CCIT Report #429
June 2003

Efficient Multipath Routing Schemes for Congestion Minimization

Ron Banner and Ariel Orda
Department of Electrical Engineering
Technion – Israel Institute of Technology
Haifa 32000, Israel
{ banner@tx, ariel@ee .technion.ac.il

Abstract

Unlike traditional routing schemes that route all traffic along a single path, multipath routing strategies split the traffic among several paths in order to ease congestion. We identify the major essential requirements of multipath routing. A multipath routing scheme should limit the number of paths per destination, the end-to-end delay of each path and the delay variance (delay-jitter) between different paths that ship traffic towards the same destination. In spite of the important benefits provided by multipath routing schemes, they got relatively little attention in the literature; moreover, most studies focused on heuristic methods. This work provides the first comprehensive study that establishes *practical* multipath routing strategies with *provable performance guarantees*, in terms of load balancing and congestion minimization.

Keywords: Multipath Routing, Delay-Jitter, Load Balancing, QoS.

1. Introduction

1.1. General

Practical routing schemes typically focus on discovering a single "optimal" path for routing, according to some desired metric. Accordingly, traffic is always routed over a single path, which often results in substantial waste of network resources. *Multipath routing* is an alternative routing scheme that distributes the traffic among multiple "good" paths instead of routing all traffic along a single "best" path.

Multipath routing can significantly reduce congestion in "hot spots", by deviating traffic to unused network resources, thus improving network utilization and providing load balancing [1]. Moreover, congested links usually result in poor performance and high variance. For such circumstances, multipath routing can offer steady and smooth data streams [2].

In spite of these important benefits, multipath routing schemes suffer from several drawbacks. The first and most problematic one results from routing over paths with variable latencies. Quoting [5]: "Since each of the redundant paths may have a different latency involved, having packets take separate paths can cause packets to always arrive out of order, increasing delivery latency and buffering requirements." Packet reordering causes TCP to enter a mode termed "fast retransmit", which consumes extra bandwidth as it attempts to unnecessarily retransmit reordered packets that are assumed to have been lost. Hence, reordering can be detrimental to network performance [5]. In order to avoid these drawbacks, the delay difference (also termed *delay jitter*) must be small with respect to the packet serialization time. Quoting [6]: "Delay differences greater than three times the packet serialization time can cause terrible TCP performance."

Another problem that arises in the use of multipath routing schemes is the considerable overhead associated with establishing and maintaining many routes for the same destination [7]. Moreover, the complexity of a scheme that distributes traffic among multiple paths considerably increases as the number of paths increases. Finally, there may be a limit on the number of explicitly routed paths that can be set up between a pair of nodes, as is the case with label switched paths in MPLS. Therefore, it is desirable to use as few paths as possible while at the same time exploit the benefits of multipath routing. Fortunately, usually just a few paths are needed in order to significantly exploit these benefits. Indeed, it has been concluded that congestion is largely reduced when such one or two paths are identified in addition to the traditional single path [8]; moreover, survivability techniques usually establish only a single backup path for each active path [9].

The rest of this document is organized as follows. In the reminder of this section, we outline our main contributions. In section 2, we introduce some terminology and definitions, and formulate the main problems considered in this study. In section 3, we present some background and survey related previous works. In section 4, we consider multipath routing schemes with additive QoS requirements and end-to-end reliability constraints. In section 5, we provide an approximation solution to the fundamental multipath routing problem of minimizing congestion under a restriction on the delay jitter. In section 6, we consider the problem of minimizing congestion

subject to a restriction on the number of paths per destination. Finally, in section 7, we propose several directions for future work.

1.2 Our Contribution

To the best of our knowledge, this work provides the first comprehensive study that establishes *practical* multipath routing strategies with *provable performance guarantees*. Based on the literature, we identify the major essential requirements of multipath routing. More specifically, we observe that, in practice, a multipath routing scheme should restrict the number of paths per destination, the end-to-end delay of each path and the delay-jitter among all paths. Accordingly, we establish multipath routing schemes that address these requirements while balancing the network's loads.

We show that minimizing congestion under either delay or delay-jitter constraints is computationally intractable. Accordingly, for both problems, we establish ε-optimal approximation schemes; in particular, we note that this is the first scheme with provable approximation bounds for the problem that restricts the delay jitter, which is of major importance in the context of multipath routing [5],[6],[16],[17]. Finally, for the scheme that restricts the end-to-end delay, we present an important application, which minimizes the congestion under end-to-end reliability constraints.

Consider now the essential restriction on the number of routing paths per destination. We show that minimizing the congestion under a restriction on the number of paths per destination is computational intractable. However, for this problem we establish a $\left(1+1/r\right)$-approximation scheme that, for any $r \geq 1$, violates the constraint on the number of paths by a factor of at most r. Finally, we broaden the scope of this problem by studying the dual problem, which restricts the congestion while minimizing the number of paths per destination. For this problem we establish an r-approximation scheme, which, for any $r \geq 1$, violates the restriction on the congestion by at most a factor of $(1+1/r)$.

2. Model and Problems Formulation

This section formulates the general model and main problems that are addressed in this study. We begin with a definition of a general communication network.

A *network* is represented by a directed graph $G(V,E)$, where V is the set of nodes and E is the set of links. Let $N=|V|$ and $M=|E|$. A *path* is a finite sequence of nodes $p=(v_0,v_1,\cdots,v_h)$, such that, for $0\le n\le h-1, (v_n,v_{n+1})\in E$. A path is *simple* if all its nodes are distinct. A *cycle* is a path $p=(v_0,v_1,\cdots,v_h)$ together with the link $(v_h,v_0)\in E$ i.e., $(v_0,v_1,\cdots,v_h,v_0)$. Denote the set of all cycles in a network G by $T(G)$.

A *commodity* is a pair of nodes $(i,j)\in V\times V$ that is assigned with a non-negative demand $\gamma^{(i,j)}$. Let β be the set of all commodities with *positive* demand $\beta=\left\{(i,j)\middle|(i,j)\in V\times V,\gamma^{(i,j)}>0\right\}$. Given a commodity $(i,j)\in V\times V$, we say that node i is the *source node* of the given commodity and node j is the *target node*. If $|\beta|\le 1$, we say that the network has a *single commodity* flow demand. Otherwise, we say that the network has a *multi-commodity* flow demand.

