

## Analytical Model for a Host-based Service Differentiation Scheme

Henrik Lundqvist and Gunnar Karlsson

KTH, Royal Institute of Technology  
P.O. Box Electrum 229, 164 40 Kista, Sweden  
henrik.lundqvist@imit.kth.se, gk@imit.kth.se

**Abstract:** Probe-based admission control has been proposed as a traffic control that can provide real-time traffic with quality of service and prevent congestion without any support from the network. We study methods to provide isolation and fairness between real-time traffic and elastic traffic by adding forward error correction (FEC) and by using appropriate probing procedures. An analytical model that can be used for evaluation of the scheme is presented and shows good correspondence with simulations.

**Keywords:** Differentiated services, quality of service, traffic control, Internet, modeling

### 1. INTRODUCTION

The deployment of quality of service on the Internet has had very slow progress. This is disappointing considering that it has been an active research area for many years and several schemes have been proposed and implemented. However, deployment of quality of service requires new business models to make it worthwhile for operators. To define traffic classes with different prices, deploy billing systems and negotiate service level agreements between different ISPs is not a trivial task. Therefore, it would be an advantage if traffic with different requirements could be differentiated at the edges without any need for upgrading networks. Such differentiation would work well with flat rate or volume based charging.

With a volume based charging there is no need for the service provider to distinguish between different types of traffic, and no class of traffic is prioritized above the other. However, it is in the interest of the user to get as high utility as possible out of each transmitted and received bit: In particular, for applications that have strict requirements on the quality, like interactive video, it would be better not to start a session if the quality is likely to be poor. The requirement of file transfers is quite different; since the goal is only to minimize the transfer time it is possible to use the network almost regardless of the loss rate.

The scheme investigated here uses the same type of differentiation between two traffic types, streaming and elastic, as suggested by Roberts and Oueslati-Boulahia in [1]. While that proposal is based on modifying the network infrastructure, our scheme is based on changes in the end hosts only. The traffic control for real-time traffic uses probe-based admission control whereas the elastic traffic uses normal TCP congestion control. Fair differentiation of streaming and elastic traffic has also been investigated analytically by Key et al. [6]. In this paper we study the details of the specific mechanisms required to achieve differentiation from

the edges.

The most well known of previously proposed congestion control schemes for streaming traffic is TCP friendly rate control, which implements a rate control that reacts slower than TCP congestion control [10]. Fairness between streaming traffic and TCP is provided by using an equation for TCP throughput as a function of loss rate and adjust the rate accordingly. The same type of fairness comparison is used in our scheme, but the traffic control for streaming traffic is based on admission control rather than rate control. The motivation for this is that human perception tends to favor consistent quality throughout a session. With admission control an admitted session does not need to reduce its quality during the life-time of the session, whereas rate control will react on changes in congestion level during the session.

Probe-based admission control has earlier been investigated as a separate service class, separated from best-effort service by scheduling in the routers [3]. The efficiency of probing as a predictor of the quality of a session has also been investigated by measurements of IP-telephony sessions on the Internet, with promising results [9]. A previous simulation based study of probe-based admission control without separation from best effort in the network nodes indicates the feasibility of this kind of service differentiation [2]. In this paper we develop models and use them to improve and evaluate the scheme. In order to make the analysis tractable we constrain ourselves to relatively simple models and focus on putting them together to get useful results and improve the proposed traffic control methods. In particular the inclusion of FEC and probabilistic guarantees for the admitted sessions is treated.

In Section 2 the probe-based admission control and the parameter settings are described. Furthermore, two probing strategies are evaluated with respect to TCP fairness and delay. In Section 3 a network model is introduced in order to evaluate the effect of the probabilistic admission control on the network performance and the dynamic behavior of the scheme. Section 4 contains a comparison between the results from the model and from simulation to validate the relevance of the model. The conclusions of the study are presented in Section 5.

## 2. Probe-based Admission Control

The principle of probe-based admission control is to measure the properties of the channel, to determine whether the quality is sufficient, before starting a session. For this purpose a number of packets, a probe, is sent to the receiver. The receiver measures the loss rate of the probe and depending on the result the session is either accepted or rejected. For a more detailed description of probe-based admission control refer to [3].

