Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Computer Networks 42 (2003) 65–80

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

Predictive routing to enhance QoS for stream-based flows sharing excess bandwidth

Xun Su *, Gustavo de Veciana [1]

Department of Electrical and Computer Engineering, University of Texas, Engineering Science Building, (ENS) 516, Austin, TX 78712-1084, USA

Received 4 June 2002; received in revised form 4 December 2002; accepted 4 December 2002

Responsible Editor: G. Kesidis

Abstract

We propose a new routing algorithm based on online estimation of the link load dynamics and prior information on flow holding times. The motivation for this proposal lies in supporting traffic flows such as VBR or associated with rate adaptive applications. Such traffic requires a minimal guaranteed bandwidth, but can see improved performance when sharing excess bandwidth not used to meet guarantees. The key idea underlying our approach is to route traffic flows so that they see minimal expected flow-perceived loads during their sojourn in the network. To this end we establish a routing framework where links estimate and advertise the parameters associated with their load dynamics in addition to their *current* load. New flows are routed based on this information and prior knowledge of their holding times so as to minimize the expected flow-perceived load. Simulations of this routing scheme in a (weighted) max–min bandwidth sharing framework show gains of 20–50% in the average flow bandwidth share over baseline routing schemes.

Keywords: Dynamic routing; Load prediction; Bandwidth sharing; Expected flow-perceived load

1. Introduction

We investigate routing mechanisms for stream-based traffic flows. The traffic and service model we consider can be summarized as follows: upon arrival to the network the traffic flows require a minimal level of guaranteed service, e.g., in terms of a minimal bandwidth guarantee. The flow is admitted if there are sufficient resources to do so along the selected route, otherwise the flow is rejected. After admission into the network, the flows [2] may achieve improved performance by sharing network resources which are not in use to guarantee service for ongoing flows. In general, the performance achieved by a given flow depends on the resource allocation policy at the flow level and the packet scheduling policy. At the flow level, the shared resources are allocated to ongoing flows

* Corresponding author.
E-mail addresses: xsu@ece.utexas.edu (X. Su), gustavo@ece.utexas.edu (G. de Veciana).
URL: http://www.ece.utexas.edu/gustavo.
[1] Tel.: +1-512-471-1573; fax: +1-512-471-5532.

[2] In this paper we refer to traffic flows and their associated users interchangeably.

1389-1286/02/$ - see front matter
doi:10.1016/S1389-1286(02)00456-5

based on the resource sharing policies employed in the network, e.g., TCP [1] or max–min sharing [2]. At the packet level, the packet scheduling policy determines the packet service rate for a given flow in accordance with the flow level resource allocations. In this paper we focus on analyzing routing schemes that improve the overall network performance at the flow level. We believe such improvements in flow level performance, coupled with a suitable work-conserving scheduling policy, can lead to better user-perceived QoS.

For a given resource sharing policy, the performance achieved by a given flow during its sojourn in the network depends on a number of factors including the number of flows in the network, the resources available for sharing, and the set of links traversed by these flows, i.e., their routes. This motivates us to investigate routing mechanisms that not only optimize system metrics such as the *flow blocking rate*, but also enhance the *user-perceived* performance to the individual flows. Prominent service classes that fit in this generic service model include ATM VBR service [3] and rate adaptive applications [4]. Specifically, ATM VBR connections would request a level of QoS, e.g., cell loss rate, upon arrival to the network. The call admission control (CAC) mechanism employed by the network might then translate this user-centric QoS specification into an estimate of the resources required to satisfy the user QoS demand, e.g., *effective bandwidth* [5]. Given an estimate for the effective bandwidth the network decides to admit or reject connections. Note that from a user's perspective, it is beneficial to route the admitted VBR connections on a path with additional spare resources, so that the inaccuracies in the estimates of the effective bandwidth can be better tolerated. In the case of rate adaptive applications, traffic flows arriving to the network are given a minimal bandwidth guarantee, and expect variable transmission rates, i.e., when the load is lower (higher), the flows adapt to higher (lower) transmission rates, possibly by subscribing (unsubscribing) to additional service layers [6]. In this case the excess bandwidth seen by the flows might be used to support lower priority layers. These observations suggest that it might be beneficial to route these flows so as to minimize the average load a flow is likely to see during its sojourn in the network. [3] We shall refer to the average load seen by a flow as the *flow-perceived load*, and set out to design a routing scheme that aims at improving this performance measure, in addition to minimizing the flow blocking rate.

To achieve this goal, we consider routing schemes that use prior knowledge of the flow holding time. For example, the holding time might be known or characterized via its mean or distribution. We propose to model the link load dynamics as a means to estimate the expected flow-perceived load on the network links. As in [7], where a CAC scheme is studied, we will use the queuing-theoretic results in [8] to propose a parametric model for the link load dynamics. In our routing framework, links estimate and advertise the parameters associated with their loads in addition to their current states. New flows are routed based on this information and prior knowledge of their holding times so as to minimize the expected flow-perceived load. We will show that even with limited information on flow holding times, e.g., their means, one can often improve both the flow blocking rate and flow-perceived load [4] *simultaneously*. Considering that the improved flow blocking rate implies an increase in the load supported by the network, it is remarkable that one can also achieve better performance in terms of flow-perceived load, and thus better eventual QoS. To substantiate this claim, we will show that in a network where available bandwidth is shared in fair fashion, a significant increase in a flow's bandwidth share can be realized when using our routing approach versus two baseline schemes.

In order to study the effectiveness of our approach, we examined various operational issues by simulations. We believe the proposed routing scheme operates effectively in a wide range of contexts, and its performance is robust to various

[3] Equivalently, we might attempt to maximize the average available bandwidth a flow is likely to see during its sojourn in the network.

[4] The flow-perceived load is measured by averaging individual flows's perceived load over all flows that are served by the network.

uncertainties in the network's operating environments.

1.1. Related work

As mentioned above, in this paper we propose a routing scheme that routes traffic flows based on both link load dynamics and prior knowledge on flow holding time. We will use an auto-regressive process to model the link load dynamics, and estimate its parameters based on load samples. The key idea is to integrate such information in the notion of the *expected flow-perceived load*, and route the traffic flows so that the expected load seen by flows during their sojourn in the network is minimized. Our work contributes to ongoing research on routing QoS traffic, by considering the role that prior knowledge of flow holding times might play.

Specifically, in [9] a number of competitive routing algorithms are presented for ATM networks. The results indicate that one can design online routing algorithms to achieve different degrees of competitiveness with respect to the optimal offline algorithm, depending on the assumptions made concerning prior knowledge of connection holding times. Rather than focusing on designing a good routing scheme relative to the worst case arrival process, in this paper we optimistically assume that link loads follow quasi-stationary stochastic dynamics.

In [10], a routing scheme is proposed which provides differentiated handling of short versus long-lived flows. Data packets are routed on static shortest paths, until a flow classifier is triggered to switch the flow routing based on a dynamic algorithm that is load-sensitive. Our work differs from [10] in that we use dynamic routing for all the traffic flows, but the differentiation is done through the use of different routing metrics for different flows. Instead of relying on a flow classification trigger as in [10], our scheme explicitly determines per-flow routing behavior by integrating into the routing decision the (mean) flow holding time and the estimated parameters characterizing link load dynamics.

