

Opportunistic Traffic Scheduling Over Multiple Network Paths

Coskun Cetinkaya
Wichita State University
Wichita, KS 67250-0044
coskun.cetinkaya@wichita.edu

Edward W. Knightly
Rice University
Houston, TX 77005
knightly@rice.edu

Abstract—Multipath routing enables a network’s traffic to be split among two or more possibly disjoint paths in order to reduce latency, improve throughput, and balance traffic loads. Yet, once the control plane establishes multiple routes, a policy is needed for efficiently splitting traffic among the selected paths. In this paper, we introduce Opportunistic Multipath Scheduling (OMS), a technique for exploiting short term variations in path quality to minimize delay, while simultaneously ensuring that the splitting rules dictated by the routing protocol are satisfied. In particular, OMS uses measured path conditions on time scales of up to several seconds to opportunistically favor low-latency high-throughput paths. However, a naive policy that always selects the highest quality path would violate the routing protocol’s path weights and potentially lead to oscillation. Consequently, OMS ensures that over longer time scales relevant for traffic management policies, traffic is split according to the ratios determined by the routing protocol. We develop a model of OMS and derive an asymptotic lower bound on the performance of OMS as a function of path conditions (mean, variance, and Hurst parameter) for self-similar traffic. An example finding from the model is that long-time-scale traffic fluctuations represented by a larger Hurst parameter improve the performance gain of OMS vs. round-robin scheduling, even under paths that are statistically identical. Finally, we use an extensive simulation-based performance study to evaluate the accuracy of the analytical model, explore the impact of OMS on TCP throughput, and study the impact of factors such as delayed measurements.

I. INTRODUCTION

Multipath routing establishes multiple network paths between pairs of routers to provide more efficient load balancing and higher-performance paths as compared to unipath routing. In practice, multipath routing is implemented via Equal Cost Multi-path as specified in [1], an extension to OSPF that establishes multiple paths with identical hop count.

Once multiple routes are established, the ingress traffic splitter (the router at the initial branch point) requires a policy to determine how to allocate individual packets to the paths. Round Robin (RR) allocation of packets among paths is the most commonly deployed policy due to its simplicity. A second scheme is to divide traffic according to a hash function applied to the source and destination pair, possibly including port numbers and protocol ID, e.g., [2]. This has the advantage of having each TCP micro-flow follow the same path thereby improving TCP performance, as packets within a TCP flow are not

reordered. However, this policy requires computationally complex per-packet operations and is not guaranteed to result in the desired traffic splitting ratio as flow rates are not known in advance. A third policy specified in [1] divides traffic according to destination prefixes of the forwarding table. While also preventing packet reordering within TCP flows, it also results in unpredictable loads on each path, as the traffic to each prefix is not known in advance nor is it easily controllable, as the majority of traffic is often destined to a single prefix [3].

In any case, all three splitting policies ignore the relative quality of the paths in making the traffic splitting decision. In this paper, we introduce Opportunistic Multipath Scheduling (OMS) as a traffic splitting policy that opportunistically favors low-delay high-throughput paths while simultaneously ensuring that the traffic splitting ratios defined by the routing policy are satisfied. In particular, routes and path weights can be expected to change on the timescale of minutes due to load balancing policies, faults, and other factors. However, path conditions are continuously changing due to traffic burstiness and traffic dynamics. OMS monitors one-way path delays and exploits differences in path conditions at moderate time scales (e.g., 10s of msec to seconds) to schedule packets over the highest quality paths.¹ Moreover, OMS ensures that over longer time scales (e.g., greater than seconds) the fraction of bytes transmitted on each path satisfies the path ratios defined by the multipath routing algorithm, ensuring that the network’s traffic engineering and load balancing policies are satisfied.

OMS is inspired by an analogous wireless scheduling problem [4], [5], [6], [7]. In wireless networks, each user’s channel condition is continuously varying due to fading and mobility. Wireless opportunistic scheduling refers to selection of the user with the best channel conditions while simultaneously ensuring that fairness constraints are satisfied over long time scales. Thus, algorithms such as developed in [4], [5], [6], [7] exploit high-quality channels when they occur, yet ensure that no user is starved due to perpetually poor channel conditions.

Our first contribution is to formulate the multipath scheduling problem as an optimization problem in which the traffic splitter seeks to select a path for each packet to minimize its queuing delay subject to satisfying the traffic splitting rules. We demonstrate OMS’ optimality and convergence to optimality via application of techniques developed in [6] despite the

This research is supported by NSF Grants ANI-0085842 and ANI-0099148, by a Sloan Fellowship, and by a gift from Intel Corporation.

¹We will show that clocks need not be synchronized despite our use of one-way delays.

significant differences in the multipath vs. wireless scheduling problems (selection of paths vs. users, paths contain queues vs. wireless channels, load balancing vs. fairness constraints, etc.).

Second, we devise an analytical model to characterize the performance of OMS as compared to Round Robin scheduling. In particular, we consider a system model consisting of a single bottleneck queue per path, and (self similar) fractional Brownian motion (fBm) fluid traffic inputs. While a highly simplified model of a realistic network, it allows us to characterize the performance of OMS and RR as a function of traffic parameters (mean rate, variance coefficient and Hurst parameter) and study several key aspects of the problem. For example, we derive an expression to characterize the performance gain of OMS vs. RR as a function of the Hurst parameter, and show that because increased long-time-scale traffic correlation (larger H) results in more highly variable path conditions, OMS has an increased opportunity to select better paths and provide further reduced delays as compared to RR.

Finally, we perform an extensive set of numerical investigations and ns-2 simulations. We first compare the results of the analytical model with simulations of fBm traffic and show that the model serves as a lower bound to the OMS vs. RR performance gain that is increasingly tight with higher Hurst parameter, mean rate, and variance coefficient. We then use simulations to explore the key performance factors that are not captured by the analytical model. For example, we show that even if information of the path conditions is delayed by as much as 100's of msec to one second, OMS can still achieve high performance provided that the Hurst parameter is sufficiently large (e.g., greater than 0.6). The reason is that the long-time-scale temporal correlation in traffic characterized by a large H results in long-time-scale correlation of the path conditions. Consequently, OMS' scheduling decisions are robust to moderate delays in obtaining the information on path quality. We also study the impact of OMS on TCP performance. As described above, multipath routing can potentially hinder the throughput of TCP flows due to misordered packets. We show that under round-robin scheduling and TCP NewReno [8] as well as TCP SACK [9], multipath TCP flows obtain goodputs as low as 20% of their ideal fair rate. In contrast, we show that under OMS scheduling, multipath TCP flows obtain throughputs of nearly 100% of their ideal fair rate, provided that the level of aggregation is sufficiently high (e.g., at least 10 TCP flows sharing a path). The key reason is that the reduced delay and optimized path selection of OMS increases TCP throughput in a way that overwhelms any adverse effects of occasional packet misordering.

The remainder of this paper is organized as follows. We first present a review of related work in multipath routing. In Section III, we describe the multipath scheduling problem formulation and devise OMS. Next, in Section IV, we devise an analytical performance model to study OMS and RR under fBm traffic. Finally, in Section V, we present the results of a simulation-based performance study of OMS and RR.