### 2.1. Admission probability

We assume that an application has a certain requirement on the packet loss rate,  $p_{req}$ , and on the maximum delay. To simplify the analysis in the rest of the paper we will assume that losses experienced by a session are independent, which means that the number of losses within a block of packets is binomially distributed. Note that the multiplexing in a network helps in reducing the dependence between losses in an individual stream. Based on the loss rate, a decision is taken on whether to start the session or not. If the threshold for the admission control is  $T$  packets, i.e. the session is rejected if more than  $T$  probe packets are lost, and the actual loss rate of the path is  $p$ , the probability that a probe of length  $N_{probe}$  packets experiences too high loss rate is given by:

$$p_{adm} = 1 - \sum_{i=T+1}^{N_{probe}} \binom{N_{probe}}{i} p^i (1-p)^{N_{probe}-i}. \quad (1)$$

## 2.2. TCP fairness

The threshold value for admitting a session should not only take into account the requirements of the application but also the fairness between elastic and streaming traffic. A first approach to provide fairness is that if the loss rate is high, TCP sessions get low throughput and therefore real-time sessions with bit rates exceeding the TCP throughput should not be admitted. A relation between the TCP throughput and the loss rate is required for this purpose. We will use a simple formula proposed by Mathis et al [4]:

$$R_{TCP} = \frac{MSS}{RTT} \sqrt{\frac{C}{p}}, \quad (2)$$

where  $MSS$  is the maximum segment size,  $RTT$  is the round trip time,  $p$  is the loss rate and  $C$  is a constant, which is set to 1.5 following the result in [4]. Note that the choice of the TCP equivalent rate requires knowledge of the maximum segment size for a TCP session and the RTT. We will assume that these are known, for example path MTU discovery and ICMP echo requests could be used to measure the values before setting up a session, or the probe packets themselves could be used for RTT estimation.

For a session with peak rate  $R_{UDP}$ , equation (2) with  $R_{TCP} = R_{UDP}$  can be solved for  $p$  to find the admission threshold  $p_{eq}$  as:

$$p_{eq} = \frac{MSS^2 C}{RTT^2 R_{UDP}^2}. \quad (3)$$

A session with peak rate  $R_{UDP}$  and a requirement on the loss rate,  $p_{req}$ , would therefore be admitted if the experienced loss rate during the probe is lower than  $\min(p_{req}, p_{eq})$ . This guarantees that there is no unfair advantage to the benefit of admission-controlled traffic. However, if the TCP equivalent loss rate is higher than the requirement of the UDP session, there is instead an advantage for TCP since the average TCP session will get a higher throughput than the UDP sessions. This can be mitigated using FEC.

## 2.3. Forward error correction

The packet loss rate in the Internet depends both on applications with different loss tolerance and the properties of TCP congestion control. Since TCP will adapt to different loss rates in ways that depend on parameters such as the RTT and the segment size, the loss rate of the network can vary strongly. By using FEC, applications can get a loss rate that is acceptable even if the loss rate in the network is high. This also means that real-time applications with different requirements can coexist, which is otherwise a problem when there is no separation of traffic classes in the network, since the applications with the highest loss threshold can increase the loss level for all other traffic classes.

We consider block codes, where  $M$  redundant packets are added to a block of  $N$  data packets and allow the recovery of  $M$  lost packets in the block of totally  $N+M$  packets. For simplicity the packet losses are still assumed to be uncorrelated when we evaluate the gain of

FEC. In this case the average packet loss rate after FEC when the loss rate of the path is  $p$  can be calculated as:

$$p^* = \sum_{i=M+1}^{N+M} \binom{N+M}{i} p^i (1-p)^{N+M-i} \frac{i}{N+M} \quad (4)$$

Since the efficiency of FEC increases with longer block lengths the block should be as long as the delay tolerance of the application, that is  $R_{UDP} \cdot d / P_{size}$ , where  $d$  is the delay and  $P_{size}$  is the size of UDP packets. The assumption here is that the application requires a specific packet loss rate, hence the last factor in (4) which converts the block loss probability to a packet loss probability. Of course it would be possible to define the loss requirement in block losses if it were more appropriate for the application.