In [11] a number of routing algorithms are examined in a network where bandwidth is shared among best effort traffic flows according to the max–min fair criterion. The authors propose a routing metric which approximately estimates the max–min rate for the new connection upon arrival. The resulting shortest path algorithm outperforms minimum hop routing and shortest–widest path routing in terms of packet throughput. Our work differs from [11] in several aspects: (1) we focus on improving the performance (i.e., blocking *and* flow bandwidth share) of stream-based flows instead of max–min rate share of the best-effort file transfers; (2) we use a link state information that scales better than that used in [11], where each link needs to maintain a sufficient number of "rate scales" in order to obtain an adequate estimate for the rate share of the new connection, and (3) we believe that the notion of expected flow-perceived load effectively captures the resource-sharing potential in a network, thus routing schemes incorporating this notion will apply to other resource sharing criteria such as proportional fair share [12,13] and size-based sharing [14].

More generally, there has been extensive study of dynamic routing, e.g., on its instability if done at the packet level [15], or on approaches to minimize blocking rate at the flow level by ensuring a better "load-balancing" [16–24]. Our approach can also be categorized as a load-balancing scheme. However, it differs from the afore-mentioned work, not only in terms of the specific routing metrics we propose, but also in that as a routing objective we explicitly identify the improvement of the individual flow's perspective of resource sharing potential.

In the following sections we present asymptotic approximations for the link load dynamics and the associated parameter estimation techniques, based on which our routing algorithm is constructed. We then examine various factors related to this routing scheme, propose an extension to mesh networks, and discuss simulation results that validate the effectiveness of our approach.

2. Analysis: a simple parallel-link model

Let us consider a simple parallel link model, where a source node s and a destination node d are

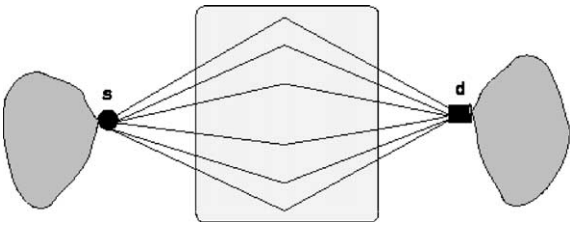

Fig. 1. A simple parallel-link topology.

connected by n links, see Fig. 1. Each link i has a capacity of c units, and serves an exogenous flow load which arrives according to a Poisson process with rate λ_i. Each flow has an exponentially distributed holding time with mean μ_i^{-1}, and requires one unit of bandwidth to ensure its minimal QoS guarantee. In this paper we will model the link load dynamics associated with the *minimal* bandwidth commitments the network has made and make routing decisions based on this model to improve the QoS of flows sharing *excess* (or additionally available) bandwidth. We denote the number of flows in progress on link i, or equivalently the load at time t, by $X_i^c(t)$, where the superscript c indicates that this is the load process on a link with capacity c. We will subsequently consider two asymptotic regimes where c and λ_i grow.

2.1. A new routing metric: expected flow-perceived load

We consider routing flows that arrive at node s and are destined to node d. Let us assume the considered flow load from node s to node d is *small* in comparison with the load generated by the exogenous flow processes described above, so that the routing of the flows from s to d will not affect the stationary link loads $X_i^c(t)$, $i = 1, 2, \ldots, n$. Suppose a single flow with a known holding time h is to be routed at time 0 from node s to node d. Let link loads be $X_i^c(0) = x_i^c(0)$, $i = 1, 2, \ldots, n$. As discussed in the introduction we propose to route the flow to the link where it is likely to experience a minimal load during its sojourn in the network.

We define the *flow-perceived load* as the time average of the load during the flow's sojourn in the network. Thus suppose a new flow is to be routed at time 0, we can express the *expected flow-perceived load* on link i as

$$u_i(h, x_i^c(0)) := E\left[\frac{1}{h}\int_0^h X_i^c(t)\,\mathrm{d}t | X_i^c(0) = x_i^c(0)\right] = \frac{1}{h}\int_0^h E[X_i^c(t)|X_i^c(0) = x_i^c(0)]\,\mathrm{d}t.$$

As mentioned in the introduction this metric quantifies the expected load a flow with known holding time h would see on link i. We propose to route the flow to the link i with maximum *expected flow-perceived available bandwidth*, i.e., $c - u_i(h, x_i^c(0))$. When all links have the same capacity this is equivalent to minimizing the expected flow-perceived load $u_i(h, x_i^c(0))$. We will later relax the assumption that h is known and examine the sensitivity of such routing algorithms to the knowledge of the flow holding time. Below we consider some approximations for the expected flow-perceived load, assuming the link load dynamics are independent of the routing decisions.

2.2. First approximation: a fluid model

Consider an asymptotic regime where λ_i and c approach infinity, but $\lambda = \theta_i \cdot c$, i.e., the flow arrival rate increases linearly as link capacity increases, irrespective of the link load condition. As shown in [8], it follows that $(X_i^c(t)/c) \xrightarrow{a.s.} x_i(t)$ as $c \to \infty$, for $0 \leqslant t \leqslant q$, $\forall q < \infty$, where $\{x_i(t)\}$ satisfies the following ordinary differential equation:

$$\dot{x}_i(t) = \theta_i - \mu_i x_i(t).$$

Thus if $(X_i^c(0)/c) \xrightarrow{a.s.} x_i(0)$, we have that

$$x_i(t) = x_i(0) \cdot \mathrm{e}^{-\mu_i t} + \frac{\theta_i}{\mu_i}(1 - \mathrm{e}^{-\mu_i t}). \tag{1}$$

Hence for a link with large capacity c and such that $X_i^c(0) = x_i^c(0)$ we have that roughly

$$X_i^c(t) \approx x_i^c(0) \cdot \mathrm{e}^{-\mu_i t} + \frac{\lambda_i}{\mu_i}(1 - \mathrm{e}^{-\mu_i t}). \tag{2}$$

Using this asymptotic regime we can approximate the expected flow-perceived load introduced in Section 2.1 as follows. Assume the link has a large capacity c and its load is $X_i^c(0) = x_i^c(0)$ then by (2) we have that

$$u_i(h, x_i^c(0)) \approx \lim_{c\to\infty} E\left[\frac{1}{h}\int_0^h X_i^c(t)\cdot \mathrm{d}t | X_i^c(0) = x_i^c(0)\right] = \left(x_i^c(0) - \frac{\lambda_i}{\mu_i}\right)\cdot\frac{1-\mathrm{e}^{-\mu_i h}}{\mu_i h} + \frac{\lambda_i}{\mu_i}.$$

Observe that for short flow holding times the expected flow-perceived load corresponds to the link's state $x_i^c(0)$, and for long flow holding times the expected flow-perceived load tends to the long-term average load λ_i/μ_i. Hence if $x_i^c(0) < (\lambda_i/\mu_i)$, i.e., the initial load is lower than the long-term average load, flows with short holding times will see a lower expected flow-perceived load than those with longer holding times. Conversely, if the initial load is higher than the long-term average load, flows with longer holding times will see a lower expected flow-perceived load than those with shorter holding times.

