Estimation of the number of virtual connections in a rate-based flow control problem.

Andreas Pitsillides
Computer Science Department
University of Cyprus
Nicosia, Cyprus
E-mail:Andreas.Pitsillides@ucy.ac.cy

Petros Ioannou, Marios Lestas
Electrical Engineering Department
University of Southern California
Los Angeles, California

***Abstract*— In this paper we deal with the rate-based flow control problem which we formulate as a tracking problem where the goal is to track a reference queue size chosen by the network administrator. We consider the single bottleneck link topology and use a non-linear model for the queueing dynamics. A known algorithm is then extended to improve performance. Integral action is added to remove any steady state errors and the control signal is then normalized by the estimated number of flows. The main contribution of this work is the estimation algorithm that we propose, which is based on on-line parameter identification techniques and is shown analytically to converge to the correct number of flows exponentially fast. The algorithm does not require maintenance of per flow states and can thus form the basis for the development of an end-to-end congestion control protocol. The performance of the proposed algorithm is evaluated through simulations carried out on the Ns simulator. Good estimation of the number of sessions and good tracking of the reference queue size are observed.** [1]

I. Introduction

As bandwidth-delay products increase dramatically, there is strong belief that congestion control mechanisms currently serving the Internet will lead to underutilization of the network and degradation of the quality of service provided to the users. This has stimulated the research community to explore ways to design new congestion control algorithms in a control theoretic framework, through mathematical modelling and analysis. This design approach leads to solutions with analytically provable performance characteristics as opposed to the intuitive ad-hoc design methods where performance could only be evaluated through excessive simulations and actual implementation. Some early attempts based on non-linear and adaptive control which is also considered here are recorded in [1] and [2].

The principle aim of any congestion control algorithm is to provide low-delay, low loss services to each user and in the case where the demand for resources exceeds availability, to distribute the network resources in a fair way. The resources under consideration are the link bandwidths. So the objective is to find algorithms which at steady state converge to a fair allocation of sending rates among the users, while at the same time achieve bounded queue sizes at the buffers within the network in order to ensure low propagation delays.

[1]This work was partly funded by IST-2001-34900 SEACORN project.

In the literature, this is referred to as the rate-based flow control problem and it is common practice to consider the single bottleneck link case because of its simplicity and the fact that the algorithms developed can be easily extended to more complex network topologies. Several algorithms have been proposed to calculate the explicit rate and most of them were intended to provide solutions for ABR services in ATM networks ([3]). One class of algorithms attempt a direct calculation of the fair share by dividing the available bandwidth at each link with the number of bottlenecked connections ([4], [5]). The exact computation is usually corrupted by inaccuracies and so additional control mechanisms are introduced in the proposed solutions. Another approach is to use bandwidth availability and queue length information to compute the explicit rate ([6], [7]). The problem then reduces to a standard tracking problem where the goal is to track a reference queue size chosen by the designer. Both discrete and continuous time formulations appear in the literature. Standard proportional ([8]) and PI ([9]) compensators have been suggested and analyzed in time delay environments where the link dynamics are modelled as simple integrators. A common feature of some of the algorithms is that the controller gains are normalized by the number of the locally bottlenecked connections. This is done in order to decouple closed loop properties of the system from the number of network users which is a time varying parameter. So, the stability and performance properties of the designed system are effectively unaffected by the number of active sessions.

It is evident from the above that the estimation of the number of connections is a very important part of a wide class of congestion control algorithms. The estimation must be done without maintaining per flow state at each link so that there is no increase in the computational overhead. This makes the estimation a very challenging research topic. The difficulty lies in that the dynamics of the estimation algorithm may interact with the dynamics of the queue control algorithm leading to closed loop instability, when the two are designed independently. The problem has been considered by a number of researchers but the proposed algorithms rely on heuristic methods ([9], [4], [7], [10]). In this paper we follow a different approach and we propose an estimation algorithm which is based on online parameter identification techniques. We consider the single bottleneck link case and

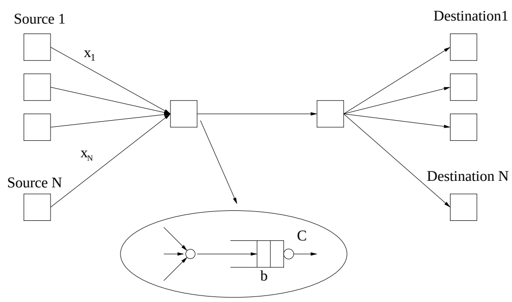


Fig. 1. Single bottleneck link topology.

we formulate the problem as a tracking problem. We use a non-linear model of the queueing dynamics, we extend the algorithm developed in [11] to improve performance and we then develop the estimation algorithm based on both the model and the controller. The estimator is shown analytically to converge to the correct number of flows exponentially fast. The algorithm is evaluated using simulations on Ns and is found to fulfill the design objectives.

