Modelling TCP Throughput and Fairness

D.J.Leith, R.Shorten

Hamilton Institute, NUI Maynooth, Ireland

Abstract. Since TCP traffic is elastic, a fundamental issue is the behaviour of multiple TCP flows competing for bandwidth on a shared link. Motivated by the ubiquity of drop-tail queueing in current networks, our focus in this paper is on developing analytic models suited to characterising the throughput and fairness of competing TCP flows in drop-tail environments. Building on recent ideas from the theory of positive linear systems, we obtain simple, insightful closed-form expressions for throughput and fairness. The accuracy of these expressions is confirmed in extensive simulations across a range of network conditions. In particular, they are found to provide accurate estimate of mean fairness and throughput even when flows are not synchronised.

1 Introduction

TCP traffic continues to account for the majority of traffic on the internet. Since TCP traffic is elastic, a fundamental issue is the behaviour of multiple TCP flows competing for bandwidth on a shared link. A variety of fluid models have been studied, but while they have been found to provide useful insight and predictive power for active queueing disciplines such as RED (see, for example [1], [2]), their utility for drop-tail queuing has yet to be established. Our focus in this paper is on developing analytic models that characterise the throughput and fairness of competing TCP flows in drop-tail environments. This is motivated by the ubiquity of drop-tail queueing in current networks. With drop-tail queueing and a single TCP flow, the seminal work by Padhye et al [3] establishes a simple approximate model of the mean transmit rate as a function of network parameters (round-trip time, bandwidth, drop probability etc). However, while a number of empirical studies have been reported (e.g. [4], [5]), few analytic results for multiple flows sharing a link are available. Notable exceptions include the work of Chiu and Jain [6] in an abstract setting, and more recently that of Brown [7]. The latter is of most relevance to the the present paper.

While [7] adopts a fluid approach, here we take a modelling approach based on new results in [8] using ideas from the theory of positive systems to analyse network dynamics. In this paper we extend the results in [8] to account for buffering and flows with different round trip times. This allows us to obtain simple, insightful closed-form expressions for fairness and throughput. Both the model in [8] and that used in [7] assume synchronisation. The synchronisation assumption is that all flows experience a drop when the network "pipe" becomes full. Clearly, this is an unrealistic assumption in real networks. In this paper we show that the fairness and throughput predicted under the assumption of synchronisation, while inaccurate with respect to instantaneous values, provides a rather accurate estimate of the *mean* fairness and throughput even when flows are not synchronised.

The paper is organised as follows. We begin in Section 2 by briefly considering the case of a single TCP flow. In Section 3, we present our main result, an analytic expression for the throughput efficiency of TCP flows competing for shared band-

width the accuracy of which is verified by extensive packet-level simulations. We conclude in Section 4 by summarising the results presented.

2 Preliminaries: Throughput of a Single TCP Flow in a Buffered Network

We begin by briefly considering the case of a single TCP flow. This was previously considered by Padhye et al [3] and others. Unlike this earlier work, here we explicitly take account of the influence of buffering on the throughput efficiency of a flow. The latter is defined here as the rate at which packets leave the bottleneck link normalised by the bandwidth of the bottleneck link; that is, the efficiency is 100% when the link operates continuously at its maximum bandwidth.

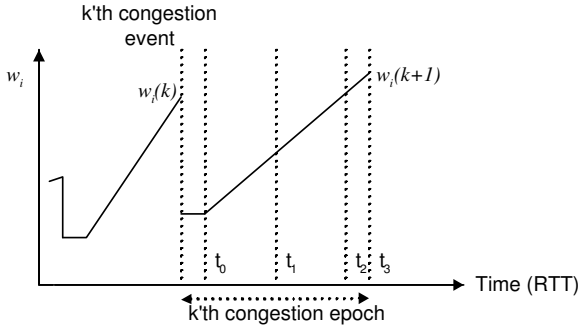

Fig. 1. Congestion window time history.

