Distributed Measurement-Aware Routing: Striking a Balance between Measurement and Traffic Engineering

Chia-Wei Chang*, Han Liu† Guanyao Huang†, Bill Lin*, and Chen-Nee Chuah†
* Department of Electrical and Computer Engineering, University of California, San Diego
† Department of Electrical and Computer Engineering, University of California, Davis

***Abstract*—Network-wide traffic measurement is important for various network management tasks, ranging from traffic accounting, traffic engineering, and network troubleshooting to security. Existing techniques for traffic measurement tend to be sub-optimal due to poor choice of monitor deployment location or due to constantly evolving monitoring objectives and traffic characteristics. It is not feasible to dynamically reconfigure/redeploy monitoring infrastructure to satisfy such evolving measurement requirements. In this paper, we present a *distributed measurement-aware* traffic engineering protocol based on a game-theoretic re-routing policy that attempts to optimally utilize existing monitor locations for maximizing the traffic measurement gain while ensuring that the traffic load distribution across the network satisfies some traffic engineering constraint. We introduce a novel cost function on each link that reflects both the measurement gain and the traffic engineering (TE) constraint. Individual routers compete with each other (in a game) to minimize their own costs for the downstream paths, i.e., each router dynamically gathers its cost information for upstream routers and use it to locally decide how to adjust traffic split ratios for each destination to the next-hop routers among these multiple equal-cost paths. Our routing policy guarantees not only a provable Nash equilibrium, but also a quick convergence without significant oscillations to an equilibrium state in which the measurement gain of the network is close to the best case performance bounds We evaluate the protocol via simulations using real traces/topologies (Abilene, AS6461 and GEANT). The simulation results show fast convergence (as expected from the theoretical results), improved measurement gains (e.g., 12 % higher) and much lower TE-violations (e.g., up to 100X smaller) compared to static, centralized measurement-aware routing framework in dynamic traffic scenario.**

I. Introduction

Achieving accurate and efficient network-wide traffic measurement is often plagued with multi-faceted challenges. While packet and flow sampling mechanisms are widely deployed (e.g, NetFlow [1]), detailed packet capture and analysis (e.g., deep packet inspection) is computationally expensive. Hence, typically only a subset of nodes are equipped with such high-fidelity monitoring capabilities. To reap the maximum measurement benefits without incurring huge deployment costs, these high-fidelity monitors need to be configured properly and strategically placed across the network. Most previous work on the latter domain focused on deriving the optimal monitor placement that maximizes the monitoring utility for a given routing and traffic profile. They are typically intended for longer time-scales and assume a priori knowledge about the traffic characteristics. However, both traffic characteristics and measurement objectives can dynamically change over time, rendering a carefully designed placement of monitors sub-optimal. To address these limitations, a measurement-aware routing framework, MeasuRouting, was recently proposed to assist traffic measurement [2]. It introduces routing as another degree of freedom and intelligently routes traffic sub-populations over pre-deployed monitors to maximize the traffic measurement gain. However, MeasuRouting requires the existence of centralized controller and offline analysis to find the optimal routing strategies for every traffic sub-populations, which is unrealistic in production IP networks. It can therefore only be interpreted as the best-case performance bounds for routing-assisted measurement.

In this paper, we present *Distributed MeasuRouting (DisMR)*, a new traffic engineering protocol that attempts to optimally utilize existing monitor locations for maximizing the traffic measurement gain while distributing the traffic load evenly across the network. DisMR takes advantage of alternative paths in a network by leveraging existing equal-cost multi-path (ECMP) routing. It maximizes the traffic measurement gain by adjusting the traffic split ratios among these paths to the same destination. DisMR is derived from a game-theoretic re-routing policy that captures the dynamic decision-making process and interactions among distributed routers. In our model, we design a cost function on each link that reflects both the measurement capability and TE constraint, i.e., links with larger measurement resources have a smaller cost but links with a larger TE score (e.g., link utilization) have a larger cost. The cost function is designed such that flows are attracted to links with better measurement capabilities while avoiding TE violations. Routers compete with each other in a game-theoretic manner in order to minimize their own costs for the downstream paths. In DisMR, every router periodically gathers/propagates sub-path cost information for upstream routers. Based on this information, each router makes local decisions on how to adjust routing split ratios for each destination traffic to the next-hop routers among these multiple equal-cost paths. Our routing policy guarantees not only a provable Nash equilibrium but also a fast convergence without significant oscillations. Meanwhile, the measurement gain of the network at the equilibrium state is close to the maximum achievable gain calculated using offline, centralized MeasuRouting. Previously, REPLEX [3], TeXCP [4] and MATE [5] have been proposed as dynamic TE solutions to minimize the path latency or the link utilization by adjusting the split ratios of traffic among the paths with the

