Online Traffic Engineering and Connection Admission Control Based on Path Queue States

Sven Krasser, Henry L. Owen
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250, USA
{sven, owen}@ece.gatech.edu

Jochen Grimminger, Hans-Peter Huth, Joachim Sokol
Siemens AG, CT IC2 Corporate Technology
81730 Munich, Germany
{jochen.grimminger, hans-peter.huth, joachim.sokol}@mchp.siemens.de

Abstract

In this paper, we outline a new connection admission control and online traffic engineering framework for small networks using differentiated services. Decisions are made at the edge routers of the network. Multiple label switched paths are set up between each pair of edge routers. When a new connection arrives at an edge router, each path is evaluated to see whether it is able to carry the connection. Then, the best path of the remaining ones is picked. The evaluation of paths is based on state information gathered from the queues on each path. We show the results this scheme achieves based on simulations.

1. Introduction

The need for different service qualities in Internet protocol-based networks is growing stronger as the Internet protocol (IP) turns out to become the universal network architecture. Different applications require different qualities of service (QoS) from the network. To address this in IP, two approaches have been developed. First, strict QoS guarantees are accomplished by the integrated services (IntServ) architecture [1] in conjunction with the resource reservation protocol (RSVP) [2] used for signaling. This framework allows reserving resources on a path through the network to achieve an end-to-end QoS guarantee, but it has shortcomings with regard to scalability since every router on that path has to maintain per-flow state information. Second, the differentiated services (DiffServ) architecture [3], which gives a loose notion of QoS, enables the network to optimize the transport of data packets according to certain requirements. DiffServ only uses different per-hop behaviors (PHBs) for different classes of traffic rather than giving guarantees on these transport characteristics. Such PHBs are implemented on every DiffServ-enabled router by mapping different traffic aggregates to different queues. These traffic aggregates are distinguished by the DiffServ codepoint (DSCP) in the IP header. Several PHBs have already been defined:

- Expedited forwarding (EF), a high priority service trying to achieve zero packet loss, minimal queuing delay, and minimal jitter [4],
- Assured forwarding (AF), a group of several PHBs giving a variety of different forwarding assurances by defining four classes (distinguished by the resources available per class, namely buffer space and bandwidth) with three different drop precedences [5],
- Best effort (BE), a low priority service equivalent to the service in DiffServ-unaware networks.

DiffServ alone does not guarantee any quality of service in an end-to-end fashion. All it does is providing differentiated service to packets on a hop-by-hop basis. To address this, DiffServ can be used in conjunction with connection admission control (CAC) to ensure that the network can support additional data without degrading the QoS of the data already admitted. Moreover, out-of-profile flows or flow aggregates can be addressed by means of traffic shapers or policers.

Several connection admission control schemes have been proposed to address these QoS requirements in IP networks. Bianchi *et al.* proposed in [6] a short probing phase at the beginning of a connection to measure the achievable throughput. In [7], Borgonovo *et al.* proposed a similar scheme that incorporates different priorities for probe packets. Li *et al.* introduced the fair intelligent admission control (FIAC) scheme in [8]. In FIAC, end nodes send resource discovery packets through the DiffServ domain, in which the core routers in that domain fill in information on their QoS state.

Another issue is traffic engineering (TE). Multiple paths can be set up between the edge router of a network domain. There are multiple frameworks that address how to

0-7803-8367-2/04/$20.00 ©2004 IEEE

decide which path to choose for new incoming traffic flow. In [9], Elwalid *et al.* presented a framework for MPLS adaptive traffic engineering (MATE). MATE gathers state information of paths by using probe packets that are sent periodically from the ingress to the egress router of a path. These statistics and a path cost function are used to decide to which path a traffic flow is shifted. Barlow *et al.* proposed the local state fair share bandwidth (LSFSB) algorithm, a traffic engineering algorithm for radio access networks, in [10]. The algorithm relies – as the name implies – on local state information only. The routers on the edge of the network choose the path on which admitted traffic is forwarded.

In this paper, we propose a related approach. The edge routers of the DiffServ domain query the queue states in the network core to render a decision whether to admit traffic and on which path this traffic traverses the network. These paths are realized by means of multiprotocol label switching (MPLS) [11]. To evaluate the performance of this algorithm, we show simulation results based on an augmented version of the network simulator version 2.1b9a (ns-2.1b9a) available at [12].

