The Role of Information Update in Flow Control

Eitan Altman [a], Tamer Başar [b] and Naceur Malouch [c]

Abstract

Congestion control protocols have been used both in the ATM architecture as well as in the Internet world allowing one to improve throughputs, decrease delays and avoid oscillations and instabilities. A basic common element in such protocols are information packets that are used to signal congestion. The goal of this paper is to study the question of how frequently congestion control protocols need to generate such packets in order to optimize their performance. Through a number of congestion control models, we identify and quantify different types of effects of the frequency of generating information packets. We consider both TCP-type protocols, in which controlling the frequency of information packets is done through static or dynamic delayed ACK options, as well as ATM type flow control, where the optimal time spacing between the generation of network management packets is computed. We show how the spacing between information packets influences the throughput and the stability of the system.

I. Introduction

Congestion and flow control protocols make use of special control packets to indicate to the traffic sources actions to be taken in order to adapt the transmission rates to the (possibly dynamically) available bandwidth. In TCP/IP these are the acknowledgement packets; in the ABR (Available Bit Rate) class of ATM these are the resource management packets. Even when the size of these packets is significantly smaller than that of the data packets, they may still compete over network resources with the data packets and thus decrease the amount of resources available to them. In many cases, they may require an amount of resource much larger than what their size would suggest. For example, when using the IEEE 802.11 MAC protocol, each data as well as ACK packet of the TCP flow requires the same (large) overhead of three link layer packets (RTS, CTS and a link layer ACK). Even in the absence of such overheads, the processing time of an ACK at the destination might require an additional overhead that can be pretty large with respect to the transmission time when very high speed networks are considered. We therefore raise the question of at what frequency flow control protocols should send control packets. The aim of this paper is to answer this question using mathematical tools.

We should note that TCP/IP already has the ”delayed ACK” option that allows it to reduce the ACK frequency from one ACK for every received packet to one ACK every $d = 2$ received packets. For Ad-Hoc networks using the IEEE 802.11 MAC protocol, it has been established through simulations that further improvement can be obtained when using TCP with $d > 2$ [6]. Note that ACKs could also be filtered within the network (see e.g. [8] and references therein). However we shall not investigate here the question of how ACKs should be thinned.[1]

[a]Postal address: INRIA, 2004 Route des Lucioles B.P. 93, 06902 Sophia Antipolis, France
[b]Postal address: University of Illinois, 1308 West Main Street, Urbana, IL 61801-2307, USA
[c]Postal address: LIP6, Université Pierre et Marie Curie 8, rue du capitaine Scott 75015 Paris - FRANCE

[1]Although we shall study through simulations the delayed ACK mechanism in which the destination node has the responsibility of generating “fewer” ACKs, the advantage of thinning ACKs at the destination using the delayed ACK option is that if d packets have not yet arrived at the

We introduce three frameworks in which to study the optimization of the frequency of control packets. The first is the case of TCP/IP traffic sources with routers using drop tail queues. A simple mathematical model is derived for optimizing the amount of ACK thinning at the destination so as to maximize the system's throughput. A simulation study validates the conclusions we obtained from the mathematical model. In the second framework, we study the dynamics of an AIMD (additive increase multiplicative decrease) flow control interacting with a RED type buffer. We model the system's dynamics through a system of delay-differential equations, and study the stability of the system as a function of the frequency of ACKs. We finally propose an abstract model for an optimal rate control with sampled delay information, similar to models used for rate control in the ABR class in ATM [2]. A linear quadratic model is considered and we optimize the time between the sending of information feedback.

II. A Fixed Point Method to Model ACK thinning

In this section, we use the expression for TCP throughput developed in the literature and standard queueing models to quantify the impact of the delay factor d on the throughput. The throughput T of a TCP connection can be approximated by [17]

$$\frac{1}{RTT\sqrt{\frac{2\,d\,p}{3}} + RTO\,\min\left(1,\,3\sqrt{\frac{3\,d\,p}{8}}\right)p\left(1+32p^2\right)} \tag{1}$$

where p is the loss probability of TCP packets, RTT is the round-trip delay experienced by the TCP connection, and RTO is the retransmission timeout.

The above formula was established under the assumption that the TCP sender increases its congestion window for each arriving ACK. It is worthwhile to notice that the TCP sender can rather increase its congestion window using the number of previously unacknowledged bytes each ACK covers [20], [21]. In this case, the factor d disappears from the formula, and the throughput increases. Since in this paper our goal is to find the optimal value of d that *maximizes* the total throughput of the TCP connections in the network, we keep the factor d in equation (1). However, our study could also serve as a worst-case analysis for the other scenario.

We start by modeling the network as a bidirectional link. Each direction of the link is modeled by a queue system. Two sets of N symmetric TCP sources send data from both end-points of the link (Figure 1). Each source of the first set connects to a receiver that does not belong to the second set and vice versa. Thus, in each queue TCP packets and ACK packets from different connections are multiplexed in the same queue and served by the same server. Using this model, we assume that losses that occur in the system are only due to congestion, i.e. buffer overflow.

Denote by α the "effective" size of ACK packets and by Z the size of TCP packets. The parameter α would not only model the actual size of ACK packets but also the eventual overheads introduced in the processing time. Denote by $b(t)$ the service time distribution, which can be expressed as

$$b(t) = \begin{cases} \dfrac{\alpha}{Z\,C} & \text{if an ACK packet is in service at time } t \\[2ex] \dfrac{1}{C} & \text{if a TCP packet is in service at time } t \end{cases} \tag{2}$$

destination but some timer has expired, the destination will generate an ACK thus avoiding situations in which the source will interpret the lack of ACKs as a loss of a packet detected through the source's timeout expiration.

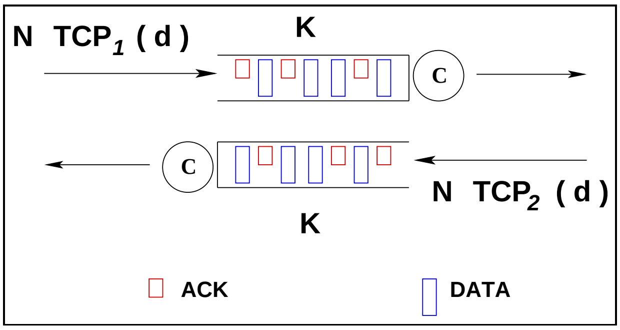

Fig. 1. A network model for ACK thinning

where C is the capacity of the link in TCP packets per unit of time. In order to develop a tractable model, we need a simple formula that relates the throughput at the buffers to the losses that will be experienced there. To that end we shall assume that the packet arrival process at each queue can be approximated by a Poisson process. (We note that the validity of this approximation in a similar context has been discussed and examined in [1], [7].) Then, the packet loss probability is the loss probability of an $M/G/1/K$ system. Note that the loss probability seen by ACK packets or TCP packets is theoretically the same because of the PASTA property.

