



Analisis kinerja *leaky bucket* terintegrasi pada jaringan ATM

F. Ahmadi Djajasugita*, Sony Sumaryo**

*Staf pengajar Jurusan Teknik Elektro ITB, **Staf pengajar STT Telkom Jl. Radio Palasari 31, Bandung.

Masuk: Maret 1998; revisi masuk: Januari 1999; diterima: Januari 1999

Sari

Pada makalah ini dibahas analisis kinerja *leaky bucket* terintegrasi pada jaringan ATM dengan kombinasi kontrol adaptif, metode *marking* dan metode prioritas. Juga ditinjau kinerja *leaky bucket* hasil *dimensioning*. Sebagai ukuran kinerja digunakan peluang sel hilang di jaringan dan *throughput* jaringan. Trafik masukan diasumsikan model 2-phase MMBP. Penelitian serupa yang telah dilakukan oleh X. Wu^[10,11] menggunakan model trafik masukan *batch traffic* dengan proses kedatangan *Poisson* dan panjang *batch* terdistribusi geometris. Hasil simulasi menunjukkan peluang sel hilang di jaringan makin berkurang dengan ukuran *token pool* K yang makin kecil; berkurangnya laju pembangkitan *token* mengakibatkan bertambahnya *throughput* jaringan; panjang *burst* rata-rata trafik masukan yang semakin besar akan memperbesar hilangnya sel di jaringan; dan kemudian, kinerja *leaky bucket* makin meningkat dengan digabungkannya mekanisme pengontrol kongesti lainnya seperti pengontrolan adaptif, *marking*, dan prioritas. Untuk *leaky bucket* hasil *dimensioning*, simulasi antara lain menunjukkan bahwa jika laju kedatangan sel *conforming* (sesuai dengan kontrak trafik), maka peluang sel hilang di jaringan makin kecil jika *token pool* makin kecil. Untuk $(M+K)$ yang sama, *leaky bucket* dengan *buffer* menghasilkan peluang sel hilang di jaringan lebih kecil dibandingkan dengan tanpa *buffer*. *Leaky bucket* dengan *buffer* M besar dan pembangkitan *token* N besar menghasilkan peluang sel hilang di jaringan terkecil. Jika laju kedatangan sel makin jauh dari *conforming*, *leaky bucket* dengan N besar menghasilkan *throughput* jaringan yang lebih besar.

Kata Kunci: analisis, model, simulasi, proses kedatangan, *burstiness*, *leaky bucket*, kinerja, jaringan ATM, multiplexer, adaptif, prioritas, *marking*, *pendimensioning*.

Abstract

Performance analysis of leaky buckets integrated in ATM networks.

This paper is an analysis of the performance of leaky buckets, which are integrated within ATM networks combined with adaptive control, marking and prioritizing. Also discussed is the performance of leaky bucket dimensioning. This performance is evaluated through network cell loss probabilities and network throughput. The arrival process is a 2-phase MMBP. Similar research has been done by Wu^[10,11] using incoming traffic model batch traffic with a Poisson arrival process and a geometrically distributed batch length. The result of the simulation shows that the network cell loss probability is reduced as the token pool size K becomes smaller and smaller; the decrease of the token generation rate causes an increase in the network throughput; the network cell loss probability is higher in the case of a larger mean burst length; and the performance of the leaky bucket can be increased by incorporating other control mechanisms such as adaptive control, marking and prioritizing. For the leaky bucket dimensioning, the results of the simulation show, among other things, that if the incoming cell rate is conforming (appropriate to traffic contract) and the token pool is reduced, then the probability of network cell loss is also reduced. If $(M + K)$ are the same, the buffered leaky bucket results in a network cell loss probability is smaller than for an unbuffered leaky bucket. The leaky bucket with big buffer M and big token generation N result in the smallest network cell loss probability. If the incoming cell rate is far from conforming, the leaky bucket with big N results in a bigger network throughput.

Key words: Analysis, model, simulation, arrival process, *burstiness*, *leaky bucket*, performance, ATM network, multiplexer, adaptive, prioritizing, marking, *dimensioning*.

1 Pendahuluan

Mekanisme *policing* (*Usage Parameter Control* (UPC) dalam terminologi ITU) merupakan teknik kontrol kongesti secara preventif yang mengontrol aliran trafik dari sumber pada *user network interface* pada jaringan ATM. Mekanisme *policing* yang populer dan banyak mendapat perhatian para ahli adalah *leaky bucket*.

Mekanisme *policing* menjamin sumber trafik memenuhi kontrak trafik yang telah disepakati antara pemakai dengan jaringan pada saat pembukaan hubungan. Parameter kontrak trafik antara lain *mean rate*, *peak rate* dan *burstiness* dari sumber trafik. Sepanjang sumber memenuhi batas kontrak trafik, mekanisme *policing* tetap transparan bagi sumber.

Ada dua pendekatan untuk menganalisis kinerja *leaky bucket*. Pendekatan yang pertama adalah mengkarakterisasi proses keluaran mekanisme *policing leaky bucket* tunggal. Pendekatan kedua adalah *leaky bucket* terintegrasi pada jaringan ATM. Dalam pendekatan kedua ini dapat digunakan *traffic descriptor*, antara lain laju rata-rata kedatangan sel, dan panjang *burst* rata-rata yang juga merupakan ukuran *burstiness*. Pendekatan kedua ini dilakukan dengan beberapa model *leaky bucket* yang terhubung dengan sebuah multiplexser. Kemudian, *leaky bucket* dikombinasi dengan beberapa metode kontrol kongesti lainnya, yaitu kontrol adaptif, metode *marking* dan metode prioritas. Pendekatan kedua ini diilhami oleh penelitian yang telah dilakukan oleh Wu^[10,11]. Dalam penelitiannya yang berupa simulasi digunakan model trafik masukan berupa *batch traffic* dengan proses kedatangan *Poisson* dan panjang *batch* terdistribusi geometris. Dalam makalah ini, sumber trafiknya diasumsikan *2-phase MMBP* yang kemudian diturunkan menjadi IBP.

Makalah ini juga meninjau kinerja *leaky bucket* hasil *dimensioning* pada jaringan ATM. Maksud dari *dimensioning* adalah agar *leaky bucket* transparan terhadap sumber *conforming* (sumber yang sesuai dengan kontrak trafik saat pembukaan hubungan). Sebagai ukuran kinerja digunakan parameter peluang sel hilang di jaringan dan *throughput* jaringan. Karena kompleksnya perhitungan secara matematik, analisis hanya berdasarkan data yang diperoleh dari simulasi.