II. RELATED WORK

A significant literature addresses protocol design for computing multiple routes and their associated weights to minimize

delay and optimize use of network resources; see for example [10], [11], [12] and the references therein. Such work is complementary to OMS as OMS presupposes a control-path protocol to set up the routes: such a protocol may be a simple hop-count protocol with equal splitting [1] or more sophisticated distributed optimizations as in [11], [12].

The *control path* for multipath routing has also been studied in the context of connection-oriented networks with reservations; see for example [13], [14], [15], [16], [17] and the references therein. Protocols devised in this context seek to balance signaling overhead and the number of established paths with throughput optimization and quality-of-service constraints. In contrast, our work is more suited towards connectionless IP traffic in which flow traffic descriptors are not known in advance.

Next, there is a literature on the *forwarding path* of multipath routing that addresses techniques for traffic splitting. Hashing schemes [2], prefix matching schemes [1], and suffix matching schemes [18] have all been introduced to address the problem of misordered TCP packets. For example, the conclusion of [2] is that a 16-bit CRC of the complete five tuple of source and destination address and port along with protocol ID is required to ensure an equal traffic split. In Section V, we show that OMS' substantial delay reduction significantly increases TCP throughput and eliminates the need for ensuring that all packets within a TCP flow use a single path, thereby eliminating the need for computations associated with hashing and prefix matching.

Finally, sender side modifications of TCP have been proposed to make TCP more robust to out-of-order segment arrival due to multipath routing [19]. The study found that the proposed modified TCP obtains 58% of the throughput as compared to flow hashing vs. 28% of the throughput for round robin and unmodified TCP. In contrast, OMS obtains 100% of the throughput of flow hashing without modification to TCP.

III. OMS SCHEDULER DESIGN

A. System Model and Problem Formulation

We consider a network supporting multipath routing as depicted in Figure 1(a), in which a routing protocol such as [1] associates a weight ϕ_i with path i such that $\sum \phi_i = 1$. The weights are computed by the routing protocol to reflect load balancing or policy objectives and are interpreted as the fraction of bytes that should be transmitted on the respective paths. The objective of OMS is to minimize the mean delay of the multipath packets by exploiting short time scale path conditions, while ensuring that the routing objectives (weights) are satisfied over long time scales.

The path conditions are time-varying due to the bursty nature of cross traffic as well as the multipath traffic itself. Therefore, we use a stochastic model to characterize the delays incurred on different paths and denote $\{D_i^k\}$ as a stochastic process associated with path i , where D_i^k is the delay that packet k would encounter if it is scheduled to path i .

The scheduling algorithm is deployed at splitters, or routers where traffic is split over multiple paths, such as routers A and B in Figure 1(a). To obtain the one-way path delays, we use active probing techniques in which a source splitter sends packets

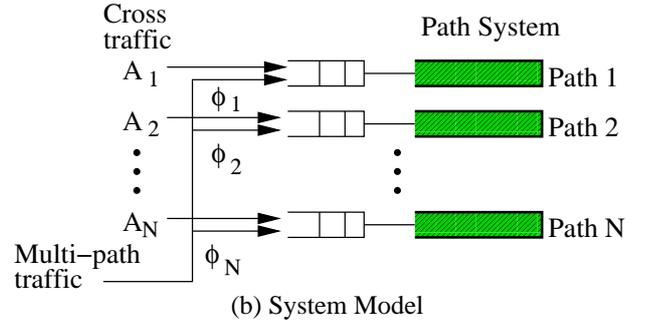
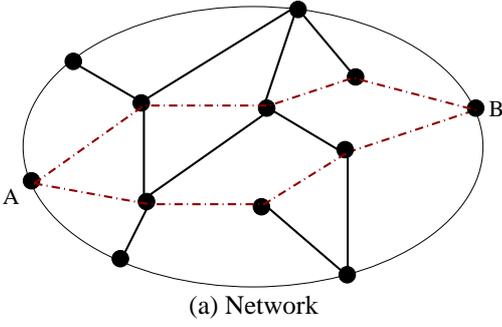


Fig. 1. Network and System Model for Multipath Routing

to its destination splitter located at the end of the multipath, and each probing packet is time-stamped at both the source and destination splitters. Ideally, splitters will have synchronized clocks using NTP [20]. However, OMS does not require that clocks be synchronized as probing packets share the same ingress and egress node. Namely, while unsynchronized clocks will bias the measured queue lengths, it will bias all paths equally. We make one further comment on the effect of unsynchronized clocks at the end of this section after presenting OMS.

To develop OMS, we consider a simplified system model as illustrated in Figure 1(b). As above, we consider N paths with weight ϕ_i for path i , where $\sum_i \phi_i = 1$. Moreover, let path i have arrivals A_i^k at time k . Let $s_k \in \{1, 2, \dots, N\}$ be the path that packet k is scheduled on. Our objective is given by the following.

$$\begin{aligned} & \min E\{D_{s_k}^k\} \\ \text{s.t. } & E\left\{\frac{\sum_k X^k I(s_k, i)}{\sum_k X^k}\right\} = \phi_i \end{aligned}$$

where X^k is the length of packet k and $I(s_k, i)$ is an indicator function that is one if packet k is scheduled to path i and zero otherwise.

Finally, we model path i as a single bottleneck queue with capacity C_i and denote $\{Q_i^k\}$ as a stochastic process associated with path i , where Q_i^k is the queue size that packet k would encounter if it is scheduled to path i .

We can then rewrite the above problem as

$$\begin{aligned} & \min E\left\{\frac{Q_{s_k}^k}{C_{s_k}}\right\} \\ \text{s.t. } & E\left\{\frac{\sum_k X^k I(s_k, i)}{\sum_k X^k}\right\} = \phi_i \end{aligned}$$

where Q_i^k has the following recursive structure [21]

$$Q_i^k = (Q_i^{k-1} + A_i^k + X^k I(s_k, i) - C_i)^+$$

and x^+ denotes $\max(x, 0)$.

B. The OMS Policy

Here we describe OMS, an algorithm that schedules packets over multiple paths at a source splitter router. The objective of OMS is to minimize the average delay of the multipath traffic by

exploiting time-varying path conditions, while also satisfying the route-splitting weights $\phi_1, \phi_2, \dots, \phi_N$.

For simplicity, let each path have the same link capacity. In this case, the problem is to minimize the expected queue size that the multipath packets encounter, i.e.,

$$\begin{aligned} & \min E\{Q_{s_k}^k\} \\ \text{s.t. } & E\left\{\frac{\sum_k X^k I(s_k, i)}{\sum_k X^k}\right\} = \phi_i. \end{aligned}$$

In other words, the scheduler determines which path should be selected to transmit packet k such that we minimize average delay and satisfy the routing objective. In developing OMS in this section, we assume we have perfect information of path conditions, i.e., for a given packet k , the scheduler knows the queue size vector $\vec{Q}^k = (Q_1^k, \dots, Q_N^k)$ where Q_i^k is the queue size of path i at the scheduling time of packet k . Later, we address the case of imperfect information due to delays. The delay-minimizing scheduler chooses the path having minimum queue size, yet may not satisfy the routing objective. Thus we consider only policies that also satisfy the routing objective.