Below we propose two modeling approaches for the service time: the exponential service time which provides a simple expression for the losses but gives a rough approximation, and the deterministic service time (whose duration varies according to whether it is an ACK or a data packet) which gives a better approximation but with a more complex expression for the losses. In the deterministic model there are thus two possible values of service times: α/ZC or $1/C$.

A. The Exponential Service Time Case

In this case, we assume that the service time distribution of packets is exponentially distributed. Using the exponential assumption, the loss probability is then given by the loss probability of an $M/M/1/K$ system:

$$p = \rho^K \frac{1-\rho}{1-\rho^{K+1}} \tag{3}$$

where ρ is the load of the system and it is computed as follows:

$$\begin{aligned}\rho &= \left(NT + \frac{NT}{d}\right)\left(\frac{1}{d+1}\frac{\alpha}{ZC} + \frac{d}{d+1}\frac{Z}{ZC}\right) \\ &= \frac{1}{C}(NT + NT\frac{\alpha}{dZ}) \end{aligned} \tag{4}$$

We use the fixed point method to solve numerically the system of equations (1), (3) and (4). The advantage of this simple model is that we can compute the loss probability and thus the throughput for large values of K and ρ.

B. The Deterministic Service Time Case

Here, we change only the assumption that the service times are exponentially distributed and thus only equation (3) is replaced by the expression of the loss probability of an $M/G/1/K$ queue [18], [19].

$$p = \frac{1 + (\rho - 1)\, f}{1 + \rho\, f}, \text{ where } f = \frac{1}{2\pi\, i} \oint_{D_r} \frac{1}{G(s)\, s^{K-1}}\, \mathrm{d}s \tag{5}$$

D_r is any circle in the complex plane with center 0 and with radius r chosen small enough so that all the zeros of the function $G(s)$ are outside the circle, i.e. $r < |G(z)|\ \forall\ z$ such that $G(z) = 0$.

The complex function $G(s)$ is defined as $G(s) = LST(b(\lambda(1 - s))) - s$ where $LST(b())$ is the Laplace Stieltjes Transform of the service time distribution:

$$\begin{aligned} LST(b(s)) &= \int_0^\infty b(t)\, e^{-st}\, dt \\ &= \underbrace{\frac{1}{d+1} e^{-\frac{\alpha s}{ZC}}}_{ACK} + \underbrace{\frac{d}{d+1} e^{-s/C}}_{TCP} \end{aligned} \tag{6}$$

The parameter λ is the total arriving rate at the entrance of each queue, which is equal to $(NT + NT/d)$. Computation of f is detailed in the Appendix.

Again, we solve the three equations (1), (5) and (4) numerically, and we compare the results to those obtained using the exponential time distribution.

C. Numerical Results and Simulations

In this section we use our model to study numerically the trade-off controlled by the delayed factor d. We will show that a relative gain in the throughput, ranging from 5% to 50%, can be achieved by setting $d > 1$. We also perform simulations using NS-2 to evaluate the accuracy of the two models at capturing the impact of d on the throughput.

Figure 2 plots the throughput of TCP as function of the delayed factor d. We consider an "effective" size of TCP data packet of 500 bytes and various "effective" sizes of ACKs, ranging between $\alpha = 40$ to 250. (as mentioned in the introduction, the difference between actual and effective size of a packet is that additional overhead may be added to its real size due to other protocols of other layers; in addition, processing an ACK at some nodes may take longer than its relative size with respect to a TCP data packet). The queue size is 20 packets and the round-trip time RTT is 200 ms. We set C to 125 TCP packets/s, and we set N to 1 since the throughput depends only on the fraction C/N. The figure was obtained numerically using the more precise model of deterministic service times.

The figure shows that for the small ACK size 40, spacing the ACK (using $d > 1$) results in a small improvement of the performance (6%). However, the throughput is maximized when $d = 4$. For higher values of "effective" ACK size ($\alpha \geq 100$, we see an improvement of 12% to 31%, with the optimum obtained for example at $d = 5$ for $\alpha = 100$. Using a value of $d > 2$, which is the default value in TCP, results in a relative gain of around 10%. We have observed similar trends with larger link capacities C (larger congestion window) and with much smaller buffer sizes K (higher loss probabilities). When $K \geq 60$, the throughput approaches the value obtained simply by solving $\rho = 1$, which gives $Thpt = \frac{C}{N(1+\alpha/(dZ))}$, and the relative gain can reach 50%.

Figure 3 plots the throughput vs. d with the same parameters as the previous scenario, but using the exponential service time approximation. In this case, the packet size can represent the average of the "effective" packet size in the

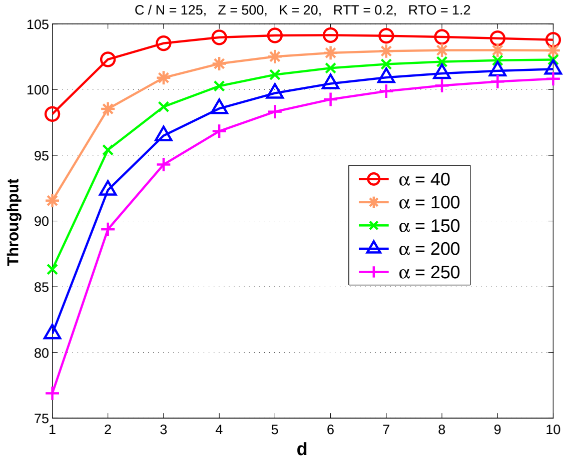

Fig. 2. Effect of d on the throughput for various ACK size

network. This model is useful when the packet size in the backbone is variable and the exact distribution is unknown. The Figure shows that the trend of the throughput is almost the same, except for the smallest ACK size 40 where spacing the ACK results in deterioration of the performance for all values of $d > 1$. For larger ACK size $\alpha \geq 200$, once more, the gain of delaying ACKs goes beyond 20%.

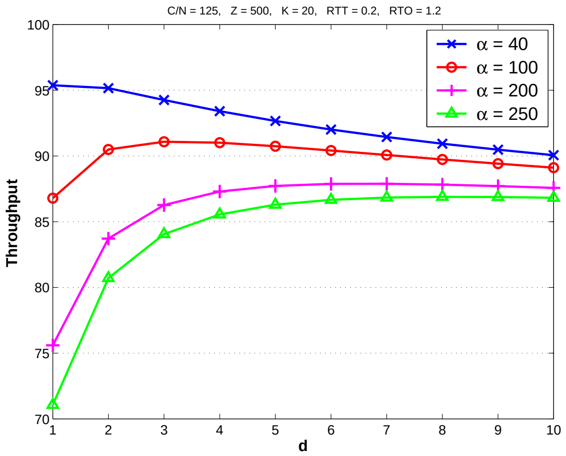

Fig. 3. Effect of d on the throughput for various ACK size, using the exponential service time approximation

Next, we perform two sets of simulations to study the robustness of the models described herein. First, we study the effect of the number of connections N. Then, the effect of the receiver timeout[2] which is not considered by the two models. We use a similar network configuration as the one described in the model (Figure 1), and we add N access links of capacity 1 M each to the bottleneck link. The average round-trip propagation delay ($\approx RTT$) is set to 200 ms.

