yang dibangkitkan oleh kartu ethernet pada komputer, sebelumnya kita membahas terlebih dahulu kartu ethernet. Pada Diagram 1.1 dibawah ini menunjukkan komputer linux dengan satu kartu ethernet.

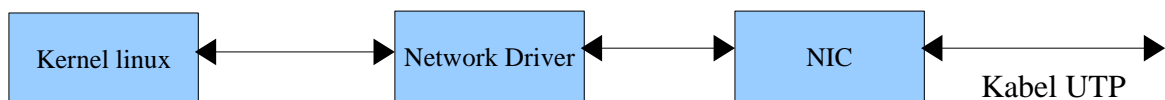


Diagram 1.1 komputer dgn satu kartu ethernet

Paket data yang dikirimkan oleh komputer lain diterima NIC (kartu ethernet), kemudian teruskan oleh driver kartu ethernet (Network Driver) ke bagian kernel linux untuk diproses. Proses ini hanya mengatur paket data yang keluar maupun masuk melalui satu kartu ethernet. Kernel linux yang bertanggung jawab mengatur aliran data disebut kernel traffic control.

Sedangkan pada diagram 1.2 menggambarkan komputer linux yang dioperasikan sebagai gateway atau router, aliran paket data dapat diatur secara bidirectional (dua arah) melalui NIC0 dan NIC1. Gateway linux dikonfigurasi untuk memisahkan trafik dari jaringan lain atau koneksi internet yang disediakan oleh ISP. Hubungan komputer klien yang dibagian NIC1 ke ISP dapat dikendalikan, misalnya bandwidth smtp di jatah 64Kbps, & ftp mendapatkan bandwidth 10Kbps.

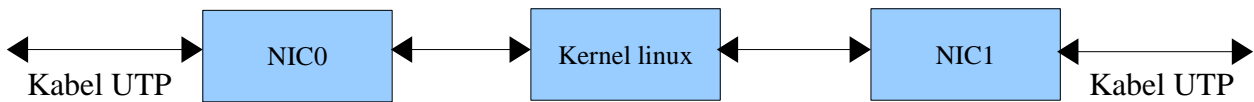


Diagram 1.2 komputer linux sebagai router / gateway

Struktur kernel traffic control

Pada dasarnya kernel traffic controll memiliki 3 bagian, yang pertama perangkat ingress yaitu jika paket data diterima oleh kartu LAN maka paket tersebut akan diproses oleh ingress, biasanya ingress dipakai untuk mengendalikan traffic upload / uplink. Kemudian perangkat egress dipergunakan untuk mengendalikan paket data yang keluar dari kartu ethernet, sehingga trafik download oleh komputer klien dapat dibatasi sesuai konfigurasi.

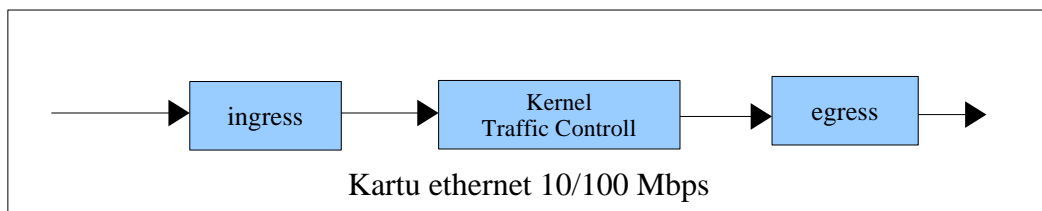


Diagram 1.3. Struktur kernel traffic control

Metode pengendalian trafik

Dalam mengendalikan trafik administrator jaringan bisa memilih beberapa metode tergantung dari situasi pada jaringan LAN atau backbone. Tiap trafik akan dikendalikan dengan metode tertentu yang akan berdampak pada kecepatan akses, jadi administrator jaringan perlu membaca dan mengerti bagian ini terlebih dahulu, beberapa metode pengendalian trafik sebagai berikut:

Prioritas

Pada metode prioritas paket data yang melintasi gateway diberikan prioritas berdasarkan port, alamat IP atau sub net. Jika trafik pada gateway sedang tinggi maka prioritas dengan nilai terendah (nilai paling rendah berarti prioritas tertinggi) akan di proses terlebih dahulu, sedangkan yang lainnya akan di berikan ke antrian atau dibuang. Metode prioritas paling cocok diterapkan pada koneksi internet yang memiliki bandwidth sempit, hanya trafik paling penting saja yang dilewatkan seperti smtp dan pop3.

FIFO

Pada metode FIFO jika trafik melebihi nilai set maka paket data akan dimasukkan ke antrian, paket data tidak mengalami pembuangan hanya tertunda beberapa saat. Metode FIFO cocok diterapkan pada koneksi internet dengan bandwidth menengah 64kbps, untuk menghindari bootle neck pada jaringan LAN. Paket data jika melebihi batas konfigurasi akan di masukkan ke dalam antrian dan pada saat jaringan LAN tidak sibuk maka paket data dalam antrian akan dikeluarkan.

Penjadwalan

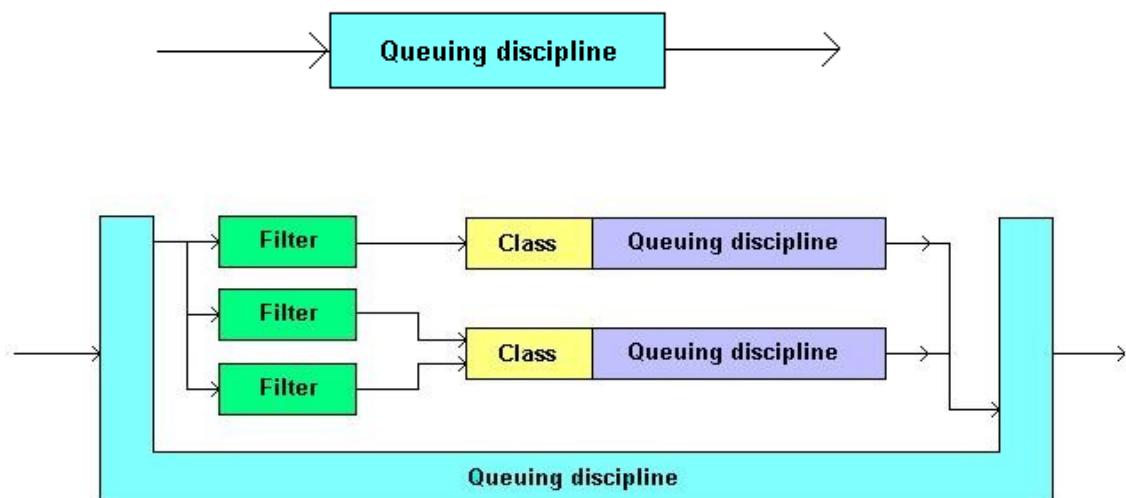
Metode penjadwalan atau scheduling ini paling sering dipakai karena memiliki kemampuan membagi paket data ke dalam ukuran yang sama besar kemudian memasukkan ke dalam beberapa antrian. Antrian itu kemudian di keluarkan oleh scheduler dengan algoritma round robin.

Shape & drop

Shape & drop merupakan metode paling cocok serta efektif untuk jaringan yang memiliki beban trafik sangat tinggi. Jika trafik melebihi nilai set maka paket data akan di masukan ke dalam antrian sehingga trafik menurun secara perlahan, metode ini disebut pemotongan bandwidth, kemudian jika trafik terus menerus melebihi nilai set maka paket data akan dibuang (drop).

qdisc

Antrian dalam setiap kartu ethernet di sebut qdisc (queuing discipline) yang dipergunakan untuk menyimpan antrian paket data, paket data masuk ataupun keluar melalui qdisc. Paket data yang memasuki qdisc akan dipisahkan oleh bagian filter untuk menentukan port / alamat ip yang akan di atur aliran trafiknya. Bagian class atau klasifikasi trafik akan dibahas pada bagian berikutnya, sedangkan qdisc yang berwarna ungu dipergunakan untuk mengeluarkan paket data ke kartu ethernet.

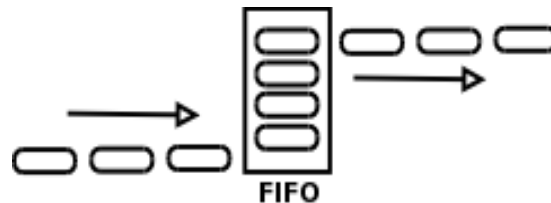


Gambar 1.5. Queuing discipline (qdisc)

Teknik antrian

First In First Out

Teknik antrian FIFO mengacu pada FCFS (First Come First Server), paket data yang pertama datang diproses terlebih dahulu. Paket data yang keluar terlebih dahulu di masukan ke dalam antrian FIFO, kemudian dikeluarkan sesuai dengan urutan kedatangan. Teknik antrian FIFO sangat cocok untuk jaringan dengan bandwidth menengah 64kbps tetapi cukup menghabiskan sumber daya prosessor dan memori.



Gambar 2.1. Antrian FIFO

Gambar 2.1 diatas menunjukkan kedatangan beberapa paket data yang berbeda waktu, paket pertama (1) dari flow 8 yang tiba lebih awal dikeluarkan ke port terlebih dahulu oleh antrian FIFO. Untuk men-set antrian kita memerlukan perintah “tc” dengan qdisc pfifo, parameter limit untuk menentukan batas maksimum antrian.

