Integration of TCP-Friendly Streaming Sessions and Heavy-Tailed Elastic Flows*

René Bekker[†,⋆], Sem Borst[†,⋆,‡], Rudesindo Núñez-Queija[†,⋆]

[†]Department of Mathematics & Computer Science, Eindhoven University of Technology,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[⋆]CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
[‡]Bell Laboratories, Lucent Technologies, P.O. Box 636, Murray Hill, NJ 07974, USA

ABSTRACT

We consider a fixed number of streaming sessions sharing a bottleneck link with a dynamic population of elastic flows. We assume that the sizes of the elastic flows exhibit heavy-tailed characteristics. The elastic flows are TCP-controlled, while the transmission rates of the streaming applications are governed by a so-called TCP-friendly rate control protocol.

Adopting the Processor-Sharing (PS) discipline to model the bandwidth sharing, we investigate the tail distribution of the deficit in service received by the streaming sessions compared to a nominal service target. The latter metric provides an indication for the quality experienced by the streaming applications. The results yield valuable qualitative insight into the occurrence of persistent quality disruption for the streaming users. We also examine the delay performance of the elastic flows.

1. INTRODUCTION

Over the past decade, TCP has gained ubiquity as the predominant congestion control mechanism in the Internet. While TCP is adequate for best-effort elastic traffic, such as file transfers and Web browsing sessions, it is less suitable for supporting delay-sensitive streaming applications. In particular, the inherent fluctuations in the window size adversely impact the user-perceived quality of real-time streaming applications. As a potential alternative, UDP could be used to avoid the wild oscillations in the transmission rate. Since UDP does not respond to congestion, it may cause severe packet losses however, and give rise to unfairness in the competition for bandwidth with TCP-controlled flows.

Discriminatory packet scheduling mechanisms provide a further alternative to achieve some form of prioritization of streaming applications. However, the implementation of scheduling mechanisms is surrounded with substantial controversy, because it entails major complexity and scalability issues. In addition, prioritization of streaming applications may cause performance degradation and even starvation of TCP-controlled flows that back off in response to congestion. Evidently, the latter issue gains importance as the amount of streaming traffic in the Internet grows.

The above considerations have motivated an interest in *TCP-friendly* or *equation-based* rate control protocols for streaming applications [5, 10, 12]. The key goal is to eliminate severe fluctuations in the window size and adjust the transmission rate in a smoother manner. In order to ensure fairness with competing TCP-controlled flows, the specific aim is to set the transmission rate to the 'fair' bandwidth share, i.e., the throughput that a long-lived TCP flow would receive under similar conditions.

In the present paper we explore the performance of streaming applications under such TCP-friendly rate control protocols. We consider a fixed number of streaming sessions which share a bottleneck link with a dynamic population of elastic flows. The assumption of persistent streaming users is motivated by the separation of time scales between the typical duration of streaming sessions (minutes to hours) and that of the majority of elastic flows (seconds to minutes). We assume that the TCP-friendly rate control results – at the flow level – in a fair sharing of the link rate in a Processor-Sharing (PS) manner, see for instance [2, 8]. We further suppose that the sizes of the elastic flows exhibit heavy-tailed characteristics. The latter assumption is based on extensive measurement studies which show that file sizes in the Internet commonly have heavy-tailed features, see for instance [4].

We consider the probability that a possible deficit in service received by the streaming sessions compared to a nominal service target exceeds a certain threshold. The latter probability provides a measure for the degree of disruption in the quality experienced by the streaming users. We furthermore examine the delay performance of the elastic flows.

In recent papers [3, 7], the authors also consider mixtures of elastic transfers and streaming users sharing the network bandwidth. These papers however focus on different performance metrics.

2. MODEL DESCRIPTION

We consider two traffic classes sharing a link of unit rate. Class 1 consists of a static population of $K \geq 1$ statistically identical streaming sessions. These sessions stay in the system indefinitely. Class 2 consists of a dynamic configuration of elastic flows. These users arrive according to a renewal process with mean interarrival time $1/\lambda$, and have generic service requirements B with distribution $B(\cdot)$ and mean $\beta < \infty$. Let B^r be a random variable distributed as the residual lifetime of B, i.e., $B^r(x) = \mathbb{P}\{B^r < x\} = \frac{1}{\beta}\int_0^x (1 - B(y))\mathrm{d}y$. We assume that $B(\cdot)$ is regularly varying of index $-\nu$, i.e., $B(\cdot) \in \mathcal{R}_{-\nu}$ (and hence $B^r(\cdot) \in \mathcal{R}_{1-\nu}$).