The single flow case with buffering may be analysed as follows. With reference to Figure 1, let $t_0(k)$ be the time at the beginning of the k^{th} congestion epoch, $t_1(k)$ the time when queue starts to fill, $t_2(k)$ the time when a drop occurs and $t_3(k)$ the time when the source detects the drop. We use $w_i(k)$ to denote the source TCP congestion window immediately before backoff and note that for a single flow $w_i(k)$ equals the "pipe" size $P = BT + q_{max}$, with B the link bandwidth, T the propagation delay and q_{max} the queue size. Let α, β be the AIMD increase and decrease parameters respectively, and define the provisioning parameter $\gamma = q_{max}/BT$ where B is the bandwidth of bottleneck link, q_{max} the buffer size and T is the end-to-end propagation delay. For simplicity, assume that $1/(1+\gamma) \leq \beta$ i.e. that the queue empties at backoff. We immediately have that

$$t_1 - t_0 = (1/(1+\gamma) - \beta)P/\alpha \tag{1}$$
$$t_2 - t_1 = (1 - 1/(1+\gamma))P/\alpha \tag{2}$$
$$t_2 - t_0 = (1-\beta)P/\alpha \tag{3}$$

That is,

$$(t_1 - t_0)/(t_2 - t_0) = \frac{(1/(1+\gamma) - \beta)}{(1-\beta)},\ (t_2 - t_1)/(t_2 - t_0) = \frac{(1 - 1/(1+\gamma))}{(1-\beta)} \tag{4}$$

The throughput (packets per second) averaged over one congestion epoch, y, is

$$y = 0.5(1/(1+\gamma) + \beta)(P/T)(t_1 - t_0)/(t_2 - t_0) + B(t_2 - t_1)/(t_2 - t_0) \tag{5}$$

(where, for simplicity, we have neglected the contribution from the one RTT interval between the drop time t_2 and the time t_3 that this is detected at the source). Substituting from (4) and normalising by B yields the efficiency

$$\eta = \frac{1}{(1-\beta)}[(0.5+\gamma)/(1+\gamma) - 0.5\beta^2(1+\gamma)] \tag{6}$$

where we have made use of the fact that $P/BT = 1+\gamma$ to eliminate P from the expression. The accuracy of (6) can be seen from Figure 2, where the theoretical prediction (6) is compared against packet-level simulation results from *ns-2* [9].

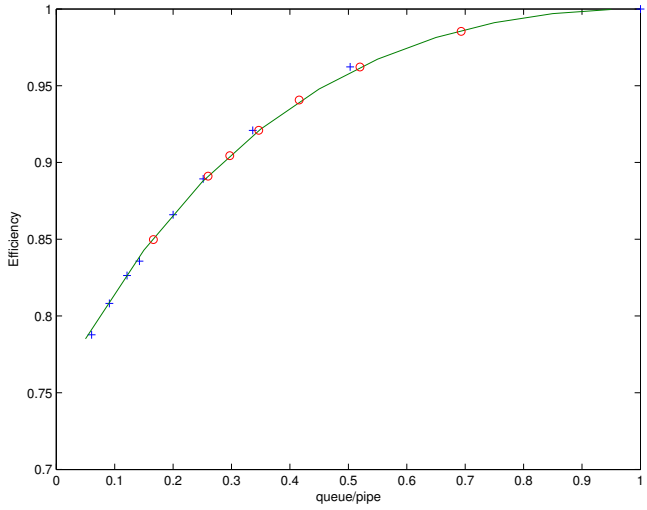

Fig. 2. Efficiency of TCP vs ratio of queue to delay-bandwidth product. The solid line is the theoretical efficiency curve given by (6). Key: + denotes simulation data as queue size varies (bandwidth 100Mb, RTT 40ms, queue is varied from 330 to 20 packets), ∘ denotes data from Figure 3.

Comment: *RTT dependence.* In (6) the efficiency is determined by the AIMD backoff factor β and the queue provisioning parameter γ. Notice that $\gamma = q_{max}/BT$ is inversely proportional to propagation delay, T. Hence, by (6), for given bandwidth B and queue size q_{max} the efficiency decreases as the propagation delay T increases. Consider the data shown in Figure 3, which plots simulation results illustrating the change in efficiency as RTT varies. Re-plotting the same data against γ rather than RTT yields the points marked by ∘ in Figure 2.

