Reaching approximate Wardrop equilibrium at reduced costs of link state updates

Yong Oh Lee, B. Rengarajan† and A. L. Narasimha Reddy
Dept. of Electrical and Computer Engineering, Texas A & M University
†IMDEA Networks networks
Email:yongoh.lee@tamu.edu,†balaji.rengarajan@imdea.org, reddy@ece.tamu.edu

***Abstract*—Several routing protocols have been proposed to take advantage of the dynamic metrics on links such as link delays, queueing lengths, and available link bandwidths. Wardrop routing, QOS routing are few such example routing protocols. Even when these protocols can be shown to offer convergence properties without oscillations, the protocols have not been widely adopted for a number of reasons. The expected cost of keeping the link metrics updated at various nodes in the network being one of them.**

In this paper, we study the problem of reducing the cost of propagating dynamic link metrics while keeping the quality of paths within a margin of error. We propose a simple technique of threshold-based metric propagation that is shown, through analysis and simulations, to offer bounded guarantees on path quality while significantly reducing the cost of propagating the dynamic link metric information. Our results indicate that threshold based updates can reduce the number of link updates by up to 90-95% in some cases.

***Index Terms*—Dynamic routing, link state updates, wardrop routing, efficiency, performance**

I. Introduction

Current routing algorithms utilize static link costs to compute routing tables between different nodes in the network. The link costs are static for long periods of time (over the duration of several hours) and are determined by the traffic engineering constraints of the network. The problem of determining the link costs has received significant attention [1]–[3]. Typically, the traffic matrix and several considerations such as keeping maximum link utilization low etc. are factored into obtaining link costs. The problem of determining link costs may be simultaneously coupled with the problem of computing routing paths in some approaches [4], [5], [9]. The current approaches to determine link costs take traffic matrices over several hours into account such that varying traffic matrices may reasonably accommodated with one set of link costs [26], [27].

Dynamic link metrics such as link delay, queueing lengths and available link bandwidth have been considered earlier as potential link cost metrics for routing purposes. For example, routing high bandwidth video flows might benefit from an idea of available link bandwidth in QOS routing [6]. Similarly, path lengths or delays can be useful in Wardrop routing [7]. Early ARPAnet considered link delays as a cost metric and the resulting oscillations prompted the use of other metrics based on capacity. QOS routing has explored the use of different dynamic metrics in routing traffic, for example in [16]–[19]. Dynamic metrics such as available bandwidth and path delay have been proposed for use in routing video and audio traffic in the network. This body of work considered the tradeoff in keeping the link state information disseminated and the quality of paths that can be computed. Some of this work proposed techniques for finding new paths efficiently, for example [19].

Most current networks, however, do not employ dynamic link metrics for various reasons. Since these metrics are dynamic, as the link metrics change over time, the traffic might be routed at different times through different paths in the network, potentially causing oscillations with incorrect choice of link metrics or routing algorithms. Even when the routing algorithms are carefully designed to not cause oscillations, the cost of propagating the link metrics has been one of the obstacles to the adoption of these algorithms.

As the dynamic link metrics change over time, these metrics need to be measured and propagated to other nodes in the network in order to keep the routing paths from deviating far from ideal. The more frequently the link metric information is propagated, the more accurate the information that the nodes have about the state of the network, and the better the efficiency of the computed network paths. However, higher frequency of updates leads to higher cost in propagating the link metric information. This tension or tradeoff has been studied through simulations, for example, in QOS routing [8].

Recently, dynamic routing algorithms, have received renewed interest for balancing load in wireless networks [7], for dynamic traffic management in wired networks [9] and traffic management across multiple paths in a multi-homed network [25]. These algorithms have used link delays or utilizations for dynamic routing.

Link updates can be sent periodically or triggered on link up/down events in OSPF routing. In order to prevent spurious link up/down events from generating excessive link update traffic, timers may be employed. These timers are in the range of several seconds (typically 30s). The importance of conveying reliable link information without generating excessive number of link updates has been earlier recognized [28], [29].

A related question that arises with the quality of link information is whether the routing algorithm can converge to a stable state despite the delay or inaccuracies in the link state information that is used in making the routing decisions.

In this paper, we try to address these problems of reducing

978-1-4244-8953-4/11/$26.00

the cost of propagating dynamic link metric information across the network while ensuring stability of the routing algorithm. We focus our attention on Wardrop routing (as an example dynamic routing approach) that employs link delays as a link cost metric. In Wardrop routing, the traffic is split across available paths in such a way as to equalize the delay across all the available paths at a node. The traffic splitting can be done at the end hosts [22] or further split at the routers in the network as traffic moves from one hop to the next [9]. However, our results can be equally applied to other algorithms, with suitable modifications.

This paper makes the following significant contributions: (1) proposes a simple technique, called threshold-based propagation, for propagating link metric information; (2) presents an analysis that threshold propagation can guarantee that the observed path quality will be within an error bound of the optimal path quality if the exact information is available; and (3) shows, through simulations, that threshold propagation reduces the cost of propagating link cost information significantly, in some cases by up to 90-95%.

II. Threshold based updates

Most earlier approaches assume that nodes measure the state of their links at regular intervals and propagate this information to the other nodes in the network. The nodes compute new routing tables or new traffic distribution ratios when all the information is received. In order to keep the information from getting too stale, the measurements and the propagation of the link state is carried out at regular intervals. The cost of propagating this information and the staleness of link state is controlled by controlling the frequency or rate of measuring and the propagation of the link state. A possible approach to reducing the cost of updates is based on observing the local link state. Every node keeps track of the last link state that is propagated to the rest of the network. When the currently measured link state differs from the last updated state considerably, and the difference exceeds a threshold, only then does the node propagate the link state information to other nodes in the network. We call such a policy *threshold based updates*. The thresholds can be based on allowable absolute error in link state or on the maximum allowable relative error. For example, thresholds can be such as 1ms or 20%. Absoulte error thresholds may not be universally applicable. A 1ms error threshold may be reasonable when link delay is say 10ms, but may not be reasonable when link delays are in the range of 100ms or 1ms. Relative error thresholds can cover wide range of link states. However, relative thresholds can be problematic as the links get heavily loaded. While, in this regime, it may be necessary to propagate information more quickly in order to distribute the load to other parts of the network and higher thresholds would not be beneficial.

In order to accommodate all the conflicting needs, we pursue a policy here that tries to combine both absolute and relative error thresholds. A node propagates its link state if its current link state exceeds the minimum of absolute or relative error thresholds. More formally, if $|l_i - lj| \geq min(e_{abs}, e_{rel} l_j)$, where l_j is the last link state that is propagated by the node, l_i is the current link state and e_{abs} and e_{rel} are the absolute and relative error thresholds, then link state l_i is propagated and remembered locally as the last propagated link state.