The remainder of this paper is organized as follows. Section II states the network setup chosen for this research and some assumptions we made. Section III outlines the TE and CAC algorithm. Section IV shows results based on our simulations. The paper is concluded by Section V.

2. Network Setup and Assumptions

2.1. Topology

For this work, we assume a topology related to our previous research. The network we assume is a simple radio access network (RAN) setup as in [10]. Figure 1 shows this topology. Mobile devices connect via the radio access servers (RASs) on the rim of the network. Those connect to edge routers (ERs) on the edge of the core network. There is one special ER called the edge gateway (EGW) in the core network that interfaces other RANs and the Internet. To keep this nomenclature general, we also consider the possibility of other routers in the core network that do not inject traffic. These routers are referred to as core routers (CRs). In the scope of this research, all ERs and CRs make up one traffic engineering and CAC domain. The ERs make CAC decisions and pick a path for incoming traffic on a per-connection basis.

Between each pair of ERs, three disjoint label switched paths (LSPs) are established. This results in a total of 60 LSPs in the network, which are set up at the beginning of the simulation.

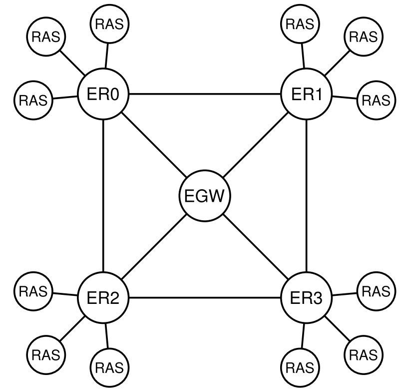

Figure 1. Simple radio access network topology

2.2. Queues and Queue States

Every router in the DiffServ domain has three physical queues per outgoing link, one for EF traffic, one for AF traffic, and one for BE traffic. The queues are serviced using a weighted round robin scheme with weights 7, 4, and 1. Three different metrics are considered: the average queue length (as seen by the random early detection algorithm [13] of the queue), the drop rate for each of the three physical queues, and the link utilization (based on the idle time of the queue server). The two latter values are measured over a 100 ms interval. The resulting value is then filtered using an exponentially weighted moving average (EWMA).

The edge routers can read these values without any delay, i.e. there is no signaling packet involved. This is a vast simplification for the implementation for the simulation software. Moreover, we do not expect this property to have a big impact on the results since our algorithm incorporates EWMA values rather than relying on exact instantaneous measurements.

2.3. Traffic Load

We consider four different services with distinct traffic flow characteristics in our simulator. The characteristics used were randomly chosen to demonstrate proof of concept. The first one uses the EF PHB and the User Datagram Protocol (UDP) for data transport. 162 byte-sized packets are sent with a constant bit rate of 64.8 kbps in both directions. The second class uses the AF PHB and UDP. Packets are sized 500 bytes, and the bit rate is 100 kbps in both directions. The third class uses the AF PHB and the Transmission Control Protocol (TCP) for data transport. The packet size is 500 bytes. The forth class uses the BE PHB with TCP and a packet size of 1000 bytes. Both TCP traffic classes send with variable bit rates, which is inherent to TCP. However, at the ingress ER, the TCP flows are policed to 100 kbps using a token bucket. Moreover, both TCP traffic classes are unidirectional. The data on the reverse path consists only of 40 byte-sized acknowledgment packets. To uniquely identify each class, we label them by their PHB and their transport protocol: *ef-udp*, *af-udp*, *af-tcp*, and *be-tcp*. The probability that a new call is an *ef-udp* call is $\frac{1}{3}$. For *af-udp*, the probability is $\frac{1}{6}$; for *af-tcp*, it is $\frac{1}{6}$; and for *be-tcp*, it is $\frac{1}{3}$. When a call fails to achieve its minimum QoS requirements, it is terminated. For *ef-udp*, no more than one loss per 2 second interval is tolerated. Both *af-udp* and *af-tcp* have to maintain a loss rate of no more than 2 packets during the last 2 seconds. Currently, there is no minimum QoS requirement for *be-tcp*.