Fig. 2 illustrates a special case with two links between source node s and destination node d. The incoming flow may encounter a number of situations with different initial link loads and long-term average link loads. Specifically, for Case (a), Link 1 is preferred even though the initial link loads at time 0 are the same for the two links. For Case (b), Link 1 is preferred since both its initial load and long-term average load are lower than those of Link 2. For Case (c), there exists a "cross-over" flow holding time $\tilde{h}$ where $u_1(\tilde{h}, x_1^c(0)) = u_2(\tilde{h}, x_2^c(0))$. For flows with holding time shorter than $\tilde{h}$ Link 1 is preferred, and for flows with holding time longer than $\tilde{h}$ Link 2 is preferred. For Case (d), there exists a "cross-over" flow holding time where for flows with holding time shorter than $\tilde{h}$ Link 1 is preferred, and for flows with holding time longer than $\tilde{h}$ Link 2 is preferred. These cases exemplify the potential gains that can be achieved by judiciously accounting for both the flow holding time and link load dynamics.

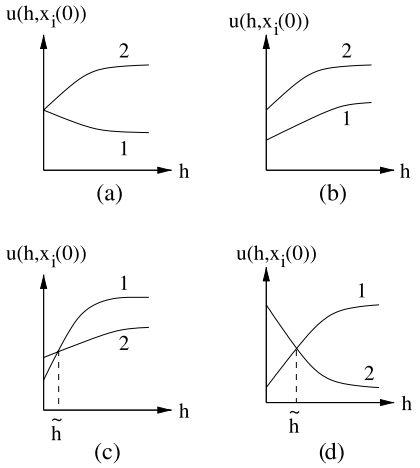

Fig. 2. Routing in two-parallel-link network.

2.3. Second approximation: a diffusion model

The fluid model presented in Section 1 allows us to approximately characterize the evolution of the link load dynamics. This model arises when we examine the scaled link load $X_i^c(t)/c$ in the limiting regime where the link capacity c and load $\lambda_i = \theta_i \cdot c$ grow linearly. We can also establish a similar relationship characterizing link load dynamics by investigating the scaled stochastic fluctuations of the link load about its mean.

Suppose a "mode" exists for the limiting regime, i.e., $(X_i^c(t)/c) \xrightarrow{a.s.} x_i(t) = (\theta_i/\mu_i)$, then as proven in [8] as $c \to \infty$ the fluctuation process about the mode converges to an Ornstein–Uhlenbeck process. In particular as $c \to \infty$, $((X_i^c(t) - c\theta_i/\mu_i)/\sqrt{c}) \xrightarrow{\text{dist.}} X_i(t)$, where $\{X_i(t)\}$ satisfies the following stochastic differential equation:

$$\mathrm{d}X_i(t) = -\mu_i X_i(t) + \sqrt{2\theta_i}\,\mathrm{d}B_i(t),$$

where $\{B_i(t)\}$ is a standard Brownian motion. Thus we can approximately model the link load process as an Ornstein–Uhlenbeck process, which is the solution to the following stochastic differential equation:

$$\mathrm{d}X_i^c(t) = -\alpha_i(X_i^c(t) - \rho_i)\,\mathrm{d}t + \sigma_i\,\mathrm{d}B(t),$$

where $\alpha_i = \mu_i$, $\rho_i = c\theta_i/\mu_i$, $\sigma_i = \sqrt{2c\theta_i}$.

Consider again a flow with holding time h to be routed to a link i with $X_i^c(0) = x_i^c(0)$ and whose load dynamics are characterized by the above Ornstein–Uhlenbeck process. The expected flow-perceived load in this regime would be given by

$$u_i(h, x_i^c(0)) \approx \rho_i + (x_i^c(0) - \rho_i)\frac{1-\mathrm{e}^{-\alpha_i h}}{\alpha_i h}, \qquad (3)$$

since $E[X_i^c(t)|X_i^c(0) = x_i^c(0)] = (x_i^c(0) - \rho_i)e^{-\alpha_i t} + \rho_i$. This is similar to the expected flow-perceived load obtained for the fluid model, even though in this case the load dynamics are modeled by stochastic fluctuations about the mode. The reason for this similarity lies in the fact that we are focusing on the mean of the link load process versus the second order statistics inherent in the diffusion approximation. Indeed, the proposed routing metric does not depend explicitly on σ_i, and thus in a sense, does not capture the degree of fluctuation in the perceived load a flow might see. However, as shown in the sequel, using only the first order characteristics for the perceived loads already achieves significant performance gains. The impact of the second order statistics on the resulting QoS seen by flows is left for future study. In the following sections we will use (3) as our link metric and will assume the load process can be adequately modeled by an Ornstein–Uhlenbeck process.

2.4. Link load characteristics: parameter estimation

In order to make routing decisions based on the proposed link metric we will estimate the parameters (i.e., ρ_i, α_i) for the load process model for each link. Note that in practice the flow arrivals seen by a link would not be Poisson with a constant rate, as assumed above. Instead the arrival rates are likely to depend on the current state of the network, i.e., if the link load is low, one might expect to see a higher arrival rate, and if the link load is high the arrival rate might go down. However, in general the dynamics of this process will exhibit the "mean reversion" property of the Ornstein–Uhlenbeck process, i.e., there exists a "mode", and the link load exhibits fluctuations about this mode due to arrivals to, and departures from the system. These in turn are influenced by the routing decisions that are being made.

Let us thus consider modeling the link load process $\{X_i(t)\}$ associated with link i as an Ornstein–Uhlenbeck process with parameters $(\rho_i, \alpha_i, \sigma_i)$. To estimate the needed parameters we sample the link loads every Δ time units. Define the sampled process $y_i(k) = X_i(k\Delta)$ for $k \in Z$. The parameters can be estimated using the following [25]:

$$\hat{\rho}_i = \frac{1}{n}\sum_{k=1}^{n} y_i(k), \quad \hat{\alpha}_i = -\frac{\ln \hat{\beta}_i}{\Delta},$$

where

$$\hat{\beta}_i = \frac{\sum_{k=2}^{n} (y_i(k) - \hat{\rho}_i)(y_i(k-1) - \hat{\rho}_i)}{\sum_{k=1}^{n} (y_i(k) - \hat{\rho}_i)^2}.$$

Note that the selection of sampling period Δ and sampling window n impacts the quality of the parameter estimates. In the sequel we use simulations to assess the importance of these sampling parameters. It is known that the spectrum of the Ornstein–Uhlenbeck process is of the "low-pass" type, i.e., with a cut-off frequency (3dB point) at α_i, hence one might roughly argue that the sampling rate should be at least $2\alpha_i$. For the queuing models discussed earlier the cut-off frequency α_i equals to μ_i, i.e., the flow departure rate. However, in practice the routing mechanism itself would accelerate the mean reversion thus one should expect to require a sampling rate faster than $2\mu_i$, i.e., $\Delta < (1/(2\mu_i))$.

2.5. Dynamic or adaptive routing?

Routing algorithms are often said to be either *dynamic*, i.e., using most up-to-date link states, or *adaptive*, i.e., using averaged/filtered link states. The proposed routing metric is based on *both* the most up to date link states *and* the averaged parameters quantifying the "stationary" or long-term characteristics of the link loads. As observed earlier as the flow holding time h becomes small the proposed metric is essentially a dynamic one, i.e., the current link state, while for large h the longer term characteristics of the link's load are used to make the routing decisions.

