Estimating Traffic Intensities at Different Nodes in Networks via a Probing Stream

Vinod Sharma
Electrical Communication Engineering Dept.
Indian Institute Of Science, Bangalore-560012, India
email: vinod@ece.iisc.ernet.in, Fax: (+)91-80-3340563

Abstract. In future it will be increasingly important for the customers and the service providers to make measurements in the Internet to ensure the QoS required by various services. But not everyone will have access to all the information available at the various network components (e.g. at the routers) for such measurements. Thus it is important to devise new methods to estimate traffic intensity at various nodes based on partial information available to a user or a service provider (say from a subnetwork not owned by it). The measurement tools available in Internet, for example, the PING probe and the TRACEROUTE (available in various refinements) have been used by users to obtain the connectivity and some congestion information in the Internet. In this paper using these tools we suggest and compare different estimators to estimate aggregate traffic intensities at various nodes in the network. One can use this information for selection of routes as well as for admission control in the Integrated and Differentiated Services Architectures in the Internet and also in a Connection Admission Controller in ATM networks. Our method works for general stationary traffic streams as well as for general network topologies.We also show the convergence of these estimators and provide simulation results.

Key words: Network monitoring, estimation of queues, decentralized estimation.

1. Introduction

It is becoming increasingly important to provide widely different Quality of Service(QoS) to users in the Internet as well as in the other high speed networks. Customers will be required to pay more for better services. This will ensure that the users will also be interested in knowing whether they are actually receiving better QoS for the extra money (see Ferguson and Huston[8], Chapter 2 for various such issues). Also if the service provider charges according to the congestion pricing (see various proposals for pricing in Mcknight and Bailey[13]) the customers will have to pay more for transporting data during congestion period. Then the users will be interested in knowing the congestion in the different places in the network at any particular time. Similarly a network service provider is interested in knowing the congestion at different places. If a subnet is not owned by the service provider then he may not have available the detailed information about the number of packets passing through an output link of a router in that subnet.

Currently many groups are working on developing tools and infrastructure for the measurement of traffic intensity and available bandwidth and obtaining connectivity information in the Internet (see Almes[1], [5], Paxson et.al.[14]). A general survey of the various tools available is in [5]. Also, various related Internet drafts and RFCs can be obtained from IETF working groups on Network and Real Time Flow measurements (see e.g.[14]). We complement these efforts by suggesting techniques for estimating the aggregate traffic intensities at different routers and nodes in the network which can be used on the data available from these tools. Although we will make specific comments on how to use our estimates from data obtained from a PING or a TRACEROUTE probing stream ([5]), our techniques can be used with other tools also. Our techniques are discussed in the idealistic setting so that the problems of obtaining accurate measurements in the network are isolated from the inherent limitations of the accuracy of the estimators themselves.

Another scenario where our estimation procedure can be used is in the measurement based admission control in the Internet and ATM networks. The main quantity to estimate for making the admission control decision is the aggregate traffic intensity at different routers or switches on the path (see Sharma and Mazumdar[18], Gelembe et.al.[9]). A particularly pertinent class of admission control algorithms is in Turanyi et al.[19] where they use the aggregate traffic intensity to make admission decisions.

There are various proposals on the use of probing streams for QoS routing in the Internet(see Chen and Nahrstedt

0-7803-5796-5/99/$10.00 © 1999 IEEE

[6]).Our estimation procedures should be useful in this context also.

Now we provide our model and the estimation problem in more specific terms (called analytical framework in [14]). Initially we consider a queue with two input Poisson streams with rates λ_1 and λ_2 and iid service times with the distributions of $s(1)$ and $s(2)$ (the model and the probabilistic assumptions will be considerably generalized below). Stream 1 is called a "test stream"(or the probe stream) which is sent in the queue to estimate λ_2. Depending upon the delays experienced by stream 1, we try to estimate λ_2.

Sharma and Mazumdar [18] suggested several algorithms for estimation in such a scenario. They also considered the more general case of general stationary ergodic arrivals and tandem and product-from queueing networks. However, all the systems considered had infinite buffers at each queue. The case of finite buffers is obviously of practical interest. We address such systems in this paper. As against the algorithms in [18], our present method works for infinite and finite buffer nonproduct form general networks also. In addition, for our present method the "test stream" can be arbitrary. This will allow us to use this method for the test stream generated by the PING probe or by TRACEROUTE.