same ingress/egress nodes. In contrast to them, our introduced link cost function is a novel combination of link measurement ability and TE constraint. Moreover our path cost is defined as the product of link costs instead of traditional summation operation. We outline our contributions as follows:

- We de-centralize MeasuRouting in a game-theoretic setting and propose a novel cost function that balances the potentially contradicting measurement and traffic engineering objectives. The cost function is designed to encourage flows to be routed through monitors with abundant resources while avoiding TE violation and we prove the existence of Nash equilibrium on the new cost function.
- We design a new traffic engineering protocol, *Distributed MeasuRouting (DisMR)*, based on the routing game. DisMR converges fast to equilibrium point and achieves comparable measurement gain with centralized MeasuRouting in static traffic scenario.
- We evaluate DisMR via simulations using real traces from Abilene [6], AS6461 [7], and GEANT [8]). The simulation results show fast convergence (as expected from the theoretical results), improved measurement gains (e.g., 12 % higher) and much lower TE-violations (e.g., up to 100X smaller) compared to static, centralized MeasuRouting in dynamic traffic scenario.

The rest of this paper is organized as follows: We first prove the existence of equilibrium on the new cost function in Section II. We next study the rerouting policies in a "dynamic round-based" variant of equilibrium and present practical *Distributed MeasuRouting* algorithm in Section III. We show it to be stable and converge quickly in a game-theoretic model under realistic conditions and present performance evaluation in Section IV. Section V concludes the paper.

II. Adaptive traffic measurement problem

In this section, we formulate the Distributed MeasuRouting problem in a game-theoretic setting. It strikes the balance between maximizing measurement gain of the network and minimizing the TE violations by introducing two novel definitions: Ψ (effective non-sampling rate) and ζ (link penalty function). We present theoretical results regarding the static convergence of the game. Note that our work differs fundamentally from Beckmann's work [9] in that our introduced link cost function is a novel combination of link measurement ability and TE constraint while their link cost function represents only TE metric (e.g., latency or link utilization). Moreover, our path cost function is defined as the product of link costs, which makes the proofs of existence of Nash Equilibrium different from [9]. The dynamic behavior of this game and its distributed implementation are presented in next section.

We consider a measurement objective of maximizing G (*sampling resolution function*), which characterizes the overall measurement utility of the whole network. In contrast to MeasuRouting [2],we assume independent uniform sampling across all participated routers, where each router independently selects a packet with a sampling probability (typically between 0.001 and 0.01) and aggregates the selected packets into flow records (e.g., via Netflow [1]). Let S_a be the given fixed sampling rate at every arc $a \in \mathcal{A}$ where $\mathcal{A}$ stands for the set of all arcs in the network. The total effective sampling rate of a path $P \in \mathcal{P}$ with respect to flow set, $[f] = \{f_P, P \in \mathcal{P}\}$ is defined as: $S_P(f_P) = 1 - \prod_{a \in P} (1 - S_a)$. Therefore $G(f) = \sum_{P \in \mathcal{P}} S_P(f_P) \cdot f_P$. We define Ψ_a to be the effective non-sampling rate at arc $a \in \mathcal{A}$: $\Psi_a = 1 - S_a$. The total non-sampling rate of a path $P \in \mathcal{P}$ with respect to f_P is then the product of the non-sampling rate of the arcs on that path: $\Psi_P(f_P) = \prod_{a \in P} \Psi_a(f_P), P \in \mathcal{P}$. Therefore the total non-sampled amount is defined as $C(f) = \sum_{P \in \mathcal{P}} \Psi_P(f) \cdot f_P$. Given fixed traffic demand, maximizing $G(f)$ could be equivalent to minimize the cost function $C(f)$.

