Dynamic multi-path routing: asymptotic approximation and simulations *

Xun Su, Gustavo de Veciana
Department of Electrical and Computer Engineering
University of Texas, Austin, TX 78712
{xsu,gustavo}@ece.utexas.edu

ABSTRACT

In this paper we study the dynamic multi-path routing problem. We focus on an operating regime where traffic flows arrive at and depart from the network in a bursty fashion, and where the delays involved in link state advertisement may lead to "synchronization" effects that adversely impact the performance of dynamic single-path routing schemes.

We start by analyzing a simple network of parallel links, where the goal is to minimize the average increase in network congestion on the time scale of link state advertisements. We consider an asymptotic regime leading to an optimization problem permitting closed-form analysis of the number of links over which dynamic multi-path routing should be conducted. Based on our analytical result we examine three types of dynamic routing schemes, and identify a robust policy, *i.e.,* routing the traffic to a set of links with loads within a factor of the least loaded, that exhibits robust performance. We then propose a similar policy for mesh networks and show by simulation some of its desirable properties. The main results suggest that our proposal would provide significant performance improvement for high speed networks carrying bursty traffic flows.

1. INTRODUCTION

As the Internet continues growing and new technologies emerge to meet this growth, networking researchers are faced with the increasingly daunting task of controlling and/or managing extraordinary amounts of traffic. Traditionally, network operators have relied on buffers at network nodes and/or congestion control to deal with fluctuations in traffic loads. However, as traffic loads become more bursty, the size and the speed of the buffers in the network are not growing commensurately with the link speed, making buffering techniques less effective. Moreover, although link speeds increase dramatically, propagation delays stay roughly unaffected, calling into question the effectiveness of flow/congestion control mechanisms. Indeed, to control the congestion inside the network, we have traditionally relied on end-to-end reactive flow control schemes, *e.g.,* TCP[5] and Random Early Detection[2]. These schemes rely on coordination among links within the network and traffic sources at the network edge. Links detect the congestion and send back "congestion indications" (*e.g.,* drop/mark packets) to the traffic sources which in turn respond by adjusting their transmissions. However, in an operating regime with high bandwidth-delay product, *i.e.,* transmission rate times the round trip time from the ingress node to the egress node, this reactive approach is not effective. The problem is twofold: (1) since the link capacity is huge, the traffic in flight when a congestion signal is generated is enormous so the network must be able to buffer a large amount of data and (2) since access speeds may be very high, a traffic burst that induces congestion may finish by the time the traffic source receives the corresponding congestion indications. In both cases the response occurs too late to effectively avoid congestion. A similar phenomenon is observed in the dynamic routing context, exemplified by the routing "synchronization" problem, where link updates are late and ineffective in navigating the traffic flows across a congested network. An interesting question that stands out, is whether we can avoid network congestion without having to slow down the user's transmission rates.

In this paper we focus on an operating regime in which traffic flows come and go within the time scale of link state advertisements. We view such flows as high speed transmissions, *i.e.,* a sequence of IP packets transmitted at a high rate, and following the same path. As a result, network congestion in this context exhibits a relatively short term dynamics and can not be effectively controlled through per source feedback schemes like TCP. Instead of slowing down user transmission rates to enable better congestion control, we propose routing schemes that alleviate network congestion while allowing users to send traffic at their full access rates. The idea is to disperse traffic flows sharing the same ingress/egress points via multiple paths on the network, in order to achieve "statistical multiplexing" of the flows over

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

available network resources [4]. This in itself is not a new idea, and is part of a tradition of alternative routing and dispersion used in some circuit switched networks [8, 11, 13, 12, 6, 3, 15].

Traffic dispersion, with its early origins in "dispersity routing" [9], has been an active research area. Dispersion at the packet, burst and flow/connection levels have been considered, see [4] and the references therein. In particular, [9] originated the idea of packet-level dispersion in the context of store-and-forward data networks, and showed that by spreading the traffic over two (or three) paths the average delay of a message is significantly reduced. Dispersity routing, now at the flow/connection level, was further adapted to the ATM networks[10], where it has been shown to equalize traffic loads and increase overall network utilization for short flows with durations in the same order as the propagation time or less. It also points out the possibility of dispersing flows adaptively. The combination of these two issues, short flows and adaptive multipath routing, is the starting point for our study. However, as pointed out in [4], the problem of determining the optimal set of paths over which such dispersion should be performed, remains open.

Somewhat akin to this problem, alternative routing and trunk reservation have been studied extensively, see *e.g.,* [8, 11, 13]. In the context of circuit-switched networks [3, 12, 6], a trade-off is sought between increasing routing options and resource utilization. This means that if the primary (usually short) paths experience congestion, secondary paths will be used to carry the traffic load. However, secondary paths are only used if they are not loaded beyond a certain *threshold*, otherwise new arrivals are blocked. The key parameters, the primary/secondary paths and the *threshold*, are difficult to optimize for general network topologies.

This paper addresses one of the key issues that needs to be addressed in dispersing flows over multiple paths, *i.e.,* how to (dynamically) select the set of candidate routes over which traffic flows will be dispersed based on potentially outdated link state information, or even adapt this set to achieve better overall performance. This contributes to the ongoing research efforts that extend the functionality of the OSPF[17], MPLS[16], or Diffserv[14], where deterministic[17, 16] or probabilistic[14] header processing mechanisms have been proposed to facilitate the dispersion of traffic flows over multiple paths from an ingress node to an egress node. With these hashing operations packets with the same attributes (*e.g.,* source address, destination address, QoS requirement), will form a flow that traverses the *same* path. Our study provides sensible routing decisions based upon which the packet-level hashing decisions can be constructed, *i.e.,* the set of paths over which the packet flows are sent.

In the ensuing sections we first consider a simple model consisting of parallel links between an ingress-egress node pair. The main result suggests a simple and robust policy to select a subset of candidate links over which to spread incoming traffic flows. We then propose and evaluate a dynamic multi-path routing scheme for mesh networks.

2. A STOCHASTIC PARALLEL-LINK MODEL

2.1 Problem Setup

In this section we study the idealized model shown in Figure 1, where a pair of ingress and egress nodes are interconnected via a set of n links, $L = \{1, 2, \cdots, n\}$, each having the same capacity c, $l \in L$. Notice that these parallel links can be used to model either real links, or *disjoint* routes between an ingress node and an egress node. Let λ_{sd} denote the flow arrival rate from node s to node d. Without loss of generality we assume each flow transmits packets at a fixed unit rate for a random duration with mean μ^{-1}, along the route it is assigned. The offered load associated with node s and node d, is thus λ_{sd}/μ. The traffic load on link $l \in L$ at time t, denoted by $x_l(t)$, is the sum of the total number of flows currently routed across it. We let $\hat{x}(t) = (x_1(t), x_2(t), \cdots, x_n(t))$. The flow arrival rate to link l is denoted by γ_l, *i.e.,* the part of total arrivals between node s and node d, *i.e.,* λ_{sd}, that is routed to link l. As an approximation, in the sequel we examine the dynamics of the link loads via a fluid model. The flow departure rate is proportional to the current link load $x_l(t)$ and is given by $x_l(t) \cdot \mu$. We consider a time scale of interest, t, that represents the potential delays involved in updating link states. To model the fact that the routing decisions are made based on outdated information, we assume traffic arrivals during $[0, t]$ are routed based on the link state $\hat{x}(0)$ available at time 0. Under an routing scheme which results in a total traffic arrival rate γ_l to link l and does not lead to link overflows, the link state $x_l(t)$ tracks the following differential equation,

$$\dot{x}_l(t) = \gamma_l - x_l(t) \cdot \mu, \tag{1}$$

hence

$$x_l(t) = x_l(0)e^{-\mu t} + \frac{\gamma_l}{\mu}(1 - e^{-\mu t}), \tag{2}$$

where $x_l(0)$ is the state of link l at the beginning of the time interval $[0, t]$, and γ_l, *i.e.,* the routing/assignment of incoming traffic flows to link l, remains fixed over $[0, t]$.