Menambahkan qdisc FIFO

```
[root@server root]# tc qdisc add dev eth0 root pfifo limit 100
[root@server root]# tc qdisc show dev eth0
qdisc pfifo 8001: limit 100p
```

untuk melihat paket data yang telah diterima dengan perintah

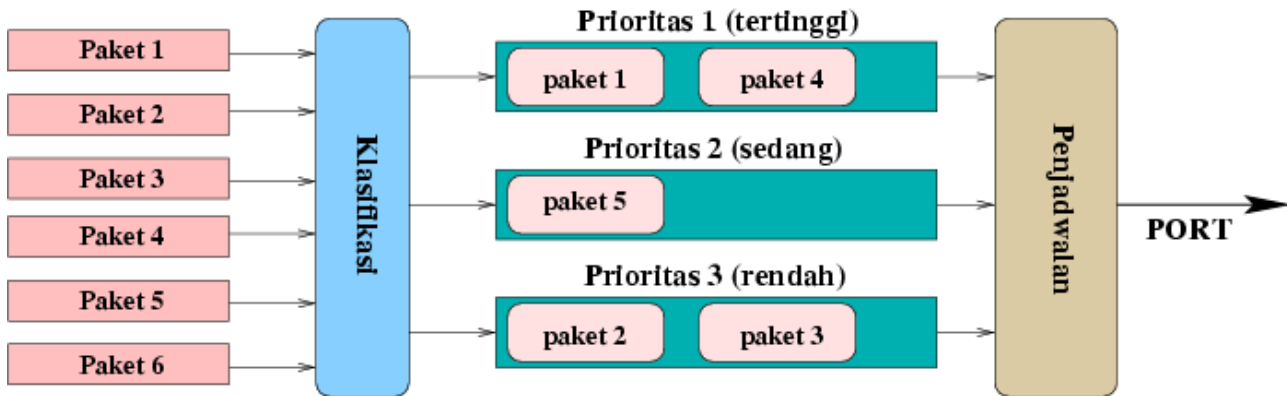
```
[root@server root]# tc -s -d qdisc ls dev eth0
qdisc pfifo 8001: limit 100p
  Sent 267399 bytes 318 pkts (dropped 0, overlimits 0)
```

Menghapus qdisc FIFO

```
[root@server root]# tc qdisc del dev eth0 root
```

Prioritas antrian

Pada situasi tertentu kadangkala kita harus memutuskan suatu permasalahan dengan memilih salah satu solusi yang perlu di laksanakan terlebih dahulu dan hal ini di sebut prioritas. Hal ini sama jika kita memiliki bandwidth internet yang sempit, sedangkan ada transaksi penting yang tidak boleh di interupsi seperti layanan email smtp dan pop3. Traffic controll pada linux memberikan kita suatu fasilitas pengendalian prioritas trafik TCP/IP, kita dapat memberikan prioritas berdasarkan alamat port, IP address, subnet maupun jenis type of service. Prioritas di tandai dengan angka numerik 1 sampai dengan 100 misalnya, prioritas tertinggi dimiliki oleh angka 1 kemudian di ikuti dengan angka berikutnya.



Gambar 3.1 prioritas antrian

Perintah untuk menentukan prioritas adalah sebagai berikut;

```
[root@pc01 root]# tc qdisc add dev eth0 root handle 1: prio
[root@pc01 root]# tc filter add dev eth0 parent 1:0 prio 1 protocol ip u32 \
> match ip dst 192.168.0.3 flowid 1:1
[root@pc01 root]# tc filter add dev eth0 parent 1:0 prio 2 protocol ip u32 \
> match ip dst 192.168.1.1 flowid 1:2
[root@pc01 root]# tc filter add dev eth0 parent 1:0 prio 3 protocol ip u32 \
> match ip dst 192.168.1.3 flowid 1:3
```

Paket yang telah diterima dapat kita lihat dengan perintah;

```
[root@pc01 root]# tc -s -d qdisc ls dev eth0
qdisc prio 1: bands 3 priomap 1 2 2 2 1 2 0 0 1 1 1 1 1 1 1
Sent 4324463 bytes 4655 pkts (dropped 0, overlimits 0)
```

Dari perintah diatas terlihat paket yang dikirimkan kartu ethernet 4324463 bytes dengan 4655 paket, filter yang lain tidak terlihat karena belum diklasifikasikan.

Stochastic Fairness Queuing (SFQ)

Stochastic Fairness Queuing (SFQ) memiliki kemampuan membagi setiap paket data yang diterima dalam jumlah yang sama rata, setiap paket data yang telah terbagi dimasukkan ke dalam suatu antrian dan menunggu dikeluarkan oleh penjadwalan, antrian dikeluarkan dengan algoritma round robin. Berikut ini adalah opsi dari teknik antrian SFQ.

perturb **nilai[detik]**

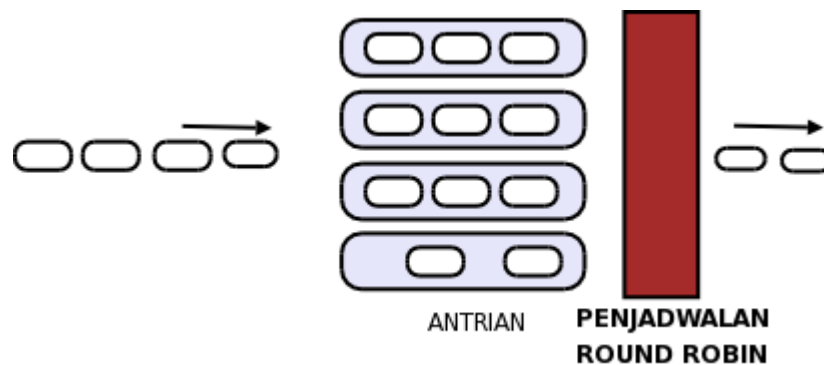
algoritma hash akan di konfigurasi ulang secara otomatis tergantung dari nilai yang diberikan [detik].

quantum **nilai[bytes]**

Jumlah paket data stream yang diizinkan untuk di keluarkan (dequeue) sebelum antrian lain diproses. Nilainya tidak boleh dibawah MTU (Maximum Transfer Unit), untuk ethernet 10/100Mbit nilai MTU nya 1500 bytes, periksa dengan perintah "ifconfig".

MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

linux.multimedia@gmail.com

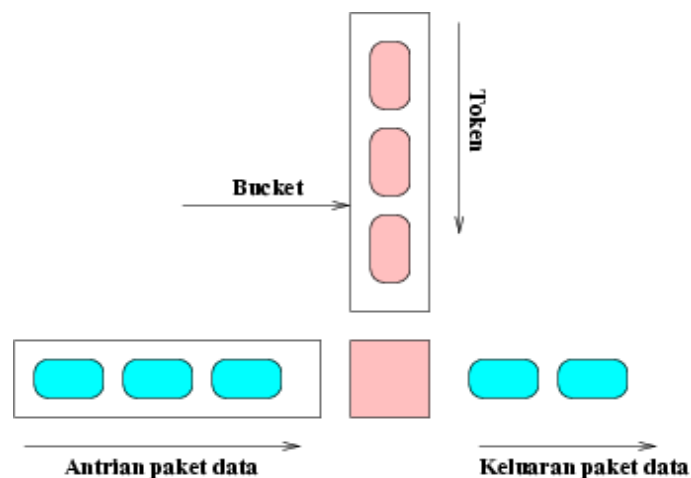


Contoh:

```
[root@server root]# tc qdisc add dev eth0 root handle 1: sfq perturb 10
```

Token bucket filter

Token bucket filter (TBF) membatasi bandwidth dengan metode shape & drop, prinsip kerja menggunakan aliran token yang memasuki bucket dengan kecepatan (rate) konstan, jika token dalam bucket habis maka paket data akan di antri dan kelebihannya dibuang, setiap paket data yang dikeluarkan identik dengan token. Token dalam bucket akan lebih cepat habis jika aliran paket data melampaui kecepatan token memasuki bucket, jadi kita asumsikan bahwa trafik melebihi batas konfigurasi.



Gambar 3.4. Token Bucket Filter

Perintah berikut ini menambahkan qdisc pada eth0 komputer gateway, bandwidth diset pada 64kbps (rate), kapasitas bucket (burst/buffer) 20kbit dan latency 70ms.

```
[root@server root]# tc qdisc add dev eth0 root handle 1:0 tbf rate 64kbit  
burst 20kb latency 70ms
```

```
[root@server root]# tc qdisc show dev eth0  
qdisc tbf 1: rate 64Kbit burst 2559b lat 85.4ms
```

```
[root@server root]# tc -s -d qdisc ls dev eth0  
qdisc tbf 1: rate 64Kbit burst 2559b/8 mpu 0b lat 85.4ms  
Sent 1350 bytes 11 pkts (dropped 0, overlimits 0)
```

MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

linux.multimedia@gmail.com

Paket yang dibuang karena overlimit (melebihi batas kapasitas bucket), memberikan efek aplikasi not responding pada MS win 9x/2000/NT/XP.

```
[root@server root]# tc -s -d qdisc ls dev eth0
qdisc tbf 1: rate 64Kbit burst 2559b/8 mpu 0b lat 85.4ms
Sent 724124 bytes 535 pkts (dropped 114, overlimits 810)
backlog 3028b 2p
```

Parameter TBF

rate

batas bandwidth yang di set oleh administrator, jika aliran paket data melebihi nilai ini maka data akan di buang (drop) atau mengalami penundaan, bandwidth dipotong.