The rationale for the threshold based update policy is simple: propagate link state only when not propagating the state will lead to errors beyond acceptable tolerance limits. With such an approach, we expect that we can bound the error in the link state while reducing the cost and the number of updates of link state across the network. While triggered updates of link state are used, for example, in OSPF routing on link up/down events and available link bandwidth changes in [8], we consider granular change in link state and directly relate the impact of the thresholds used in determining the propagation of link state to the final quality of routing goals (both through analysis and simulations).

The allowed or acceptable error thresholds may depend on the link state and the role played by the link state in the routing algorithm. Again, to demonstrate the potential viability of such threshold based updates, we will focus on one routing algorithm, Wardrop routing [9], [22]. The allowed error in link state updates directly gets reflected in allowed error in the path delay metrics used in routing. We assume that nodes operate synchronously in order to make the analysis portion simple. As the link state propagation gets delayed, the routing decisions can be made on stale information. As inaccuracy is allowed in link delays, the resulting decisions can be erroneous, potentially leading to oscilations, where stablity could be guaranteed with exact information.

We will answer the following questions as we go forward: (a) how does the allowed link state update error influence the maximum observed path delay? This reflects on the path quality degradation as a result of threshold based updates. (b) can we still guarantee convergence of Wardrop routing, now albeit a looser notion of convergence i.e., do different paths converge to approximately equal delays (the errors or bounds being determined by (a))? (c) how much gain can be had in reducing the cost of link state updates through threshold based updates?

III. Convergence and Error Analysis

We consider a network represented by a graph $G = (\mathcal{V}, \mathcal{E})$. The traffic demand is specified by a set of commodity flows $\mathcal{K}$ with commodity $k \in \mathcal{K}$ corresponding to traffic λ_k from a source $s_k \in \mathcal{V}$ to a destination $d_k \in \mathcal{V}$. Let $\mathcal{P}$ be the set of all allowed paths connecting source-destination pairs and $\mathcal{P}_k \subseteq \mathcal{P}$ be the set of paths connecting s_k to d_k. Note that each path $p \in \mathcal{P}$ is a set of edges $e \in \mathcal{E}$. We assume that the maximum length of a path in the network is bounded by L. For simplicity, we assume that $\mathcal{P}_k$ are disjoint. The routing state of the network is given by a flow vector $\vec{x} = \{x_p\} \in \mathbb{R}^{|\mathcal{P}|}$, where $|\mathcal{P}|$ denotes the cardinality of $\mathcal{P}$. For a flow vector to correspond to a feasible routing state requires $\vec{x} \geq 0$ and $\sum_{p \in \mathcal{P}_k} x_p = \lambda_k$. The flow on a edge $e \in \mathcal{E}$ is denoted $x^e = \sum_{p \ni e} x_p$. The latency of an edge e supporting flow x^e is specified by a function $l_e(x^e)$, and that the function is upper bounded by $l_{\max}$ for all $e \in \mathcal{E}$. The latency of a path is then given by $l_p(\vec{x}) = \sum_{e \in p} l_e(x^e)$. We denote $l^k_{\min}(\vec{x}) = \min_{p \in \mathcal{P}_k} l_p$.

For simplicity, we focus in this section on the class of adaptive routing policies whose convergence properties under a model with periodic updates have been studied in [10]. Consider a fluid model with an infinite number of agents each make routing decisions for an infinitesimal fraction of the traffic on the network. A flow vector $\vec{x}$ corresponds to x_p agents routing traffic over path p. The route used by each agent is revised periodically at discrete points in time based on the available information about the link and path metrics. The current path delay metric for path p that is available to all agents (potentially with errors) is denoted $\hat{l}_p$. An agent controlling traffic belonging to commodity k and currently using path $p \in \mathcal{P}_k$ samples a path $q \in \mathcal{P}_k$ with probability σ_{pq} and switches to the path q, if it is better, with probability $\mu(\hat{l}_p, \hat{l}_q)$. In this paper, we consider only the subset of policies that are α-smooth, i.e., policies that satisfy $\mu(\hat{l}_p, \hat{l}_q) \leq \alpha(\hat{l}_p - \hat{l}_q)$. In the case of the threshold based scheme, an agent decides that the sampled path q is better only if the sampled latency $\hat{l}_q$ is less than the current latency of path p, $\hat{l}_p$, by an error margin which will be discussed further in the sequel.

A. Convergence to approximate Wardrop equilibria

Through such threshold based updates, we do not aim to converge precisely to a Wardrop equilibrium but instead to an approximate equilibrium defined below.

Definition 1: A flow vector $\vec{x}$ corresponds to a δ - approximate Wardrop equilibrium if: $x_p > 0$ only if $\exists$ a commodity k such that $l_p(\vec{x}) \leq l^k_{\min}(\vec{x}) + \delta$.

As we will show through simulations, such an approximate equilibrium allows the network to leverage the benefits of Wardrop routing at a fraction of the cost.

Lemma 2: If link updates are propagated with an absolute error threshold $e_{abs} < \frac{\delta}{2L}$, and an agent shifts traffic from path p to q only if $(\hat{l}_p - \hat{l}_q) \geq 2Le_{abs}$, and σ_{pq} assigns positive probability to all paths, the flow vector $\vec{x}$ converges to a δ-approximate Wardrop equilibrium.

Proof: Assume that the network is in a routing state $\vec{x}$ that is not a δ-approximate Wardrop equilibrium. When agents sample path with higher latency, no traffic is shifted. If an agent decides that path q is better than p based on the measured latencies, then $(\hat{l}_p - \hat{l}_q) > 2Le_{abs}$. Since link metrics are propagated whenever the change in link delay exceeds e_{abs} and the maximum path length is L, the maximum error in the sampled latency of a path is bounded by Le_{abs}. Thus, using an error margin of $2Le_{abs}$, we are guaranteed that if an agent decides to switch traffic from path p to q based on the propagated link metrics, then indeed $l_p > l_q$ also. Since any feasible path is sampled with positive probability, an agent not on the lowest latency path will eventually sample a better path and switch to it with a positive probability.

As shown in [10], the potential function

$$\Phi = \sum_{e \in \mathcal{E}} \int_0^{x^e} l_e(x) \mathrm{d}x \tag{1}$$

is a Lyapunov function which is minimized at the Wardrop equilibrium. We have shown that, when the network is not at a δ-approximate Wardrop equilibrium, there will eventually be a traffic shift between paths that lowers the potential function while no shifts that increase the function are possible. Thus, the gradient of the potential function is negative as long as the system is not in a δ-approximate Wardrop equilibrium. This in turn implies the lemma. ■

B. Speed of Convergence

In this part, we consider for simplicity the case with one commodity coupled with an adaptive routing scheme using

1) Uniform sampling: $\sigma_{pq} = |\mathcal{P}|^{-1}, \forall p, q \in \mathcal{P}$
2) Linear rule to switch traffic: $\mu(\hat{l}_p, \hat{l}_q) = \frac{(\hat{l}_p - \hat{l}_q)}{l_{\max}}$.

