This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Reliable Adaptive Multipath Provisioning with Bandwidth and Differential Delay Constraints

Weiyi Zhang*, Jian Tang†, Chonggang Wang‡, and Shanaka de Soysa*
*Department of Computer Science, North Dakota State University, Fargo, ND 58105
†Department of Computer Science, Montana State University, Bozman, MT 59717
‡NEC Laboratories America, Princeton, USA, 08540

***Abstract*— Robustness and reliability are critical issues in network management. To provide resiliency, a popular protection scheme against network failures is the simultaneous routing along *multiple disjoint paths*. Most previous protection and restoration schemes were designed for *all-or-nothing* protection and thus, an overkill for data traffic. In this work, we study the Reliable Adaptive Multipath Provisioning (RAMP) problem with reliability and differential delay constraints. We aim to route the connections in a manner such that link failure does not shut down the *entire stream* but allows a continuing flow for a significant portion of the traffic along multiple (not necessary disjoint) paths, allowing the whole network to carry sufficient traffic even when link/node failure occurs. The flexibility enabled by a multipath scheme has the tradeoff of differential delay among the diversely routed paths. This requires increased memory in the destination node in order to buffer the traffic until the data arrives on all the paths. Increased buffer size will raise the network element cost and could cause buffer overflow and data corruption. Therefore, differential delay between the multiple paths should be bounded by containing the delay of a path in a range. We first prove that RAMP is an NP-hard problem. Then we present a pseudo-polynomial time solution to solve a special case of RAMP, representing edge delays as integers. Next, an $(1+\epsilon)$-approximation algorithm is proposed to solve the optimization version of the RAMP problem. An efficient heuristic is also provided for the RAMP problem. We also present numerical results confirming the advantage of our schemes as the first solution for the RAMP problem.**

***Keywords:* Quality of service; multi-path routing; differential delay; polynomial time approximation; restricted maximum flow.**

I. Introduction

More and more various applications have emerged on the Internet, including IPTV, VoIP, and Video-on-Demand [12], [17], [20], [34], [38]. Driven by these applications, the Internet has also migrated from the best-effort service model to an integrated service model for data, voice, and video applications. As high bandwidth applications become increasingly popular, supporting guaranteed quality-of-service (QoS) requirements becomes an important challenging research problem. Meanwhile, considering the huge amount of data carried on the Internet, there are needs for robustness and reliability in any modern network.

In traditional data networks, routing is commonly along the shortest path [5], [6]. These protocols are inadequate to support multimedia applications such as on-demand video delivery, which require a certain minimum end-to-end bandwidth together with a bounded start-up delay to offer high user satisfaction [9]. Very often, a single path in a bandwidth limited network such as the Internet or a wireless network, may not be able to support such a high bandwidth requirement. As a result, researchers have proposed to use multi-path routing to support high bandwidth multimedia applications in bandwidth limited networks [10], [11], [30]. In many cases, particularly when the network is loaded, splitting the traffic over multiple paths may provide a lower cost solution than finding a single path meeting requested capacity (which likely will be much longer). Meanwhile, protecting a connection over multiple disjoint paths has the obvious advantage of better fault tolerance. On failure, the system switches from one path to the other. Therefore, simultaneous routing along multiple disjoint paths, can result in increased resiliency against such failures. This is especially apparent in the case of real-time data transmission, whereby if one routes along a single path, just one node (or link) failure is sufficient to cause path failure and transmission interruption. In contrast, routing along k disjoint paths makes failure much less likely, as all k disjoint paths must become disconnected in order for transmission to be interrupted. Considerable work has been done on multipath routing in both wired and wireless networks [4], [21], [25]–[27], [31], [32], [39].

We argue that most previous protection and restoration schemes were designed for the *all-or-nothing* protection and thus, an overkill for data traffic. Though the provisioning of two disjoint paths provides better survivability, it imposes at least a 100% protection bandwidth overhead. In [1], the authors argued that while achieving the reliability goals, multiple disjoint path scheme makes very inefficient use of network resource. While voice generates constant bit rate traffic, data traffic is bursty giving the advantage that data applications can continue operation, possibly at a lowered performance, even if the capacity along the path is reduced. For example,

Weiyi Zhang and Shanaka de Soysa are with the Department of Computer Science at North Dakota State University, Fargo, ND 58105. Email: {weiyi.zhang, shanaka.desoysa}@ndsu.edu. Jian Tang is with the Department of Computer Science at Montana State University, Bozman, MT 59717. Email: tang@cs.montana.edu. Chonggang Wang is with NEC Laboratories America, Princeton, USA, 08540. Email: cgwang@nec-labs.com.

The research developed in this paper is supported by North Dakota State EPSCoR under the Infrastructure Improvement Program FAR-0015846 and National Science Foundation under grants CNS-0845776 and CNS-0721880.

978-1-4244-5837-0/10/$26.00 ©2010 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

a wide-area enterprise storage network, while slowing down, can still function if failures reduce the underlying network capacity by 50%. In other words, unlike voice which has a binary service up or down condition, data services can survive gradual degradation as the available bandwidth reduces.

Acharya *et al.* [1] proposed a scheme named PESO which aims to route the connections in a manner such that link (or node failure) does not shut down the entire stream but allows significant amount of traffic to still continue to flow. PESO allows the traffic of a single path to be split and routed along multiple (not necessary disjoint) paths such that a single link failure does not affect more than $x\%$ of the total bandwidth. Thus, PESO creates a novel way to look at the bandwidth overbuild-reliability tradeoff.

Also there are extensive works having been done on how to split, reassemble and re-construct the traffic packets [14], [23]. As studied in [1], [14], [23], the source node partitions the data into several parts using coding theory and transmits each part along a different path. In this way, all packets received from these paths are needed to be intercepted and store in the buffer memory in the destination node in order to find out what is transmitted in the process.

Ability to split and route traffic separately however introduces a unique problem. When traffic is routed over different physical paths, they may incur different amounts of delay and thus, reach at different times. This difference of delay of paths is popularly called *differential delay*. Presence of differential delay requires increased memory in the destination node to buffer the traffic until the data arrives on all the paths [14], [23], and consequently forces increased requirements for memory and bandwidth at the destination node in order to compensate for the transmission. In turn, this potentially raises the network element cost, making deployments more expensive. Moreover, in the worst case, buffer overflows can cause data corruption and bring down the service.

Clearly, it is necessary to handle differential delay in order to correctly re-construct the data at the destination. Differential delay problem was introduced by Srivastava *et al.* in [28]. It motivated the need to study differential delay in routing algorithm for various classes of graph. The authors proposed the initial heuristic based solutions to the problem and studied their performances. [28] also argued the benefit of measuring differential delay values in reverse engineering the accurate link delays. [2], [3] introduced the concept of differential delay minimization when adding a new circuit to an existing group of circuits. It showed the hardness of the problem and proposed heuristics. In [29], the authors studied the cumulative differential delay problem, which seeks to minimize the sum of the difference of delays of all the paths of a solution compared to the highest delay path. The authors of [19] applied the differential delay constraint to telecom mesh networks.