In the first set of simulations, we vary the number of connections N from 30 to 200, and we vary the bottleneck capacity C in order to keep constant the ratio C/N at 125. We fix the ACK size to $\alpha = 100$. In this set of simulations, we use the default value of the receiver timeout which is set to 100 ms. Figure 4 plots the average TCP throughput vs. the delayed factor d, using the two models and simulation traces for $N = 30$, 50, 100 and 200.

[2]The receiver timeout is used to send an ACK for arriving packets even before d packets arrived at the receiver if the time since the non-acknowledged packet arrived exceeds the timeout

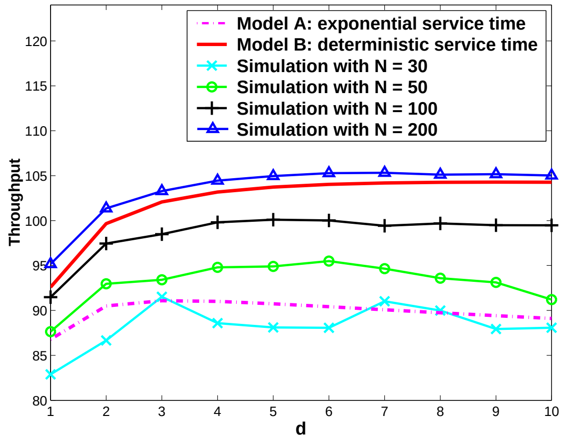

Fig. 4. Simulation results vs. numerical results

The main observation we made is that when N increases, the relative error induced by the deterministic model decreases notably. This is because the resulting process of multiplexing the TCP connections approaches a Poisson process [7]. Surprisingly, the model with the exponential service time approximation predicts very well the throughput when $N = 30$. The combination of two phenomena is responsible for that.

- First, we note that whereas the queueing models yield the same throughput when multiplying both the number of connections and the capacity by the same constant, the simulation does not: when the number of connections in the simulation is small, the capacity can not be fully utilized and the throughput of single connections decreases. Another reason for the fact that the throughput of the simulation is lower for smaller N is that there is less multiplexing so the packet arrival process is more bursty than the Poisson model.
- Second, the exponential service model gives lower throughput than the deterministic service one (this is due to the fact that the exponential model (M/M/1/K) over-estimates the loss probability with respect to the one computed by the deterministic model (M/D+D/1/K)). Combining this with the fact that the simulated throughput is lower than the deterministic model can explain the better fit of the simulation with the exponential model for low N.

If we examine closely the plots, in Fig. 4, we can observe that when d is large, the throughput in the simulations decreases (as opposed to the one estimated by the deterministic model). The explanation is that when d is large, the receiver timeout expires more often especially when the congestion window size is small. Thus, the number of ACKs increases and the bandwidth consumed by ACKs becomes greater than NT/d. Figure 5 illustrates this behavior when $N = 100$, which depicts the ratio between the TCP and the ACK throughputs. This ratio is seen to increase when the timeout value increases.

After examining the influence of the timeout on the ratio of throughputs, we wish to examine its impact directly on the TCP throughput. We use the same parameters as in the previous simulations, and we set N equal to 100. We vary the receiver timeout from $t_1 = 10$ ms to $t_2 = 180$ ms. Note that t_1 is just larger than the minimum inter-arrival time of two consecutive TCP packets sent in the same window, and t_2 is just less than RTT.

Figure 6 plots the average TCP throughput as a function of the receiver timeout for various values of d. We see clearly that when the timeout is large, then the throughput is reduced. In the case $d = 2$ and $d = 4$, the throughput is maximized when the timer is equal to 20 ms. For $d = 6$ and $d = 8$, the throughput is maximized at a timer of 50 ms.

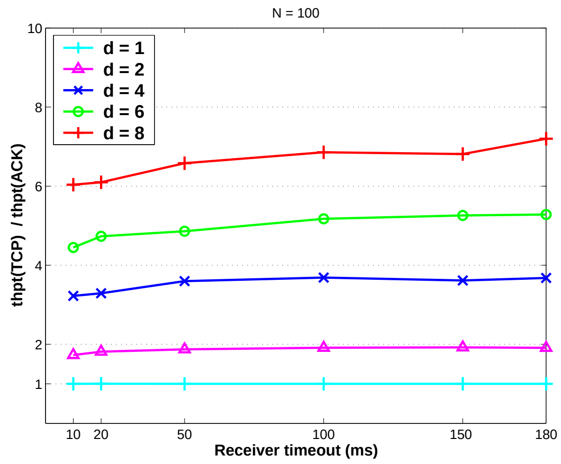

Fig. 5. Effect of the receiver timeout on the number of generated ACKs

Here, the gain in the throughput is relatively small ($\approx 3\%$), but this is because the average window size is also small (≈ 20). For larger window sizes, for example when the delay-bandwidth product is large, the gain is more significant. However, setting a small value for the timeout is risky since the packet inter-arrival time depends on the cross traffic along the path and hence could vary over time.

More generally, it is difficult to find a constant timeout that is adequate for all network scenarios. Another alternative is to compute dynamically the timeout using a similar method as the one used by the sender to compute the retransmission timeout.

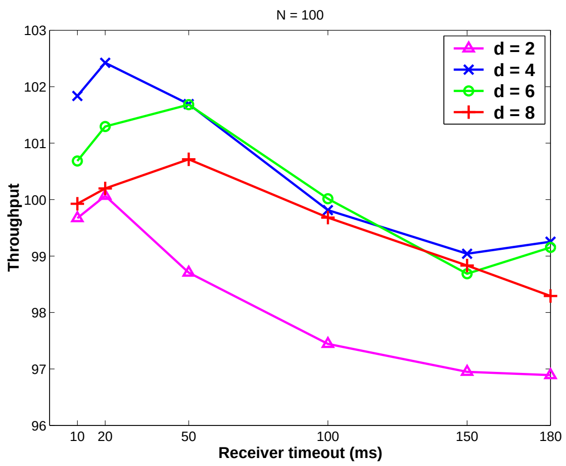

Fig. 6. Effect of the receiver timeout on the TCP throughput

To conclude this section, it is worthy to use a delay factor $d > 2$, particularly for long TCP connections and when the window size is large. In fact, the improvement obtained from reducing the number of ACKs in the network is more significant than the decrease in the throughput due to the lack of information update. Besides, by choosing an adequate receiver timeout, we can improve further the gain in the throughput.

III. A DYNAMIC CONTROL MODEL OF TCP TRAVERSING A RED BUFFER

Whereas in the previous section we focused on the quantitative impact of the rate of information on the throughput, we study in this section a more qualitative property, the stability, and show how stability conditions are influenced by the rate of information packets.

We assume that N symmetric persistent TCP connections share a bottleneck link of capacity C located close to the source, assumed to operate in the congestion avoidance regime. Let $W(t)$ be the window size of a connection at time t, R the round trip delay (including queueing delay), which is assumed to be a constant (this assumption holds when queueing delays are much smaller than propagation delays). Let $p(t)$ be the loss (or marking) probably of the RED buffer at time t. Let $q(t)$ be the amount of buffered traffic at the queue at the bottleneck link.