The set $P^{(i,j)}$ is the collection of all directed paths from the source i to the destination j in the network. In addition, let $P\triangleq\bigcup_{(i,j)\in V\times V}P^{(i,j)}$ and let $P_{simple}^{(i,j)}\subseteq P^{(i,j)}$ represents the set of *simple paths* from *i* to *j* in the network. Finally, for each path $p\in P^{(s,t)}$ and link $e\in E$, $\Delta_e(p)$ counts the number of occurrences of the link e in the path p. For example, given a non-simple path $p=(v_0,v_1,v_2,v_3,v_1,v_2,v_4)$ and a link $e=(v_1,v_2)$, we have $\Delta_e(p)=2$.

Each link $e\in E$ is assigned a *weight* $w_e\in\mathbb{Z}^+$ and a *capacity* $c_e\in\mathbb{Z}^+$. We assume that the link weighs w_e constitute an additive metric. We consider a *link state* routing environment, where each source node has an image of the entire network.

Definition 2.1 Given a (non-empty) path p, the weight $W(p)$ of p is defined as the sum of weights of its links, namely, $W(p)=\sum_{e\in p}w_e$.

Definition 2.2 Given a (non-empty) path p, the capacity $C(p)$ of p is defined as the capacity of its bottleneck link, namely, $C(p)=\underset{e\in E}{Min}\{c_e\}$.

Definition 2.3 Let $G(V,E)$ be a network. A *path flow* is a real-valued function $f:P\to\mathbb{R}^+\cup\{0\}$ that satisfies the following two properties:

Capacity constraints: For each $e\in E$, $\sum_{p\in P}\Delta_e(p)\cdot f(p)\le c_e$.

Flow demand: For each commodity $(i,j) \in V \times V$, $\sum_{p \in P^{(i,j)}} f(p) = \gamma^{(i,j)}$.

Definition 2.4 Given is a path flow $f : P \to \mathbb{R}^+ \cup \{0\}$ over a network $G(V,E)$. A *link flow of a commodity* $(i,j) \in V \times V$ is a real-valued function $f : E \times V \times V \to \mathbb{R}^+ \cup \{0\}$ that satisfies, for each link $e \in E$: $f_e^{(i,j)} \triangleq \sum_{p \in P} \Delta_e(p) \cdot f(p)$.

Denote $f_e \triangleq \sum_{(i,j) \in V \times V} f_e^{(i,j)}$.

Definition 2.5 Let $G(V,E)$ be a network. A *cycle flow of a commodity* $(i,j) \in V \times V$ is a real-valued function $f : T(G) \times V \times V \to \mathbb{R}^+ \cup \{0\}$.

Note that, for a given link flow of a commodity $(i,j) \in V \times V$, the path flow $f : P^{(i,j)} \to \mathbb{R}^+ \cup \{0\}$ is not necessarily unique. In addition, the path flow representation (which is of size $O(|P|)$) may have exponential size (with respect to the network representation). However, it follows from the flow decomposition theorem [12] that any path flow assigned for a commodity $(i,j) \in V \times V$ (i.e. the collection of pairs $(p, f(p))$ for each $p \in P^{(i,j)}$) has a corresponding path flow with at most M paths and cycles with a positive flow that share the same link flow representation.

Definition 2.6 Given a network $G(V,E)$ and a link flow $\{f_e\}$, the value $\frac{f_e}{c_e}$ is the *link congestion factor*.

Definition 2.7 Given a network $G(V,E)$ and a link flow $\{f_e\}$, the *network congestion factor* is the largest link congestion factor in the network, i.e., $\max_{e \in E} \left\{ \frac{f_e}{c_e} \right\}$.

As noted in [3],[11],[19] the network congestion factor provides a good indication of congestion.

We now establish that minimizing the network congestion factor is equivalent for the single commodity case to maximizing the flow i.e., the objective function of the well known Max Flow Problem [12].

Theorem 2.1 Given a network $G(V,E)$ two nodes $\{s,t\}$ capacities $\{c_e\}$ and a demand γ. $\{f_e\}$ is a solution to the instance $\langle G(V,E), \{c_e\}, \{s,t\} \rangle$ of the Maximum Flow Problem that transfers $F \geq \gamma$ flow units from *s* to *t iff* $\left\{ f_e \cdot \frac{\gamma}{F} \right\}$ is a link flow that transfer γ flow units from *s* to *t* such that the network congestion factor is minimized .

The proof to the Theorem appears in the Appendix.

We are now ready to formulate the main problems considered in this study. All these minimize the network congestion factor subject to different considerations. We saw that, in the single commodity case, minimizing the network congestion factor is equivalent to maximizing the total throughput. Therefore, all these problems remain equivalent for the single commodity case if we consider a different objective function, namely that of maximizing the throughput. However, whereas maximizing the throughput is not well defined for the multi-commodity case, minimizing the network congestion factor is well defined both for the single commodity and the multi-commodity cases.

We proceed to present the first problem. We are given a network with one or more commodities that need to transfer some flow demand subject to some given QoS requirements. The goal of a QoS multipath routing scheme is to identify several paths for each commodity, each meeting the QoS requirement, such that the load over the most utilized links in the network is minimized. This can be formulated as follows.

Problem RMP (Restricted Multipath) Given are a network $G(V,E)$, for each link $e \in E$ a weight $w_e > 0$ and a capacity $c_e > 0$ and, for each commodity $(i,j) \in V \times V$, a demand $\gamma^{(i,j)}$ and a weight restriction $W^{(i,j)}$. Find a path flow that minimizes the network congestion factor, such that, if $P_1^{(i,j)} \subseteq P^{(i,j)}$ is the collection of all paths in $P^{(i,j)}$ that are assigned with a positive flow, then, for each $p \in P_1^{(i,j)}$, it holds that $W(p) \le W^{(i,j)}$.

We shall later prove that Problem RMP is intractable. We will also show that the solution can be used in order to support end-to-end reliability requirements in multipath routing schemes.

Although Problem RMP captures a variety of QoS requirements in multipath routing, it does not cope with the important delay-jitter problem mentioned in the Introduction. Hence, we proceed to present a revised version of Problem RMP, which copes with the delay jitter restrictions. We note that, when attempting to minimize congestion subject to delay-jitter restrictions, we may end up with some unexpected behavior, as illustrated by the following example.

Example: Consider the network depicted in Fig. 2.1. Suppose that the delay-jitter restriction is 2 i.e., the end-to-end delay variance between any two paths that carry a positive flow must be at most 2. It is easy to see that a solution that minimizes the network congestion factor must allocate one flow unit to the path (*s*,*t*) and one flow unit to the *non-simple* path (*s*,*a*,*b*,*c*,*a*,*t*).