As in [10], we bound the time to reach a (δ, ϵ)-approximate Wardrop equilibrium, defined in [10] as:

Definition 3: If at most ϵ agents use paths p such that $l_p(\vec{x}) > l_{\min}(\vec{x}) + \delta$, then $(\vec{x})$ corresponds to a (δ, ϵ)-approximate Wardrop equilibrium.

Lemma 4: Assume that agents sample alternate paths at a rate ω, and the threshold update policy is used with the absolute error threshold chosen such that $e_{abs} < \frac{\delta}{2L}$. Then, for the uniform sampling policy that shifts traffic between paths following a linear rule, the time spent in a routing state that is not a (δ, ϵ)-approximate Wardrop equilibrium is upper bounded by

$$\frac{|\mathcal{P}| l^2_{\max}}{\omega \epsilon \delta (\delta - 2Le_{abs})} \tag{2}$$

Proof: An agent using a path p such that $l_p(\vec{x}) > l_{\min}(\vec{x}) + \delta$ samples the cheapest path with a probability of at least $\frac{1}{|\mathcal{P}|}$. The agent shifts the controlled traffic to the cheapest path with probability of at least $\frac{(\hat{l}_p - \hat{l}_{\min})}{l_{\max}} \geq \frac{(\delta - 2Le_{abs})}{l_{\max}}$. Also, until we reach the (δ, ϵ)-approximate Wardrop equilibrium, there are at least ϵ such agents. Thus, the rate at which agents switch to the current minimum latency path thereby reducing their latency by at least δ is at least

$$\frac{\omega \epsilon (\delta - 2Le_{abs})}{|\mathcal{P}| l_{\max}}, \tag{3}$$

and the rate at which the potential function decreases is then given by

$$\frac{\omega \epsilon \delta (\delta - 2Le_{abs})}{|\mathcal{P}| l_{\max}}, \tag{4}$$

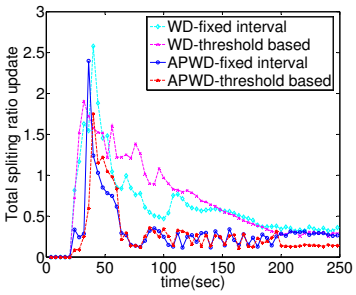
The potential function Φ is bounded above by $l_{\max}$ and below by 0. Thus, the time to reach the (δ, ϵ)-approximate Wardrop equilibrium is bounded by

$$\frac{|\mathcal{P}| l^2_{\max}}{\omega \epsilon \delta (\delta - 2Le_{abs})} \tag{5}$$

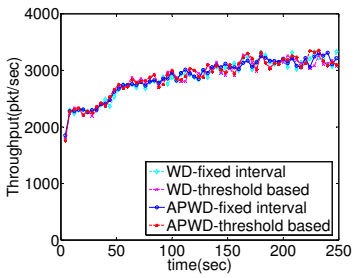
■

IV. Simulation

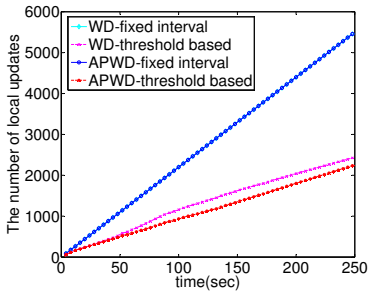
In this section, we compare the behavior of the fixed interval update scheme which sends link updates at fixed time intervals to that of the threshold based update scheme. In the fixed interval update scheme, the interval of link update is set to T = 2 seconds. The link state update, and the traffic splitting ratio changes are based on the algorithm in [9].



(a) Total spliting ratio changes

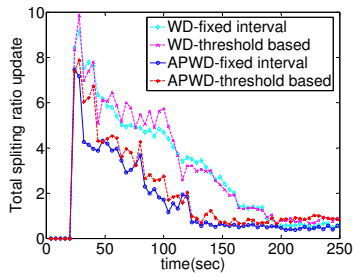


(b) Throughput

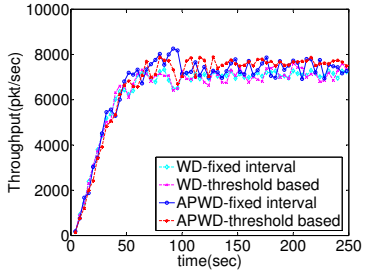


(c) The cumulative number of link updates

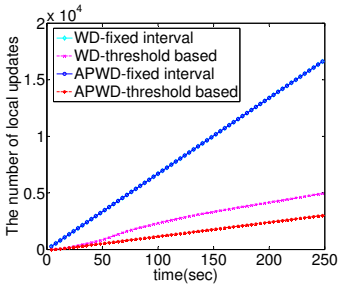
Fig. 1. Spliting ratio, number of link updates, and throughput in NSF topology.



(a) Total spliting ratio changes



(b) Throughput



(c) The cumulative number of link updates

Fig. 2. Spliting ratio, number of link updates, and throughput in Tiscali topology.

It is noted that in order to slow down the propagation of update messages during rapid traffic changes, the link state measurements are controlled by a measurement interval of T = 2 seconds or 1second, even in the threshold based update scheme.

For the link state update, the fixed interval update scheme and the threshold based update scheme set the same interval, T. Since traffic (and hence link latencies) can be very bursty at short timescales, each node measures the current latency ($l(i,j)$), and then updates its link latency by computing an exponential moving average ($\widehat{l}(i,j)$) at every T to smooth out the noise.

$$\widehat{l}(i,j) = \gamma \widehat{l}(i,j) + (1-\gamma) l(i,j) \tag{6}$$

For propagating the link state and updating the path latency, we define $L(i,j,k)$ and $L_P(k,j)$.

- $L(i,j,k)$: the expected path latency from i to j via neighbor k.

$$L(i,j,k) = \widehat{l}(i,k) + L_P(k,j) \tag{7}$$

- $L_P(k,j)$:the expected path latency from k to j

$$L_P(k,j) = \sum_{n_j \in N(k,j)} p(k,j,n_j) L(k,j,n_j) \tag{8}$$

where $p(i,j,k)$ is the splitting ratio from i to j via k, and $N(k,j)$ is the neighbor of k for the destination j.

$L(i,j,k)$ is computed and $L_P(i,j)$ is propagated to the network based on the update policy of the schemes. In the fixed interval update scheme, $L(i,j,k)$ is computed and $L_P(i,j)$ is propagated at every T. In the threshold based update scheme, $L(i,j,k)$ is computed at every T, but $L_P(i,j)$ is propagated only when the change of $\widehat{l}(i,j)$ is greater than $min(e_{abs}, e_{rel}\widehat{l}(i,j))$. In all simulation, e_{abs} is set to 7.5 msec and e_{rel} is set to 0.1. L is 3.

The splitting ratio update is determined by the condition of the (δ,ϵ)-approximate Wardrop equilibrium. The traffic splitting ratios $p(i,j,k_1)$ are changed only when $L(i,j,k_1) - L(i,j,k_2) > e$ where $e = min(e_{abs}, e_{rel} * \widehat{l}i,j) * 2L$.