Limit / latency

limit merupakan jumlah byte yang dapat diantri sebelum token tersedia, sedangkan latency adalah lama waktu (dalam mili detik [msec]) paket dapat diantri.

Burst/buffer/maxburst

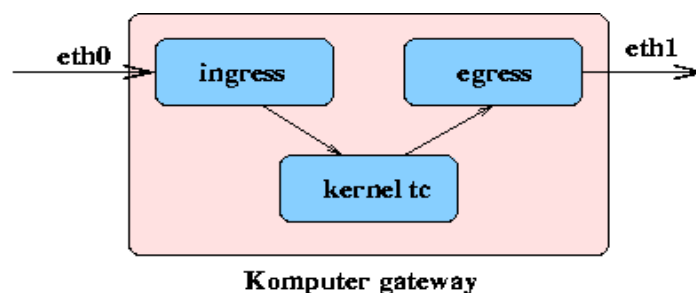
Kapasitas bucket dalam byte, paket data yang melebihi nilai ini akan dibuang atau mengalami penundaan.

Peakrate

Batas maksimum rate menangani lonjakan bandwidth sesaat dengan syarat paket data tidak boleh melebihi kapasitas bucket dan mtu.

Ingress

Paket data yang memasuki komputer gateway akan diproses terlebih dahulu oleh qdisc ingress, sebelum diberikan kernel traffic control. Jadi ingress merupakan teknik antrian untuk mengatur data yang memasuki kartu ethernet. Gambar 3.5 menunjukkan paket data diterima dari kartu ethernet 1 (eth0) kemudian dimasukkan ke dalam antrian ingress dan diproses oleh kernel traffic control. Dalam prakteknya ingress tidak mempengaruhi trafik yang memasuki gateway, mungkin ini bug kernel traffic control.



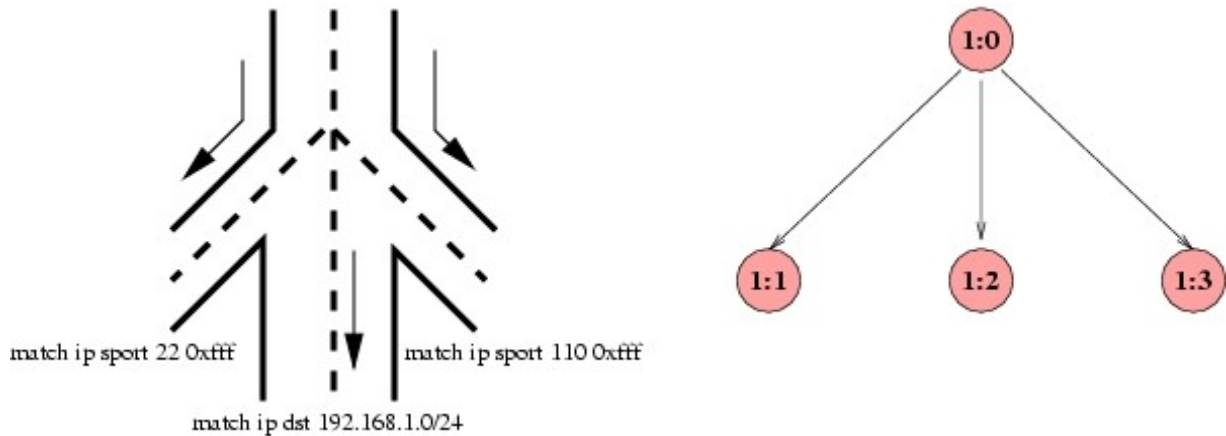
Gambar 3.5. antrian ingress dalam komputer gateway

Perintah untuk menambahkan ingress sebagai berikut;

```
[root@server root]# tc qdisc add dev eth0 handle ffff: ingress
[root@server root]# tc -s -d qdisc ls dev eth0
qdisc ingress ffff: -----
Sent 17468838 bytes 2615 pkts (dropped 0, overlimits 0)
```


Filtering

Paket data yang akan memasuki antrian melalui klasifikasi (class) sebelumnya disaring (filter) terlebih dahulu, agar setiap paket bisa ditentukan jenisnya, alamat IP, alamat PORT dan TOS nya. Kemudian fungsi yang kedua, filtering digunakan untuk mengarahkan suatu paket agar ke tujuan yang benar, ke klasifikasi paket (class) sesuai dengan arah alirannya, contoh;



Gambar 4.1. Filtering paket

Pada gambar 4.1. menunjukkan paket dibagi menjadi 3 bagian, untuk dari port ssh (22), alamat tujuan 192.168.1.0/24 dan dari port POP3 (110), sedangkan bilangan heksadesimal 0xffff menunjukkan semua paket dari sport xx. Perintah untuk membangun filtering paket sebagai berikut;

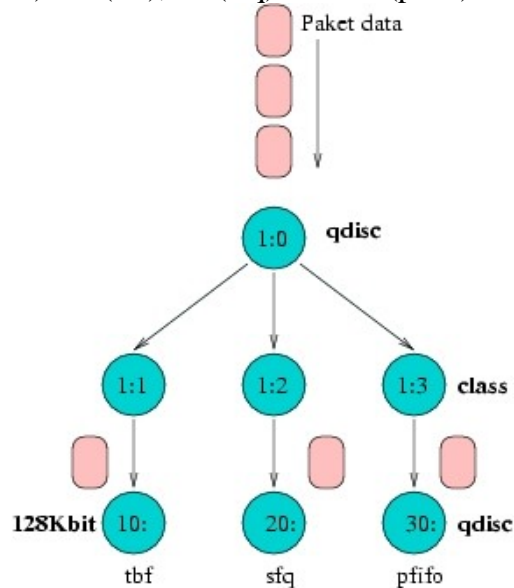
```
[root@server root]# tc filter add dev eth1 parent 1:0 protocol ip prio 1 u32
> match ip sport 22 0xffff flowid 1:1
[root@server root]# tc filter add dev eth1 parent 1:0 protocol ip prio 2 u32
match ip sport 110 0xffff flowid 1:3
[root@server root]# tc filter add dev eth1 parent 1:0 protocol ip prio 3 u32
match ip dst 192.168.1.0/24 flowid 1:2
```

Parameter “prio” digunakan untuk menentukan prioritas paket yang akan ditangani oleh teknik antrian (qdisc), angka 1 menunjukkan prioritas tertinggi dan 3 prioritas terendah.

Klasifikasi paket

Klasifikasi prioritas

Klasifikasi paket merupakan cara memberikan suatu kelas atau perbedaan pada setiap paket, hal ini dilakukan untuk mempermudah penanganan paket oleh antrian. Klasifikasi berbeda dengan filtering yang berfungsi mengarahkan dan menyaring aliran paket data. Contoh pada gambar 5.1. dibawah ini menunjukkan paket data dibagi menjadi tiga kelas 1:1, 1:2 dan 1:3 dan tiap kelas tersebut ditangani oleh teknik antrian (qdisc) 10: (tbf), 20:(sfq) dan 30: (pfifo).



Gambar 5.1. Klasifikasi paket data

Perintah ini secara otomatis membuat klasifikasi (class) paket 1:1,1:2 dan 1:3, dengan teknik antrian prioritas (PRIO), pada kartu ethernet 0 (eth0).

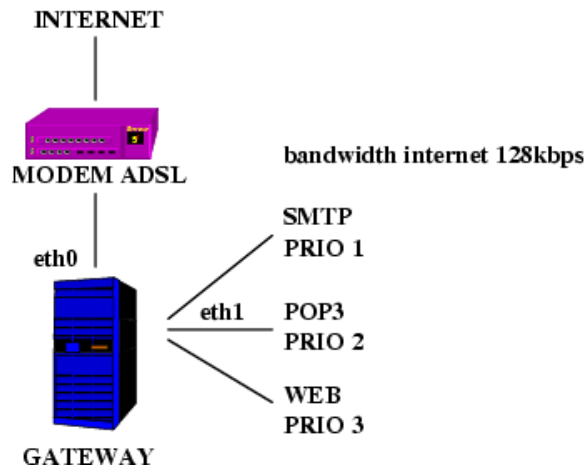
```
tc qdisc add dev eth0 root handle 1:0 prio
```

Menambahkan filter ke class 1:1, 1:2 dan 1:3 dari parent 1:0, bandwidth port ssh (22) dibatasi menjadi 128Kbps (128Kbit)

```
tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 match ip sport 22 0xffff
flowid 1:1
tc filter add dev eth0 parent 1:0 protocol ip prio 2 u32 match ip dst
192.168.0/24 flowid 1:2
tc filter add dev eth0 parent 1:0 protocol ip prio 3 u32 match ip sport 110 0xffff
flowid 1:3
```

```
tc qdisc add dev eth0 parent 1:1 handle 10: tbf rate 128kbps burst 10000 latency
70ms
tc qdisc add dev eth0 parent 1:2 handle 20: sfq
tc qdisc add dev eth0 parent 1:3 handle 30: pfifo
```

