An Approximate Calculation of Max-Min Fair Throughputs for Non-Persistent Elastic Sessions

Pinaki Shankar Chanda, Anurag Kumar, Arzad Alam Kherani
Dept. of Electrical Communication Engg.
Indian Institute of Science, Bangalore, 560012, INDIA
Submitted to Internet Performance Symposium, Globecom 2001.
`email: pinaki, anurag, alam@ece.iisc.ernet.in`

Abstract— **The general problem we consider is the analysis of a traffic engineering model in which there are several routes in a network, on each route elastic sessions arrive randomly according to some arrival process, and they need to transfer a finite volume of data sampled from some distribution. We are interested in computing a measure of average session throughput on each route, for a given bandwidth sharing mechanism. Such a model would be useful for the purpose of network planning and for setting up explicit routed paths.**

In particular, in this paper, we assume Poisson session arrivals on each route, the session volumes are fluid and arbitrarily distributed. At each instant the network shares the bandwidth among the ongoing sessions according to the max-min fair bandwidth sharing mechanism, i.e., *Instantaneous Max-Min Fair* **(IMMF) share mechanism is used. The measure of performance we consider is the time average bandwidth obtained by a session on each route. We propose a heuristic algorithm for obtaining an approximation for this performance measure for arbitrary routes in an arbitrary network topology. Simulations with different network topologies are used to evaluate the proposal. In spite of its simplicity, we find that the approximation works quite well in a variety of topologies that we have studied.**

I. Introduction

Present day packet networks predominantly carry elastic traffic, and hence the rates of flows being carried at any time can be adjusted according to the level of network congestion. In any packet networking technology there are distributed traffic control algorithms for dynamic bandwidth sharing among the elastic flows carried by the network. In ATM networks, elastic flows are carried on the ABR (available bit rate) service and the bandwidth sharing objective is to achieve max-min fairness. In the Internet there is no explicit attempt to achieve a specific bandwidth sharing objective. Based on the way TCP works, recent research (see [1], [2], [3]) has suggested models for bandwidth sharing in the Internet.

Traffic engineering in networks requires effective models that can predict the performance seen by the end users. It is clear that elastic flows transfer finite volumes of data, and are hence non-persistent, or *ephemeral*. The important performance index for elastic sessions is the average throughput of sessions belonging to a particular route. This measure is dependent on the (random) number of ongoing transfers that are taking place on the links shared by those connections. Most studies related to throughput analysis ([4], [5], [6], [7]) have assumed *persistent sessions*, i.e., the sessions have an infinite amount of data to send. In reality new sessions start, transmit some finite number of packets and then close their connections. Hence the consideration of a session arrival process and random transfer volumes in modeling elastic flows in a network is important.

In the work presented here, we consider an arbitrary network topology with fixed routes. On each route, we assume that session arrival epochs constitute a Poisson process. Each session needs to transfer a random volume of data, and the sequence of transfer volumes are independent and identically distributed. We assume that *instantaneous* max-min fair (IMMF) sharing is achieved in the network; i.e., at each instant of time the rate obtained by a session is its max-min fair share given the ongoing sessions. We adopt the average bandwidth share of sessions on different routes in the network as our performance index. In this framework, our contribution is an algorithm to approximately calculate the average bandwidth share of sessions on each route. Simulations for different network topologies have been performed to evaluate this proposal.

In an internet, bandwidth sharing is achieved by TCP's end-to-end adaptive window mechanism. Recent research has suggested that TCP's AIMD rate control, based on packet-loss feedback, is equivalent to a bandwidth sharing objective called F_A^h fairness [2]. It has been shown in [8] that TCP Vegas achieves proportional fairness. In some simple situations (simple topologies or a single bottleneck for each route) proportional fairness is equivalent to max-min fairness. Further, TCP can be viewed as a whole class of adaptive window based algorithms, and the bandwidth sharing that is achieved depends on the network feedback and source adaptation. With this point of view, it can be seen that with appropriate feedback from routers TCP can actually achieve max-min fairness ([9]). Each link computes its MMF parameter η_l, $l \in L$ (set of all links), (see [10]) and marks packets with probability $e^{-\alpha\eta_l}$, for some $\alpha > 0$. For large α it follows that the marking probability along route s is approximately $e^{-\alpha \min_{l \in L_s} \eta_l}$, where L_s is the set of all links carrying route s (see also [11]). Thus each

Submitted to Internet Performance Symposium, Globecom 2001. This work was supported by a research grant from HFCL (Himanchal Futuristic Communications Limited).

Address all correspondence to Anurag Kumar; anurag@ece.iisc.ernet.in.

source effectively learns $\min_{l \in L_s} \eta_l$ (its fair rate), measures the round trip time along the path and adjusts its window accordingly (see also [12]).

Some prior literature related to our work is the following. The importance of traffic engineering models with non-persistent flows has been highlighted in [13]. In [14] the authors consider the same stochastic model as we do, and for exponentially distributed transfer volumes they provide an ergodicity condition for the process of the number of sessions on each route. The fairness objectives considered are: max-min fairness, weighted max-min fairness and proportional fairness. In this paper some simulation results for session delays have been provided for simple networks. These results are generalised in [3] to a broad class of fairness objectives, of which max-min fairness is a special case. Computational results are provided for some simple topologies. In [15] a bandwidth sharing objective called "min bandwidth sharing" is introduced, and a technique is studied for performance analysis with this policy in large networks with certain structure. The papers [14], [3], and [15] all make the assumption that the bandwidth shared adjust instantaneously as sessions arrive and depart. This can be expected to be a good approximation to the actual network behaviour when the delay-bandwidth product is small relative to the transfer volumes. There are now a few papers on performance models of TCP controlled bandwidth sharing, with non-persistent flows, in the Internet. Simple single link networks are studied via product form queueing models in [16] and [17]. A detailed model for an additive-increase and multiplicative-decrease congestion control protocol, with session arrivals and departures is provided in [18]. An iterative technique for networks is proposed in [19].