In this work, we will study the Reliable Adaptive Multipath Provisioning (RAMP) problem with bandwidth and differential delay constraints. To the best of our knowledge, this is the first work which jointly studies the adaptive multipath routing and the differential delay problem. In Section II, we will define the RAMP problem and prove that it is NP-hard. In Section III, a pseudo-polynomial time solution is presented for a special case of RAMP, representing edge delays and differential delay bounds as integers. Next, we propose an $(1+\epsilon)$-approximation algorithm to solve the optimization version of the RAMP problem in Section IV, which is followed by an efficient heuristic for RAMP in Section V. We present the numerical results in Section VI, and conclude our work in Section VII.

II. Problem Statement

We model the network using a weighted directed graph $G(V, E, b, d)$, where V is the set of n nodes, and E is the set of m links. Each link $e = (u, v) \in E$ is associated with a *bandwidth* $b(e) > 0$ and a *delay* $d(e) \geq 0$. Let s be a *source* node and t a *destination* node. A $s - t$ path is a sequence of nodes $x_0, x_1, \ldots, x_l$ in V such that $x_0 = s, x_l = t$, and (x_{i-1}, x_i) is a link in E for $i = 1, 2, \ldots, l$. Let p be a $s - t$ path, the *bandwidth of path* p is

$$b(p) = \min_{e \in p} b(e) \tag{2.1}$$

and the *delay of path* p is

$$d(p) = \sum_{e \in p} d(e) \tag{2.2}$$

Definition 1 (Bandwidth allocation)**.** Let $\mathcal{P}$ be a set of $s - t$ paths, where each path $p \in \mathcal{P}$ is associated with a *bandwidth allocation* $\mathcal{L}(p) \leq b(p)$. We say $\mathcal{L}$ is a *feasible bandwidth allocation* of $\mathcal{P}$ if for each link $e \in E$, $\sum_{p \in \mathcal{P}, e \in p} \mathcal{L}(p) \leq b(e)$. The *aggregated bandwidth* of $\mathcal{P}$, denoted by $b(\mathcal{P})$, is the sum of all the bandwidth allocations of the paths in $\mathcal{P}$:

$$b(\mathcal{P}) = \sum_{p \in \mathcal{P}} \mathcal{L}(p) \tag{2.3}$$

Definition 2 (Differential delay)**.** Let $\mathcal{P}$ refers to a set of paths $p_1, p_2, \ldots, p_k$ for a node pair (s, t). The delay of $\mathcal{P}$, denoted by $d(\mathcal{P})$, is defined to be the delay of the longest path in $\mathcal{P}$, i.e., $d(\mathcal{P}) = \max_{p \in \mathcal{P}} d(p)$. Let d_h and d_s be the delay of highest and smallest delay paths in $\mathcal{P}$ respectively, then differential delay $\mathcal{D}_\mathcal{P}$ of paths in defined as: $\mathcal{D}_\mathcal{P} = d_h - d_s$.

Definition 3 (**R**eliable **A**daptive **M**ultipath **P**rovisioning [RAMP])**.** *Let $G = (V, E, b, d)$ be a weighted directed graph with node set V and link set E, where each link $e \in E$ is associated with a bandwidth $b(e) > 0$ and a delay $d(e) \geq 0$. Let $\mathcal{R}$ be a new connection request with source node s, destination node t, bandwidth request B, reliability requirement $x\%$, and differential delay requirements d_{min} and d_{max}. The* Reliable Adaptive Multipath Provisioning (RAMP) *problem seeks a set of paths $\mathcal{P}$ such that:*

1) *The aggregated bandwidth of all paths in $\mathcal{P}$ is no less than B: $b(\mathcal{P}) \geq B$.*
2) *Route the data traffic such that any single link failure does not affect more than $x\%$ of the total bandwidth.*
3) *Any path p in $\mathcal{P}$ must satisfy the differential delay constraint:*

$$d_{min} \leq d(p) \leq d_{max} \tag{2.4}$$

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

In RAMP problem, we aim to serve data service for connection requests adaptive to different reliability requirements. Network operators provide connections with service-level-agreement (SLAs), where an SLA is a contract documenting the availability guarantees. This reflects the case when the traffic is provisioned for the peak rate but operator needs to ensure that the average rate, say 30% below the peak rate, is maintained even after failures. With different reliability requirements, such as *no more than 60% loss of the total traffic in case of at most 2-link failure*, we can adaptively decide the bandwidth allocation on each link used for the traffic and provide reliable traffic. For example, if we allocate no more than $X(= x\% \cdot B)$ on each link (Condition 2 in **Definition 3**), then any single link failure can only affect $x\%$ of the total traffic (no matter how many paths are affected), and any two-link failure (no matter how many paths are affected) can only affect *at most* $2 \cdot x\%$ of the total traffic.

A. Hardness of RAMP

In the following, we will show that the RAMP problem is NP-hard [18], by outlining a reduction from **Partition** to RAMP.

An instance of **Partition** is given by a finite set A, where each $a \in A$ is associated with a positive integer $s(a)$, known as the size of a. It asks for the existence of a subset A' of A such that $\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a)$. This problem is known to be NP-hard [18].

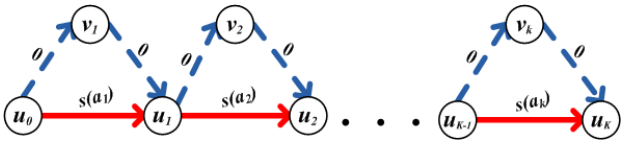


Fig. 1. Reduction from **Partition** to RAMP.

We present a reduction from **Partition** to RAMP in the following, with the help of Fig. 1. Let an instance $\mathcal{I}_1$ of **Partition** be given by $A = \{a_1, a_2, \ldots, a_k\}$ and size function s. We construct an instance $\mathcal{I}_2$ of RAMP in the following way. The set of nodes of graph $G(V, E)$ is given by $V = \{u_0, v_1, u_1, v_2, \ldots, u_{k-1}, v_k, u_k\}$. The set of directed links are (u_{i-1}, u_i), where $i = 1, 2, \ldots, k$, with delay equal to $s(a_i)$ (marked by solid red links in Fig. 1) and (u_{i-1}, v_i) and (v_i, u_i) (dashed blue links in Fig. 1) with a delay equal to 0, $i = 1, 2 \ldots, k$. All links have bandwidth equal to 1. The graph is shown in Fig. 1. Set $s = u_0, d = u_k, B = 2, x\% = 50\%, d_{min} = d_{max} = \frac{\sum_{1 \le i \le k} s(a_i)}{2}$ for the instance of RAMP.

Clearly, $\mathcal{I}_2$ can be constructed from $\mathcal{I}_1$ in polynomial time. Suppose $\mathcal{I}_1$ has a feasible partition $A' \subseteq A$ such that $\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a)$. Then in $\mathcal{I}_2$, from u_0 to u_k, there are two paths p_1 and p_2 where one path, say p_1, uses the solid links corresponding to elements $a \in A'$, and the other path p_2 uses the solid links corresponding to elements $a \in A \setminus A'$. It is easy to see that both paths provide bandwidth 1 and $d_{min} \le d(p_1) = d(p_2) \le d_{max}$. This gives us a feasible solution for $\mathcal{I}_2$.

Conversely, suppose $\mathcal{I}_2$ has a feasible solution, which provides two paths. Each path has a bandwidth of 1, and a path delay of $\frac{\sum_{1 \le i \le k} s(a_i)}{2}$. Let A' be the set of elements whose corresponding solid links are used by p_1. Then A' form a feasible partition for $\mathcal{I}_1$. This proves that RAMP is NP-hard.