## 2.4. Parameter setting for probing

The goals with the probe-based admission control are twofold: To provide fairness between streaming and elastic traffic and to provide sufficient quality to admitted sessions. Since the admission control only provides probabilistic guarantees it is necessary to include some margins to provide loss rates that are low enough for admitted sessions. To make a reasonable estimation of the loss rate, the probe should be long enough to at least contain in the order of ten lost packets for loss rates close to the threshold level. In this case the number of losses within the probe can be estimated as Gaussian distributed [3]. The admission threshold can then be set such that the percentile of sessions that are admitted when the loss rate is too high is lower than a given requirement, for example five percent. Hence, when the margin is taken into account the required threshold is reduced from  $p_{req}$  to:

$$p_{marg} = p_{req} - z_R \sqrt{\frac{p_{req}(1-p_{req})}{N_{probe}}}, \quad (5)$$

where  $z_R$  is the R-percentile of the normal distribution, for example 1.64 for the 95-percentile. This means that  $p_{marg}$  should be compared with the  $p_{eq}$  from (3) to determine whether FEC should be used. If  $p_{marg} < p_{eq}$  it is beneficial to add redundancy.

When FEC is used the safety margins should provide probabilistic guarantees for the quality of the session, without impairing the fairness between TCP and real-time traffic. Therefore a margin is added to  $p_{eq}$  from (3) to compensate for the inaccuracy in the estimation of the loss rate:

$$p_{FECmarg} = p_{eq} + z_R \sqrt{\frac{p_{eq}(1-p_{eq})}{N_{probe}}}, \quad (6)$$

This loss rate,  $p_{FECmarg}$ , is used to calculate the loss rate after decoding,  $p^*$ , from (4), which is then compared with the application requirement  $p_{req}$ . If  $p^* < p_{req}$ ,  $p_{eq}$  can be used as admission threshold, and the loss rate experienced by the application will still be low enough. Hence, when FEC and margins are taken into account a session is admitted if the loss rate during the probe is lower than  $p_{eq}$  if FEC is added or  $\min(p_{marg}, p_{eq})$  if FEC is not added.

The parameters of the FEC and the admission threshold can now be set as follows, we can all this Algorithm 1: When the data rate of the application,  $R_{UDP}$ , is lower than  $R_{TCP}$  this algorithm increases the admission threshold by adding enough redundancy to make  $p^*$  lower

than  $p_{req}$ . For the total rate including redundancy  $R_{FEC}$  a new  $p_{eq}$  is calculated from (3) and used as admission threshold. Since this threshold is higher than the original  $p_{eq}$  the admission probability is higher without being unfair.

The result of Algorithm 1 is illustrated in Fig. 1. It is assumed that the  $RTT$  is 0.1 seconds, the  $MSS$  is 500 bytes and the TCP throughput is calculated using  $C=1.5$ , following the results in [4]. For example, a session with a rate of 400 kb/s, a delay requirement corresponding to a FEC block length of 20 packets and a loss requirement of one percent would add one redundant packet per block and have an admission threshold of 1.36 percent. The other two applications both have 200 kb/s data rate and 20-packets FEC block length, the first requires a loss rate of three percent and needs to add two redundant packets per block, the second requires a loss rate of 0.1 percent and must add 5 redundant packets per block. The admission thresholds from (3) for the resulting total rates of 220 kb/s and 250 kb/s, respectively, are 4.96 percent and 3.84 percent loss rate. The probability of being admitted depends on the threshold and the loss rate of the path. As can be seen in Fig. 1, the expected goodput of a UDP session, calculated as  $p_{adm} \cdot R_{UDP}$ , is lower than the expected throughput of a TCP session at the same loss rate. Actually, if the loss rate of the path is equal to the admission threshold (i.e. at the vertical lines in Fig. 1), the admission probability,  $p_{adm}$ , is 50 percent. This implies that the expected throughput of the UDP sessions will be lower than for the TCP sessions, even when the redundancy is included in the UDP rate.

Fig. 2 shows the impact of choosing the constant  $C$  in (1) differently. A higher value for  $C$  makes the admission control less restrictive, so that more redundancy can be added and the sessions can be admitted at higher loss rates.  $C$  can be considered as a knob that can be turned in order to change the rate sharing between UDP and TCP. The curves can of course never be identical because of the different nature of the traffic types and control mechanisms, therefore the fairest choice of  $C$  cannot be easily defined.