Contoh klasifikasi prioritas:



Alokasi bandwidth eth0

Service	Bandwidth			Class	
	Uplink	Internet (kbps)	Total (kbps)	Qdisc	PRIO
SMTP	30%	128	38,4	Bfifo	1
POP3	-	128	-	-	-
WEB	30%	128	38,4	Tbf	2
Bandwidth uplink yg terpakai			76,8		

Alokasi bandwidth eth1

Service	Bandwidth			Class	
	Downlink	Internet (kbps)	Total (kbps)	Qdisc	PRIO
SMTP	30%	128	38,4	Bfifo	1
POP3	25%	128	32	Bfifo	2
WEB	30%	128	38,4	Tbf	3
Bandwidth downlink yg terpakai			108,8		

Class Based Queue (CBQ)

Teknik klasifikasi paket data yang paling terkenal adalah CBQ, mudah dikonfigurasi, memungkinkan sharing bandwidth antar kelas (class) dan memiliki fasilitas user interface. 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.

```
[root@server root]# tc qdisc add dev eth0 root handle 1:0 cbq bandwidth 100Mbit avpkt 1000 cell 8
```

```
[root@server root]# tc class add dev eth0 parent 1:0 classid 1:1 cbq bandwidth 100Mbit rate 128kbit weight 12.8kbit avpkt 1000 sharing borrow
```

```
[root@server root]# tc class add dev eth0 parent 1:1 classid 1:2 cbq bandwidth 100Mbit
```

MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

linux.multimedia@gmail.com

```
rate 56kbit weight 5.6kbit avpkt 1000 sharing borrow prio 1
[root@server root]# tc class add dev eth0 parent 1:1 classid 1:3 cbq bandwidth 100Mbit
rate 56kbit weight 5.6kbit avpkt 1000 sharing borrow prio 2
```

```
[root@server root]# tc qdisc add dev eth0 parent 1:2 handle 20: sfq
[root@server root]# tc qdisc add dev eth0 parent 1:3 handle 30: sfq
```

```
[root@server root]# tc -s -d qdisc show dev eth0
[root@server root]# tc -s -d class show dev eth0
[root@server root]# tc -s -d filter show dev eth0
```

parameter CBQ:

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

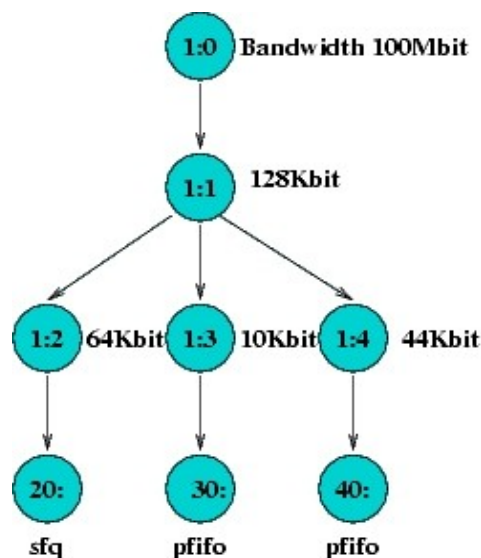
Peningkatan paket data yang dikeluarkan ke kartu ethernet berdasarkan jumlah byte, misalnya 800 ke 808 dengan nilai cell 8.

isolated / sharing

parameter isolated mengatur agar bandwidth tidak bisa dipinjam oleh kelas (class) lain yang sama tingkat / sibling. Parameter sharing menunjukkan bandwidth kelas (class) bisa dipinjam oleh kelas lain.

bounded / borrow

parameter borrow berarti kelas (class) dapat meminjam bandwidth dari kelas lain, sedangkan bounded berarti sebaliknya.



MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

linux.multimedia@gmail.com

Pembuatan antrian (queue) dan klasifikasi

```
[root@server root]# tc class add dev eth1 parent 1:0 classid 1:1 bandwidth 100Mbit rate
128Kbit avpkt 3000

[root@server root]# tc class add dev eth1 parent 1:0 classid 1:1 cbq bandwidth 100Mbit
rate 128Kbit avpkt 3000

[root@server root]# tc class add dev eth1 parent 1:1 classid 1:2 cbq bandwidth 100Mbit
rate 64Kbit avpkt 1000 sharing borrow

[root@server root]# tc class add dev eth1 parent 1:1 classid 1:3 cbq bandwidth 100Mbit
rate 10Kbit avpkt 1000 sharing borrow

[root@server root]# tc class add dev eth1 parent 1:1 classid 1:4 cbq bandwidth 100Mbit
rate 44Kbit avpkt 1000 bounded isolated

[root@server root]# tc qdisc add dev eth1 parent 1:2 handle 20: sfq perturb 10
[root@server root]# tc qdisc add dev eth1 parent 1:3 handle 30: pfifo
[root@server root]# tc qdisc add dev eth1 parent 1:4 handle 40: pfifo
```

cek konfigurasi dengan perintah

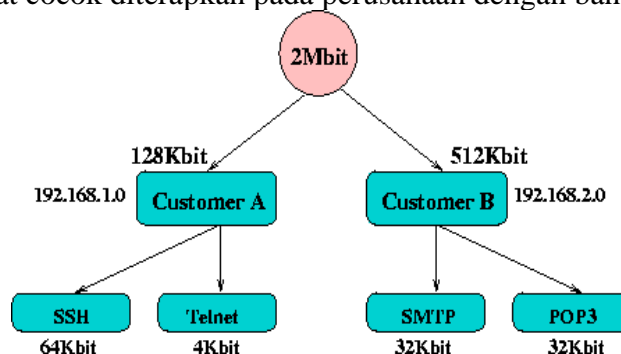
```
[root@server root]# tc -s -d qdisc show dev eth1
[root@server root]# tc -s -d class show dev eth1
```

Pembuatan filter

```
[root@server root]# tc filter add dev eth1 parent 1:0 protocol ip u32 match ip dst
192.168.1 flowid 1:2
[root@server root]# tc filter add dev eth1 parent 1:0 protocol ip u32 match ip dst
192.168.2 flowid 1:3
[root@server root]# tc filter add dev eth1 parent 1:0 protocol ip u32 match ip dst
192.168.3 flowid 1:4
[root@server root]# tc -s -d filter ls dev eth1
```

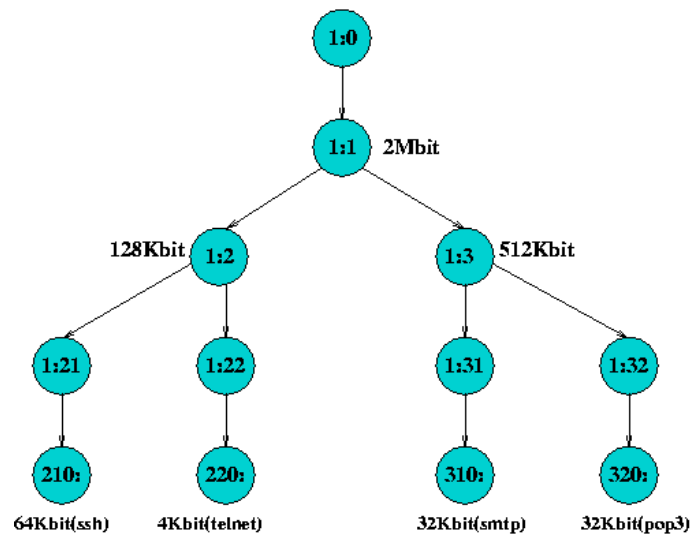
Hierarchy Token Bucket (HTB)

Teknik antrian HTB mirip dengan CBQ hanya perbedaannya terletak pada opsi, HTB lebih sedikit opsi saat konfigurasi serta lebih presisi. Teknik antrian HTB memberikan kita fasilitas pembatasan trafik pada setiap level maupun klasifikasi, bandwidth yang tidak terpakai bisa digunakan oleh klasifikasi yang lebih rendah. Kita juga dapat melihat HTB seperti suatu struktur organisasi dimana pada setiap bagian memiliki wewenang dan mampu membantu bagian lain yang memerlukan, teknik antrian HTB sangat cocok diterapkan pada perusahaan dengan banyak struktur organisasi.



MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

linux.multimedia@gmail.com



Contoh script htc.tc

```
CMD=tc
DEVICE=eth0
BANDWIDTH=2Mbit
CUSTOMER_A=128Kbit
CUSTOMER_B=512Kbit
SSH=64Kbit
TELNET=4Kbit
POP3=32Kbit
SMTP=32Kbit
QDISC_ADD="qdisc add dev"
FILTER_ADD="filter add dev"
CLASS_ADD="class add dev"
ROOT="root handle 1:0 htb"

$CMD $QDISC_ADD $DEVICE $ROOT
$CMD $CLASS_ADD $DEVICE parent 1:0 classid 1:1 htb rate 2Mbit ceil 2Mbit
$CMD $CLASS_ADD $DEVICE parent 1:1 classid 1:2 htb rate 128Kbit ceil 128Kbit
$CMD $CLASS_ADD $DEVICE parent 1:1 classid 1:3 htb rate 512Kbit ceil 512Kbit

$CMD $CLASS_ADD $DEVICE parent 1:2 classid 1:21 htb rate 64Kbit ceil 64Kbit
$CMD $CLASS_ADD $DEVICE parent 1:2 classid 1:22 htb rate 4Kbit ceil 128Kbit

$CMD $CLASS_ADD $DEVICE parent 1:3 classid 1:31 htb rate 32Kbit ceil 32Kbit
$CMD $CLASS_ADD $DEVICE parent 1:3 classid 1:32 htb rate 32Kbit ceil 32Kbit

$CMD $QDISC_ADD $DEVICE parent 1:21 handle 210: pfifo limit 10
$CMD $QDISC_ADD $DEVICE parent 1:22 handle 220: pfifo limit 10

$CMD $QDISC_ADD $DEVICE parent 1:31 handle 310: pfifo limit 10
$CMD $QDISC_ADD $DEVICE parent 1:32 handle 320: pfifo limit 10

$CMD $FILTER_ADD $DEVICE parent 1:0 protocol ip u32 match ip dst 192.168.1/24
match ip sport 22 0xff flowid 1:21
$CMD $FILTER_ADD $DEVICE parent 1:0 protocol ip u32 match ip dst 192.168.1/24
match ip sport 23 0xff flowid 1:22

$CMD $FILTER_ADD $DEVICE parent 1:0 protocol ip u32 match ip dst 192.168.2/24
match ip sport 25 0xff flowid 1:31
$CMD $FILTER_ADD $DEVICE parent 1:0 protocol ip u32 match ip dst 192.168.2/24
match ip sport 110 0xff flowid 1:32
```

Eksekusi script htb.tc

```
chmod 700 htb.tc
./htb.tc
```

Parameter HTB

rate

parameter rate menentukan bandwidth maksimum yang bisa dipakai oleh setiap class, jika bandwidth melebihi nilai "rate" maka paket data akan dipotong atau di jatuhkan (drop)

ceil

parameter ceil di set untuk menentukan peminjaman bandwidth antar class (kelas), peminjaman bandwidth dilakukan class paling bawah ke kelas di atasnya, teknik ini disebut link sharing.

Random Early Detection

Random Early Detection atau bisa disebut Random Early Drop biasanya dipergunakan 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. Parameter RED sebagai berikut;

min

Nilai rata – rata minimum antrian (queue)

max

Nilai rata – rata maksimum antrian, biasanya dua kali nilai minimum atau dengan rumus;

$$\text{max} = \text{bandwidth [Bps]} * \text{latency [sec]}$$

probability

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

limit

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

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

burst

digunakan untuk menentukan kecepatan perhitungan nilai antrian mempengaruhi antrian riil (limit). Untuk praktek nilainya kita set dengan rumus;

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

avpkt

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

MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

linux.multimedia@gmail.com

bandwidth

Lebar band (bandwidth) kartu ethernet.

ecn

Explicit Congestion Notification memberikan fasilitas gateway RED untuk memberitahukan kepada klien jika terjadi kemacetan.

```
tc qdisc add dev eth0 root limit <bytes> min <bytes> max <bytes> \
  avpkt <bytes> burst <packets> probability <float> bandwidth <kbps> ecn
```

Contoh:

1. Kantor PT. Linux menyewa bandwidth dari ISP sebesar 128kbps ditarik melalui kabel leased line telpon ke modem ADSL yang terhubung dengan komputer gateway RED, pada saat jaringan sedang sibuk nilai latency nya kita set 500msec. Buatlah setting RED pada komputer gateway ?

```
Bandwidth = 128kbps = 16000 Byte / sec
Latency = 500msec = 0.5 sec
Max      = Bandwidth * Latency
          = 16000 * 0.5
          = 8000
```

```
Min      = Max / 2
          = 4000
```

```
Limit    = 8 * max
          = 8 * 8000
          = 64000
```

```
Burst    = (2 * min + max) / (3 * avpkt)
          = (2 * 4000 + 8000) / (3 * 1000)
          = 5.33
```

Jadi setting RED telah kita dapatkan dan ketikkan perintah;

```
tc qdisc add dev eth0 root red limit 64000 min 4000 max 8000 burst 5.33 avpkt
1000 \
probability 0.02 bandwidth 128
```

```
tc -s -d qdisc show dev eth0
qdisc red 8001: limit 64000b min 4000b max 8000b ewma 2 Plog 18 Scell_log 24
  Sent 3456897 bytes 4973 pkts (dropped 163, overlimits 163)
  marked 0 early 163 pdrop 0 other 0
```


Policing & packet marking

Device egress mengendalikan paket data yang keluar dari ethernet tapi tidak bisa membatasi jumlah rata – rata paket yang memasuki ethernet. Policing mengendalikan paket data yang memasuki kartu ethernet melalui device ingress, administrator memberikan batas maksimum paket yang boleh melewati ingress dan policing akan membuang paket jika melebihi batas maksimum konfigurasi. Setting policing dapat dilakukan dengan dua cara, pertama dengan iptables mangle dan kedua melalui cara manual. Pada iptables mangle paket yang memasuki device ingress diberikan tanda dengan opsi mangle dari iptables, paket tersebut kemudian diproses oleh policing. Pada cara manual paket data di filter dengan opsi “match ip” secara langsung.

Klasifikasi paket data pada policing ada beberapa macam seperti berikut;

fw

Menggunakan iptables mangle untuk klasifikasi paket data.

route

Menggunakan ip route untuk klasifikasi paket data yang akan dikirimkan / di route ke lain subnet.

Policing paket data

drop

Paket data akan di buang atau di antrikan jika melampaui batas maksimum konfigurasi.

continue

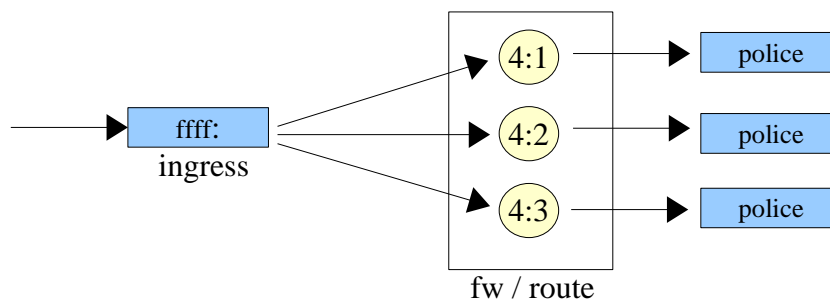
Paket data jika melebihi batas maksimum konfigurasi akan di berikan ke klasifikasi berikutnya, jadi kita dapat membuat skala bandwidth pada device ingress.

pass / ok

Paket data yang melampaui batas maksimum konfigurasi boleh melintas.

reclassify

Paket data di klasifikasi oleh kelas berikutnya, ini pilihan default.



Policing & packet marking

MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

linux.multimedia@gmail.com

Cara iptables mangle

```
iptables -t mangle -A PREROUTING -i eth0 -p tcp --dport 22 -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -i eth0 -p tcp --dport 21 -j MARK --set-mark 2
tc qdisc add dev eth0 handle ffff: ingress
tc filter add dev eth0 parent ffff: protocol ip prio 1 handle 1 fw police rate
56kbit burst 90k \
continue flowid 4:1
tc filter add dev eth0 parent ffff: protocol ip prio 1 handle 1 fw police rate
128kbit burst 100k drop flowid 4:2
tc filter add dev eth0 parent ffff: protocol ip prio 2 handle 2 fw police rate
64kbit burst 100k continue flowid 4:3
tc filter add dev eth0 parent ffff: protocol ip prio 2 handle 2 fw police rate
128kbit burst 100k drop flowid 4:4
```

Cara manual

```
tc qdisc add dev eth0 handle ffff: ingress
tc filter add dev eth0 parent ffff: protocol ip prio 1 u32 match ip dport 22
police rate 56kbit burst 90k continue flowid 4:1
tc filter add dev eth0 parent ffff: protocol ip prio 1 u32 match ip dport 22
police rate 128kbit 100k drop flowid 4:2
tc filter add dev eth0 parent ffff: protocol ip prio 2 u32 match ip dport 21
police rate 64kbit 100k continue flowid 4:3
tc filter add dev eth0 parent ffff: protocol ip prio 2 u32 match ip dport 21
police rate 128kbit 100k drop flowid 4:4
```

Traffic Control Next Generation (TCNG)

TCNG dibuat oleh Werner Almesberger dengan lisensi GNU/GPL, jadi orang boleh menggunakan software tcng tanpa membayar lisensi. Software TCNG berfungsi sebagai kompilator, file text dengan format tcng dirubah ke script tc (traffic control), sehingga memudahkan kita untuk menyusun rule antrian / traffic control. Syntax pemrograman tcng mirip dengan bahasa C, PERL ataupun java, bagi yang sudah terbiasa pasti tidak akan kesulitan.