Given the hardness of the problem, it is not possible to find a polynomial time optimal solution for RAMP unless $P = NP$ [18]. Therefore, the best solution we can expect are polynomial time approximation schemes or polynomial time approximation algorithms for RAMP. To find an approximation solution for the RAMP problem, let us study a special case of RAMP first, where edge delays and differential delay bounds are integers.

III. Pseudo-Polynomial Solution for IRAMP

In the RAMP problem, the link delay may take nonnegative *real* values. In this section, we study a special case of the RAMP problem, denoted by IRAMP where the delays on the network links and the differential delay bounds (d_{min} and d_{max}) are *integers*. For the special case, we present a novel *pseudo-polynomial time* solution.

Definition 4 (Integer Reliable Adaptive Multipath Provisioning [IRAMP])**.** Let $G = (V, E, b, d)$ be a weighted directed graph with node set V and link set E, where each link $e \in E$ is associated with a bandwidth $b(e) > 0$ and a delay $d(e) \ge 0$. Moreover, on each link e, the edge delay, $d(e)$, is assumed to be **integer**. Let $\mathcal{R}$ be a new connection request with source node s, destination node t, bandwidth request B, reliability requirement $x\%$, and **integer** differential delay requirements d_{min} and d_{max}. The IRAMP problem seeks a set of paths $\mathcal{P}$ such that:

1) The aggregated path bandwidth $b(\mathcal{P})$ is no less than B.
2) Route the requested connection such that any single link failure does not affect more than some $x\%$ of the total bandwidth.
3) Each path in $\mathcal{P}$ must satisfy the differential delay constraint. In other words, for any path p in $\mathcal{P}$, its delay is bounded by d_{min} and d_{max}.

Our algorithm is based on a novel graph transformation technique, which is described in below.

A. Graph Transformation

Let an instance of IRAMP be given by graph $G(V, E)$, reliable requirement $x\%$, bandwidth request B, differential delay bounds d_{min} and d_{max}, and source-destination node pair (s, t). We construct a *layered graph* $G^R = (V^R, E^R)$ from G in the following way.

1) Corresponding to each node $u \in V$, V^R contains $(d_{max} + 1)$ nodes $u_{[0]}, u_{[1]}, \ldots, u_{[d_{max}]}$.
2) For each link (u, v) in G, E^R contains $d_{max} - d(u, v) + 1$ links in the form $(u_{[i]}, v_{[i+d(u,v)]}), i = 0, 1, \ldots, (d_{max} - d(u, v))$.
3) E^R also contains $(d_{max} - d_{min})$ links in the form of $(t_{[i]}, t_{[i+1]}), i = d_{min}, \ldots, d_{max} - 1$. Each such a link has bandwidth ∞.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

4) For all edges $(u_{[i]}, v_{[i+d(u,v)]})$ constructed in E^R in step 2), if $b(u,v) \geq x\% \cdot B$, aggregated bandwidth of these edges is $x\% \cdot B$. Otherwise, aggregated bandwidth is $b(u,v)$ for these edges.

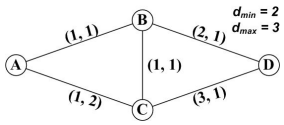


Fig. 2. Graph showing $(b(e), d(e))$ on each link e.

Each link $(u_{[i]}, v_{[i+d(u,v)]})$ in G^R demonstrates the case that if at node u a path has delay i, and the path is extended by using link (u,v), then the path delay will be increased to $i + d(u,v)$ when reaching node v. Let us use an example to demonstrate our graph transformation technique. In Fig. 2, an instance of IRAMP with $s = A, t = D, B = 2, x\% = 50\%, d_{min} = 2$ and $d_{max} = 3$ is presented. The corresponding layered graph G^R is represented in Fig. 3.

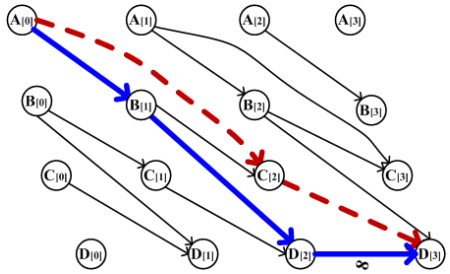


Fig. 3. Graph G^R corresponding to the graph G in Fig. 2.

Since the reliability requirement is 50%, and the bandwidth request is 2, the bandwidth that can be allocated on each link in G is at most 1. Correspondingly, in Fig. 3, all links in the form $(u_{[i]}, v_{[i+d(u,v)]}), i = 0, 1, \ldots, (d_{max} - d(u,v))$ have an aggregated bandwidth of 1. For example, link (B, D) has bandwidth 2 in Fig. 2, but only at most bandwidth 1 can be used for connection $A - D$. Thus, its corresponding links in G^R, $(B_{[0]}, D_{[1]}), (B_{[1]}, D_{[2]}), (B_{[2]}, D_{[3]})$, have aggregated bandwidth 1, shown in Fig. 3.

Algorithm 1 IRAMP$(G, s, t, B, x, d_{min}, d_{max})$

1: Construct corresponding G^R using above graph transformation technique;
2: Find a maximum restricted flow from $s_{[0]}$ to $t_{[d_{max}]}$;
3: **if** flow value is less than B **then**
4: Drop the connection request;
5: **else**
6: Decompose the flow to generate a set of path $\mathcal{P}^R$;
7: Find the corresponding path set $\mathcal{P}$ in G;
8: Output $\mathcal{P}$ as the solution.
9: **end if**

It is worth noting that the constructed graph G^R is acyclic since for each link $(u_{[i]}, v_{[j]}) \in E^R$, it is always true that $i < j$. To find a solution for an instance of IRAMP from s to t in G, we search paths in G^R from $s_{[0]}$ to $t_{[d_{max}]}$. Note that any path p^R from $s_{[0]}$ to $t_{[d_{max}]}$ guarantees that its delay is between d_{min} and d_{max}. For any p^R, there is a corresponding path p from s to t in G whose delay is also bounded by d_{min} and d_{max}. Since the aggregated flow on links $(u_{[i]}, v_{[i+d(u,v)]}), i = 0, \ldots, d_{max} - d(u,v)$ is no more than $x\% \cdot B$, link (u,v) has no more than $x\% \cdot B$ flow value on it in the corresponding $s - t$ flow in G.

With the maximum flow (whose value $\geq B$) calculated by IRAMP, we can use arc-chain decomposition [15], [16] to generate paths, each of whose delay is bounded by d_{min} and d_{max} and bandwidth is no more than $x\% \cdot B$. For our example given in Fig. 2, Algorithm 1 will find a flow which is illustrated in Fig. 3 with thick links. It will be decomposed to two paths (Line 6) $p_1^R = (A_{[0]}, B_{[1]}, D_{[2]}, D_{[3]})$ (marked by solid blue links) and $p_2^R = (A_{[0]}, C_{[2]}, D_{[3]})$ (marked by dashed red links), in the layered graph G^R. The blue path p_1^R has corresponding path $p_1 = (A, B, D)$ in G with path delay 2. The red path p_2^R has corresponding path $p_2 = (A, C, D)$ in G with path delay 1. Note that path p_1 has path delay smaller than d_{max}, due to the links $(D_{[2]}, D_{[3]})$.