We use an additive network congestion measure, $s(x(t)) = \sum_{l=1}^{n} f(x_l(t))$, where $f(x_l(t))$ is an increasing and convex function of $x_l(t)$, *e.g.,* $f(x_l(t)) = \frac{1}{c - x_l(t)}$. Our goal is to find an allocation of incoming traffic flows within a time interval $[0, t]$ to the n links such that the increase of the system congestion measure, *i.e.,* $s(x(t)) - s(x(0))$, is minimized. This is equivalent to minimizing the system congestion *at time t*, $s(x(t))$. This choice is intended to simplify the analysis, as will be seen in the following section. Eq.(1) captures link load dynamics that may impact the routing decisions, *i.e.,* the more link l is loaded, the faster traffic flows depart from it. Intuitively, this observation suggests $x_l(t)$ may underestimate the available capacity on a heavily loaded link when it comes to routing new traffic demands. In particular, the

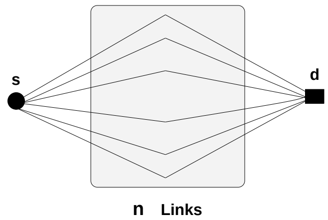

Figure 1: An abstract model of route selection.

capability of a more loaded link to accommodate traffic demands might be favorably "upgraded" since it is likely to see more departing flows.

Given the current link states $\hat{x}(0)$ and the offered load λ_{sd}, one can in principle determine the optimal routing that minimizes system congestion. Our goal, however, is to find a simple dynamic multi-path routing policy. In particular, we focus on a form of least-loaded routing scheme where *equal shares* of the traffic load are routed on a subset of k links[1]. The key problem is to determine an "optimal" k which is "robust" to a range of possible link loads, the intensity of the flow arrival process, and the mean flow holding time. In the sequel, we derive such a solution by finding a k which on average is "optimal" over a range of possible link states. We call this *dynamic multi-path routing* since based on the network state k links are selected to disperse the traffic.

2.2 Analysis

We let X_l, $l = 1, 2, \cdots, n$, denote the random loads on the links *at time 0*. We assume that they are independent and identically distributed with a continuous distribution function F and support set $[0, c]$. These distributions might be selected to reflect the typical operating regimes of the system, e.g., typically lightly loaded or heavily loaded. Alternatively one might select the prior distributions on the link loads to be uniformly distributed, so as to achieve a robust solution over a range of possible operating regimes. Note that in reality a number of factors impact F, *e.g.,* traffic arrival rates, flow holding times, and routing algorithm, thus our assumption on the random link loads may not be realistic in practice. In particular, the link loads are neither identically distributed, nor independent. However, our intent in introducing this simplifying assumption, is to enable analytical derivation of a robust policy and in turn garner interesting insights on dynamic multi-path routing.

Since our policy involves selecting the k least loaded links, we will make use of *order statistics* on link loads. We use $X^n_{(i)}$ to denote the i^{th} order statistic of the n link loads at time 0, thus $X^n_{(1)} \le X^n_{(2)} \le \cdots \le X^n_{(n)}$. Let $\lambda_{sd} = \lambda \cdot n$. Hence λ is a measure of the flow arrival rate, normalized by the number of *options*, *i.e.,* n, over which routing decisions are to be made. Suppose incoming traffic flows over a time interval $[0, t]$ is spread over k links that are the least loaded at time 0 , then the resulting congestion increase is

$$D^n(k) := \sum_{i=1}^{k} [f(X^n_{(i)} e^{-\mu t} + \frac{n \cdot \lambda}{k}(1 - e^{-\mu t})) - f(X^n_{(i)})] + \sum_{i=k+1}^{n} [f(X^n_{(i)} e^{-\mu t}) - f(X^n_{(i)})].$$

The first sum on the right hand side accounts for the change in the congestion level for the k least loaded links which share the incoming traffic flows during time interval $[0, t]$, while the second term corresponds to the links which see no additional load.

An "optimal" selection of k might correspond to solving the following minimization problem:

$$k^* = \text{argmin}_k \{E[D^n(k)] \mid k \in \{1, 2, \ldots, n\}\}. \quad (3)$$

The expectation is taken so as to obtain a choice of k that is optimal "on average" over a range of possible link loads. Yet this problem is still quite difficult to solve. In the following we consider an asymptotic regime, wherein the offered load $n \cdot \lambda$ and number of links n grow [2], *i.e.,* we consider a sequence of networks with increasing routing diversity and carried load. We parameterize k as $k = \lceil \alpha n \rceil$, hence α corresponds to the fraction of (least loaded) links over which the traffic flows will be spread. Our goal is to find α which minimizes the *normalized* average congestion increase as $n \to \infty$, *i.e.,*

$$\min_{0 \le \alpha \le 1} \lim_{n \to \infty} \frac{E[D^n(\lceil \alpha n \rceil)]}{n}. \quad (4)$$

The following theorem establishes that (4) can be expressed in two equivalent forms that are amenable to analysis. The proof is deferred to the appendix.

THEOREM 1. *The problem defined in (4) can be rewritten as*

$$\min_{0 \le \alpha \le 1} \Big\{ E[f(X \cdot e^{-\mu t} + \frac{\lambda(1 - e^{-\mu t})}{\mu \alpha}); X \le F^{-1}(\alpha)] + E[f(X \cdot e^{-\mu t}); X \ge F^{-1}(\alpha)] \Big\}, \quad (5)$$

or equivalently, as

$$\min_{0 \le y \le c} \Big\{ E[f(X \cdot e^{-\mu t} + \frac{\lambda(1 - e^{-\mu t})}{\mu F(y)}); X \le y] + E[f(X \cdot e^{-\mu t}); X \ge y] \Big\}, \quad (6)$$

[1]We opt to focus on this scheme, due to the simple cyclical implementation it implies in a highly dynamic environment, as opposed to, *e.g.,* routing weighted shares of traffic to different links, in which case a set of weights have to be *dynamically* maintained.

[2]This scaling might correspond to the practical context where additional *wavelengths* are added to an optical fiber to increase its total capacity. Each of these added wavelengths can be thought of as an additional link in our model.

where the optimal decision variables α^* and y^* are related by $\alpha^* = F(y^*)$. Here F is a continuous distribution on $[0, c]$ modeling the link loads on the parallel-link network. ■

Observe that the theorem suggests that it is *asymptotically* equivalent to use the $\alpha^* \cdot n$ least loaded links or all the links with load less than y^*, where α^* and y^* are the optimizers of problems (5) and (6), respectively. This follows by a simple change of variables. However, it is worth pointing out that in practice these correspond to two different *modes* of routing, *i.e.,* routing on $\alpha^* \cdot n$ least loaded links vs. routing on a set of links whose loads are below a threshold y^*.

Assuming the prior distribution on link loads are independent and uniform, the optimal choices for α^* and y^* can be determined. The proof of the following fact can be found in the appendix.