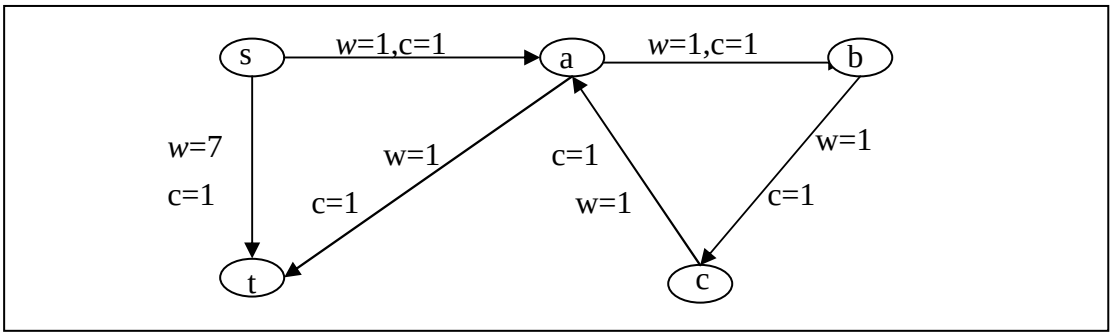

Fig. 2.1 Minimizing the congestion subject to delay jitter restrictions

This example illustrates that a solution that minimizes congestion subject to delay-jitter restrictions may use network links in order to "accumulate" delay. Thus, network links may be used for purposes different than their "traditional" goal, namely transferring flow demands. Therefore, in order to control this phenomenon, we restrict the number of hops that each path can use.

Problem RDJM (Restricted Delay-Jitter Multipath) Given are a network $G(V,E)$, for each link $e \in E$ a weight $w_e > 0$ and a capacity $c_e > 0$ and, for each commodity $(i,j) \in V \times V$, a demand $\gamma^{(i,j)}$, an hop-count restriction $1 \le H^{(i,j)} \le N\text{-}1$, a weight restriction $W^{(i,j)}$, and a delay-jitter requirement $J^{(i,j)}$. Find a path flow that minimizes the network congestion factor, such that, if $P_2^{(i,j)} \subseteq P^{(i,j)}$ is the collection of all paths in $P^{(i,j)}$ that are assigned with a positive flow, then, for each $p_1, p_2 \in P_2^{(i,j)}$, it holds that it holds that $|p_1|,|p_2| \le H^{(i,j)}$, $W(p_1), W(p_2) \le W^{(i,j)}$ and $\left| W(p_1) - W(p_2) \right| \le J^{(i,j)}$.

Although the restriction on the hop count reduces the severity of the phenomenon that was described in Fig. 2.1, it does not eliminate it completely. In Section 5, we explain how the solution to Problem RDJM can be modified so that non-simple paths can be avoided using a bounded buffer at the source node.

Problems RMP and RDJM did not limit the number of different paths over which a commodity is shipped. However, as explained in the Introduction, in practice it is essential to limit this number. Accordingly, we define the following *K-Path Routing (KPR)* problem.

Problem KPR (K-Path Routing) Given are a network $G(V,E)$, for each link $e \in E$ a capacity $c_e > 0$, and, for each commodity $(i,j) \in V \times V$, a demand $\gamma^{(i,j)}$ and a *split restriction* $\mathrm{K}^{(i,j)}$. Find a path flow that minimizes the network congestion factor, such that, if $P_3^{(i,j)} \subseteq P^{(i,j)}$ is the collection of all paths in $P^{(i,j)}$ that are assigned with a positive flow, then $\left| P_3^{(i,j)} \right| \le \mathrm{K}^{(i,j)}$.

3. Related Work

The limitations of current routing protocols in terms of traffic engineering have been largely recognized. Such traffic engineering requirements were specified in [30]. Although multipath routing plays a major role in these requirements, most multipath routing schemes suggested thus far entirely focused on heuristic methods.

Congestion Avoidance in Multipath Routing In [14], a multipath routing scheme, termed Equal Cost Multipath (ECMP), has been proposed for balancing the load along multiple paths of equal cost. However, since ECMP splits the traffic equally, the flow distribution may not reflect the congestion state of the network. [31] suggests a scheme based on periodically probing multiple paths and re-distribute the traffic in order to balance the loads. In [15], explicit routing algorithms for traffic engineering are considered. These schemes consider traffic engineering as an optimization problem with an objective function that is identical to ours i.e., minimizing the network congestion factor. However, both [15] and [31] do not consider QoS restrictions such as end-to-end delay or delay jitter. As a matter of fact, the only restriction considered by [15] is a special case of a restriction that is discussed by us i.e., that of limiting the number of paths allocated to each commodity. More specifically, [15] restricts the flow demand of each commodity to travel only over a single path. Moreover, for that case, only *heuristic* schemes were proposed.

Delay-Jitter & End-to-End Delay In spite of the major importance of restricting the delay jitter in multipath routing schemes [5], [6], [16], [17] we are not aware of any previous algorithmic solution that addresses this problem. In addition, we are not aware of any prior work on multipath routing schemes that minimizes congestion while using only paths with bounded delay, or any algorithmic solution to the equivalent problem of maximizing the flow subject to a weight restriction on the paths.

K Path Routing Problem KPR, which restricts the number of paths that are allocated to each demand, contains as a special case the well-known *unpalatable flow problem* [18], which has a 2-approximation scheme for the single source case [19]. In [4], a 0.5-approximation scheme is provided for the *k splittable flow problem* in the single commodity case. This problem maximizes the total flow under a restriction on the number of paths that are allowed to carry a positive flow.

4. Solution of problem RMP

In this section we aim at solving problem RMP, i.e., the problem of minimizing congestion subject to additive QoS requirements. In addition, we present an important application that supports end-to-end reliability requirements. First we establish that the problem is intractable.

4.1 Intractability of Problem RMP.

We show that Problem RMP can be reduced to the *Partition* problem [23].

Theorem 4.1 Problem RMP is NP-hard.

Proof First, let us define the single-commodity case of problem RMP as a decision problem.

Given are a network $G(V,E)$, for each link $e \in E$, a weight $w_e > 0$ and a capacity $c_e > 0$, and, for a commodity $(s,t) \in V \times V$, a demand $\gamma > 0$ and a weight restriction W. Is there a path flow with network congestion factor of at most α such that, if path p transfers a positive amount of flow then $W(p) \le W$?

Consider the following instance of the Partition problem; given an ordered set of elements $a_1, a_2, \ldots, a_{2n}$ that constitute a set A with size $s(a) \in \mathbb{Z}^+$ for each $a \in A$, find a subset $A' \subseteq A$ such that A' contains exactly one element of $a_{2i-1,} a_{2i}$ for $1 \le i \le n$ and $\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a)$.