B. Restricted Maximum Flow Scheme for IRAMP

A critical part of our solution for IRAMP is in Line 2 of Algorithm 1. To find the paths, we need to find a restricted maximum flow in G^R. The restriction of our flow solution is the *aggregated bandwidth* of the corresponding links of a link in G. To solve the *restricted maximum network flow* problem in the layered graph G^R, we propose a Linear Program (LP) solution as listed below.

Objective: **Maximize** F
Subject to:

$$\sum_{u \in adj(s_{[0]})} f(s_{[0]}, u) - \sum_{u \in adj(s_{[0]})} f(u, s_{[0]}) = F \quad (3.1)$$

$$\sum_{u \in adj(t_{d_{max}})} f(u, t_{d_{max}}) - \sum_{u \in adj(t_{d_{max}})} f(t_{d_{max}}, u) = F \quad (3.2)$$

$$\sum_{(x,y) \in E^R} f(x,y) = \sum_{(y,x) \in E^R} f(y,x), \quad \forall y \neq s_{[0]}, t_{d_{max}} \quad (3.3)$$

$$\sum_{i=0}^{d_{max}-d(u,v)} f(u_{[i]}, v_{[i+d(u,v)]}) \leq b(u,v), \quad \forall (u,v) \in E \quad (3.4)$$

$$\sum_{i=0}^{d_{max}-d(u,v)} f(u_{[i]}, v_{[i+d(u,v)]}) \leq x\% \cdot B, \quad \forall (u,v) \in E \quad (3.5)$$

$$f(x,y) \geq 0, \ \forall (x,y) \in E^R \quad (3.6)$$

In our restricted maximum flow formulation, the objective is to find a maximum flow in G^R and the variables are the flow values (real valued) on the links in G^R. These variables are nonnegative (ensured by Constraint 3.6). Constraints (3.1), (3.2), and (3.3) ensure the flow conservation in the network. Constraints (3.4) and (3.5) ensure that for each link (u,v) in G, the aggregated flow on all links $(u_{[i]}, v_{[i+d(u,v)]}) \in E^R$ is

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

no more than $x\%$ of the total bandwidth request (the reliability requirement), and no more than $b(u,v)$, the actual link bandwidth. These constraints make *restricted maximum flow* different from the conventional network flow problem [16]. It is necessary because all the flows on links $(u_{[i]}, v_{[i+d(u,v)]}) \in E^R$ corresponding to an aggregated flow on link (u,v) in G. And any link (u,v) cannot carry any traffic exceeding its capacity, or more than $x\%$ of the total traffic due to the reliability requirement.

It is worth noting that our linear programming problem has $O(m \cdot d_{max})$ variables. Meanwhile, assume that $\mathcal{L}$ is the input size of the instance of IRAMP$(G, s, t, B, x, d_{min}, d_{max})$. the input size of our linear programming is of d_{max} times of the input size of IRAMP$(G, s, t, B, x, d_{min}, d_{max})$, say $O(d_{max}\mathcal{L})$. Based on the theory in [37], it takes $O((m \cdot d_{max})^3(d_{max}\mathcal{L}))$ time to solve the LP formulation, This shows that our solution is a polynomial time scheme.

IV. Approximation Scheme for ORAMP

The RAMP problem studied in Secton II and III is a decision problem [13]. In this section we first introduce its corresponding optimization problem, denoted as ORAMP. Then we will study a special case of the ORAMP problem (denoted as SPRAMP), and present an $(1+\epsilon)$-approximation scheme for SPRAMP. Finally, we propose an $(1+\epsilon)$-approximation algorithm for the ORAMP problem based on the approximation scheme for SPRAMP.

Definition 5 (ORAMP)**.** Let $G = (V, E, b, d)$ be a weighted directed graph with node set V and link set E, where each link $e \in E$ is associated with a bandwidth $b(e) > 0$ and a delay $d(e) \geq 0$. Let $\mathcal{R}$ be a new connection request with source node s, destination node t, bandwidth request B, reliability requirement $x\%$, and delay requirements d_{min}. The *optimization version* of the RAMP problem ORAMP seeks a set of $s-t$ paths, $\mathcal{P}$, together with a feasible bandwidth allocation $\mathcal{L}$ such that:
Objective: **Minimize** D
Subject to:

$$(1)\ b(\mathcal{P}) \geq B; \quad (2)\ d_{min} \leq d(p) \leq D, \quad \forall p \in \mathcal{P}$$

$$(3)\ \sum_{p \in \mathcal{P}, e \in p} \mathcal{L}(p) \leq \min\{b(e), x\% \cdot B\}, \forall e \in E;$$

We proved that RAMP is NP-hard in Section II. It also implies that ORAMP is NP-hard. Thus, the best we can do is to present an approximation scheme or algorithm for the ORAMP problem. We first study a *special case* of ORAMP (denoted by SPRAMP), where $d_{min} = 0$. A Fully Polynomial Time Approximation Scheme (FPTAS) is presented for this spacial case. Then we provide an $(1+\epsilon)$-approximation algorithm for the ORAMP problem based on the FPTAS for the SPRAMP problem. Before we introduce our FPTAS for the SPRAMP problem, we first give our observation on the relation between ORAMP and SPRAMP.

Lemma 1. If there is no feasible solution for an instance of SPRAMP(G, s, t, x, B), then there is no feasible solution for the instance ORAMP(G, s, t, x, B, d_{min}). □

Proof. Let us prove the Lemma by contradiction. Assume that SPRAMP(G, s, t, x, B) does not have a feasible solution while corresponding ORAMP(G, s, t, x, B, d_{min}) has a feasible solution path set $\mathcal{P}$. Therefore, we have $d_{min} \leq d(p) \leq d(\mathcal{P})$ for each path $p \in \mathcal{P}$ and $b(\mathcal{P}) \geq B$ with each link having no more than $x\%$ of the total bandwidth on it. It is easy to see that this path set $\mathcal{P}$ is also a feasible solution for SPRAMP(G, s, t, x, B) because of $0 \leq d(p) \leq d(\mathcal{P})$ for each path $p \in \mathcal{P}$. This will contradict our assumption. Therefore, if an instance of SPRAMP(G, s, t, x, B) has no feasible solution, an instance of ORAMP(G, s, t, x, B, d_{min}) must have no feasible solution too. ■

A. A Fully Polynomial Time Approximation Scheme for SPRAMP

For a given positive real number θ and an instance SPRAMP(G, s, t, x, B), we construct an auxiliary graph $G^\theta = (V, E, b, d^\theta)$ of graph $G = (V, E, b, d)$. The edge delay is changed as $d^\theta = \lfloor d(e) \cdot \theta \rfloor + 1$ for each edge e. This is the scaling and rounding technique used in [22], [36].

Algorithm 2 FPTAS-SPRAMP(G, s, t, x, B)

1: Construct $G^\theta = (V, E, s, d^\theta)$ by changing the edge delay $d^\theta = \lfloor d(e) \cdot \theta \rfloor + 1$ on each link e;
2: Find the *minimum integer* D, say D_m, such that IRAMP$(G^\theta, s, t, x, 0, D)$ is feasible;
3: Solve IRAMP$(G^\theta, s, t, x, 0, D_m)$ and find the corresponding path set for the SPRAMP problem.