A standard way to study the interaction between TCP/IP and the RED buffer is to consider a fluid approximation of the window size of TCP. The window size evolution is then given by

$$\frac{\mathrm{d}W}{\mathrm{d}t} = \frac{1}{R} - \frac{W(t)W(t-R)}{2R}p(t-R),$$

see [13]. This equation is obtained under the assumptions that (i) the delayed ACK mechanism is not used; and (ii) a new ACK is generated with each TCP packet that arrives at the destination, resulting in an increase of the window size by one unit every round trip time.

We shall now consider the possibility of using a (dynamic) delayed ACK approach in which an ACK is generated for every d TCP packets that arrive at the destination. d will be considered as a control variable and will thus be allowed to be time dependent. Both the increase rate as well as the decrease rate are divided by a factor d since the rate of ACKs that arrive at the sources is d time smaller; In particular indications for decreasing the rate (we assume that ACKs have marks indicating congestion) return less frequently. The window size evolution then becomes

$$\frac{\mathrm{d}W}{\mathrm{d}t} = \frac{1}{Rd(t-R)} - \frac{W(t)W(t-R)}{2Rd(t-R)}p(t-R).$$

The queue dynamics is given in [13] by

$$\frac{\mathrm{d}q}{\mathrm{d}t} = \frac{N}{R}W(t) - C.$$

We shall assume that not only the TCP packets have to queue but also the ACKs. To model the difference between the size of an ACK and that of a TCP packet, we assume that an ACK requires a fraction γ of buffer space required by a TCP packet. Under this assumption, the queue dynamics above is modified to

$$\frac{\mathrm{d}q}{\mathrm{d}t} = \frac{N}{R}W(t) - C + \frac{\gamma W(t-R)N}{d(t-R)R}.$$

The tradeoff that influences the choice of the control d will be the following: on one hand, when the window size is small, we may wish d to be small so that the window size can grow quickly so as to achieve higher throughput. On the other hand, when the window size is large then we may wish to increase d so as to limit the congestion due to ACKs in the bottleneck queue.

We shall analyze in this section a linear control mechanism in which d has the form

$$d(t) = \alpha(1 + \beta W(t)) \quad \alpha > 0,\ \beta \geq 0.$$

Finally, we shall consider the RED buffer marking probability (ignoring the averaging of the queue size) in its linear operation regime:

$$p(t) = \eta_1 q(t) - \eta_2.$$

We now summarize the system's overall dynamics below:

$$\frac{\mathrm{d}W}{\mathrm{d}t} = \frac{1}{Rd(t-R)} - \frac{W(t)W(t-R)}{2Rd(t-R)}p(t-R) \tag{7}$$

$$\frac{\mathrm{d}q}{\mathrm{d}t} = \frac{N}{R}W(t) - C + \frac{\gamma W(t-R)N}{d(t-R)R} \tag{8}$$

$$d(t) = \alpha(1 + \beta W(t)) \tag{9}$$

$$p(t) = \eta_1 q(t) - \eta_2 \tag{10}$$

Equilibrium: The equilibrium point is obtained by equating to 0 the time derivatives in the previous equations. We obtain

$$\frac{N}{R}W_o + \frac{N\gamma W_o}{d_o R} = C \Rightarrow W_o = \frac{CR}{N(1+\gamma/d_o)} \tag{11}$$

$$d_o = \alpha(1 + \beta W_o) \tag{12}$$

$$W_o^2 p_o = 2 \Rightarrow p_o = 2/W_o^2 \tag{13}$$

$$p_o = \eta_1 q_o - \eta_2 \Rightarrow q_o = (p_o + \eta_2)/\eta_1 \tag{14}$$

Hence the throughput is given by:

$$Thp = \frac{W_o}{R} = \frac{C}{N(1+\gamma/d_o)}$$

The throughput is seen to increase with d_o.

A linearization of the dynamical system in a neighborhood of the equilibrium point yields

$$\begin{aligned}\frac{\mathrm{d}\delta W}{\mathrm{d}t} &= -\frac{1}{RW_o d_o}\delta W(t) - \frac{1}{RW_o d_o}\delta W(t-R)\\ &\quad -\frac{\eta_1 W_o^2}{2Rd_o}\delta q(t-R)\\ \frac{\mathrm{d}\delta q}{\mathrm{d}t} &= \frac{N}{R}\delta W(t) + \frac{\alpha\gamma N}{d_o^2 R}\delta W(t-R),\end{aligned}$$

where δ stands for the shifted version of variables in which the equilibrium value is subtracted (e.g. $\delta W := W - W_o$). Taking the Laplace transform of these equations we obtain

$$s\delta W(s) = -\frac{1}{RW_o d_o}\delta W(s) - \frac{e^{-sR}}{RW_o d_o}\delta W(s)$$

$$
\begin{aligned}
& -\frac{\eta_1 W_o^2}{2Rd_o} e^{-sR}\delta q(s) \\
s\delta q(s) &= \left(\frac{N}{R} + \frac{\alpha\gamma N}{d_o^2 R} e^{-sR}\right)\delta W(s)
\end{aligned}
$$

With $z = sR$, the stability condition is then given by requiring that all zeros of $g(z) = 0$ have strictly negative real parts, where $g(z)$ is given by

$$
z^2 + \frac{z}{W_o d_o}(1 + e^{-z}) + \frac{\eta_1 W_o^2 N}{2d_o}(1 + \frac{\alpha\gamma}{d_o^2} e^{-z})e^{-z} \tag{15}
$$

A. Solution of $g(z) = 0$

Denote $x = \Re(z), y := \Im(z)$. Then $g(z) = 0$ is equivalent to

$$
\begin{aligned}
0 &= x^2 - y^2 + 2ixy + \frac{x + iy}{W_o d_o} \\
& +(\frac{x + iy}{W_o d_o} + \frac{\eta_1 W_o^2 N}{2d_o})e^{-x}(\cos(y) - i\sin(y)) \\
& +\frac{\eta_1 W_o^2 \alpha\gamma N}{2d_o^3} e^{-2x}(\cos(2y) - i\sin(2y))
\end{aligned}
$$

or equivalently,

$$
\begin{cases}
x^2 - y^2 + \frac{x}{w_o d_o} + \\
\quad (\frac{x}{W_o d_o} + \frac{\eta_1 W_o^2 N}{2d_o})e^{-x}\cos(y) + \frac{y}{W_o d_o} e^{-x}\sin(y) \\
\quad +\frac{\eta_1 W_o^2 \alpha\gamma N}{2d_o^3} e^{-2x}\cos(2y) = 0 \\
\\
2xy + \frac{y}{W_o d_o} - (\frac{x}{W_o d_o} + \frac{\eta_1 W_o^2 N}{2d_o})e^{-x}\sin(y) \\
\quad +\frac{y}{W_o d_o} e^{-x}\cos(y) \\
\quad -\frac{\eta_1 W_o^2 \alpha\gamma N}{2d_o^3} e^{-2x}\sin(2y) = 0
\end{cases}
$$

a) The goal.: One may identify two possible goals: (i) maximize the system throughput while keeping it stable, and (ii) for a given desired throughput, make the system "as stable as possible" by which we mean to choose the parameters so as to have the real part of the largest zero of $g(z)$ as negative as possible.