The difficulty of exact computation of performance in our model has been recognised in [20] and [21]. In the latter thesis some approximations are provided for simple topologies. In the work reported here we provide a heuristic algorithm, motivated by an exact algorithm for a single link. In spite of its simplicity, we find that in a variety of arbitrary, and even random, network topologies considered, the algorithm produces a good estimate of the average bandwidth share of sessions on each route.

The paper is organised as follows. In Section II we describe the traffic engineering model and assumptions. In Section III we discuss various performance measure of throughput of ephemeral sessions. Section IV contains the proposed heuristic algorithm to estimate throughput of non-persistent flows in different routes of an arbitrary network. In Section V we present the comparison of throughput predicted by this algorithm and that obtained from simulations for different network topologies.

II. Traffic Model and Notation Used

A. Traffic Model

We consider a backbone packet network of a service provider or an enterprise, and assume that this backbone network is the bottleneck for the traffic it carries. The users connected to this backbone initiate sessions and transfer random volumes of data over the backbone network over different routes. Thus the backbone network sees a random arrival process of transfer requests at each of its edge routers. Each request causes the transfer of a random volume of data. We depict this scenario in Figure 1. We assume that the traffic for

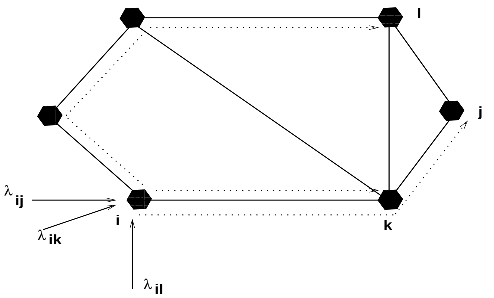

Fig. 1. A backbone network showing arrival processes of requests for transferring random volumes of data; the dotted lines show the fixed routes between (i,j), (i,k) and (i,l).

each source and destination pair uses a fixed route. For i and j being two edge nodes of the network, $\lambda_{i,j}$ is the rate of arrival of transfer requests from i to j. In later parts of this work we assume that traffic flow in each route is bi-directional, i.e., it offers the same load on the links in both forward and reverse direction, i.e., $\lambda_{i,j}$ denotes the arrival rate of transfer requests from i to j as well as from j to i. The assumption can be relaxed by considering each link of the network as two unidirectional links, i.e., thinking of the network as a directed graph and, while calculating average bandwidth share of a route, considering the load offered by routes sharing links with this route in the same direction of flow. It requires a simple modification to the proposed algorithm.

We make the following modeling assumptions: session arrival epochs on each route constitute a Poisson process; each session needs to transfer a random volume of data with some distribution; the sequence of transfer volumes are independent and identically distributed and they are independent of the arrival process. The Poisson approximation for the arrival process is reasonable ([22], [23]) when we consider the superposition of arrivals from a large number of independent users. We assume that at every instant of time the ongoing sessions on various routes of the network are using the bandwidth of the network in the max-min fair share fashion, i.e., Instantaneous Max-min fairness (IMMF) is used. Note that for the case of a single link, IMMF reduces to bandwidth sharing in the Processor Sharing manner. We consider this model and address the problem of computing a measure of average throughput of sessions on different routes.

Under the assumption of exponential filesize distribution and Poisson arrival of file transfer requests, Theorem 3.1 of [14] gives the necessary condition for existance of the stationary distribution of random vector representing the number of sessions belonging to different routes of the network.

B. Notation Used

In a network with R fixed routes and a fixed set of links L ($|L|$ denote total number of links) we use the following notations.

$\Re$	: Set of all routes in the network; $\lvert\Re\rvert =: R$
λ_r	: Rate of the Poisson process modeling the session arrival epochs in route $r \in \Re$.
$E(V_r)$	: Mean of the distribution modeling the session file sizes in route $r \in \Re$.
C_j	: The capacity of the link $j \in L$.
ρ_r^j	: The normalised load on link j due to arrival process in route r that uses link j defined as, $\rho_r^j = \frac{\lambda_r E(V_r)}{C_j}$
$N_r(t); r \in \Re$	: Number of active sessions in route r at time t.
$(N_1, N_2 ... N_R)$	: Stationary random vector for $\{N_r(t) : r \in \Re\}$
$\pi(n_1, n_2 ... n_R)$	: Stationary distribution of $(N_1, N_2 ... N_R)$.

In the case of a single link with single class (or single route) traffic, we use this notation without the subscript and superscript. This notation also applies to a single link with multiclass (or multi-route) traffic.

III. The Performance Measure

A. Single Class Traffic on a Link

On a single link the IMMF bandwidth sharing reduces to an M/G/1 processor sharing model. There are different measures of session throughput. An important performance measure is *file transfer throughput*, i.e., the total volume of the transfer divided by the file sojourn time in the system avearged over all the files transferred [24]. Let V_k be the service requirement of the kth session, and W_k the sojourn time of the kth session. The *average session throughput* is defined as

$$\tau := \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \frac{V_k}{W_k} \tag{1}$$

For $\rho < 1$ $(V_k, W_k) \to (V, W)$ in distribution, where (V, W) is the stationary random vector of the service time and the sojourn time in the system. It is easy to see that, $\tau \overset{a.s.}{=} E(\frac{V}{W})$. While this measure of throughput may be an appropriate measure, it is difficult to compute analytically [25].