The queue process $\{Q_i^k\}$'s can be stationary or non-stationary and we begin with the stationary case. Let A_i^k and X_i^k be the stationary stochastic process of cross traffic i over path i and multipath traffic respectively. From [22], we have that for a stationary input process, the queue process is also stationary. Let Q_i be a random variable representing the queue size of path i at a generic scheduling time. Assuming equal packet size for simplicity of presentation and replacing expectation with probability, we have

$$\begin{aligned} & \min E\{Q_{s_k}^k\} \\ \text{s.t. } & P\{s_k = i\} = \phi_i. \end{aligned}$$

Let us define a policy S to be a scheduling policy that satisfies $P\{S(\vec{Q}) = i\} = \phi_i$ where \vec{Q} is the queue size vector at a generic packet scheduling time. If $S(\vec{Q}) = i$, then the packet is routed over path i , and it encounters queue size $Q_{S(\vec{Q})}$ (i.e. Q_i). Therefore, $E\{Q_{S(\vec{Q})}\}$ is the average queue size the multipath packets encounter under policy S . Let Ω be the set of all scheduling policies that satisfy the routing constraints as we are interested only in scheduling policies that split traffic in accordance with the routing weights. For example, a greedy policy which always schedules traffic on the lowest delay path and ignores path weights is not a valid policy as it violates the traffic

splitting rules. Thus, our goal is to find a policy S that minimizes the average queue size, namely

$$\min_{S \in \Omega} E\{Q_{S(\vec{Q})}\}. \quad (1)$$

In general, the scheduling policy itself affects the distribution of the queue size since queues are filled by cross-traffic as well as multipath traffic. Yet, to simplify the problem and to have a tractable analysis, we assume that in any time slot (previously, index k represents packet id, henceforth, it represents time-slot under our assumption of fixed-sized packets), the amount of multipath traffic is sufficiently small such that it does not affect the queue size distribution. With this assumption, we have a unique distribution for each path which depends only on the cross-traffic. We explore the impact of the multipath traffic itself using simulations and present the results in Section V.

In [6], Liu et al. investigate a scheduling problem in wireless networks in which a single wireless channel is shared by N users, and selection of user i will result in utility (e.g., throughput) U_i^k according to the user's wireless channel condition. The problem is to determine which user should be scheduled to maximize throughput given constraints of total system resources as well as constraints on temporal fairness, namely, over long time scales, user i should access the channel a fraction ϕ_i of time. From [6], the optimal (utility maximizing) scheduling policy under the above constraints and stationary U_i^k is shown to be

$$S^*(\vec{U}) = \arg \max_i U_i + v_i^* \quad (2)$$

with

$$v_i^* \text{ s.t. } P(\max_{j \neq i} U_j + v_j^* < U_i + v_i^*) = \phi_i. \quad (3)$$

In other words, at each scheduling instant k , the user i that maximizes $\{U_i^k + v_i^*\}$ is selected and scheduled. If the channel conditions are stationary, v_i^* are fixed real values and can be computed using the distributions of U_i and Equations (2) and (3). If U_i^k is unknown or non-stationary, [6] shows how stochastic approximation techniques [23] can be used to adaptively estimate v_i^* based on measurements.

Here, we have one data packet (vs. N packets from N users) to be transmitted on 1 of N paths (vs. a single channel), where the paths have constant capacity and variable congestion (vs. variable capacity due to wireless conditions). Moreover, our objective is to select the minimum delay path (vs. maximum utility) subject to route-splitting byte-count rules (vs. fairness). In any case, we observe that the two problems have a dual mathematical representation such that the optimal (delay minimizing) scheduling policy subject to the multipath splitting ratios is given by

$$S^*(\vec{Q}) = \arg \min_i Q_i + v_i^* \quad (4)$$

with

$$v_i^* \text{ s.t. } P(\min_{j \neq i} Q_j + v_j^* > Q_i + v_i^*) = \phi_i$$

In other words, for the case of stationary cross traffic and a negligible impact of multipath traffic on the queue size distribution,

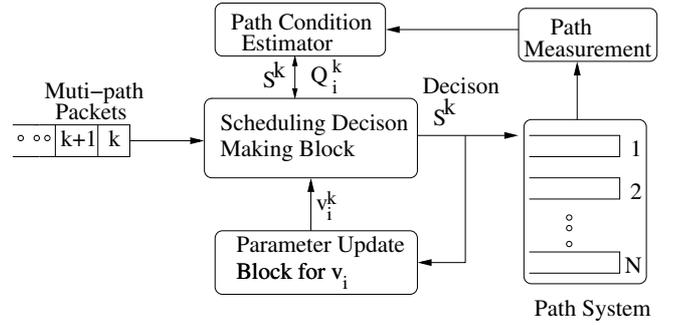


Fig. 2. OMS Block Diagram

the policy S^* minimizes the average queue size encountered by multipath packets while satisfying the routing constraints.

The OMS scheduling policy itself is illustrated in Figure 2. The path measurement module estimates the relative delays of the N paths using active probing as discussed above, such that the estimated delay or queue length of path i for packet k is Q_i^k . To schedule packet k , the scheduling block simply selects the path i minimizing $Q_i^k + v_i^k$. If the distributions of the path characteristics Q_i are known in advance, then v_i^k is a constant and can be computed as in Section IV. If the distributions are unknown or the path characteristics are non-stationary, then v_i^k is updated in the parameter update block according to

$$v_i^{k+1} = v_i^k - \frac{1}{k}(\phi_i - I(S^k, i)). \quad (5)$$

As shown in [23], [6], v_i^k is guaranteed to converge to its optimal value for a quite general class of scenarios including noisy observations. Intuitively, if $v_i = 0$, the path with the lowest delay is always selected. However, adapting v_i^k according to Equation (5), ensures that the weights that define the traffic splitting ratios ϕ_i are obeyed.

Finally, we return to the issue of unsynchronized clocks. The adaptive search for v_i^* begins with an initialization of $v_i^k = 0$ for all i . Observe that if the egress node's clock lags or leads the ingress node's by ϵ , the optimal values will simply shift from v_i^* to $v_i^* + C\epsilon$ or $v_i^* - C\epsilon$ respectively. Consequently, OMS will take correspondingly additional or fewer steps to converge to the new optimal value from its starting point of $v_i^k = 0$.

IV. ANALYTICAL PERFORMANCE MODEL

In this section, we study the performance of OMS by computing the mean delay of each of the multiple paths under the OMS algorithm. As a baseline for comparison, we also compute the mean delay of a Round-Robin (RR) policy which simply alternates paths according to the weights ϕ_i without regard to path conditions.

To maintain tractability while still capturing the essential aspects of the system, we model each path by a single bottleneck queue as in Figure 1, model traffic by a (self-similar) fractional Brownian motion (fBm) process, and assume that the amount of multi-path traffic itself does not change the queue size distribution.

In this case, the mean delay of a path is given by EQ/C where C is the link capacity and henceforth we use queue

length and delay interchangeably. We define the gain of OMS over Round Robin as

$$G = \frac{E\{Q_{RR}\} - E\{Q_{OMS}\}}{E\{Q_{RR}\}} \quad (6)$$

such that $0 < G \leq 1$ indicates a delay reduction of OMS vs. Round Robin, with 1 representing the best achievable. A gain $G < 0$ would indicate that Round Robin has lower delay. Our results provide the performance gain of OMS as compared to round-robin multi-path scheduling as a function of cross-traffic parameters (mean rate, variance coefficient and Hurst parameter). In special cases, we compute closed-form expressions.