*This work was financially supported by a grant from Philips Research and by the Dutch Ministry of Economic Affairs (via its agency SENTER; project EQUANET)

The elastic flows are TCP-controlled, while the transmission rates of the streaming sessions are adapted in a TCP-friendly fashion. Abstracting from packet-level details, we assume that this results in a fair sharing of the link rate according to the PS discipline. Thus, when there are $N(u)$ elastic flows in the system at time u, the available service rate for each of the users – either elastic or streaming – is $1/(K+N(u))$. Denote by $C_1(s,t) := \int_{u=s}^{t} K/(K+N(u))\mathrm{d}u$ the total amount of service available for the streaming sessions during the time interval $[s,t]$.

We will mainly be interested in the quantity $V_1(t) := \sup_{s\leq t}\{A_1(s,t) - C_1(s,t)\}$, where $A_1(s,t)$ denotes the amount of service which ideally should be available for the streaming traffic during the interval $[s,t]$. For example, $A_1(s,t)$ may be taken as the amount of streaming traffic that would nominally be generated during the interval $[s,t]$ if there were ample bandwidth. We assume $A_1(s,t) \equiv Kr(t-s)$ (see the end of Section 3 for an extension). Thus, $V_1(t)$ may be interpreted as the shortfall in service for the streaming traffic at time t compared to what should have been available in ideal circumstances. For conciseness, we will henceforth refer to $V_1(t)$ as the *workload* of the streaming traffic at time t.

Let $C_2(s,t)$ be the amount of service available for the elastic flows during $[s,t]$. Evidently, $C_2(s,t) \geq t-s-C_1(s,t)$, with equality in case the streaming sessions always claim the full service rate available. For the elastic traffic, the latter case is equivalent to a G/G/1 PS queue with K permanent customers, accounting for the presence of the competing streaming sessions.

However, we allow for possible strict inequality in case the streaming sessions do not always consume the full service rate available, and the unused surplus is granted to the elastic class, i.e., $C_2(s,t) = t-s-B_1(s,t)$, with $B_1(s,t) \leq C_1(s,t)$ denoting the actual amount of service received by class 1 during the interval $[s,t]$. For example, when the 'workload' of the streaming sessions is zero, the actual service rate may be set to the minimum of the aggregate input rate and the total service rate available. It may be checked that the two scenarios described above provide lower and upper bounds for the general case with $t-s-C_1(s,t) \leq C_2(s,t) \leq t-s-B_1(s,t)$.

Define $\rho := \lambda\beta$ as the traffic intensity of class 2. For class 2 to be stable, we require that $\rho < 1$. For class 1 to be stable as well, we need to assume that $\rho + Kr < 1$. Here class i ($i = 1,2$) is said to be stable if the 'workload' $V_i(t)$ converges to a finite random variable V_i as $t \to \infty$. We additionally assume that $(K+1)r > 1-\rho$, which implies that the system is critically loaded in the sense that one extra streaming session – or a 'persistent' elastic flow – would cause instability. Combined, the above two assumptions give $Kr < 1-\rho < (K+1)r$. We refer to [1] for cases in which $\rho + (K+1) < r$.

3. MAIN RESULTS

In this section we present the main results of the paper. The performance of the elastic class is mainly determined by the sojourn time distribution of an elastic flow. The tail of this distribution is determined in the first part of this section. In the second part, we derive the asymptotic tail distribution of the streaming workload. Finally, we extend the workload results to the case in which the streaming users have a stochastic nominal service target.

Delay performance of the elastic flows

As mentioned above, our model shows strong resemblance with a G/G/1 PS queue with K permanent customers. For tractability, we assume in the first part that elastic flows arrive according to a Poisson process. Denote by S_2 the delay of an elastic flow. The asymptotic tail distribution of S_2 is given by the next proposition (similar delay asymptotics were obtained in [6, 9]).

PROPOSITION 1. *If $B(\cdot) \in \mathcal{R}_{-\nu}$ and $Kr < 1-\rho < (K+1)r$, then*

$$\mathbb{P}\{S_2 > x\} \sim \mathbb{P}\left\{B > \frac{(1-\rho)x}{K+1}\right\}.$$

Informally speaking, the result shows that a large delay of an elastic flow is due to a large service requirement of the flow itself. The ratio between the two quantities is simply the average service rate received by the large flow. Over the duration of the large flow, the other elastic flows receive service roughly equal to their average input rate ρ. The remaining service capacity is shared among the large elastic flow and the streaming users, each entitled to a fair share $(1-\rho)/(K+1)$. A detailed proof is given in [1].