To be precise consider a link whose load dynamics is characterized by parameters (ρ_i, α_i). In this case the link load relaxes exponentially to the long-term average ρ_i, see (1) and (2). The "relaxation time" is roughly $1/\alpha_i$. By contrast the expected flow-perceived load is defined as the expectation of the time-averaged link load, and thus relaxes more slowly. Its effective "relaxation time" is roughly e/α_i. Let $h_i^r = e/\alpha_i$. Thus for sufficiently large holding times, i.e., $h > h_i^r$, we have

that $u_i(h, x_i(0)) \approx \rho_i$ and the routing of a flow with such holding times may be said to be adaptive. By contrast if its holding time is smaller than h_i^r one might say the metric accounts for the dynamic characteristics of the link's load.

For our simple topology with two links, let h_i^r corresponds to the "critical" flow holding time for link i, where $i = 1, 2$. We observe that for all flows with flow holding times greater than $\max\{h_1^r, h_2^r\}$, the routing mechanism is adaptive. Similarly, for all flows with holding time less than $\min\{h_1^r, h_2^r\}$ the routing mechanism is essentially a dynamic one.

In summary, these criteria roughly show a "split" between flows with different holding times, according to which flows are routed in a dynamic or adaptive manner.

2.6. Impact of the delays in advertising link states

In a link-state routing scheme, there usually exists a broadcasting mechanism through which the link states at the routers are updated. Inevitably updating delays are involved in such broadcasting schemes, due to overhead constraints on message propagation and processing delays. In this subsection we examine the impact of updating delays on the proposed routing metrics. Consider the scenario where we make a routing decision at time t, but only have access to the advertised link state at time $t - d$. Without loss of generality suppose $t = d$ and at that time we have access to $x_i(0)$ as well as the parameters (α_i, ρ_i) characterizing the link load. If the delay d is known one can compensate for this by computing the expected flow-perceived load as follows:

$$\begin{aligned} u_i(h, x_i(0); d) &= \frac{1}{h}\int_d^{h+d} E[X_i(s)|X_i(0) = x_i(0)]\,\mathrm{d}s \\ &= \rho_i + (x_i(0) - \rho_i)\left(\frac{1 - \mathrm{e}^{-\alpha_i h}}{\alpha_i h}\right)\mathrm{e}^{-\alpha_i d} \\ &\approx u_i(h, x_i(d)), \end{aligned}$$

since $x_i(d) \approx (x_i(0) - \rho_i)\mathrm{e}^{-\alpha_i d} + \rho_i$. We observe that as d increases $u_i(h, x_i(0); d)$ converges to ρ_i. Thus if significant delays are involved in link-state updates, the routing algorithm that accounts for the (known) updating delays would be essentially adaptive.

Note that this discussion assumes that the delay associated with the current update for the link state is known. In practice this can be done by time-stamping link state updates. However, in the sequel we will, for the most part, not assume such delays are known. Instead, outdated link states are treated as "current" and directly used in estimating the expected flow-perceived load according to (3), i.e., when making routing decisions at time d we use $x_i(0)$ in place of $x_i(d)$. Let $\tilde{x}_i(d) = x_i(0)$. In this case

$$\begin{aligned} u_i(h, \tilde{x}_i(d)) &= \rho_i + (x_i(0) - \rho_i)\left(\frac{1 - \mathrm{e}^{-\alpha_i h}}{\alpha_i h}\right) \\ &\approx \rho_i + (x_i(d) - \rho_i)\left(\frac{1 - \mathrm{e}^{-\alpha_i h}}{\alpha_i h}\right)\mathrm{e}^{\alpha_i d}. \end{aligned}$$

Hence if $d \ll \alpha_i^{-1}$, $u_i(h, \tilde{x}_i(d)) \approx u_i(h, x_i(d))$. We will see in the sequel that even in the case where $d \approx \alpha_i^{-1}$ the predictive flow-time aware routing scheme still provides performance improvements over our baseline schemes. However, the "time-stamping" mechanism can contribute to additional performance improvements.

2.7. Uncertainty in flow holding times

Previously we assumed that flow holding times were known in advance. In practice this may not be the case. In this subsection we consider the sensitivity of the routing decisions to uncertainty in the flow holding time. We approach this via two different avenues, (1) what is the impact of uncertainty in the flow holding time on the *routing metric*, i.e., the expected flow-perceived load? and (2), when do the *routing decisions* change as flow holding times vary? We shall write the expected flow-perceived load on link i as

$$u_i(h, x_i(0)) = \rho_i + (x_i(0) - \rho_i) l_i(h),$$

where $l_i(h) = (1 - \mathrm{e}^{-\alpha_i h})/\alpha_i h$ and h denotes a known holding time. Note that $l_i(h)$ is decreasing and convex in h, thus the proposed routing metric is fairly insensitive to the uncertainty in h when h is large. Suppose only the mean $\bar{h}$ of a flow's holding time distribution is known. Let H be a random variable with that distribution. One might consider using $u_i(\bar{h}, x_i(0))$ as a routing metric. We note that

if $\bar{h}$ is large and the variance of H is small then this metric is fairly representative of the *actual* expected flow-perceived load.

Even if h is moderate or small and so that $l_i(h)$ is relatively sensitive to h, we argue that although the routing metric $u_i(h, x_i(0))$ may vary if we use $\bar{h}$ instead of the actual flow holding time, the routing decisions based on this may not. Fig. 2 provides an illustration. For Fig. 2(a) and (b), we observe that no matter how h varies, the routing decisions remain the same. For Fig. 2(c) and (d), there exists a certain timescale $\tilde{h}$ such that for all h less than $\tilde{h}$ Link 1 is favored, and for all h greater than $\tilde{h}$ Link 2 is favored. Hence in these cases if h and $\bar{h}$ remain on the same side of $\tilde{h}$, the routing decisions made based on $\bar{h}$ will not change.

3. Predictive flow-time aware routing in a mesh network

In Section 2 we proposed a routing scheme based on the notion of *expected flow-perceived load*, in the context of a simple parallel-link topology. The basic ideas generalize to mesh networks with multiple source–destination pairs routing flows simultaneously. Note that in this case the *link metrics* must be used to construct *path metrics*. The task here is to compute paths for the incoming traffic flows so that (1) the network can carry as many traffic flows as possible, and (2) the perceived loads by the admitted flows during their sojourn in the network are as low as possible. To achieve these goals, we will have to make a number of design choices:

- Whether to use additive or concave link metrics to construct path metrics;
- How to incorporate the notion of expected flow-perceived load into the link metrics;
- How to effectively estimate the parameters that characterize the link load dynamics and the expected flow-perceived load.

To systematically address these issues, we have performed extensive simulations of the proposed routing approach. Below we show the performance of our predictive flow-time-aware routing (FTAR) scheme, and illuminate a number of factors that may impact its performance.

3.1. Simulation setup

We performed simulations for different network topologies and traffic matrices. In the following we present a set of results for the network shown in Fig. 3. In our simulations, the flows arrive to the network according to a Poisson process, and the flow holding times are randomly distributed. We experimented with various flow holding time distributions, e.g., exponential, Pareto, hyper-exponential, bi-modal. The general trends of the results are similar under different holding time distributions. We will only show results corresponding to the exponential distributions. The ingress and the egress nodes for the new flows are selected according to Table 1, which corresponds to a typical WAN traffic pattern, i.e., the ingress and egress nodes of a flow are at least two hops away from each other.