3 Fairness and Throughput of Competing TCP Flows

We extend our analysis to n flows with a shared bottleneck link as follows. Let $w_i(k)$ denote the congestion window of flow i (the total number of packets in flight) immediately before the kth backoff event, α_i denote the AIMD increase parameter of flow i (number of extra packets added to the network per second) and β denote the backoff factor (which is assumed to be the same for all flows). With knowledge of these quantities, we can apply essentially the same analysis as in the single flow

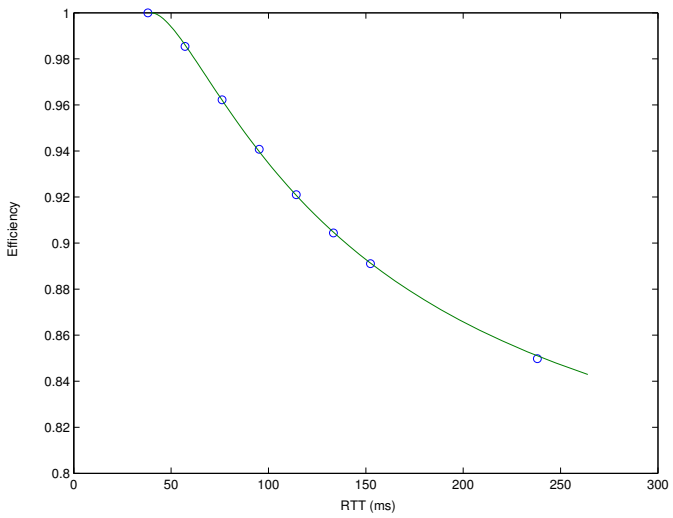

Fig. 3. Efficiency of TCP as RTT varies (bandwidth 100Mb, queue 330 packets). The solid line is the corresponding theoretical prediction (6).

case to derive an expression for the efficiency of flow i. In the single flow case, $w(k)$ is simply the pipe size. With competing flows, to determine $w_i(k)$ we need to establish how the available bandwidth is divided up between the flows.

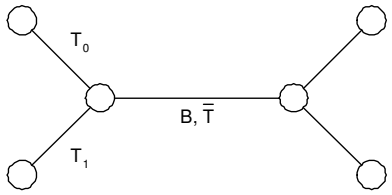

Fig. 4. Dumbell topology

3.1 Distribution of Bandwidth Between Flows

We proceed as follows. Let B denote the bandwidth of the bottleneck link, q_{max} the queue size and $\bar{T}$ the propagation delay of the bottleneck link. Let T_i denote the combined propagation delay of all other (non-congested) links traversed by flow i, see Figure 4. The queue at the bottleneck link does not begin to fill until

$$\sum_{i=1}^{n} cwnd_i(t_1)/(\bar{T} + T_i) = B \tag{7}$$

where $cwnd_i(t)$ is the instantaneous value of the congestion window of flow i at time t and t_1 is the time when the queue just begins to fill. We re-write this as

$$\sum_{i=1}^{n} \overline{cwnd}_i(t_1) = B\bar{T} \tag{8}$$

with $\overline{cwnd}_i(t) = [\bar{T}/(\bar{T}+T_i)]cwnd_i(t)$; we might interpret $\overline{cwnd}_i(t)$ as the number of packets in flight in the congested link and $B\bar{T}$ is the delay-bandwidth product of this link. The pipe then becomes full at time t_2 after an additional q_{max} packets are added,

$$\sum_{i=1}^{n} \bar{cwnd}_i(t_2) = B\bar{T} + \bar{q}_{max} \tag{9}$$

where $\bar{q}_{max} = [\sum_{i=1}^n \bar{\alpha}_i / \sum_{i=1}^n \alpha_i] q_{max}$ and $\bar{\alpha}_i = [\bar{T}/(\bar{T}+T_i)]\alpha_i$. Observe that $\bar{\alpha}_i$ can be interpreted as the number of extra packets added to the *bottleneck* link by flow i every second.