Our goal is to let the flow sets at each end point route their traffic selfishly to better learn a Nash equilibrium of non-sampling rate while adhering to traffic engineering constraints. However, in a distributed environment, flow sets will all choose the best paths with minimum $\Psi_P(f)$ and may overload some specific arcs. This is because Ψ_a at every arc $a \in \mathcal{A}$ is constant (e.g., sampling rates do not adapt to the traffic amount). In order to reflect TE constraints, we add penalty function $\zeta(f)$ to Ψ_a, i.e., $\Psi_a(f) = \Psi_a + \zeta(f)$ for each arc $a \in \mathcal{A}$. We design the $\zeta(f)$ such that its value increases sharply when the traffic amount is above the TE-constraint (e.g., maximum link utilization), otherwise it will stay at zero. Therefore, $\Psi_a(f)$ becomes a function of traffic for every arc $a \in \mathcal{A}$ (i.e., a non-decreasing and continuous function). Suppose every flow set tends to minimize its own cost, $C(f_P) = \Psi_P(f_P) \cdot f_P$, we prove the existence of static Nash equilibrium for this game in Section II-A. The details about how to design the penalty function are discussed in Section II-B.

A. The Existence of Nash Equilibrium

We consider a model for selfish routing where each of an infinite population of agents wants to send an infinitesimal amount of traffic (flows) through a network $G = (V, \mathcal{A})$ with vertex set V, arc set $\mathcal{A}$, and k source-to-destination vertex pairs, $\{s_i, t_i\}, i \in [k] = \{1, \ldots, k\}$ with flow demand r_i. Each agent belongs to one of the $\{s_i, t_i\}, i \in [k]$. Let $\mathcal{P}_i$ denotes the set of multiple equal-cost routing paths from s_i to t_i in G and $\mathcal{P} = \bigcup_i \mathcal{P}_i$, the set of all possible routing paths. The flow set $f_P, P \in \mathcal{P}$ is feasible if for all $i \in [k]$, $\sum_{P \in \mathcal{P}_i} f_P = r_i$. For a given flow set $f_P, P \in \mathcal{P}$, we define the aggregated flows on arc $a \in \mathcal{A}$ as $f_a = \sum_{P \in \mathcal{P}: a \in P} f_P$. The non-sampling rate of a path $P \in \mathcal{P}$ is $\Psi_P(f) = \prod_{a \in P} \Psi_a(f)$ where $\Psi_a(f) = \Psi_a + \zeta(f)$ for each arc $a \in \mathcal{A}$. We are interested in flow assignments that are stable in the sense that no agent can improve their $\Psi_P(f)$ by changing their paths selfishly.

Definition 1: A feasible flow set $f_P, P \in \mathcal{P}$ is at a Wardrop (Nash) equilibrium if for each $i \in [k]$ and every path $P, R \in \mathcal{P}_i$ with $f_P > 0$, it holds that $\Psi_P(f) \leq \Psi_R(f)$.

To prove that the Nash flows always exist in our non-sampling rate case and the achieved cost is unique, we use the Karush-Kuhn-Tucker optimality conditions as in the studies by Beckmann et al. [9] and Dafermos et al. [10]. Let $Q_a(x) = \ln(\Psi_a(x))$ for every arc $a \in \mathcal{A}$ (i.e., also non-decreasing and continuous). Similar to [9] and [10], we con-

struct a convex program (CP) as following with continuously differentiable and convex functions $(h_a)_{a\in\mathcal{A}}$, which is defined as $h_a(f_a) = \int_0^{f_a} Q_a(x)dx$:

$$\texttt{Minimize} \quad \sum_{a\in\mathcal{A}} h_a(f_a) \tag{1}$$

$$s.t. \sum_{P\in\mathcal{P}_i} f_P = r_i \qquad \forall i \in [k] \tag{2}$$

$$f_a = \sum_{P\in\mathcal{P}:a\in P} f_P \qquad \forall a \in \mathcal{A} \tag{3}$$

$$f_P \geq 0 \qquad \forall P \in \mathcal{P} \tag{4}$$

$$h_a(f_a) = \int_0^{f_a} Q_a(x)dx \tag{5}$$

Based on the Karush-Kuhn-Tucker optimality conditions, a feasible flow set $f_P, P \in \mathcal{P}$ is an optimal solution for this convex program if and only if

$$\forall i \in [k], \forall P, R \in \mathcal{P}_i, f_P > 0 \tag{6}$$

$$h'_P(f) = \sum_{a\in\mathcal{P}} h'_a(f_a) \leq \sum_{a\in\mathcal{R}} h'_a(f_a) = h'_R(f), \tag{7}$$

where $h'_a(x)$ refers to the first derivative of $h_a(x)$. Therefore

$$h'_P(f) = \sum_{a\in\mathcal{P}} h'_a(f_a) = \sum_{a\in\mathcal{P}} Q_a(f_a) = \sum_{a\in P} \ln(\Psi_a(f_a)) \tag{8}$$

$$= \ln(\prod_{a\in P} \Psi_a(f_a)) = \ln(\Psi_P(f)) \tag{9}$$

$$\leq \sum_{a\in\mathcal{R}} h'_a(f_a) = \sum_{a\in\mathcal{R}} Q_a(f_a) = \sum_{a\in R} \ln(\Psi_a(f_a)) \tag{10}$$

$$= \ln(\prod_{a\in R} \Psi_a(f_a)) = \ln(\Psi_R(f)) \tag{11}$$

It means $\ln(\Psi_P(f_a)) \leq \ln(\Psi_R(f_a))$, which implies $\Psi_P(f_a) \leq \Psi_R(f_a)$ for $\forall i \in [k], \forall P, R \in \mathcal{P}_i, f_P > 0$. The optimality condition of the convex problem coincides with the condition of the Nash equilibrium.

infocom-mini

B. Design of Penalty Functions

In the routing game, after the current link capacity U_a exceeds U_{max}, we add a sharp penalty to the metric $\Psi_a(f)$ such that selfish agents are aware of the TE constraints. The more U_a exceeds U_{max}, the larger the penalty $\Psi_a(f)$ will be. $U_a = \frac{f}{C_a}$, where C_a is the link capacity and f is the current traffic on link a. Here we use *additive* operator to embed penalty function $\zeta(f)$ into $\Psi_a(f)$, i.e., $\Psi_a(f) = (1 - S_a) + \zeta(f)$. We keep $\zeta(f) = 0$ if $U_a < U_{max}$ and make $\zeta(f)$ increase sharply if $U_a \geq U_{max}$ as follows:

$$\zeta(f) = \begin{cases} 0, & \text{if } U_a < U_{max}; \\ (U_a - U_{max}) \cdot m_\zeta, & \text{if } U_a \geq U_{max}; \end{cases}$$

and therefore

$$\Psi_a(f) = \begin{cases} (1 - S_a) + 0, & \text{if } \frac{f}{C} < U_{max}; \\ (1 - S_a) + (\frac{f}{C} - U_{max}) \cdot m_\zeta, & \text{if } \frac{f}{C} \geq U_{max} \end{cases}$$

where m_ζ controls the sharpness of the penalty. Usually with a larger m_ζ, it will have fewer TE-violations in the equilibrium state but with longer convergence time. We find $m_\zeta = 10^6$ provides a good trade-off between those two effects described above.