One measure of per session throughput is mean volume of transfer divided by mean sojourn time of a session in the system. Thus it is defined as

$$\phi := \frac{EV}{EW} \tag{2}$$

For the $M/G/1$ PS queue this reduces to $\phi = 1-\rho$. This measure is analytically tractable, but is only a loose lower bound to τ. Yet another measure is the time averaged bandwidth share per ongoing session, conditioned on there being at least one session active. This was introduced in [21] and is defined as,

$$\sigma := \lim_{t\to\infty} \frac{\int_0^t \frac{C}{N(u)} I_{\{N(u)>0\}} du}{\int_0^t I_{\{N(u)>0\}} du} \tag{3}$$

$$\text{where } \mathrm{I}_{\{\mathrm{N(u)>0}\}} = \begin{cases} 1 & \text{if } N(u) > 0 \\ 0 & \text{otherwise} \end{cases}$$

σ has been found to approximate τ well on a single link while being computationally tractable [21]. For $\rho < 1$ it is easy to see that,

$$\sigma = \frac{\sum_{n=1}^{\infty} C \frac{\pi(n)}{n}}{\rho} \tag{4}$$

For the M/G/1 PS model $\pi(n) = (1-\rho)\rho^n$ and hence,

$$\frac{\sigma}{C} = \frac{1-\rho}{\rho} \ln \frac{1}{1-\rho} \tag{5}$$

Thus σ is easily computable and since π is insensitive to the distribution of V, so is σ.

B. Multi-Class Traffic on a Link

When multiple routes share a link we view the link as being shared by multiple traffic classes. We extend the above defination to multiple traffic classes sharing a link(as shown in Figure 2).

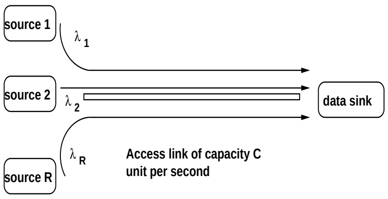

Fig. 2. R classes of traffic stream sharing a single link

Define, $n = n_1 + n_2 + ... + n_R$. The steady state distribution of $(n_1, n_2 .... n_R)$ is given by [26]

$$\pi(n_1, n_2 .... n_R) = (1-\rho) n! \prod_{r=1}^{R} \frac{\rho_r^{n_r}}{n_r!} \tag{6}$$

where $\rho = \sum_{i=1}^{R} \rho_i$
The time average per session bandwidth share of class r, i.e.,

$\sigma_r; r \in \Re$, is defined as the time averaged rate per ongoing session, conditioned on there being at least one class r active session. Hence,

$$\sigma_r \quad := \quad \lim_{t\to\infty} \frac{\int_0^t \frac{C}{N(u)} I_{\{N_r(u)>0\}} du}{\int_0^t I_{\{N_r(u)>0\}} du} \tag{7}$$

where $N(u) = \sum_{r\in\Re} N_r(u)$, i.e., the total number of sessions active at time u on this link. Note the difference between σ_r and σ as defined earlier for single class traffic. For $\rho < 1$,

$$\sigma_r \quad = \quad \frac{\sum_{(n_1,n_2,\ldots.n_R):n_r>0} \frac{C}{(n_1+n_2+\ldots.+n_R)} \pi(n_1, n_2 \ldots . n_R)}{\sum_{(n_1,n_2,\ldots.n_R):n_r>0} \pi(n_1, n_2, \ldots, n_R)}$$

and we can easily simplify this to (details are given in [27]),

$$\sigma_r \quad = \quad C\frac{(1-\rho)(1-\bar{\rho_r})}{\rho_r} \ln(\frac{1-\bar{\rho_r}}{1-\rho}) \tag{8}$$

$$\tag{9}$$

where $\bar{\rho}_r = \sum_{\forall i\in\Re:i\neq r} \rho_i$
Note that when there is only one class of traffic, $\bar{\rho_1} = 0, \rho_1 = \rho$ and Equation 8 yields $\sigma_1 = C\frac{1-\rho}{\rho} \ln(\frac{1}{1-\rho})$, which is the same as σ, as expected.

IV. Approximately Calculating σ_r in a Network

We define the per session bandwidth share of a route in a network as the time averaged IMMF rate per on-going session belonging to a particular route, conditioned on there being at least one session active on that route. Denoting this by σ_r, we see that

$$\sigma_r \quad := \quad \lim_{t\to\infty} \frac{\int_0^t x_r(u) I_{\{n_r(u)>0\}} du}{\int_0^t I_{\{n_r(u)>0\}} du} \tag{10}$$

where $x_r(u)$ is the max-min fair rate of as session in route r in the network at time u. To get a closed form expression for σ_r in a network we need to know the stationary distribution of the number of sessions in different routes. This is a difficult problem to solve ([15], [20], [21]). We, therefore, propose a heuristic algorithm to obtain $\sigma_r, r \in \Re$. For a single link with multi-class traffic we can obtain a recursive calculation of σ_r as shown in the next subsection. The algorithm for the network is motivated by this recursive algorithm for a single link.