FACT 1. *Suppose the link loads are uniformly distributed on $[0, c]$, then the minimizers for (5) and (6) are respectively*

$$\alpha^* = \min\{\sqrt{\frac{\lambda}{\mu c}\frac{(1-e^{-\mu t})}{e^{-\mu t}}}, 1\},$$

and

$$y^* = \min\{\sqrt{\frac{\lambda c}{\mu}\frac{(1-e^{-\mu t})}{e^{-\mu t}}}, c\}. \qquad ■$$

Note that for uniformly distributed link loads, the optimal parameters α^* and y^* is not sensitive to the exact form of the system congestion measure f, it need only be increasing and convex.

2.3 Observations

Based on Fact 1 we can make a number of fairly interesting observations.

As the traffic arrival rate λ increases, or the mean flow holding time μ^{-1} decreases, or λ increases for a fixed offered load $\rho = \lambda/\mu$, one should disperse the traffic flows over a *larger* set of paths. Hence as an engineering guideline, it makes sense to differentiate the operational parameters at various network nodes with different types of incoming traffic requests. In particular, a network node should actively route its connection requests over *a large set of paths* if these requests are mostly *short and arrive frequently*, or direct its connection requests to the *least loaded* path if these requests are mostly *long and arrive infrequently.*

The intuition for the above observation, lies in that the system congestion measure $s(t)$ is a symmetric sum of increasing convex functions. This suggests that to approach optimality one should route the traffic flows so that link loads at time t are balanced. From (2) we see that when if the flow departure rate is large the differences in the *initial* link loads should be "discounted" and we should "perceive" all

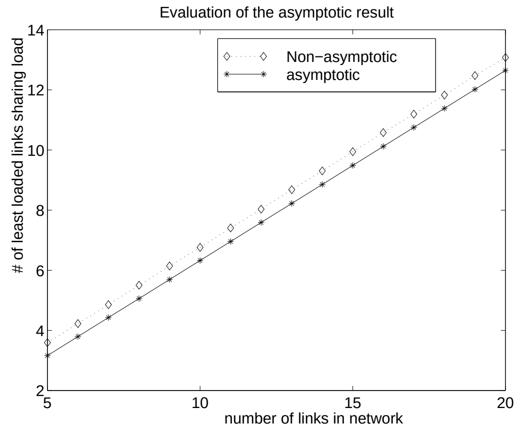

Figure 2: The least loaded links used to spread traffic: increasing total links.

link loads as approximately "equal." In this context the natural routing decision to make, in order to bring the link loads at time t close to each other, is to spread the incoming traffic flows over a large set of links.

Suppose one scales the offered load ρ (while keeping mean flow holding time μ^{-1} invariant) and link capacity c in proportion, then the optimal fraction of links over which one should route the traffic remains unchanged. This suggests that in practice if we build up the network capacity and the traffic load grows in proportion then α^* and y^*, *i.e.,* the range of paths over which we route traffic flows, remain invariant. On the other hand, we should adjust the range of multi-path routing mechanism if the rate of capacity expansion does not match that of the traffic growth. More specifically, one might need to modify the operational parameters α^* and y^* if user traffic outgrows the network capacity.

Now suppose the time scale of interest t grows, *e.g.,* one has to limit link state advertisements. Notice that as t increases the impact of the initial link load diminishes. Intuitively, if this is the case, the optimal allocation is to spread the load evenly among a large set of links. This is verified by the result in Fact 1.

In practice, we might not only have incomplete knowledge of the link states but also of the arrival rate λ. From Fact 1 we see that the optimal parameters α^* and y^* are square root functions of λ. This suggests that these optimal parameters are relatively insensitive to the exact value of λ as the routing diversity increases, thus a reasonable estimate may suffice.

2.4 Validation Of The Asymptotic Result

In the previous section we obtained an asymptotic result concerning the number of links over which to disperse traffic, in a regime where the load and number of links (disjoint paths) grow in proportion. Here we evaluate the quality of the result, in the case where n is small, or modest, via sim-

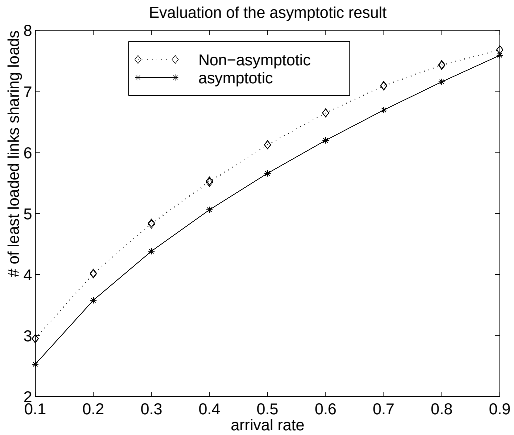

Figure 3: The least loaded links used to spread traffic: increasing arrival rates.

ulation. Suppose $X \sim \text{Uniform}[0, 10]$, $\lambda = 4$, $t = 1$, and $\mu = 0.1$. We compare the number of least loaded links that would be selected based on the asymptotic model versus the corresponding values obtained by successively sampling the link loads and computing the averaged optimal number of links over which to disperse traffic. In Figure 2 we exhibit the comparison between theoretical/asymptotic and simulated average optimal choices, in terms of the optimal number of least loaded links over which to route the traffic flows. We observe that the number of least loaded links obtained via the asymptotic formula is within 10% of the average obtained via simulation, where the number of links n ranges from 5 to 20. In Figure 3 we evaluate the effectiveness of Fact 1. Here we fix the number of links $n = 8$, $t = 1$, $\mu = 0.1$, $X_i \sim \text{Uniform}[0, 10]$, and increase λ from 1 to 9. The asymptotic prediction matches its simulated counterpart to a satisfactory degree. We conclude that the asymptotic result provides a reasonable approximation to select the k links over which to disperse traffic flows.

2.5 Realizing Dynamic Multi-path Routing — Three Alternatives

In the previous sections we explored a dynamic multi-path routing scheme, where equal shares of the traffic flows were routed over a dynamically selected set of network links. An asymptotic analysis on how to dynamically select such links is embodied in (5) and (6). We can interpret the solutions to these optimization problems as suggesting two different schemes: routing over the $k = \alpha^* n$ least-loaded paths (or shortest paths, if we equate the *length* of a link to its load), or routing over all the paths that have a load less than y^*.

The implementation of the first scheme corresponds to the classic k-shortest path algorithm[19]. We name the first scheme DKSP, which is short for Dynamic k Shortest Paths. Note that this scheme spreads the traffic flows over k links/paths with potentially different loads. This is in contrast with the traditional least-loaded routing scheme, where one randomly

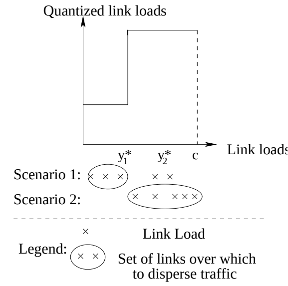

Figure 4: A two-level quantizer.

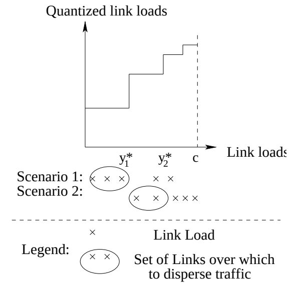

Figure 5: A multi-level quantizer.

selects a link only among those with least load.

For the second scheme, notice that for a given parameter y^* and a particular set of network loads there may not be any candidate links with load less than y^*. To address this problem, we consider two solutions that correspond to 1) a *pre-determined* link load quantization mechanism, named DQSP, which is short for Dynamic Quantized Shortest Paths, and 2) a *dynamic* threshold mechanism, named DTMP, which is short for Dynamic Threshold Multi-Paths.