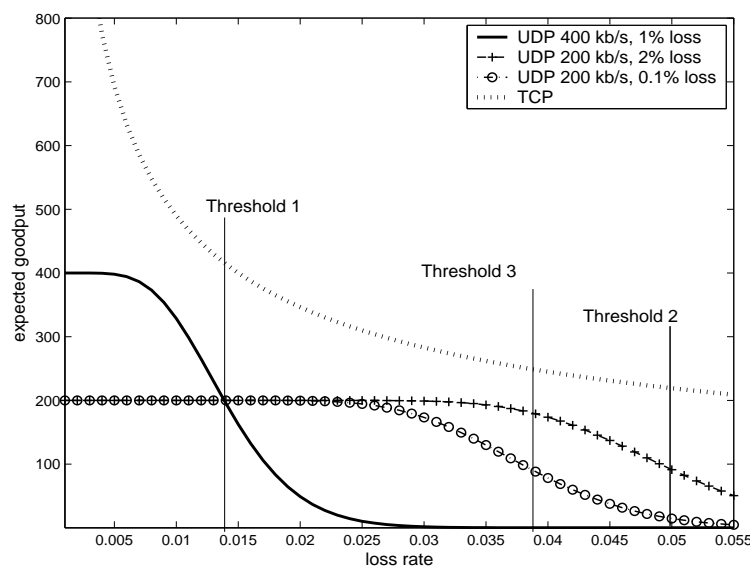


Figure 1. The admission threshold is higher for sessions with lower demands on rate and higher acceptance for losses. In the figure the redundancy is not included in the rate of the UDP sessions.

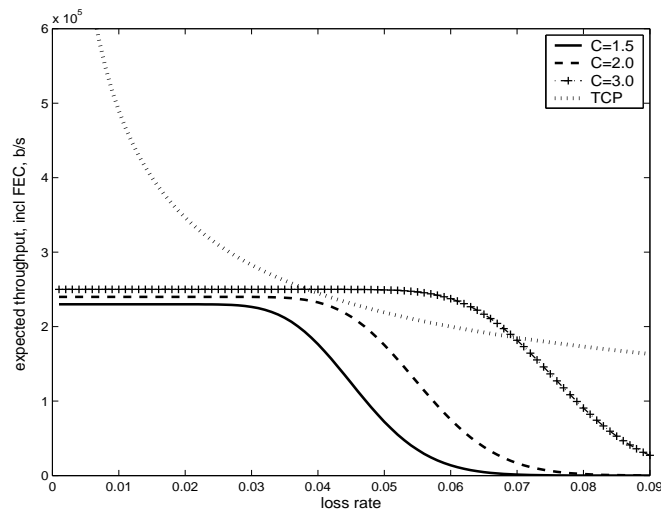


Figure 2. The constant  $C$  has an impact both on how much redundancy is added and on how the capacity is shared between TCP and UDP traffic. The UDP rates include redundancy.

### 3. Operating Point

To evaluate the proposed admission control scheme we use a model that relates the loss rate in the network to the admission control and TCP congestion control is required.

#### 3.1. Network Model

The network consists of a set of links  $J$  where each link has a limited capacity  $C_j$ . The traffic in the network consists of different classes of flows on different paths through the network. The number of elastic flows on a path  $r \in R$  is called  $N_r$  and the throughput of each such flow is  $x_r$ . The number of streaming flows on a path  $s \in S$  is called  $M_s$  and the rate, including redundancy, of each such flow is  $R_s$ . If there are streaming flows with different rates using the same path through the network, the paths will be given different indices but will be routed the same way. The routing of the elastic flows is determined by a  $|J| \cdot |S|$  matrix  $A$ , where element  $A_{jr} = 1$  if route  $r$  passes through link  $j$ . The routing of the streaming flows is determined by a  $|J| \cdot |S|$  matrix  $B$ , with  $B_{js} = 1$  if route  $s$  passes through link  $j$ . The load from probes on path  $s$  is denoted  $Q_s$  and is calculated as the probe length / average session length times the arrival intensity.