A. *Obtaining σ_r Recursively on a Link*

Interestingly, a simple recursive procedure works for computing $\sigma_k, 1 \leq k \leq R$ when there are R routes on a single link each with Poisson arrival of sessions with random volume of data to be transferred, i.e., for the multi-class $M/G/1$ PS model.

Let, $\sigma_k(C; \rho_1, \rho_2, \ldots.\rho_R)$ denote σ_k with total link capacity C and normalized load ρ_j of route j, $1 \leq j \leq R$. As shown above,

$$\begin{aligned} &\sigma_k(C; \rho_1, \rho_2, \ldots.\rho_R) \\ &\quad = \quad C\frac{(1-\rho)(1-\bar{\rho_k})}{\rho_k} \ln(\frac{1-\bar{\rho_k}}{1-\rho}) \end{aligned} \tag{11}$$

with $\bar{\rho_k} = \sum_{i=1:i\neq k}^{R} \rho_i$.

Theorem IV.1: Consider a given indexing of the R routes. Then for $k \in \{1, 2 \ldots R\}$
$\sigma_k(C; \rho_1, \rho_2, \ldots.\rho_R) = \sigma_k(C(1 - \sum_{i=1}^{k-1} \rho_i); \rho_k, \rho_{k+1} \ldots \rho_R)$
□
The proof of this theorem involves trivial checking (see [27]). The above theorem implies the following recursive approach for calculating $\sigma_k, 1 \leq k \leq R$, on a single link, paralleling the approach for persistent sessions.

B. *Algorithm on a Single Link*

Fix an indexing of the route $k, 1 \leq k \leq R$. Then compute $\sigma_1 = \sigma_1(C; \rho_1, \rho_2, \ldots.\rho_R)$. The average bandwidth share of the sessions in route 1 is given by σ_1. The total bandwidth utilised by route 1 is $C\rho_1$ (unlike persistent session it does not depend on the bandwidth share of the other routes). Hence compute $\sigma_2 = \sigma_2(C(1-\rho_1); \rho_2, \ldots.\rho_R)$. The average bandwidth share of the session in route 2 is given by σ_2. The total bandwidth utilised by route 1 and route 2 is $C(\rho_1 + \rho_2)$. Hence compute $\sigma_3 = \sigma_3(C(1-\rho_1-\rho_2); \rho_3, \ldots.\rho_R)$. Thus at each step the link capacity is reduced by the amount taken up by the routes examined until that step and the bandwidth share of the remaining routes is computed assuming that the remaining routes share the reduced link capacity.

C. *The Proposed Algorithm for Approximate Calculation of σ_r in a Network*

Motivated by the above recursive algorithm for obtaining average bandwidth share of different classes of traffic over a single link, we propose a heuristic algorithm for computing the average bandwidth shares $\sigma_r, 1 \leq r \leq R$ in a network. Here we assume that each route represents two identical traffic streams in both forward and reverse directions over the links on which this route flows (see Subsection II-A for discussion of this assumption made here for convenience). In this iterative algorithm we determine average bandwidth share of a single route in each iteration. We remove this route from the routing topology and adjust the capacity of the links carrying this route. Hence on every itearation the route-incidence matrix and the remaining capacity of the links change.

C.1 Additional Notation Used

$R^{(k)}$: Total number of remaining routes in the network in iteration k. Initially, $R^{(0)} = R$.
$\gamma^{(k)}$: ith element of this vector is the remaining capacity of link $i \in L$ of the network (say $\gamma_i^{(k)}$)

in iteration k. $\gamma^{(k)} = [\gamma_i^{(k)}]_{(1\times|L|)}$ Initially, $\gamma_i^{(0)} = C_i, \ \ \forall i \in L$

$A^{(k)}$: Route incidence matrix in iteration k. $A^{(k)} = [a_{ij}^{(k)}]_{(|\Re|\times|L|)}$ where,

$$a_{ij}^{(k)} = \begin{cases} 1 & \text{if route } i \text{ is crossing link } j \\ 0 & \text{otherwise} \end{cases}$$

ψ : the rth element of this vector is the approximate to the avearage bandwidth share σ_r of route $r \in \Re$ in the network.

$\rho_i^{j(k)}$: Load of link j due to route i normalized to remaining link capacity in iteration k. $\rho_i^{j(k)} = \frac{\lambda_i E(V_i)}{\gamma_j^{(k)}}, \ i \in \Re, \ j \in L$ such that $\mathrm{a}_{\mathrm{ij}}^{(\mathrm{k})} = 1$

$\sigma_i^{j(k)}$: Average bandwidth share of route i on link j considering link j alone in iteration k.

C.2 Algorithm

We take a route r in the network and consider the set of links carrying this route. We calculate σ_r considering each link separately (e.g., for route r on link l it is σ_r^l). This computation is repeated for all the routes. We now take the minimum of all $\sigma_r^l \ \forall r \in \Re, \forall l \in L$. We assign this minimum value as the average bandwidth share of the corresponding route. Now, this route is removed from the routing topology and the capacities of all the links that carry this route is reduced by $\lambda_r EV_r$. The above procedure is repeated to find the average bandwidth share of all the routes. We describe the steps of the iterative algorithm as follows:

Step (1): We consider each link of the topology separately and calculate $\sigma_i^{j(0)}$. Thus in this step while calculating the average bandwidth share we ignore the presence of all other links. Hence, $\sigma_i^{j(0)}$ i.e., average bandwidth share of route i over link j when link j is considered alone is given as:

$$\sigma_i^{j(0)} = \gamma_j^{(0)} \frac{(1 - \sum_{r=1:a_{rj}^{(0)}\neq 0}^{R} \rho_r^{j(0)})(1 - \sum_{r=1:r\neq i, a_{rj}^{(0)}\neq 0}^{R} \rho_r^{j(0)})}{\rho_i^{j(0)}} \ln\left(\frac{1 - \sum_{r=1:r\neq i, a_{rj}^{(0)}\neq 0}^{R} \rho_r^{j(0)}}{1 - \sum_{r=1:a_{rj}^{(0)}\neq 0}^{R} \rho_r^{j(0)}}\right) \quad (12)$$

Step(2): We find

$$\sigma_n^{m(0)} = min(\sigma_i^{j(0)}, \ \ \forall i \in \Re \, \forall j \in L \ such \ that \ a_{ij}^{(0)} = 1)$$

We say that route n is bottlenecked in link m in some average sense. Throughput of this route is determined by link m. We assign it to $\psi_n \ i.e$ the average bandwidth share of nth route.

$$\psi_n = \sigma_n^{m(0)}$$

Step (3): This route n is now removed from the network. Necessary changes are made in the link incidence matrix for this and a new link-incidence matrix $A^{(1)}$ is formed.

$$a_{ij}^{(1)} = \begin{cases} a_{ij}^{(0)} & \forall \text{ i} \in \Re \text{ and } \forall \text{ j} \in L \text{ such that i} \neq \text{n} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

We are thus making all entries of the n th column of the link incidence matrix equal to zero to account for the route n which has been removed.

$$\text{And,} \ \ \mathrm{R}^{(1)} = \mathrm{R}^{(0)} - 1$$

We reduce the capacity of each links in the network over which this route n flows by an amount of $\lambda_n E(V_n)$. Thus, $\gamma^{(1)} = \{\gamma_j^{(1)}; j \in L\}$ such that

$$\gamma_j^{(1)} = \begin{cases} \gamma_j^{(0)} - \lambda_n E(V_n) & \forall \text{ j} \in L \text{ such that } a_{nj}^{(0)} = 1 \\ \gamma_j^{(0)} & \text{otherwise} \end{cases} \quad (14)$$

Step (4): After computation of the average bandwidth share of route n we have modified the link incidence matrix and vector of link capacities. We now follow the same set of steps (from step (1) to step (3)) to find out the next route that is going to get the minimum average bandwidth share on some link. We now assign this minimum share to the average bandwidth share of this route and remove the route from the topology as we have done for route n earlier. We continue till kth iteration when we have $R^{(k)} = 0$.

V. Comparison of Analysis with Simulations

A. Simulation Setup

After choosing a particular network topology with some fixed routing plan over that network, we do a discrete event simulation. We simulated assuming data source as a fluid source and fluid flow rate at any instant is perfectly controlled and equals the IMMF rate. In all the simulations we have taken Poisson distribution for arrival process. Exponential distribution, and a hyper-exponential distribution matched to a heavy tailed distribution have been used for file size distribution.

B. Case 1: A Line Network

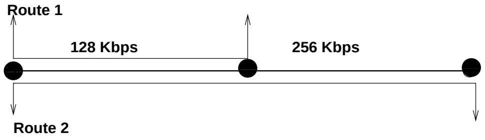

Fig. 3. Network topology for case 1

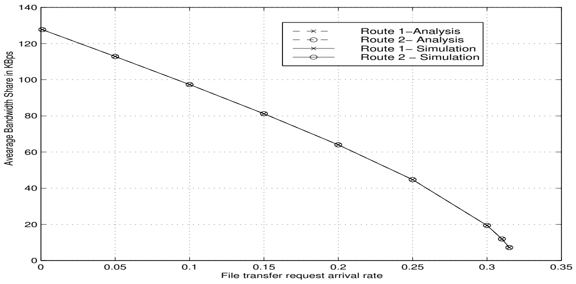

Fig. 4. Average bandwidth share vs. arrival rate for line network shown in Figure 3; exponentially distributed file sizes

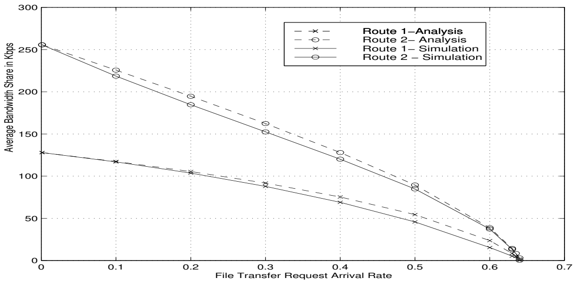

Fig. 6. Average bandwidth share vs. arrival rate for line network shown in Figure 5 ; exponentially distributed filesizes

The network and routing topology is shown in Figure 3. There are two routes, one spanning both the links and the other spanning a single link. The arrival rate of sessions in both the routes are the same (say λ) and having mean file size 200 Kilobits (exponentially distributed). We plot the simulation result and the average bandwidth share predicted from the algorithm in Figure 4. Observe that in this particular configuration both the routes are bottlenecked on the 128 Kbps link and as the arrival rates of sessions in both the routes are same, the average bandwidth share of both the routes are the same. Thus, the average bandwidth share of both the routes over link 1 will be less than the average bandwidth share of route 2 over link 2. The algorithm actually reduces to considering a network of a 128 Kbps link having two routes of the same arrival rate and equal mean service requirement. Note that the σ_1 and σ_2 obtained from the algorithm is exactly the same as when we consider two classes of traffic in a single link. This explains the exact match of simulation result and analysis.

