Study of rate-control policies for streaming sources based on feedback from a congested buffer

Erling Austreim and Peder J. Emstad

Centre for Quantifiable Quality of Service in Communication Networks[1]
Department of Telematics

Norwegian University of Science and Technology, Norway

Keywords: congestion control, rate-controlled streaming sources, fluid-flow

This paper presents a study of some policies for exercising control of the output from sources based on information from a congested buffer. The study of these control policies is motivated by the increasing amount of traffic from applications using transport protocols that has no built-in congestion-control mechanisms. The sources are here modeled as simple ON-OFF sources, and the study is carried out using Markov modulated fluid-flow modeling. We also study the different control policies based on discrete Markov chains. By doing this discretization it is possible to take the feedback delay from the congested buffer into account. There is also made a study of the accuracy of this simplification. This paper is based on part of a thesis written to fulfill the requirements of a master degree in technology at NTNU.

1. Introduction

Up to now the traffic volume in the Internet has mostly been TCP traffic due to browsing applications and file transfer. Streaming applications require high bandwidths, and now when this bandwidth becomes available, such applications have become more popular. This type of traffic also typically require bounds on delay and loss. The use of TCP as transport protocol is therefore inappropriate, and most of these applications therefore use RTP on top of UDP as their transport protocol. These protocols don't have any built-in mechanisms for congestion-control, in contrast to the traditional TCP protocol.

In the case of congestion, the TCP sources will typically reduce their sending rates due to implicit feedback from the network. The streaming sources on the other hand will send with

---

1. "Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence" appointed by The Research Council of Norway, funded by the Research Council, NTNU and UNINETT.
http://www.ntnu.no/Q2S/

the same high rate, leading to a situation where sources which react to congestion in the network may get less than their fair rate. Generally the exact behavior of an admitted streaming source like video is unknown.

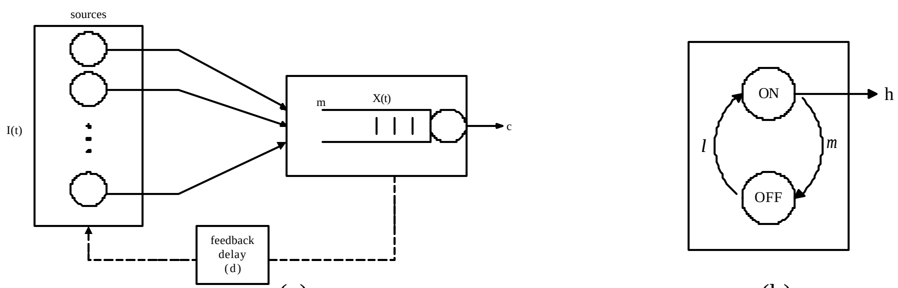
To ensure fairer bandwidth sharing and control with the quality of streaming applications in the case of congestion, one therefore may use mechanisms at higher layers which reduce the sending rates from the streaming sources in the case of congestion in the network. One way to do this is to use explicit feedback from the network to control the rate from such sources [1]. This would be most effective when the delay from the network to the sources is low. Streaming audio and video are examples of popular streaming applications, which can make use of such explicit feedback from the network. The rate can here be controlled directly by the encoder by dynamically adjusting the quantization parameters [2], or with use of scalable video encoding [3]. This will however reduce the decoded quality of the audio and video at the receiver, but allows the sources to cooperate and adapt in case of congestion and thus helps avoiding congestion.

We have looked at different analytical models and studied methodology that can be used to assess such a control scheme based on feedback from the network. We have not studied the consequences at the receiving end by applying such a control scheme. Without such a rate control, the quality of the streaming application will be degraded more in the case of congestion. By applying such a scheme the quality of the application can be controlled in a better way.

In part two we present the different models used to analyze the control scheme based on feedback from the network. These are based on markov modulated fluid-flow modeling and discrete Markov chains (DMC). The delay in the control loop affects the efficiency and this is studied using a DMC. Some results based on these analyses are presented in part three and part four concludes the paper.