Workload asymptotics of the streaming traffic

We now turn the attention to the workload distribution of class 1. The main result is presented in the following theorem.

THEOREM 2. *If $B(\cdot) \in \mathcal{R}_{-\nu}$ and $Kr < 1-\rho < (K+1)r$, then*

$$\mathbb{P}\{V_1 > x\} \sim \frac{\rho}{1-\rho-Kr}\mathbb{P}\left\{B^r > \frac{x\frac{1-\rho}{K+1}}{K(r-\frac{1-\rho}{K+1})}\right\}. \quad (1)$$

Next, we provide a heuristic derivation of the asymptotic behavior of $\mathbb{P}\{V_1 > x\}$. We will specifically argue that in the present context the most likely way for a large class-1 workload V_1 to occur arises from the arrival of a class-2 user with a large service requirement $B_{\rm tag}$, while the system shows average behavior otherwise. We will refer to the class-2 user as the "tagged" user.

Now, suppose that the tagged user arrives at time $-y-z_0$, with $z_0 = \frac{x}{K(r-\frac{1-\rho}{K+1})}$, and has service requirement $B_{\rm tag} \geq x + (1-\rho-Kr)(y+z_0)$, and $y \geq 0$. For class 2 to remain stable, other class-2 users together approximately require a service $\rho(y+z_0)$ during the interval $(-y-z_0, 0]$. Class 1 together with the tagged class-2 user consume what is left over and thus roughly receive $(1-\rho)(y+z_0)$. The cumulative amount of service received by the tagged user up to time 0 roughly equals either $(1-\rho)(y+z_0)/(K+1)$ or $B_{\rm tag}$, depending on whether the user is still present at time 0 or not. Class 1 is entitled to the unused capacity of class 2 and the amount of service received by class 1 can thus be estimated by $\max\{Kr(y+z_0)-x, \frac{K}{K+1}(1-\rho)(y+z_0)\}$. Moreover, since the nominal service target of class 1 equals

Kr, we deduce

$$\begin{aligned} V_1(0) &\geq Kr(y+z_0) - \max\{Kr(y+z_0) - x, \\ &\qquad \frac{K}{K+1}(1-\rho)(y+z_0)\} \\ &= \min\{x, K(r - \frac{1-\rho}{K+1})(y+z_0)\} \\ &\geq \min\{x, K(r - \frac{1-\rho}{K+1})z_0\} = x. \end{aligned}$$

Integrating with respect to y (in case of Poisson class-2 arrivals), then gives the right-hand side of Equation (1), and thus indicates that the scenario described above is sufficient for the class-1 workload to reach level x. Of course, there are alternative scenarios that could potentially lead to a large class-1 workload, but these are extremely unlikely compared to the one described above.

A rigorous proof of Theorem 2 can be found in [1] and involves lower and upper bounds that asymptotically coincide. The proof of the lower bound is based on the above heuristics. For the upper bound, we use an alternative interpretation of the most likely scenario. Note that the arrival of a class-2 user with a large service requirement also results in a large total workload after its arrival. In fact, we show in [1] that the event $V_1(-z_0)+V_2(-z_0) \geq x+(1-\rho-Kr)z_0$ corresponds to the most likely scenario described above. To see that this alternative characterization results in the same asymptotics, we use the following equivalence relation for the total workload.

Let $V_2^c(t)$ be the workload at time t in an isolated queue with service rate c fed by class 2 only, and let V_2^c be its steady-state version. We then have the following asymptotic equivalence between $V := V_1 + V_2$ and V_2^{1-Kr}:

$$\mathbb{P}\{V > x\} \sim \mathbb{P}\left\{V_2^{1-Kr} > x\right\} \sim \frac{\rho}{1-Kr-\rho}\mathbb{P}\{B^r > x\}.$$

The first asymptotic equivalence is evident, and in fact applies in a sample-path sense, if the full service rate is always used when there is any work present. However, it also holds for a critically loaded system where the unused surplus of the streaming class is lost. The second equivalence is due to Pakes [11]. Now, substituting $z_0 = \frac{x}{K(r-\frac{1-\rho}{K+1})}$ in the bound for $V_1(-z_0)+V_2(-z_0)$ and some rewriting yields Equation (1).