III. Distributed MeasuRouting Algorithm

Up to this point, our traffic model is based on the assumption that agents at end hosts have full control over their traffic and they can access the current TE cost value of all paths. Obviously, none of these is true in the real-world IP networks. In this section, we study our Nash equilibrium model that both considers effective non-sampling rate and TE-violation penalty in a dynamic/distributed, round-based variant. Suppose agents at end hosts are activated every T_s seconds and are allowed to change their routes simultaneously. Since they all intend to migrate traffic to a path with minimal cost value, such global migration behavior will result in greatly increased congestion on the optimal path (from measurement's perspective) and lead to oscillations. Fischer et al. proposed the so-called (α-β)-exploration-replication policy in [11] to avoid traffic migration oscillation by using adaptive path-sampling methods. Although [11] is designed for the cost model defined for latency, we apply and modify it to our newly defined non-sampling rate cost model.

In this section, we present our adaptive algorithm, *Distributed MeasuRouting (DisMR)*, which runs on each individual routers to make routing decisions on how to adjust routing split ratios for each destination traffic. In order to do this, each router first needs to measure the non-sampling rate $\Psi(R, V_i)$ for each link to next-hop routers V_i and exchanges information with other routers by using *Distributed Ψ-Propagation Algorithm*. After receiving $A(V_i, D)$, *the expected average non-sampling rate* of the path to every destination D via V_i from next-hop routers, each router can compute $\Psi(R, D, V_i)$ locally and use this information to conduct the *Adaptive Weight Calculations*. $A(V_i, D)$ can be treated as the condensed information of *expected non-sampling rate* beyond V_i. Here we assume synchronized routing-updates of these link/path costs. The impacts of asynchronous update issue could be solved similarly in [5] where we defer as our future work. In summary, each router R needs to maintain the following sets of information for all possible next-hop routers $V_i \in N(R, D)$ to every destination D:

1) $\Psi(R, V_i)$: the non-sampling rate value that also includes the penalty value to reflect the current link utilization on link $R \rightarrow V_i$.
2) $A(V_i, D)$: the expected average non-sampling rate value to destination D via V_i ($V_i \dashrightarrow D$) which is received periodically from neighbor router V_i.
3) $w(R, D, V_i)$: current dynamically changeable weights for traffic routed from current router R to destination D via V_i.

Algorithm 1 describes the distributed Ψ-metric propagation procedure of DisMR in details. Every T_s seconds, the set of $\Psi(R, D, V_i)$ values are updated at each router by using the information of current $\Psi(R, V_i)$ and previous $A(V_i, D)$ from neighbors (line 7) where T_s controls how often the participated routers update their traffic split ratios. Subsequently, the new $A(R, D)$ values are re-calculated by using the current

Algorithm 1 Distributed Ψ-Propagation Algorithm

1: assume current node is R
2: **while** every T_s secs **do**
3: initialize new update message $M(T_s)$
4: **for** each destination D in routing table **do**
5: **for** every next-hop nodes $V_i \in N(R,D)$ **do**
6: measure $\Psi(R,V_i)$
7: $\Psi(R,D,V_i) = \Psi(R,V_i) \cdot A(V_i,D)$
8: **end for**
9: $A(R,D) = \sum_{V_i \in N(R,D)} w(R,D,V_i) \cdot \Psi(R,D,V_i)$
10: Append $A(R,D)$ in $M(T_s)$
11: **end for**
12: Execute one of the **Adaptive-Weights calculations**
13: Send $M(T_s)$ to all neighbor nodes
14: After receiving $M(T_s)$ from neighbor node U_i
15: **for** each $A(U_i,D)$ in $M(T_s)$ **do**
16: **if** $U_i \in N(R,D)$ **then**
17: Update $A(U_i,D)$ from $M(T_s)$
18: **end if**
19: **end for**
20: **end while**