The paper is organized as follows. In section II we describe our model and formulate the problem mathematically, in section III we develop the estimation algorithm and analyze its properties and in section IV we demonstrate the functionality of the proposed solution through simulations.

II. Problem Formulation

We consider a packet switched network which accommodates elastic applications. The applications are assumed to be persistent in the sense that they always have data to send. The data traffic is viewed as a deterministic fluid flow and continuous time queueing and flow dynamics are assumed. For control design and analysis purposes time delays within the network are neglected. Their effect however, on the performance of the control design is examined on simulations. The network comprises of N sources indexed by $i = \{1, 2, 3, ..N\}$, utilizing a single bottleneck link as shown in figure 1. The bandwidth of the bottleneck link is denoted by C and is assumed to be less than the bandwidth of the all the links attached to both the sources and the destinations. The dynamics of these links can thus be neglected. Let x_i be the sending rate of source i and let y be the incoming data rate at the bottleneck link. We associate one physical queue at the bottleneck link and its buffer size at time t is denoted by $b(t)$. A congestion controller is installed at the link and a congestion signal p is generated. This signal is communicated back to the sources through an acknowledgement mechanism and the sending rate of each source is updated using this congestion signal. In the system under investigation the congestion controller is actually generating the desired sending rate such that $x_i = p$, $\forall i$. In addition since all sources receive the same feedback signal their sending rates are always the same and so we denote by x the common sending rate of all sources. It is implied by the above discussion that $x = p$. The queuing dynamics are described by the following non-linear model which was derived in [12] using the approximate fluid flow modelling approach proposed by Agnew ([13]).

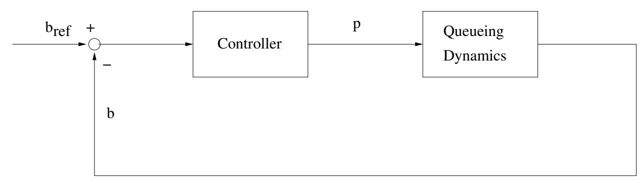


Fig. 2. Control System.

$$\frac{db(t)}{dt} = y(t) - C\frac{b(t)}{1+b(t)}, \qquad b(0) = b_o \tag{1}$$

A block diagram of the control system is shown in figure 2. The objective is then to design the controller (i.e to find how to update the congestion signal p) such that at steady state, the queue size b converges to a reference value b_{ref} chosen by the network operator. The following control algorithm was proposed in [11] to fulfill the design objective.

$$p(t) = max[0, C\frac{b(t)}{1+b(t)} - a_r(b(t) - b_{ref})] \tag{2}$$

Here, we modify the algorithm in three ways. Integral action is introduced to eliminate steady state errors and the congestion signal calculated, is normalized by an estimation of the number of users sharing the bottleneck link ($\hat{N}$), in order to achieve decoupling of the stability and transient properties of the system from time varying parameters such as the number of virtual connections. In addition the lower bound of the signal $p(t)$ imposed by the max operator is changed from 0 to a small positive constant $\alpha > 0$ which is a design parameter. This is done to ensure that the regressor in the estimation algorithm is persistently exciting and thus leads to a correct estimation of the unknown parameter. So, the control algorithm takes the following form:

$$p(t) = \frac{1}{\hat{N}} max[\alpha, C\frac{b(t)}{1+b(t)} - a_p\bar{b}(t) - a_i\bar{k}(t)], \quad \alpha > 0 \tag{3}$$

where $\bar{b}(t) = b(t) - b_{ref}(t)$ and $\bar{k}(t) = \int_0^t (b(\tau) - b_{ref})d\tau$. In this paper we propose an algorithm to calculate $\hat{N}$ based on online parameter identification techniques. The performance of the algorithm and the way it interacts with the proposed controller is evaluated through simulations on Ns.