2 Model simulasi dan asumsi

2.1 Model proses kedatangan

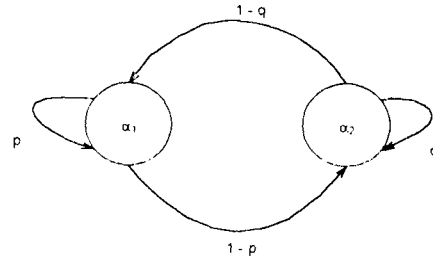
Proses kedatangan sel yang masuk ke *leaky bucket* berupa model *Markov Modulated Bernoulli Process* (MMBP). Model ini lebih sesuai untuk analisis jaringan kecepatan tinggi^[3,4]. Skala waktu dalam MMBP dibagi dalam slot-slot dengan panjang yang tetap. Peluang bahwa sebuah slot berisi sel sesuai dengan proses *Bernoulli* dengan parameter yang berubah sesuai dengan fase (kondisi) dari *r* proses *Markov* (*r-state Markov process*), yang bebas terhadap proses kedatangan.

Pada awal tiap slot, proses *Markov* berpindah dari fase (*state*) *i* ke fase *j* dengan peluang p_{ij} atau tetap berada di fase *i* dengan peluang p_{ii} . Maka :

$$\sum_{j=1}^r p_{ij} = 1 \quad i = 1, \dots, r \quad (1)$$

Pada fase *i*, slot berisi sel dengan peluang α_i dan tak ada sel dengan peluang $(1 - \alpha_i)$. Peluang kedatangan sel dan

proses *Markov* yang mendasarinya diasumsikan bebas satu sama lainnya^[4,8].



Gambar 1 Proses 2-phase MMBP

Model *2-phase MMBP* ditunjukkan pada Gambar 1. Pada awal slot, proses kedatangan pindah dari fase 1 ke fase 2 dengan peluang $(1 - p)$, dan pindah dari fase 2 ke fase 1 dengan peluang $(1 - q)$. Selama fase 1, kedatangan terjadi dengan peluang α_1 , dan selama fase 2, kedatangan terjadi dengan peluang α_2 . Apabila $\alpha_1 = \alpha_2$ maka proses MMBP hanya mempunyai satu fase sehingga proses MMBP menjadi proses *Bernoulli*. Dan, apabila α_1 atau α_2 sama dengan nol, maka proses MMBP menjadi proses *Interrupted Bernoulli Process* (IBP).

2.2 Burstiness

Dalam model IBP, *burstiness* dinyatakan dalam parameter panjang periode *burst* rata-rata^[3]. Kemudian, dalam IBP, jika $\alpha_1 = 1$, fasenya dinamakan fase aktif dan slot selalu berisi sel. Didefinisikan juga peubah acak *B* yang menyatakan panjang periode *burst* dalam satuan slot. Distribusi waktu yang dihabiskan dalam fase aktif atau *burst* dan durasi periode aktif rata-rata atau panjang periode *burst* rata-rata berturut-turut dinyatakan sebagai^[3]:

$$P(B = i) = (1 - p)p^{i-1} \quad i \geq 1 \quad (2)$$

$$E[B] = 1/(1-p) \quad (3)$$

Jika $\alpha_2 = 0$, fasenya dinamakan fase diam. Didefinisikan pula peubah acak *S* yang menyatakan panjang periode diam dalam satuan slot. Demikian juga distribusi waktu yang dihabiskan dalam fase diam dan durasi periode diam rata-rata berturut-turut dinyatakan sebagai^[3]:

$$P(S = i) = (1 - q)q^{i-1} \quad i \geq 1 \quad (4)$$

$$E[S] = 1/(1-q) \quad (5)$$

Peluang *steady state* IBP berada dalam fase aktif dan diam berturut-turut dinyatakan dengan^[8]:

A
I
k
S
d
s
P
d
r
se
af
m
di
sl
se
me
dig
sae
ter
ber

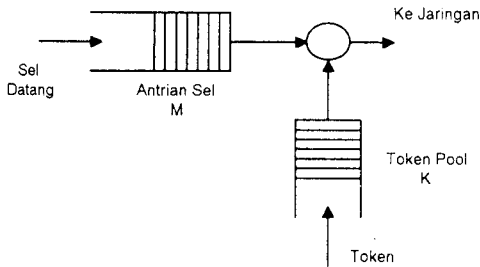
2.4
Mo
Gar
buc.
para
dapa
prio
(mui
mela

$$\pi_B = (1-q) / (2-p-q) \quad ; \quad \pi_S = (1-p) / (2-p-q) \quad (6)$$

Peluang sebuah slot berisi sel :