Losses can occur at a router connecting to a congested link. Our approach is to use a simple loss model, namely an M/M/1/K-queue, and see whether the results are accurate enough. The load is defined as:

$$\rho_j = \frac{\sum_r A_{jr} N_r x_r + \sum_s B_{js} (M_s R_s + Q_s R_s)}{C_j} \quad (7)$$

The loss rate of the M/M/1/K queue as a function of the load is:

$$p_j = \frac{(1 - \rho_j) \rho_j^K}{1 - \rho_j^{K+1}} \quad (8)$$

For the evaluation of the proposed scheme the TCP model used is the one from Padhye et al. [8]:

$$x_r = MSS \min \left( \frac{W_m}{RTT}, \frac{1}{RTT \sqrt{\frac{2bp}{3}} + T_0 \min \left( 1, 3 \sqrt{\frac{3bp}{8}} \right) p(1+32p^2)} \right), \quad (9)$$

where  $b$  is the number of segments per acknowledgement,  $T_0$  the timeout period and  $W_m$  the maximum window size. Of course, this model could also be used instead of (2) in the parameter setting. However, since we only strive for an approximate fairness, (2) will be sufficiently exact and the definition of the parameter settings is simplified.

We are interested in knowing how the network resources are shared between TCP and streaming traffic and what the loss rate in the network will be for a given offered load of  $\lambda_s$  UDP sessions and  $N_r$  TCP sessions per path. We define an *operating point* as the number of admitted UDP sessions of each specific type and path,  $M_s$ , the throughput  $x_r$  of each TCP session and the loss rates for each link. The reason for defining the load differently for UDP and TCP is the different types of traffic control. Since the streaming traffic reacts to congestion only at the start of a session, the number of admitted sessions is a function of the loss rate. For the TCP sessions there is no blocking of sessions, but the congestion control will result in a throughput that depends on the load. Since we are interested in an operating point, which should reflect a long term average of the resource sharing, the average number of sessions is appropriate to use as TCP load. A load defined as arrivals of files of finite lengths can be added as an extension to this model, as in [6].

Now we have the following non-linear equation system:

$$\rho_j = \frac{\sum_r A_{jr} N_r x_r + \sum_s B_{js} (M_s R_s + Q_s R_s)}{C_j}, \quad j \in J$$

$$p_j = \frac{(1 - \rho_j) \rho_j^K}{1 - \rho_j^{K+1}}, \quad j \in J$$

$$p_r = 1 - e^{-\sum_j A_{jr} \ln(1-p_j)}, \quad r \in R$$

$$p_s = 1 - e^{-\sum_j B_{js} \ln(1-p_j)}, \quad s \in S$$

$$x_r = MSS \min \left( \frac{W_m}{RTT_r}, \frac{1}{RTT_r \sqrt{\frac{2bp_r}{3}} + T_0 \min \left( 1, 3 \sqrt{\frac{3bp_r}{8}} \right) p_r(1+32p_r^2)} \right), \quad r \in R$$

$$M_s = \lambda_s \left( 1 - \sum_{i=T+1}^{N_{probe}} \binom{N_{probe}}{i} p_s^i (1-p_s)^{N_{probe}-i} \right), \quad s \in S$$

Matlab has been used to simultaneously solve this equation system numerically.

### 3.2. Probe length and estimation accuracy

In this section the usefulness of the model is demonstrated by an investigation of the effect of the probe length. Because of the stochastic nature of the probing process, the number of admitted sessions will increase when the session arrival rate is high. Since this is potentially a problem for the scheme we analyze to what degree the loss rate increases due to probe load and due to higher number of accepted sessions. The scenario considered here is a 45 Mb/s link shared between 100 persistent TCP sessions and admission controlled UDP sessions with a rate of 100 kb/s each and a loss requirement of 1 percent from the application. Since there is a single link and one class each for TCP and UDP the matrices  $A$  and  $B$  will only contain one element each. Following algorithm 1, four redundant packets will be added to each block of 20 data packets and the resulting admission threshold is 5.5 percent. The probe length in this case is either one second or three seconds, corresponding to 313 and 938 packets respectively, where each packet is 40 bytes long. The average session length is 50 seconds, which means that the average load from probe traffic is either 2.4 kb/s or 7.2 kb/s multiplied by number of offered UDP sessions. The buffer size is 30 packets, TCP acknowledges every segment ( $b=1$ ) and the window size is large enough not to limit the throughput. Fig. 3 shows the loss rate in the network as a function of the offered load of UDP traffic. With a hypothetical deterministic admission control without loss estimation by probing, the loss rate would be limited to 5.5 percent since all UDP sessions would be blocked at higher loss rates. This can be used as a reference that the probe-based admission control should approximate. From the figure it is possible to see that three seconds probe length improves the accuracy of the loss estimation and makes the curve closer to the deterministic case. The effect of the extra load from the longer probes can be seen from the fact that the curve for three second probes crosses the 5.5 percent admission threshold at lower load than the one second probes. Since it is clear that there is only a short interval where the shorter probe length is closer to the deterministic admission control, we can conclude that the accuracy of the loss estimation is a more critical factor here than the load from the probes. The length of the probes must, however, also be limited in order to restrict the delay at session set up.