A. General Formulation

Let path i have fBm traffic with mean rate m_i , variance coefficient a_i and Hurst parameters H_i . In [22], Norros showed that the queue occupancy distribution is asymptotically Weibull,² i.e.,

$$P\{Q_i > x\} \approx e^{-\beta_i x^{\gamma_i}} \quad (7)$$

where

$$\gamma_i = 2(1 - H_i) \\ \beta_i = \frac{1}{2a_i m_i (1 - H_i)^2} \left(\frac{(1 - m_i)(1 - H_i)}{H_i} \right)^{2H_i}$$

for links with unit capacity. Let us assume we have N paths with capacity 1 and the queue size distribution is Weibull. The OMS algorithm is

$$S^k(\vec{Q}^k) = \arg \min_i Q_i^k + v_i^*.$$

Observe that we can calculate \vec{v}^* by using the joint density function of the queue size distribution of each of the paths. In particular, as Q_i denotes a random variable representing the stationary queue size of path i , from the definition of OMS, we have that

$$Q_{OMS} = Q_S$$

where

$$S = \arg \min(Q_1 + v_1^*, \dots, Q_N + v_N^*).$$

Moreover, since the Q_i 's can be assumed independent since they are serving separate cross traffic flows, $E\{Q_{OMS}\}$ is given by

$$\begin{aligned} E\{Q_{OMS}\} &= \int_0^\infty \dots \int_0^\infty q_S \Pi_{i=1}^N f_{Q_i}(q_i) dq_i \\ &= \int_0^\infty \dots \int_0^\infty q_S \Pi_{i=1}^N \gamma_i \beta_i q_i^{\gamma_i - 1} e^{-\beta_i q_i^{\gamma_i}} dq_i \\ &= \int_0^\infty \dots \int_0^\infty q_S e^{-\sum_{i=1}^N \beta_i q_i^{\gamma_i}} \Pi_{i=1}^N \gamma_i \beta_i q_i^{\gamma_i - 1} dq_i \end{aligned} \quad (8)$$

where $S = \arg \min(q_1 + v_1^*, \dots, q_N + v_N^*)$.

Since the Weibull distribution does not have a moment generating function, a general closed form expression for Equation

²Measurement studies have found the Weibull distribution effective in characterizing queueing delay [24].

(8) cannot be obtained and we resort to numerical integration as described below. However, a closed form expression for $E\{Q_{OMS}\}$ can be obtained for the case of independent but statistically homogeneous paths that have identical mean, variance, and Hurst parameters, as also described below.

For the case of round-robin multipath scheduling, the average delay or queue size encountered by multipath traffic is computed as follows. Let $Q_{RR} = \phi_1 Q_1 + \dots + \phi_N Q_N$. The expected queue size is given by

$$\begin{aligned} E\{Q_{RR}\} &= E\{\phi_1 Q_1 + \dots + \phi_N Q_N\} \\ &= \phi_1 E\{Q_1\} + \dots + \phi_N E\{Q_N\} \end{aligned}$$

Since the mean of the Weibull distribution is given by

$$E\{Q_i\} = \frac{\Gamma(1 + \frac{1}{\gamma_i})}{\beta_i^{1/\gamma_i}}$$

where

$$\Gamma(z) = \int_0^\infty e^{-x} x^{z-1} dx$$

we have

$$E\{Q_{RR}\} = \sum \frac{\phi_i \Gamma(1 + \frac{1}{\gamma_i})}{\beta_i^{1/\gamma_i}}.$$

B. Homogeneous Paths

Here we consider the case of N homogeneous or *iid* paths, i.e., each path has statistically independent fBm traffic, and all paths have the same mean, variance, and Hurst parameter and weight $\phi_i = 1/N$.

In this case, $\vec{v}^* = \vec{0}$ and $Q_{OMS} = \min(Q_1, \dots, Q_N)$. Therefore, the expected queue size under OMS is given by

$$\begin{aligned} E\{Q_{OMS}\} &= \int_0^\infty \Pi_{i=1}^N (1 - F_{Q_i}(x)) dx \\ &= \int_0^\infty \Pi_{i=1}^N e^{-\beta x^\gamma} dx \\ &= \int_0^\infty e^{-N\beta x^\gamma} dx \\ &= \frac{\Gamma(\frac{1}{\gamma})}{\gamma \beta^{1/\gamma} N^{1/\gamma}} \end{aligned}$$

Thus, we can compute the exact gain of OMS vs. RR as

$$\begin{aligned} G &= 1 - \frac{\Gamma(\frac{1}{\gamma})}{\gamma \beta^{1/\gamma} N^{1/\gamma}} \cdot \frac{\beta^{1/\gamma}}{\Gamma(1 + \frac{1}{\gamma})} \\ &= 1 - \frac{\Gamma(\frac{1}{\gamma})}{\gamma N^{1/\gamma} \Gamma(1 + \frac{1}{\gamma})} \\ &= 1 - \frac{1}{N^{1/\gamma}} \end{aligned} \quad (9)$$

where $\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1)$.

Observe from Equation (9) and Figure 3 that the gain of OMS vs. RR depends only on the number of paths and the Hurst parameter under statistically homogeneous path conditions. In

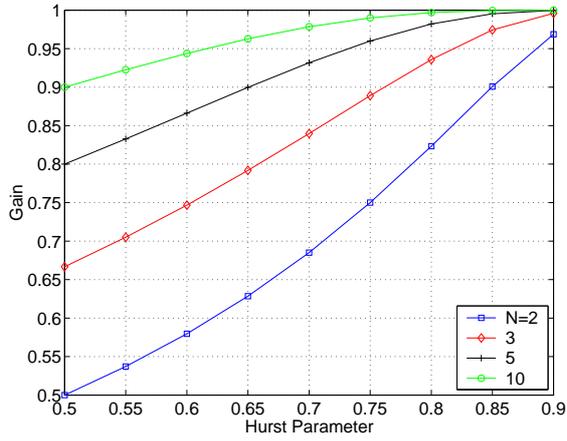


Fig. 3. OMS Gain for Homogeneous Paths

particular, observe that gain increases with increasing Hurst parameter and with an increasing number of paths. When the Hurst parameter increases, the long-time-scale autocorrelation of the traffic increases, thereby increasing both the mean and variance of the queue length. With more highly variable path conditions, OMS has a greater opportunity to opportunistically reduce delay. Similarly, with a larger number of paths, it becomes increasingly likely that one of the paths is in a low-delay state, perhaps even with an empty queue, as is occurring as the gain approaches 1.