The amount of change (Δ) in traffic splitting ratios is given by, from [9],

$$\Delta = \lambda \left((1-\beta)\, p\,(i,j,k_1) + \frac{\beta}{|N(i,j)|} \right) \frac{L(i,j,k_1) - L(i,j,k_2)}{L(i,j,k_1) + \alpha} \tag{9}$$

The details about weigh shift factor(λ), virtual latency offset α, and exploration ratio (β) are desgined for preventing oscillations and exploring new paths. (Please refer [9]).

We consider the topologies ranging from simple four node graphs to medium and large topologies in [13]. We assign identical weights for links so that hop count is the routing cost, and it increases equal cost multiple paths. To simulate traffic on the networks, we generate a workload based on the Web workload generators in [12]. The workload mimics that generated by a user requesting a web page, and then remaining idle for a period while reading the page and then requesting another web page and so on. The sizes of the files (requests) are drawn from a heavy-tail distribution. This results in a mix of short-term flows and a considerable number of long-term flows.

In all simulations, the threshold-based scheme is simulated

along with the fixed interval update scheme to compare their performance. We measure the following metrics.

- Total spliting ratio changes: the sum of the spliting ratio change(Δ) at time t (for subsection A)
- Throughput: the sum of the throughput at time t (for subsection A)
- Spliting ratio: $p(r_1, r_4, r_2)$ and $p(r_1, r_4, r_3)$ (for subsection B)
- Route utilization: $U(r_1 - r_2 - r_4)$ and $U(r_1 - r_3 - r_4)$ where $U(P)$ is the utilization of path(P) (for subsection B)
- Path latency gap($G_p(t)$): $L(r_1, r_4, r_2)$-$L(r_1, r_4, r_2)$ at time t (for subsection C and D)
- Cumulative path latency gap: $\sum_{i=1}^{t} G_p(i)$ (for subsection C
- The number of link updates: the cumulative number of update message (for subsection A,B,C, and D)

A. Approximate versus Exact Wardrop comparison

In this section, we present results from a simulation to compare the approximate Wardrop routing (apWD) with exact Wardrop routing (WD). We employed NSF topology for this simulation [13] with 14 nodes and 22 links for medium size network simulation, and tiscali topology with 40 nodes and 67 links for large size network simulation. The results from simulations in other topologies and workloads are similar.

The results are shown in Fig. 1 and Fig. 2. We present both timer-based link updates and threshold-based updates with exact and approximate Wardrop equilibria. The results show that approximate Wardrop routing reaches similar performance as exact Wardrop routing, within the allowed error bounds. It is noted that both the schemes employed threshold based (or timer based) link updates and that the only difference in the two schemes is the goal for convergence. In the exact Wardrop routing case, the traffic splitting ratios are updated to make the differences in path delays go to zero and in the approximate case, the traffic splitting ratios are updated to make the differences in the path delays approach the allowed error bounds for the paths.

We measure the sum of all route spliting ratio changes at all routers to test the oscilation in the schemes. We expect the measurement to show quick convergenc and small fluctuations after the convergence. Also, we expect adjustment of spliting ratios to contribute to the improvement of throughput of TCP connections.

In this experiment, 6 clients generate HTTP traffic between every source-destination pair during the simulation time (=250 sec). The splitting ratio chage is allowed after 20 sec.

Both schemes show small fluctuations of total spliting ratio changes after 70 seconds as seen in Fig. 1(a) and after 120 seconds as seen in Fig. 2(a). As the traffic spliting ratio is changed, the total throughput increased in both the networks, especially during the period between 30s and 60s. Even though both schemes show similar performance, the link update overhead of the threshold based update scheme is less than 50% compared to that of the fixed interval update scheme. (in Fig.1(c) and Fig.2(c))

The results show that approximate Wardrop routing provides similar performance, with a slightly lower number of traffic splitting updates and slightly lower number of link updates, within each type of link update mechanism. It is also noted that the threshold based mechanisms required far fewer link updates than timer based link update mechanisms.

From here on, we will only consider approximate Wardrop equilibrium as we consider more experiments.

B. Simple topology comparison

The first simulated network topology consists of 4 routers. (As shown in Fig. 6). The 20 clients connected with r_4 download the HTTP files from the routers connected with r_1. There are two possible paths: r_1-r_2-r_4, and r_1-r_3-r_4 between r_1 and r_4.

To study the performance of the Wardrop routing scheme using threshold based updates, we examine the spliting ratio and the route utilization of the two routes over time under three cases. We assign bandwidths of 1, 5, and 10 Mbps to link(r_1, r_2) and link(r_2, r_4) while keeping the other links at 10Mbps. These three cases are denoted 1/10 Mbps, 5/10 Mbps and 10/10 Mbps. If the algorithm works well, we expect the load on both routes to be balanced. The optimal spliting ratio, ignoring the traffic variability, should consequently converge to 0.09/0.91, 0.33/0.67, and 0.5/0.5 respectively.

In 3(a),4(a), and 5(a), both schemes converge to near optimal spliting ratios. Then we can see the route utilizations converge to nearly the same utilization on both paths in both schemes (in fig 3(b),4(b),and 5(b)). However, the number of link updates of the fixed interval update scheme is much higher than the number of link updates of the threshold based update scheme (in fig 3(c),4(c),and 5(c)). The fixed interval update scheme keeps sending the path latency information after convergence. However, the threshold based update scheme sends the path latency information scarcely after convergence.

C. Impact of traffic bursts

We assume that the traffic demands are static in the previous subsection. However, the dynamic routing protocol should respond to the changes in the traffic rapidly. In this simulation, we examine if the threshold based update scheme can adjust quickly to chages in the traffic.

We simulate adding a traffic burst to different links to see the impact of how quickly the traffic burst results in making traffic adjustments across the two paths and how much overhead is required for adjusting to changes in traffic. We generate

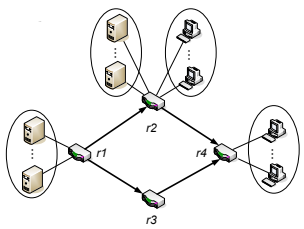
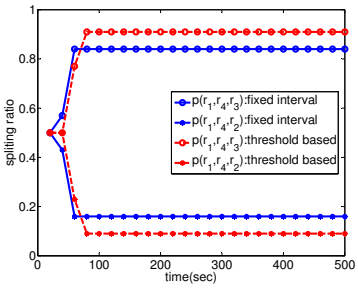
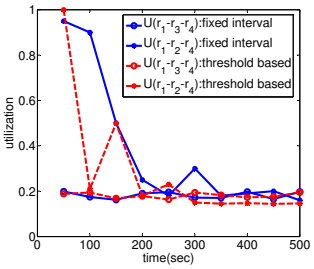