In Algorithm 2, we first update the delay of each link in G^θ. After the change, each link in G^θ has an integer delay. Next, we try to find the minimum integer D which can return feasible solution for an instance of IRAMP with $d_{min} = 0$ and $d_{max} = D$. Denote the minimum value of D as D_m, we try to solve an IRAMP instance with edge delay d^θ, $d_{min} = 0$ and $d_{max} = D_m$. For the returned path set $\mathcal{P}$, we have the following theorem.

Theorem 1. The solution found by Algorithm 2 is an $(1+\epsilon)$-approximation to SPRAMP(G, s, t, x, B). □

Proof. Assume that the *optimal solution* for SPRAMP(G, s, t, x, B) is a set of paths denoted by $\mathcal{Q}$. And the optimal value is d_o. So we know that $0 \leq d(p) \leq d_o, \forall p \in \mathcal{P}$. Let L and U be given positive constants such that $L \leq d_o \leq U$. For any given $\epsilon > 0$, we calculate $\theta = \frac{n-1}{L \cdot \epsilon}$.

For any path p in the optimal solution $\mathcal{Q}$, its delay value in G^θ is denoted as $d^\theta(p)$. We have

$$\begin{aligned} d^\theta(p) &= \sum_{e \in p} d^\theta(e) = \sum_{e \in p} (\lfloor \theta \cdot d(e) \rfloor + 1) \qquad (4.1) \\ &\leq \sum_{e \in p} (\theta \cdot d(e) + 1) \leq \theta \cdot d(p) + (n-1) \end{aligned}$$

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Since d_o is the optimal value for SPRAMP, and p is a path from the optimal solution to SPRAMP, it is easy to see that $0 \leq d(p) \leq d_o$. In other words, every path in the optimal solution has its delay no less than 0 and no more than d_o. Combining with (4.1), we know that for each path $p \in \mathcal{Q}$:

$$\begin{aligned} d^\theta(p) &\leq \theta \cdot d_o + n - 1 \\ &= \frac{(n-1) \cdot d_o}{L \cdot \epsilon} + n - 1 \end{aligned} \tag{4.2}$$

Because $d^\theta(p)$ can only take integer values, it implies that

$$d^\theta(p) \leq \lfloor \frac{(n-1) \cdot d_o}{L \cdot \epsilon} \rfloor + n - 1, \quad \forall p \in \mathcal{Q} \tag{4.3}$$

This proves that the optimal solution $\mathcal{Q}$ for SPRAMP(G, s, t, x, B) is also a *feasible solution* for IRAMP$(G^\theta, s, t, x, B, 0, \lfloor \frac{(n-1) \cdot d_o}{L \cdot \epsilon} \rfloor + n - 1)$.

In Line 2 in Algorithm 2, D_m is found as the *minimum value* of D which guarantees that IRAMP$(G^\theta, s, t, x, B, 0, D)$ has a feasible solution. Since IRAMP$(G^\theta, s, t, x, B, 0, \lfloor \frac{(n-1) \cdot d_o}{L \cdot \epsilon} \rfloor + n - 1)$ has a feasible solution $\mathcal{Q}$, it is clear that $D_m \leq \lfloor \frac{(n-1) \cdot d_o}{L \cdot \epsilon} \rfloor + (n-1)$. This implies that

$$\begin{aligned} D_m &\leq \lfloor \frac{(n-1) \cdot d_o}{L \cdot \epsilon} \rfloor + (n-1) \\ &= \theta \cdot d_o + L \cdot \epsilon \cdot \theta \end{aligned} \tag{4.4}$$

In Line 3 of Algorithm 2, we solve IRAMP$(G^\theta, s, t, x, 0, D_m)$. Let us assume that the returned feasible solution is $\mathcal{P}^\theta$. For each $p_\theta \in \mathcal{P}^\theta$, we have

$$d^\theta(p_\theta) \leq D_m \leq \theta \cdot d_o + L \cdot \epsilon \cdot \theta \tag{4.5}$$

Meanwhile, we know that:

$$\begin{aligned} d^\theta(p_\theta) &= \sum_{e \in p_\theta} (\lfloor \theta \cdot d(e) \rfloor + 1) \\ &\geq \sum_{e \in p_\theta} \theta \cdot d(e) \\ &\geq \theta \cdot d(p_\theta) \geq 0 \end{aligned} \tag{4.6}$$

Combine (4.6) with (4.5), it is clear that

$$\begin{aligned} &\theta \cdot d(p_\theta) \leq d^\theta(p_\theta) \leq D_m \leq \theta \cdot d_o + L \cdot \epsilon \cdot \theta \\ &\Rightarrow d(p_\theta) \leq d_o + L \cdot \epsilon \leq (1+\epsilon) \cdot d_o, \quad \forall p_\theta \in \mathcal{P}^\theta \end{aligned} \tag{4.7}$$

Meanwhile, it is obvious that the reliability requirement will be satisfied for a SPRAMP instance if it is satisfied for the corresponding IRAMP instance. This proves that $\mathcal{P}^\theta$ returned by IRAMP$(G^\theta, s, t, x, 0, D_m)$ is an $(1+\epsilon)$-approximation to SPRAMP(G, s, t, x, B). ■

B. Discussion

The most important part of the FPTAS to SPRAMP is *how to find* D_m in Line 2 of Algorithm 2. In equation (4.4), we proved that $D_m \leq \lfloor \frac{(n-1) \cdot d_o}{L \cdot \epsilon} \rfloor + (n-1)$. Therefore, we know that:

$$0 \leq D_m \leq \lfloor \frac{(n-1) \cdot U}{L \cdot \epsilon} \rfloor + (n-1)$$

Using binary search, we can find D_m by testing $\log(\frac{n \cdot U}{\epsilon \cdot L})$ instances of IRAMP. Solving each instance takes $O((m \cdot \frac{n \cdot U}{\epsilon \cdot L})^3 (\frac{n \cdot U}{\epsilon \cdot L} \mathcal{L}))$ time (showed in Section III-B).

It is clear that to calculate D_m, we first need to find a pair of lower and upper bounds of d_o, denoted by L and U. Then we can find D_m and an $(1+\epsilon)$-approximation solution for SPRAMP(G, s, t, x, B) in polynomial time.

Now we need to find an efficient way to compute a lower bound L and a upper bound U. Assume that there are m links in the network, the number of distinct link delay values is no more than m. Let these distinct link delay values be $d_1 < d_2 < \ldots < d_k$. Clearly, for any simple path in G, its delay will be no less than d_1 and no more than $(n-1) \cdot d_k$. However, it is a very loose bounds for d_o. And we can see that the running time of our FPTAS is proportional to $(\frac{U}{L})^4$ times. Therefore, it is important to compute a tight pair of lower and upper bounds.