Note that whereas the throughput only depends on d_o, γ, N and C, and not directly on the values of $\alpha, \beta, \eta_1, \eta_2$, the stability regime does depend directly on α and η_1, and hence these parameters do enter into the optimization under goal (i). For the second goal, we may first compute d_o and then optimize stability with respect to the other parameters.

We need to (numerically) verify that there are no zeros of g with $x \geq 0$. We show that we can restrict the numerical search to a bounded domain.

Assume $x \geq 0$. Then

$$
|g(z)| \geq |z|^2 - \frac{2|z|}{w_o d_o} - \frac{\eta_1 w_o^2 N}{2d_o}(1 + \frac{\alpha\gamma}{d_o^2}) =: f(|z|)
$$

A necessary condition for $g(z) = 0$ is then $f(|z|) \leq 0$. Note that

$$f(|z|) = (|z| - z_1)(|z| - z_2),$$

where

$$z_{1,2} = \frac{1}{w_o d_o} \pm \sqrt{\frac{1}{(w_o d_o)^2} + \frac{\eta_1 w_o^2 N}{2d_o}(1 + \frac{\alpha\gamma}{d_o^2})}$$

Thus, for $x \geq 0$, a necessary condition for $g(z) = 0$ is that $|z| \leq z_1$ (where z_1 is the positive zero).

Note: in a similar way, we can show that to find the zeros of $g(z) = 0$ in the region $x \geq v$ (where v may be negative or positive), it suffices to consider $|z| \leq z(v)$ where

$$\begin{aligned} z(v) \quad &:= \quad \frac{1}{2}\left(\frac{1+e^{-v}}{w_o d_o}\right. \\ &+ \quad \left.\sqrt{\left(\frac{1+e^{-v}}{w_o d_o}\right)^2 + 4\frac{\eta_1 w_o^2 N e^{-v}}{2d_o}(1 + \frac{\alpha\gamma e^{-v}}{d_o^2})}\right) \end{aligned}$$

b) A numerical example: This numerical example is picked to show that the system could be unstable for meaningful choices for the values of the parameters.

$\eta_1 = 0.001$, $\eta_2 = 0.02$, $\alpha = 1$, $d_0 = 3$. Further let $N = 5$, $\gamma = 250/500 = 0.5$, $RC = 100$ so that $RC >> q_o$.

$$W_o = \frac{CR}{N(1 + \gamma/d_o)} = 100/(5 \times 1.166) = 17.142857$$

Hence

$$p_o = 2/W_o^2 = 0.006806,\ \beta = (d_0/\alpha - 1)/W_o = 0.11666$$

$$\text{and } q_0 = \frac{p_0 + \eta_2}{\eta_1} = 26.80555$$

which is indeed small with respect to RC.

We can verify that $z = 0.09964430039 + 0.4704656808 * I$ is a zero of (15), and hence the system is unstable.

B. Using the Model: Impact on the Stability

Examining the form of the equation $g(z) = 0$ (whose solutions provide the stability condition), we observe the following.

- The stability condition is not a function of the round trip delay. This is seen directly from (7)-(10): if we scale time so that a time unit corresponds to a round trip time, we get a new system of equations that does not involve R.
- From the form of the expression $g(z)$ we see that for the same values of $C, R, N, \gamma, \eta_1, \eta_2$ and for the same value d_o at equilibrium, the stability region can change according to the choice of the parameters α and β. $\beta = 0$

corresponds to a non-dynamic value of d (i.e. a value that does not change with W) in which case $\alpha = d_o$. (Other advantages of dynamic d have already been illustrated in [6] in the context of mobile communications.)

- We also see that for fixed $C, R, N, \gamma, \alpha, \beta$ and a fixed queue size q_o, at equilibrium, $g(z)$ will be influenced by η_1 (and hence the stability region). In fact $g(z)$ does not depend on η_2, but note that since we assume that R includes the mean queueing delay (which is proportional to q_o), this means that fixing q_o and η_1 already determines η_2.

In the following, we choose a scenario that shows clearly the impact of using a dynamic delayed ACK factor d on the RED stability. We keep unchanged the parameters used in the numerical example. Then, we set $N = 14$, and we vary α from 1 to 4. Since we fix d_0, β is deduced. Table I presents the results concerning the stability of the system.

TABLE I
IMPACT OF THE DYNAMIC CONTROL ON THE STABILITY

$d0$	α	$\Rightarrow \beta$	The system is
3	1	0.33 ($W \nearrow \; d \nearrow$)	stable
3	2	0.08 ($W \nearrow \; d \nearrow$)	stable
3	3	0 (no control)	unstable
3	4	-0.04 ($W \nearrow \; d \searrow$)	unstable

The first observation we made is that when $\alpha = 3 (= d_0)$, the system is unstable, which means that when there is no dynamic variation of the delayed factor d, the system is unstable. However, for $\alpha = 1$ or 2, the system is stable. Besides, β is positive, which means that the linear control is correct ($d(t)$ is a non-decreasing function of $W(t)$). When $\alpha = 4$, β is negative and the system is unstable. These results confirm the fact that when the TCP window increases, we should increase the delay factor d. Moreover, using an adequate adaptive control of d, we can improve the stability of the system.

IV. Linear Quadratic approaches for flow control

In this section we study the problem of the optimal time interval between successive transmission of information using a control theoretic approach within a linear quadratic framework. Such frameworks have frequently been used for approximating rate-based flow control by ignoring the nonlinearity at queue boundaries (empty or full buffers) [5].

More precisely, we make three simplifying (but realistic) assumptions:

1. *Fluid approximation.* We replace a discrete number of packets by a continuous fluid. The fluid approximation is justified by the fact that in today's technology, buffering capabilities are very large (several thousands) in terms of the number of packets they can store, so that the error of replacing an integer number of packets by a real number is small relative to the size of buffers. This type of approximation is common both in the design of controllers in high-speed telecommunication networks (see e.g. [11]) as well as in performance evaluation of existing controllers [15], [16].

2. *Linearized dynamics.* It is assumed that the network has linearized dynamics for the control of queue length; see (16) below: we neglect losses when the buffer is full, and we neglect the boundary effect of an empty queue. To motivate this linearization, we use the fact that the controllers that we derive operate in a region close to full throughput utilization, so that the queue will almost never empty. The full utilization is indeed common in the control of ABR switches, see e.g., [14], and is possible by *regulating* the (controlled) input rate and adapting it to the available capacity.