C. Case 2: A Line Network

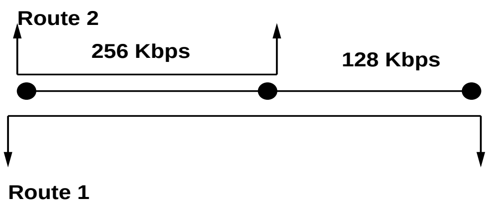

Fig. 5. Network topology for Case 2.

Here we use the same topology as shown in Figure 3 but different link capacities as shown in Figure 5. The plot of average bandwidth shares (with expontially distributed file size having mean 200 Kbits) is shown in Figure 6. Arrival rates of sessions in both routes are same. Observe that route 1 can get at most 128 Kbps while route 2 can get at most 256 Kbps. We also show the simulation result using a Pareto distribution (Figure 7). We take file size as $x = 200v$Kilobits when v is distributed in a *Pareto distribution* with ccdf given

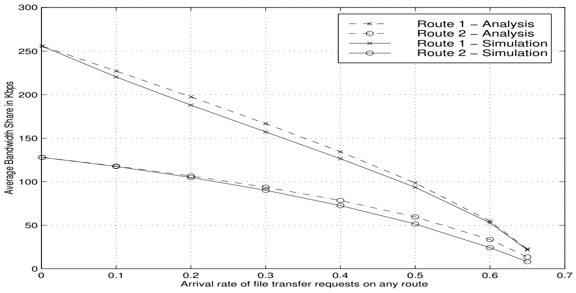

Fig. 7. Average bandwidth share vs. arrival rate for line network shown in Figure 5; file sizes distributed as a mixture of 7 exponentials matched with Pareto distribution

by $F^c(v) = (1 + bv)^{-a}$ with $a = 2.2$ and $b = .8333$ (i.e., with mean 1). In the simulation we approximate the heavy-tailed distribution as a mixture of 7 exponential distributions (see [28] for this Pareto distribution and this approximation). We get a hyper-exponential distribution with mean 0.9559 approximating the Pareto distribution.

Notice that the algorithm overestimates the simulation results but provides a good approximation. In the algorithm, for a fixed λ, we consider a particular link as the bottleneck link for a particular route. But in the simulation depending on the number of sessions in the two routes, the bottleneck can shift from one link to the other. When the other link becomes the bottleneck link then, it seems that particular route actually receives much less average badwidth share causing the algorithm to overestimate the average bandwidth share.

D. Case 3: A Mesh Network

We take a complicated mesh network with 9 links and 16 routes as shown in Figure 8. We plot the simulation and analysis results in Figure 9, Figure 10, Figure 11, Figure 12, Figure 13. Arrival rates of sessions in all the routes are the same, and the mean file sizes on all the routes are equal (200 Kilobits, exponentially distributed). The capacity of all the links is 1Mbps.

We now discuss the results in the plots. Note that the algorithm overestimates the bandwidth share in every case, but is

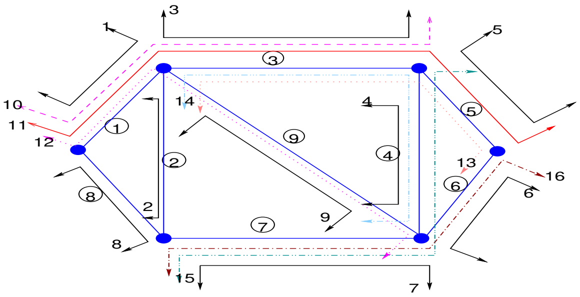

Fig. 8. Mesh network considered in Case 3 The numbers in the circles indicate link numbers, the others are route numbers.

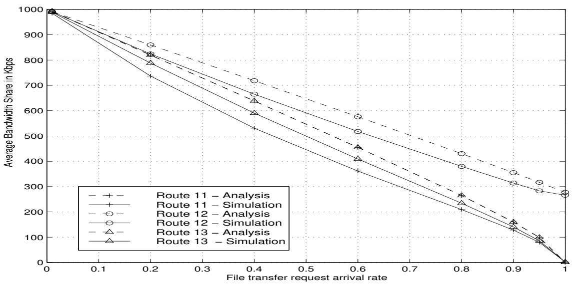

Fig. 12. Average bandwidth share in Kbps vs. arrival rate for mesh network shown in Figure 8; exponentially distributed file sizes

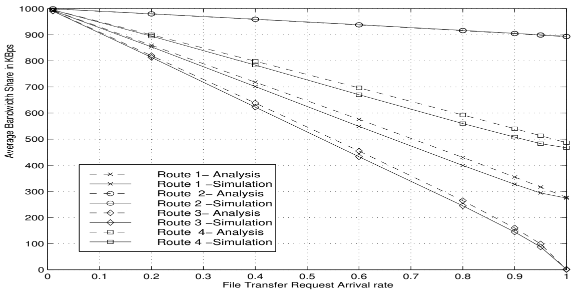

Fig. 9. Average bandwidth share in Kbps vs. arrival rate for the mesh network in Figure 8; exponentially distributed file sizes