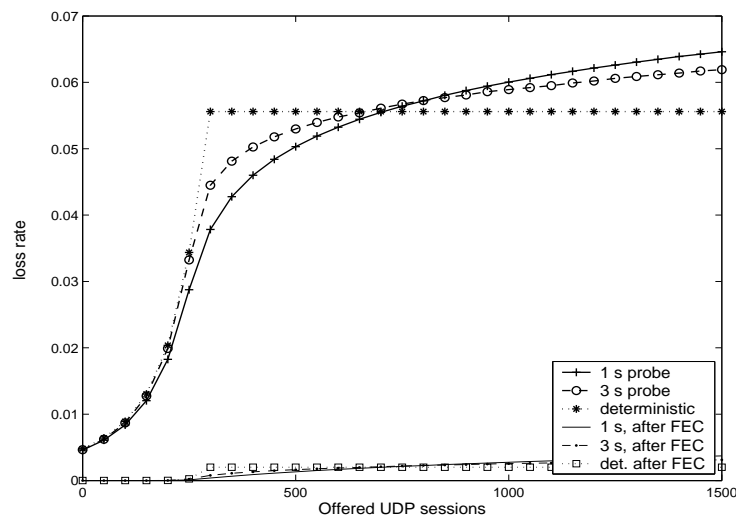


Figure 3. The probabilistic admission control decision has a larger impact on the operating point than the probe traffic.





## 5. CONCLUSIONS

A scheme for probe-based admission controlled has been adapted to coexist with TCP with the aid of models of protocol functions. The aim is to offer a useful service to applications with strict requirements on loss, throughput and delay, while still being fair to TCP controlled traffic. The analytical model developed here gives results that closely match simulation results and is therefore useful to give an understanding of the scheme and an efficient way of analyzing its properties. The evaluation shows that it is feasible to implement differentiation between elastic and inelastic services from the edges of a classless network. However, without network support there is no complete isolation between the different traffic types. Only probabilistic guarantees can therefore be given, but owing to the simple deployment it would be a first step towards introducing differentiated services in the Internet.

## REFERENCES

1. J. W. Roberts, S. Oueslati-Boulahia: Quality of Service by Flow Aware Networking, in *Phil. Trans. of The Royal Society of London, series A*, vol. 358, no. 1773, August 2000.
2. G. Karlsson, H. Lundqvist, I. Mas: Single Service Quality Differentiation, in *Proc. IWQOS*, June 2004.
3. I. Mas, G. Karlsson: PBAC: Probe-Based Admission Control, in *Proc. QoFIS*, Coimbra, Portugal, Sept. 2001.
4. M. Mathis, J. Semke, J. Mahdavi, T. Ott: The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. *ACM Computer Communication Review*, vol. 27, pp. 67-82, July 1997.
5. I. Marsh, F. Li: Wide Area Measurements of VoIP Quality, in *Proc. QoFIS*, October 2003.
6. P. Key, L. Massoulié, A. Bain, F. Kelly: Fair Internet Traffic Integration: Network Flow Models and Analysis, in *Proc. ITC 18*, 2003.
7. M. Roughan, A. Erramilli, D. Veitch: Network Performance for TCP Networks Part 1: Persistent Sources, in *Proc. ITC 17*, 2001.
8. J. Padhye, V. Firoiu, D. Towsley, J. Kurose: Modeling TCP Reno Performance: a Simple Model and its Empirical Validation, *IEEE/ACM Trans. on Networking*, Vol. 8, Issue 2, April 2000.
9. O. Hagsand, I. Mas, I. Marsh, G. Karlsson: Self-Admission Control for IP Telephony using Early Quality Estimation, In *Proc. Networking 2004*.
10. S. Floyd, M. Handley, J. Padhye and J. Widmer: Equation-based congestion control for unicast applications, in *Proc. ACM SIGCOMM*, 2000.