2. Analytic models

We look at a system consisting of a number of sources sending data into a network, where the rate from each source is dependent on the state of the network. We assume that there is one bottleneck node in the network, so the sending rate *r* for each source is a function of the buffer content in this node. The system to be analyzed is then as sketched in figure 1, consisting of *n* sources and one node with buffer size of *m* data units. The buffer size has to be set such that the maximum delay encountered in the node satisfies the requirements for the streaming sources. We seek the distribution of the queuesize *X(t)* and especially the probability of overflow.



(a) (b)

Figure 1: System to be analysed (a), with given source model (b).

2.1. Source model

We assume the streaming sources to be simple ON-OFF sources, which are modeled as Markov Modulated Rate Processes. A source is then either on and sends data at its peak rate *h* or off where it doesn't send any data (as depicted in figure 1b). The time in the ON and OFF states is negative exponentially distributed with mean $1/\mu$ and $1/\lambda$, respectively.

The total rate into the buffer is than given by $h \cdot i$, where *i* is the number of active sources. The sources are independent and thus the probability of *i* active sources is given by

$$P(I = i) = b_i = \binom{n}{i} \cdot \alpha \cdot (1 - \alpha)^{n-i} \tag{1}$$

where *n* is the total number of sources and α is the probability that a source is active given by

$$\alpha = \frac{\lambda}{\lambda + \mu} \tag{2}$$

2.2. Fluid-flow model

To analyze the system sketched in figure 1 with the given source model, we first look at a model based on a continuous fluid-flow approximation of the traffic [4]. In such a model the data sent from the sources is assumed to be so fine grained that the data flow can be represented with a liquid. The model then represents sources that generate fluid which flow into and out of the buffer. If we denote the buffer content and number of active sources at time t as *X(t)* and *I(t)* respectively, the state of the system can here be described in terms of

$$P_i(t, x) = P(X(t) \le x, I(t) = i) \tag{3}$$

2.2.1 General model

We first look at the case without feedback from the congested buffer. The rate from each source is then independent of the queuesize and each source sends with it's peak rate h when in ON state. A statistical balance equation between the state probabilities at t and $t + \Delta t$ can be set up for $0 < X < m$ leading to [4]:

$$\frac{d}{dt}P_i(t, x) + \frac{d}{dx}P_i(t, x) \cdot (i \cdot h - c) = P_{i-1}(t, x) \cdot p(i-1, i) + P_{i+1}(t, x) \cdot p(i+1, i) - P_i(t, x) \cdot p^*(i) \tag{4}$$

where $p(i, j)$ is the probability of going from *i* to *j* active sources and $p^*(i)$ is given by:

$$p^*(i) = p(i, i+1) + p(i, i-1) = (n-i) \cdot \lambda + i \cdot \mu \tag{5}$$

We are interested in the time independent equilibrium probabilities

$$F_i(x) = P(X \le x, I = i) = \lim_{t \to \infty} P_i(t, x) \tag{6}$$

obtained by setting $\frac{d}{dt}P_i(t, x) = 0$.
Then the system is described by the following system of differential equations:

$$(i \cdot h - c) \cdot \frac{d}{dx} F_i(x) = F_{i-1}(x) \cdot p(i-1, i) + F_{i+1}(x) \cdot p(i+1, i) - F_i(x) \cdot p^*(i) \quad (7)$$

$, 0 \le i \le n, 0 < x < m$

These equations can be rewritten in matrix notation

$$\underset{\sim}{D} \frac{d}{dx} \underset{\sim}{F}(x) = \underset{\sim}{M} \underset{\sim}{F}(x) \quad (8)$$

Let φ_i and z_i denote the eigenvectors and eigenvalues of $\underset{\sim}{D}^{-1} \underset{\sim}{M}$. The solution to (7) is then given by:

$$\underset{\sim}{F}(x) = \sum_k a_k \cdot e^{z_k \cdot x} \cdot \underset{\sim}{\varphi}_k \quad (9)$$

The constants a_k are determined by looking at the boundary conditions:

- For all states where $i \cdot h > c$ the queue is always increasing, so the queuelength cannot be zero $\Rightarrow F_i(0) = 0$.
- For all states where $i \cdot h < c$ the queue is always decreasing, so the queuelength cannot be on its limit $\Rightarrow F_i(m) = P(I = i)$.