C. Two Paths with $\phi_1 = \phi_2$

An important special case for simulation experiments and model validation is two paths having identical weight. This scenario occurs if the cross traffic of the two paths has the same mean rate and in Equal Cost Multi-Path. In this case, we have $\phi_1 = \phi_2 = 0.5$ and can calculate v_i^* by using the joint density of the queue sizes. That is,

$$\begin{aligned}
0.5 &= P(Q_1 > Q_2 + v^*) \\
&= \int_0^\infty \int_{q_2+v^*}^\infty f(q_1, q_2) dq_1 dq_2 \\
&= \int_0^\infty \int_{q_2+v^*}^\infty \gamma_1 \beta_1 q_1^{\gamma_1-1} e^{-\beta_1 q_1^{\gamma_1}} \gamma_2 \beta_2 q_2^{\gamma_2-1} e^{-\beta_2 q_2^{\gamma_2}} dq_1 dq_2 \\
&= \int_0^\infty \gamma_2 \beta_2 q_2^{\gamma_2-1} e^{-\beta_2 q_2^{\gamma_2}} e^{-\beta_1 (q_2+v^*)^{\gamma_1}} dq_2 \quad (10)
\end{aligned}$$

for a normalized link capacity of 1. We then use numerical techniques to solve Equation (10) since a closed form solution does not exist for $\gamma \leq 1$. Once v^* is numerically obtained, we compute the mean queue size $E\{Q_{\text{OMS}}\}$ numerically using Equation (8) which simplifies to

$$\begin{aligned}
E\{Q_{\text{OMS}}\} &= \int_0^\infty \int_0^{q_2+v^*} q_1^{\gamma_1} q_2^{\gamma_2-1} \beta_1 \beta_2 e^{-\{\beta_1 q_1^{\gamma_1} + \beta_2 q_2^{\gamma_2}\}} dq_1 dq_2 \\
&\quad + \int_0^\infty \int_{q_2+v^*}^\infty q_2^{\gamma_2} q_1^{\gamma_1-1} \beta_1 \beta_2 e^{-\{\beta_1 q_1^{\gamma_1} + \beta_2 q_2^{\gamma_2}\}} dq_1 dq_2.
\end{aligned}$$

V. NUMERICAL AND SIMULATION RESULTS

In this section, we perform numerical investigations of the multipath model for OMS and RR presented in Section IV.

Moreover, we perform an extensive set of ns-2 simulations to explore the accuracy of the analytical model and study performance factors not incorporated by the model such as effects of delayed information and the impact of multipath routing and scheduling on TCP throughput.

We consider a scenario as depicted in Figure 4(a) in which the two paths between I_2 and E_2 have the same capacity, same propagation delay (1 msec) and identical weight, i.e., the multipath traffic is split equally among the two paths. We study the following performance factors and ranges for both the analytical model and simulations: Hurst parameter [0.5,0.9], mean rate [0.3,0.9], and variance coefficients [0.5, 4]. Moreover, with simulations we study the ratio of multipath to cross-traffic [0,0.5] and the feedback delay for path quality estimation [0,1 sec]. We consider two traffic models: self-similar traffic generated from a multi-fractal wavelet model [25] as well as ‘‘closed-loop’’ TCP flows. For TCP, we study the level of aggregation [1,60 flows] and the version of TCP [New-Reno, SACK], which impacts TCP’s ability to recover from out-of-order packets due to multipath routing.

A. Homogeneous Paths

Our first experiments investigate the case where each path has the same mean rate, variance coefficient and Hurst parameter. Figure 5(a) depicts the gain predicted by the analytical model along with that obtained via simulations. In this case, the gain is given by the expression in Equation (9). The general trend of all five curves is that the gain of Opportunistic Multipath Scheduling vs. Round Robin increases with increasing Hurst parameter for the reasons described in Section IV-B. Moreover, the upper four simulation curves of Figure 5(a) indicate that the gain decreases with increasing variance coefficient a as well as with increased mean m . The reason is that while increased variance of traffic conditions also provides OMS with an increased opportunity to exploit low-delay paths, increased variance also increases mean queue lengths (even in unipath routing). This effect is explored in Figure 5(b), in which we vary the variance coefficients from 0.5 to 4 for a mean rate of 0.7 and Hurst parameter of 0.5. The figure shows the effect on both gain (the decreasing curve) as well as the mean queue length (the two increasing curves). Observe that while the difference in mean delay between RR and OMS is increasing with increasing variance coefficient, the gain is decreasing, as gain is normalized to the RR delay as in Equation (6). Moreover, observe that for larger mean rates and Hurst parameters, increasing variance has a lesser effect on gain.

Finally, returning to Figure 5(a), observe that the analytical result suggests that gain only depends on the Hurst parameter, whereas the simulation results show that it also depends on mean rate and variance coefficient. Consequently, in practice, the analytical result becomes a lower bound on the gain of OMS vs. RR. The reason is that the queue occupancy distribution of Equation (7) is an asymptotic lower bound (asymptotic in buffer size) that becomes increasingly tight for large queue sizes.

B. Impact of Variance Coefficient Ratio

Here we consider paths with heterogeneous statistical properties and study the effect of the variance coefficient ratio. In

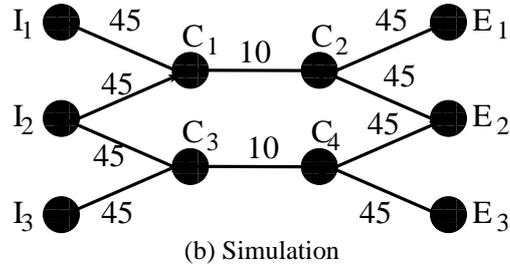
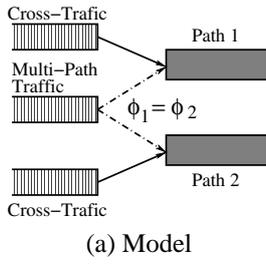


Fig. 4. Scenario

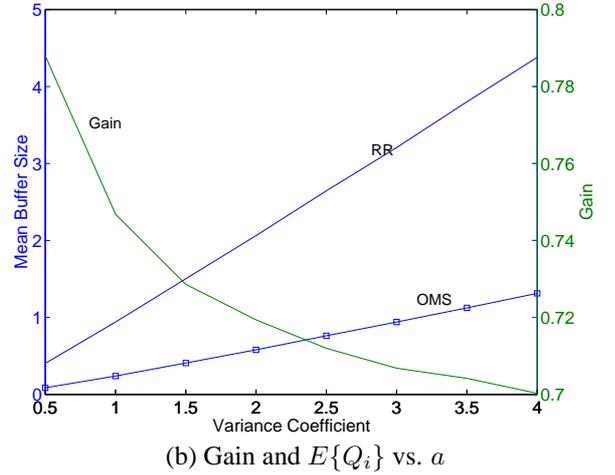
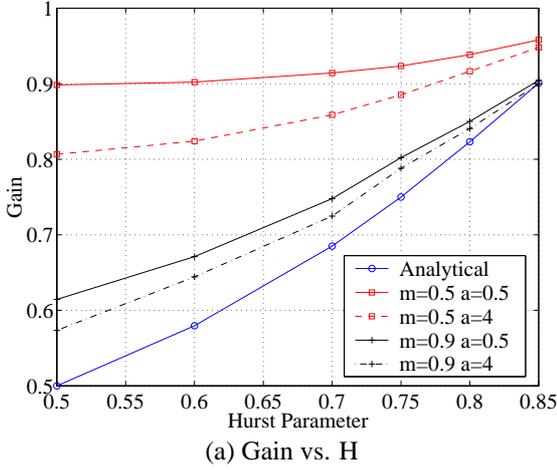