We transform Partition to RMP as follows (see also Fig. 1):

- Given an element $a_i \in A$ with size $s(a_i)$, define a unit capacity link $u_i \to v_i$ with weight $s(a_i)$.
- For each link $u_{2i-1} \to v_{2i-1}$ $1 \le i \le n$, define a link $v_{2i-1} \to u_{2i+1}$ and a link $v_{2i-1} \to u_{2i+2}$. Assign to both a unit capacity and a zero weight.
- For each link $u_{2i} \to v_{2i}$, $1 \le i \le n$, define a link $v_{2i} \to u_{2i+1}$ and a link $v_{2i} \to u_{2i+2}$. Assign to both a unit capacity and a zero weight.
- Define links $s \to u_1, s \to u_2$ and links $v_{2n-1} \to t, v_{2n} \to t$. Assign to each a unit capacity and a zero weight.
- Set: $W \leftarrow \frac{1}{2} \cdot \sum_{a \in A} s(a)$ and $\gamma \leftarrow 2$.

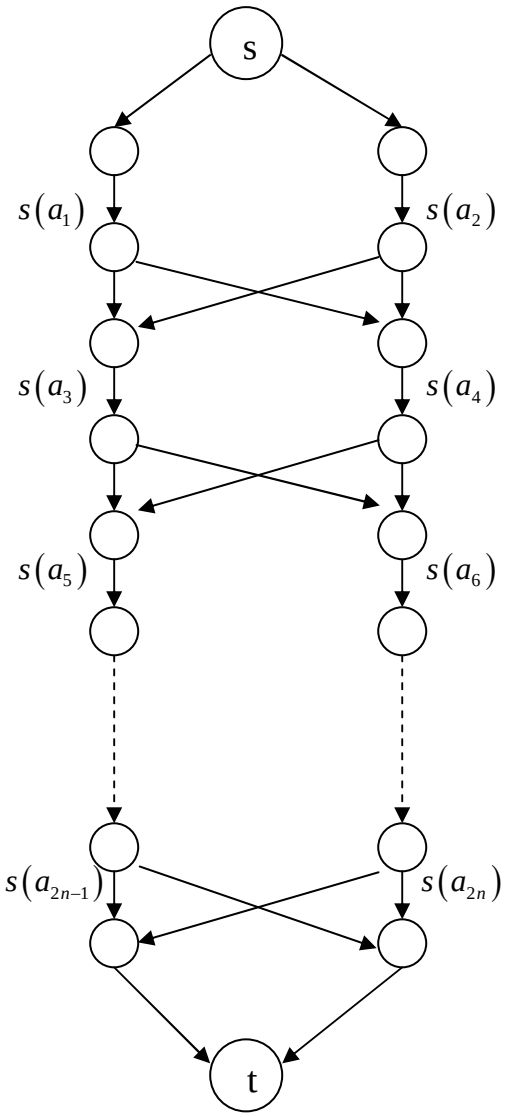

Fig. 4.1: Reduction of Partition to RMP

We shall prove that it is possible to transfer 2 flow units over paths whose weights are not larger than W without exceeding the network congestion factor of $\alpha = 1$ *iff* there is a subset $A' \subseteq A$ such that A' contains exactly one element of $a_{2i-1,}a_{2i}$ for $1 \le i \le n$ and $\sum_{a \in A'} s(a) = \sum_{a \in A-A'} s(a)$. (Remark: We refer to elements and their sizes interchangeably.)

$\Leftarrow$: Suppose there is a subset $A' \subseteq A$ such that A' contains exactly one of $a_{2i-1,}a_{2i}$ for $1 \le i \le n$ and $\sum_{a \in A'} s(a) = \sum_{a \in A-A'} s(a)$. Then, it is easy to see that the selection of the links that represents the elements in A' and the zero weight links that connect those links constitutes a path. Also, it is easy to see that this path is disjoint to the path that the complement subset $A - A'$ defines. Since all capacities equal to 1, we have two disjoint paths that can transfer together exactly 2 units of flow without violating the

congestion constraint $\alpha = 1$. The weight restriction is preserved since the two defined paths have weight of $\frac{1}{2} \cdot \sum_{a \in A} s(a)$, which was defined to be the weight restriction W.

$\Rightarrow$: Suppose there is a path flow that transfers two flow units over paths that are not bigger than W. It is easy to see that all paths in the graph must be simple since the graph is a DAG. Select one path that transfers a positive flow and denote it as p. Define an empty set S. For every link in p, with weight $s(a_i)$, insert the element a_i into S. Since all links in the graph have one unit of capacity, the selected path p is not able to transfer more than one unit of flow. Now, delete all the links that constitute path p. Since p is simple and since it transfers at most one unit of flow, there must be another path that is disjoint to the selected path that transfers a positive flow over the links that were left in the graph. For each link in that path with size $s(a_i)$, insert the element a_i into a different set S'. We will now prove that $A = S \cup S'$, $S \cap S' = \phi$ and, finally, $\sum_{a \in S} s(a) = \sum_{a \in S'} s(a)$.

Since S and S' were constructed out of disjoint paths, it is obvious that $S \cap S' = \phi$. Since every path must traverse either $s(a_{2i-1})$ or $s(a_{2i})$ for each $1 \le i \le n$, and since both paths are disjoint, $S \cup S' = \{a_i\}_{i=1}^{2n} = A$.
Since both paths have weights that are not longer than W, we have:

$$\sum_{a \in S} s(a) \le W, \quad \sum_{a \in S'} s(a) \le W. \qquad (1)$$

Since $W \triangleq \frac{1}{2} \cdot \sum_{a \in A} s(a)$ and $S \cup S' = A$, we get:

$$\sum_{a \in S} s(a) + \sum_{a \in S'} s(a) = \sum_{a \in A} s(a) = 2 \cdot W. \qquad (2)$$

Note that if variables x_1, x_2 satisfy $x_1 \le B$, $x_2 \le B$ and in addition $x_1 + x_2 = 2 \cdot B$, it follows that $x_1 = x_2 = B$. Accordingly, we conclude from (1) and (2) that $\sum_{a \in S} s(a) = \sum_{a \in S'} s(a) = W$.

Thus, problem RMP is NP hard. ■

4.2 Linear Programming Formulation

In this section we present a linear programming formulation for Problem RMP. To that end, we need some additional notation.