$$\rho = \pi_B \alpha_1 = \frac{\alpha_1(1-q)}{(2-p-q)} \quad (7)$$

2.3 Model Leaky Bucket^[4]



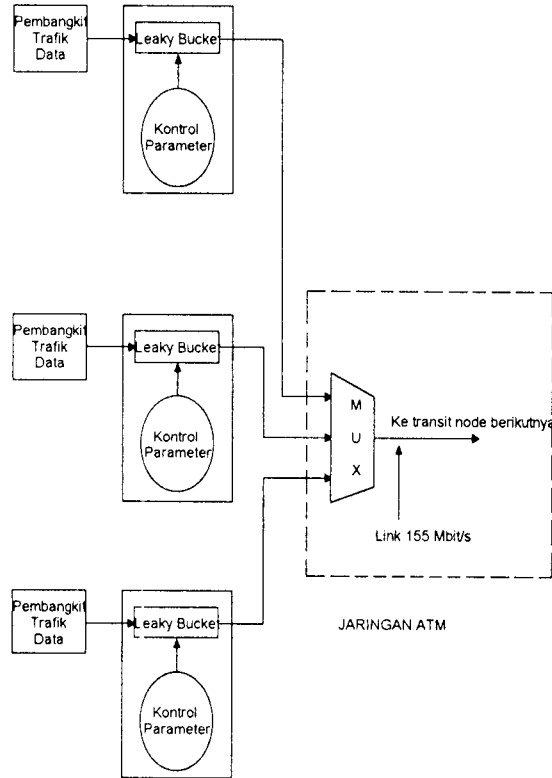
Gambar 2 Model Leaky Bucket

Dalam model *leaky bucket* terdapat dua macam antrian yaitu sebuah antrian sel (*buffer*) berukuran M dan *token pool* berukuran K. Yang menjadi dasar pendekatan metode *leaky bucket* adalah bahwa sel sebelum masuk ke jaringan harus mendapatkan *token* terlebih dahulu. Setiap kedatangan sel akan membutuhkan satu *token* dan akan segera meninggalkan *leaky bucket* apabila setidaknya ada satu *token* yang berada dalam *token pool*. *Token* dibangkitkan pada laju konstan dan ditempatkan dalam *token pool*. Jumlah *token* yang dapat menunggu di dalam *token pool* terbatas, sehingga pada saat *token pool* penuh, *token* yang datang berikutnya akan dibuang. Sementara sel-sel datang dan menggunakan *token*, *token-token* baru akan dibangkitkan pada saat periode pembangkitannya tiap N slot dan ditempatkan di dalam *pool*. Apabila masih ada sel di dalam *buffer*, tidak akan ada *token* yang menunggu dalam *token pool*. Ukuran *buffer* yang digunakan adalah terbatas, dan sel yang datang pada saat *buffer* telah penuh akan dibuang. Jadi, tak pernah terjadi *token* dan sel berada di *leaky bucket* pada waktu bersamaan.

2.4 Model sistem jaringan ATM

Model jaringan ATM yang digunakan ditunjukkan pada Gambar 3^[10,11]. Model tersebut memperlihatkan *leaky bucket* yang dapat dimodifikasi dengan pengontrolan parameter masukan. Pengontrolan parameter tersebut dapat berupa pengontrolan adaptif, *marking* atau prioritas. Keluaran *leaky bucket* diumpankan ke MUX (multiplexer) yang dihubungkan ke jaringan ATM melalui *link* kecepatan tinggi. Sumber trafik masukan

semua *leaky bucket* diasumsikan hanya berupa data. Sumber trafik data tersebut diasumsikan dapat dimodelkan dengan model *Interrupted Bernoulli Process* (IBP).



Gambar 3 Model simulasi *leaky bucket* terintegrasi pada jaringan ATM

Kapasitas *link* diasumsikan 155 Mbit/s, yang ekuivalen dengan $155 \times 10^6 / (53 \times 8) \approx 3.65 \times 10^5$ sel / detik dan satu sel *payload* ATM sebanyak 48 oktet^[2]. Diasumsikan dalam satu slot waktu cukup untuk mengirimkan satu sel ATM ke *link* 155 Mbit/s pada puncak laju transmisi. Maka, lebar satu slot sama dengan $53 \times 8 / (155 \times 10^6) \approx 2.73 \times 10^{-6}$ detik.

Didefinisikan throughput jaringan sebagai perbandingan dari banyaknya sel yang meninggalkan multiplexer terhadap banyaknya sel yang dibangkitkan oleh seluruh sumber selama n slot waktu. Sementara itu, peluang sel hilang di jaringan diasumsikan sebagai perbandingan banyaknya sel yang hilang di multiplexer terhadap banyaknya sel yang masuk ke multiplexer.

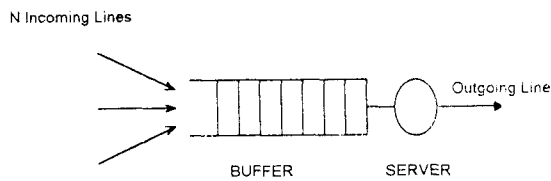
2.5 Model multiplexer

ATM multiplexer menerima sel dengan panjang tetap dari sejumlah *incoming line* yang berbeda dan kemudian mengirimkannya keluar pada *outgoing line* tunggal.

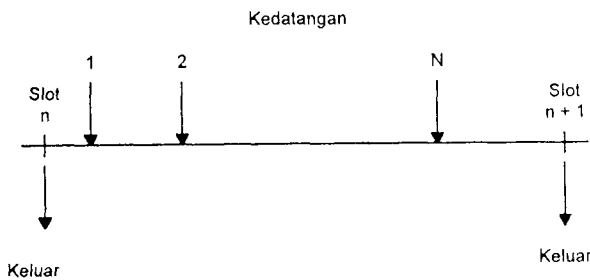
Dalam multiplekser terdapat *buffer* dengan kapasitas terbatas. Sel yang datang akan hilang di jaringan ketika *buffer* multiplekser sudah penuh.

ATM multiplekser umumnya dimodelkan berupa *buffer* dengan *server* tunggal dan dengan layanan deterministik dengan N *stream* kedatangan (lihat Gambar 4) [5]. Dalam simulasi ini *stream* kedatangan berasal dari keluaran *leaky bucket*. Waktu pelayanan adalah deterministik yang lamanya satu slot waktu ($\approx 2.73 \times 10^{-6}$ detik) dari *outgoing line*; diasumsikan satu slot tersebut cukup untuk mentransmisikan satu sel keluar dari multiplekser. Setiap proses kedatangan juga dibagi-bagi dalam slot, dan satu slot kedatangan lebarnya sama dengan slot dari *outgoing line* (*service slot*). Disiplin pelayanan sel adalah FIFO (*First In First Out*).

Batas-batas slot dari *stream* kedatangan berada di antara batas *service slot*. Jika sel datang ketika *buffer* multiplekser kosong, sel akan menunggu sampai awal dari *service slot* berikutnya, dan kemudian sel tersebut mulai dilayani. Sel meninggalkan sistem multiplekser pada akhir *service slot* tersebut. Untuk jelasnya, lihat Gambar 5 [5].



Gambar 4 Model antrian multiplekser



Gambar 5 Hubungan sel datang dan sel keluar pada multiplekser

3 Leaky bucket dasar terintegrasi pada jaringan ATM

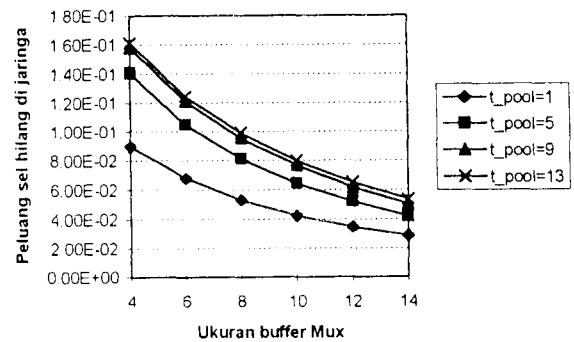
Untuk simulasi *leaky bucket* dasar, sistem *leaky bucket* belum dimodifikasi dengan mekanisme pengontrol lainnya. Untuk mengevaluasi kinerja *leaky bucket*

tersebut, pada jaringan dievaluasi hubungan antara periode pembangkitan *token* dengan *throughput* jaringan, hubungan ukuran maksimum *token pool* terhadap peluang sel hilang di jaringan, dan hubungan panjang *burst* rata-rata terhadap *throughput* jaringan.