2.2.2 Controlled rate

Now we look at a system where the sources' rates are dependent on the content in the buffer. The way the node controls the rate can take different forms. One possible way is to divide the buffer into different levels, and let the rate of a source change from level to level. The allowed rate from a source in the ON state, is then given by $h \cdot k(x)$, where *k(x)* is the rate reduction as a function of the queuesize. An example of this is shown in figure 2a.

Such a system can be solved as for the general model, but we have to look at each level separately. For each level we then get a system of differential equations describing the system, in the same form as in (7), with h replaced by $r(x) = h \cdot k(x)$. If we look at level j, the solution is found as in the general model by

$$\underset{\sim}{F}_j(x) = \sum_k a_{k,j} \cdot e^{z_{k,j} \cdot x} \cdot \underset{\sim}{\varphi}_{k,j} \quad (10)$$

which is only valid in the interval where *k(x)* is constant. As an example, the solution to a system with a rate reduction as shown in figure 2a is given by:

$$\underset{\sim}{F}(x) = \begin{cases} \underset{\sim}{F}_1(x) & , 0 \le x < q_1 \\ \underset{\sim}{F}_2(x) & , q_1 \le x < q_2 \\ \underset{\sim}{F}_3(x) & , q_2 \le x < m \end{cases} \quad (11)$$

For each level we have to determine the constants a_k. If we have j_{max} levels, we have to determine $j_{max} \cdot n$ constants. These are found by looking at the following conditions:

- The state probabilities are continuous and we then have:

$$F_{i,j}(q_j) = F_{i,j+1}(q_j) \qquad , j = 1, 2, \ldots j_{max} - 1 \tag{12}$$

- In level 1: for all states where $i \cdot r(x) > c$ the queue is always increasing, so the queuelength cannot be zero $\Rightarrow F_{i,1}(0) = 0$.
- In the last level: for all states where $i \cdot r(x) < c$ the queue is always decreasing, so the queuelength cannot be on its limit $\Rightarrow F_{i,j_{max}}(m) = P(I = i)$.

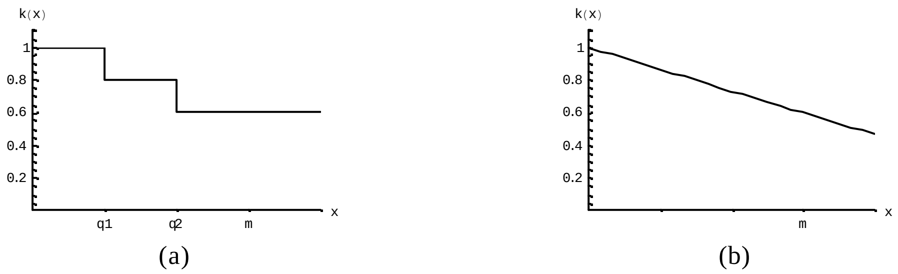


Figure 2: Step wise (a) and linear (b) rate reduction as a function of the queuesize.

Another approach to control the source's rate is to have a continuous rate reduction function based on the content in the buffer. This is similar to the technique used in RED [5] (fluid-based analysis in [6]), where the rate is reduced by dropping packets or setting a bit in packet headers with an increasing probability as the queuesize grows.

One example of such a continuous rate reduction function is shown in figure 2b. For this case we have looked at a system with one source. From (7) we then get the following system of differential equations:

$$-c \cdot \frac{d}{dx}F_0(x) = \mu \cdot F_1(x) - \lambda \cdot F_0(x) \tag{13}$$

$$(r(x) - c) \cdot \frac{d}{dx}F_1(x) = \lambda \cdot F_0(x) - \mu \cdot F_1(x) \tag{14}$$

Solving (13) for $F_1(x)$ and substituting into (14), we get:

$$(r(x) - c) \cdot \left(\frac{\lambda}{\mu} \cdot \frac{d}{dx}F_0(x) - \frac{c}{\mu} \cdot \frac{d^2}{dx^2}F_0(x)\right) = c \cdot \frac{d}{dx}F_0(x) \tag{15}$$

$$\Rightarrow \frac{\frac{d^2}{dx^2}F_0(x)}{\frac{d}{dx}F_0(x)} = \frac{\lambda}{c} - \frac{\mu}{r(x) - c} \tag{16}$$

Solving (16) for $F_0(x)$ we then get:

$$F_0(x) = \int k_1 \cdot e^{-g(x) + \frac{\lambda}{c} \cdot x} dx + k_2 \qquad , g(x) = \frac{\mu}{r(x) - c} \tag{17}$$

The constants k_1 and k_2 can be determined using the same boundary conditions as above. We have here only looked at a system with one source. A system with several sources is not easy to solve.