Similarly to the single flow case, let $t_0^i(k)$ be the start of the k^{th} congestion epoch for flow i, $t_2^i(k)$ be the time at which a drop occurs and $t_3^i(k)$ the time at which the source detects the loss. Let $w_i(k)$ denote the congestion window size of flow i immediately before backoff in the kth congestion epoch . Source i detects this packet loss one RTT, i.e. $T_i + \bar{T} + q_{max}/B$, later at time $t_3^i(k)$. Assuming for the moment (this assumption will be dropped later) that drops are synchronised, i.e. $t_2^i(k) = t_2(k) \forall i$, and neglecting the impact of RTT variation due to queueing,

$$\bar{w}_i(k+1) = \beta \bar{w}_i(k) + \bar{\alpha}_i(t_3^i(k+1) - t_0^i(k+1)) \tag{10}$$

and

$$t_0^i(k+1) = t_2(k) + 2RTT_i \tag{11}$$

$$t_2(k+1) = t_2(k) + 2\tau + \frac{\sum_{i=1}^n (1-\beta)\bar{w}_i(t_3^i(k)) - \sum_{i=1}^n \bar{\alpha}_i(2\tau - RTT_i)}{\sum_{i=1}^n \bar{\alpha}_i} \tag{12}$$

$$t_3^i(k+1) = t_2(k+1) + RTT_i \tag{13}$$

where $RTT_i = T_i + \bar{T} + q_{max}/B$, $\tau = \max_i(T_i + \bar{T} + q_{max}/B)$ and we have made use of

$$\bar{P} = B\bar{T} + \bar{q}_{max} = \sum_{i=1}^{n} \bar{w}_i(k) - \bar{\alpha}_i RTT_i \tag{14}$$

Hence,

$$t_3^i(k+1) - t_0^i(k+1) = \frac{\sum_{i=1}^n (1-\beta)\bar{w}_i(k)}{\sum_{i=1}^n \bar{\alpha}_i} + \epsilon \tag{15}$$

where $\epsilon = (2\tau - RTT_i) - \frac{\sum_{i=1}^n \bar{\alpha}_i(2\tau - RTT_i)}{\sum_{i=1}^n \bar{\alpha}_i}$. The bottleneck link population of packets therefore evolves according to

$$\bar{w}_i(k+1) = \beta \bar{w}_i(k) + \frac{\bar{\alpha}_i}{\sum_{i=1}^n \bar{\alpha}_i}[\sum_{i=1}^{n}(1-\beta)\bar{w}_i(k)] + \bar{\alpha}_i \epsilon \tag{16}$$

When ϵ is sufficiently small that it may be neglected (e.g. when the congestion epoch duration is sufficiently long), the dynamics (16) constitute a positive linear system and the analysis from [8] may be immediately applied. In particular, the fixed point is determined by the Perron eigenvector and we have $\bar{w}_i = [\bar{\alpha}_i / \sum_{j=1}^n \bar{\alpha}_j]\bar{P}$. Using $\alpha_i = 1/(\bar{T} + T_i)$, then we have that in steady-state

$$\bar{w}_i = \frac{(\bar{T}+T_i)^2}{\sum_{j=1}^n (\bar{T}+T_j)^2}\bar{P} \tag{17}$$

3.2 Throughput of Flow i

As before, define $t_1^i(k)$ to be the time during the kth congestion epoch when the queue just begins to fill. We have that

$$(t_1^i - t_0^i)/(t_2^i - t_0^i) = \frac{(1/(1+\bar{\gamma}) - \beta)}{(1-\beta)},\ (t_2^i - t_1^i)/(t_2^i - t_0^i) = \frac{(1 - 1/(1+\bar{\gamma}))}{(1-\beta)} \quad (18)$$

where $\bar{\gamma}$ is the provisioning parameter of the bottleneck link,

$$\bar{\gamma} = \bar{q}_{max}/B\bar{T} \quad (19)$$

The throughput (packets per second) of flow i averaged over one congestion epoch is

$$y_i = 0.5(\beta + \frac{1}{1+\bar{\gamma}})\frac{\bar{w}_i}{\bar{T}}\frac{(t_1^i - t_0^i)}{(t_2^i - t_0^i)} + \frac{1}{1+\bar{\gamma}}\frac{\bar{w}_i}{\bar{T}}\frac{(t_2^i - t_1^i)}{(t_2^i - t_0^i)} \quad (20)$$