Some other relevant literature on estimating parameters in queuing systems is Bhat and Subba Rao[3], Pickand III and Stine[ ] and Daley and Servi[ ]. The problems and the methods of estimation in these studies are necessarily different from ours.

In section 2 we study a single queue. Section 3 considers a tandem of such queues. In section 4 we consider nonproduct form general networks. Simulation results are provided in section 5.

2. Single Queue

In this section we consider a single queue with two Poisson arrival streams, (this assumption will be removed later) each with iid general service requirement. The stream 1 is a test stream with rate λ_1, which will be used to "monitor" or estimate the system state (as mentioned below). Stream 2 has arrival rate λ_2. A generic service time of stream i is $s(i), i = 1, 2$. We assume a FCFS discipline. The queue can store upto M packets (in addition to the packet in service). If any packet arrives to a full buffer, it is lost. The rate λ_1 and the packet lengths of stream 1 are known. The aim is to estimate λ_2 given the sojourn times of stream 1 in the system. In the following we will consider different scenarios and then mention extra assumptions for the system.

Consider the situation where we know when a stream 1 arrival occurs, when it departs from the queue (may be via an immediate acknowledgment) and its packet lengths. If a customer (packet) of stream 1 is lost, its ack is not received and hence we know which packets are lost. The packets of stream 2 are not observable. This scenario can be abstracted to varying degrees of accuracy from the various measurement tools in Internet. This happens for example, if we use PING or TRACEROUTE except that instead of the departure times from the queue, we only know the round trip time (the time for the packet to reach the destination and for the ack to return to the source). We will assume that the time taken by the ack is known (it happens if its queueing delays are negligible and its route is known). By putting time stamps on the packets (and synchronizing the clocks) one can obtain the time taken by the packet alone (in the forward path). From the available information, we can calculate the fraction of packets lost from stream 1. One can show that if $E[(s(i))^\alpha] < \infty$, $\alpha \geq 1, i = 1, 2$ then the regeneration length τ (the arrival epochs of stream 1 finding the system empty can be taken as the regeneration epochs) satisfies $E[\tau^\alpha] < \infty$. Then from the law of large numbers (LLN) for the regenerative processes (Asmussen[2]) we obtain that the fraction of packets lost provides a consistent estimate of $P(1)$, the stationary probability of loss of packets of stream 1. From the Central Limit Theorem (CLT) for regenerative processes (Asmussen[2], Sharma[16]) it is also asymptotically normal if $E[(s(i))^2] < \infty$ for $i = 1, 2$. From Sharma[16] we also obtain other rates of convergence.

By PASTA, the stationary probability $P(2)$ of packet loss of type 2, equals $P(1)$ and hence we have a consistent estimate of it also. Furthermore from the available information, we know the packets of stream 1 which find the system empty (because we know their sojourn times and service times). Again by PASTA, the fraction of such packets gives a consistent estimate of $1 - \rho$ where ρ is the total traffic intensity *entering* the system, and

$$\rho = \lambda_2(1 - P(2))E[s(2)] + \lambda_1(1 - P(1))E[s(1)]. \quad (1)$$

Therefore, we obtain a consistent estimate of λ_2 if we know $E[s(2)]$ (otherwise we get a consistent estimate of $\lambda_2 E[s(2)]$, which being the workload arrival intensity of stream 2 is anyway more relevant). Asymptotic normality of the estimate of λ_2 is also obtained if $E[(s(i))^2] < \infty, i = 1, 2$. This is because, from the CLT for regenerative processes any linear combination of the above estimates of ρ and $P(i)$ is asymptotically normal and λ_2 is a continuously differentiable function of $P(1), \rho$. Now use the Cramer-Wold device to obtain the asymptotic normality of the estimate of λ_2. We also obtain the rate of convergence from Sharma[16].

Actually the consistency of the above estimator holds even when $\{(s_n(1), s_n(2))\}$ is strictly stationary and ergodic. For asymptotic normality stronger conditions will be required.

In a practical network scenario knowing the total time a packet takes on its path, we can find if it experiences queueing delay as follows. The total time a packet takes on its path

includes the transmission time, the propagation delays on the links and the processing times. By sending many packets on a route over a certain time the minimum time taken by these packets can approximate all these delays without the queueing delay. Knowing this we can compute the number of packets which did not face the queueing delays. Similar comments will hold for our estimation techniques later on also but we will not repeat them.