To do that, we adopt the *scaling and testing* technique used in [22], [24], [36]. First we try to find the initial pair of L and U such that the ratio between them is n, the number of nodes in the network. Define E_i to be the set of links with delay not greater than $d_i, 1 \leq i \leq k$, that is $E_i = \{e \in E | d(e) \leq d_i\}, 1 \leq i \leq k$, and $E_0 = \phi$. Let $G_i = (V, E_i)$, then $G_k = G$ and $G_i \subset G_{i+1}, 0 \leq i \leq k-1$. If there is no $s-t$ flow of B in G_k, then the instance has no solution. Otherwise, there must exist a unique index $l (1 \leq l \leq k)$ such that G_l has an flow of value B, and G_{l-1} has no such flow. Then in [22], [24], it is proved that $d_l \leq d_o \leq (n-1) \cdot d_l$. Next, using $L = d_l$ and $U = (n-1) \cdot d_l$, we use the *approximation testing* scheme [22], [36] to refine the pair of lower and upper bounds until U is within a *constant factor* of L [35], [36]. It takes $O(m^3 n^4 \mathcal{L} \log\log n)$ to find such pair of lower and upper bounds [35], [36]. Then we need to calculate D_m (Line 2 in Algorithm 2) using binary search (D_m *is an integer*), D_m can be computed by solving $O(\log(\frac{n}{\epsilon}))$ instances of IRAMP, each requiring $O(m^3 (\frac{n}{\epsilon})^4 \mathcal{L})$ time since $\frac{U}{L}$ is a constant. Finally, Line 3 of Algorithm 2 takes $O(m^3 (\frac{n}{\epsilon})^4 \mathcal{L})$ time to find an approximation solution. Therefore, the running time of Algorithm 2 is $O(m^3 (\frac{n}{\epsilon})^4 \mathcal{L} \log(\frac{n}{\epsilon}))$, which is polynomial.

C. Approximation Algorithm for ORAMP

In Section IV-A, we presented an $(1+\epsilon)$-approximation scheme to SPRAMP, a special ORAMP problem with $d_{min} = 0$. In this section, we will propose an $(1+\epsilon)$-approximation algorithm to solve the ORAMP problem with $d_{min} \geq 0$.

Our approximation algorithm is based on the FPTAS for SPRAMP, and is listed in Algorithm 3. Given an instance of ORAMP(G, s, t, x, B, d_{min}), we first solve an instance of SPRAMP(G, s, t, x, B) (Line 1). This is like a *necessary condition* check for the ORAMP problem. As we proved in **Lemma** 1, we can drop an instance of ORAMP if its corresponding SPRAMP instance is not not feasible. Then if there is a feasible solution $\mathcal{P}$ for SPRAMP(G, s, t, x, B), we check every path p in $\mathcal{P}$ (Lines 6-11). If p does not satisfy the differential delay constraint, we remove this path

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Algorithm 3 Approximation to ORAMP(G, s, t, x, B, d_{min})

1: Solve SPRAMP(G, s, t, x, B);
2: **if** (**NO** feasible solution is found) **then**
3: Drop the connection request.
4: **end if**
5: Assume the found path set is $\mathcal{P}$;
6: **for** each path $p \in \mathcal{P}$ **do**
7: **if** $d(p) < d_{min}$ **then**
8: Remove p from $\mathcal{P}$;
9: $b(\mathcal{P}) = b(\mathcal{P}) - b(p)$;
10: **end if**
11: **end for**
12: **if** $b(\mathcal{P}) \geq B$ **then**
13: Output $\mathcal{P}$ as the solution.
14: **else**
15: Drop the request.
16: **end if**

from $\mathcal{P}$, and reduce the aggregated path bandwidth (Line 9). If all the remaining paths, which all satisfy the differential delay constraint, can still provide enough bandwidth (no less than B), we return $\mathcal{P}$ as the solution. Otherwise, we drop the request.

Theorem 2. The path set $\mathcal{P}$ found by Algorithm 3 is an $(1+\epsilon)$-approximation solution for ORAMP(G, s, t, x, B, d_{min}). □

Proof. It is obvious that reliability requirement will be satisfied for the returned paths. Meanwhile, it is ensured that each path has its delay no less than d_{min} (**for**-loop from Line 6 to Line 9). Assume that the optimal value of the instance ORAMP(G, s, t, x, B, d_{min}) is d_o. What we need to prove is that $d(p) \leq (1+\epsilon) \cdot d_o, \forall p \in \mathcal{P}$.

Since $\mathcal{P}$ is a feasible solution to SPRAMP, it has been proven in Theorem 1 that

$$0 \leq d(p) \leq (1+\epsilon) \cdot d_o^S, \forall p \in \mathcal{P} \tag{4.8}$$

where d_o^S is the optimal value of the instance SPRAMP(G, s, t, x, B).

From **Lemma** 1, we know that any feasible solution for ORAMP(G, s, t, x, B, d_{min}) is a feasible solution for SPRAMP(G, s, t, x, B). Therefore, $d_o \geq d_o^S$. Otherwise (if $d_o < d_o^S$), d_o^S cannot be the optimal value of SPRAMP(G, s, t, x, B, d_{min}). Then it is easy to see that $d(p) \leq (1+\epsilon) \cdot d_o, \forall p \in \mathcal{P}'$. Therefore, this proves that the solution found by Algorithm 3 is an $(1+\epsilon)$-approximation to the ORAMP problem. ■

V. Efficient Heuristic for RAMP

We have presented an FPTAS for SPRAMP and an approximation algorithm for ORAMP. Though both are polynomial time solutions, they still can be time consuming if the given network size is large, as shown in the following simulation results. In practice, it is still necessary to provide fast and efficient heuristics for the RAMP problem.

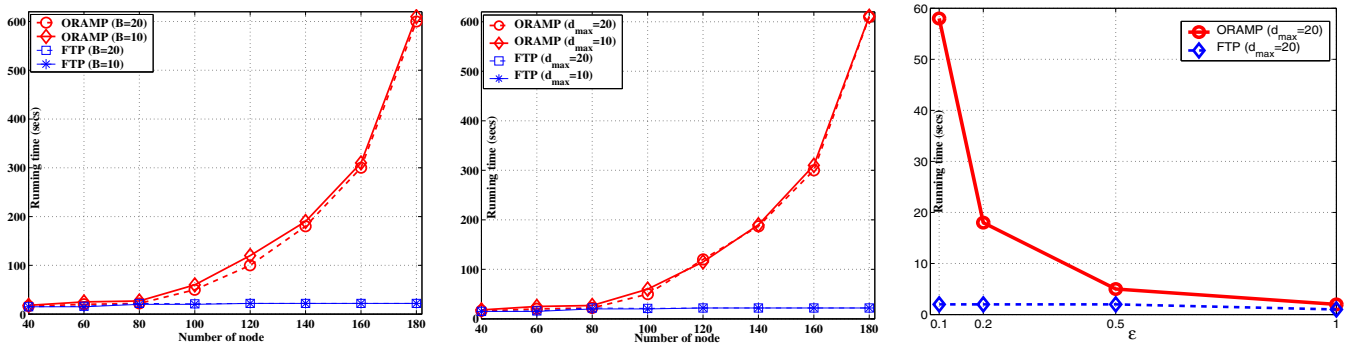
Algorithm 4 *Flow_to_Path(G, s, t, x, B,* d_{min}, d_{max}*)*

1: **for** each edge $e \in G$ **do**
2: **if** $b(e) > x\% \cdot B$ **then**
3: $b(e) = x\% \cdot B$;
4: **end if**
5: **end for**
6: Compute a maximum flow from s to t.
7: **if** the found maximum flow value is less than B **then**
8: Drop the connection request;
9: **else**
10: The returned flow and the links with positive flow form a subgraph G';
11: *TotalFlow* = 0;
12: Calculate a path set $\mathcal{P}$ having K paths, $p_1, p_2, \ldots, p_K$ in the non-decreasing order of delays;
13: **for** $i = 1, 2, \ldots, K$ **do**
14: **if** $d_{min} \leq d(p_i) \leq d_{max}$ **then**
15: *TotalFlow* = *TotalFlow* + $b(p_i)$;
16: **else**
17: Remove p_i from $\mathcal{P}$;
18: **end if**
19: **end for**
20: **if** *TotalFlow* $\geq B$ **then**
21: Output $\mathcal{P}$ as the feasible solution.
22: **else**
23: Drop the connection request.
24: **end if**
25: **end if**