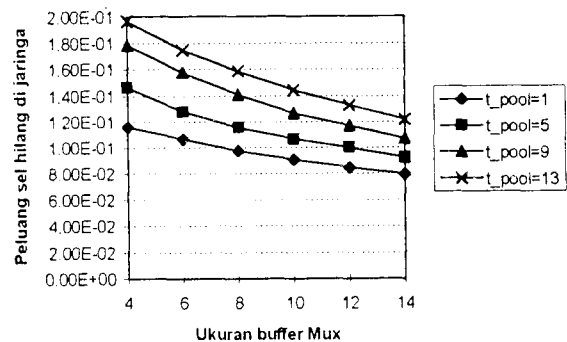
Untuk simulasi *leaky bucket*, dasar ukuran *buffer* tiap *leaky bucket* dibuat sebesar mungkin, karena kita ingin meminimalkan pengaruh sel hilang pada *leaky bucket*.

3.1 Hasil simulasi dan analisis

Gambar 6 dan 7 menunjukkan hasil simulasi ukuran maksimum *token pool* terhadap peluang sel hilang di jaringan (di multiplekser). Pada simulasi tersebut digunakan periode pembangkitan *token* N sebesar 2 slot, laju kedatangan sel rata-rata $\rho = 0.2$, jumlah sel yang diumpukan ke multiplekser sebanyak 5 000 000 sel, dan untuk Gambar 6 digunakan panjang *burst* rata-rata $E[B] = 5$ dan untuk Gambar 7 digunakan $E[B] = 20$.



Gambar 6 Pengaruh *token pool* terhadap peluang sel hilang di jaringan, $E[B]=5$



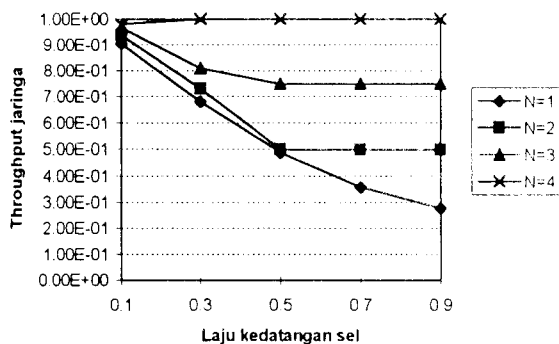
Gambar 7 Pengaruh *token pool* terhadap peluang sel hilang di jaringan, $E[B]=20$

Sel akan hilang di jaringan apabila sel datang pada saat multiplekser penuh. Dari simulasi terlihat peluang sel hilang di jaringan makin berkurang dengan ukuran maksimum *token pool* yang makin kecil. Ukuran *buffer*

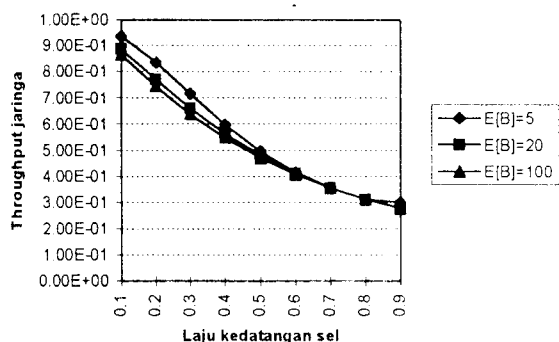
multiplekser yang semakin besar juga akan semakin mengurangi peluang sel hilang di jaringan. Panjang *burst* rata-rata yang semakin besar juga akan memperbesar hilangnya sel di jaringan.

Hasil simulasi hubungan periode kedatangan token terhadap *throughput* jaringan diperlihatkan pada Gambar 8. Pada simulasi ini *throughput* jaringan merupakan perbandingan dari banyaknya sel yang meninggalkan multiplekser terhadap banyaknya sel yang dibangkitkan seluruh sumber yang sampai pada multiplekser selama 2 500 000 slot waktu. Untuk simulasi digunakan ukuran multiplekser sebesar 4 sel, $E[B]=10$, ukuran *token pool* tiap *leaky bucket* sebesar 5 *token*.

Hasil simulasi menunjukkan dengan bertambahnya periode kedatangan token, *throughput* jaringan semakin bertambah. Ini berarti makin banyak sel yang sampai di jaringan tanpa mengalami kongesti. Bagaimanapun, dengan bertambahnya periode kedatangan token, makin bertambah pula sel yang hilang di *buffer leaky bucket* apabila ukuran *buffer* tersebut tidak cukup panjang.



Gambar 8 Hubungan periode kedatangan token terhadap *throughput* jaringan



Gambar 9 Hubungan panjang *burst* rata-rata terhadap *throughput* jaringan

Hasil simulasi hubungan panjang *burst* rata-rata $E[B]$ terhadap *throughput* jaringan ditunjukkan pada Gambar 9. Ukuran-ukuran yang digunakan adalah laju kedatangan token $N=1$, panjang multiplekser = 4 sel, *token pool* = 5 *token*. Panjang *burst* rata-rata yang besar menunjukkan tingkat *burstiness* yang lebih tinggi yang mengakibatkan bertambahnya peluang terjadinya kongesti di jaringan. Hasilnya berupa *throughput* jaringan yang lebih rendah sebagaimana ditunjukkan pada Gambar 9.

4 Kontrol *leaky bucket* secara adaptif

Kontrol kongesti secara preventif semata tidak bisa menghilangkan sama sekali kongesti yang terjadi pada jaringan ATM. Ketika kongesti terjadi, diperlukan reaksi untuk mengatasi masalah itu; karena itulah diperlukan juga kontrol kongesti reaktif secara adaptif.