As discussed below, we set some desirable threshold on the queue length which we attempt to track, precisely so as to avoid large queues (which might result in losses) or empty queues (which might result in loss of potential throughput). When a control mechanism has a full utilization, then the nonlinearity around zero disappears. For similar models with a single controller, simulations have confirmed [4], [3] that *controlled* linearized models lead to trajectories that are very close to the original one. The fact that the other boundary is ignored is motivated by similar arguments since our optimal control will be shown to be symmetric with respect to positive or negative deviations around the target queue value.

3. *Bottleneck assumption.* We assume that all performance measures (such as throughput, delays, loss probabilities, etc.) are determined essentially by a bottleneck node. This assumption admits theoretical as well as experimental justifications; see [10].

4. *Information flow.* We assume that information is sent to the controller on the queue length periodically, each $1/\lambda$ seconds.

We now introduce the model. Let $q(t)$ denote the queue length at a bottleneck link. We assume that the information packets use the same link and have priority over data packets. The link capacity available to data packets, C, is thus assumed to depend on λ:

$$C = C(\gamma) = C - \frac{a}{\lambda}$$

We assume that other non-controlled inputs share the buffer, and their total input rate is given by $C_1 + v(t)$ where C_1 is some constant and v represents a stochastic process with zero mean. The average rate of the controlled source is assumed to be $C_2 = C - C_1$. Thus we let $u(t) + C_2$ be the rate of the controlled input into the queue at time t. Then the queue length dynamics is given by

$$dq = u\mathrm{d}t + \mathrm{d}v \tag{16}$$

which is *idealized* because the end-point effects have been ignored. The objective of the flow controller are *(i)* to ensure that the bottleneck queue size stays around some desired level $\overline{Q}$, and *(ii)* to minimize variations of the rates. The choice of $\overline{Q}$ and the variability around it have a direct impact on loss probabilities and throughput. We therefore define a shifted version of q: $x(t) := q(t) - \overline{Q}$, in view of which (16) now becomes

$$dx = u\mathrm{d}t + \mathrm{d}v \tag{17}$$

An appropriate local cost function that is compatible with the objectives stated above would be the one that penalizes variations in $x(t)$ and $u(t)$ around *zero* — a candidate for which is the weighted quadratic cost function: $x^2 + ku^2$.

We seek an optimal control policy among those which choose $u(t)$ as a function of the queue length at the times when information is available:

$$u(t) = \mu(x(0), x(\lambda), ..., x(n\lambda)),$$

for

$$t \in [n\lambda, (n+1)\lambda), \quad n = 0, 1, 2, ...$$

A. LQG model

We assume that v is a zero mean Brownian motion with variance r. The expected average cost for a given policy μ and initial state x is defined as

$$J(x,\mu,\lambda) = \lim_{T\to\infty} \frac{1}{T} E_x^{\mu} \left[\int_0^T (x^2(t) + ku^2(t))dt \right]$$

where E_x^{μ} is the expectation with respect to the probability measure induced by a policy μ and an initial state x. We first seek to obtain the optimal policy and value for a given λ:

$$J(x,\lambda) := \min_{\mu} J(x,\mu,\lambda).$$

Theorem IV.1: The optimal value of the LQG problem is independent of the initial state x and is given by

$$J(\lambda) = \frac{\lambda^2}{2} r + \sqrt{k} r$$

and the unique policy that attains the minimum is given by

$$u^*(t) = -\frac{1}{\sqrt{k}} \exp\Big(-\frac{t - n\lambda}{\sqrt{k}} \Big) x(n\lambda) \tag{18}$$

for $t \in [n\lambda, (n+1)\lambda)$.

Proof. A proof can be found in the Appendix.

We thus conclude that as long as λ^2 is much smaller than $\sqrt{k}$, the value is quite insensitive to changes in the spacing λ. On the other hand, when it is much larger than $\sqrt{n}$, we see that the spacing of information packets has a huge impact on the performance: the cost grows quadratically in the spacing.

B. Optimal spacing of information for the LQG model

We found the optimal control policy for a given parameter λ of spacing of information packets. Our next goal is to optimize λ. In addition to the cost $J(\lambda)$ which we have obtained, we assume that the flow control has some utility $U(\lambda)$ that can represent the utility for the average throughput $C - a/\lambda$ available for the data packets. We shall consider

1) A utility linear in the average throughput: $U(\lambda) = C - a/\lambda$.
2) A utility which is logarithmic in the average throughput: $U(\lambda) = \log(C - a/\lambda)$.

The cost to be minimized is in both cases $Z(\lambda) = J(\lambda) - \gamma U(\lambda)$. Taking the derivative and equating it to zero we obtain the following:

Theorem IV.2: The optimal information spacing is given by

(i) linear case:

$$\lambda^* = \Big(\frac{\gamma a}{r} \Big)^{1/3}$$

(ii) logarithmic case:

$$\begin{aligned}\lambda^* &= \frac{1}{6}\left(108\beta + 8\alpha^3 + 12\sqrt{81\beta^2 + 12\beta\alpha^3}\right)^{1/3} \\ &+ \frac{\alpha}{3} + \frac{2\alpha^2}{3\left(108\beta + 8\alpha^3 + 12\sqrt{81\beta^2 + 12\beta\alpha^3}\right)^{1/3}}\end{aligned}$$

where $\alpha = a/C$ and $\beta = \gamma a/(rC)$.

C. An H^∞ approach

In this subsection we do not make statistical assumptions on the distribution of the noise process, and we choose instead a robust approach that guarantees the best performance under the worst case conditions. More precisely, we define

$$V(\mu, \lambda) = \sup_{\{v_n\}_{n=-\infty}^{\infty}} \frac{L(\mu, \nu, \lambda)}{\|v\|} \tag{19}$$

where

$$L(\mu, \nu, \lambda) = \int_{-\infty}^{\infty} (x^2(t) + ku^2(t))dt$$

and where $\|v\|$ is the L^2 norm of v: $\|v\| = \sqrt{\sum_{n=-\infty}^{\infty}(v_n)^2}$. One then wishes to find μ that minimizes $V(\mu, \lambda)$; denote the infimum over μ by $(\Gamma^*)^2$.

Define a soft-constrained cost function

$$L_\Gamma(\mu, v, \lambda) := L(\mu, v, \lambda) - \Gamma^2 \|v\|^2,$$

and consider a two player game where L_Γ is to be minimized by Player 1 (controlling μ) and maximized by Player 2 (controlling (v)). If there exists some policy μ^* for the problem of minimizing $L(\mu, \lambda)$, then it has the property [9]:

$$\sup_v L_{\Gamma^*}(\mu^*, v, \lambda) = \inf_\mu \sup_v L_{\Gamma^*}(\mu, v, \lambda).$$

The quantity above is the upper value of the zero-sum dynamic game with kernel L_{Γ^*}, which is in fact equal to zero. It can actually be shown that for any $\Gamma \geq \Gamma^*$, the upper value of the game with parameterized kernel L_Γ is zero, and for $\Gamma < \Gamma^*$, its upper value is infinite. Hence, Γ^* is the smallest positive scalar Γ for which the zero-sum game with kernel L_Γ has a finite upper value.