The feedback signals from a congested node are typically delayed some time before the sources get the information and are able to react to this information. This feedback delay is hard to take into account in the fluid-flow analysis presented here. We therefore wanted to look at a similar model, but with discrete time. Based on the exact solutions found by the fluid-flow analysis, we can study the accuracy of this discrete model.

2.3. Discrete Markov Chain

To be able to take this delay into account we look at a different model of the system based on discrete Markov chains. The delay in the control loop is for simplification lumped, see figure 1. In this model we look at the state of the system at times Δt apart, where the system state is given by the number of active sources and the queuesize, denoted by $\underset{\sim}{\pi} = \{I, X\}$. The idea here is to find the transition probabilities between all such states in Δt, and find the steady state probabilities by iteration.

By doing this discretization, we have to modify our source model slightly. The sources will now go from ON to OFF state in Δt with a probability *q*, and from off to on with a probability *p*. The time in each state is thus geometrically distributed, and *q* and *p* are given by

$$\begin{aligned} p &= \lambda \cdot \Delta t \\ q &= \mu \cdot \Delta t \end{aligned} \tag{18}$$

The probability, $P_{i,l}$, of going from *i* to *l* active sources in Δt is now found (details not shown due to space limitations):

$$\begin{aligned} P_{i,l} &= \sum_{j=0}^{i} \sum_{k=0}^{n-i} \binom{i}{j} \cdot \binom{n-i}{n-j-l} \cdot q^j \cdot (1-q)^{i-j} \cdot (1-p)^{n-j-l} \cdot p^{j+l-i} \\ &= \sum_{j=0}^{n-l} \binom{i}{j} \cdot \binom{n-i}{n-j-l} \cdot q^j \cdot (1-q)^{i-j} \cdot (1-p)^{n-j-l} \cdot p^{j+l-i} \end{aligned} \tag{19}$$

We know the probability of going from *i* to *l* active sources in Δt, but we also must look at how the queuesize develops. The total amount of data arriving the node in $[t, t + \Delta t]$ is given by $i \cdot h \cdot \Delta t$ and the amount of data departing from the node in the same interval is given by $c \cdot \Delta t$. To be able to solve such a system we have to choose *h*, *c* and Δt such that $(i \cdot h \cdot -c) \cdot \Delta t$ is an integer for all *i*.

In the case where each source has to regulate it's rate based on feedback from the congested node, the rate is dependent on the content in the buffer at time $t - d$, where *d* is the delay. The total amount of data arriving the node in $[t, t + \Delta t]$ is then given by $i \cdot r(x_{t-d}) \cdot \Delta t$. Here, we have to choose *r(x)*, *c* and Δt such that $(i \cdot r(x) - c) \cdot \Delta t$ is an integer for all combinations of *i* and *x*.

2.3.1 Solution without feedback delay

If we don't have any feedback delay, we can find the content in the buffer at time $t + \Delta t$ if we know the content in the buffer and number of active sources at time t by $(i \cdot r(x_t) - c) \cdot \Delta t$. The transition probabilities is thus given by

$$\underset{\sim}{P}_{\{i, x\}\{l, w\}} = \begin{cases} P_{i,l} & , (i \cdot r(x_t) - c) \cdot \Delta t = w - x \\ 0 & , (i \cdot r(x_t) - c) \cdot \Delta t \neq w - x \end{cases} \tag{20}$$

If we denote the system state at time $n \cdot \Delta t$ by $\underset{\sim}{\pi}^{(n)}$, the system state at time $(n+1) \cdot \Delta t$ is given by

$$\underset{\sim}{\pi}^{(n+1)} = \underset{\sim}{\pi}^{(n)} \underset{\sim}{P} \tag{21}$$

We can now find the steady state probabilities by iteration using (21). This iteration can be done until some precision criteria are fulfilled.