Instalasi

Software tcng dapat didownload di url <http://tcng.sourceforge.net> dengan nama file [tcng-9l.tar.gz](#). File tersebut berisi kompilator tcc dan tcsim (traffic control simulator), tcsim tidak dibahas pada bagian ini. Langkah untuk instalasi sebagai berikut;

```
tar xzvf tcng-9l.tar.gz
cd tcng
./configure --no-tcsim
make
make install
```

Bahasa TCNG

Syntax dasar

Syntax tcng mirip dengan bahasa C, terdiri atas file header, macros / preprocessor dan bagian utama. Untuk lebih jelasnya lihat contoh dibawah ini;

```
#include "fields.tc"
#include "ports.tc"
```

MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

linux.multimedia@gmail.com

```
dev eth0{
    egress{
        class(<$customer1>)
            if ip_dst == 192.168.0.1;
        class(<$customer2>)
            if ip_dst == 192.168.0.2;
        htb() {
            class(rate 2Mbps, ceil 2Mbps) {
                $customer1 = class(rate 128kbps, ceil 128kbps) {sfq;};
                $customer2 = class(rate 64kbps, ceil 128kbps) {sfq;};
            }
        }
    }
}
```

Interfaces

Pada bagian utama tcng harus disertakan perangkat jaringan (interfaces) yang akan dikendalikan aliran trafiknya. Syntax dasar interfaces harus diawali dengan “dev” kemudian diikuti nama perangkat jaringan.

```
dev eth0 {
    .....
    isi program
}
```

Teknik antrian

Setiap perangkat jaringan memiliki teknik antrian ingress dan egress, dimana ingress mengatur paket data yang memasuki kartu ethernet, sedangkan egress mengendalikan keluaran paket data.

```
dev eth0{
    egress{
        class(<$test1>)
            if ip_dst == 192.168.0/24;
        class(<$test2>)
            if ip_dst == 192.168.0/24;
        prio{
            $test1 = class(1) { fifo(limit 20kB);}
            $test2 = class(1) { fifo(limit 20kB);}
        }
    }
}
```

Klasifikasi

Seperti pada perintah manual tc, tcng memiliki syntax dasar untuk meng-klasifikasikan paket data yang melintasi perangkat jaringan.

```
class(<$ssh>)
    if tcp_sport == 22;
class(<$pop3>)
    if tcp_sport == 110;
prio{
    $test1 = class(1) { fifo(limit 20kB);}
    $test2 = class(1) { fifo(limit 20kB);}
}
```

Konstruktor if

Konstruktor / statement if dipergunakan untuk mengendalikan aliran trafik agar memasuki kelas (class) yang benar;

```
class(<$test1>
    if ip_dst == 192.168.0/24;
```

Konstruktor drop

Konstruktor drop dipergunakan untuk menghentikan paket data yang sesuai dengan konstruktor if, penggunaannya sebagai berikut;

```
class(<$test1>
    if ip_dst == 192.168.0.1 && drop ;
atau
drop if ip_dst == 192.168.0.1;
```

Variabel

Pada bahasa pemrograman fungsi variabel digunakan untuk menyimpan data yang akan dipakai lagi. Syntax variabel tng mirip dengan PERL diawali dengan karakter dolar "\$" kemudian diikuti nama variabel;

```
$rate = 10Mbps
$gateway = "192.168.1.1";
$hostname = "www.linux.org";
```

Parameter

Beberapa teknik antrian seperti sfq, bfifo,tbf dan htb meminta admin untuk memasukkan parameter yang diperlukan. Masukan berupa parameter diawali dengan nama teknik antrian " cbq/htb" kemudian diikuti dengan karakter kurung buka "(" dan tutup ")", lihat contoh dibawah ini;

```
tc qdisc add dev eth0 parent 1:0 handle 1:1 htb rate 2Mbps ceil 2Mbps
menjadi
htb(rate 2Mbps, ceil 2Mbps)

tc qdisc add dev eth0 parent 1:2 handle 20: sfq perturb
menjadi
..... {sfq(perturb 10s)};
```

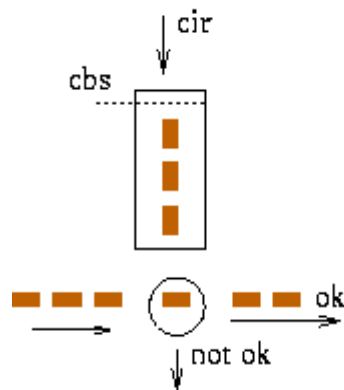
Traffic metering

Traffic metering digunakan untuk mengukur paket yang melewati jaringan pada kondisi tertentu, kemudian hasilnya diperlukan untuk membuat skala bandwidth pada QoS atau memberikan penanganan lebih lanjut terhadap trafik paket data. Pada pelatihan ini kita bahas tiga metode traffic metering yaitu; single leak bucket, single rate three color meter & double rate three color meter.

Single Leaky Bucket

Traffic metering Single Leak Bucket (SLB) mengukur kecepatan aliran (flow) paket data sesuai dengan kecepatan datangnya token (cir) mengisi bucket, setiap paket data yang melintas keluar akan mngurangi jumlah token yang berada di dalam bucket. Jika kecepatan aliran paket data melebihi

kecepatan rata – rata datangnya token dalam bucket sampai mencapai nilai “cbs” maka token akan segera habis.



Single Leak Bucket (SLB)

```
#include "fields.tc"
#include "meters.tc"
#include "ports.tc"
$pop3 = SLB(cir 64kbps, cbs 20kB);
dev eth0{
    egress{
        class(<$normal>)
            if tcp_sport == 110 && SLB_ok($pop3);
        drop if tcp_sport == 110 && SLB_else_drop($pop3);
        htb() {
            class(rate 2Mbps, ceil 2Mbps) {
                $normal = class(rate 64kbps, ceil 64kbps)
            }
        }
    }
}
```

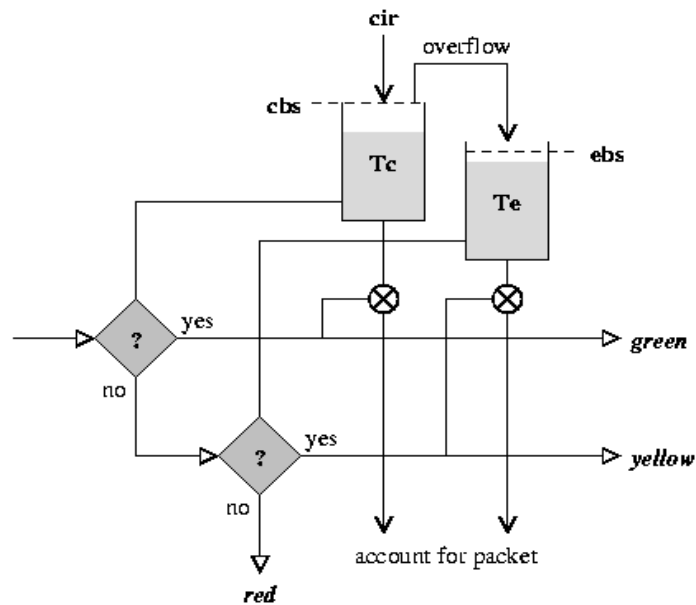
Single Rate Three Color Meter

Traffic metering jenis srTCM memiliki tiga warna untuk membedakan setiap aliran paket data, hijau (green), kuning (yellow) dan merah (red). Bucket Te menerima limpahan dari Tc saat trafik data lebih kecil dari cir (committed information rate) dan meter menunjuk warna hijau (green), saat trafik data melebihi nilai cir maka token yang berada dalam Tc habis, kemudian untuk melewatkan paket data dipergunakan token dari bucket Te dan meter menunjuk warna kuning. Jika token dalam bucket Tc dan Te habis maka tidak ada trafik data keluar sehingga meter menunjuk warna merah.

```
#include "fields.tc"
```

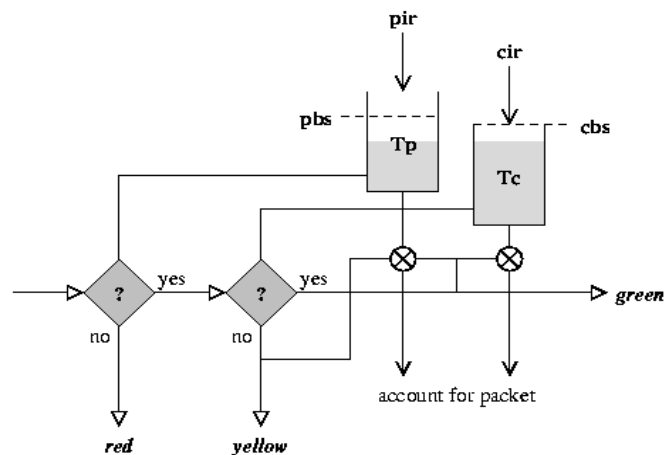
MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

linux.multimedia@gmail.com



```
#include "meters.tc"
$meter = srTCM(cir 64kbps, cbs 50kB, ebs 40kB);
dev eth0{
    egress{
        class(<$setting>)
        if tcp_sport == 22 && srTCM_green($meter);
        class(<$bawah>)
        if tcp_sport == 22 && srTCM_yellow($meter);
        htb() {
            class(rate 2Mbps, ceil 2Mbps) {
                $setting = class(rate 64kbps, ceil 64kbps) {sfq;};
                $bawah = class(rate 32kbps, ceil 32kbps) {sfq;};
            }
        }
    }
}
```

Double Rate Three Color Meter



```
#include "fields.tc"
#include "meters.tc"

$meter = trTCM(cir 64kbps, cbs 50kB, pir 128kbps, pbs 70kB);
```

MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

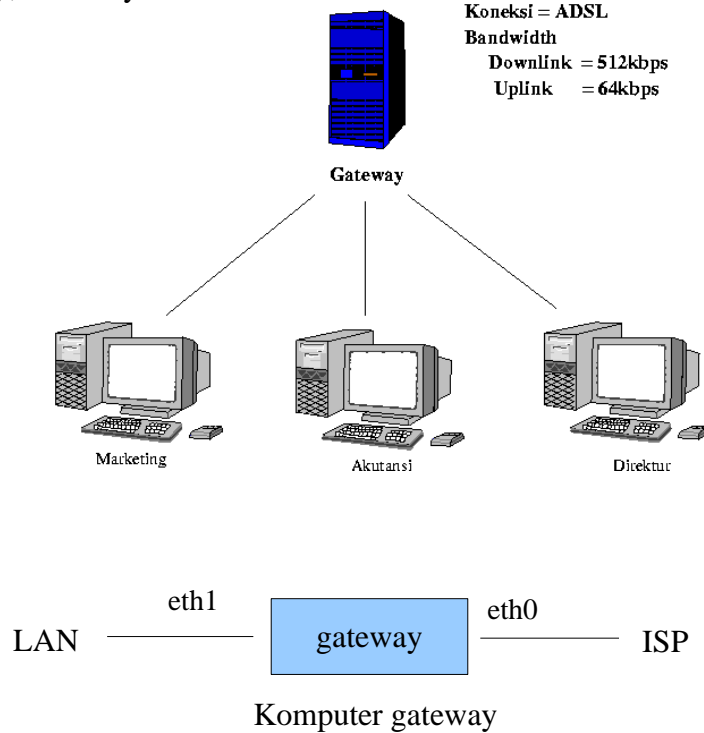
linux.multimedia@gmail.com

```
dev eth0{
    egress{
        class(<$setting>)
        if tcp_sport == 22 && __trTCM_green($meter);
        class(<$sedang>)
        if tcp_sport == 22 && __trTCM_yellow($meter);
        htb() {
            class(rate 128kbps, ceil 128kbps) {
                $sedang = class(rate 64kbps, ceil 64kbps) {sfq;};
                $setting = class(rate 32kbps, ceil 32kbps) {sfq;};
            }
        }
    }
}
```

Implementasi QoS

Small Office Home Office

Small Office Home Office atau lebih dikenal sebagai SOHO, pada jaringan ini rata – rata komputer klien 50 buah tetapi memiliki load bandwidth yang tinggi akibat adanya transaksi bisnis melalui jaringan internet. Transaksi bisnis biasanya memakai http (web browser), transaksi email smtp dan pop3, sehingga pengendalian bandwidth kita fokuskan pada tiga tersebut. Sebelum kita melakukan setting pada gateway, sebaiknya terlebih dahulu dibuat tabel fasilitas dan alokasi bandwidth.



Tabel fasilitas dan alokasi bandwidth

<i>Fasilitas</i>	<i>Maks user</i>	<i>Alokasi</i>	<i>Status</i>	<i>Bandwidth (kbps)</i>	<i>Total (kbps)</i>
HTTP	30	25%		512	128
SMTP	50	75%		64	48
POP3	50	15%		512	76,8
FTP	20	10%		512	51,2
Other	10	15%		512	76,8
Squid	50	64%		10000	6400

MANAJEMEN BANDWIDTH INTERNET DAN INTRANET
linux.multimedia@gmail.com

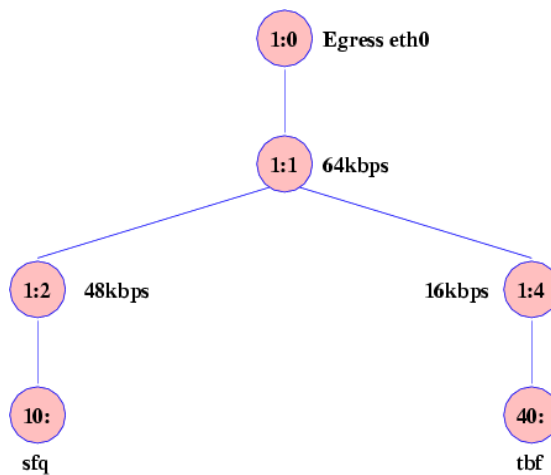
Tabel QoS

Fasilitas		eth1 (kbps)		eth0 (kbps)		Qdisc	Parameter
Nama	Port	egress	ingress	egress	ingress		
SMTP	25	128	128	48	48	sfq	
POP3	110	76,8	76,8	-	76,8	sfq	
FTP	21	51,2	51,2	-	51,2	bfifo	
HTTP	80	128	128	16	128	tbw	
Squid	8080	128	128	-	-	tbw	
Other		76,8	76,8	76,8	76,8	pfifo	

Konfigurasi QoS

Perangkat eth0

Egress



```

tc qdisc add dev eth0 root handle 1:0 htb
tc class add dev eth0 parent 1:0 classid 1:1 htb rate 64kbit ceil 64kbit
tc class add dev eth0 parent 1:1 classid 1:2 htb rate 48kbit ceil 64kbit
tc class add dev eth0 parent 1:1 classid 1:4 htb rate 16kbit ceil 64kbit

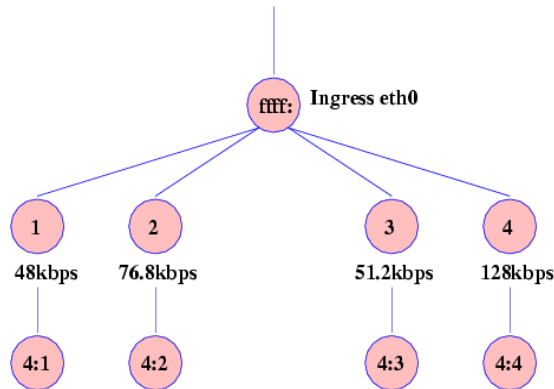
tc qdisc add dev eth0 parent 1:2 handle 10: sfq
tc qdisc add dev eth0 parent 1:4 handle 40: tbw rate 16kbit burst 20kbit latency 70ms

tc filter add dev eth0 parent 1:0 protocol ip u32 match ip sport 25 0xffff flowid 1:2
tc filter add dev eth0 parent 1:0 protocol ip u32 match ip sport 80 0xffff flowid 1:4
  
```

MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

linux.multimedia@gmail.com

Ingress

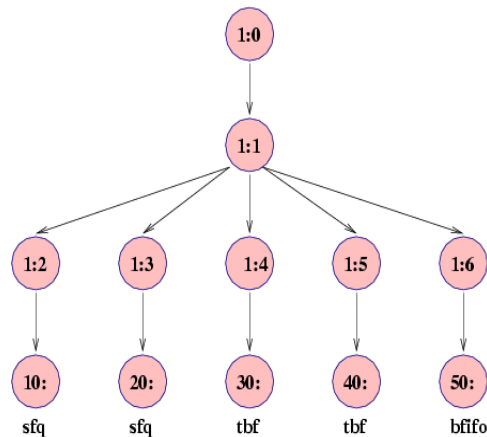


```
iptables -t mangle -A PREROUTING -i eth0 -p tcp --sport 25 -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -i eth0 -p tcp --sport 110 -j MARK --set-mark 2
iptables -t mangle -A PREROUTING -i eth0 -p tcp --sport 21 -j MARK --set-mark 3
iptables -t mangle -A PREROUTING -i eth0 -p tcp --sport 80 -j MARK --set-mark 4
```

```
tc qdisc add dev eth0 ingress handle ffff:
tc filter add dev eth0 parent ffff: protocol ip prio 1 handle 1 fw police rate 48kbit
burst 50k drop flowid 4:1
tc filter add dev eth0 parent ffff: protocol ip prio 2 handle 2 fw police rate 77kbit
burst 50k drop flowid 4:2
tc filter add dev eth0 parent ffff: protocol ip prio 3 handle 3 fw police rate 51kbit
burst 50k drop flowid 4:3
tc filter add dev eth0 parent ffff: protocol ip prio 4 handle 4 fw police rate 128kbit
burst 50k drop flowid 4:4
```

Perangkat eth1

Egress



```
tc qdisc add dev eth1 root handle 1:0 htb
tc class add dev eth1 parent 1:0 classid 1:1 htb rate 512kbit ceil 512kbit
tc class add dev eth1 parent 1:1 classid 1:2 htb rate 128kbit ceil 512kbit
tc class add dev eth1 parent 1:1 classid 1:3 htb rate 76kbit ceil 512kbit
tc class add dev eth1 parent 1:1 classid 1:4 htb rate 128kbit ceil 512kbit
tc class add dev eth1 parent 1:1 classid 1:5 htb rate 128kbit ceil 512kbit
tc class add dev eth1 parent 1:1 classid 1:6 htb rate 51kbit ceil 512kbit
```

```
tc qdisc add dev eth1 parent 1:2 handle 10: sfq
tc qdisc add dev eth1 parent 1:3 handle 20: sfq
tc qdisc add dev eth1 parent 1:4 handle 30: tbf rate 128kbit burst 100kb latency 70ms
```