Our *flow to path* heuristic is listed in Algorithm 4. First we update the link bandwidth according to the reliability requirement (any link failure will not affect $x\%$ of the total traffic bandwidth) from Line 1 to Line 5. Then we compute a maximum flow from s to t. If the value is smaller than B, the heuristic correctly claims that the given instance of RAMP has no solution. Otherwise, the links in the maximum flow form a subgraph G' of G. And we know that an $s-t$ flow satisfying the reliability and bandwidth requirement exists in G'. Next, we find a number of candidate paths (no more than a predefined number K) in G' and sort them in the non-decreasing order of path delays. Then we check the feasibility of the paths one by one. If a path does not satisfy the differential delay constraint, we remove it from the path set. Finally if all the remaining paths can provide enough bandwidth, we output a feasible solution. Otherwise, we drop the connection request.

VI. Numerical Results

In this section, we present numerical results to confirm the effectiveness of our solutions. We implemented the approximation algorithm for ORAMP, which is denoted as ORAMP in the figures, and our *flow to path* heuristic, which is denoted as FTP. We compare the results of ORAMP with those obtained from FTP. All our simulation runs were performed on a 2.8 GHz Linux PC with 1G bytes of memory.



(a) Running times with different B, $x = 30\%$, $\epsilon = 0.5, d_{max} = 10$.

(b) Running times with different d_{max}, $x = 30\%$, $\epsilon = 0.5, B = 10$.

(c) Running times with different ϵ, $x = 30\%$, $B = 10, d_{max} = 10$.

Fig. 4. Running times of ORAMP and FTP

We used well-known Internet topologies to *verify the suitability of the algorithms*, and randomly generated topologies to *verify the computational scalability of the algorithms*. The well-known Internet topologies used are ItalianNet (which has 33 nodes and 67 links), ArpaNet (which has 20 nodes and 32 links), and NSFNET (which has 14 nodes and 21 links) [35], [36]. Each link in the graph has two weights, the first represents the bandwidth, and the second represents the delay. Similar to [35], [36], the link bandwidth and delay values were generated uniformly in a range (we use the interval $[1, 10]$). For the random graphs, we used the widely used BRITE graph generator [8]. We used the default values of $\alpha = 0.15$ and $\beta = 0.2$ for the Waxman model [33] in BRITE. For the setup, the nodes were randomly generated in a square field of size 1000×1000 square meters. According to the Waxman model, if E_{uv} denotes the Euclidean distance between two nodes u and v, the probability of having a bidirectional link (u, v) connecting u and v is given by $\beta \times e^{\frac{E_{uv}}{\alpha M}}$, where e is the base for natural logarithms, M is the maximum distance between any two nodes. We have used 8 different network sizes, where the number of nodes are 40, 60, 80, 100, 120, 140, 160, and 180, respectively. For each case, the source and destination pairs were chosen randomly for both algorithms.

To study the scalability of our algorithms with the network size, we first tested both algorithms on large, randomly generated network topologies. For each network topology and a given value of ϵ, we generated 10 source-destination pairs for testing. Here we have used reliability requirement $x = 30\%$ for all the test cases. For FTP, the number of precomputed paths, K, is defined as 10. In other words, we calculated 10 paths (Line 12 of Algorithm 4) for FTP testing if possible.

First, Fig. 4(a) illustrates the running times with different bandwidth request B. With $\epsilon = 0.5, d_{max} = 10$, we tested of both ORAMP and FTP with bandwidth request $B = 20$ and $B = 10$. It can be expected that our algorithms performed similarly on various bandwidth requests because the running time is independent of the value of B. We also observed that the running times of ORAMP increased faster than the running times of FTP did with the increase of the the network size. Next, in Fig. 4(b), with $x = 30\%, \epsilon = 0.5, B = 10$, we compared the running times of both algorithms with different d_{max} on the randomly generated networks. As shown in our analysis, the running time of ORAMP is independent of d_{max}. For both Fig. 4(a) and Fig. 4(b), ORAMP required longer running time than FTP. However, the running time of ORAMP grew polynomially with the network size. We do not show the running time for the well-known network topologies because both ORAMP and FTP took little time to find the solutions.

Fig. 4(c) illustrates the running times of both algorithms as functions of ϵ, using the case of $x = 30\%, B = 10, d_{max} = 20$ for ItalianNet. Clearly, ϵ has no effect on FTP. Meanwhile, as expected, the running times of ORAMP increase with $1/\epsilon$.

Another important performance metric is the *delay of the set of paths*. To evaluate the performance in term of delay of path, we introduced a new definition of *delay ratio*, which is defined as $\frac{d(\mathcal{P})}{d_{max}}$. If there is no feasible solution, then this delay ratio is defined as ∞ (denoted as **inf** in Fig. 5). It is clear that *the lower the delay ratio, the better the performance*. It is worth noting that if the ratio is no more than 1, then we have a feasible solution for RAMP.

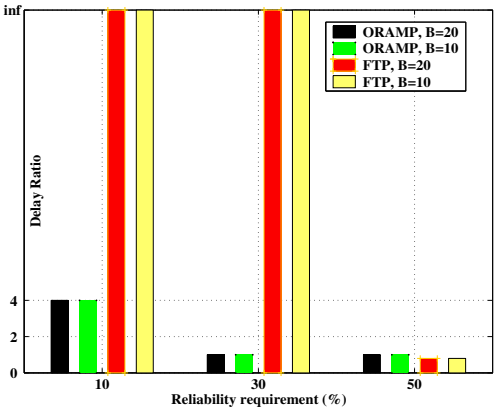


Fig. 5. Delay ratio for ItalianNet.

Fig. 5 shows the delay ratio values obtained by ORAMP and FTP for the ItalianNet for $B = 20$ and $B = 10$. With different reliability requirements ($x\% = 10\%, 30\%, 50\%$), The delay ratio values of the solutions found by ORAMP are mostly less than the delay ratio values of the solutions found by FTP.

When $x\% = 10\%$ or 20%, FTP could not find any feasible solution for RAMP. So the delay ratio is ∞, marked as **inf** in Fig. 5. On the other hand, ORAMP returned approximation or feasible solutions for all cases. Note that we have tested for ArpaNet and NSFNET. The delay ratio results for these network topologies, which are not shown here due to the space limit, follow the similar pattern.

VII. Conclusions

In this work, we studied the Reliable Adaptive Multipath Provisioning (RAMP) problem, which seeks to provide connection with predefined reliability requirement and differential delay constraint. Adapting to different reliability requirements, we can define the bandwidth allocation on the links in the network to guarantee the required network reliability. Differential delay is also an important research issue in multipath routing. This paper is the first work which jointly studies the adaptive multipath routing and the differential delay problem. We first proved that the RAMP problem is NP-hard. Then we proposed a novel pseudo-polynomial time algorithm for a special case of RAMP where edge delays and differential delay bounds are integers. Next, we presented a fully polynomial time approximation scheme for the SPRAMP problem and an $(1+\epsilon)$-approximation algorithm for the ORAMP problem. Since the problem is NP-hard, our approximation scheme and algorithm are the best one can hope for. We also provide an efficient heuristic for the RAMP problem. Numerical results confirm the advantage of our solutions.