We are given a network $G(V,E)$, for each link $e \in E$ a weight $w_e \in \mathbb{Z}^+$ and a capacity $c_e > 0$, and, for each commodity $(i,j) \in V \times V$ a demand $\gamma^{(i,j)}$ and a weight restriction $W^{(i,j)}$. Let α be the network congestion factor. Define $f_e^{\omega,(i,j)}$ as the flow of commodity (i,j) over link $e=(u,v) \in E$ that has traversed through paths $p \in P^{(i,u)}$ of total weight $W(p)=\omega$. Finally, for each $v \in V$, denote by $O(v)$ the set of links that emanate from v, and by $I(v)$ the set of links that enter that node, namely $O(v)=\{(v,l)|(v,l) \in E\}$ and $I(v)=\{(w,v)|(w,v) \in E\}$. Then, Problem RMP can be formulated as the following linear program:

Minimize α

s.t

$$\sum_{e \in O(v)} f_e^{\omega,(i,j)} - \sum_{e \in I(v)} f_e^{\omega - w_e,(i,j)} = 0 \quad ,\forall (i,j) \in \beta \quad ,\forall v \in V - \{i,j\} \quad ,\forall \omega \in \left[0, W^{(i,j)}\right] \tag{1}$$

$$\sum_{e \in O(i)} f_e^{\omega,(i,j)} - \sum_{e \in I(i)} f_e^{\omega - w_e,(i,j)} = 0 \quad ,\forall (i,j) \in \beta \quad ,\forall \omega \in \left[1, W^{(i,j)}\right] \tag{2}$$

$$\sum_{e \in O(i)} f_e^{0,(i,j)} = \gamma^{(i,j)} \quad ,\forall (i,j) \in \beta \tag{3}$$

$$\sum_{(i,j) \in \beta} \sum_{\omega=0}^{W^{(i,j)}} f_e^{\omega,(i,j)} \le c_e \cdot \alpha \quad ,\forall e \in E \tag{4}$$

$$f_e^{\omega,(i,j)} = 0 \quad ,\forall \omega < 0 \ ,\forall (i,j) \in \beta \ , \forall e \in E \tag{5}$$

$$f_e^{\omega,(i,j)} \ge 0 \quad ,\forall (i,j) \in \beta \ ,\forall \omega \in \left[0, W^{(i,j)}\right] \ ,\forall e \in E \tag{6}$$

$$\alpha \ge 0 \tag{7}$$

Fig. 4.2 Program RMP

Note that the variables of the linear program are the link flows $\left\{ f_e^{\omega,(i,j)} \right\}$ and the network congestion factor α.

In this linear programming formulation the objective function is to minimize the network congestion factor. Constraints (1), (2) and (3) are nodal flow conservation constraints. Equation (1) states that the traffic flowing out of node v, which has traversed through paths $p \in P^{(i,v)}$ of weight $W(p)=\omega$, has to be equal to the traffic flowing into node v, through paths $p' \in P^{(i,u)}$ and links $e=(u,v) \in E$, such that $W(p')+w_e=\omega$; since $\omega \in \left[0, W^{(i,j)}\right]$, the weight restriction is preserved for each commodity $(i,j) \in \beta$; finally, equation (1) must be satisfied for each node other than the source node and the destination node for each commodity with a positive demand.

Equation (2) extends for each $(i,j) \in \beta$ the validation of equation (1) to hold for traffic that encounters node i after it has already traveled over paths with non-zero weights. Informally, equation (2) states that "old" traffic that was already traversing over at least one link must satisfy constraint (1). Equation (3) states that, for each commodity $(i,j) \in \beta$, the traffic flowing out of source i, which has already traversed paths of weight $\omega = 0$, must be equal to the demand $\gamma^{(i,j)}$. Equation (4) is the link capacity utilization constraint. Expression (5) rules out non-feasible flows, and Expression (6) and (7) restricts all variables to be non-negative.

We can solve Program RMP using any polynomial time algorithm for linear programming [24]. The solution to problem RMP is then achieved by decomposing the link flow $\left\{ f_e^{\omega,(i,j)} \right\}$ into a path flow that satisfies the weight restriction. This is done by Algorithm PFC, specified in Fig. 4.3, which is an (elaborated) generalization of the Flow Decomposition Algorithm [12].

For each commodity $(i,j) \in V \times V$ the algorithm uses the link flow $\left\{ f_e^{\omega,(i,j)} \right\}$ in order to define paths that transfer a total flow of $\gamma^{(i,j)}$ flow units over paths that have total weight not larger than $W^{(i,j)}$. At each iteration, the algorithm uses Procedure Path Construction, specified in Fig. 4.4, in order to define a path with end-to-end weight of at most $W^{(i,j)}$ whose corresponding link flows $\left\{ f_e^{\omega,(i,j)} \right\}$ are all positive. The flow over this path is defined to be equal to the smallest flow $f_e^{\omega,(i,j)}$ that belongs to the path. Then, the algorithm subtracts the flow that traverses through that path from the demand $\gamma^{(i,j)}$ and from each variable $f_e^{\omega,(i,j)}$ in the path. The algorithm stops when the demand $\gamma^{(i,j)}$ is zeroed for each $(i,j) \in V \times V$. Thus, the resulting path flow transfers $\gamma^{(i,j)}$ flow units from source *i* to destination *j* over paths with weight of at most $W^{(i,j)}$ for each $(i,j) \in V \times V$.

Algorithm PFC $\left(G(V,E),\left\{f_e^{\omega,(i,j)}\right\},\left\{\gamma^{(i,j)}\right\}\right)$

Initialization:

For each commodity $(i,j)\in V\times V$ and each path $p\in P^{(s,t)}$: $f(p)\leftarrow 0$.

For each commodity $(s,t)\in V\times V$:

While $\gamma^{(s,t)}>0$ do:

1. $S\leftarrow$ Path_Construction$\left(G(V,E),\{s,t\}\left\{f_e^{\omega,(i,j)}\right\}\right)$.
2. $f_{e_k}^{\Delta_k,(s,t)}\leftarrow f_{e_k}^{\Delta_k,(s,t)}-\min\limits_{(e_k,\Delta_k)\in S}\left\{f_{e_k}^{\Delta_k,(s,t)}\right\}$, for each $(e_k,\Delta_k)\in S$.
3. $\gamma^{(s,t)}\leftarrow\gamma^{(s,t)}-\min\limits_{(e_k,\Delta_k)\in S}\left\{f_{e_k}^{\Delta_k,(s,t)}\right\}$.
4. Denote path $s\xrightarrow{e_1}v_1\xrightarrow{e_2}v_2\cdots v_{|s|}\xrightarrow{e_{|s|}}t$ as p, where e_k corresponds to the pair $(e_k,\Delta_k)\in S$. $f(p)\leftarrow\min\limits_{(e_k,\Delta_k)\in S}\left\{f_{e_k}^{\Delta_k,(s,t)}\right\}$.