DQSP can be interpreted as a link state quantization scheme. In particular, the ingress nodes (or the links) quantize the link loads based on a threshold y^*, and traffic is routed over the links with the least quantized load. Figure 4 illustrates two scenarios where each "$\times$" indicates the amount of load on a given link, and the circled links are those over which traffic flows will be spread based on the threshold y^*. It is evident that by quantizing the link load, we can increase the number of links that are "equally" loaded. Notice that under this scheme if there is no link with load less than y^*, we can use all the links to route the traffic flows. This can be refined as follows. Conditional upon all the link loads exceeding y^*, we can formulate a modified version of the previous optimization problem, and recursively obtain a sequence of quantization thresholds: $\hat{y}^* = (y_1^*, y_2^*, \cdots, y_n^*)$, where

$$y_1^* = y^*,$$
$$y_{i+1}^* = y_i^* + \sqrt{\frac{\lambda \cdot (c - y_i^*) \cdot (1 - e^{-\mu t})}{\mu \cdot e^{-\mu t}}}, \quad i = 1, 2, \cdots, n.$$

The procedure terminates at the index i where $y_i \leq c$ and $y_{i+1} > c$. With these quantization thresholds in place, the ingress node simply examines the current link loads and identifies the set of links with the least quantized load. Fig. 5 shows a multi-level link state quantizer, where we illustrate two loading conditions in which only the circled links are used in routing the traffic load.

Our DTMP scheme, is aimed at addressing the possible void of links with load less than y^* based on a dynamic threshold mechanism. Instead of routing traffic flows on all links with load less than y^* we route the traffic over all the links that do not have a load exceeding a multiple of the least loaded one. That is, we use all the links with load no more than

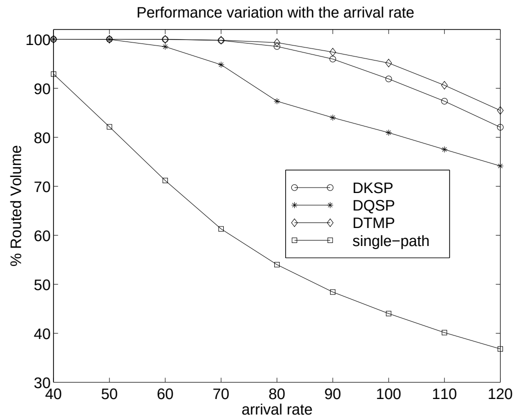

Figure 6: Performance comparison of the routing schemes: nominal load = 100.

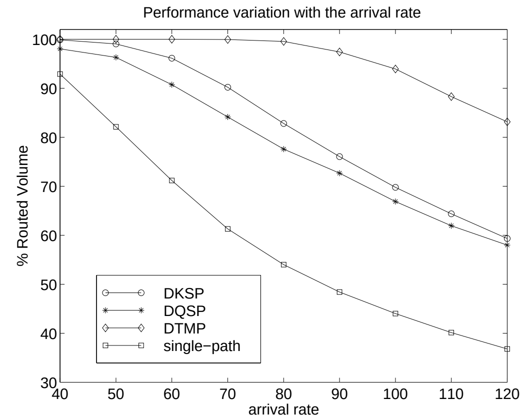

Figure 7: Performance comparison of the routing schemes: nominal load = 10.

$(1+\beta) \cdot \min_l x_l(0)$, where $x_l(0)$ is the load on link l at time 0 and β is a positive scaling factor. This guarantees that there is at least one link on which traffic can be routed, *i.e.,* the least loaded link(s). In the sequel we show by simulation that the DTMP scheme performs well against the other two schemes, especially in the operating regime where the number of links n is modest and the arrival rate at the ingress node is not accurately modeled, *e.g.,* it may vary. We might however expect these routing schemes to be equivalent in the asymptotic regime considered in Section 2.2. By considering the associated asymptotic regimes one can show that β should be approximately set to $\alpha^* \cdot n$.

Let us assess the performance achieved by setting $\beta^* = \alpha^* \cdot n$. Specifically, we compare these values with the best β^* value obtained via simulation, where a collection of possible values for β^* was examined and that corresponding to the least flow blocking rate was identified. Consider a network with 12 parallel links, each with capacity 20 units. Traffic flows arrive according to a Poisson process with rate equal to 50 flows per second. The flow holding time is exponentially distributed with mean μ^{-1} and each flow requests one unit of bandwidth. We assume a periodic link state update mechanism with period t. Table 1 summarizes the comparison across a range of μ and t values. In the simulation β^* was incremented by 0.5 each step in the process of searching for the best value. It is fair to conclude that as a simple approximation, $\beta^* = \alpha^* \cdot n$ provides a crude, but reasonable setting for β^* resulting in good performance.

Table 1: The selection of β^*.

μ	t	$\beta^* = \alpha^* n$	β^* via simulation
0.25	0.5	3.6	4.5
0.25	1.0	5.4	6
0.1	1.0	8.4	9

2.6 Three Dynamic Multi-path Routing Schemes: A Comparison

In this section we present our simulations comparing the performance of the three routing schemes discussed above. As a base case, we will use a routing scheme that routes traffic to the least loaded link. The performance metric we use, is "% routed volume", that is the percentage of the bandwidth demand that is successfully routed. In the sequel we also use " % improved routed volume", which is defined as $\frac{x-y}{y} \cdot 100$, where x is the performance achieved by our dynamic multipath routing schemes, and y is that achieved by a dynamic single-path scheme. In order to investigate the robustness of the proposed schemes to varying arrival rates we will consider an operational scenario where the network is designed to carry traffic flows with a *nominal* arrival rate, but the *actual* flow arrival rate is different. Note that the performance of the routing schemes depends on the parameters α^*, y^* and β^*. Note that these parameters are all functions of the nominal flow arrival rate λ. Thus we wish to establish the sensitivity of the routing performance to the nominal λ, for the three schemes proposed above.

We first compare the performance of the three routing schemes. We set the parameters α^*, y^*, and β^* based on a nominal flow arrival rate equal to 100 flows per sec, and the actual flow arrival rate varies between 40 to 120 flows per sec. From Figure 6 we observe that DTMP scheme exhibits the most significant performance improvement over single path least loaded routing. In particular, the % improved routed volume for DTMP ranges from 7% to 136%, as the actual flow arrival rate increases.

Next we examine another case where network load was underestimated, *i.e.,* we set the parameters based on a nominal flow arrival rate equal to 10 flows per sec, and let the actual flow arrival rate varies between 40 and 120 flows per sec. From Figure 7 We observe that in this case with underestimated operational parameters, DTMP policy again performs adequately, while other schemes see a significant performance degradation. For example, if we compare Figure 6 and 7 at arrival rate 100 flows per sec, we see that the performance of DTMP remains almost unchanged while those of DKSP and DQSP degrade significantly.

3. DYNAMIC MULTI-PATH ROUTING IN MESH NETWORKS: AN APPROXIMATION

3.1 From Parallel Links To Mesh Networks: Extending DTMP

In the previous section we considered dynamic multi-path routing problem for a symmetric parallel-link network. It is difficult to extend these results to mesh networks. Specifically, the available routes between a pair of ingress and egress nodes are not necessarily disjoint, so there may be interactions among the traffic loads on various routes. Moreover, there are usually multiple pairs of ingress and egress nodes that make independent routing decisions based on network states, and these decisions may be synchronized, which in turn aggravates congestion in the network.