Hence, the efficiency is

$$\eta_i = \frac{1}{(1-\beta)(1+\bar{\gamma})}\frac{\bar{w}_i}{B\bar{T}}[\frac{0.5+\gamma}{1+\bar{\gamma}} - 0.5\beta^2(1+\gamma)] \quad (21)$$

Observing that $B\bar{T} = (1+\bar{\gamma})\bar{P}$ and substituting from (17) for $\bar{w}_i/B\bar{T}$ yields

$$\eta_i = \frac{\phi_i}{(1-\beta)}[\frac{0.5+\bar{\gamma}}{1+\bar{\gamma}} - 0.5\beta^2(1+\bar{\gamma})] \quad (22)$$

where $\phi_i = \frac{(\bar{T}+T_i)^2}{\sum_{j=1}^{n}(\bar{T}+T_j)^2}$. It can be seen that this is simply the throughput expression (6) scaled by ϕ_i.

Comments

(i) *Fairness.* The factor ϕ_i captures the unfairness that can be introduced when AIMD flows with different effective increase and decrease factors compete for shared bandwidth. It is interesting to note that ϕ_i is quadratic in round-trip time and so the unfairness arising from competition between flows with different round-trip times is indicated as potentially much larger than that observed for isolated flows with different round-trip times (in which case throughput scales approximately linearly with RTT).

(ii) *Overall efficiency.* Since $\sum_{i=1}^{n}\phi_i = 1$, we have from (22) that the overall efficiency $\eta = \sum_{i=1}^{n}\eta_i$ achieved by multiple competing flows is identical to the efficiency (6) for a single flow, with γ replaced by $\bar{\gamma}$.

3.3 Comparison with Packet-Level Simulation Results

The foregoing analysis is based on a number of simplifying assumptions. In this section we compare the efficiency predicted using (22) with results obtained from *ns-2* packet-level simulations for a range of network conditions.

Our analysis predicts that with multiple competing TCP flows the throughput efficiency of flow i is linear in ϕ_i (for given bottleneck buffer provisioning $\bar{\gamma}$). Figure 5 presents *ns-2* simulation results for competing TCP flows. The data is for two flows , one flow with round-trip time fixed at 22ms and the other with round-trip varied from 22ms to 272ms in 20ms steps. In addition, a small amount (less than 1% of the bottleneck link bandwidth) of background web traffic was included to better

capture realistic network conditions and to reduce artefacts associated with phase effects [10] . For each value of round-trip time, a total of 1000s of data was collected. The throughput binned over 100s intervals plotted in Figure 5. It can be seen that the data from the packet-level simulation does, indeed, exhibit a near linear dependence on ϕ_i. It is important to note that while the analytic expression (6) is derived under the assumption that the flows experience synchronised packet drops, it is readily verified that in these simulation results this assumption is significantly violated. See, for example, the congestion window time histories in Figure 6. The lack of synchronisation is reflected in the spread of throughput values observed.

The effect of changing the queue size is to alter $\bar{\gamma}$ thereby, according to (22), altering the slope of the line relating throughput to ϕ_i. Once again this analytic prediction is supported by simulation results: see Figure 7. These results are not specific to the network parameters used: see, for example, Figure 8.