Return the path flow f.

Fig 4.3: Program RMP Algorithm Path Flow Construction (PFC)

We turn to explain the main idea behind Procedure Path Construction, which is specified in Fig. 4.4. The procedure identifies a path $p:s\xrightarrow{e_1}u_1\xrightarrow{e_2}u_2\cdots u_{h-1}\xrightarrow{e_h}t$ whose corresponding link flows $\left\{f_{e_1}^{0,(s,t)},f_{e_2}^{w_{e_1},(s,t)},\cdots,f_{e_h}^{w_{e_1}+\cdots+w_{e_h},(s,t)}\right\}$ are all positive. This is done by employing the following property that characterizes the solution to Program RMP. If a *positive* flow $f_e^{\omega-w_e,(s,t)}$ enters through link $e=(u,v)$ into node $v\in V-\{s,t\}$, then there exists a *positive* flow $f_{e'}^{\omega,(s,t)}$ that emanates out of v through a link $e'=(v,w)$. Since each positive flow $f_e^{\omega,(s,t)}$ must satisfy $\omega\le W^{(s,t)}$ it follows that, if we follow these positive flows from the source s, we establish a path that satisfies the weight restriction $W^{(s,t)}$. We now prove that if we follow these positive flows for a finite number of times we eventually find a positive flow that enters into the destination i.e., we eventually identify a directed path from s to t. We note that this path is not necessarily simple.

Lemma 1: Consider the nodes $\{u_k\}$ identified by Procedure Path Construction (Fig. 4.4) for the input $\left\langle G(V,E),\{s,t\},\left\{f_e^{\omega,(i,j)}\right\}\right\rangle$. There exists an h, $h\le W^{(s,t)}/\min\limits_{e\in E}\{w_e\}$, such that the sequence $(u_0,u_1,\ldots,u_h)$ is a path from s to t with a weight of at most $W^{(s,t)}$.

Proof: It follows from constraint (3) that, if $\gamma^{(s,t)}>0$, then there exists some link $e_0=(u_0,u_1)$ such that the variable $f_{e_0}^{0,(s,t)}$ is positive. Then, from constraints (1) and (2), it follows that, if $u_1\neq j$, then there exists some link $e_1=(u_1,u_2)$ such that the variable $f_{e_1}^{w_{e_0},(s,t)}$ is positive. Thus, applying constraints (1) and (2) for any index k, it follows

that, if there exists a positive variable $f_{e_k}^{\Delta_k,(i,j)}$ where $\Delta_k \triangleq \sum_{l\in[0,k-1]} w_{e_l}$, then, unless $u_{k+1}=t$, there exists a link $e_{k+1}=(u_{k+1},u_{k+2})$ such that the variable $f_{e_{k+1}}^{\Delta_k+d_{e_k},(s,t)}$ is positive. However, since it follows from constraint (5) that $f_e^{\omega,(s,t)}=0$ for each $\omega > W^{(s,t)}$, and since $\Delta_k > W$ for each $k \geq W^{(s,t)}/\min_{e\in E}\{w_e\}$, it follows that there exists a k, $k \leq W^{(s,t)}/\min_{e\in E}\{w_e\}$, such that $u_k=t$. Finally, as the procedure selects only positive variables $f_e^{\omega,(s,t)}$, it follows from (5) that $\omega \in \left[0, W^{(s,t)}\right]$. Thus the path $(u_0,u_1,\ldots,u_k)$ has a weight of at most $W^{(s,t)}$. ■

Procedure Path Construction $\left(G(V,E),\{s,t\},\left\{f_e^{\omega,(i,j)}\right\}\right)$

Initialization

$S \leftarrow \phi$, $u_0 \leftarrow s$, $\Delta_0 \leftarrow 0$, $k \leftarrow 0$

While $u_k \neq t$ do

1. Select a positive variable $f_{e_k}^{\Delta_k}$ such that $e_k \triangleq (u_k,u_{k+1}) \in E$.
2. $S \leftarrow S \cup (e_k,\Delta_k)$, $\Delta_{k+1} \leftarrow \Delta_k + w_{e_k}$, $k \leftarrow k+1$.

Return S

Fig 4.4: Procedure Path Construction

Note that each iteration of Algorithm PFC zeroes at least one variable $f_e^{\omega,(i,j)}$. Therefore, Algorithm PFC iterates for no more than the number of variables $\left\{f_e^{\omega,(i,j)}\right\}$. In addition, it follows from Lemma 1 that the complexity of Procedure Path Construction is at most $W^{(s,t)}$ for the input $\left\langle G(V,E),\{s,t\},\left\{f_e^{\omega,(i,j)}\right\}\right\rangle$. Since $W^{(s,t)} \leq \left|\left\{f_e^{\omega,(i,j)}\right\}\right|$, it follows that Algorithm PFC has a polynomial complexity with respect to the number of variables $\left\{f_e^{\omega,(i,j)}\right\}$ that are used by Program RMP.

For completion, in Fig. 4.5 we specify Algorithm RMP, which solves Problem RMP.

Algorithm RMP $\left(\mathrm{G(V,E)},\{w_e\},\{c_e\},\left\{\gamma^{(i,j)}\right\},\left\{W^{(i,j)}\right\}\right)$

1. $\left\{f_e^{\omega,(i,j)}\right\} \leftarrow$ **RMP** $\left(\mathrm{G(V,E)},\{w_e\},\{c_e\},\left\{\gamma^{(i,j)}\right\},\left\{W^{(i,j)}\right\}\right)$
2. $f \leftarrow$ **PFC** $\left(G(V,E),\left\{f_e^{\omega,(i,j)}\right\},\left\{\gamma^{(i,j)}\right\}\right)$
3. **Return** path flow f.

Fig 4.5 Algorithm RMP

In Algorithm RMP, the complexity incurred by solving Program RMP (step 1) is polynomial in the number of variables $\left\{f_e^{\omega,(i,j)}\right\}$ [13]. In addition, as shown, the

complexity of Algorithm PFC (step 2) is polynomial in $\left|\left\{f_e^{\omega,(i,j)}\right\}\right|$. Thus, the complexity of Algorithm RMP is polynomial in the number of variables $\left\{f_e^{\omega,(i,j)}\right\}$ that are used by Program RMP.

Nonetheless, each commodity with a positive demand $\gamma^{(i,j)}$ has at least $M \cdot W^{(i,j)}$ variables. Hence, the total number of variables that are used in order to formulate the problem is $M \cdot \sum_{(i,j)\in\beta} W^{(i,j)}$, which may be exponential with respect to the input. Note, however, that when the hop count metric is considered (i.e., $w_e \equiv 1$), the number of variables formulating the problem is polynomial. Therefore, for that case we have a polynomial solution.