If the service times of both the streams are exponential with the same known rate μ, then, we can use the above provided estimate of probability of loss in the formula

$$(1-\hat{\rho})\hat{\rho}^{M+1}/(1-\hat{\rho}^{M+2}) \qquad (2)$$

to obtain the estimate of $\hat{\rho}$ the total traffic intensity arriving at the system $(=(\lambda_1+\lambda_2)/\mu)$ and hence then of λ_2. For general service times such a formula is not available and hence we used the above procedure. However, interestingly, (1) provides a more efficient estimator even for M/M/1/k case (also see the comments below).

Next we consider the algorithm we provided in section 2.2 of Sharma and Mazumdar [18]. Now the stream 1 can have arbitrary arrival epoches as well as service times. The stream 2 remains Poisson but the service times of stream 2 form an ergodic stationary sequence. Information about the arrival epochs and departure epochs of stream 1 enables one to know when two consecutive packets of stream 1 are in the system at the same time. For the infinite buffer case, the difference of their departure epochs minus the service time of the second packet of such a consecutive pair gives the amount of work of stream 2 entering the queue during the interarrival time of this pair. But for the finite buffer queue it is not true because there is a possibility of loss of some of the packets of stream 2 that arrived during this time. Thus we suggest a modification of this scheme. The modification provides only an approximate estimate but the advantage with respect to the first scheme (via (1)) provided above is that, we can extend it to the case where the second stream is a general stationary, ergodic process.

We send the packets of stream 1 at regular intervals $n\Delta t$, where Δt is a small appropriate constant. Now consider the number of packets of stream 2 which arrive between a consecutive pair of stream 1 which are in the system at the same time. Since interval Δt is small, there is a good probability that if a packet of stream 1 enters the queue then the next one will also enter. The probability that there is no arrival in stream 2 in time Δt is $e^{-\Delta t\lambda_2}$. When Δt is small, this probability is large. We estimate this probability by the fraction of consecutive pairs of stream 1 which are in the queue at the same time, and find no type 2 packet between them (this will be known by looking at the arrival epochs, the departure epochs and the service times of type 1) to the total number of such pairs. This estimate is an approximation of $e^{-\Delta t\lambda_2}$ (and hence provides an approximate estimate of λ_2) because the following event has been ignored. On arrival, a type 1 packet enters the queue, then during next Δt time enough packets of type 2 arrive such that some of them overflow, then a departure takes place and then the type 1 packet arrives Δt time later. When Δt is small, this event will be very rare and hence we can have a good approximation of λ_2. The smaller the Δt, the better is the approximation. However, then the test stream will load the queue more and more making it heavily loaded. This will not make the estimation procedure invalid but it may not be desirable in a network. Thus actually we will modify the scheme as follows. The $2n$th packet of stream one is generated at time $nT, T>0$ and the $(2n+1)$st packet is generated at time $nT+\Delta t$ where $0<\Delta t<T$. We can take Δt small and T to be a suitably large constant. Then we can consider only the pairs sent at times $nT, nT+\Delta t$ and take the fraction of those pairs with no arrival between them to the total number of pairs which are both admitted to the queue. There is one more undesirable effect of very small Δt. Denoting $e^{-\Delta t\lambda_2}$ by p, if it has an estimation error of Δp then this leads to an error of $\Delta p/\Delta tp$ in the estimate of λ_2. Our simulation experience in section 5 shows that this requires that Δt should be carefully chosen for the case when there are small buffers and the ρ is large. For larger buffers (≥ 10) the estimate is not that sensitive to Δt. This observation is valid for the other systems described below where this scheme has been used. Observe that in this scheme we did not require the knowledge of $E[s(2)]$. The asymptotic normality of this estimator will be obtained if the service times of each stream are iid and $E[(s(i))^2]<\infty, i=1,2$. Similarly, other rates of convergence are obtained by using the fact that if $E[(s(i))^\alpha]<\infty$, $\alpha\geq 1, i=1,2$ then $E[\tau^\alpha]<\infty$.

As mentioned above, this procedure can be generalized to the case when the stream 2 is a general time stationary ergodic sequence. Then the probability of the event that in a small interval Δt, no arrival comes is $1-\lambda_2\Delta t+o(\Delta t)$ and that of the event that one batch arrival comes is $\lambda_2\Delta t+o(\Delta t)$. Ignoring $o(\Delta t)$, the above procedure will provide an estimate of $1-\lambda_2\Delta t$ (and hence of λ_2). Again the above comments regarding the size of Δt vs the accuracy of the estimate hold.