Calls arrive exponentially distributed at the rim of the network with an inter-arrival time of λ. Two parameters determine the distribution of connections in the network: p_{EGW} is the probability that the EGW is part of a connection, and p_{ER0} is the probability that ER 0 is part of a connection.

3. The Path Queue State-based TE/CAC Algorithm

Our proposed algorithm works in two steps. The first step is to check if new incoming traffic flows can be admitted to one of the three possible LSPs, and LSPs not capable of carrying more traffic are pruned from the set of feasible paths. Then a path is selected from the set of remaining paths. The decision for both steps is based on the three metrics described in Section II B. The highest link utilization on a path is used to calculate an estimated available bottleneck bandwidth for that path, denoted A. A path is not pruned from the set of feasible paths if the bandwidth requirement B (see Section II C) of the connection to be admitted satisfies the condition $B \leq \beta \cdot A$ where β is a parameter depending on the traffic class. Additionally, all average queue lengths are add up. Since the average packet sizes and the link capacity are known, this yields a delay estimate δ. However, δ is just an estimate because it depends on the ratios of the service times each per-hop behavior gets at each link of the path. This delay estimate has to be lower than a parameter $\delta_{\max}$, which depends again on the traffic class. The last requirement for a path to stay in the set of feasible paths is that the sum of all drop rates l is lower than a traffic class-dependent parameter $l_{\max}$. The values for β, $\delta_{\max}$, and $l_{\max}$ we chose are presented in Table 1.

To avoid that multiple ERs admit connections concurrently to paths that share a common link, each queue on a path a connection is admitted to receives a penalty. For EF connections the delay estimate for each queue is increased by 1 msec for an interval of 1 sec. Moreover, for EF and AF the EWMA is decreased by $1.5 \cdot B$. For BE, the EWMA value is decreased by $0.5 \cdot B$.

Table 1. Admission control parameters

	β	$\delta_{\max}$	$l_{\max}$
ef-udp	3.0	29 msec	$5 \cdot 10^{-8}$
af-udp	3.0	233 msec	$1 \cdot 10^{-7}$
af-tcp	3.0	233 msec	$1 \cdot 10^{-7}$
be-tcp	0.5	∞	∞

Currently, the selection of a path from the set of feasible paths is implemented rather straightforward. For EF and BE connection, the path with lowest estimated delay is selected. AF connections are routed over the path with the lowest estimated loss.

The actual implementation of the queue state querying and the setting of a queue penalty in real networks could be done by using setup packets that are sent between the two participating ERs before the actual data is flowing. These setup packets are intercepted and interpreted by other QoS-aware routers on the path. Note that these mechanisms do not require per-flow state information. However, for distributing flows over different paths, ERs have to be aware of different flows to assign the appropriate MPLS labels to them.

4. Results

The following results are gathered by simulating the outlined networking system for 150 seconds after a 60 second warm-up period. In Table 2 we show the different parameters we simulated. To be able to easily name a set of parameters, we assign a tag consisting of three letters to each of them. The first letter indicates the traffic distribution, 'h' for a hotspot and 'e' for an evenly distribution. The second letter shows the call arrival rate, 'l' for low, 'm' for medium, and 'h' for high. The third letter denotes whether traffic engineering (TE) was enabled ('t') or not (shortest path routing, 's'). In the non-TE case, all calls are admitted to the network. Hence, there is no CAC. However, if a call cannot maintain the minimum QoS outlined in Section II C, it is terminated. Therefore, all calls maintain the minimum QoS as long as they are active. We use this property as a baseline to evaluate our algorithm.