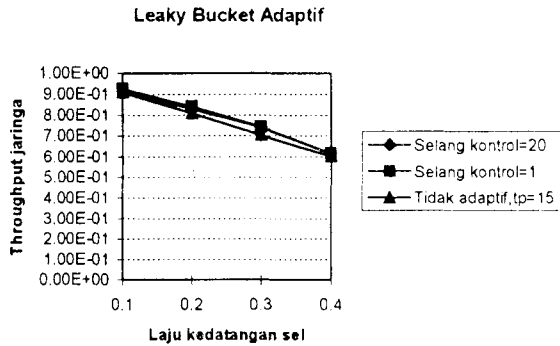
Pada kontrol adaptif ukuran-ukuran *token pool* atau laju pembangkitan token, berubah tergantung pada kondisi beban trafik jaringan^[10,11]. Jika beban trafik sangat rendah, ukuran *token pool* atau laju pembangkitan token dapat ditambah. Jika beban trafik tinggi, ukuran parameter tersebut dapat dikurangi. Beban trafik jaringan dapat ditentukan dengan memantau jumlah sel yang ada di *buffer* jaringan (multiplekser). Jaringan mengirimkan pesan status trafik jaringan ke pemakai sehingga pemakai dapat mengatur parameter-parameter *leaky bucket* tersebut.

Dalam simulasi yang dilakukan, pengontrolan adaptif dilakukan dengan pengaturan ukuran maksimum *token pool* untuk harga laju pembangkitan token tertentu. Untuk memantau kondisi *buffer* multiplekser, digunakan dua harga *threshold*; di sini dipilih 40% dan 80% dari ukuran maksimum. Dalam proses adaptasi, ukuran *token pool* dikurangi jika isi *buffer* multiplekser lebih dari 80% maksimum. Jika *buffer* isinya kurang dari 40% maksimum, ukuran *token pool* ditambah. Ukuran *token pool* dijaga konstan jika isi *buffer* antara 40% sampai 80%.

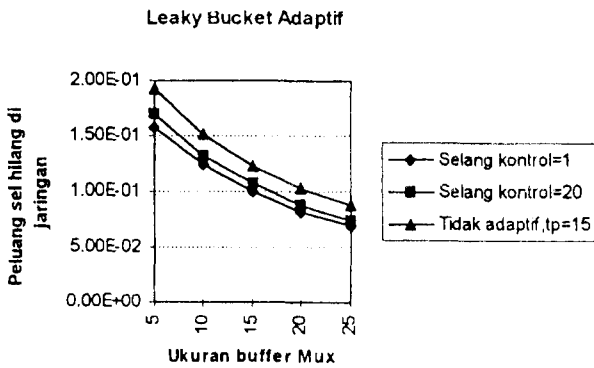
4.1 Hasil simulasi dan analisis

Untuk simulasi *throughput* jaringan periode pembangkitan token $N = 2$ slot, ukuran *buffer* multiplekser sebesar 5 sel, $E[B]=20$, waktu simulasi selama 2 500 000 slot waktu. Ukuran *buffer* tiap *leaky bucket* dibuat sebesar mungkin. Untuk penyederhanaan semua ukuran awal *token pool* untuk *leaky bucket* adaptif dibuat sama sebesar 15 *token*, sedangkan untuk yang tidak adaptif *token pool* juga berukuran 15 *token*. Ukuran *token pool* berkurang 2 bila isi *buffer* mux lebih dari 80% ukuran maksimumnya, dan bertambah 2 bila isi *buffer* mux kurang dari 40%. Ukuran *token pool*

dijaga tetap bila isi *buffer* mux antara 40% dan 80%. Ukuran minimum *token pool* sebesar 1 *token*.



Gambar 10 Hubungan perubahan *token pool* terhadap *throughput* jaringan



Gambar 11 Hubungan perubahan *token pool* terhadap peluang sel hilang di jaringan

Gambar 10 menunjukkan hasil simulasi *throughput* jaringan dengan dua harga selang waktu kontrol umpan balik, yaitu 1 slot waktu dan 20 slot waktu. Selang waktu kontrol umpan balik ini merepresentasikan selang waktu pengiriman pesan status trafik jaringan ke pemakai; dalam hal ini *leaky bucket*. Selang waktu kontrol yang lebih kecil akan mempunyai *throughput* yang lebih besar dibandingkan dengan selang waktu kontrol yang lebih besar. Selang waktu kontrol yang lebih kecil akan membuat *leaky bucket* cepat tanggap terhadap kondisi trafik di jaringan, sehingga *leaky bucket* tidak terlalu lama membanjiri jaringan apabila jaringan sedang dalam keadaan kongesti. Gambar 10 juga memperlihatkan *leaky bucket* dasar dengan ukuran maksimum *token pool* tetap sebesar 15 *token*

menghasilkan *throughput* yang lebih kecil dibandingkan dengan yang adaptif.

Kinerja *leaky bucket* kontrol adaptif dalam peluang sel hilang di jaringan menunjukkan peningkatan, yaitu lebih kecil dibandingkan dengan *leaky bucket* dasar seperti yang ditunjukkan dalam Gambar 11.

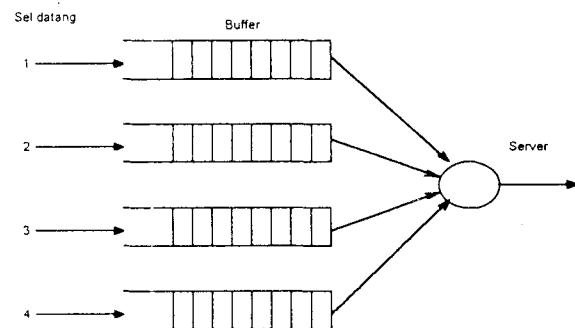
Untuk simulasi peluang sel hilang, jumlah sel yang masuk ke multiplekser sebanyak 5 juta sel, $N=2$ slot, laju kedatangan sel rata-rata $\rho=0.2$, $E[B]=20$, ukuran *buffer* tiap *leaky bucket* sebesar mungkin. Juga, semua ukuran awal *token pool* untuk *leaky bucket* adaptif dibuat sama, yaitu 15 *token*, sedangkan untuk yang tidak adaptif, *token pool* juga berukuran 15 *token*.

5 Kombinasi *leaky bucket* dengan metode prioritas

Sel yang datang akan dibuang pada *node* jaringan yang sedang mengalami kongesti. Dengan metode prioritas, sel dengan prioritas rendah akan dibuang lebih dahulu daripada sel dengan prioritas yang lebih tinggi. Ini berguna pada jaringan ATM yang menyediakan layanan dengan berbagai tingkat (*multiple grades of services*) untuk kelas layanan (*service classes*) yang berbeda.