The parameters for the simulation were set as follows: link capacity is 200 bandwidth units. The flow arrival rate between each pair of source and destination nodes in Table 1 is set to be 50 flows per time unit. The mean flow holding time is 1 time unit, which might represent, say a 3-minute period for voice applications, or a 1-hour period for video transmissions. In the following simulations we will use the mean instead of the exact value of flow holding time to evaluate the expected flow-perceived load. *The various timescales we will encounter in this section*, e.g., *link load sampling period, sampling window size, and link state updating delays,*

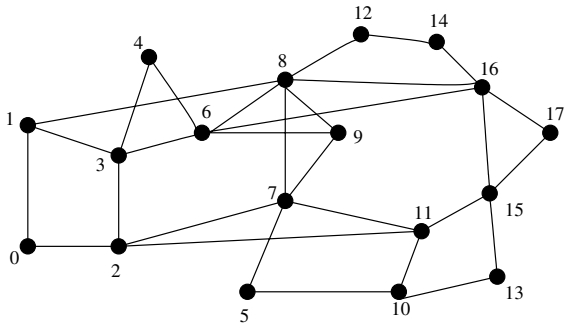

Fig. 3. NSF topology.

Table 1
Traffic sources and destinations

Ingress node	Egress node	Hop distance
0	16	4
1	17	3
2	16	3
2	13	3
3	9	2
4	13	4
5	14	4
8	10	3
10	4	5
11	4	4

will all be set relative to the mean flow holding time. The bandwidth request of each flow is uniformly distributed between 0.5 and 1.5 bandwidth units.

This setup is referred to as the base case. We increase the traffic load by scaling the arrival rates of the base case by a sequence of factors. The links in the network estimate the parameters that characterize their load dynamics, i.e., ρ_i, α_i, and distribute these parameters along with the current link load periodically. We will refer to our routing scheme FTAR.

3.1.1. Three routing algorithms

We will compare FTAR with two baseline routing schemes. The first is referred to as dynamic single path (DSP), and uses the reciprocal of the *current* available capacity as the routing metric [17]. The second baseline scheme is referred to as mean single path (MSP) and uses the reciprocal of the *estimated mean* available capacity as the routing metric, i.e., $c - \rho_i$. Here ρ_i is the mean load estimated by FTAR. FTAR is a revised version of DSP, i.e., we use expected flow-perceived load instead of current link load to evaluate the available capacity. For an incoming flow, we compute the shortest path based on the inverse of the *expected flow-perceived bandwidth*, i.e., the link capacity minus the expected flow-perceived load, and establish the flow on the resulting path if the available bandwidth along the path allows it. Otherwise the flow is blocked.

3.1.2. Performance metrics

We compare routing schemes based on the *percentage of demands* that are successfully routed, and the *average flow-perceived excess bandwidth* seen by flows. The latter is determined by first sampling the residual bandwidth seen by a given flow during its sojourn in the network, then averaging these samples to get its *perceived excess bandwidth* when it departs, and finally averaging over all the departed flows. Note that this is a measure of how much bandwidth there is in the network for a given flow to share with other ongoing flows during its sojourn, i.e., the *potential* for better performance, but not necessarily the bandwidth *achieved* by the flow. Clearly, the *flow bandwidth share* depends on the specific bandwidth sharing policy used in the network, e.g., max–min sharing, proportional sharing [13], or size-based bandwidth sharing [14]. In the sequel we use (weighted) max–min sharing to illustrate the effectiveness of our routing scheme in terms of average flow bandwidth share.

Moreover, in the following sections we will evaluate "% improved routed volume" and "% improved average flow-perceived excess bandwidth", which are defined as $((x-y)/y)\cdot 100$, where x is the performance (% routed volume or average flow-perceived excess bandwidth) achieved by our FTAR scheme, and y is that achieved by the corresponding baseline scheme.

3.2. Parameter estimation: the optimal sampling rate and window size

Let us first examine the impact on the routing performance of the parameter estimation procedure. In particular, we focus on determining a good choice for the sampling rate, i.e., the speed at which a link takes samples of its loads, and the sampling window, i.e., the duration of the time over which the samples are kept in memory.

The discussion in Section 2.4 suggests that the sampling rate should be fast enough to obtain accurate parameter estimates, i.e., $\Delta^{-1} \geqslant 2\mu_i$. Estimates are based on samples within a *moving window* [5] so the size of the window might impact

[5] It is also feasible to use an "exponentially weighted-averaging" mechanism to estimate these parameters.

the routing performance. In the following we shall vary the sampling rate and sampling window size to identify the set of operational values. The results show that the performance of the FTAR routing scheme is robust to the selection of sampling rate and sampling window size, unless very poor choices are made.

Fig. 4 shows the performance of our routing scheme for different sampling periods and window sizes. Observe that when the sampling window is small, i.e., equal to 0.01 time units, the routing performance in terms of routed volume is unsatisfactory. Indeed if the sampling window is not large enough we are not able to capture the link load dynamics. In addition, note that when the sampling rate is small, i.e., with a sampling period of 1 time unit, the routing performance also deteriorates. This is consistent with our assertion in Section 2.4 that if the sampling is not done frequently enough, we will not have sufficient samples to be able to estimate the parameters for the Ornstein–Ulenbeck model.

Note that other than the specific cases described above, the routing performance is robust to the choice of sampling rate and sampling window size. In the sequel we will use a sampling period equal to 0.1 time units and a sampling window size equal to 1.5 time units.

3.3. Performance gains using predictive flow-time-aware routing

Let us now compare the performance of FTAR with DSP and MSP. In Fig. 5 we show a typical result for the case where the current link states are assumed to be known, i.e., no updating delays (in which case DSP performs "ideally"). We consider nonzero updating delays in Section 3.5. We see that FTAR improves the routing performance over both DSP and MSP, by up to 10% in terms of routed volume. We observe that FTAR performs consistently better than MSP, and that only in the heavily loaded regime where FTAR is supporting a higher traffic volume, does its average flow-perceived excess bandwidth become less than that for DSP. Note that in the lightly loaded regime the flow blocking performance of FTAR is better than DSP and MSP, and FTAR also provides better performance in terms of average flow-perceived excess bandwidth. This is surprising since the network using FTAR is admitting a higher number of flows. Hence the overall routing of traffic must be significantly improved by using a predictive FTAR mechanism.

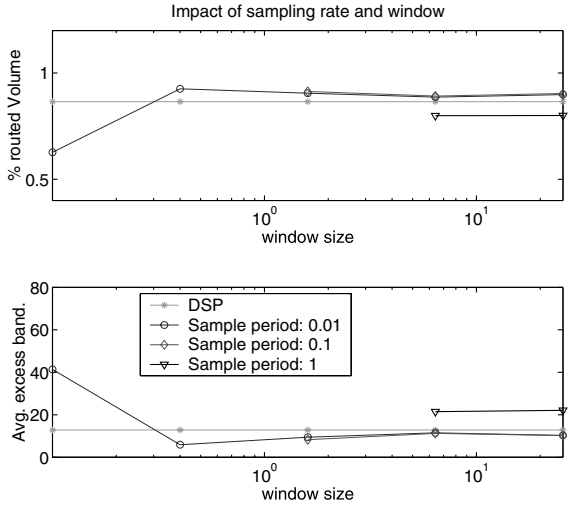

Fig. 4. Performance in terms of routed volume and average flow-perceived excess bandwidth as functions of link load sampling rates and sampling window sizes.