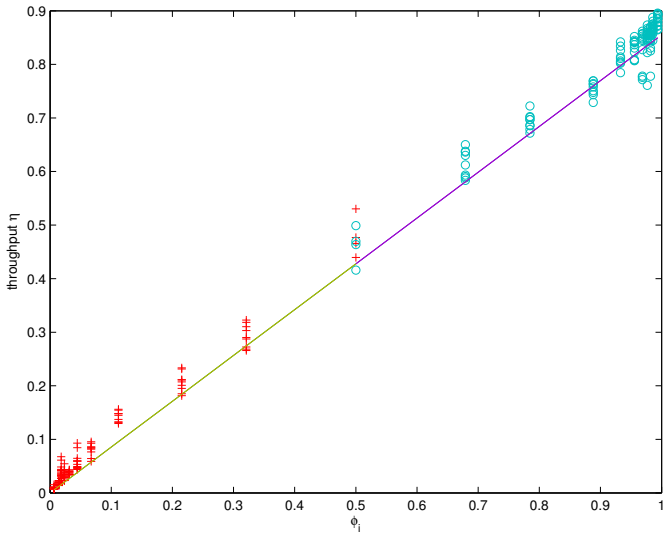

Fig. 5. Throughput efficiency η as RTT varies (B=100Mb, $\bar{T}$=20ms, T_1=2ms, T_2=2-252ms, queue 80 packets). Key: ∘ denotes data for flow 1, + data for flow 2; the solid line is the analytic prediction given by (22)

4 Concluding Remarks

Since TCP traffic is elastic, a fundamental issue is the behaviour of multiple TCP flows competing for bandwidth on a shared link. Motivated by the ubiquity of drop-tail queueing in current networks, our focus in this paper is on developing analytic models suited to characterising the throughput and fairness of competing TCP flows in drop-tail environments. Our approach builds on recent results for synchronised networks [8], extending these earlier results to account for buffering and flows with different round trip times. This allows us to obtain simple, insightful closed-form expressions for fairness and throughput. The synchronisation assumption is that all flows experience a drop when the network "pipe" becomes full. Clearly, this is an unrealistic assumption in most real networks. Nevertheless, we find that the fairness and throughput predictions, while inaccurate with respect to instantaneous values,

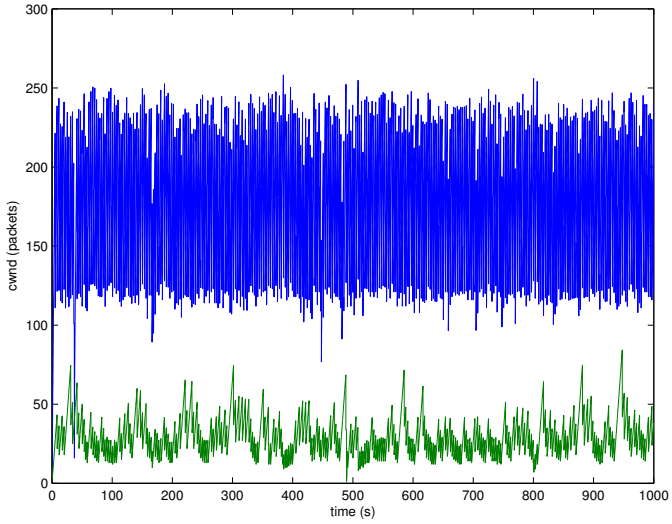

Fig. 6. Typical congestion window time histories of TCP (B=100Mb, $\bar{T}$=20ms, T_1=2ms, T_2=162ms, queue 80 packets).

provide a rather accurate estimate of the *mean* fairness and throughput even when flows are not synchronised.

5 Acknowledgements

This work was supported by Science Foundation Ireland grant 00/PI.1/C067. This work was also partially supported by the European Union funded research training network *Multi-Agent Control*, HPRN-CT-1999-00107 and by the Enterprise Ireland grant SC/2000/084/Y.