2.3.2 Solution with feedback delay

When $d > 0$, the amount of data arriving the node in $[t, t + \Delta t]$ will be dependent on the number of active sources at time t and the buffer content at time $t - d$. That means that the transition probabilities is not the same all the way as in (20), because the amount of data arriving the node in $[t, t + \Delta t]$ is here dependent on the state of the system at time $t - d$ and at time t. To be able to solve such a system we then have to require that $d = a \cdot \Delta t$, where a is an integer.

If we look at a system with a step wise rate reduction function with two levels, the sources send with max rate or reduced rate at time t based on the buffer content at time $t - d$. For each iteration we then need to use the state probabilities a step back, and update the transition probabilities. The probability for the sources to send with high rate or reduced rate is then given by:

$$\begin{aligned} &P(x_{t-d} < q_1) \\ &P(x_{t-d} \geq q_1) \end{aligned} \tag{22}$$

To find the steady state probabilities for a system with feedback delay from the node to the sources, we then need to save the state probabilities a steps back. In addition, the iteration can be time consuming because of the fact that the transition probabilities have to be updated for every iteration.

3. Results

Based on the analysis carried out in the last chapter, we can now find the distribution of the queuesize and the probability of overflow in the buffer.

Table 1: Parameters used in analysis.

Parameter	Value	Description
c	30 Mb/s	Node’s capacity
h	60 Mb/s	Source’s peak rate
$1/\lambda$	5.55 ms	Expected time in OFF state
$1/\mu$	5.13 ms	Expected time in ON state
m	300 kb/10 ms	Buffer size
q1, q2	100 kb, 200 kb	Set point where rate is changed in case of step wise rate reduction function
A	0.96	Offered traffic

3.1. Fluid-flow

Based on the results for $\underset{\sim}{F}(x)$ in (9), (10), and (17), we now find the queuesize distribution:

$$P(X \le x) = \sum_{i=0}^{n} F_i(x) \tag{23}$$

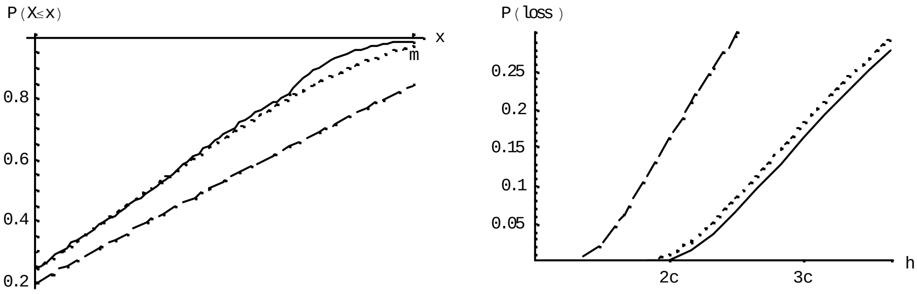
From (23) we can also find the probability of overflow in the buffer by the amount of data arriving when it is full:

$$P(loss) = \frac{\sum_{i=0}^{n} (i \cdot (k(m) \cdot h - c))^{+} \cdot (b_i - F_i(m))}{\sum_{i=0}^{n} i \cdot k(m) \cdot h \cdot b_i} \tag{24}$$

where $(x)^{+}$ is defined as $max(x, 0)$.

We have compared the queuesize distribution for a case without rate control, with the distribution for the two different rate reduction functions shown in figure 2. The analysis was carried out with traffic from one source, and the parameters as shown in table 1. We have also looked at the probability of buffer overflow as a function of the source’s peak rate for these three cases.

By having a continuous rate function, the congested node has to send feedback information back to the sources all the time, leading to very much signaling. This is thus unrealistic to implement, but is interesting as a reference. Such a continuous rate function can be approximated by a step wise function as shown in figure 3a, with a few rate levels, leading to less signaling. The resulting queuesize distributions and loss probabilities are shown in figure 3.



(a) (b)

Figure 3: Queuesize distribution (a) and loss probability (b) with no rate control (dashed), with step wise rate reduction function as shown in figure 2a (solid) and with linear rate reduction function as shown in figure 2b (dotted).