Untuk simulasi diasumsikan data dari *leaky bucket* 1 mempunyai prioritas ke-4, data dari *leaky bucket* 2 dengan prioritas 2, data dari *leaky bucket* 3 dengan prioritas 3, dan data dari *leaky bucket* 4 dengan prioritas 1. Model simulasi multiplekser dengan metode prioritas dapat dilihat pada Gambar 12. Data dengan prioritas lebih tinggi akan dilayani lebih dahulu oleh *server* daripada data dengan prioritas yang lebih rendah.

Model prioritas yang digunakan sebagai contoh simulasi di sini adalah model prioritas *non-preemptive*, yaitu sel dengan prioritas yang lebih tinggi tak dapat menginterupsi sel yang prioritasnya lebih rendah yang sedang dilayani *server*¹¹. Disiplin pelayanan untuk sel-sel yang datang dengan prioritas yang sama sesuai dengan prinsip FIFO.



Gambar 12 Model antrian multiplekser dengan metoda prioritas

5
C
la
d
p
se
se
E
di
Ga
ak
ke
thr

Peluang sel hilang di jaringan

Gambar

6 K
m

Dalam
hubung
masuk
melebih
pembuk
kondisi
simulasi
prioritas
dianggap

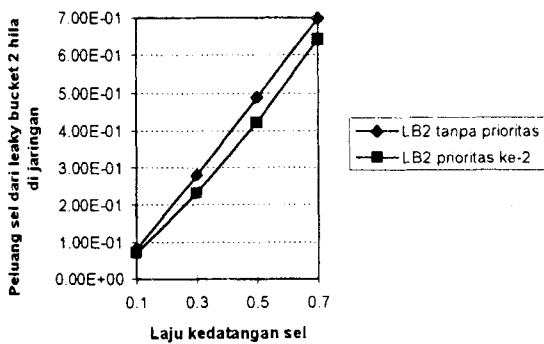
token
datang

Misaik

5.1 Hasil simulasi dan analisis

Gambar 13 menunjukkan perbandingan peluang sel dari *leaky bucket 2* dengan prioritas ke-2 hilang di jaringan dibandingkan dengan bila tanpa prioritas. Digunakan periode pembangkitan *token* $N=1$, ukuran multiplexer sebesar 4 sel, ukuran *buffer* tiap *leaky bucket* berukuran sebesar-besarnya, ukuran *token pool* sebesar 5 *token*, $E[B]=20$, jumlah sel dari *leaky bucket 2* yang diumpankan ke multiplexer sebanyak 5 juta sel.

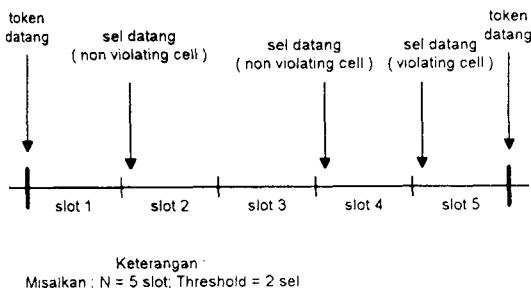
Gambar 13 juga menunjukkan bahwa metode prioritas akan mengurangi peluang sel hilang di jaringan. Makin kecilnya peluang sel hilang berarti makin bertambahnya *throughput* jaringan.



Gambar 13 Peluang sel dari *leaky bucket 2* dengan prioritas ke-2 hilang di jaringan

6 Kombinasi *leaky bucket* dengan metode *marking*

Dalam jaringan ATM, selama berlangsungnya hubungan mungkin bisa terjadi perubahan trafik masukan ke jaringan sehingga trafik yang terjadi melebihi *bandwidth* yang dialokasikan pada waktu pembukaan hubungan. Sel-sel yang mengakibatkan kondisi tersebut dinamakan *violating cell*. Dalam simulasi *leaky bucket* dasar, dengan kontrol adaptif dan prioritas, semua sel yang masuk ke *leaky bucket* dianggap sel yang tidak *violating*.



Gambar 14 *Violating cell*

Dalam simulasi ini diuji dua macam pendekatan^[10,11] untuk mengatasi *violating cell*. Yang pertama adalah dengan menandai (*marking*) sel datang yang *violating* dan memperbolehkannya masuk ke *buffer leaky bucket*. Sel bertanda tersebut akan disalurkan ke jaringan apabila mendapat *token*. Apabila *node* jaringan mengalami kongesti, *violating cell* (sel bertanda) akan dibuang dari *node* jaringan bila sel yang normal (sel tak bertanda) datang. Pendekatan yang lain adalah *violating cell* dibuang sebelum memasuki *buffer* di *leaky bucket*.

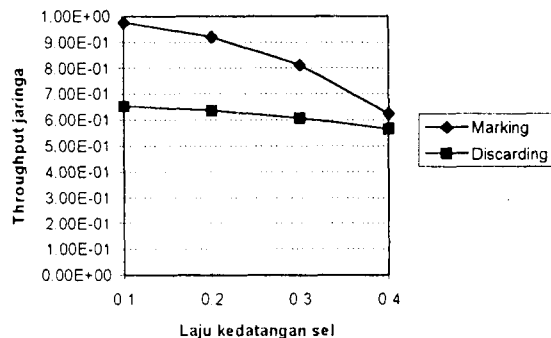
Dalam simulasi seperti yang ditunjukkan pada Gambar 14, *violating cell* terjadi apabila sel datang ke *leaky bucket* pada hitungan lebih tinggi dari harga *threshold* yang ditentukan tiap periode N slot. Pencacah untuk mengidentifikasi *violating cell* selalu di-*reset* di akhir slot ke- N .

Untuk simulasi dengan pendekatan *marking*, apabila *buffer* multiplexer penuh dan di dalamnya terdapat sel bertanda, maka sel bertanda tersebut akan dibuang bila sel tak bertanda datang ke multiplexer. Sel tak bertanda akan dibuang pula di multiplexer bila *buffer* multiplexer penuh dan tak ada sel bertanda di *buffer* multiplexer.