III. Estimation Algorithm

The first step in designing the on line parameter identification algorithm is to manipulate equations (1) and (3) in order to derive a linear static parametric model (SPM) of the form

$$z = \theta^* \phi \tag{4}$$

The parameter $\theta^* = N$ to be estimated is the number of users sharing the bottleneck link whereas z and ϕ are known signals. Since, as pointed out in the previous section, $x = p$ it is possible to set $y(t) = Np(t)$ in equation (1) to obtain the following:

$$\frac{db(t)}{dt} = Np(t) - C\frac{b(t)}{1+b(t)}, \quad b(0) = b_o \tag{5}$$

We now filter both sides of (5) with the stable filter $\frac{1}{s+\lambda}$, where $\lambda > 0$ is a design parameter we can choose arbitrarily, to obtain:

$$\frac{sb}{s+\lambda} + \frac{1}{s+\lambda}C\frac{b}{1+b} = \frac{Np}{s+\lambda} \tag{6}$$

The above equation is in SPM form, with the following signal correspondence:

$$z = \frac{sb}{s+\lambda} + \frac{1}{s+\lambda}C\frac{b}{1+b} \tag{7}$$

$$\phi = \frac{p}{s+\lambda} \tag{8}$$

$$\theta^* = N \tag{9}$$

Once the parametric model has been generated there exists a wide class of adaptive laws which can be used to estimate the unknown parameter. Since it is known priori that the number of users is greater or equal to 1 we use a gradient projection algorithm which is described by the following equations:

$$\epsilon = \frac{z - \hat{N}\phi}{1 + a_1\phi^2} \tag{10}$$

$$\frac{d\hat{N}}{dt} = \{ \begin{array}{ll} \gamma\epsilon\phi & if\ \hat{N} > 1,\ or\ \hat{N} = 1, \gamma\epsilon\phi > 0 \\ 0 & otherwise \end{array} \tag{11}$$

The signal ϕ is called the regressor, ϵ is referred to as the estimation error and the denominator of ϵ, $m_s^2 = 1 + a_1\phi^2$ is a normalization design signal which improves the robustness and stability properties of the adaptive law. The scaling constant γ is a design parameter and we refer to it as the adaptive gain. It affects the convergence and stability properties of the system. The algorithm has the following properties which are stated as a lemma:

Lemma 3.1: The gradient projection algorithm with instantaneous cost function guarantees that:

- i $\quad \epsilon, \hat{N}, \frac{d\hat{N}}{dt} \in L_\infty$ (bounded)
- ii $\quad \epsilon, \frac{d\hat{N}}{dt} \in L_2$ (square integrable)
- iii $\quad \phi$ is bounded and persistently exciting, i.e. it satisfies $\frac{1}{T}\int_t^{t+T} \phi(\tau)^2 d\tau > \mu$ for all $t \geq 0$ and some constants T, $\mu > 0$, and so $\hat{N}(t) \to N$ exponentially fast

Proof: The proof for (i) and (ii) is found in [14] and is thus omitted. It is important to note that (i) and (ii) are valid independent of whether the regressor ϕ is bounded. The latter is a result of the effect of the normalizing signal m_s^2 which guarantees that the signal $\frac{|\phi|}{m_s} \in L_\infty$. Here we show that ϕ is persistently exciting and bounded. The regressor ϕ is given by:

$$\phi = \frac{1}{s+\lambda}p = \frac{1}{s+\lambda}\frac{r(t)}{\hat{N}(t)} \tag{12}$$

where

$$r(t) = max[\alpha, C\frac{b(t)}{1+b(t)} - a_p\bar{b}(t) - a_i\bar{k}(t)] \tag{13}$$

The effect of the max operator establishes that $r(t) > \alpha$. In addition, $\hat{N}$ is strictly positive and bounded from below due to the projection operator which guarantees that it is greater than 1 and also bounded from above due to (ii). So, $\frac{r(t)}{\hat{N}} > c > 0$ for some positive constant c. It follows that $\phi > \frac{c}{s+\lambda}$. Assuming, without loss of generality, zero initial conditions we deduce that:

$$\phi(t) > \frac{c}{\lambda}(1 - e^{-\lambda t}) \tag{14}$$

The right hand side of the inequality is a bounded strictly increasing function of time and this establishes the existence of constants T, μ such that $\frac{1}{T}\int_t^{t+T} \phi(\tau)^2 d\tau > \mu$ for all $t \geq 0$. The latter shows that ϕ is persistently exciting. $\hat{N}$ was shown above to be bounded and strictly positive. r(t) is also bounded due to the stability of the control law. So, from (12) it follows that ϕ is bounded as well. The fact that ϕ is bounded and persistently exciting establishes that the estimator will converge to the correct number of flows exponentially fast. The proof can be found in [14] and is thus omitted.