In fact, it is possible to match the loss probabilities for such a continuous rate reduction function with only two levels. The loss probabilities for this step wise function and the continuous function are compared in figure 4. By using such a step wise rate reduction function the node only has to signal to the sources when the queuesize cross the point where the rate is changed. And the sources only need to adapt to two rate levels, which can be realized by a base layer and an enhancement layer in the case of scalable encoding. The rate is here reduced by cutting off the enhancement layer.

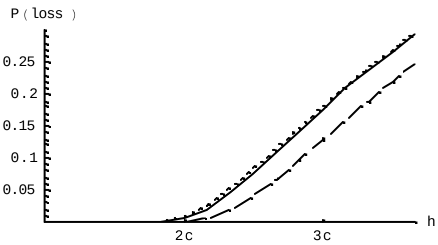


Figure 4: Loss probability with linear rate reduction function (dotted), compared with step wise rate reduction with two levels, where the step is at 100 kb (dashed) and 200 kb (solid).

Given which two rate levels the sources can adapt to, and the requirements on the loss probability, one have to choose an appropriate value of the queuesize where the rate is reduced (q1). With the parameters as listed in table 1, we have looked at how the loss probability is affected by the value of q1 (figure 5). From this curve the value of q1 can be found when the acceptable loss level is given. This makes it possible to give a statistical guarantee for the loss in the node.

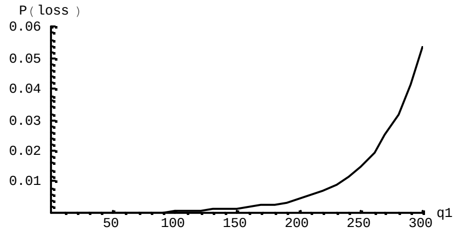


Figure 5: Loss probability as a function of q1 (set point where rate is changed).

3.2. Discrete Markov chain

Based on the state probabilities found by using discrete Markov chains and the iterative formula in (21), we now can find the queuesize distribution:

$$P(X = x) = \sum_{i=0}^{n} \pi_{i,x} \qquad (25)$$

$$P(X \le x) = \sum_{k=0}^{x} \sum_{i=0}^{n} \pi_{i,k}$$

The probability of overflow can be found similarly as in the fluid-flow analysis by (27).

To see how good approximation is possible by doing this discretization, we have compared the results for the queuesize distribution for the case of a step wise rate reduction function with two levels (step at 150 kb, the rest of the parameters as in table 1), with the results found by the fluid flow analysis. We have here looked at the error made in the queuesize distribution for different values of Δt. The largest error in percent of the exact solution based on the fluid flow analysis for different values of Δt is shown in table 2. This error tells us the time resolution needed in the DMC approach.

Table 2: Error made in the approach based on DMC in percent of the exact solution.

Δt (ms)	Max error (%)
1	15.28
1/2	8.66
1/3	5.81
1/6	2.54

Based on this discrete model we now looked at how the feedback delay affects these rate control policies. We considered the same simple system with two rate levels (parameters as above, Δt equal to 1/6), and compared the queuesize distribution and probability of buffer overflow for different feedback delays (figure 6). We also found the limiting loss probability. Here, the sources will send with high or low rate with probability given by $P(x < q_1)$ and $P(x \ge q_1)$ respectively. These probabilities and thus the probability of buffer overflow is found by iteration. We see that the loss probability is close to this limit for large values of d.

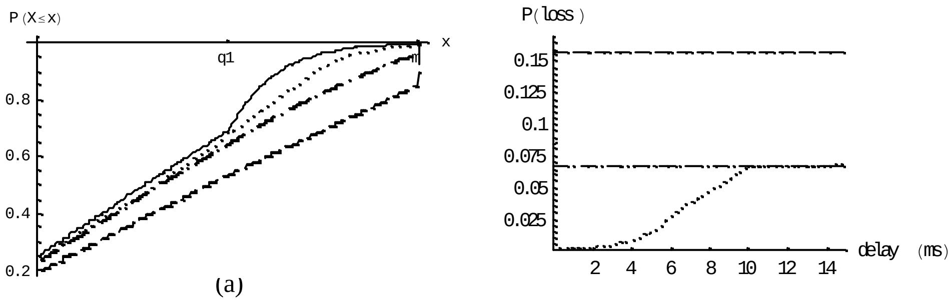


Figure 6: (a): Queuesize distribution for system with feedback delay (2ms: dotted, 5ms: dotted-dashed). (b): Loss probability for varying feedback delay (dotted) and with infinite delay (dotted-dashed). Both compared with loss probability for the system without delay (solid) and without feedback at all (dashed).