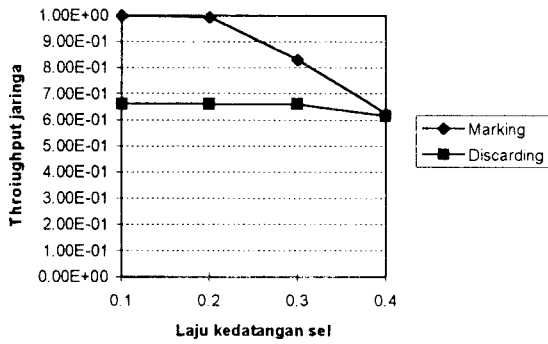
6.1 Hasil simulasi dan analisis

Gambar 15 dan 16 menunjukkan hasil simulasi *throughput* jaringan dari dua metode, yaitu penandaan (*marking*) dan pembuangan (*discarding*) *violating cell*. Untuk simulasi ini digunakan periode pembangkitan *token* $N= 3$ slot, *threshold* sebesar 2 sel, $E[B]=20$, ukuran *buffer* dan *token pool* berturut-turut masing-masing sebesar 9999 sel dan 5 *token*, ukuran *buffer* mux 4 sel untuk Gambar 15 dan 70 sel untuk Gambar 16, dan waktu simulasi selama 1 000 000 slot waktu.

Dari Gambar 15 dan 16 terlihat bahwa *throughput* jaringan meningkat dengan diterapkannya metode *marking* pada *leaky bucket*. Dari gambar itu terlihat pula *throughput* meningkat dengan bertambahnya ukuran *buffer* multiplexer.



Gambar 15 *Leaky bucket* dengan metode *Marking*, mux=4 sel



Gambar 16 Leaky bucket dengan metode Marking, $\mu = 70$ sel

7 Pendimensian leaky bucket

Agar operasi kerja leaky bucket transparan terhadap sumber conforming, secara umum telah disetujui bahwa peluang sel hilang pada leaky bucket sebaiknya kurang dari 10^{-9} (atau 10^{-10})^[3,4]. Peluang sel hilang tergantung pada nilai $M + K$, bukan pada nilai individu M atau K . Pendimensian leaky bucket dilakukan dengan menentukan parameter $M + K$ dan N (periode pembangkitan token). Setelah $M+K$ ditentukan, nilai individu M dan K dapat ditentukan berdasarkan toleransi delay dari aplikasi masukan leaky bucket.

Dalam percobaan yang telah dilakukan, digunakan proses kedatangan (sumber) conforming dengan kasus model proses kedatangan Bernoulli dengan $p+q=1$ dan $\alpha_1 = \alpha_2 = 0,1$ sehingga laju kedatangan sel rata-rata $\rho = 0,1$ dan C^2 waktu antar-kedatangan sel sebesar $0,9$ ^[3,4]. Beberapa hasil percobaan pendimensian melalui simulasi ditunjukkan pada Tabel 1. Metode analisis keluaran data hasil simulasi yang digunakan adalah metode replikasi bebas (independent replication) dengan 10 replikasi, selang keyakinan 90%, dan pada tiap replikasi dibangkitkan sebanyak 10^6 sel akibat keterbatasan program simulasi. Ini berarti bahwa untuk simulasi, agar leaky bucket transparan, diperlukan peluang sel hilang kurang dari 10^{-5} .

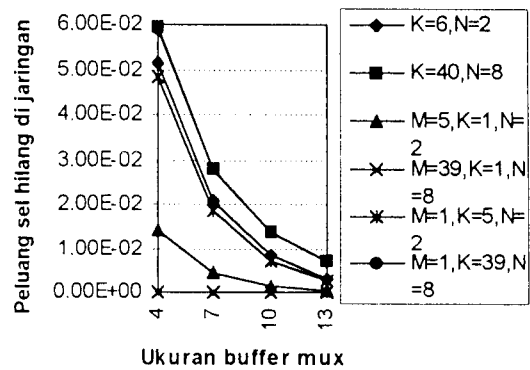
Tabel 1 Pendimensian leaky bucket melalui simulasi dengan sumber model Bernoulli

M+K	N	Peluang sel hilang
6	2	0 ± 0
6	4	$6.16162e-6 \pm 9.3395e-7$
3	2	$5.252525e-6 \pm 1.37451e-6$
40	8	0 ± 0
100	2	0 ± 0
1000	8	0 ± 0

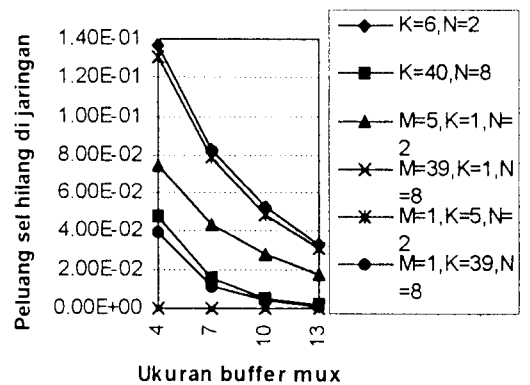
8 Hasil simulasi leaky bucket dimensioning pada jaringan ATM

Jika laju kedatangan sel rata-rata conforming ($\rho=0.1$), maka peluang sel hilang di jaringan makin kecil jika ukuran token pool makin kecil. Untuk $(M+K)$ yang sama, leaky bucket dengan buffer menghasilkan peluang sel hilang di jaringan yang lebih kecil dibandingkan dengan tanpa buffer. Leaky bucket dengan buffer besar dan N besar menghasilkan peluang sel hilang di jaringan yang kecil: ini terlihat pada ukuran $M=39$; $K=1$; $N=8$ yang menghasilkan loss jaringan terkecil.

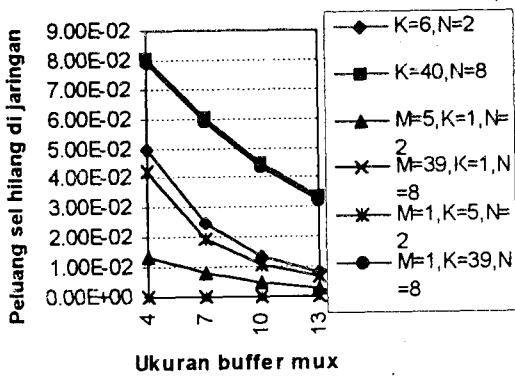
Jika laju kedatangan sel rata-rata makin jauh dari conforming, leaky bucket dengan N besar menghasilkan throughput jaringan yang lebih besar. Throughput diasumsikan sebagai perbandingan banyaknya sel yang keluar dari multiplexer terhadap banyaknya sel yang dibangkitkan seluruh sumber yang sampai ke multiplexer selama n slot waktu. Leaky bucket dengan peluang sel hilang di jaringan terbesar menghasilkan throughput jaringan terkecil.