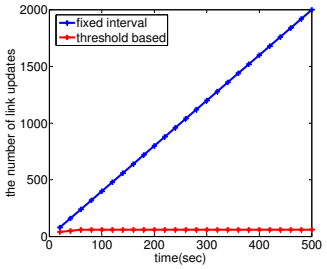
Fig. 6. Simple topology



(a) Spliting ratio to the routes

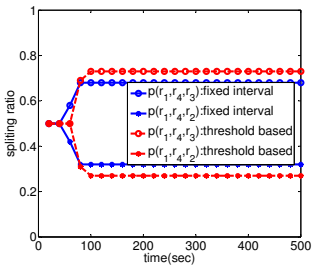


(b) Route utilization

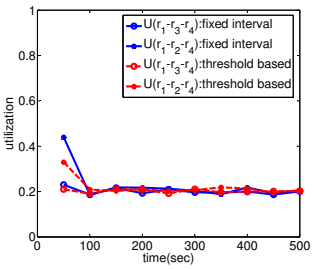


(c) The cumulative number of link updates

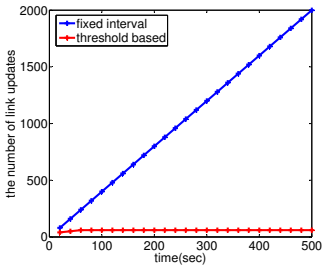
Fig. 3. Spliting ratio, route utilization, and the number of link updates for the case of 1/10 Mbps



(a) Spliting ratio to the routes

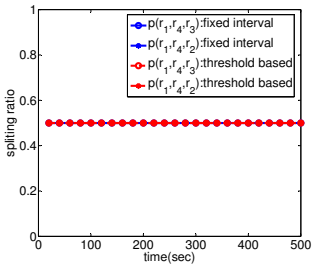


(b) Route utilization

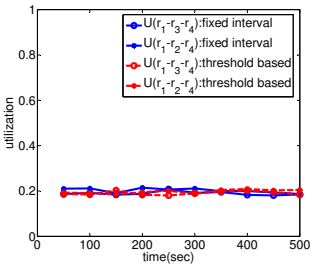


(c) The cumulative number of link updates

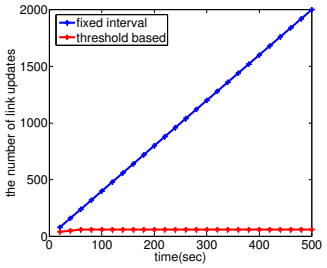
Fig. 4. Spliting ratio, route utilization, and the number of link updates for the case of 5/10 Mbps



(a) Spliting ratio to the routes



(b) Route utilization



(c) The cumulative number of link updates

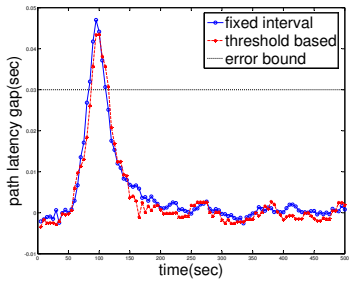
Fig. 5. Spliting ratio, route utilization, and the number of link updates for the case of 10/10 Mbps

additional traffic on one or more links in the network of Fig. 6. At the begining of the simulation, clients connected to r_4 download HTTP files from the routers connected with r_1. Additional HTTP clients, become active after a burst start time (=50 sec). Before the burst of traffic due to additional HTTP clients, the traffic is balanced well between the two routes. The traffic ratios are changed as a result of this traffic burst.

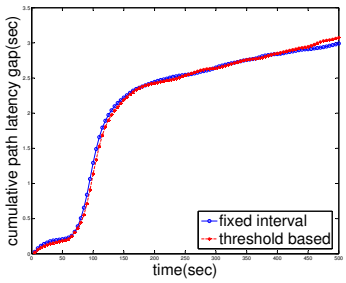
In scenario 1, all link capacities are 2 Mbps and the burst traffic is put on link (r_1, r_2). In this case, r_1 detects the traffic change through link latency measurements. In addition, r_1 has an alternative path (path$(r_1 - r_3 - r_4)$) avoiding the congested path (path$(r_1 - r_3 - r_4)$).

In scenario 2, all link capacities are 2Mbps and the burst traffic is put on link(r_2, r_4). Node r_1 cannot detect the traffic change directly. Instead, r_2 can detect the traffic change, but it does not have any alternative path. In this case, r_2 propagates the traffic change with link updates, and r_1 changes the spliting ratio based on the received updates. We examine how fast link update is propagated with the threshold based update scheme.

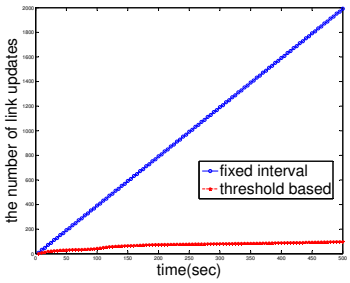
In scenario 3, we consider different link capacities by setting link capacities on link(r_1, r_3) and link(r_3, r_4) to 4Mbps and 4/3Mbps respectively and the burst traffic is put on link(r_1, r_2) while the capacities of link(r_1, r_2) and link(r_2, r_4) are 2 Mbps. In this setting, the path latencies of two paths is the same as before the burst traffic. We put the burst traffic on link(r_1, r_2). Similar to scenario 1, r_1 detects the traffic change and it has an alternative path. However, the link update is important in this scenario. Since, the capacity of link(r_3, r_4) is less than that of link(r_2, r_4), swithcing the traffic on link(r_1, r_2) to link(r_1, r_3) can cause another congestion on link(r_3, r_4). The spliting ratio change is dependent both on the link state update in this scenario.



(a) Path latency gap

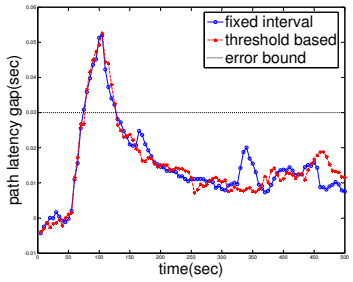


(b) Cumulative path latency gap

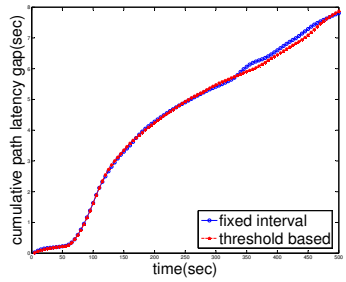


(c) Cumulative number of link updates

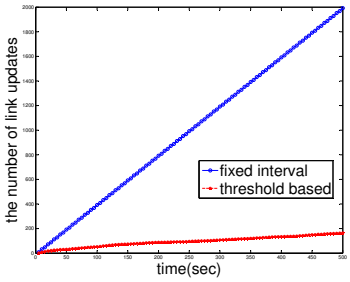
Fig. 7. The path latency gap and the number of link updates for scenario 1