Instead of obtaining μ^* defined above, we will in fact solve a parameterized class of controllers, $\{\mu^\Gamma, \Gamma > \Gamma^*\}$, where μ^Γ is obtained from $\sup_v L_\Gamma(\mu^\Gamma, v, \lambda) = \inf_f \sup_v L_\Gamma(\mu, v, \lambda)$. The controller μ^Γ will clearly have the property that it ensures a performance level Γ^2 for the index adopted at (19), i.e. the attenuation is bounded by

$$\frac{\{L(\mu^\Gamma, v, \lambda)\}^{1/2}}{\|v\|} \leq \Gamma \ \text{ for all } v. \tag{20}$$

It will turn out that the limit $\lim_{\Gamma\to\infty}\mu^\Gamma =: \mu^\infty$ is a well-defined controller, and solves uniquely the control problem with the previous Gaussian model.

Note that Γ^* will be a function of λ. Our goal is to determine the largest value of spacing for which there exists a policy that guarantees a given level of attenuation.

Theorem IV.3: For a given level of attenuation Γ, the largest spacing λ of information for which there is a policy μ^Γ such that (20) holds is given by

$$\lambda = \Gamma\frac{\pi}{2} - \Gamma \arctan \frac{1}{\sqrt{\Gamma^2/k - 1}}$$

for $\Gamma > k$.

Proof. The result follows from the material in [9, Chapter 5.3].

Figure 7 shows how the spacing grows as a function of the desired achievable attenuation for $k = 1$. The spacing (y-axis) is seen to be concave increasing in the required attenuation, and it grows asymptotically linear in it.

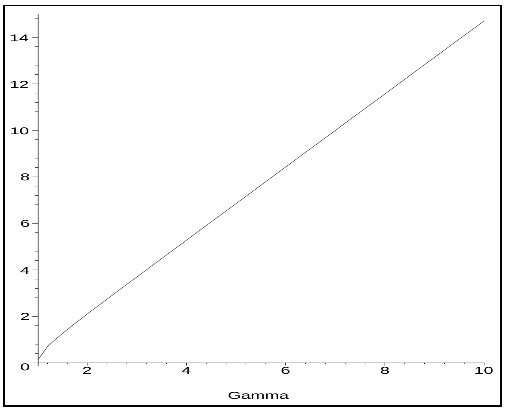

Fig. 7. Maximum spacing as a function of the attenuation Γ

V. Conclusion

We investigated in this paper different facets of the problem of determining how frequently information should be sent to the source in congestion control mechanisms. We showed that the rate of information can have quantitative impacts, in particular on the throughput of the system, as conveying information requires resources that are then not available to the data packets. It can also have qualitative impacts: it may impact the stability of a congestion control algorithm. More precisely, we show a scenario where a dynamic delayed ACK mechanism make the system stable.

References

[1] K. Abratchenkov, U. Ayesta , E. Altman, P. Nain, C. Barakat ”The effect of router buffer size on the TCP performance”, in Proc. LONIIS workshop on Telecommunication Networks and Teletraffic Theory, St. Petersburg, Russia, pp. 116-121, January 2002.

[2] E. Altman and T. Başar, “Multi-user rate-based flow control”, *IEEE Trans. on Communications*, pp. 940-949, 1998.

[3] E. Altman, F. Baccelli and J. C. Bolot, “Discrete-time analysis of adaptive rate control mechanisms”, *High Speed Networks and their performance*, H.G. Perros and Y. Viniotis Eds., North Holland, 121-140, 1994.

[4] E. Altman and T. Başar, “Optimal rate control for high speed telecommunication networks, *Proc. 34th IEEE Conference on Decision and Control*, New Orleans, Louisiana, Dec. 1995.

[5] E. Altman, T. Başar and R. Srikant, "Robust Rate Control for ABR Sources" *IEEE INFOCOM*, San-Francisco, California, USA, March 29 - April 2, 1998.

[6] E. Altman and T. Jiménez, "Novel delayed ACK techniques for improving TCP performance in multihop wireless networks", Proc. PWC 2003 Personal Wireless Communications, Sept. 2003, Venice, Italy. Available at http://www-sop.inria.fr/mistral/personnel/Eitan.Altman/mobile.html

[7] U. Ayesta, K. Avrachenkov, E. Altman, C. Barakat, P. Dube, "Simulation Analysis and Fixed Point Approach for Multiplexed TCP flows", In the *Proc. PFLDnet'03: Workshop on Protocols for Fast Long-Distance networks*, CERN-Geneva, Switzerland, February 2003.

[8] C. Barakat and E. Altman, "On ACK filtering on a slow reverse channel", *Proc. first international Workshop on Quality of future Internet Services (QOFIS)*, Berlin, Germany, September 2000.

[9] T. Başar and P. Bernhard, *H^∞-Optimal Control and Relaxed Minimax Design Problems: A Dynamic Game Approach*, Birkhäuser, Boston, MA, 1991 (2nd edition, 1995).

[10] J-C. Bolot, "End-to-end packet delay and loss behavior in the Internet", *Proc. ACM Sigcomm '93, pp. 289-298,* San Francisco, CA, Sept. 1993.

[11] J.-C. Bolot, A. U. Shankar, "Analysis of a fluid approximation to flow control dynamics", *Proc. IEEE Infocom*'92, pp. 2398-2407, Florence, Italy, May 1992.

[12] W. H. Fleming and H. M. Soner, *Controlled Markov processes and viscosity solutions*, Springer Verlag, NY, 1993.

[13] C. Hollot, V. Misra, D. Towsley and W.-B. Gong, "A control theoretic analysis of RED" *CMPSCI Technical Report TR 00-41*, July 2000.

[14] R. Jain, "ERICA+", contribution ATM Forum/95-1346.

[15] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda and H. Miyahara, "Rate-based control for ATM Networks," *Computer Communication Review, ACM-Sigcomm,* special issue in ATM, ed. R. Jain and K.Y. Siu, Vol 25, No. 2, pp. 60-71, 1995.

[16] M. Ritter, "Network buffer requirements of the rate-based control mechanism for ABR services," *IEEE INFOCOM'96,* San Francisco, USA, March 1996.

[17] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation", *ACM SIGCOMM'98*, 1998.

[18] J. W. Cohen "The Single Server Queue", *North Holland publishing company*, 1982.

[19] E. Altman, C. Barakat and V. M. Ramos R., "Queuing analysis of simple FEC schemes for Voice over IP", *Communications Networks, vol. 39, no. 2, pp. 185-206*, June 2002.

[20] M. Allman, "TCP Congestion Control with Appropriate Byte Counting (ABC)", *Request for Comments: 3465*, February 2003.

[21] M. Allman, V. Paxson and W. Stevens, "TCP Congestion Control", *Request for Comments: 2581*, April 1999.

[22] S. Alouf, "Parameter Estimation and Performance Analysis of Several Network Applications", *Ph.D. Thesis, Department of Sciences, University of Nice Sophia-Antipolis*, November, 2002.