In the next section we extend the above approximation scheme to a tandem of queues and in section 4 to nonproduct form networks.

Now we compare the last scheme with the estimate provided via (1) and (2). The first scheme has to take samples when the stream 1 sees the system full and when it sees the system empty. One of these events can be somewhat rare in many situations (e.g., in heavy traffic case, the empty epochs will be rare and in the light traffic case overflow epochs will be rare). However, the last scheme can be formed so that

more frequent events are sampled. In addition the simulation experience (see also the justification below) in section 5 shows that the approximate estimate can be quite accurate. But whenever the estimator based on (1) is applicable, it seems to be more efficient (based on our simulation results) than the last estimate. Next we compare the estimates based on (1) and (2). The estimate in (2) is based on only estimating the probability of packet loss for stream 1. As just commented, if the ρ is small and the buffer size is somewhat large, then this event will be rare and estimating it will not be very efficient (for the estimator via (1), under such conditions estimating ρ will be more efficient and it seems to compensate the loss in efficiency in estimating $P(1)$). If the event is not rare then this should provide a good estimate (although it is valid only for an M/M/1/k system). Our simulation experiments indicate that indeed in most of our experiments, the estimate based on (1) out performs that based on (2). This is a useful observation because (1) provides the estimate under more general conditions.

From the practical point of view, in the networks, the robustness of the estimators is also an important issue. By taking derivative of $\hat{\rho}$ in (2) with respect to the probability of loss, we observe that a small error in the estimate of this probability leads to several orders of magnitude increase in error in the estimate of ρ even for a small buffer length of 10 (the error increases with the buffer length). However for the estimates from (1) or from the last scheme, the estimation error for λ_2 is of the same order of magnitude as of the primary quantity estimated. Therefore both of these estimators are much superior to (2) even though the error in estimation of fraction of pairs without any other traffic in between them should have more estimation error.

3. Tandem of Queues

We first consider a tandem of two queues. The test stream is stream 1. Stream 2 enters queue 1 and after service leaves the system (this assumption is not required for our algorithms). Stream 3 enters queue 2 and leaves the system after service. Streams 2 and 3 are Poisson processes (this assumption will be removed later) but their service times can be arbitrary iid sequences. Both the queues have finite buffers. If a packet arrives at a queue and the buffer is full, the packet leaves the system. We will know the exact arrival epochs, service times (in both queues) and the departure epochs of stream 1 from *the system*. The aim is to estimate λ_2 and λ_3.

Our comments at the beginning of section 2 regarding the applicability of these assumptions if PING or TRACEROUTE probe streams are used, remain valid for this system as well as the system considered in the next section.

First send the test stream only through the first queue (after service from queue 1, the test stream leaves the system). Then we know the exact departure times of stream 1 from queue 1. Now the methods of section 2 can be used to estimate λ_2. Next pass the test stream through both the queues 1 and 2. Now the departure instants of stream 1 from queue 1 are not available to us; only the departure instants from queue 2. Because of this the methods in section 2.2 and 2.3 in Sharma and Mazumdar [18] could not be used for tandem queues. Now we provide a modification of that procedure which provides exact (asymptotic) estimates for the infinite buffer queue but an approximation for the finite buffer system. For the scheme to be presented actually the PING probe stream is not appropriate but the TRACEROUTE can be used. The reason is that in case of TRACEROUTE one can send the probe stream upto different hops of a route. Also one knows the exact path followed by a packet. The actual path followed by a PING packet is not available to us – only the destination node.