(a) Path latency gap

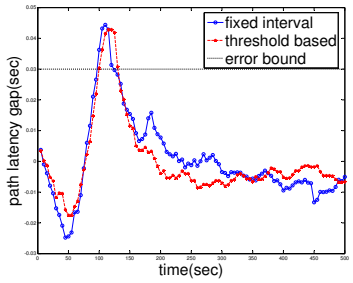


(b) Cumulative path latency gap

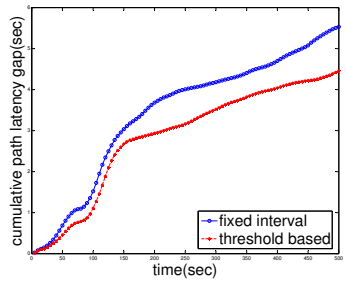


(c) Cumulative number of link updates

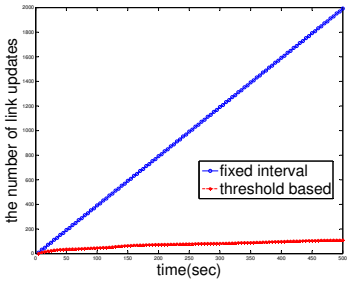
Fig. 8. The path latency gap and the number of link updates for scenario 2



(a) Path latency gap



(b) Cumulative path latency gap



(c) Cumulative number of link updates

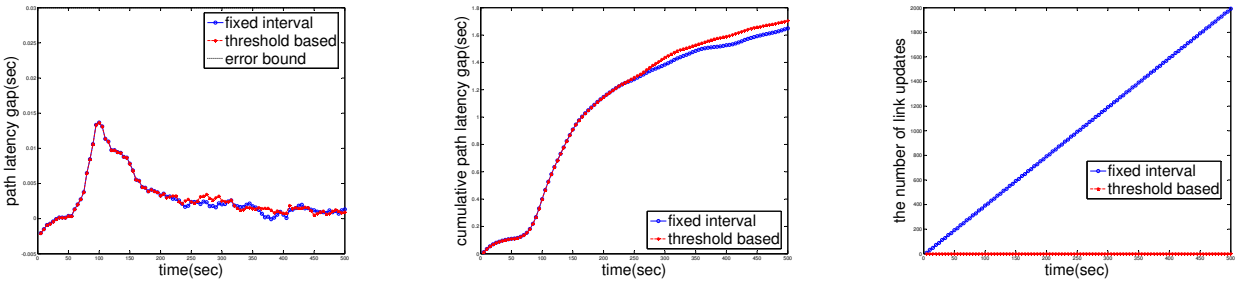
Fig. 9. The path latency gap and the number of link updates for scenario 3

In scenario 1,2 and 3, the fixed interval update scheme and the threshold based update scheme both change the traffic splitting ratios. We measure the path latency gap ($|L(r_1, r_4, r_2)$-$L(r_1, r_4, r_3)|$). The behaviors of the path latency gap are similar in both schemes (in fig. 7(a),8(a), and 9(a)). However, the difference in the number of link updates in the two schemes is significant as seen in 7(c),8(c), and 9(c).

In scenario 1, 2, and 3, the link utilizations are high before the burst traffic. If the link utilization is high when the additional burst of traffic joins the link, the link latency is likely to change and hence impact the path latency. However, if the link utilization is low before the new burst of traffic joins the link, the path latency may not be affected much. As a result, the traffic splitting ratio across available paths may not change. The threshold based update scheme does not exchange the link latency information in this case, so we can reduce the overhead. For simulation of this case, all link capacities are 4 Mbps and the burst traffic is put on link (r_1, r_2) (we refer this as scenario 4).

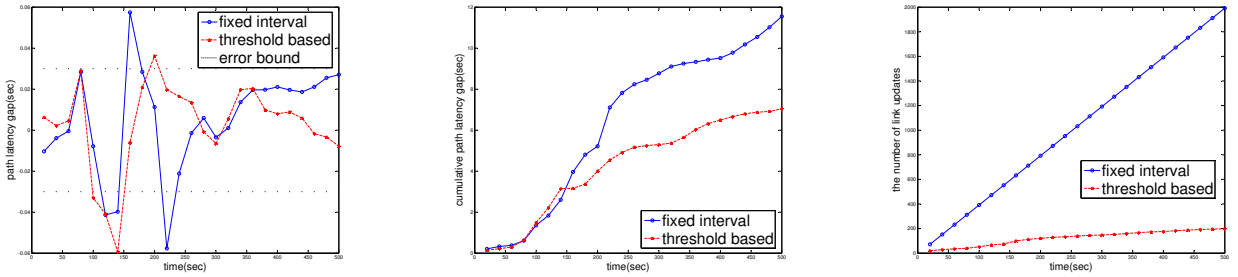
Under the low utilization scenario, the burst traffic does not make the path latency gap higher than the error bound (in fig. 10(a)). As a result, there is no change to traffic splitting ratios across the paths. In this case, there are little or no link updates in the theshold based scheme while the fixed interval update scheme continues sending the same number of link updates as seen in Fig. 10(c).

To see the adoption for dynamic change of the traffic, we do the following simulations. The capacities of all links are set to 2Mbps. The burst traffic is on link(r_1, r_2) between 50 sec and 450 sec, and the burst traffic is on link(r_1, r_3) between 100 sec and 400 sec. Node r_1 can react these traffic changes with measuring the link state. In addition, The burst traffic is



(a) Path latency gap (b) Cumulative path latency gap (c) Cumulative number of link update

Fig. 10. The path latency gap and the number of link updates for scenario 4



(a) Path latency gap (b) Cumulative path latency gap (c) Cumulative number of link updates

Fig. 11. The path latency gap and the number of link updates for scenario 5

on link(r_2, r_4) between 150 sec and 350 sec, and the burst traffic is on link(r_3, r_4) between 200 sec and 300 sec. We can test how both schemes react to the dynamic change of traffic (We refer this scenario 5).

Even in such dynamic situation of rapid fluctuations in traffic across different links, threshold-based updates maintain the paths within the error bounds (As seen in fig. 11(a)). Fig. 11(b) shows that the cumulative path latency gap, that measures the total cumulative difference in path quality across the simulation time, is slightly better for threshold-based updates. It is again noted that the threshold-based propagation requires far fewer updates to reach similar routing goals. The number of link updates in the threshold based update scheme is also much lower than the fixed interval update scheme.

D. Impact of the update interval

In this section, we test the impact of the update interval. We use the different T: T= 2 sec and T=1 sec. The more frequent updates are expected to lead to faster convergence.

For the simulation, we put a periodic burst traffic from 50 second on link(r_2, r_4) of the network (fig. 6). The burst traffic is active in the first half of the burst traffic period, and then the burst traffic is inactive in the second half of the burst traffic period. We exame three burst traffic periods: 100, 50, and 20 seconds. If the burst traffic period is long enough to make spliting ratio near to the optimal spliting ratio, the path latency gap does not exceed the error bound in next burst traffic period. This simulation is expected to test the speed of convergence.