Finally, to obtain an upper bound, we also show that scenarios in which both $V_1(-z_0)+V_2(-z_0) < x+(1-\rho-Kr)z_0$, and $V_1(0) > x$, are extremely unlikely compared to the one described above. This involves many technical details, see [1].

Generalization to stochastic service targets

The assumption that the ideal service target for the streaming traffic is constant, is actually not crucial (see also [1]). Theorem 2 remains valid in case $A_1(s,t)$ behaves according to a general stationary process with mean $\mathbb{E}[A_1(t,t+1)] = Kr$, provided that significant deviations from the mean are sufficiently unlikely. More specifically, we require that $A_1(s,t)$ satisfies

ASSUMPTION 1. *For all $\phi > 0$, $\psi > 0$, and for $x \to \infty$,*

$$\mathbb{P}\left\{\sup_{t\geq 0}\{A_1(-t,0) - K(r+\psi)t\} > \phi x\right\} = o(\mathbb{P}\{B^r > x\}).$$

In such a scenario, the variations in the class-1 service target do not matter asymptotically, because they average out. Assumption 1 is satisfied by a wide range of traffic processes, such as instantaneous bursts and On-Off sources.

The proof of this generalization of Theorem 2 also involves lower and upper bounds. The lower bound only requires minor modifications. For the upper bound, we relate the class-1 workload $V_1^{\text{var}}(t)$ to that in a system with a constant service target. Heuristically speaking, if the service target exceeds Kr, the overshoot $A_1(s,t) - Kr(t-s)$ is postponed to moments at which the service target is below Kr. This "smoothing" can be converted into a strict sample-path relation ($\phi > 0$):

$$V_1^{\text{var}}(t) \leq V_1^{K(r+\phi)}(t) + V_1^{\text{cst}}(t),$$

where $V_1^{K(r+\phi)}(t)$ is the deficit in service in an isolated system of capacity $K(r+\phi)$, and 'var', 'cst' refer to the systems with variable and constant nominal service targets, respectively. Next, Assumption 1 may be applied to control $V_1^{K(r+\phi)}$ after which the appropriate lower bound remains.

4. REFERENCES

[1] Bekker, R., Borst, S.C., Núñez-Queija, R. (2004). Performance of TCP-friendly streaming sessions in the presence of heavy-tailed elastic flows. *Perf. Eval., Special Issue on Long Range Dependence and Heavy Tail Distributions*, to appear.

[2] Ben Fredj, S., Bonald, T., Proutière, A., Régnié, G., Roberts, J.W. (2001). Statistical bandwidth sharing: a study of congestion at the flow level. In: *Proc. SIGCOMM 2001*, 111–122.

[3] Bonald, T., Proutière, A. (2004). On performance bounds for the integration of elastic and adaptive streaming flows. In: *Proc. ACM Sigmetrics/Performance 2004*, 235–245.

[4] Crovella, M., Bestavros, A. (1996). Self-similarity in World Wide Web traffic: evidence and possible causes. In: *Proc. ACM Sigmetrics '96*, 160–169.

[5] Floyd, S., Handley, M., Padhye, J., Widmer, J. (2000). Equation-based congestion control for unicast applications. In: *Proc. ACM SIGCOMM 2000*, 43–54.

[6] Guillemin, F., Robert, Ph., Zwart, A.P. (2004). Tail asymptotics for processor sharing queues. *Adv. Appl. Prob.* **36**, 525–543.

[7] Key, P.B., Massoulié, L., Bain, A., Kelly, F.P. (2003). A network flow model for mixtures of file transfers and streaming traffic. In: *Proc. ITC-18*, 1021–1030.

[8] Núñez-Queija, R. (2000). *Processor-Sharing Models for Integrated-Services Networks.* Ph.D. Thesis, Eindhoven University of Technology.

[9] Núñez-Queija, R. (2002). Queues with equally heavy sojourn time and service requirement distributions. *Ann. Oper. Res.* **113**, 101–117.

[10] Padhye, J., Kurose, J., Towsley, D., Koodli, R. (1999). A model-based TCP-friendly rate control protocol. In: *Proc. IEEE NOSSDAV '99.*

[11] Pakes, A.G. (1975). On the tails of waiting-time distributions. *J. Appl. Prob.* **12**, 555–564.

[12] Rejaie, R., Handley, M., Estrin, D. (1999). RAP: an end-to-end rate-based congestion control mechanism for real-time streams in the Internet. In: *Proc. Infocom '99*, 1337–1346.