We take the test stream to be a batch arrival process where each batch is of size N_1, the inter-batch distance is arbitrary and the service times of the customers of type 1 in each queue are constants Δt (actually as will be clear from the last paragraph of this section, instead of taking a batch of packets arriving at exactly one instant, one could take these packets to arrive at regular intervals– the modifications required are as in the last paragraph). Since the test stream is under our control, we can choose $N_1, \Delta t$ and inter-batch distances appropriately. The packets of a batch in queue 1 will occupy consecutive positions (whichever are admitted to the queue). Thus, interdeparture times of those customers from queue 1 will be Δt. Some (or none) of these will be admitted to queue 2. When they depart queue 2, we will know if there was any packet of type 3 in between those packets. This is because when there is no type 3 packet between two consecutive packets of a batch, their interdeparture distance from queue 2 will be Δt. Thus as in the last section, from a sample of such packets we can estimate $e^{-\Delta t \lambda_3}$ (and hence λ_3) via the fraction of the admitted pairs with no type 3 packet between them to the total number of the pairs admitted to queue2. This will be an exact estimate for the infinite buffer case. In that case actually we don't need to consider only the pairs with no arrival of type 3, but could consider all the packets of a batch and estimate $\lambda_3 \Delta t E[s(3)]$ – this will be more efficient (although our simulation experiments did not show any noticeable improvement in efficiency). For the finite buffer queue the estimate of $e^{-\Delta t \lambda_3}$ will be approximate (as for a single queue) and the approximation will improve as Δt decreases (although after a while, because of the sensitivity effect on the estimation error, mentioned in the last section, the estimate will actually start becoming worse). The consistency of the (approximate) estimator is obtained by the strong law of large numbers (SLLN). For asymptotic normality, we further assume that the service times of streams 2 and 3 are iid and $E[(s(i))^2] < \infty, i = 2, 3$. Then one can show (for instance

by the results in Sharma [17]) that the regeneration length τ (the regeneration epochs can be taken as the arrival epochs of type 1 to the system which find the system empty) satisfy $E[\tau^2] < \infty$. Now the asymptotic normality can be obtained as in section 3. If stream 2 also passes through queue 2 then again the algorithm works in the same way because there will be no packet of stream 2 within packets of a batch of stream 1.

This method can be used when the arrival streams 2 and 3 are general stationary ergodic. Then λ_2 is estimated as in section 3. For stream 3 also, we can estimate $1 - \lambda_3 \Delta t$ in the same way. These comments will hold for a tandem of n queues studied below. This allows us to consider this tandem of queues to be the queues on the route of the virtual circuit of a connection in a network because all the other traffic on a node could be represented by the general stationary ergodic stream.

The above procedure can be extended to more than two queues. The approximation will *not* become worse as we estimate the arrival rate at a queue at a later stage in the network. Thus for a queue at stage n, $n \geq 2$ we will pass the test stream through the first n queues only. Hence we will know exactly when the packets of stream 1 depart from queue n. Since in each queue the service times of stream 1 are Δt, we will know, after looking at their departure instants between which packets from an original batch of stream 1 there were no arrivals from streams $2, \ldots, n+1$ (then their interdeparture times will be Δt). For the infinite buffer case, we can get from this a consistent estimate of $e^{-\Delta t(\lambda_3 + \ldots + \lambda_{n+1})}$ while for the finite buffer case an approximate estimate. Since we have already estimated $\lambda_3, \ldots, \lambda_n$, now we will have an estimate of λ_{n+1}. The consistency and asymptotic normality of the estimate is obtained as for the two queues case.

4. Queueing Networks

We can use the method of section 3 for networks with feed-back also. We illustrate by a few examples. The overall scenario is as in section 3. We know the arrival epochs as well as the departure epochs of stream 1 from the system. In addition we know the service times (which will be fixed ,Δt in this section) of packets of sream 1. Based on this information, we want to estimate the arrival rates of other traffic streams entering the system. In section 3 we did not need the queueing system to be stable (ergodic). However, now we will assume stability with the test stream in the network because, otherwise the traffic rate we may be estimating, may not exist. We will not prove the ergodicity of the systems. In literature, under various conditions, stability of such networks is available. Sharma and Mazumdar [18] provided an algorithm to estimate traffic intensities under this scenario but only for the product-form, infinite buffer systems. The algorithm in the present section is valid for nonproduct-form and finite buffer systems.

First consider two queues, $Q1$ and $Q2$ with stream 2 a stationary ergodic sequence (when entering the system) with rate λ_2. The stream 2 follows the route $Q2 \rightarrow Q1 \rightarrow Q2$. The service times of stream 2 in both the queues are assumed to form stationary ergodic sequences. We pass the test stream through $Q1$ and then it leaves the network. The $2n$th customer of the test stream, $n \geq 1$ will enter the queue at times $nT, T > 0$ and $(2n+1)st$ customer at time $nT + \Delta t$. The constants $\Delta t, T$ are suitably chosen, Δt a small constant and T a relatively large constant $\Delta t < T$. The service times of the test stream can be arbitrary (say iid or constant and such that total traffic intensity ρ at queue 1 is less than 1 in case of infinite buffer at $Q1$). It is assumed that the traffic stream 2 entering Q1 is asymptotically stationary, ergodic. By observing the arrival times and the departure times of the test stream through the system, we will know in between which pairs $(2n, 2n+1)$ of the test stream there are arrivals of stream 2. Now the estimation procedure of λ_2 remains as in section 3. Asymptotic stationarity of stream 2 entering Q1 provides the consistency of the estimate. For the estimate to be good Δt should be small. For the finite buffer case the approximation will be worse than for the infinite buffer system. A slight modification of this scheme can take care of probabilistic routing also.