Both update schemes with T=1 update the increased latency information faster than those with T=2. As a result, both update schemes with T=1 achieve convergence earlier than those with T=2 (in fig. 12(a), 12(b), and 12(c)). Even worse, both update schemes with T=2 do not converge at the end of the simulation if the burst traffic period is 20 sec. The update scheme with T=1 outperform the update scheme with T=2, but the problem is overhead of link updates. As seen in fig. 13(a), 13(b), and 13(c), the number of link updates in the update schemes with T=1 is twice of that in the update schemes with T=2. However, the threshold based update scheme with T=1 has lower overhead than that with the fixed interval update scheme with T=2.

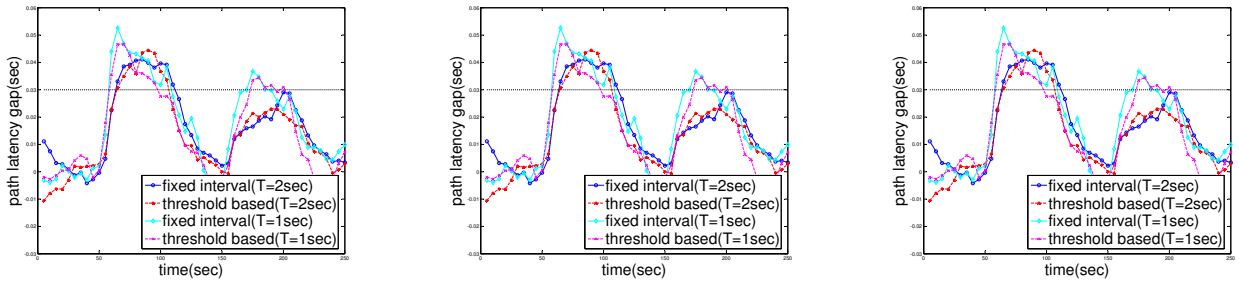
V. Related Work

Dynamic routing has received much attention. Early ARPAnet considered link delays as a cost metric and the resulting oscillations prompted the use of other metrics based on capacity.

In most networks, the link costs are now determined based on traffic engineering considerations [1], [14].

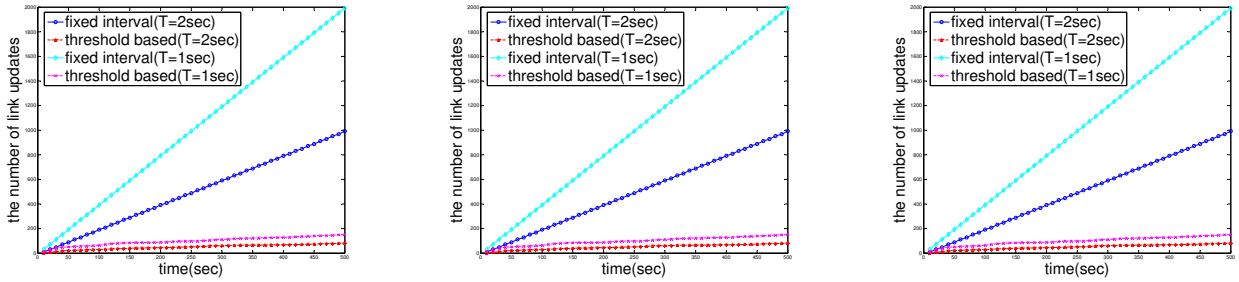
Simultaneous traffic engineering and routing table computation is considered in DEFT [15]. DEFT splits a flows's traffic across multiple paths based on an exponential function of the path delay differences, preferring smaller delay paths.

Recently, intelligent route control devices have been employed to route traffic efficiently when stub networks are multi-homed [25]. These devices measure path delays through Internet and utilize this information in making decisions on



(a) Burst traffic period=100sec (b) Burst traffic period=50sec (c) Burst traffic period=20sec

Fig. 12. Path latency gap with the different T on the periodic burst traffic



(a) Burst traffic period=100sec (b) Burst traffic period=50sec (c) Burst traffic period=20sec

Fig. 13. The number of link updates with different T with periodic burst traffic

which of the available network connections will be utilized for routing traffic. Oscillations and convergence issues are considered [20], [21]. In such systems, path delay information is obtained at the end stub network, through passive or active measurements and individual network link state is not propagated by the network elements.

Dynamic routing has been studied widely. Dynamic routing is proposed recently for wireless networks [22], and for dynamic traffic engineering in wired networks [9] and for multipath adaptive routing [23]. These approaches utilize network delays for making routing decisions. These approaches rely on network elements propagating the dynamic link state information around the network and hence can directly benefit from our work reported here. The problem of reducing link state updates have been studied previously in [30], [31] for QOS routing and shortest-path routing. Our threshold based update scheme combines both fixed and relative error thresholds in order to be more widely applicable and we formally prove the impact of our scheme on stability of the routing paths along with bounding the expected errors.

The work [8] considered triggered updates of available bandwidth for QOS routing. The updates are triggered based on inverse of the available bandwidth at a link. It was shown that the triggered updates can be effective for QOS routing. Available link bandwidth is a dynamic link metric, as it fluctuates with call admission and departure in QOS routing. Our work is similar to this earlier work, but focuses on utilizing link delays as the dynamic link metric in Wardrop routing, and rigorously examines the effect of the triggered update rate on algorithm performance.

While the focus in [30] and [8] is on simulation study, we additionally analytically bound the errors in link delays and study the impact on the performance of the routing algorithm in reaching approximate Wardrop equilibrium. In [8], no generally applicable answer to the question of "how big a change in the link metric is significant?" is determined. In this work, we present a method to set the threshold and thus the rate of triggered updates based on the deviation from the exact equilibria that is tolerable as well as the acceptable rate at which the algorithm converges.

Delay and convergence analysis of Wardrop routing with regular update of link state information is studied in [10], [11]. Our analysis here shows that threshold-based updates can reach convergence (within error bounds instead of exact equalization of delays) with less restrictive assumptions. Note that, in [11], the link latency function is assumed to have a bounded slope. This condition is satisfied, for example, when the growth of the latency function is polynomially bounded. Both the interval at which the regular link updates must be sent as well as the speed of convergence depend on the upper bound of the rate at which link latency can grow. Thus, the periodic update policy will have to be reconfigured for different networks with different link latency characteristics and also when networks expand by adding new links. The threshold based update policy proposed in this paper will adaptively vary the rate of updates, implicitly taking into account changes in the rate of growth of the latency function both as the regime of operation (low or high load) as well as the

network topology vary. Also, since link latency also includes queueing latency, the rate of growth of link latency in many practical systems cannot be bounded by polynomial functions. Consider for example, even a simple M/M/1 queueing model to see that this is true. The convergence and indeed even the speed of convergence of the threshold based policy does not depend on the characteristics of the latency function, enabling its use in various practical systems with queueing delay at the links. Thus, the threshold based policy will adaptively vary the rate of updates and converge to the approximate Wardrop equilibrium without requiring configuration based on knowledge of the link latency growth rates and can be used even in systems with link delays that can grow at unbounded rates depending on the load on the link.