Table 3 gives a general idea about the load the network has to cope with for different parameter sets. For low and medium call arrival rates (i.e. long interarrival times), TE/CAC outperforms the non-TE approach. This is due to the fact that calls are terminated in the non-TE case quite frequently and cannot be replaced by new calls. The load

Table 2. Simulation parameters

tag	λ [sec]	p_{EGW}	p_{ER0}	TE enabled
hls	0.1	0.90	0.70	no
hms	0.05	0.90	0.70	no
hhs	0.005	0.90	0.70	no
hlt	0.1	0.90	0.70	yes
hmt	0.05	0.90	0.70	yes
hht	0.005	0.90	0.70	yes
els	0.1	0.20	0.20	no
ems	0.05	0.20	0.20	no
ehs	0.005	0.20	0.20	no
elt	0.1	0.20	0.20	yes
emt	0.05	0.20	0.20	yes
eht	0.005	0.20	0.20	yes

Table 3. Number of packets delivered and average link utilization

tag	packets delivered	avg. util.	tag	packets delivered	avg. util.
hls	2410923	0.24	hlt	3616202	0.50
hms	3743116	0.37	hmt	5338285	0.82
hhs	8872623	0.89	hht	6054646	0.94
els	3583866	0.44	elt	3679181	0.52
ems	5691657	0.69	emt	5726649	0.87
ehs	8530194	0.98	eht	6539365	0.94

for high call arrival rates is higher in the non-TE case. Even though calls are terminated rather quickly, they are replaced by new calls. Moreover, since no CAC is limiting the amount of calls, the network becomes extremely utilized.

4.1. Hotspot Load

First, we investigate results for a hotspot scenario. In this scenario, there is a 70% chance that the EGW is part of a connection. Moreover, ER0 is part of a connection with a probability of 80%. Table 4 gives on overview over the CAC results for EF traffic. Our algorithm performs generally well. However, about 8.62% of the calls in the medium arrival rate setup fail. Surprisingly, there is no problem for a higher loads.

Figure 2 and Figure 3 show the link utilizations for medium connection arrival rates for the non-TE and TE scheme. The thickness corresponds to the utilization. The

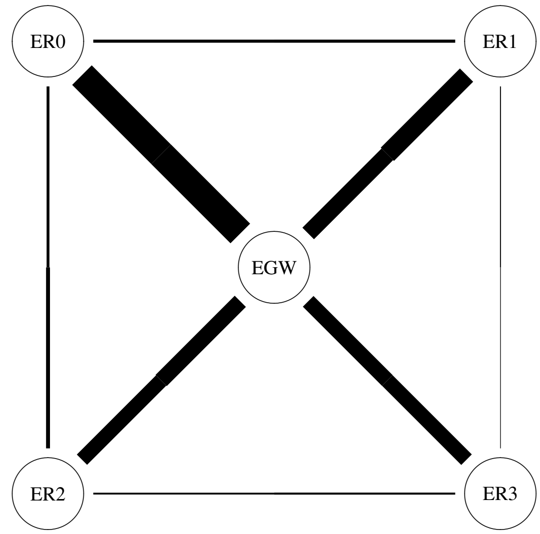

Figure 2. Link utilization for parameter set ‘hms’

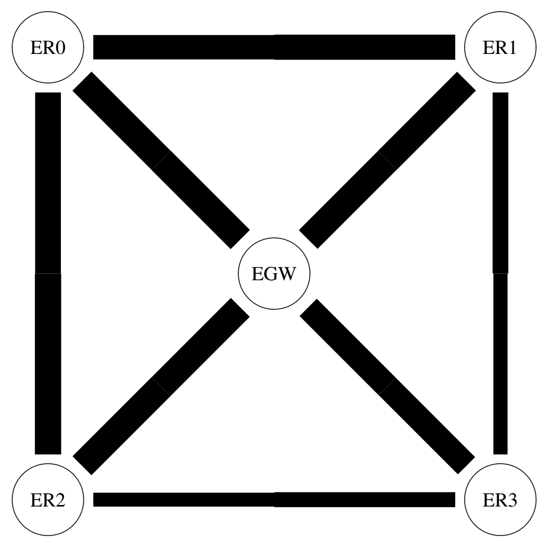

Figure 3. Link utilization for parameter set ‘hmt’

Table 4. Results for EF and a hotspot load

tag	packets dropped	calls accepted	calls successful [%]
hls	661	679	64.06
hlt	0	667	100.00
hms	1901	1382	48.19
hmt	193	998	91.38
hhs	32075	13771	10.20
hht	0	981	100.00