As expected the probability of buffer overflow increases with the delay, since it takes an increasing amount of time until the sources react to the feedback signals. This will therefore have to be taken into account when choosing an appropriate value of q1 (set point where rate is changed). In figure 7 it is shown how the probability of buffer overflow is affected by q1 for different feedback delays.

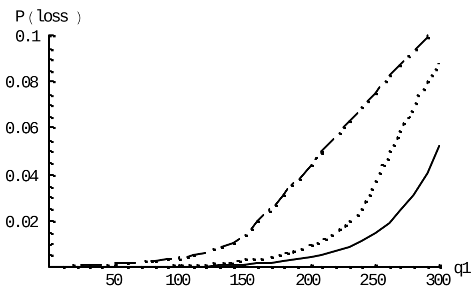


Figure 7: Loss probability as function of q1 for system with feedback delay (2ms: dotted, 5ms: dotted-dashed) compared with system without delay (solid).

4. Conclusions

The paper presents two different approaches to analyze a rate control policy based on feedback from a congested node. We have made use of a fluid-flow analysis to find an exact solution when simple ON-OFF sources generate traffic into the node, and looked at a similar model based on DMC to study how the feedback delay affects the efficiency of this control scheme. We have here only looked at systems with one source, but analysis with several sources is being done now. The DMC approach is simple, but gives many states which may require much memory and lead to time consuming computations.

By using such a rate control policy based on feedback from the network, the results presented here show that the loss probability is reduced and one can achieve better control of the output rate from the sources which apply such a policy. Such a policy can take different forms based on the rate reduction function applied in the node. For this to be feasible one would require little signaling, and thus a function with different rate levels could be the way to go. We have here shown that such a function can be used to approximate other continuous rate reduction functions, and give similar queuesize distributions and loss probabilities.

Another implementation issue is the fact that there usually is a delay before the sources get the feedback information and are able to react to this. As expected we found that this feedback delay will affect the loss probability. With a given acceptably loss level this delay therefore have to be taking into account when choosing an appropriate rate reduction function. Such a rate control policy would therefore only be feasible if the delay in the feedback path is low enough, and the sources are able to react to the feedback information fast.

By using such a rate control policy the sources have to reduce their rates', and thus the quality of the application is degraded. We have here only looked at how such a policy will affect the quality of the application in terms of loss in a congested node, and not at the perceived quality at the receiving end. The quality will be degraded both with and without such a control policy, but by applying this policy one have better control and congestion can be reduced considerably.

In our analysis we have assumed that the parameters describing the behavior of the sources are known. In a real case, some of these parameters may be unknown and need to be estimated. This will be for later work. It could also be useful to do a sensitivity analysis and find how variations in the parameters would affect on the results found for this control scheme.

References

[1] P. Gevros, J. Crowcroft, P. Kirstein, and S. Bhatti. Congestion control mechanisms and the best effort service model. *IEEE Network*, 15(3):16-25, 2001

[2] D. Wu, Y. Hou, and Y-Q. Zhang. Transporting real-time video over the internet: Challenges and approaches. *PROCEEDINGS OF THE IEEE*, 88(12), December 2000

[3] D. Wu, Y.T. Hou, W. Zhu, Y-Q. Zhang. Streaming video over the internet: Approaches and directions. *IEEE Transactions on circuits and systems for video technology,* 11(3), March 2001

[4] R.C.F Tucker. Accurate method for analysis of a packet-speech multiplexer with limited delay. *IEEE Transactions on Communications*, 36(4), April 1988

[5] S. Floyd, V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on networking*, 1(4), August 1993

[6] V. Misra, W-B. Gong, D. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. *ACM SIGCOMM*, 30(4), October 2000