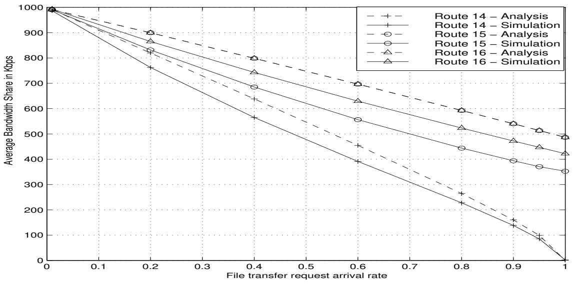

Fig. 13. Average bandwidth share in Kbps vs. arrival rate for mesh network shown in Figure 8; exponentially distributed file sizes

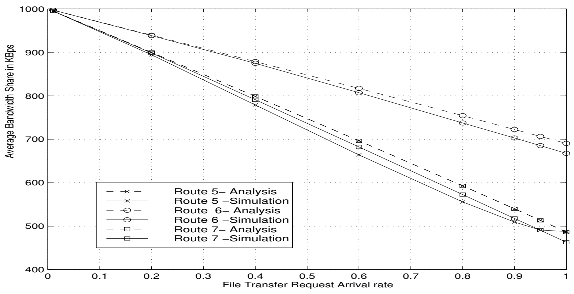

Fig. 10. Average bandwidth share in Kbps vs. arrival rate for the mesh network in Figure 8; exponentially distributed file sizes

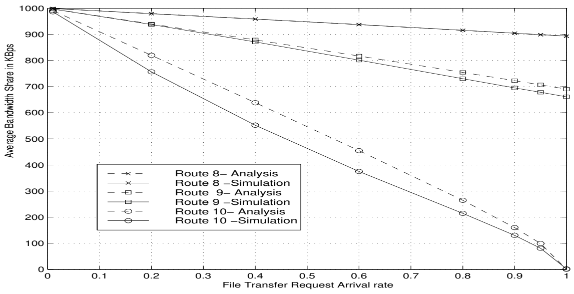

Fig. 11. Average bandwidth share in Kbps vs. arrival rate for mesh network shown in Figure 8; exponentially distributed file sizes

a good approximation. For most of the routes the error is less than 10%, with worst case error 25% (route 15 in Figure 13). Route 2 and route 8 are the only routes over link 2 and link 8 respectively. Hence, link 2 is the bottleneck link for route 2 and link 8 is the bottleneck link for route 8 all the time and their average bandwidth shares are same as when a single class of traffic stream is active on a single link. This explains the exact match between the analysis and simulation results. On link 5 and on link 7 three routes are active and all the routes having the same arrival rates. Hence both these links are equally loaded. Route 5 sessions are using only link 5 and route 7 sessions are using only link 7. Since we have two routes with equal arrival rates sharing two different links of same capacity and as well equally loaded, they get the same average bandwidth share from the computation of the algorithm. For route 11 link 1 and link 3 are the most occupied links in its path and the load on both the links are the same. For route 13 link 3 is maximum loaded link. These two routes have link 3 common in their path which happens to be maximum loaded link in their paths. It causes them to get the same average bandwidth share. Route 15 and route 16 are sharing the bandwidth of link 7 and this link is the maximum loaded link in their path in the network. As a result both of them get the same average bandwidth share.

We also show the simulation results using a *Pareto distribution* of file sizes (Figure 14, Figure 15). We take the file size as $x = 200v$ Kilobits when v is *Pareto* distributed. with ccdf

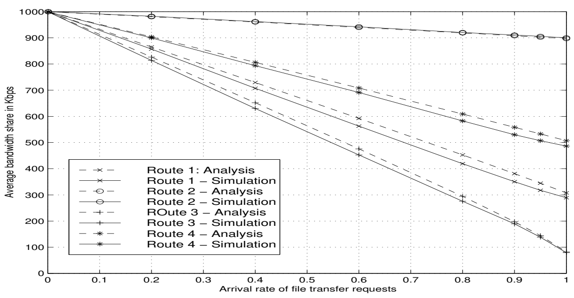

Fig. 14. Average bandwidth share in Kbps vs. arrival rate for mesh network shown in Figure 8; file sizes distributed in a mixture of 7 exponentials matched with Pareto distribution

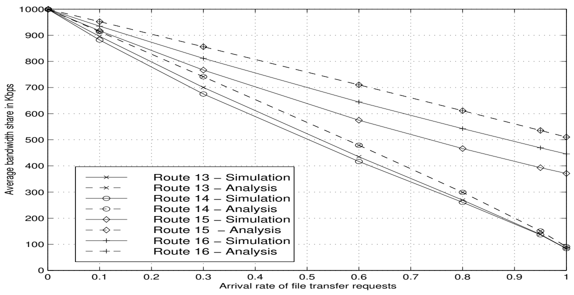

Fig. 15. Average bandwidth share in Kbps vs. arrival rate for mesh network shown in Figure 8; file sizes distributed in a mixture of 7 exponentials matched with Pareto distribution

given by $F^c(v) = (1 + bv)^{-a}$ with $a = 2.2$ and $b = .8333$ (i.e., with mean 1) . In the simulation we approximate the heavy-tailed distribution as a mixture of 7 exponential distributions (see [28] for this *Pareto distribution* and this approximation). We get a hyper-exponential distribution with mean 0.9559 approximating the Pareto distribution. We only show the result for a subset of all the routes in the network. Note that the simulation results are close to the approximation, and the approximation continues to be good.

VI. Summary

Our primary objective in this work has been to address the problem of computing the average bandwidth share obtained by ephemeral elastic sessions (using IMMF as bandwidth sharing policy) on different routes in a network, for a given network topology, routing plan, and offered load on each route.