4.3 Approximation Scheme for Problem RMP

In this section, we establish an approximation scheme for problem RMP. Specifically, we present an approximation scheme for the case where the weight restrictions $W^{(i,j)}$ are of the same order of magnitude for each $(i,j) \in \beta$, namely $\frac{W^{(i_1,j_1)}}{W^{(i_2,j_2)}} = O(1)$ for every pair of commodities $(i_1 j_1),(i_2 j_2) \in \beta$. We note that since end-to-end delays are normally in the order of milliseconds, the case of $\frac{W^{(i_1,j_1)}}{W^{(i_2,j_2)}} = O(1)$ for every pair of commodities $(i_1 j_1),(i_2 j_2) \in \beta$ is the typical case.

The following approximation scheme (Fig. 4.6) reduces the complexity of Algorithm RMP by reducing the number of variables that are used by Program RMP. Since we have already seen that the number of variables is $M \cdot \sum_{(i,j)\in\beta} W^{(i,j)}$, it follows that, in order to have a polynomial number of variables (and therefore polynomial complexity) it is essential to reduce the value of each weight restriction. To that end, we scale the weight restriction of each commodity into a smaller integral value. As a result, we must also scale the weight of each link. However, in order to ensure that the optimal network congestion factor does not increase, we relax the new weight restriction with respect to its original value. This is done by *rounding up* the weight restrictions and *rounding down* the weight of each link.

RMP Approximation Scheme $\left(G, \{c_e\}_{e\in E}, \{w_e\}_{e\in E}, \{\gamma^{(\mathrm{i,j})}\}_{(\mathrm{i,j})\in\beta}, \{W^{(\mathrm{i,j})}\}_{(\mathrm{i,j})\in\beta}, \varepsilon\right)$

Parameters:

G – network

$\{c_e\}_{e\in E}$ – capacities

$\{w_e\}_{e\in E}$ – weights

$\{\gamma^{(\mathrm{i,j})}\}_{(\mathrm{i,j})\in\beta}$ – demands

$\{W^{(\mathrm{i,j})}\}_{(\mathrm{i,j})\in\beta}$ – weight restrictions

ε – approximation parameter

variables

$W^{\max}$ – largest weight restriction

$W^{\min}$ – smallest weight restriction

$\widetilde{G}(\widetilde{V}, \widetilde{E})$ – auxiliary graph

1 $W^{\min} \leftarrow \min_{(\mathrm{i,j})\in\beta}\{W^{(\mathrm{i,j})}\}$, $W^{\max} \leftarrow \max_{(\mathrm{i,j})\in\beta}\{W^{(\mathrm{i,j})}\}$

2 $\delta \leftarrow \dfrac{W^{\min}\cdot\varepsilon}{N}$

3 Construct network $\widetilde{G}(\widetilde{V}, \widetilde{E})$ as follows:

a. $\widetilde{V} \leftarrow V$

b. $\widetilde{E} \leftarrow \{e \in E \mid w_e \le W^{\max}\}$

c. For each $\tilde{e} \in \widetilde{E}$:

$$\widetilde{w_e} \leftarrow \left\lfloor \frac{w_e}{\delta} \right\rfloor, \quad \widetilde{c_e} \leftarrow c_e.$$

4 For each commodity $(\mathrm{i,j}) \in \beta$:

$$\widetilde{W^{(\mathrm{i,j})}} \leftarrow \left\lceil \frac{W^{(\mathrm{i,j})}}{\delta} \right\rceil.$$

5 Solve instance $\left\langle \widetilde{G}, \{\widetilde{c_e}\}_{e\in\widetilde{E}}, \{\widetilde{w_e}\}_{e\in\widetilde{E}}, \{\gamma^{(i,j)}\}_{(\mathrm{i,j})\in\beta}, \{\widetilde{W^{(i,j)}}\}_{(\mathrm{i,j})\in\beta} \right\rangle$ of problem RMP using Algorithm RMP. Let path flow g represent the solution.

6 Construct a path flow f as follows:

a. Let $h: P^{(i,j)} \to P^{(i,j)}_{\text{simple}}$ map each path $(v_0, v_1, \cdots v_{h-1}, v_h, \cdots v_h, v_{h+1}, \cdots v_n) \in P^{(i,j)}$ into a simple path $(v_0, v_1, \cdots v_{h-1}, v_h, v_{h+1}, \cdots v_n) \in P^{(i,j)}_{\text{simple}}$.

b. For each $p \in P^{(i,j)}_{\text{simple}}$, $f(p) \triangleq \sum_{p' \mid h(p')=p} g(p')$

7. Return path flow f.

Fig. 4.6 RMP Approximation Scheme

Definition 4.1 Given an instance of problem RMP, α^* is the network congestion factor of the optimal solution.

Theorem 4.2 Given an instance $\left\langle G,\{c_e\}_{e\in E},\{w_e\}_{e\in E},\{\gamma^{(i,j)}\}_{(i,j)\in\beta},\{W^{(i,j)}\}_{(i,j)\in\beta}\right\rangle$ of problem RMP and an approximation parameter ε, the output of the *RMP Approximation Scheme*, specified in Fig. 4.6, is a path flow f that satisfies the following:

a. For each commodity $(i,j)\in\beta$, $\sum_{p\in P^{(i,j)}} f(p)=\gamma^{(i,j)}$ i.e., the flow demand requirement is satisfied for all commodities.
b. If α^* is the network congestion factor of the optimal solution, then, for each e$\in E$, it holds that $\sum_{(i,j)\in\beta}\sum_{p\in P^{(i,j)}}\Delta_e(p)\cdot f(p)\le\alpha^*\cdot c_e$, i.e., the network congestion factor is at most α^*.
c. For each commodity $(i,j)\in\beta$, if $p\in P^{(i,j)}$ and $f(p)>0$ then $W(p)\le(1+\varepsilon)\cdot W^{(i,j)}$ i.e., the end-to-end delay is violated by a factor of at most $(1+\varepsilon)$.