Algorithm 2 Adaptive Weight Calculation

1: after $\Psi(R,D,V_i)$ information is updated
2: **for** each destination D in routing table **do**
3: **for** every next-hop node $V_i \in N(R,D)$ **do**
4: $w_{new}(R,D,V_i) = w(R,D,V_i)$
5: **end for**
6: **for** every pair of next-hop nodes $V_1, V_2 \in N(R,D)$ **do**
7: **if** $\Psi(R,D,V_1) > \Psi(R,D,V_2) + \epsilon \times m_\zeta$ **then**
8: Calculate $P_M = \frac{\Psi(R,D,V_1) - \Psi(R,D,V_2)}{\Psi(R,D,V_1) + \alpha}$
9: **if** with probability P_M **then**
10: **if** $w(R,D,V_2) \neq 0$ **then**
11: $\Delta = (1-\beta) \cdot w(R,D,V_2) \cdot \Delta_{fix}$
12: **else**
13: $\Delta = \frac{\beta}{N(R,D)} \cdot \Delta_{fix}$
14: **end if**
15: $w_{new}(R,D,V_1) = w(R,D,V_1) - \Delta$
16: $w_{new}(R,D,V_2) = w(R,D,V_2) + \Delta$
17: **end if**
18: **end if**
19: **end for**
20: Use $w_{new}(R,D,V_i)$ to distribute the traffic
21: **end for**

weights $w(R,D,V_i)$ and broadcast to all of the neighbor routers (line 9-10). Meanwhile each router will execute the Adaptive Weight Calculation procedure to reassign the weights $w(R,D,V_i)$ for all possible next-hop routers $V_i \in N(R,D)$ to every destination D by using updated information of $\Psi(R,D,V_i)$ (line 12).

Algorithm 2 presents the Adaptive Weight Calculation procedure of DisMR. For every pair of next-hop routers (e.g., say V_1, V_2), it first compares their cost metric $\Psi(R,D,V_i), i = 1,2$ and conducts the migration procedure if the difference of their cost values is more than the *migration threshold* [1] ($\epsilon \times m_\zeta$) (line 7). Otherwise, DisMR will not change the weights of V_1 and V_2.

Subsequently, it computes the migration probability (line 7-9) and the adaptive migration amount (line 10-14) according to the (α-β)-exploration-replication policy [11]. For every pair of next-hop nodes in each round (line 3), we denote V_1 to be the node with larger cost value, $\Psi(.)$ and V_2 to be the alternate node. From statistic point of view, the adaptive migration amount Δ should be calculated depending on node V_2. If V_2 is already used (e.g., $w(R,D,V_2) \neq 0$), then $\Delta = (1-\beta) \cdot w(R,D,V_2) \cdot \Delta_{fix}$ from *proportional sampling* perspective. If V_2 is unused (e.g., $w(R,D,V_2) = 0$), then $\Delta = \frac{\beta}{N(R,D)} \cdot \Delta_{fix}$ from *uniform sampling* perspective where Δ_{fix} is the unit of weight shifted in one round and it controls the convergence speed of DisMR (details are discussed in Section IV-A). The migration probability is decided as $P_M = \frac{\Psi(R,D,V_1) - \Psi(R,D,V_2)}{\Psi(R,D,V_1) + \alpha}$ based on [11] in order to avoid oscillations from global synchronized migrations (line 8). This adaptive migration policy ensures that smaller non-sampling rate gains, $\Delta_\Psi = \Psi_P - \Psi_Q$, only cause a smaller migration possibility and avoid oscillation. The implementation of distributing traffic according to $W(R,D,V_i)$ for each router can use the hashing methods described in [3–5]. If $W(R,D,V_i)$ are constant, there is no packet reordering occurred. However once $W(R,D,V_i)$ are shifted, a fraction of the traffic needs to be rerouted and probably causes packet reordering. The solution is to make the time interval when $W(R,D,V_i)$ shifts occur not smaller than the time TCP needs to recover from packet losses in [3].