Fig. 5. Analytical and Simulation Results for Homogeneous Paths

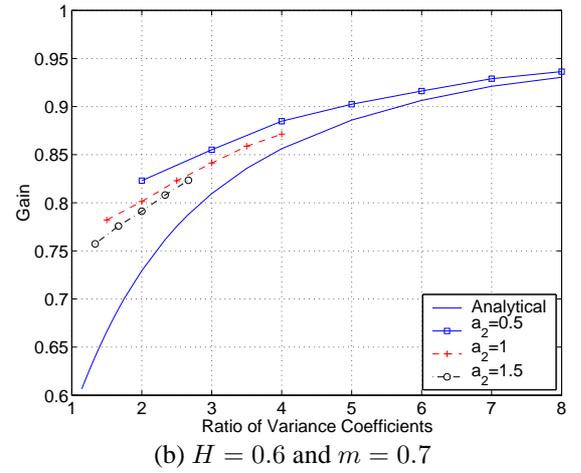
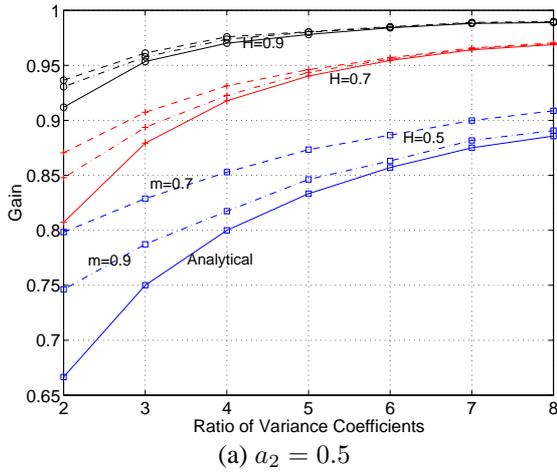


Fig. 6. Effect of a_1/a_2 on Gain

particular, we consider paths 1 and 2 having respective variance coefficients a_1 and a_2 and both paths having the same mean rate m and Hurst parameter H . Figure 6 illustrates the effect of the ratio of variance coefficients and compares analytical and simulation results. Figure 6(a) depicts three groups of curves corresponding to $H = 0.9$ (top three curves), $H = 0.7$ (middle three), and $H = 0.5$ (lower three). The variance coefficient for path 2 is $a_2 = 0.5$ and a_1 is varied such that the x-axis depicts a_1/a_2 . First, the general trend is that gain increases with an increasing variance coefficient ratio. Namely, OMS exploits increasing path heterogeneity to reduce queuing delay to the maximal extent possible subject to the routing

constraints. Second, note that the analytical bound is increasingly tight for higher mean rates and Hurst parameter for the reasons described above.

Figure 6(b) shows the result for a fixed Hurst parameter of 0.6 and a fixed mean rate of 0.7. The upper three curves depict cases of $a_2 = 0.5, 1, 1.5$, and a_1 is again varied with the x-axis depicting a_1/a_2 . Observe that the effects of path heterogeneity captured through the ratio a_1/a_2 dominate the effect of the particular value of a_2 . The analytical result again behaves as a lower bound that is tighter when the Hurst parameter or ratio increases.

C. Multipath vs. Cross Traffic Ratio

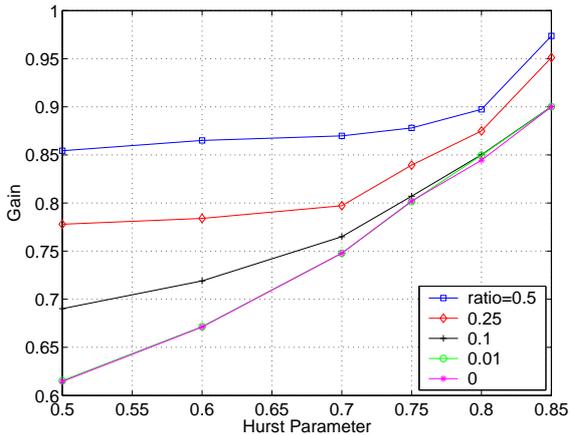


Fig. 7. Effect of Multipath vs. Cross Traffic Ratio

To maintain tractability, the analytical model assumes that queue performance is dominated by cross traffic and that the multipath traffic itself has negligible impact. In this section, we use simulation to study the effects of the multipath traffic itself. In particular, we vary the amount of multipath traffic from 1% to 50% of the cross traffic while holding the mean rate of the paths constant.

Figure 7 depicts gain as a function of Hurst parameter for ratios of multipath to cross traffic of 0 to 0.5, for a fixed mean rate of $m = 0.9$ and variance coefficient of $a = 0.5$. The figure shows that gain increases with an increasing ratio of multipath traffic since the multipath traffic has higher impact on the paths' conditions, and OMS is incorporating these conditions into the scheduling decision.

However, the gain increment is smaller for larger Hurst parameters, as in such cases, OMS already extracts a significant gain from long-time-scale correlation of cross traffic, and limited further gain is available (cf. Figure 5). This behavior is also observed in other experiments (not shown) for different mean rates and variance coefficients including heterogeneous cases.

D. Information Delay

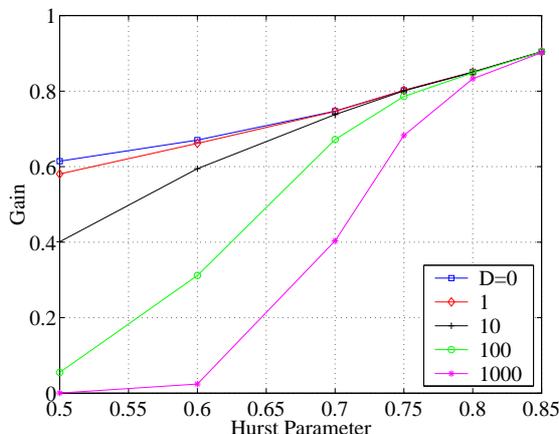


Fig. 8. Effect of Information Delay

In the analytical model, we assume that the conditions of network paths are immediately available at the traffic splitter. However, in practice, information will be available only after a round trip time that includes propagation and queuing delays. For larger networks, this round-trip delay is expected to be on the order of tens of milliseconds [24].

To explore this issue, we perform a number of experiments in which the feedback delay of the queue state varies from 0 to 1 second for paths with mean rate 0.9 and variance coefficient 0.5. As illustrated in Figure 8, the gain decreases with increasing feedback delay, as expected. However, the degradation for large Hurst parameters is almost negligible since the temporal correlation of the buffer occupancy increases with the Hurst parameter such that the buffer occupancy does not change dramatically over moderate timescales. As measurement studies indicate that Hurst parameters incurred in practice are typically within a range of 0.7 to 0.85, delayed information will not be a major impediment in achieving high performance with OMS in many cases. We also find that the relative degradation decreases when the variance coefficient and mean rate increases for reasons discussed previously, and that the results are similar for heterogeneous variance coefficients and Hurst parameters.

In any case, we note that additional techniques can be incorporated into OMS to mitigate the effects of information delay. First, estimation techniques can be used to predict current path conditions based on the correlation structure of measurements (see [26] for example). Moreover, the information delay could be reduced significantly if the core routers directly exchange queue-state information with edge routers vs. use of edge-to-edge probing.