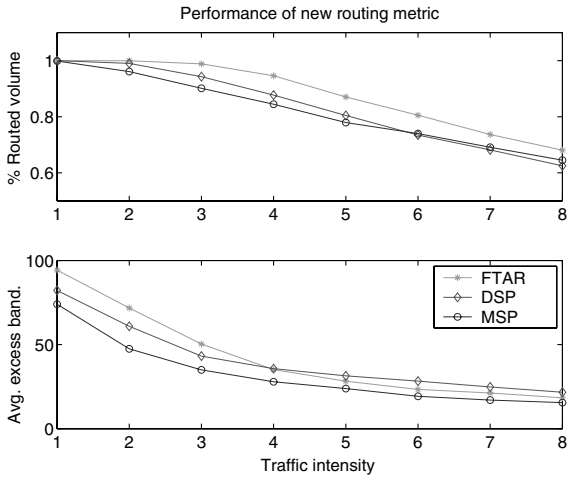

Fig. 5. Performance gain in routed volume and average flow-perceived excess bandwidth by FTAR over DSP and MSP.

3.4. Concave or additive path metrics: choice of mesh routing algorithms

To determine good path between a pair of source–destination nodes for an incoming flow, one often resorts to a notion of "shortest path" or "widest path". On the one hand, the construction of a *shortest* path often proceeds by adding up link metrics. On the other hand, the construction of a *widest* path usually involves taking the minimum (a "concave" operation) of several link metrics. It is not entirely clear what routing metrics and their associated algorithms one should use for a specific routing scenario, though [17] suggests that the inverse of the residual bandwidth might be a good additive routing metric to achieve network load-balancing. Note that in a simple parallel topology like the one we used in the previous sections the "shortest" and "widest" routing schemes are equivalent, i.e., the difference arises only when there are multi-link paths in question.

In the context of predictive FTAR, we believe the choice of routing strategy, i.e., "shortest" or "widest" criterion, depends on the characteristics of the incoming flow and the corresponding network load condition. In particular, we note that the "dominating link" on a path, i.e., the link that exhibits the "worst" load level, might vary during a flow's sojourn in the network. Fig. 6 shows the performance comparison between the shortest–widest routing scheme using concave metric and the shortest path routing scheme using additive metric. We see that the routing scheme using additive metric outperforms the routing scheme using concave metric, by up to 12% in terms of routed volume and by up to 120% in terms of average flow-perceived excess bandwidth.

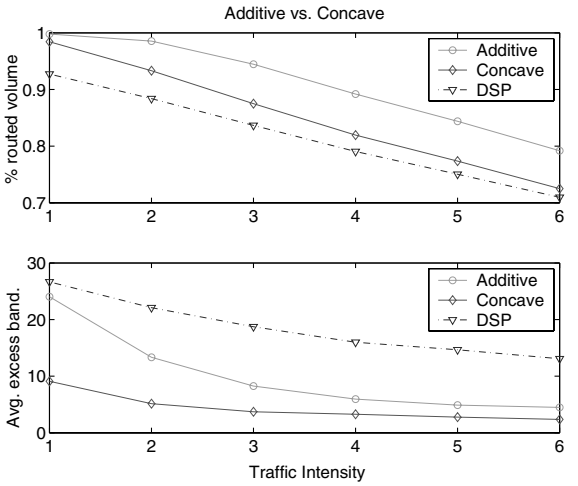

Fig. 6. Performance comparison in routed volume and average flow-perceived excess bandwidth of concave versus additive path metric.

3.5. Effect of state advertising delays

As often is the case in practice, there are delays involved in link state broadcasts. Since dynamic routing schemes make use of link states, it is important to gage the impact these delays have on routing performance. In this section we first compare the performance of FTAR and DSP as such delays increase. In particular, we will have a "slow update" scenario, where the link states are updated every 1 time unit, and a "fast update" scenario, where the link states are updated every 0.1 time units. These may correspond to networks with different geographical coverage, i.e., long versus short-haul networks, or simply different limitations on the signaling overheads. We will use delayed link load in computing routing metrics for FTAR and DSP. As shown in Fig. 7 the

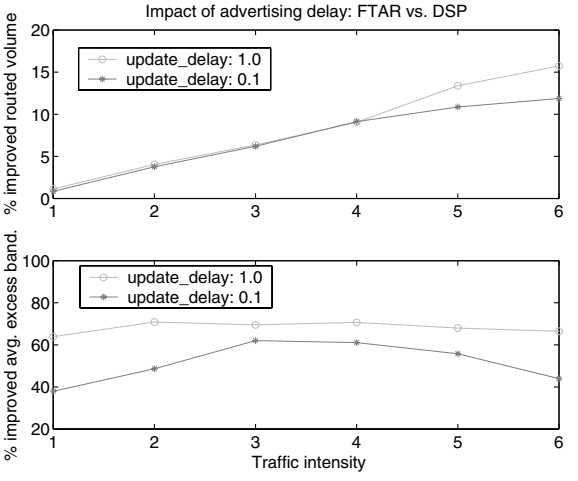

Fig. 7. Performance improvement in routed volume and average flow-perceived excess bandwidth for FTAR over DSP for different advertising delays.

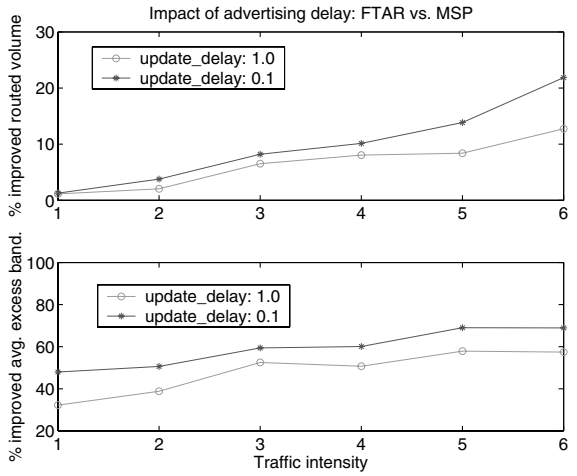

Fig. 8. Performance improvement in routed volume and average flow-perceived excess bandwidth for FTAR over MSP for different advertising delays.

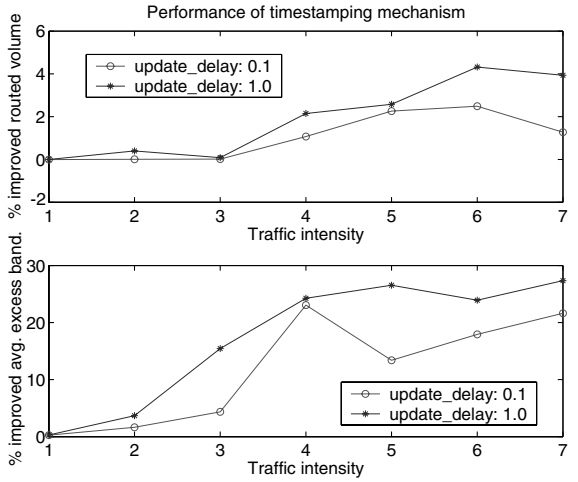

Fig. 9. Performance improvement in routed volume and average flow-perceived excess bandwidth for FTAR over DSP by using time-stamping mechanism under different broadcast delays.

performance improvement by FTAR over DSP is more significant when the advertising delays are larger. This confirms our intuition in that the larger delays lead to a diminishing effect on the routing performance of the "current" link states, or alternatively, the more significant contribution by the long-term average load information, which is captured and utilized by FTAR. Furthermore, in Fig. 8 we plot the performance improvement of FTAR over MSP under different link state advertising delays. We see that this improvement decreases as the link state advertising delay increases. However, FTAR still consistently outperforms MSP, even under the (relatively) large delay case we considered, i.e., 1 time unit.