Since the algorithm will be implemented in discrete time, we present the discrete time version of the algorithm below:

$$\epsilon(k) = \frac{z(k) - \hat{N}(k-1)\phi(k)}{1 + a_1\phi(k)^2} \tag{15}$$

$$\hat{N}(k) = \{ \begin{array}{ll} \hat{N}(k-1) + \gamma\epsilon(k)\phi(k) & if\ \hat{N}(k) \geq 1 \\ \hat{N}(k-1) & otherwise \end{array} \tag{16}$$

The design parameters γ and a_1 are required to satisfy the following inequality: $0 < \frac{\gamma}{a_1} < 2$

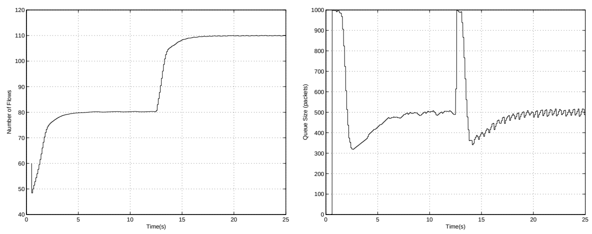
IV. Simulations

The performance of the proposed algorithm was evaluated through simulations carried out on the Network Simulator (Ns). What was investigated, was the ability of the estimator to find the correct number of sessions at steady state and also the ability of the controller to achieve good tracking of the reference queue size at steady state. The transient response of the system was also evaluated. A custom rate protocol was developed on Ns to simulate the proposed algorithm. The network topology shown in figure 1 was examined. The bottleneck link capacity was set to 50Mb/s and the propagation delay of all source destination pairs was at first chosen to be 10msec. The first simulation scenario involved 80 users originally sharing the bottleneck link. Their number was increased to 110 users after 10 seconds. The values of the design parameters of the controller and the estimator are shown in table I. It is important to note that the discrete time version of the algorithm is implemented and so an additional design parameter is the sampling time. This was chosen to be 100msec.

Parameter	Value
a_p	3
a_i	2
b_{ref}	500
γ	1.1
λ	3
a_1	1
α	10
T_s	0.1sec

TABLE I

DESIGN PARAMETER VALUES

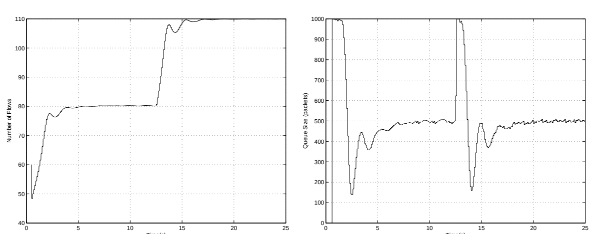


(a) Number of flows (b) Queue size

Fig. 3. Responses with 10msec delay.

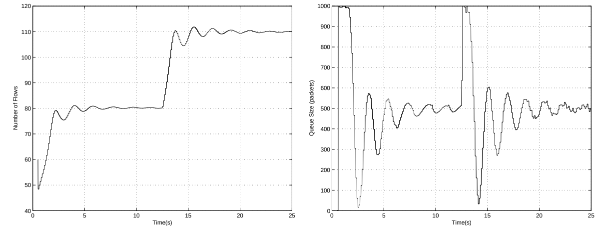
The number of flows calculated by the estimator and the time evolution of the queue size of the buffer at the bottleneck link are shown in figure 3. It is evident that the number of sessions is correctly estimated at steady state and that the proposed controller achieves good tracking of the reference queue size. The responses are also characterized by overdamped transients and good speed of response which are desirable properties.

Since the effect of time delays is not accounted for in the analysis, the effect of increasing delays on the performance of the proposed algorithm was investigated through simulations. The first simulation scenario involved setting the propagation delay to 60msec whereas in the second it was increased to 80msec. The resulting time responses are shown in figures 4 and 5 respectively. The algorithm exhibits good robustness properties with respect to variations in the time delay, since good estimation and tracking is still achieved at steady state. However, the stability margins seem to decrease as the delay increases and the degradation in the transient performance is evident.



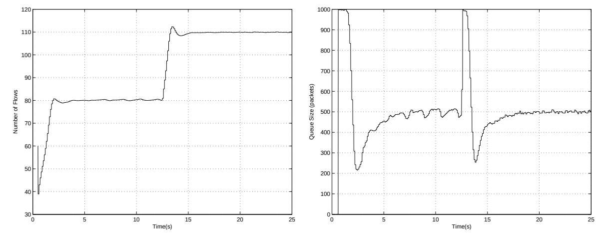
(a) Number of flows (b) Queue size

Fig. 4. Responses with 60msec delay.