Let us consider routing a set of traffic flows on a mesh network $G(N, L)$ with a set of nodes N and a set of links L, so that an additive network congestion measure is minimized. Formally, suppose we have a set of ingress-egress node pairs S, a set of available routes R, and a 0-1 matrix H where $H_{sr} = 1$ if $s \in S$ can be served by route $r \in R$, and $H_{sr} = 0$ otherwise. Let $s(r)$ denote the set of routes r that serve flow s, *i.e.,* $H_{sr} = 1$. Moreover, let us define a matrix A such that $A_{jr} = 1$ if route $r \in R$ passes through link $j \in L$. Let us model the network dynamics with a fluid approximation. Suppose the traffic flows arrive with rate g_s, the mean flow holding time μ^{-1} is set to 1, and each flow transmits at unit rate. We define the routing objective as follows:

$$\min_{\lambda_r, r\in R} \sum_{i\in L} \int_0^{x_i} z(y)dy$$
$$\text{s.t.} \quad H\lambda = g, A\lambda = x,$$

where $f(x_i) = \int_0^{x_i} z(y)dy$ is a convex function of x_i, $g = (g_s, s \in S)$ is the vector of the flow arrival rates (or offered load in our setup), and $x = (x_l, l \in L)$ is the vector of the link loads. The solution to this network flow problem can be characterized as follows:

$$\lambda_r > 0 \Rightarrow \sum_{i\in r} z(x_i) \le \sum_{i\in r'} z(x_i), \forall r' \in s(r).$$

i.e., only the *shortest* paths where link lengths are $z(x_i)$, will carry *positive* amounts of flow. This is known as a Wardrop equilibrium [7]. As a special case, if we were to minimize the network congestion measure given by $-\sum_{l\in L} \log(c_l - x_l)$, the link cost on a link l with capacity c_l becomes $z(x_l) = 1/(c_l - x_l)$, *i.e.,* the inverse of the available bandwidth. In later sections we will use $1/(c_l - x_l)$ as link metric.

The Wardrop equilibrium suggests that one should route the traffic in such a way that only the "shortest" paths carry positive amounts of flow. However, this is meaningful only in a static or quasi-static network scenario. In the highly dynamic environment we consider in this study, where traffic flows arrive and depart quickly(*e.g.,* less than link state updating period), link loads often exhibit bursty changes, and the link state information used to compute the "shortest" paths is often outdated. Hence the "perfect load-balancing" suggested by the Wardrop equilibrium is neither practical nor achievable[18]. Instead of restricting ourselves to shortest paths alone and trying to adapt to the exact flow proportions, we propose to randomly route the traffic flows between an ingress-egress node pair s, among all the paths with length no more than $(1+\beta^*)\cdot l_s$, where l_s is the length of the shortest path associated with the node pair s, and $\beta^* > 0$ is a design parameter to be determined.

This approximation is similar to the DTMP scheme proposed for the parallel-link model. We will again use "DTMP" to refer to this approximation scheme for mesh networks. Notice that in the mesh network setup, the set of paths which are selected may not be disjoint, hence some links may be traversed by several paths used for routing the traffic between a given ingress and egress node. The load on these links could "build up." The dynamic aspect of our scheme, *i.e.,* choosing the paths whose length is within a certain range of the shortest path length, helps to avoid this build-up process, as long as the dynamic link metrics, *i.e.,* $1/(c_l - x_l)$ reflect the link loads on the network.

Notice that by letting the length of the paths over which one disperses traffic be dependent upon the shortest path length l_s, we achieve the following intuitive behavior: if the network is lightly loaded, it is beneficial to consistently use only the shortest paths, whose unused capacity is high; if the network is more congested, it is advantageous to spread the load over a larger set of paths in order to accommodate the incoming (bursty) flows. In the next section we examine various aspects of this routing scheme via simulation. Based on our simulations we made the following observations:

1. The DTMP scheme outperforms *dynamic* single-path routing, *i.e.,* least loaded routing (LLR).

2. In networks with "hot-spots" DTMP offers more significant performance improvement than in networks with "balanced" traffic, thus if such hot spots arise one can might resort to DTMP to alleviate the impact of congestion.

3. If traffic flows arrival processes are bursty, DTMP provides a greater performance gain over its single-path counterpart.

4. As the portion of co-located traffic, *i.e.,* traffic between nodes that are one hop away from each other, increases the performance gains from using DTMP decrease.

5. In the network where link state updates are relatively slow as compared to flow arrivals/departures, DTMP offers significant performance improvement.

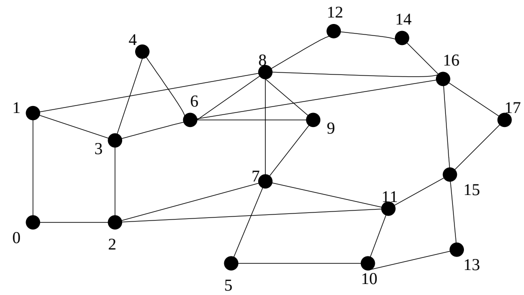

Figure 8: NSF topology.

6. As we scale up the capacity of the network, the use of DTMP scheme is more important as it offers greater performance gains.

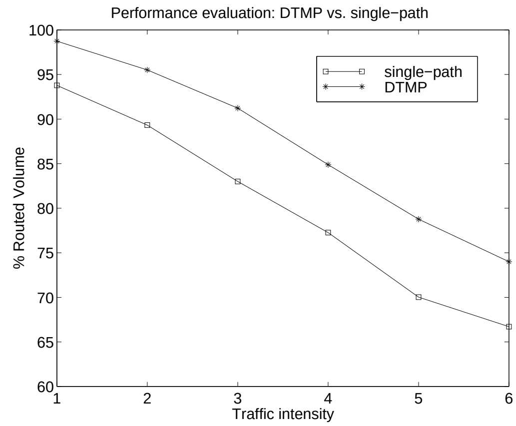

Figure 9: DTMP vs. single path.

Table 2: The traffic matrix.

ingress node	egress node	hop distance	arrival rate
0	16	4	1
1	17	3	1
4	13	4	1
5	14	4	1
8	10	3	1

3.2 Simulation Setup

We present a set of results for the network shown in Figure 8. In the simulation, the flows arrive to the network according to a Poisson process, and the flow holding times are Pareto distributed. The ingress and the egress nodes of the flows are selected according to Table 2, which are set up to model a typical WAN traffic pattern, *i.e.,* the ingress and egress nodes of a flow are at least three hops away from each other. In the sequel we will examine the effect of this setup, and evaluate the impact of the "co-located" ingress and egress nodes, *i.e.,* within two hops or less. The parameters for the simulation were set as follows: link capacity is 25 units, mean flow holding time is 12 msec, and the bandwidth request of each flow is uniformly distributed between 0.5 and 1.5 units. This 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 numbers, shown on the horizontal axis, see Figure 9. Unless explicitly stated, we use dynamic link metric $1/(c_l - x_l(t))$, and the routers exchange link states periodically, with an updating period of 10 msec.

3.3 Performance Evaluation: Is DTMP Routing Effective?

We first compare our DTMP with dynamic single path routing. The performance improvement of the DTMP scheme is evident from Figure 9. Specifically, the relative performance improvement ranges from 5% to 13%, as the traffic load grows.

In the above simulation β^* was set to 1.6. In general, it is hard to pin-point the best β^*. It depends on network topology, traffic demands, as well as various timing factors involved in the network, *e.g.,* flow arrival rate, flow holding time, and link state updating period. Our experience, however, suggests that DTMP's performance is quite robust to the choice of β^*. For the set of simulations we conducted it was observed that in the interval $0.2 \leq \beta^* \leq 3.2$ our multi-path routing scheme outperforms single-path routing scheme. However we did see the performance degradation caused by excessive multi-path routing, in the cases where $\beta^* > 3.2$. We conjecture that in practice it is relatively easy to tune β^* to achieve good overall performance.