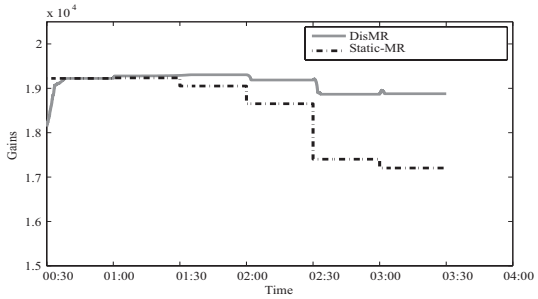
IV. Performance Evaluation

In this section, we evaluate DisMR using Abilene [6](with 11 nodes and 28 links), GEANT [8](with 23 nodes and 74 links) to AS6461 topology obtained using RocketFuel (with 19 nodes and 68 links) [7]. In each set of topology, we first calculate multiple paths for every OD (origin-destination) pair nodes to simulate the (ECMP)-like algorithm in practical scenarios, and run DisMR on those multiple paths. Our simulations have three goals: (1) determine good parameters for the algorithm to quickly reach equilibrium state without significant oscillations; (2) show that the measurement gain of the network at equilibrium state is close to the offline maximum achievable gain calculated by static centralized MeasuRouting; (3) show that it indeed improves measurement gain in dynamic traffic scenario compared to static centralized MeasuRouting.

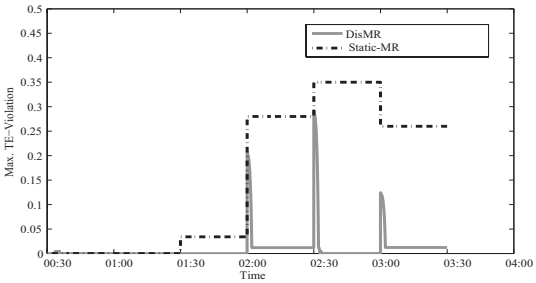
A. DisMR Applied in Realistic Topologies

We first evaluate the performance of DisMR in three realistic topologies: Abilene, GEANT and AS6461 in static traffic scenario. In order to accentuate DisMR's performance, we only consider the traffic traces of the OD pairs with at least two multiple paths. Our simulation results show that our performance is largely independent of α and β. We

[1] $\epsilon \times m_\zeta$ controls the granularity of equilibrium DisMR wants to achieve where m_ζ is the severeness of the penalty and ϵ is the inaccurate-rate DisMR can tolerate.



(a) Measurement Gain Comparison



(b) Max. TE-violation Comparison

Fig. 1. Dynamic traffic scenario

find $\alpha = \beta = 10^{-5}$ offers a good trade-off between the consequences discussed in [11] and we use them for all of our evaluations. About the choice of *migration threshold*, $\epsilon \times m_\zeta$, we observe that the performance of DisMR is also less sensitive to the sharpness of the penalty, m_ζ and we suggest to use $m_\zeta = 10^6$. However with smaller ϵ, DisMR has less TE-violation but with longer convergence time and more oscillations while with larger ϵ, it has more TE-violation but with shorter convergence time and less oscillations. Therefore, choosing the right ϵ is a tradeoff between convergence speed and TE-violations. We suggest using $\epsilon = 10^{-3}$. The more sensitive parameters are mostly *migration rate*, Δ_{fix}. Table I compares the performance of DisMR with different choices of migration rate, Δ_{fix} in Abilene where the fixed migration threshold used in this section is 1000 ($\epsilon = 10^{-3}$, $m_\zeta = 10^6$) and the TE-constraint is $U_{max} = 0.9$. We show that DisMR with smaller Δ_{fix} incurs less TE-violation but with longer convergence time, while DisMR with larger Δ_{fix} incurs more TE-violation but with shorter convergence time. The same property could be observed in both AS6461 and GEANT network topologies. The simulation results with different choices of Δ_{fix} all show that the measurement gain of DisMR is close to the maximum achievable gain using offline, centralized MeasuRouting which is denoted as "Static-MR" in the table but with subtle TE-violations in static traffic scenario.