VI. APPENDIX

A. Computation of f for Section II

Let $s = re^{i\theta} = r\cos\theta + i\,r\sin\theta$, and $\beta = \frac{\alpha}{Z}$ and $L = \frac{\lambda}{C}$ then we can write

$$\begin{aligned} f &= \frac{1}{2\pi\, i} \oint_{D_r} \frac{1}{G(s)\, s^{K-1}}\, \mathrm{d}s \\ &= \frac{1}{2\pi\, r^{K-2}} \int_0^{2\pi} \frac{e^{-i\theta(K-2)}\, \overline{G(\theta)}}{|G(\theta)|^2}\, \mathrm{d}\theta \end{aligned} \tag{21}$$

where $G(s) = \frac{1}{d+1} e^{-\frac{\alpha\lambda(1-s)}{ZC}} + \frac{d}{d+1} e^{-\frac{\lambda(1-s)}{C}} - s$

$$e^{-i\theta(K-2)}\, \overline{G(\theta)} =$$

$$
\begin{aligned}
& \cos(\theta K - 2\theta)\left(\frac{e^{-\beta L(1-r\cos(\theta))}\cos(\beta Lr\sin(\theta))}{d+1}\right.\\
+ \quad & \left.\frac{de^{-L(1-r\cos(\theta))}\cos(Lr\sin(\theta))}{d+1} - r\cos(\theta)\right)\\
+ \quad & \sin(\theta K - 2\theta)\left(-\frac{e^{-\beta L(1-r\cos(\theta))}\sin(\beta Lr\sin(\theta))}{d+1}\right.\\
& \left.-\frac{de^{-L(1-r\cos(\theta))}\sin(Lr\sin(\theta))}{d+1} + r\sin(\theta)\right)\\
+ \quad & i\left[-\sin(\theta K - 2\theta)\left(\frac{e^{-\beta L(1-r\cos(\theta))}\cos(\beta Lr\sin(\theta))}{d+1}\right.\right.\\
+ \quad & \left.\frac{de^{-L(1-r\cos(\theta))}\cos(Lr\sin(\theta))}{d+1} - r\cos(\theta)\right)\\
+ \quad & \cos(\theta K - 2\theta)\left(-\frac{e^{-\beta L(1-r\cos(\theta))}\sin(\beta Lr\sin(\theta))}{d+1}\right.\\
& \left.\left.-\frac{de^{-L(1-r\cos(\theta))}\sin(Lr\sin(\theta))}{d+1} + r\sin(\theta)\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
& |G(\theta)|^2 = \\
& -\left(-d^2e^{2L(-1+r\cos(\theta))} - r^2\right.\\
& +2\,e^{\beta L(-1+r\cos(\theta))}\cos(\beta Lr\sin(\theta))\,r\cos(\theta)\\
& +2\,e^{\beta L(-1+r\cos(\theta))}\sin(\beta Lr\sin(\theta))\,r\sin(\theta)\\
& -2\,e^{L(-1+r\cos(\theta))(\beta+1)}\sin(\beta Lr\sin(\theta))\,d\sin(Lr\sin(\theta))\\
& -2\,e^{L(-1+r\cos(\theta))(\beta+1)}\cos(\beta Lr\sin(\theta))\,d\cos(Lr\sin(\theta))\\
& +2\,e^{\beta L(-1+r\cos(\theta))}\cos(\beta Lr\sin(\theta))\,r\cos(\theta)\,d\\
& +2\,d^2e^{L(-1+r\cos(\theta))}\cos(Lr\sin(\theta))\,r\cos(\theta)\\
& +2\,de^{L(-1+r\cos(\theta))}\cos(Lr\sin(\theta))\,r\cos(\theta)\\
& +2\,e^{\beta L(-1+r\cos(\theta))}\sin(\beta Lr\sin(\theta))\,r\sin(\theta)\,d\\
& +2\,d^2e^{L(-1+r\cos(\theta))}\sin(Lr\sin(\theta))\,r\sin(\theta)\\
& +2\,de^{L(-1+r\cos(\theta))}\sin(Lr\sin(\theta))\,r\sin(\theta)\\
& \left.-e^{2\beta L(-1+r\cos(\theta))} - r^2d^2 - 2\,r^2d\right)\Big/(d+1)^2
\end{aligned}
$$

We can check that the real parts of $e^{-i\theta(K-2)}\,\overline{G(\theta)}$ and $|G(\theta)|^2$ are even functions in θ, and that the imaginary part of $e^{-i\theta(K-2)}\,\overline{G(\theta)}$ is odd. Then, equation (21) is reduced to:

$$f = \frac{1}{\pi\, r^{K-2}} \int_0^{\pi} \frac{Real(e^{-i\theta(K-2)}\,\overline{G(\theta)})}{|G(\theta)|^2}\, \mathrm{d}\theta \tag{22}$$

Choosing a radius r for the integration First, we notice that when $\beta = 1$, i.e. $\alpha = Z$, then $L = \rho$ and the queueing system is reduced to $M/D/1/K$. It is possible to find quickly a radius in this case [22]. Let $r(\rho)$ denotes such radius. We can show that $r(\rho)$ can also be used to compute the integral in the case where $\beta < 1$.

B. *Sketch of proof of Theorem IV.1*

We can view the cost to be minimized as

$$J(\lambda, u) = \lim_{n\to\infty} \frac{J_n(\lambda, u)}{n},$$

where

$$J_n(\lambda, u) = \sum_{i=1}^{n} \int_{i\lambda}^{(i+1)\lambda} E[x_t^2 + k u_t^2] dt.$$

We first consider the problem on a finite horizon, i.e. the problem of minimizing $J_n(\lambda, u)$ over the feasible u. We know from the theory of LQG control that the optimal value at times $n\lambda, n = 0, 1, 2, ...$ is quadratic in the initial state x, so that it can be written as $P(n)x^2$ where $P(n)$ are constants that should be determined. For $t \in [n\lambda, (n+1)\lambda)$, we know from the certainty equivalence approach that the optimal action u_t is of the form

$$u_t = -p(t, n\lambda)\hat{x}_t/k \tag{23}$$

where $p(t, n\lambda)$ is the solution of the Riccati equation

$$\dot{p}(t, n\lambda) = \frac{p(t, n\lambda)^2}{k} - 1$$

[12, p. 143] and where $\hat{x}_t$ is the estimation of x_t given simply by $\mathrm{d}\hat{x}_t/\mathrm{d}t = u_t$. Combining with (23) we obtain $\mathrm{d}\hat{x}_t/\mathrm{d}t = -p(t, n\lambda)\hat{x}_t/k$. For the limiting case corresponding to the infinite horizon, we have $p^2 = k$ so that $\mathrm{d}\hat{x}_t/\mathrm{d}t = -\hat{x}_t/\sqrt{k}$. We then obtain $\hat{x}_t = \exp(-(t - n\lambda)/\sqrt{k})$ for $t \in [n\lambda, (n+1)\lambda)$ which then yields (18).