3.6. Time-stamping mechanism

It is of interest to compare the routing performance using delayed link states, as presented in Section 2, with that where a "time-stamping" mechanism is used to determine exactly the delay associated with a given link state, i.e., the links attach a time-stamp to the link states when they are advertised. As discussed earlier when making routing decisions routers can use this time-stamp information to determine the delay of link states and thus estimate the expected flow-perceived load according to (4). Fig. 9 shows the performance improvement achieved by FTAR augmented with time-stamp over FTAR without the knowledge of link state advertising delays. We see that this time-stamping scheme improves the routed volume by 4% and average flow-perceived excess bandwidth by 25%. Moreover, we note that when the update delay is larger, the performance improvement obtained by the time-stamping mechanism is more significant. This is intuitive considering the fact that the difference in routing metrics increases when the update delay increases between the cases with and without time-stamps, and hence the difference in the routing decisions.

3.7. Bursty arrivals: Markov modulated Poisson process

In the previous simulations we modeled the flow arrivals by Poisson processes. This is a relatively "smooth" random process. In this section we examine the effect of a more bursty arrival process. Specifically, we use Markov modulated Poisson process (MMPP) to model the flow arrivals. There

are two "modulating" states, "high" and "low". In each state traffic flows arrive as a Poisson process. We will consider two MMPPs with different flow arrival rates in the "high" state. For the first, the flow arrival rate in the "high" state is three times the mean given in Table 1. For the second, the flow arrival rate in the "high" state is 1.5 times the mean given in Table 1. In the "low" state, traffic flows arrive with rate 1/3 of the mean given in Table 1, for both MMPPs. Besides the rates associated with the "high" and "low" states, the MMPPs are also characterized by the mean time they stay at "high" and "low" states. For the first MMPP, we set the mean time at "high" state and "low" state to be $0.5 \cdot \text{MMPP\_TIME}$ and $1.5 \cdot \text{MMPP\_TIME}$, respectively, where MMPP_TIME is a scaling variable which we vary from 10 to 90 time units. For the second MMPP, we set the mean time at both modulating states to be MMPP_TIME. The flow holding time is again exponentially distributed, with mean 1 time unit. Note that the first MMPP is more bursty than the second MMPP.

In Fig. 10 we compare the performance improvement for FTAR over DSP, under different flow arrival processes. Observe that as the flow arrival process becomes more bursty the improvement in terms of routed volume increases, while the improvement in terms of average flow-perceived excess bandwidth decreases. These seemingly diverging trends make sense, since as FTAR allows increasing traffic load into the network, the average flow-perceived excess bandwidth reported by the (larger amount of) supported traffic decreases. This indicates that in an operating regime with bursty flow arrivals, it will be beneficial to use information on link load dynamics in addition to the "current" link load.

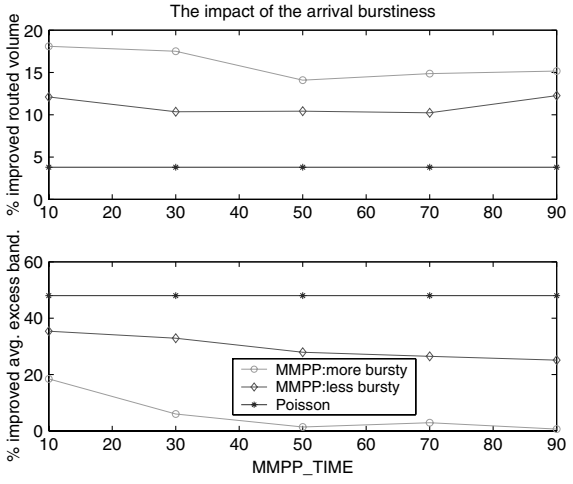

Fig. 10. Performance improvement of FTAR over DPS in routed volume and average flow-perceived excess bandwidth with bursty flow arrivals.

4. Application: routing max–min rate adaptive sessions

In the previous sections we showed that by using a link metric associated with the expected flow-perceived load, the routing performance improves in terms of both routed volume and average flow-perceived excess bandwidth. The former metric corresponds to the ability of the network to support traffic flows having minimal guaranteed bandwidth requirement. The latter metric corresponds to the *potential* for the admitted traffic flows to improve their "achieved" performance by sharing the excess bandwidth in addition to the guaranteed minimal rate. In this section we show by simulation that in a *max–min* bandwidth sharing framework the proposed routing scheme can indeed realize the potential and yield improved "achieved rate". We assume that upon arrival and departure of the flows the *excess* bandwidth allocated to the ongoing flows are instantaneously re-computed according to the max–min rate allocation scheme [2], and the traffic sources are responsive enough to adjust their transmission rates accordingly. In the following we examine the performance improvement achieved by our routing scheme in terms of the additional bandwidth seen by flows, i.e., the *average flow bandwidth share*, which is measured by first taking sampled-average of the additional bandwidth allocated to the individual flows during their sojourn in the network, then averaging over all the departed flows.

In principle the max–min bandwidth sharing is fair in the sense that it does not discriminate against flows traversing long routes. Bandwidth

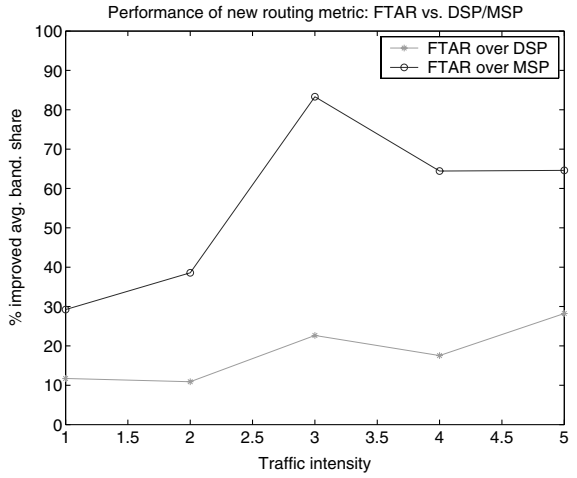

Fig. 11. Performance improvement for FTAR over DSP and MSP in terms of average flow bandwidth share using weighted max–min fair sharing.

sharing schemes used in practice, e.g., proportional-fair sharing, or TCP will however do so [13]. We consider a *weighted max–min* sharing scheme where larger (smaller) weights are given to the flows that traverse shorter (longer) routes. This corresponds to larger (smaller) amount of bandwidth being allocated to the flows that traverse shorter (longer) routes.