References

1. E. Altman, C. Barakat, E. Laborde, P. Brown, and D. Collange, "Fairness analysis of tcp/ip," in *Proceedings IEEE Conference on Decision and Control, Sydney*, 2000.
2. S. Low, F. Paganini, and J. Doyle, "Internet congestion control," *IEEE Control Systems Magazine*, vol. 32, no. 1, pp. 28–43, 2002.
3. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling tcp throughput: A simple model and its empirical validation," in *Proceedings SIGCOMM'98*, 1998.
4. R. Yang, M. Kim, and S. Lam, "Transient behaviors of tcp-friendly congestion control protocols," in *Proceedings INFOCOM 2001*, 2001.
5. S. Floyd, M. Handley, and J. Padhye, "A comparison of equation-based and aimd congestion control," in *Int. Computer Science Institute Technical Report*, 2000.
6. D. Chiu and R. Jain, "Analysis of the increase/decrease algorithms for congestion avoidance in computer networks," *Journal of Computer Networks*, vol. 17, no. 1, pp. 1–14, 1989.
7. P. Brown, "Resource sharing of tcp connections with different round trip times," in *Proceedings INFOCOMM 2000*, 2000.
8. R. Shorten, D. Leith, J. Foy, and R. Kilduff, "Towards an analysis and design framework for congestion control in communication networks," in *Proceedings of the 12th Yale Workshop on Adaptive and Learning Systems*, 2003.

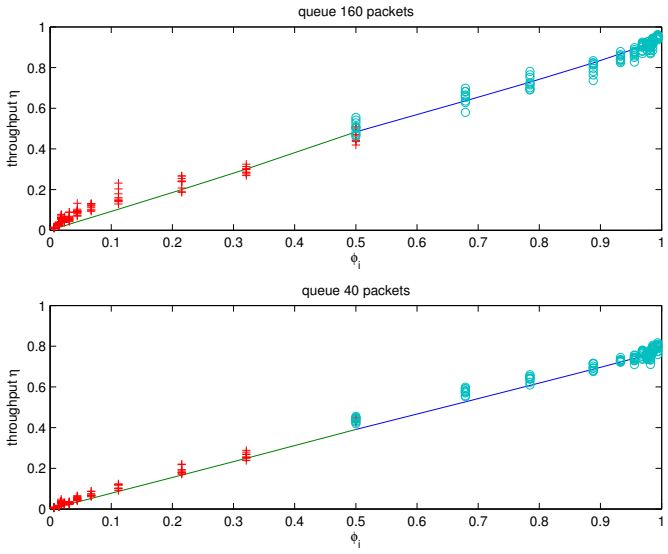

Fig. 7. Throughput efficiency as RTT and queue varies (B=100Mb, $\bar{T}$=20ms, T_1=2ms, T_2=2-252ms in 20ms steps, queue 40/160 packets). Key: ∘ denotes data for flow 1, + data for flow 2; the solid line is the analytic prediction given by (22)

9. "Network simulator, http://www.isi.edu/nsnam/ns/," tech. rep., University of California, Berkeley.
10. S. Floyd and V. Jacobson, "Traffic phase effects in packet-switched gateways," *ACM SIGCOMM Computer Communication Review*, vol. 21, no. 2, pp. 26–42, 1991.

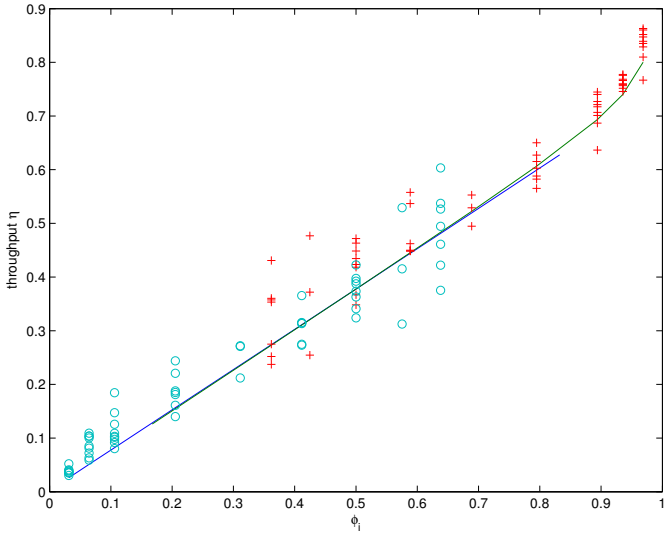

Fig. 8. Throughput efficiency as RTT varies (B=100Mb, $\bar{T}$=20ms, T_1=100ms, T_2=2-252ms in 20ms steps, queue 80 packets). Key: ∘ denotes data for flow 1, + data for flow 2; the solid line is the analytic prediction given by (22)