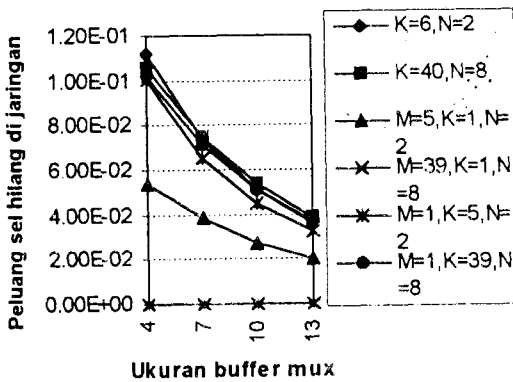
Gambar 17 Simulasi peluang sel hilang di jaringan, $E[B]=5$, ρ masukan = 0.1



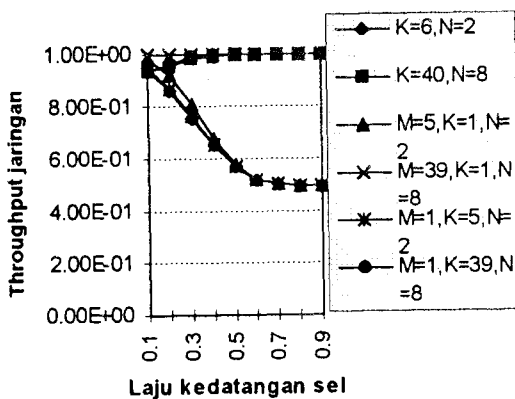
Gambar 18 Simulasi peluang sel hilang di jaringan, $E[B]=5$, ρ masukan = 0.2



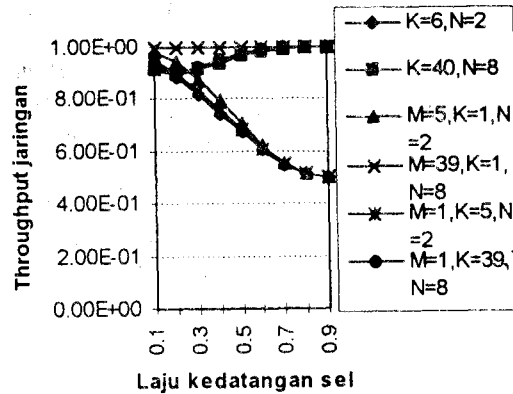
Gambar 19 Simulasi peluang sel hilang di jaringan, $E[B]=20$, ρ masukan = 0.1



Gambar 20 Simulasi peluang sel hilang di jaringan, $E[B]=20$, ρ masukan = 0.2



Gambar 21 Simulasi throughput jaringan, $E[B]=5$, ukuran mux = 4 sel



Gambar 22 Simulasi throughput jaringan, $E[B]=20$, ukuran mux = 4 sel

9 Kesimpulan

Peluang sel hilang di jaringan makin berkurang dengan ukuran *token pool* yang makin kecil. Dengan berkurangnya laju pembangkitan *token*, *throughput* jaringan semakin bertambah. Panjang *burst* rata-rata trafik masukan yang semakin besar akan memperbesar hilangnya sel di jaringan dan juga akan mengurangi *throughput* jaringan. Kinerja *leaky bucket* di jaringan dapat ditingkatkan dengan menggabungkannya dengan mekanisme pengontrol kongesti lainnya seperti pengontrolan adaptif, *marking*, dan prioritas. Tetapi, hal ini akan menambah beban pemrosesan dalam sistem *leaky bucket* dan jaringan.

Jika laju kedatangan sel *conforming*, maka peluang sel hilang di jaringan makin kecil jika *token pool* makin kecil. Untuk $(M+K)$ yang sama, *leaky bucket* dengan *buffer* menghasilkan peluang sel hilang di jaringan lebih kecil dibandingkan dengan tanpa *buffer*. *Leaky bucket* dengan *buffer* besar dan periode pembangkitan *token* N besar menghasilkan peluang sel hilang di jaringan terkecil. Jika laju kedatangan sel makin jauh dari *conforming*, *leaky bucket* dengan N besar menghasilkan *throughput* jaringan yang lebih besar. *Leaky bucket* dengan peluang sel hilang yang besar di jaringan menghasilkan *throughput* jaringan yang kecil.

10 Daftar pustaka

1. Akimaru, H. and K. Kawashima, *Telettraffic: Theory and Applications*, Springer-Verlag, New York, 1993.
2. De Prycker, M., *Asynchronous transfer mode: solution for broadband ISDN*, Prentice Hall, 1995.
3. Holtsinger, D., & H. Perros, *Waiting Time and Cell Loss Probability Analysis for the Buffered Leaky*

- Bucket*, Technical Report, Department of Computer Science, North Carolina State University. North Carolina, September 1992.
4. Holsinger, D.S., and H.G. Perros, *Performance Analysis of Leaky Bucket Policing Mechanisms*, Technical report, Department of Computer Science, North Carolina State University, November 1991.
 5. Hong, Suk.W and H.G. Perros, *An Approximate Analysis of an ATM Multiplexer with Multiple Heterogenous Bursty Arrivals*, Technical report, Department of Computer Science, North Carolina State University, August 1991.
 6. Kleinrock, Leonard, *Queueing System, Vol. 1: Theory*, Wiley, New York, 1976.
 7. M.Law. Averill & W. David Kelton, *Simulation Modeling & Analysis*, 2nd edition, Mc Graw-Hill International Editions, 1991, Singapore.
 8. Onvural O., Raif, *Asynchronous Transfer Mode. Performance Issues*. Artech House, Inc. 1994.
 9. S. Lavenberg, Stephen, *Computer Performance Modeling Handbook*, Academic Press, New York, 1983.
 10. Wu. X., I. Lambadaris, H. Lee and A.R. Kaye, *A Comparative Study of Some Leaky Bucket Network Access Schemes*, ICC '94, pp. 1586-1591.
 11. Wu, X., *Leaky Bucket Congestion Control in ATM Networks*, M.Eng. Thesis, Dept. of Systems and Computer Engineering, Carleton University, 1993.