From these experiments we conclude that this simple dynamic multi-path routing scheme works well at improving performance over the traditional dynamic single-path routing. The performance of the proposed scheme is relatively robust to the choice of parameter β^*. However, we note that one should not be overly aggressive in setting a high value for this parameter.

To further evaluate the effectiveness of the DTMP scheme, we vary the flow arrivals to the network so that certain "hot-spots" are present. Specifically, we increase the arrival rate from Node 1 to Node 17 to 3 flows per msec, and decrease the arrival rates to other pairs of ingress-egress nodes to 0.5 flows per msec. Figure 10 compares the performance gain, *i.e.,* the improvement of the performance achieved by DTMP over the single-path routing, between the case with traffic matrix in Table 2 and the case here with "hot-spot" traffic. A more significant performance improvement (12-21%) is evident when "hot-spots" are present, as compared with the base case (5-13%). Hence we maintain that DTMP is conducive to alleviating the impact of the "uneven" network loads.

As observed in practice, even traffic flow arrivals themselves may be bursty, *e.g.,* the access to CNN web site before and after a major news event. We believe that in such an operational scenario, DTMP can deliver more significant performance improvements over its dynamic single-path counterpart. In the previous simulations we modeled the flow arrivals by a Poisson process, which is generally considered a "smooth" random process. To model bursty flow arrivals, we used the Markov Modulated Poisson Process (MMPP) illustrated in Figure 11. 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 3 times the mean given in Table 2. For the second, the flow arrival rate in the "high" state is 1.5 times the mean given in Table 2. In the "low" state, the traffic flows arrive with rate 1/3 of the mean given in Table 2, for both MMPPs. Besides the rates associated with the "high" and "low" states, the MMPPs are also characterized by the average holding time at "high" and "low" states. For the first MMPP, we set the average holding time at "high" state and "low" state to be $0.5 \cdot$ MMPP_TIME and $1.5 \cdot$ MMPP_TIME, respectively, where MMPP_TIME is a scaling variable which we vary from 10 to 90 msec. For the second MMPP, we set the average holding time at both modulating states to be MMPP_TIME.

In Fig. 12 we illustrate the performance improvement achieved when such bursty arrival processes are present. It is evident that the DTMP scheme is more effective in a network supporting bursty arrivals processes. In addition, note that when MMPP_TIME equals to 50 msec the performance gains are the highest. This suggests an optimal time scale for which DTMP is most effective. The intuition is as follows: when the MMPP_TIME is small, the "high" and "low" states alternate frequently relative to the link state updates, hence the routing decisions that have to be made in "high" state by the dynamic single path routing scheme "averaged out" with those in "low" state. If MMPP_TIME is large the network states get updated often enough to track changes in the traffic. The "critical" time scale, however, is the one where a burst of flows arrive in "high" state and the updates are not quite frequent enough for the single path routing scheme to track such changes. At this "critical" time scale DTMP provides the most performance improvement over single path routing scheme.

Next let us examine the impact on network performance of traffic locality with respect to the ingress and egress nodes. The traffic arrival pattern in the above simulations roughly models a WAN. One might ask what happens if a significant amount of traffic is between network nodes that are "colocated", *i.e.,* having direct links to each other? Notice that in the topology under consideration, the closer the ingress and egress nodes, the fewer paths there are that have similar characteristics in terms of hop count. Intuitively, this implies that we have fewer *options* over which to support the proposed multi-path routing scheme. Hence we should see a decrease in the performance improvement achieved by DTMP over its single-path counterpart. To verify this intuition, we introduce additional traffic between Nodes 6 and 16, and also between Nodes 7 and 11, each with rate 1 flows per msec, while decreasing the flow arrival rates associated

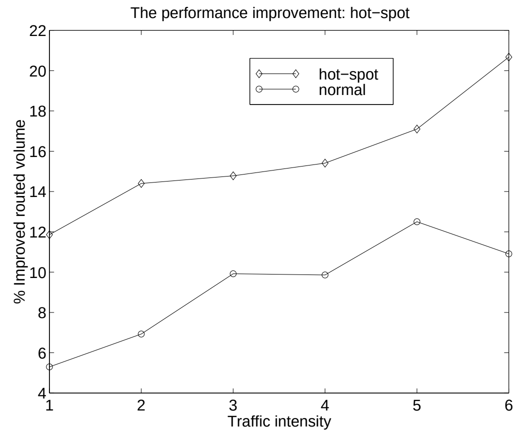

Figure 10: The effect of the hot-spot traffic.

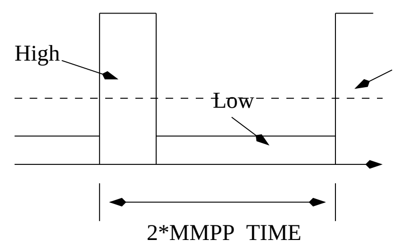

Figure 11: Markov Modulated Poisson Process

with the other node pairs in Table 2 to 0.6 flows per msec. This is done so that the total flow arrival rate to the network is kept to be 5 flows per msec. The results in Fig. 13 support this insight.

The performance of the DTMP routing scheme depends on the quality of the set of paths over which dispersion will take place. The quality of a path is captured by its length, which in turn depends on timely link metrics. In our next simulation we considered how often updates would be gen-

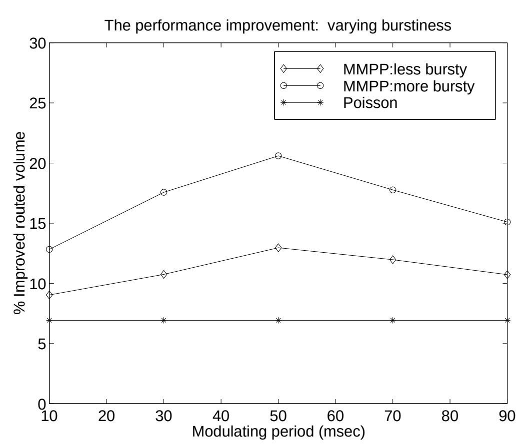

Figure 12: The effect of the bursty traffic.

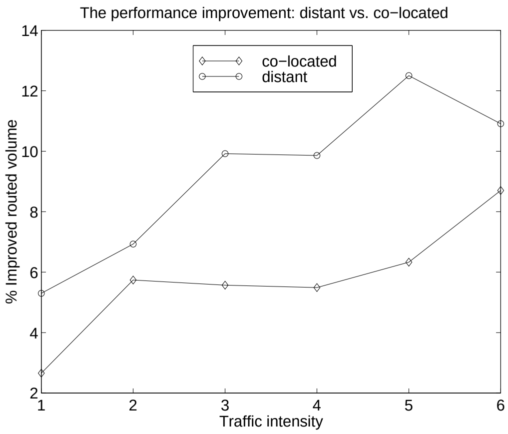

Figure 13: The effect of the local traffic.

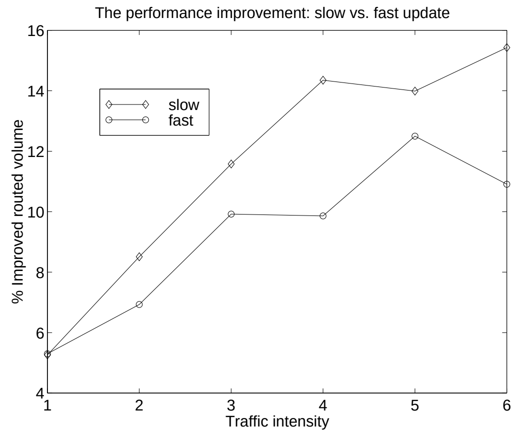

Figure 14: The impact of the link state update period.

erated, namely "slow-updates" and "fast-updates". [3] Here by "slow-updates" we refer to an operating regime where link metrics are updated every 1 sec, and by "fast-updates" we refer to an operating regime where link metrics are updated every 10 msec. This distinction in link state updating rate may correspond to networks with different geographical coverage, *i.e.,* long vs. short-haul networks, or simply limitations on signaling overheads. As seen in Figure 14, for "slow-updates" the performance improvement is more significant. The reason is that for "slow-updates", the unevenness and/or buildup in network loads are more pronounced in the single-path routing scheme due to the longer delay in link state update. Hence our DTMP scheme alleviates the impact of delays in distributing link state information.