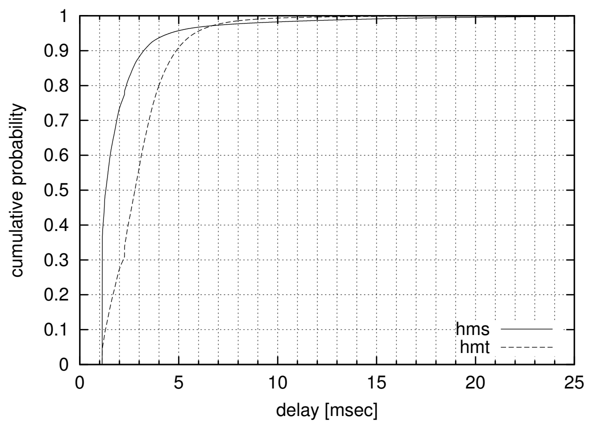

Figure 4. Cumulative distribution functions of EF packet delay for a hotspot load and a medium connection arrival rate

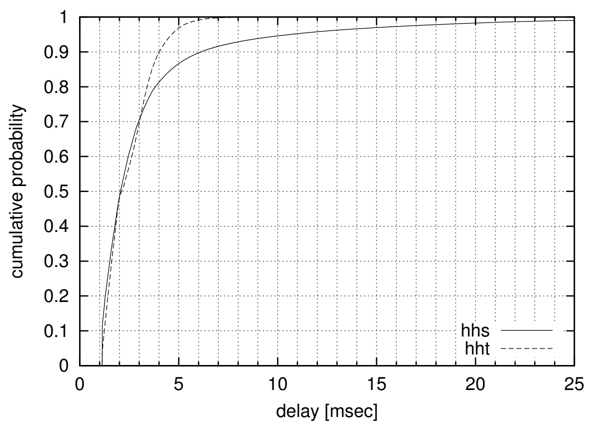

Figure 5. Cumulative distribution functions of EF packet delay for a hotspot load and a high connection arrival rate

part of the line adjacent to a node depicts the amount of traffic sent out on this link while the distant part of the line corresponds to the amount of traffic received on this link. The spare capacity in the network (denoted by the thin lines in Figure 2) is put to use by the TE scheme.

Figure 4 shows the cumulative distribution functions (CDFs) for the delay of EF packets for both the TE and non-TE scenario for a medium connection arrival rate. Since some links in the non-TE case have very low utilizations, the overall delay distribution profits from that for low delay thresholds (left part of the curve). However, the TE case guarantees a tighter upper delay bound and breaks even at a delay threshold of about 7 msec.

The CDFs for high connection arrival rates are shown in Figure 5. Since most links are highly utilized, the curves are more comparable than those in Figure 4. For very low

Table 5. Results for AF and a hotspot load

tag	packets dropped	calls accepted	calls successful [%]
hls	836	627	81.02
hlt	144	625	99.52
hms	3514	1300	60.00
hmt	54	935	99.79
hhs	69453	13496	19.08
hht	8	1047	100.00

Table 6. Results for EF and an evenly distributed load

tag	packets dropped	calls accepted	calls successful [%]
els	75	668	95.96
elt	0	668	100.00
ems	1046	1371	73.60
emt	221	1101	90.92
ehs	33013	13687	8.94
eht	0	1052	100.00

delays, both curves are nearly equal. At about 3 msec the advantage of the TE scheme becomes apparent. Furthermore, the TE curve indicates that 99% of all packets are below a 6 msec threshold.

Table 5 shows results for packets forwarded with the AF PHB. Our algorithm achieves the targeted QoS goals well. Note, that the performance of the algorithm increases as the network gets more loaded. This is due to the fact that a loaded network is generally more predictable and tends to less bursty traffic characteristics.

Since we do not consider particular QoS requirements for BE traffic, we do not investigate BE performance in the scope of this paper.

4.2. Evenly Distributed Load

Generally, an evenly distributed traffic load is unfavorable for our TE scheme since using any path other than the shortest path uses up more network resources. However, because of short imbalances and with well-chosen parameters in our algorithm that favor the shortest path, our algorithm functions well, and thus, these effects do not become apparent in the results presented.

Table 6 contains simulation results for the CAC of connections using EF packets. Again, our algorithm needs refinement regarding medium arrival rates of connections.