We have proposed a simple heuristic algorithm to compute per session bandwidth share on different routes in the network. Simulations for different network topologies using exponential and heavy tailed distribution (approximated by a mixture of exponentials) as file size distribution, have been used to evaluate the validity of this proposal. We observed that even in complex network topologies and routing plans, though the algorithm overestimates the average bandwidth share of sessions, it provides good approximations. In ongoing work we are investigating whether the algorithm provides an upper bound to σ_r for every route.

References

[1] F.P. Kelly, A.K. Maulloo, and D.K.H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness, and stability," *Journal of the Operational Research Society*, vol. 49, 1998.

[2] M. Vojnovic, J.-Y. Le Boudec, and C. Boutremans, "Global Fairness of Additive Increase and Multiplicative Decrease with Heterogenous Round Trip Times," in *Proc. IEEE Infocom 2000*, 2000.

[3] Thomas Bonald and Laurent Massoulie, "Impact of fairness on internet performance," in *Sigmetrics*, 2001.

[4] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," *SIGCOMM'98*, 1998.

[5] Anurag Kumar, "Comparative Performance Analysis of Versions of TCP in Local Network with a Lossy Lnk," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, August 1998.

[6] Robert Morris, "TCP Behavior with Many Flows.," *Proc. IEEE International Conference on Network Protocols.*, October 1997.

[7] T.V. Lakshman and Upamanyu Madhow, "The Performance of TCP/IP for Networks with High Bandwidth Delay Products and Random Loss," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 336–350, June 1997.

[8] Steven Low, Larry Peterson, and Limin Wang, "Understanding TCP vegas: A Duality Model," in *Proc. SIGMETRICS 2001*, 2001, paper obtained from http://www.cs.princeton.edu/nsg/publications.html.

[9] Jeonghoon Mo and Jean Walrand, "Fair End-to-End Window-Based Congestion Control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, October 2000.

[10] Santosh P.Abraham and Anurag Kumar, "A New Approach for Asynchronous Distributed Rate Control of Elastic Sessions in Integrated Packet Networks," *IEEE/ACM Transactions on Networking*, February 2001.

[11] Steven H. Low and David E. Lapsley, "Optimization Flow Control-I: Basic Algorithm and Convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, December 1999.

[12] Aditya Karnik and Anurag Kumar, "Performance of TCP Congestion Control with Rate Feedback: Rate Adaptive TCP (RATCP)," in *Proc. IEEE Globecom 2000*, 2000.

[13] J. W. Roberts, "Engineering for quality of service," *K. Park and W. Willinger, editors, Self-Similar Network Trac and Performance Evaluation. Wiley-Interscience*, 2000.

[14] Gustavo De Veciana, Tae-Jin Lee, and Takis Konstantopoulos, "Stability and Performance Analysis of Network Supporting Services With Rate Control- Could the Internet be Unstable ?," *Proceedings of IEEE INFOCOM'99*, 1999.

[15] Guy Fayolle, Arnaud de La Fortelle, Jean-Marc Lasgouttes, Luarent Massoulie, and James Roberts, "Best-effort networks: modeling and performance analysis via large networks asymptotics," *Proc. IEEE Infocom*, 2001.

[16] D. P. Heyman, T. V. Lakshman, and A. L. Neidhardt, "A new method for analyzing feedback-based protocols with applications to engineering web traffic over the internet," *Proc. ACM-SIGMETRICS'97*, 1997.

[17] Arthur W. Berger and Yaakov Kogan, "Dimensioing bandwidth for elastic traffic in high speed data networks," 1998.

[18] A.A.Kherani and Anurag Kumar, "Performance analysis of tcp with nonpersistent sessions," workshop on modelling of flow and congestion control, inria, ecole normale superieure, paris, september 4-6, 2000," in *Workshop on Modelling of Flow and Congestion Control, INRIA, Ecole Normale Superieure, Paris*, September 2000.

[19] Claudio Casetti and Michela Meo, "A New Approach to Model Stationary Behavior of TCP Connections," *Proc. IEEE INFOCOM'2000*, April 2000.

[20] James Roberts and Laurent Massoulie, "Bandwidth sharing and admission control for elastic traffic," October, 1998.

[21] A. A. Kherani, "Rate Based Congestion Control for Ephemeral Best-Effort Sessions in Packet Networks," Master of Engineering Thesis, Indian Institute of Science, Bangalore, Jan 1999.

[22] Vern Paxson and Sally Floyd, "Wide-area traffic: the failure of Poisson modeling," *Proc. ACM SIGCOMM 94*, 1994.

[23] V. Paxson and S. Floyd, "Difficulties in simulating the internet," *To appear in IEEE/ACM Transactions on Networking*, February, 2001.

[24] Leonard Kleinrock, *Queueing Systems.*, John Wiley and Sons, 1996.

[25] Arzad Alam Kherani, Anurag Kumar, Pinaki Shankar Chanda, and Vijay H. Shankar, "On the Throughput of nonpersistent elastic sessions on an internet link with small propagation delay," Submitted.

[26] E. Gelenbe and L. Mitrani, *Analysis and Synthesis of Computer Systems*, 1980.

[27] Pinaki Shankar Chanda, "Models for Traffic Engineering of Packet Networks with Non-Persistent Elastic Flows," Master of Engineering Thesis, Indian Institute of Science, Bangalore, Jan 2001.

[28] Anja Feldmann and Ward Whitt, "Fitting mixtures of exponentials to long-tail distributions to analyze network performance models," Tech. Rep., AT&T Laboratory - Research, December 16, 1996.