Finally we examined the impact on the routing performance as the capacity of the network is increased. Let us denote the network used in the previous discussions by NET-SMALL, and construct a new network, NET-BIG, that has the same topology as NET-SMALL but 100 times the link capacity. In order to derive a meaningful comparison, we scaled the flow arrival rates to NET-BIG to be 250 times those of NET-SMALL. A comparison of the performance improvements achieved by DTMP is shown in Figure 15.

We observe that performance improvement brought about by the DTMP mechanism are more significant in the network with large capacity. The reason is that with delays in link updating, single-path routing is somewhat oblivious to the network load condition, which leads to poor load balancing on the network. The key point is that such imbalances are more pronounced in the large capacity network and a multi-path routing scheme like ours is able to alleviate this problem more substantially.

Note that in the above simulations we opted to *linearly* scale the flow arrival rate and the network capacity. To further generalize this result, we also experimented with other scaling schemes. In particular, we scaled the flow arrival rates so that in both networks dynamic *single-path* routing scheme achieved roughly the same performance. We compared the performance improvement attained by DTMP schemes on these two networks and found that in networks with larger arrival rate and link capacity the DTMP scheme again achieved a more significant performance improvement over dynamic single-path routing scheme.

[3] We are using a simple periodic updating scheme. Other mechanisms exist and a comparison study can be found in [1].

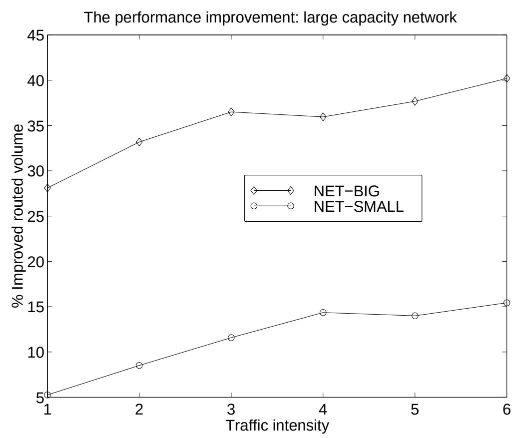

Figure 15: The impact of the network capacity.

4. CONCLUSION

In this paper we studied dynamic multi-path routing. We formulated a stochastic optimization problem in a parallel-link network model. We analyzed a set of routing policies intended to optimally select the links over which to disperse traffic flows. For a limiting regime we exhibited an associated optimization problem which permits the closed-form analysis. These results provide a number of insights addressing the interaction among traffic arrivals, flow holding time, link capacity, and network updating time scales. In particular, we identified a robust dynamic multi-path routing scheme, *i.e.,* the DTMP scheme, that performs well in various network environments.

We then extend the findings to networks with mesh topologies. We adapted the DTMP scheme in this context and conducted extensive simulations to examine its performance, including the impact of the link state updating rate, burstiness in traffic arrivals, and various issues concerning traffic load distribution. Based on our simulations we believe we have identified a robust dynamic multi-path routing scheme that can be used effectively to route/disperse traffic in high speed networks.

APPENDIX

A. PROOF OF THEOREM 1

To prove Theorem 1, we will use the following three lemmas.

LEMMA A.1. *Suppose $X_1, X_2, \cdots$ are iid. uniform random variables on the interval [0,1]. Let $X^n_{(\lceil n\alpha\rceil)}$ be the $\lceil n\alpha\rceil$ order statistic based on the first n random variables in the sequence. Then $X^n_{(\lceil n\alpha\rceil)} \xrightarrow{a.s.} \alpha$.*

Proof : By definition,

$$X^n_{(\lceil n\alpha\rceil)} \xrightarrow{a.s.} \alpha \quad \text{iff} \quad P(\{\omega \in \Omega | \lim_{n\to\infty} X^n_{(\lceil n\alpha\rceil)}(\omega) = \alpha\}) = 1.$$

Now $X^n_{(\lceil n\alpha\rceil)}(\omega) \to \alpha$ if $\forall\epsilon > 0$, $\exists m > 0$, such that $\forall n > m$, $\alpha - \epsilon \le X^n_{(\lceil n\alpha\rceil)}(\omega) \le \alpha + \epsilon$. Note that

$$\begin{aligned} & X^n_{(\lceil n\alpha\rceil)}(\omega) \le \alpha + \epsilon \\ \Longleftrightarrow & \sum_{i=1}^{n} 1\{X_i(\omega) \le \alpha + \epsilon\} \ge \lceil n\alpha\rceil \\ \Longleftrightarrow & \frac{1}{n} \cdot \sum_{i=1}^{n} 1\{X_i(\omega) \le \alpha + \epsilon\} \ge \frac{\lceil n\alpha\rceil}{n}. \end{aligned}$$

By the Strong Law of Large Numbers,

$$\lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} 1\{X_i(\omega) \le \alpha + \epsilon\} = \alpha + \epsilon \ge \alpha = \lim_{n\to\infty} \frac{\lceil n\alpha\rceil}{n},$$

so

$$\exists m_1, \forall n > m_1, X^n_{(\lceil n\alpha\rceil)}(\omega) \le \alpha + \epsilon, \forall\omega.$$

Similarly, $\exists m_2, \forall n > m_2, X^n_{(\lceil n\alpha\rceil)}(\omega) \ge \alpha - \epsilon, \forall\omega$. Whence $X^n_{(\lceil n\alpha\rceil)}(\omega) \to \alpha, \forall\omega$ and $X^n_{(\lceil n\alpha\rceil)} \xrightarrow{a.s} \alpha$.

■

LEMMA A.2. *If $X_1, X_2, \cdots$ are iid random variables with distribution function F, where F is continuous and has a finite support $[0, c]$, then the order statistics are such that $X^n_{(\lceil n\alpha\rceil)} \xrightarrow{a.s.} F^{-1}(\alpha)$.*

Proof: Since $X \sim F(x)$, $F(X) \sim U(0, 1)$. By the continuity assumption, we have that $\forall\epsilon > 0, \quad \exists\epsilon' > 0$,

$$\begin{aligned} & P(\{\omega \in \Omega | \lim_{n\to\infty} X^n_{(\lceil n\alpha\rceil)}(\omega) = \lim_{n\to\infty} F^{-1}(\alpha) \\ & = P(\{\omega \in \Omega | \lim_{n\to\infty} F(X^n_{(\lceil n\alpha\rceil)})(\omega) = \alpha) = 1. \end{aligned}$$

The last step follows from the fact that F is increasing, thus $F(X^n_{(i)})$ is the i-th order statistic of a uniformly distributed random variable $F(X)$. By definition we obtain almost surely convergence.

■

LEMMA A.3. *For the continuous function f, if $X \sim F$, then $\lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{\lceil n\alpha\rceil} f(X^n_{(i)}) = E[f(X) \cdot 1\{X \le F^{-1}(\alpha)\}]$.*

Proof: This follows from Lemmas A.1 and A.2. Details are omitted.