We next study the combined effects of the ratio of multipath to cross traffic and information delay and present the results in Figure 9. The key observation is that if the multipath traffic ratio and delay are sufficiently large, and H is sufficiently small, then RR can outperform OMS as indicated by a negative gain. Intuitively, if the multipath ratio is 1, then the queue lengths are determined solely by the multipath traffic. If the information delay is simultaneously sufficiently large and the queue process is not sufficiently correlated (H near 0.5), then OMS increasingly selects the “wrong” path by basing its current scheduling decision on its now-irrelevant past scheduling decisions. In any case, we observe that OMS still obtains significant gains as compared to RR for parameter values observed in practice consisting of larger values of H and moderate round trip times.

E. TCP Traffic

Thus far, we have considered “open loop” traffic in which the flow input rates are not affected by network performance. Here, we consider TCP traffic in which flows adapt to network conditions according to the TCP congestion avoidance policy.

We use ns-2 simulations and consider the scenario of Figure 4(b) with links of 10 and 45 Mb/sec. Multipath flows are established between I_2 and E_2 and weights of the paths are equal. In addition to multipath traffic, we have cross-traffic flows between $I_1 - E_1$, $I_1 - E_2$, and $I_2 - E_1$ over link $C_1 - C_2$, and $I_3 - E_3$, $I_3 - E_2$, and $I_2 - E_3$ over link $C_3 - C_4$. To obtain the path condition, we use active probing and insert a time-stamp

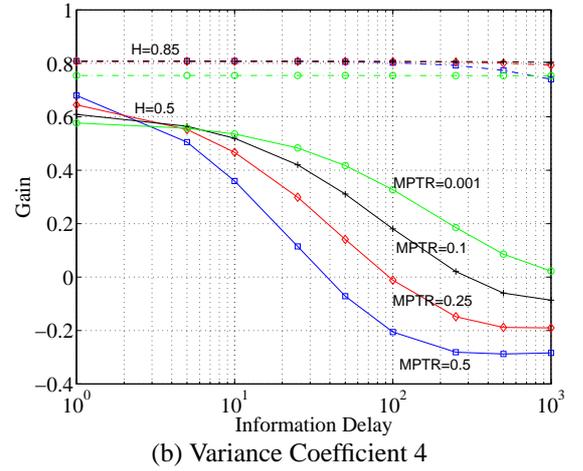
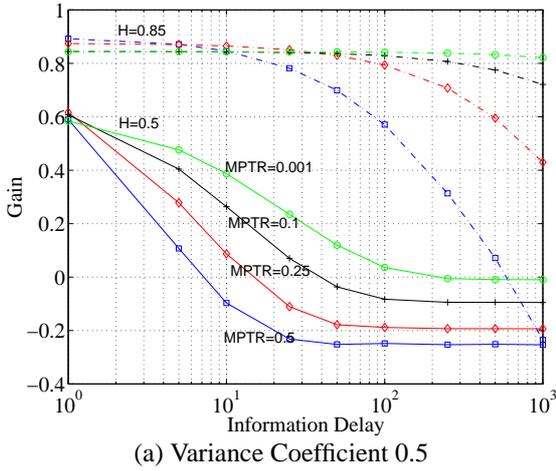


Fig. 9. Joint Effect of Information Delay and Cross Traffic Ratio

into the headers of probe packets at I_2 , and use this information to calculate the path delay at E_2 . Node E_2 then sends this measured delay back to I_2 for implementing the OMS policy.

In all experiments, we report TCP goodput normalized to the ideal fair-share rate. Namely as the system is “balanced,” all TCP flows (multipath and unipath) should obtain an identical goodput and the normalized goodput should be 1, with values above or below 1 indicating that a flow is getting more or less than its fair share. Moreover, we consider both TCP NewReno [8] as well TCP SACK [9], with the latter mitigating the effects of mis-ordered packets due to multipath routing.

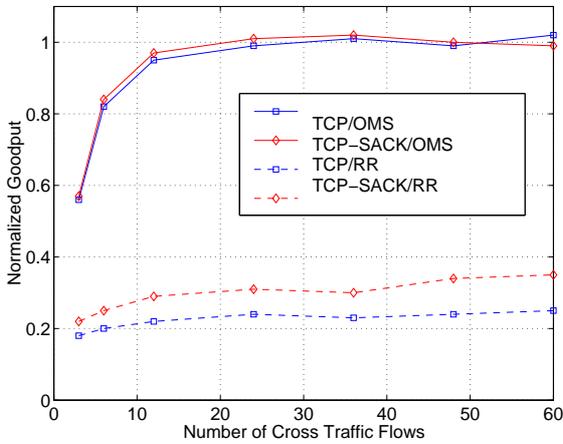


Fig. 10. TCP Aggregation and SACK

We begin with a single multipath TCP flow and consider scenarios of TCP NewReno (simply referred to as TCP) and TCP SACK over both OMS and RR multipath scheduling. We repeat the experiment for different numbers of cross-traffic flows ranging from 3 to 60 per path as reported on the x-axis. For each experiment, the probing period is 10 msec and the buffer size is 166 kB. The probing overhead per path is either 32 kb/sec regardless of the number of multi-path flows, as the probing interval is 10 msec and only a packet header needs to be transmitted (40 bytes under IPv4).

Figure 10 depicts the resulting goodput of the multipath TCP flow normalized to its fair share. First, observe that with RR

scheduling, the multipath TCP flow obtains a goodput that is 20% to 35% of its fair share. Thus, despite having two paths to choose from, TCP over multipath routing with round robin scheduling significantly decreases TCP throughput due to packet misordering. However, with Opportunistic Multipath Scheduling, the multipath TCP flow encounters significantly-reduced delays and reduced delay variability. As delay reduction increases TCP throughput proportionately [27] and delay-variability reduction reduces the number of out-of-order packets, the multipath TCP flow can achieve its ideal goodput of 1 across a broad range of scenarios.

Second, observe that while OMS always outperforms RR, an aggregation level of at least 10 cross traffic flows is required for the multipath flow to achieve its ideal fair goodput. The reason is the ratio of multipath to cross traffic explored previously: OMS has increasingly high performance when cross traffic dominates multipath traffic.

Third, note the difference between TCP NewReno and TCP SACK. With RR, selective acknowledgement improves the throughput of the multipath TCP flow by as much as 40% (.25 to .35). In contrast, with OMS, the difference in performance of TCP variants is overwhelmed by the effects of selecting better transmission paths.

Our final experiments explore the effects of probing interval and buffer size on TCP throughput. In both cases, three curves are depicted for 3, 12, and 48 cross-traffic flows with all experiments having one multipath flow. Figure 11(a) illustrates the effect of varying the probing interval from 1 msec to 1 sec. With very rapid probing and 12 and 48 cross-traffic flows, OMS selects paths so efficiently that the multipath TCP flow obtains higher throughput than the cross-traffic TCP flows as indicated by goodputs greater than 1. Otherwise, provided that probing occurs at least once per 10 msec and the ratio of multipath to cross-traffic is sufficiently small, OMS can attain goodputs near 1.