B. DisMR Applied in Dynamic Traffic Scenario

Here we compare the performance of DisMR with static centralized MeasuRouting in dynamic traffic scenario. We conducted these experiments using GEANT topology with the traffic snapshots on Apirl 11 and we change the traffic patterns in every 30 minutes based on the traces in [8]. Here Static-MR consistently uses the same traffic splitting strategy based on the initial traffic snapshot (00:30), while DisMR will adaptively adjust its traffic splitting policy with the new traffic pattern. Fig. 1 shows the real-time $\max$ TE-violations and the changes of measurement gain for DisMR and Static-MR in GEANT network/trace. Initially, DisMR has similar gain as Static-MR after it reaches equilibrium state (00:38) in Fig. 1(a). We observed that the measurement gain of Static-MR decreases a lot when traffic pattern changed. When the time interval increases (03:30), the degradation becomes severe but DisMR can still outperform Static-MR (e.g., $\frac{1.9-1.7}{1.7} \approx 11.7\%$). In Fig. 1(b), both DisMR and Static-MR have large TE-violation when the traffic suddenly changes but DisMR can quickly improve its TE-violation in short period of time compared to Static-MR (e.g., up to $\frac{0.35}{0.003} \approx 100X$ at time (03:00)). In brief, DisMR has improved higher measurement gains and much lower TE-violations compared to static, centralized MeasuRouting in dynamic traffic scenario.

TABLE I
Δ_{fix} VARIATIONS WITH $m_\zeta = 10^6$, $\epsilon = 0.001$ IN ABILENE

Δ_{fix}	10^{-1}	$5 \cdot 10^{-2}$	10^{-2}	$5 \cdot 10^{-3}$	10^{-3}
iterations	322	440	3416	7965	23068
TE-violation	$3.524 \cdot 10^{-5}$	$1.062 \cdot 10^{-5}$	$9.236 \cdot 10^{-6}$	$1.33 \cdot 10^{-6}$	$1.10 \cdot 10^{-6}$
$Gain(DisMR)$	2671.9	2671.72	2671.57	2671.54	2671.537
$Gain(Static-MR)$	2671.8	2671.8	2671.8	2671.8	2671.8

V. CONCLUSION

In this paper we propose a distributed measurement-aware traffic engineering protocol, DisMR, based on game-theoretic rerouting policy. It achieves the decent balance between measurement-aware routing and traffic engineering objectives by the introduction of a new routing game and distributed routing control. We show that DisMR guarantees both a provable Nash equilibrium and a fast convergence without significant oscillations. The measurement gain of DisMR at the equilibrium state is close to the maximum achievable gain calculated by offline/centralized MeasuRouting in static traffic case. DisMR also improves the measurement gain and TE-violations of MeasuRouting in dynamic traffic scenario.

REFERENCES

[1] C. Estan, K. Keys, D. Moore, and G. Varghese, "Building a Better NetFlow," in *Proceedings of ACM SIGCOMM*, 2004.
[2] S. Raza, G. Huang, C.-N. Chuah, S. Seetharaman, and J. P. Singh, "Measurouting: A framework for routing assisted traffic monitoring," in *IEEE Infocom*, 2010.
[3] N. K. S. Fischer and A. Feldmann, "Replex - dynamic traffic engineering based on wardrop routing policies," in *CoNext'06*, Dec 2006.
[4] B. D. S. Kandula, D. Katabi and A. Charny, "Walking the tightrope: responsive yet stable traffic engineering," in *Proceedings of ACM SIGCOMM*, 2005.
[5] S. L. A. Elwalid, C. Jin and I. Widjaja, "Mate: Mpls adaptive traffic engineering," in *Proc. IEEE INFOCOM Conference*, 2001.
[6] "The abilene network," http://www.internet2.edu.
[7] R. M. N. Spring and T. Anderson, "Quantifying the causes of path inflation," in *Proceedings of ACM SIGCOMM*, 2003.
[8] "Geant topology," http://www.geant.net/Media_Centre/Media_Library/Media\%20Library/10Gfibre2004all.jpg.
[9] C. B. M. M. Beckmann and C. B. Winsten, "Studies in the economics of transportation," Yale University Press 1956.
[10] S. C. Dafermos and F. T. Sparrow, "The traffic assignment problem for a general network," *Journal of Research of the National Bureau of Standards*, vol. 73B, no. 2, pp. 91–118, 1969.
[11] H. R. S. Fischer and B. Vöcking, "Fast convergence to wardrop equilibria by adaptive sampling methods," in *Proc. 38th Ann. ACM. Symp. on Theory of Comput. (STOC)*, May 2006.