■

Proof of Theorem 1: By Lemma A.3, we have

$$\begin{aligned} & \min_{0\le\alpha\le1} \lim_{n\to\infty} \frac{1}{n} \cdot E[\sum_{i=1}^{\lceil\alpha n\rceil} [f(X^n_{(i)} e^{-\mu t} + \frac{\lambda}{\alpha\mu}(1 - e^{-\mu t})) \\ & \quad -f(X^n_{(i)})] + \sum_{i=\lceil\alpha n\rceil+1}^{n} [f(X^n_{(i)} e^{-\mu t}) - f(X^n_{(i)})]] \\ = & \min_{0\le\alpha\le1} E[f(X \cdot e^{-\mu t} + \frac{\lambda}{\mu\alpha}(1 - e^{-\mu t})) \\ & \quad 1\{0 \le X \le F^{-1}(\alpha)\}] \\ & \quad +E[f(X \cdot e^{-\mu t})1\{(X \ge F^{-1}(\alpha)\}] \\ & \quad -E[f(X)1\{(0 \le X \le F^{-1}(\alpha)\}] \\ & \quad -E[f(X)1\{X \ge F^{-1}(\alpha)\}], \\ = & \min_{0\le\alpha\le1} E[f(X \cdot e^{-\mu t} + \frac{\lambda}{\mu\alpha}(1 - e^{-\mu t})) \\ & \quad 1\{0 \le X \le F^{-1}(\alpha)\}] \\ & \quad +E[f(X \cdot e^{-\mu t})1\{X \ge F^{-1}(\alpha)\}] \\ & \quad -E[f(X)1\{(0 \le X \le F^{-1}(\alpha)\}] \\ & \quad -E[f(X)1\{X \ge F^{-1}(\alpha)\}], \\ = & \min_{0\le\alpha\le1} E[f(X \cdot e^{-\mu t} + \frac{\lambda}{\mu\alpha}(1 - e^{-\mu t})) \\ & \quad 1\{0 \le X \le F^{-1}(\alpha)\}] \\ & \quad +E[f(X \cdot e^{-\mu t})1\{X \ge F^{-1}(\alpha)\}] - E[f(X)], \end{aligned}$$

which is equivalent to

$$\begin{aligned} & \min_{0\le\alpha\le1} E[f(X \cdot e^{-\mu t} + \frac{\lambda}{\mu\alpha}(1 - e^{-\mu t})); \qquad (7) \\ & \quad 0 \le X \le F^{-1}(\alpha)] + E[f(X \cdot e^{-\mu t}); X \ge F^{-1}(\alpha)] \end{aligned}$$

By a change of variable, $\alpha = F(y)$, we have that (7) is equivalent to

$$\begin{aligned} & \min_{0\le y\le c} E[f(X \cdot e^{-\mu t} + \frac{\lambda}{\mu F(y)}(1 - e^{-\mu t})); \\ & \quad 0 \le X \le y] + E[f(X \cdot e^{-\mu t}); X \ge y] \end{aligned}$$

■

B. PROOF OF FACT 1

Proof: If $X \sim \text{Uniform}[0,c]$, the first order optimality condition for (6) is given by:

$$\begin{aligned} & f(ye^{-\mu t} + \frac{\lambda c}{\mu y}\cdot(1-e^{-\mu t})) - f(ye^{-\mu t}) \\ = \quad & \frac{\lambda c\cdot(1-e^{-\mu t})}{y^2\cdot\mu\cdot e^{-\mu t}}\cdot[f(ye^{-\mu t}+\frac{\lambda c}{\mu y}\cdot(1-e^{-\mu t})) \\ - \quad & f(\frac{\lambda c}{\mu y}\cdot(1-e^{-\mu t}))]. \end{aligned}$$

Let $\eta = e^{-\mu t}$ and $\theta = \frac{\lambda c}{\mu}\cdot(1-e^{-\mu t})$, then we can write this condition as

$$f(\eta y + \frac{\theta}{y}) - f(\eta y) = \frac{\theta}{\eta y^2}\cdot[f(\eta y+\frac{\theta}{y}) - f(\frac{\theta}{y})],$$

or equivalently,

$$\frac{f(\eta y+\frac{\theta}{y}) - f(\eta y)}{\frac{\theta}{y}} = \frac{f(\eta y + \frac{\theta}{y}) - f(\frac{\theta}{y})}{\eta y}, \tag{8}$$

since we assume f is convex it follows that there exists a unique solution y^+ to (8) and $\eta y^+ = \frac{\theta}{y^+}$, or equivalently, $y^+ = \sqrt{\frac{\theta}{\eta}}$.

Now the optimizer y^* is either the stationary point y^+ or a boundary point 0 or c. Note that

$$E[f(X\cdot e^{-\mu t} + \frac{\lambda c}{\mu y}(1-e^{-\mu t}); 0 \le X \le y)] \to \infty$$

as $y \to 0$, so $y^* = \min\{y^+, c\}$ and $\alpha^* = y^*/c$.

■

C. REFERENCES

[1] G. Apostolopoulos, R. Guérin, and S. Tripathi. Quality of service based routing: A performance perspective. In *Proc. ACM Sigcomm*, pages 17–28, 1998.

[2] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Networking*, 1(4):397–413, Aug. 1993.

[3] R. Gibbens. Dynamic routing in fully connected networks. *IMA Journal of Mathematical Control and Information*, 7:77–111, 1990.

[4] E. Gustafsson and G. Karlsson. A literature survey on traffic dispersion. *IEEE Network*, pages 28–36, March/April 1997.

[5] V. Jacobson. Congestion avoidance and control. In *Proc. ACM Sigcomm*, pages 314–329, Aug. 1988.

[6] F.P. Kelly. Loss networks. *Ann. Appl. Prob.*, 1:317–378, 1991.

[7] F.P. Kelly. Network routing. *Proc. R. Soc. Lond. A*, 337:343–367, 1991.

[8] S.A. Lippman. Applying a new device in the optimization of exponential queuing systems. *Operations Research*, 23(4):687–710, 1975.

[9] N.F. Maxemchuk. Dispersity routing. In *Proc. ICC'75*, pages 41.10–41.13, June 1975.

[10] N.F. Maxemchuk. Dispersity routing on ATM networks. In *Proc. IEEE Infocom*, volume 1, pages 347–357, 1993.

[11] V. Nguyen. On the optimality of trunk reservation in overflow processes. *Probability in the Engineering and Informational Sciences*, 5:369–390, 1991.

[12] R.Gibbens, P. Hunt, and F. Kelly. Bistability in a communications network. In G. Grimmett, editor, *Disorder in Physical Systems*, pages 113–128. Claredon Press, 1990.

[13] S. Sibal and A. DeSimone. Controlling alternate routing in general-mesh packet flow networks. *Proc. of SIGCOMM94*, pages 168–179, 1994.

[14] I. Stoica and H. Zhang. LIRA: an approach for service differentiation in the internet. In *NOSSDAV*, pages 115–128, 1998.

[15] S.R.E. Turner. The effect of increasing routing choice on resource pooling. *Probability in the Engineering and Informational Sciences*, 12:109–124, 1998.

[16] C. Villamizar. MPLS Optimized Multipath (MPLS-OMP). *Internet Draft*, Nov. 1998.

[17] C. Villamizar. OSPF Optimized Multipath (OSPF-OMP). *Internet Draft*, Feb. 1999.

[18] S. Vutukury and J.J. Garcia-Luna-Aceves. A simple approximation to minimum-delay routing. In *Proc. ACM Sigcomm*, pages 227–238, 1999.

[19] J.Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17:712–716, 1971.