In Fig. 11 we show a performance comparison for FTAR, DSP, and MSP routing schemes. We know from the previous simulations that the blocking performance of FTAR is superior to the baselines. To highlight the capability of FTAR in obtaining improved max–min shared rates, here we show an operating regime where the load is light, i.e., no blocking occurs for all routing schemes. An improvement of 10–30% is achieved over DSP and the improvement over MSP can be up to 80%. [6]

5. Conclusion and discussion

In this paper we proposed a new dynamic routing scheme which improves the overall performance *achieved* by stream-based flows during their sojourn in the network. The novelty of our approach lies in (1) the identification of the notion of *expected flow-perceived load*, which quantifies the "potential" for improvement of the user's performance, that exists at a given link from a specific flow's perspective, (2) the construction of a practical routing algorithm which realizes the above potential, based on an auto-regressive load model and the prior information on flow holding time. For a large class of traffic and service models, e.g., VBR and rate adaptive applications, an effective use of our approach will result in better flow QoS. Specifically, we constructed a routing algorithm that aims at minimizing expected flow-perceived load during a flow's sojourn in the network. We showed that this routing algorithm leads to not only better load balancing in the network, but also improved flow-perceived performance. This allows the flows admitted to the network to realize a greater share of "achieved" bandwidth, in addition to their minimal requested amount.

The implementation of the proposed routing scheme would require updating routing software. We use prior information on the holding time of the traffic flows. This can be either presented by the traffic flows upon arrival to the network, or obtained through traffic statistics gathered by the network operator. In addition, routers in the network need to maintain link load models. The effort here includes estimating the parameters of the model and advertising the estimated parameters along with current link loads. This implies additional computational and signaling overhead. However, we note that in a distributed routing environment a given router need only maintain the link load models for the adjacent links, which scales at most linearly with the number of the routers in the network. [7] Moreover, these estimated parameters are "stable" since they correspond to the mean and the rate of variation for a quasi-stationary stochastic process, thus they need not be updated as frequently as the current link

[6] The performance comparison using *unweighted* max–min sharing yields similar result.

[7] That is, in a fully connected network. In a mesh network it grows much slower.

load. It is shown in Section 3.5 that FTAR achieves higher performance gain with larger advertising delays when compared to DSP. This suggests FTAR is more robust than DSP when link state advertisements become less frequent, enabling reduced overhead by using larger advertising delays. These observations lead us to believe that the performance advantage of our routing scheme outweighs concerns with overhead. In conclusion, the routing designer can improve the routing performance of the stream-based flows by taking advantage of information regarding link load dynamics and flow holding time, without having to significantly increase the routing overheads.

Acknowledgements

This work is supported by National Science Foundation Career Grant NCR-9624230, and by an Intel Technology for Education 2000 equipment grant.

References

[1] V. Jacobson, Congestion avoidance and control, Proc. ACM Sigcomm, 1988, pp. 314–329.
[2] D. Bertsekas, R. Gallager, Data Networks, Prentice Hall, Englewood Cliffs, NJ, 1992.
[3] A. Forum, P-NNI draft specification, Tech. rep., March 1995.
[4] S. Shenker, Fundamental design issues for the future Internet, IEEE J. Select. Areas Commun. 13 (7) (1995) 1176–1188.
[5] F. Kelly, Notes on effective bandwidths, in: F. Kelly, S. Zachary, I. Ziedins (Eds.), Stochastic Networks: Theory and Applications, Oxford University Press, Oxford, 1996, pp. 141–168.
[6] S. McCanne, Scalable compression and transmission of internet multicast video, Ph.D. Thesis, University of California at Berkeley, 1996.
[7] C. Courcoubetis, A. Dimakis, M. Reiman, Providing bandwidth guarantees over a best-effort network: call-admission and pricing, in: Proc. IEEE Infocom, 2001, pp. 459–467.
[8] A. Mandelbaum, W. Massey, M. Reiman, Strong approximations for Markovian service networks, Queue. Sys. 30 (1998) 149–201.
[9] S. Plotkin, Competitive routing of virtual circuit in ATM networks, IEEE J. Select. Areas Commun. 13 (6) (1995) 1128–1136.
[10] A. Shaikh, J. Rexford, K. Shin, Load-sensitive routing of long-lived IP flows, Proc. ACM Sigcomm, 1999.
[11] Q. Ma, P. Steenkiste, H. Zhang, Routing high-bandwidth traffic in max–min fair share networks, Proc. ACM Sigcomm, 1996, pp. 206–217.
[12] R. Gibbens, F. Kelly, Resource pricing and the evolution of congestion control, Automatica (1999) 1969–1985.
[13] L. Massoulié, J. Roberts, Bandwidth sharing: objectives and algorithms, in: Proc. IEEE Infocom, 1999, pp. 1395–1403.
[14] S. Yang, G. de Veciana, Size based adaptive bandwidth allocation: Optimizing the QoS for elastic flows, in: Proc. IEEE Infocom, 2002.
[15] Z. Wang, J. Crowcroft, Analysis of shortest-path routing algorithms in a dynamic network environment, ACM Comput. Commun. Rev. 22 (2) (1992) 63–71.
[16] G. Apostolopoulos, R. Guérin, S. Tripathi, Quality of service based routing: A performance perspective, in: Proc. ACM Sigcomm, 1998, pp. 17–28.
[17] Q. Ma, P. Steenkiste, On path selection for traffic with bandwidth guarantees, Fifth IEEE International Conference on Network Protocols, 1997.
[18] M. Kodialam, T. Lakshman, Minimum interference routing with applications to MPLS traffic engineering, in: Proc. IEEE Infocom, 2000, pp. 884–893.
[19] F. Kelly, Network routing, Proc. Roy. Soc. Lond. A 337 (1991) 343–367.
[20] R. Gibbens, Dynamic routing in fully connected networks, IMA J. Math. Control Inf. 7 (1990) 77–111.
[21] M. Alanyali, B. Hajek, On simple algorithms for dynamic load balancing, Proc. IEEE INFOCOM, 1995, pp. 230–238.
[22] Z. Wang, J. Crowcroft, Quality-of-service routing for supporting multimedia applications, IEEE J. Select. Areas Commun. 14 (7) (1996) 1228–1235.
[23] S. Nelakuditi, Z. Zhang, R. Tsang, Adaptive proportional routing: A localized QoS routing approach, Proc. IEEE Infocom, 2000, pp. 1566–1575.
[24] R. Gibbens, F. Kelly, S. Turner, Dynamic routing in multiparented networks, IEEE/ACM Trans. Network. 1 (2) (1993) 261–270.
[25] P.J. Brockwell, R.A. Davis, Introduction to Time Series and Forecasting, Springer, New York, 1996.

Xun Su received his BSEE from University of Electronic Science and Technology of China in 1992. He earned his MSEE from Southeast University, China in 1995. He entered University of Texas at Austin in September, 1996 and joined Dr. Gustavo de Veciana's networking research group in Spring 1998. He concluded his research at UT Austin in December 2002 with a focus on network routing algorithms and obtained a Ph.D. in Electrical Engineering. His research interests include network routing, wireless ad hoc networking, peer to peer systems and network measurement.

Gustavo de Veciana received his B.S., M.S. and Ph.D. in electrical engineering from the University of California at Berkeley in 1987, 1990 and 1993 respectively. In 1993, he joined the Department of Electrical and Computer Engineering at the University of Texas at Austin where he is currently an Associate Professor. His research focuses on issues in the analysis and design of telecommunication networks. Dr. de Veciana has been an editor for the IEEE/ACM Transactions on Networking. He is the recipient of a General Motors Foundation Centennial Fellowship in Electrical Engineering and a 1996 National Science Foundation CAREER Award, and co-recipient of the IEEE Bill McCalla Best ICCAD Paper Award for 2000.