(a) Number of flows (b) Queue size

Fig. 5. Responses with 80msec delay.



(a) Number of flows (b) Queue size

Fig. 6. Responses with $\gamma = 1.8$.

What was finally investigated was the effect of the parameter γ as this appears in the estimation algorithm. There are no analytical tools to find the correct parameter value but it is known that the choice is a trade-off between speed of convergence and stability. We increase the value from 1.1 to 1.8 and observe the effect on the time response which is shown in figure 6. As expected the increase does increase the speed of response of the system at the expense of lower damping, which results in the slight overshoot of the flow estimation. This effect becomes more evident at even higher values of the adaptive gain γ.

V. CONCLUSIONS AND FUTURE WORK

In this paper we extend an algorithm presented in [11] and propose a novel way to estimate the number of active sessions through a bottleneck link, using on-line parameter identification techniques. Simulations demonstrate the effectiveness of the algorithm in both its tracking ability and the accuracy of the estimation. The algorithm is found to perform well even in the presence of delays. This work has been motivated by the need to improve congestion control schemes in the IDCC framework proposed in [11]. The latter is intended to serve applications in the core and RAN networks of UMTS systems ([15]). Since the analysis performed in this work applies only to the single bottleneck link case the main objective is to extend the proposed algorithm to more general network topologies. It is also important to relax the no delay assumption. We aim at providing analytical proofs for both the local and global behavior of the system and evaluate the performance through simulations on Ns (Network Simulator) with the help of packages which we have already developed.

REFERENCES

[1] A. Pitsillides and J. Lambert. Adaptive connection admission and flow control: quality of service with high utilisation. *IEEE INFO-*

COM'94, 13th Conference on Computer Communications, Toronto, Ontario, Canada, pages 1083–1091, 1994.

[2] A. Pitsillides, P. Ioannou, and D. Tipper. Integrated control of connection admission, flow rate, and bandwidth for atm based networks. *IEEE INFOCOM'96, 15th Conference on Computer Communications, San Francisco, USA, March 24-28, 1996, pp. 785-793.*, pages 785–793, 1996.

[3] E. J. Hernandez-Valencia, L. Benmohamed, S. Chong, and R. Nagarajan. Rate control algorithms for the atm abr service. *Europ. Trans. Telecom.*, 8(1), 1997.

[4] S. Fahmy R. Goyal B. Vandalore S. Kalyanaraman, R. Jain. The erica switch algorithm for abr trafiic management in atm networks. *IEEE/ACM Transactions on Networking*, 8(1):87–98, 2000.

[5] Lampros Kalampoukas, Anujan Varma, and K. K. Ramakrishnan. An efficient rate allocation algorithm for ATM networks providing max-min fairness. In *HPN*, pages 143–154, 1995.

[6] Meerkov SM. Benmohamed L. Feedback control of congestion in packet switching networks: The case of a single congested node. *IEEE/ACM Transactions on Networking*, 1(6):693–707, 1993.

[7] Ching-Fong Su, Gustavo de Veciana, and Jean Walrand. Explicit rate flow control for ABR services in ATM networks. *IEEE/ACM Transactions on Networking*, 8(3):350–361, 2000.

[8] S. Chong, R. Nagarajan, and Y. Wang. Designing stable abr flow control with rate feedback and open loop control:first-order control case. *Performance Evaluation*, 34:189–206, 1998.

[9] Song Chong, Sangho Lee, and Sungho Kang. A simple, scalable, and stable explicit rate allocation algorithm for MAX-MIN flow control with minimum rate guarantee. *IEEE/ACM Transactions on Networking*, 9(3):322–335, 2001.

[10] J. W. Cho H. W. Lee. A distributed max-min flow control algorithm for mulri-rate mulricast flows. *IEEE ACM Trans. on Networking, in review*.

[11] A. Pitsillides, P. Ioannou, and L. Rossides. Congestion control for differentiated-services using non-linear control theory. *In Proceedings of the Sixth IEEE Symposium on Computers and Communications*, pages 726–733, 2001.

[12] J. Filipiak. Modelling and control of dynamic flows in communication networks. *Springer Verlag*, 1988.

[13] C. Agnew. Dynamic modelling and control of congestion prone systems. *Oper. Res.*, 24(3):400–419, 1976.

[14] J. Sun P. Ioannou. Robust adaptive control. *Prentice Hall*, 1996.

[15] Ist-seacorn project - www.seacorn.ptionavacao.pt.