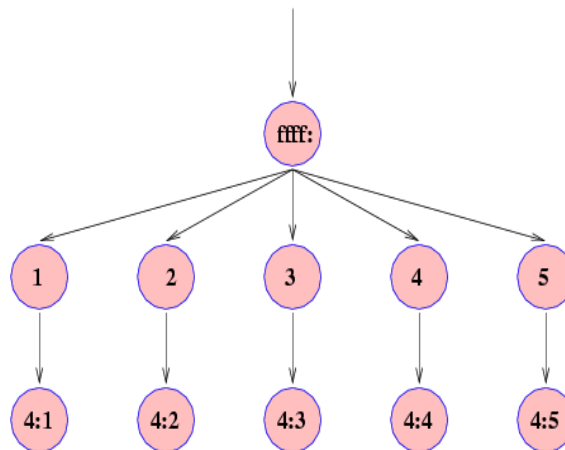
MANAJEMEN BANDWIDTH INTERNET DAN INTRANET

linux.multimedia@gmail.com

```
tc qdisc add dev eth1 parent 1:5 handle 40: tbf rate 128kbit burst 100kb latency 70ms
tc qdisc add dev eth1 parent 1:6 handle 50: bfifo
```

```
tc filter add dev eth1 parent 1:0 protocol ip u32 match ip sport 25 0xffff flowid 1:2
tc filter add dev eth1 parent 1:0 protocol ip u32 match ip sport 110 0xffff flowid 1:3
tc filter add dev eth1 parent 1:0 protocol ip u32 match ip sport 80 0xffff flowid 1:4
tc filter add dev eth1 parent 1:0 protocol ip u32 match ip sport 8080 0xffff flowid 1:5
tc filter add dev eth1 parent 1:0 protocol ip u32 match ip sport 21 0xffff flowid 1:6
```

Ingress



```
iptables -t mangle -A PREROUTING -i eth0 -p tcp --dport 25 -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -i eth0 -p tcp --dport 110 -j MARK --set-mark 2
iptables -t mangle -A PREROUTING -i eth0 -p tcp --dport 80 -j MARK --set-mark 3
iptables -t mangle -A PREROUTING -i eth0 -p tcp --dport 21 -j MARK --set-mark 4
iptables -t mangle -A PREROUTING -i eth0 -p tcp --dport 8080 -j MARK --set-mark 5
```

```
tc qdisc add dev eth0 handle ffff: ingress
tc filter add dev eth0 parent ffff: protocol ip prio 1 handle 1 fw police rate 128kbit
burst 100kb drop flowid 4:1
tc filter add dev eth0 parent ffff: protocol ip prio 1 handle 2 fw police rate 76kbit
burst 90kb drop flowid 4:2
tc filter add dev eth0 parent ffff: protocol ip prio 1 handle 3 fw police rate 128kbit
burst 78kb drop flowid 4:3
tc filter add dev eth0 parent ffff: protocol ip prio 1 handle 4 fw police rate 51kbit
burst 65kb drop flowid 4:4
tc filter add dev eth0 parent ffff: protocol ip prio 1 handle 5 fw police rate 128kbit
burst 100kb drop flowid 4:5
```

RINCIAN BIAYA KURSUS LINUX IN HOUSE TRAINING

Kursus	Durasi	Biaya / peserta
Linux professional full package	30 jam	Rp2.890.000
Linux professional desktop	12 jam	Rp700.000
Linux system administrator	18 jam	Rp900.000
Linux networking & security	30 jam	Rp1.180.000
Manajemen bandwidth intranet (QoS)	12 jam	Rp700.000
Kursus on request	-	Rp0
Linux Enterprise Mail Server	8 jam	Rp650.000
Linux Samba PDC & Antivirus Solution	8 jam	Rp650.000
Linux Antivirus Gateway (NEW)	8 jam	Rp650.000
High Availability Linux Cluster (NEW)	8 jam	-

PAKET WORKSHOP LINUX ANTIVIRUS SOLUTION

Kursus	Durasi	Biaya / peserta	Keterangan
Linux Enterprise Mail Server	8 jam	Rp650.000	Integrasi mail server smtp dan pop3 dengan antivirus clamav. Scan email incoming dan outgoing, mencegah penyebaran virus dengan double scanning.
Linux Samba PDC & Antivirus Solution	8 jam	Rp650.000	Integrasi samba PDC dengan antivirus clamav. File maupun direktori yang akan di baca / tulis di scan terlebih dahulu. Mencegah penyebaran virus dan worm melalui file sharing.
Linux Antivirus Gateway (NEW)	8 jam	Rp650.000	Integrasi squid proxy server dengan antivirus clamav, situs web yang dibuka oleh user terlebih dahulu discan oleh antivirus clamav. Mencegah penyebaran virus melalui html, javascript, active-x, gif animation dan worm jpeg.

Catatan:

Pelatihan paket workshop khusus untuk system administrator yang sudah biasa menjalankan perintah dasar dan jaringan linux. Paket workshop antivirus solution di tujukan untuk mencegah penyebaran virus melalui media internet dan intranet terutama file sharing.

Silabus kursus linux in house training

Linux professional desktop

- | | |
|---|--|
| <ul style="list-style-type: none"> - sekilas linux - pengenalan linux mandrake 10.0 desktop - User dan password - Konfigurasi desktop - Konfigurasi perangkat keras (hardware) - Filesystem - Aplikasi perkantoran - Openoffice - Multimedia | <ul style="list-style-type: none"> - ODBC connection - Report database & mail merge - Windows dan DOS emulator - Aplikasi konverter database - Koneksi internet web browser & email - Aplikasi grafis GIMP, Xfig & DIA - Lyx latex editor - Rendering 3D kpovray modeller - Linux development tools |
|---|--|

Linux system administrator

- | | |
|---|---|
| <ul style="list-style-type: none"> - Pengenalan administrasi OS linux - Standar hirarki filesystem linux - Informasi proses - Kompilasi kernel linux - Shell bash programming - Administrasi user dan group - Quota user dan group - Monitoring sistem operasi - Instalasi linux dan paket rpm - Remote maintenance | <ul style="list-style-type: none"> - Direktori dan file - Manajemen filesystem - Penggunaan editor Vi - Run Level - Linux Loader (LILO) - Backup - Manajemen proses - System log linux - Konsep TCP / IP - Perintah dasar linux |
|---|---|

Linux networking & security

- | | |
|--|--|
| <ul style="list-style-type: none"> - LAN dan internet - Resolusi alamat - Service daemon dan super server xinetd - Samba file server - Samba domain controller - Squid proxy server - Apache web server - Firewall & NAT - Koneksi internet + gateway - Sendmail mail server | <ul style="list-style-type: none"> - POP3, IMAP & webmail - PostgreSQL - DNS bind - DHCP - FTP server - Network Filesystem (NFS) - OpenLDAP - Keamanan jaringan - Port scanner - Web scanner - Packet capture |
|--|--|

Manajemen bandwidth intranet (QoS)

- | | |
|---|---|
| <ul style="list-style-type: none"> - Pengenalan QoS - Pengendalian trafik jaringan - Teknik antrian - FIFO & SFQ - Random Early Detection (RED) - General RED - DSMARK | <ul style="list-style-type: none"> - Token Bucket Filter (TBF) - Filtering - Klasifikasi paket - Class Based Queue (CBQ) - Hierarchy Token Bucket (HTB) - TCNG / Traffic Control Compiler programming - Implementasi manajemen bandwidth |
|---|---|

LINUX ENTERPRISE MAIL SERVER (MODUL WORKSHOP)

Integrated E-mail Server With Antivirus Clamav

- Postfix SMTP Server
- IMAP4 dan POP3
- Webmail client
- Spamassassin
- Antivirus Clamav
- SMTP antivirus scanner
- POP3 antivirus scanner
- Pengujian integrasi antivirus

- Email statistik dan monitoring

LINUX SAMBA PDC & ANTIVIRUS SOLUTION (MODUL WORKSHOP)

- Samba File Server
- Samba Primary Domain Controller
- Manajemen user samba
- Windows Internet Name Server
- WINS Proxy
- Windows Logon dan konfigurasi MS Windows 9x/XP

- Antivirus clamav 0.86
- Integrasi samba dengan antivirus clamav
- Pengujian integrasi antivirus samba pdc

LINUX ANTIVIRUS GATEWAY (MODUL WORKSHOP)

- Koneksi internet
- Firewall IPTables dan NAT
- Squid proxy server
- Clamav antivirus 0.86
- Dansguardian antivirus (http av scanner)
- Konfigurasi dansguardian

- Pengujian http antivirus scanner
- NTOP bandwidth monitoring

HIGH AVAILABILITY LINUX CLUSTER (MODUL WORKSHOP)

- Linux PC Cluster
- Konfigurasi IP Address
- Pengkabelan server cluster
- Pengujian koneksi TCP/IP
- Instalasi dan konfigurasi heartbeat

- IP Take Over
- Service/Server Take Over
- Resource release (IP Failback)
- Implementasi HA Linux Cluster untuk server internet

Untuk informasi lebih lanjut hubungi:
Budi Santosa (0852 1616 2008 / 0251 – 617 854)
Yahoo Messenger : budi_santosa24
ICQ : 229 529 810
Email : linux.multimedia@gmail.com