References

[1] S. Acharya, B. Gupta, P. Risbood, A. Srivastava, PESO: Low Overhead Protection for Ethernet over SONET Transport, *IEEE Infocom'2004*, pp.165-175.

[2] S. Ahuja, T. Korkmaz, M. Krunz, Minimizing the differential delay for virtually concatenated Ethernet over SONET systems, *IEEE ICCCN 2004*. pp. 205-210.

[3] S. Ahuja, M. Krunz, T. Korkmaz, Optimal path selection for minimizing the differential delay in ethernet-over-SONET, *Computer Networks*, Vol. 50, Issue 13, 2006, pp. 2349-2363.

[4] R. Andersen, F. Chung, A. Sen and G. Xue, On disjoint path pairs with wavelength continuity constraint in WDM networks, *IEEE INFOCOM'2004*, pp. 524-535.

[5] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi, Quality of service based routing: A performance perspective, *ACM SIGCOMM'1998*, pp. 17-28.

[6] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda, and T. Przygienda, QoS routing mechanisms and OSPF extensions, *IETF RFC 2676*, 1999.

[7] J. Bang, S. Tekinay and N. Ansari, Performance analysis of cell switching management scheme in wireless packet communications, *Proceedings of IEEE Globecom'2001*, pp. 3639-3643.

[8] BRITE; http://www.cs.bu.edu/brite.

[9] J. Chen, R. Sundaram, M. Marathe and R. Rajaraman, The confluent capacity of the Internet: congestion vs. dilation, *IEEE ICDCS'06*, pp. 5.

[10] J.C. Chen and S.H. Chan, Multipath routing for video unicast over bandwidth-limited networks, *IEEE Globecom'2001*, pp. 1963-1997.

[11] J.C. Chen, S.H. Chan and V. Li, Multipath routing for video delivery over bandwidth-limited networks, *IEEE Journal on Selected Areas in Communications*, Vol. 22(2004), pp. 1920-1932.

[12] I. Cheng, A. Basu, Y. Zhang, S. Tripathi, QoS specification and adaptive bandwidth monitoring for multimedia delivery; *EUROCON'2001*, pp. 483-486.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, second edition, The MIT Press, 2001.

[14] P. Djukic, S. Valaee, Reliable packet transmissions in multipath routed wireless networks *IEEE Transactions on Mobile Computing*, Vol. 5, Issue 5, 2006, pp. 548-559.

[15] D. Du and S. Kabadi, An improved algorithm for decomposing arc flows into multipath flows, *Operations Research Letters*, Vol. 34, Issue 1, 2006, pp. 53-57.

[16] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.

[17] L. Gao and D. Towsley, Supplying instantaneous video-on-demand services using controlledmulticast, *IEEE ICMCS'1999*, pp. 117-121.

[18] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. *W. H. Freeman & Co.*, New York, 1979.

[19] S. Huang, B. Mukherjee, C. Martel, Survivable Multipath Provisioning with Differential Delay Constraint in Telecom Mesh Networks, *IEEE Infocom'2008*, pp. 718-725.

[20] D. Jurca and P. Frossard, Media flow rate allocation in multipath networks, *IEEE Transactions on Multimedia*, Vol. 9(2006), pp. 1227-1240.

[21] S. J. Lee and M. Gerla, Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks, *Proc. IEEE ICC'2001*, pp. 3201-3205, 2001.

[22] D. Lorenz and D. Raz, A Simple Efficient Approximation Scheme for the Restricted Shortest Path Problem, *Operations Research Letters*, Vol. 28, Issue. 5, 2001, pp. 213-219.

[23] W. Lou and Y. Fang, A multipath routing approach for secured data delivery, *Proceedings of IEEE Milcom'2001*, pp. 1467C1473.

[24] S. Misra, G. Xue, and D. Yang, Polynomial time approximations for multi-path routing with bandwidth and delay constraints, *IEEE INFOCOM'2009*.

[25] A. Nasipuri and S.R. Das, On-demand multi-path routing for mobile ad hoc networks, *Proc. IEEE ICCCN'1999*, pp. 64-70, 1999.

[26] N. Singhal, L. Sahasrabuddhe and B. Mukherjee, Provisioning of survivable multicast sessions against single link failures in optical WDM mesh networks, *IEEE Journal of Lightwave Technology*, Vol. 21, Issue 11, 2003, pp. 2587-2594.

[27] A. Srinivas, E. Modiano Minimum energy disjoint path routing in wireless ad-hoc networks *ACM Mobicom'2003*, pp. 122-133.

[28] A. Srivastuva, S. Acharyu, M. Alicherry, B. Gupta and P. Risbood, Differential Delay Aware Routing for Ethernet over SONET/SDH, *IEEE Infocom'2005*, pp.1117-1127.

[29] A. Srivastava, Flow Aware Differential Delay Routing for next-generation Ethernet over SONET/SDH *IEEE ICC'2006*, pp. 140-145.

[30] H. Suzuki and F.A. Tobagi, Fast bandwidth reservation scheme with multilink and multipath routing in ATM networks, *IEEE INFOCOM'1992*, pp. 2233-2240.

[31] J. Tang and G. Xue, Node-disjoint path routing in wireless networks: tradeoff between path lifetime and total energy, *IEEE ICC'2004*, pp. 3812-3816.

[32] S. Vutukury and J. J. Garcia-Luna-Aceves, MDVA: A Distance-Vector Multipath Routing Protocol, *Proc. IEEE INFOCOM'2001*, pp. 557-564, 2001.

[33] B.M. Waxman, Routing of multipoint connections, *IEEE Journal on Selected Areas in Communications*, Vol. 6(1988), pp. 1617-1622.

[34] D. Wu, Y.T. Hou, W. Zhu, Y.Q. Zhang and J.M. Peha, Streaming video over the Internet: approaches and directions, *IEEE Transactions on CAS for Video Technology*, Vol. 11(2001), pp. 282-300.

[35] G. Xue, A. Sen, W. Zhang, J. Tang and K. Thulasiraman, Finding a Path Subject to Many Additive QoS Constraints, *IEEE/ACM Transactions on Networking*; Vol. 15(2007), pp. 201-211.

[36] G. Xue, W. Zhang, J. Tang and K. Thulasiraman, Polynomial Time Approximation Algorithms for Multi-Constrained QoS Routing, *IEEE/ACM Transactions on Networking*, Vol. 16 (2008), pp. 656-669.

[37] Y. Ye, An $O(n^3\mathcal{L})$ potential reduction algorithm for linear programming, *Mathematica Programming*, Vol. 50(1991), pp. 239-258.

[38] Q. Zhang, W. Zhu and Y.Q. Zhang, Resource allocation for multimedia streaming over the Internet, *IEEE Transactions on Multimedia*, Vol. 3(2001), pp. 339-355.

[39] W. Zhang, G. Xue, J. Tang and K. Thulasiraman, Dynamic light trail routing and protection issues in WDM optical networks, *IEEE GLOBECOM'2005*, pp. 1963-1967.