Now we consider a network of three queues Q1, Q2, Q3. Stream 2 has the routing $Q2 \rightarrow Q1 \rightarrow Q2$ and stream 3 has routing $Q3 \rightarrow Q2 \rightarrow Q1$. We assume that the arrival streams 2 and 3 are stationary, ergodic and their service times in each queue are also stationary and ergodic. We first pass the test stream only through Q1 and then it leaves the system. Then we can estimate $\lambda_2 + \lambda_3$ by the schemes mentioned earlier. Next we send the test stream through the route $Q1 \rightarrow Q3$. The test stream is generated at times $\{nT, n \geq 1\}$ with a batch of size $m > 1$ at each such epoch. The service times of this stream in each queue will be Δt. With probability $1 - \lambda_3 \Delta t$, the packets within a batch will be separated by Δt in time from each other at departure times from Q3. This will provide us an estimate of λ_3. Now will also obtain an estimate of λ_2. We will provide the simulation results for this system in section 5.

5. Simulation Experiments

In this section we describe our simulation experience. We are interested in obtaining information about the following questions: (i) In case of the M/M/1/k system, which is the most efficient estimator in section 2. (ii) How sensitive to Δt is the estimator based on $e^{\Delta t \lambda}$ and how good an approximation for the finite buffer queue. (iii) Goodness of estimators under different traffic loads. (iv) Sensitivity of approximation $1 - \Delta t \lambda$ to Δt and the underlying distributions. (v) Goodness of estimates for queueing networks.

In the following we provide our findings on the above is-

sues based on our simulation data.

The tables 1-4 contain the simulation results for a single queue. In table 1, the estimates are based on a sample size (of test stream) 10,000. This was done to show the difference in performance of the three estimators for the case of an M/M/1/k system. With a sample size of 20,000 all estimators perform very well and hence, their performance cannot be compared. For the estimators via equations (1) and (2), the test stream was a Poisson process while for the third estimator the arrivals come at times $\{nT, nT + \Delta t, n \geq 0\}$. A * in the table indicates that for that particular case no packet of stream 1 was lost and hence formula (2) would give no estimator. For the estimates it did provide, we observe that for small ρ and large buffers its quality is worse than the other estimators (in fact even for $\rho = 0.9$ and buffer size 15 its performance is the worst). For larger ρ and smaller buffer sizes the approximation $e^{-\Delta t\lambda}$ is not very accurate and hence we observe that under these conditions this method provides worse performance than via (1) and at times also worse than via (2). Overall, the estimator via (1) provides the most accurate results for all ρ and buffer sizes. We also observed that the sensitivity to Δt in general was large for buffer size five and $\rho = 0.9$ but not so much for other values. Furthermore, the sensitivity was minimal for the infinite buffer case.

Next consider table 2. Now the system has an infinite buffer size. We have the option of estimating $\lambda_2 E[s(2)]\Delta t$ or $e^{-\Delta t\lambda_2}$. We observe that the estimation via $\lambda_2 E[s(2)]\Delta t$ is more accurate, as expected (for $\lambda_2 E[s(2)]\Delta t$, we did not need Δt to be small). Also, the number of samples required for the infinite buffer case is much more than the finite buffer case because the regeneration lengths for the infinite buffer case can be much longer (which makes the samples more correlated and hence the rate of convergence of the estimates much slower).

In the above two cases we also simulated the system with deterministic service times (not reported in the paper) and obtained similar conclusions. Below we will report some results with deterministic service for tandem queues.

Table 3 provides results when the inter-arrival times are a sum of two exponentials. As expected errors are large when the ρ is large and the buffer size is small. Table 4 contains the results when the interarrival time is a mixture of two exponentials. In this case errors are more than other cases (e.g. Table 3) even when the sample size is larger. In fact the estimate was also more sensitive to Δt. One of the reasons we require larger sample size in tables 3 and 4 is that the estimate requires the arrival stream to become time stationary (we started with an arrival at time 0).