Figure 11(b) illustrates the impact of buffer size on the normalized goodput of the multipath TCP flow. As OMS is using measured delays to schedule packets, it requires sufficiently large buffers to differentiate the paths. Thus, to obtain normalized goodputs near 1, a buffer size of at least 166 kB is required,

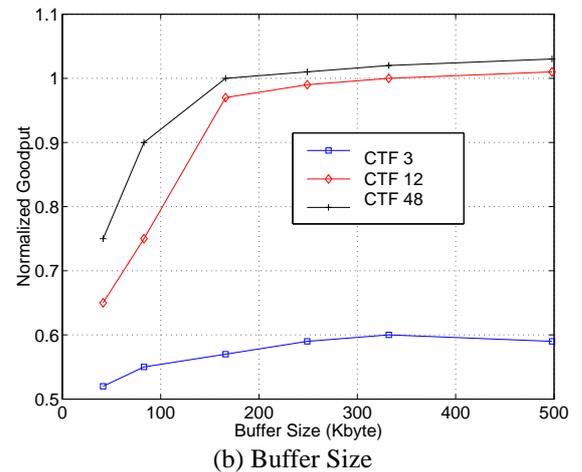
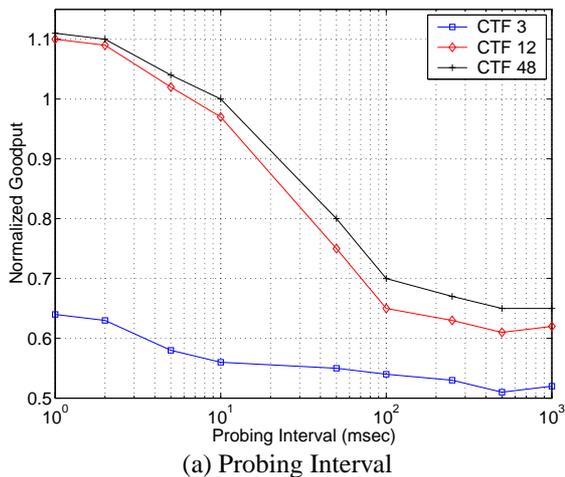


Fig. 11. Effect of Probing Interval and Buffer Size

a modest amount for this delay-bandwidth product.

In further experiments, we vary the number of multipath TCP flows as well as the number of cross traffic flows. As the conclusions above and from Figure 7 generalize to these cases, we present no further discussion here.

VI. CONCLUSIONS

Multipath routing provides a mechanism for load balancing, traffic engineering, and ensuring efficient network operation. Unfortunately, splitting traffic according to a round-robin policy can reduce throughput of TCP flows by as much as 80%. In this paper, we introduced Opportunistic Multipath Scheduling (OMS), a traffic splitting algorithm that opportunistically exploits high quality paths when they occur while simultaneously ensuring that paths are utilized according to the specified routing weights. We analytically showed that OMS has increasing efficiency with long-time-scale correlated traffic. Moreover, using simulations we showed that OMS is robust to information delay and dramatically improves throughput of multipath TCP flows.

REFERENCES

- [1] J. Moy, "OSPF version 2," 1998, Internet RFC 2328.
- [2] Z. Cao, Z. Wang, and E. Zegura, "Performance of hashing-based schemes for Internet load balancing," in *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000.
- [3] C. Villamizar, "OSPF Optimized Multipath (OSPF-OMP)," Feb. 1999, Internet Draft (draft-ietf-ospf-omp-02).
- [4] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," in *Proceedings of IEEE INFOCOM 2003*, San Francisco, CA, Apr. 2003.
- [5] S. Borst and P. Whiting, "Dynamic rate control algorithms for HDR throughput optimization," in *Proceedings of IEEE INFOCOM '01*, Anchorage, Alaska, Apr. 2001.
- [6] X. Liu, E. Chong, and N. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2053–2065, 2002.
- [7] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *Proceedings of ACM MOBICOM '02*, Sept. 2002.
- [8] S. Floyd and T. Henderson, "The newreno modification to tcp's fast recovery algorithm," 1999, Internet RFC 2582.
- [9] S. Floyd et al., "An Extension to the Selective Acknowledgement (SACK) Option for TCP," 2000, Internet RFC 2883.
- [10] H. Kaur, S. Kalyanaraman, A. Weiss, S. Kanwar, and A. Gandhi, "BANANAS: An evolutionary framework for explicit and multipath routing in the Internet," in *Proceedings of FDNA 2003*, Karlsruhe, Germany, Aug. 2003.
- [11] S. Vutukury and J.J. Garcia-Luna-Aceves, "A simple approximation to minimum-delay routing," in *Proceedings of ACM SIGCOMM '99*, Cambridge, MA, Aug. 1999.
- [12] W. Zaumen and J.J. Garcia-Luna-Aceves, "Loop-free multipath routing using generalized diffusing computations," in *Proceedings of IEEE INFOCOM '98*, San Francisco, CA, Mar. 1998.
- [13] S. Bahk and M. El Zarki, "Dynamic multi-path routing and how it compares with other dynamic routing algorithms for high speed wide area networks," in *Proceedings of ACM SIGCOMM '92*, Aug. 1992.
- [14] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 885–896, Dec. 1999.
- [15] S. Lee and M. Gerla, "Fault tolerance and load balancing in QoS provisioning with multiple MPLS paths," in *Proceedings of IWQoS 2001*, Karlsruhe, Germany, June 2001.
- [16] S. Nelakuditi and Z. Zhang, "On selection of paths for multipath routing," in *Proceedings of IWQoS 2001*, Karlsruhe, Germany, June 2001.
- [17] N. Rao and S. Batsell, "QoS routing via multiple paths using bandwidth reservation," in *Proceedings of IEEE INFOCOM '98*, San Francisco, CA, Mar. 1998.
- [18] J. Chen, P. Druschel, and D. Subramanian, "An efficient multipath forwarding method," in *Proceedings of IEEE INFOCOM '98*, San Francisco, CA, Mar. 1998.
- [19] Y. Lee, I. Park, and Y. Choi, "Improving TCP performance in multipath packet forwarding networks," *Journal of Communications and Networks*, vol. 4, no. 2, pp. 148–157, June 2002.
- [20] D. Mills, "On the accuracy and stability of clocks synchronized by the Network Time Protocol in Internet systems," *Computer Communications Review*, vol. 20, no. 1, Jan. 1990.
- [21] D. Lindley, "On the theory of queues with a single server," *Proceedings of the Cambridge Philosophical Society*, vol. 48, pp. 277–289, 1952.
- [22] I. Norros, "A storage model with self-similar input," *Queueing Systems*, 16:387–396, 1994.
- [23] H. Kushner and D. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Verlag, 1978.
- [24] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, F. Tobagi, and C. Diot, "Analysis of measured single-hop delay from an operational backbone network," in *Proceedings of IEEE INFOCOM 2002*, New York, NY, June 2002.
- [25] V. Ribeiro, R. Riedi, M. Crouse, and R. Baraniuk, "Simulation of non-gaussian long-range-dependent traffic using wavelets," in *Proceedings of ACM SIGMETRICS '99*, Atlanta, GA, June 1999, pp. 1–12.
- [26] Y. Gao, G. He, and C. Hou, "On leveraging traffic predictability in active queue management," in *Proceedings of IEEE INFOCOM 2002*, New York, NY, June 2002.
- [27] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proceedings of ACM SIGCOMM 1998*, Vancouver, British Columbia, Sept. 1998.