REPLEX [9] proposed Wardrop routing for dynamic traffic engineering purposes. Our work is motivated by this and other earlier work on Wardrop routing. REPLEX utilized periodic updates of state and used the most recent observed state in making routing decisions. Our work used REPLEX for comparing the threshold-based updates to the fixed interval updates. Our results are equally applicable whether Wardrop equilibrium is used for routing or for traffic engineering purposes among the paths made available by the underlying routing algorithm.

Our work here focused on reaching approximate equilibrium in Wardrop routing (a form of multi-path routing). However, the results can be generalized for other routing algorithms. Our work does not attempt to exactly equalize the delays across different paths. However, this loss in exact equilibrium is well compensated by the observed reduction in update traffic of the link state information.

VI. Conclusion

In this paper, we have considered the problem of reducing the cost of updating link state for dynamic routing algorithms. We have proposed threshold based link state updates that can limit the errors on paths in network to known bounds to enable reaching approximate equilibria of dynamic routing. In particular, the threshold based scheme is shown to enable reaching approximate Wardrop equilibria with quantifiable bounds in the differences in different path latencies. Our simulations have shown that threshold-based updates can reduce the cost of link updates significantly compared to a model of fixed-interval updates. Our results show that the number of link updates can be reduced by range from 50% to 90%. These results are encouraging and may point to making dynamic routing more viable.

Acknowledgement

This work is supported in part by NSF grants 0702012 and 0621410 and grants from Qatar National Research Foundation and Qatar Telecom. Part of this work was done while Narasimha Reddy was on a sabbatical at the University of Carlos III and Imdea Networks in Madrid, Spain.

References

[1] B. Fortz, J. Rexford, and M. Thorup, "Internet traffic engineering by optimizing OSPF weights" *Proc. IEEE INFOCOM*, 2000

[2] L. Buriol, M. Resende, C. Ribeiro, and M. Thorup "A memetic algorithm for OSPF routing" *Proc. the 6th INFORMS Telecom*,2002

[3] A. Sridharan, R. Guerin, and C. Diot "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks" *IEEE/ACM Trans. Network*,2005

[4] A. Elwalid, C. Jin, S. Low, and I. Widjaja "MATE:MPLS adaptive traffic engineering" *Proc. IEEE INFOCOM*,2001

[5] S. Kandula, D. Katabi, B. Davie, and A. Charny "Walking the tightrope: responsive yet stable traffic engineering" *Proc. ACM SIGCOMM*,2005

[6] Manish Jain and Constantinos Dovrolis "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput" *Proc. ACM SIGCOMM*,2002

[7] V. Raghunathan and P. R. Kumar, " Wardrop Routing in Wireless Networks " *IEEE Trans. Mobile Computing*, vol. 8, No. 5, 2009

[8] A. Shaikh, J. Rexford and K. G. Shin "Evaluating the impact of stale link state on quality-of-service routing" *IEEE/ACM Trans. Networking*, 2001

[9] S. Fischer, N. Kammenhuber, and A. Feldmann, " REPLEX: dynamic traffic engineering based on wardrop routing policies " *Proc. ACM CoNEXT*, 2006

[10] S. Fischer, H. Racke, and B. Vocking, "Fast convergence to Wardrop equilibria by adaptive sampling methods" *Proc. 38th Ann. ACM. Symp. on Theory of Comput. (STOC)*, 2006

[11] S. Fischer and B. Vocking "Adaptive routing with stale information" *Proc. 24th Ann. ACM SIGACT-SIGOPS Symp. on Principles of Distributed Computing (PODC)*, 2005.

[12] C. Vollmert. " A Web workload generator for the SSFNet network simulator. " *Bachelors thesis, Technische Universitat Munchen*,2004

[13] Rocketfuel topology mapping. *WWW http://www.cs.washington.edu*.

[14] A. Sridharan, R. Guerin and C. Diot "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks" *IEEE/ACM Trans. Networking*,2002

[15] D. Xu , M. Chiang , J. Rexford "DEFT: Distributed exponentially-weighted flow splitting" *Proc. IEEE INFOCOM*, 2007

[16] E. Crawley and R. NAir and B. Rajagopalan and H. Sandick "A framework for QOS-based routing in the internet" *Internet draft*,1997

[17] I. Matta and U. Shankar "Type-of-service routing in datagram delivery systems" *IEEE Jour. on Selec. Areas in Commun.*,1990

[18] Q. Ma and P. Steenkiste "Supporting Dynamic Inter-Class Resource Sharing: A Multi-Class QoS Routing Algorithm" *Proc. IEEE INFOCOM*,1999

[19] S. Kweon and K. Shin "A new Distributed QOS routing algorithm based on Fano' s method" *Computer Networks Journal*,2005

[20] R. Gao, C. Dovrolis and E. Zegura "Avoiding Oscillations due to Intelligent Route Control Systems" *Proc. IEEE INFOCOM*,2006

[21] Y. Liu and A. L. Narasimha Reddy, "Multihoming route control among a Group of Multihomed Stub Networks" *Journal on Computer Communication*,2007

[22] V. Borkar and P. Kumar "Dynamic Cesaro-Wardrop equilibration in networks" *IEEE Tran. Automatic Control*,2003

[23] A. Kvalbein, C. Dovrolis and C. Muthu "Multi-path load-adaptive routing: Putting on the emphasis on robustness and simplicity" *Proc. of ICNP*,2009

[24] A. Khanna and J. Zinky "The revised ARPANET routing metric" *Proc. ACM SIGCOMM*,1989

[25] A. Akella, B. Maggs, S. Seshan, A. Shaikh and R. Sitaraman "A measurement-based analysis of multihoming" *Proc. ACM SIGCOMM*,2003

[26] C.Zhanga, J. Kurose, D. Towsley, Z. Ge and Y. Liu "Optimal routing with multiple traffic matrices: Tradeoff between average and worst case performance" *Proc. ICNP*,2005

[27] Y. Zhang, M. Roughan, C. Lund and D. Donoho "An information-theoretic approach to traffic matrix estimation" *Proc. ACM SIGCOMM*,2003

[28] A. Lambert, M.-O. Buob, and S. Uhlig, Improving internet-wide routing protocols convergence with MRPC timers, *CoNEXT*, 2009

[29] A. Basu, C.-H. L. Ong, A. Rasala, F. B. Shepherd, and G. Wilfong, Route oscillations in I-BGP with route reflection, *ACM SIGCOMM,* ,2002

[30] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi, "Quality of service based routing: a performance perspective," *ACM SIGCOMM*, 1998.

[31] K. Levchenko, G. M. Voelker, R. Paturi, and S. Savage, "XL: An Efficient Network Routing Algorithm," *ACM SIGCOMM*, 2008.