In tables 5-7 we provide simulation results for a tandem of three queues. Except the test stream, the others are Poisson processes with λ_i rate in queue $i - 1, i = 2, 3, 4$. The service times are iid with distributions exponential or deterministic. We want to estimate all these λ_i. In the first queue the estimate is observed by any of the above methods (we used only the third method here). Then the test stream was sent through Q1, and Q2 and an estimate of λ_3 was obtained. Next the test stream was passed through the queues Q1-Q3. The test stream was generated with batches of 5 packets, Δt was taken between 0.1 and 0.01. The sample size of the test stream was 50,000. The buffer sizes were 10, 15, 20. We observe that both the estimates $\hat{\lambda}_3, \hat{\lambda}_4$ of λ_3 and λ_4 are quite accurate in all the cases.

In table 8 we provide the results for the three queue network described in section 4. Each queue has a finite buffer and the service times are deterministic. Streams 2 and 3 are Poisson and follow the routes $Q2 \rightarrow Q1 \rightarrow Q2$ and $Q3 \rightarrow Q2 \rightarrow Q1$ respectively. In the table ρ_i denotes the total traffic intensity at Q_i. We first sent the stream 1 only through Q1 and estimated $\lambda_2 + \lambda_3$. Then we sent the stream 1 through the route $Q1 \rightarrow Q3$ and estimated λ_3. This time the test stream was generated at Q1 with batch sizes of 5. The buffer sizes considered were 10, 15 and 20. The rates $\lambda_2 + \lambda_3$ were estimated via an estimate of $1 - (\lambda_2 + \lambda_3)$ while λ_3 at Q3 was estimated via $e^{-\lambda_3 \Delta t}$. Thus, we see that the estimate of λ_3 is generally better than that of λ_2. Also, to obtain a proper estimate of $\lambda_2 + \lambda_3$ at Q1, the network needed to be in stationarity. Thus, we had to take a lot more sample (100,000) to obtain a reasonable estimate. We also observe that at higher ρ, the estimate of λ_2 is worse than at lower ρ. This is partly because in general for higher ρ the estimation is harder and also because the approximation by $1 - (\lambda_2 + \lambda_3)\Delta t$ will be worse. At higher ρ we also found that the estimates were more sensitive to Δt.

Acknowledgements The simulation results have been obtained by my student Karthikeyan Balaji.

References

1 G. Almes, Metrics and infrastructure for IP performance, http://io.advanced.org/csg-ippm/, Sept. 1997.
2 S. Asmussen, Applied Probabilty and Queues, John Wiley and Sons,Chichester,1987.
3 U.N. Bhat and S. Subba Rao, Statistical analysis of queueing systems, Queueing Systems, Vol.1, 1987, 217-247.
4 CAIDA measurement tool taxonomy, http://www.caida.org/tools/meastools:html#traceroute
5 S. Chen and K. Nahrstedt, IEEE Networks, Nov./ Dec. 1998, 64-79.
6 D.A. Daley and L.D. Servi, Exploiting Markov chains to infer queue length from transactional data, J. Appl. Prob., Vol. 29, 1992, 713-732.
7 P. Ferguson and G. Huston, Quality of Service, J. Wiley, N.Y.,1998.
8 E. Gelembe, X Mang and R. Onvural, IEEE Communications Magazine, Vol. 35, 1997, 122-129.

9 L. W. Mcknight and J.P. Bailey, Internet Economics, MIT Press, Cambridge,1998.

10 V. Paxson, G. Almes, J. Mahdavi and M. Mathis, Internet RFC 2330, 1998.

11 J. Pickands III and R.A. Stine, Biometrika, Vol.84, 1997, 295-308.

12 V. Sharma, Reliable estimation via simlation, Queueing Systems, Vol.19, 1995, 169-192.

13 V. Sharma, Some limit theorems for regenerative queues, Queueing Systems, Vol.30, 1998, 341-363.

14 V. Sharma and R. Mazumdar, Estimating Traffic parameters in queueing systems with local information, Performance Evaluation, Vol. 32, 1998, 217-230.

15 Z. Turanyi, A. Veres and A. Olah, in Performance of Information and Communication Systems, Ed. U. Korner and A. Nilsson, Chapman and Hall,1998.