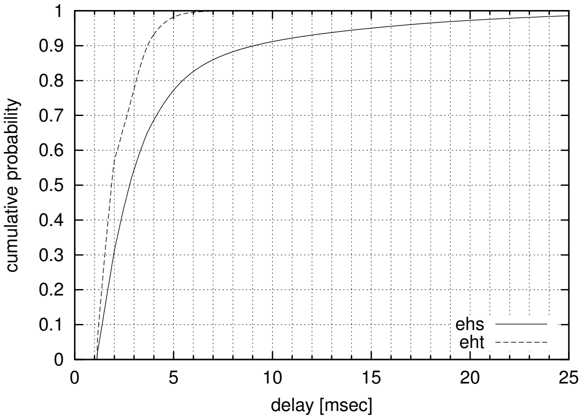

Figure 6. Cumulative distribution functions of EF packet delay for an evenly distributed load and a high connection arrival rate

Table 7. Results for AF and an evenly distributed load

tag	packets dropped	calls accepted	calls successful [%]
els	5	705	100.00
elt	0	705	100.00
ems	1259	1407	89.05
emt	67	1124	99.56
ehs	72157	13799	18.16
eht	25	1156	100.00

The corresponding CDFs for packet delay are shown in Figure 6. The TE algorithm performs equally well compared to the hotspot scenario and significantly better than the non-TE scheme. The non-TE scheme performs worse when compared to the hotspot scenario. This is due to links that are not under full load in the hotspot scenario. Results for AF traffic are shown in Table 7. The proposed algorithm is able to maintain the targeted QoS level.

5. Conclusion

We presented a new framework to achieve firm QoS goals in small DiffServ network domains such as radio access networks. The framework deals with connection admission control as a means to keep the network at a load that does not negatively impact the service ongoing connections receive. Additionally, the framework includes an adaptive traffic engineering approach. The network state is evaluated and based on this network state, new calls are distributed over a number of alternative paths.

In general, the proposed TE/CAC scheme enhances the overall service quality of the assumed radio access network. Some parts of our algorithmic framework still need additional work, e.g. the amount of failed EF calls for medium arrival rates (Table 4 and Table 6) has to be minimized. Except for this issue, QoS goals for connections are achieved. Furthermore, the TE part can harness spare capacity in the network to avoid congested links. All computations for that happen online, which renders this scheme adaptive to quick changes in the network state without the necessity of being aware of the traffic distribution.

References

[1] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview," IETF RFC 1633, June 1994.

[2] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (RSVP) – version 1 functional specification," IETF RFC 2205, September 1997.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," IETF RFC 2475, December 1998.

[4] B. Davie, A. Charny, J. Benett, K. Benson, J. Le Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, "An expedited forwarding PHB (per-hop behavior)," IETF RFC 3246, March 2002.

[5] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group," IETF RFC 2597, June 1999.

[6] G. Bianchi, A. Capone, and C. Petrioli, "Throughput analysis of end-to-end measurement-based admission control in IP," in *Proc. IEEE INFOCOM 2000*, vol. 3, 2000, pp. 1461–1470.

[7] F. Borgonovo, A. Capone, L. Fratta, M. Marchese, and C. Petrioli, "PCP: A bandwidth guaranteed transport service for IP networks," in *Proc. IEEE ICC '99*, vol. 1, 1999, pp. 671–675.

[8] M. Li, D. B. Hoang, and A. J. Simmonds, "Fair intelligent admission control over DiffServ network," in *Proc. IEEE International Conference on Networks 2003*, 2003, pp. 501–506.

[9] A. Elwalid, C. Jin, S. H. Low, and I. Widjaja, "MATE: MPLS adaptive traffic engineering," in *Proc. IEEE INFOCOM 2001*, 2001, pp. 1300–1309.

[10] D. Barlow, H. Owen, V. Vassiliou, J. Grimminger, H.-P. Huth, and J. Sokol, "Router-based traffic engineering in MPLS/DiffServ/HMIP radio access networks," in *Proc. IASTED International Conference on Wireless and Optical Communications*, 2002, pp. 360–365.

[11] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," IETF RFC 3031, January 2001.

[12] The network simulator – ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/

[13] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.