ρ	Buffer size	λ_2	Estimate $\hat{\lambda}_2$		
			via (2)	via (1)	via $e^{-\Delta t\lambda}$
0.3	5	2.000	2.162	1.977	2.031
0.3	10	2.000	*	1.940	2.010
0.3	15	2.000	*	2.016	1.958
0.6	5	4.000	4.036	4.0355	3.828
0.6	10	4.000	3.944	3.945	3.916
0.6	15	4.000	*	4.054	3.888
0.9	5	6.000	5.923	5.915	5.435
0.9	10	6.000	6.182	6.048	5.978
0.9	15	6.000	6.278	6.091	5.981

Table 1: Single queue, exp. arrivals, exp. service, finite buffers, $\Delta t = 0.01$, sample 10,000.

ρ	λ_2	$\hat{\lambda}_2$	
		via estimating $\lambda_2 E[s(2)]\Delta t$	via $e^{-\Delta t\lambda}$
0.3	2.0	2.004	1.980
0.6	4.0	3.993	3.806
0.9	6.0	5.988	5.962

Table 2: Single Queue, exp. arrivals, infinite buffers, exp. service, $\Delta t = 0.003$, sample 100,000.

ρ	Buffer size	λ_2	$\hat{\lambda}_2$
0.3	5	2.667	2.667
0.3	10	2.667	2.570
0.3	15	2.667	2.641
0.6	5	3.333	3.072
0.6	10	3.333	3.264
0.6	15	3.333	3.359
0.9	5	4.000	3.505
0.9	10	4.000	3.879
0.9	15	4.000	4.0556

Table 3: Single queue, sum of exp. arrivals, finite buffers, $\Delta t = .02, T = 1.00$, sample 30,000.

ρ	Buffer size	λ_2	$\hat{\lambda}_2$
0.3	5	6.154	6.02
0.3	10	6.154	6.30
0.3	15	6.154	6.47
0.6	5	7.692	6.88
0.6	10	7.692	7.61
0.6	15	7.692	7.65
0.9	5	9.231	7.88
0.9	10	9.231	8.54
0.9	15	9.231	8.71

Table 4: Single queue, mixt. of exp. finite buffers, $\Delta t = .005, T = 1.0$, sample 50,000.

ρ	Buf. size	λ_3	λ_4	$\hat{\lambda}_3$	$\hat{\lambda}_4$
0.65	10	4.000	4.000	3.967	3.918
0.65	15	4.000	4.000	3.949	4.048
0.65	20	4.000	4.000	4.143	3.970
0.85	10	6.000	6.000	6.001	5.822
0.85	15	6.000	6.000	5.844	5.845
0.85	20	6.000	6.000	6.0125	6.106

Table 5: Tandem queues, exp. services, finite buf.

ρ	Buf. size	λ_3	λ_4	$\hat{\lambda}_3$	$\hat{\lambda}_4$
0.65	10	4.000	4.000	3.9675	3.948
0.65	15	4.000	4.000	3.946	4.042
0.65	20	4.000	4.000	3.981	3.969
0.85	10	6.000	6.000	6.071	5.897
0.85	15	6.000	6.000	5.953	6.088
0.85	20	6.000	6.000	6.001	5.964

Table 6: Tandem queues, deter. service, finite buf.

ρ	λ_3	λ_4	$\hat{\lambda}_3$	$\hat{\lambda}_4$	Service type
0.65	4.000	4.000	4.125	3.995	Exponential
0.85	6.000	6.000	5.965	5.914	
0.65	4.000	4.000	3.989	4.013	Deterministic
0.85	6.000	6.000	6.060	6.044	

Table 7: Tandem queues, infinite buffer, exponential/deterministic service.

ρ_1	ρ_2	ρ_3	Buf	λ_2	λ_3	$\hat{\lambda}_2$	$\hat{\lambda}_3$
0.425	0.40	0.325	10	2.0	4.0	2.094	3.998
0.425	0.40	0.325	15	2.0	4.0	2.075	3.917
0.425	0.40	0.325	20	2.0	4.0	2.093	3.992
0.675	0.70	0.525	10	3.0	8.0	3.088	7.980
0.675	0.70	0.525	15	3.0	8.0	3.067	7.964
0.675	0.70	0.525	20	3.0	8.0	3.183	7.978
0.850	0.90	0.650	10	4.0	10.0	3.997	10.002
0.850	0.90	0.650	15	4.0	10.0	4.186	10.068
0.850	0.90	0.650	20	4.0	10.0	4.588	9.990

Table 8: Queueing Network, finite buf., deter. service, sample 100,000.