Proof

a. Since equation (3) in Program RMP specifies for each commodity $(i,j)\in\beta$, $\sum_{e\in O(i)} f_e^{0,(i,j)}=\gamma^{(i,j)}$, it follows that every path flow representation that corresponds to link flow $\left\{f_e^{\omega,(i,j)}\right\}$ must satisfy for each commodity $(i,j)\in\beta$, $\sum_{p\in P^{(i,j)}} f(p)=\gamma^{(i,j)}$.

b. Since the algorithm rounds down the link weights and rounds up the weight restrictions, it follows that the resulting instance $\left\langle \widetilde{G},\ \widetilde{w_e},\ \widetilde{c_e},\ \{\gamma^{(i,j)}\}_{(i,j)\in\beta},\ \{\widetilde{W^{(i,j)}}\}_{(i,j)\in\beta}\right\rangle$ of problem RMP (that appears in line (5)) *relaxes* the original constraints. Hence, the resulting network congestion factor is at most α^*.

c. Define θ_e to be the discretization error resulted from rounding the weight w_e of link $e\in E$ to a multiple of δ i.e., for each link $e\in E$, $\theta_e \triangleq w_e-\left\lfloor\frac{w_e}{\delta}\right\rfloor\cdot\delta$. Note that $\theta_e=w_e-\left\lfloor\frac{w_e}{\delta}\right\rfloor\cdot\delta\le w_e-\left(\frac{w_e}{\delta}-1\right)\cdot\delta=\delta$. Since by construction, the paths that are used by path flow f are simple, every such path contains at most $N-1$ links. Therefore, the total error in evaluating the weight of these paths is at most $(N-1)\cdot\delta$. Applying similar considerations, it is easy to see that the discretization error resulting from rounding the weight restrictions to multiples of δ is at most δ. Therefore, each path $p\in P^{(i,j)}$ that is used by

path flow f (i.e., $f(p)>0$) has total weight of at most $W^{(i,j)}+N\cdot\delta=W^{(i,j)}+N\cdot\frac{W^{\min}\cdot\varepsilon}{N}=W^{(i,j)}+W^{\min}\cdot\varepsilon\le W^{(i,j)}(1+\varepsilon)$. ■

Given an instance of Problem RMP and an approximation parameter ε, we now show that the proposed approximation scheme for Problem RMP is polynomial .

Theorem 4.3 Given an instance $\left\langle G,\{c_e\}_{e\in E},\{w_e\}_{e\in E},\{\gamma^{(i,j)}\}_{(i,j)\in\beta},\{W^{(i,j)}\}_{(i,j)\in\beta}\right\rangle$ of problem RMP and an approximation parameter ε, RMP Approximation Scheme has a polynomial complexity with respect to the input and the approximation parameter ε.

Proof Clearly, the complexity of RMP Approximation Scheme is determined by step (5). As was already explained in section 4.2, the complexity of this step is polynomial in the number of variables that are used by Program RMP to solve the corresponding instance of step (5). Thus, in order to prove the theorem, we only need to show that Program RMP solves instance $\left\langle \widetilde{G},\ \{\widetilde{c_e}\}_{e\in\widetilde{E}},\ \{\widetilde{w_e}\}_{e\in\widetilde{E}},\ \{\gamma^{(i,j)}\}_{(i,j)\in\beta},\ \{\widetilde{W^{(i,j)}}\}_{(i,j)\in\beta}\right\rangle$ of step (5), using a polynomial number of variables with respect to the input and the value of ε.

We first prove that, if $w_e\le W^{\max}$ for each $e\in E$, then the number of variables considered by Program RMP is at most $2\cdot M\cdot|\beta|\cdot W^{\max}$. To that end, consider the set of variables $\{f_e^{\omega,(i,j)}\}$ that formulates Program RMP. For a link $e\in E$ and a commodity $(i,j)\in\beta$, the number of variables $f_e^{\omega,(i,j)}$ that are used by Program RMP is determined by the number of values that ω can take. Since equations (1) - (7) of Program RMP refer both to the variable $f_e^{\omega-w_e,(i,j)}$ and to the variable $f_e^{\omega,(i,j)}$, for each $\omega\in[0,W^{(i,j)}]$, then for a given commodity $(i,j)\in\beta$ and a link $e\in E$, the set $\{f_e^{\hat{\omega},(i,j)}\}$ where $\hat{\omega}\in[0,W^{(i,j)}]\cup[-w_e,W^{(i,j)}-w_e]=[-w_e,W^{(i,j)}]$ contains all the variables that are used by Program RMP. Since $w_e\le W^{\max}$ for each $e\in E$, it follows that the linear program uses, for a commodity $(i,j)\in\beta$ and a link $e\in E$, at most $w_e+W^{(i,j)}\le W^{\max}+W^{(i,j)}\le 2\cdot W^{\max}$ different variables. Thus, the total number of variables in Program RMP is at most $\sum_{e\in E}\sum_{(i,j)\in\beta}2\cdot W^{\max}=2\cdot\sum_{e\in E}\sum_{(i,j)\in\beta}W^{\max}=2\cdot M\cdot\sum_{(i,j)\in\beta}W^{\max}=2\cdot M\cdot|\beta|\cdot W^{\max}$.

We shall now use this finding, namely that, if $w_e\le W^{\max}$ for each $e\in E$, then the number of variables for an instance $\left\langle G,\{c_e\}_{e\in E},\{w_e\}_{e\in E},\{\gamma^{(i,j)}\}_{(i,j)\in\beta},\{W^{(i,j)}\}_{(i,j)\in\beta}\right\rangle$ of Problem RMP is $2\cdot M\cdot|\beta|\cdot W^{\max}$, in order to show that the number of variables for the

instance $\left\langle \widetilde{G},\ \left\{\widetilde{c_e}\right\}_{e\in\widetilde{E}},\ \left\{\widetilde{w_e}\right\}_{e\in\widetilde{E}},\ \left\{\gamma^{(i,j)}\right\}_{(i,j)\in\beta},\ \left\{\widetilde{W^{(i,j)}}\right\}_{(i,j)\in\beta}\right\rangle$ of step (5) is $O\left(\frac{N\cdot M\cdot|\beta|}{\varepsilon}\right)$. Since, by construction, $\widetilde{E}=\left\{e\in E\,\middle|\,w_e\le W^{\max}\right\}$, it follows that, for each $\tilde{e}\in\widetilde{E}$, $\widetilde{w_e}=\left\lfloor\frac{w_e}{\delta}\right\rfloor\le\left\lceil\frac{W^{\max}}{\delta}\right\rceil=\widetilde{W^{\max}}$. Thus, the number of variables in Program RMP for instance $\left\langle \widetilde{G},\ \left\{\widetilde{c_e}\right\}_{e\in\widetilde{E}},\ \left\{\widetilde{w_e}\right\}_{e\in\widetilde{E}},\ \left\{\gamma^{(i,j)}\right\}_{(i,j)\in\beta},\ \left\{\widetilde{W^{(i,j)}}\right\}_{(i,j)\in\beta}\right\rangle$ is at most $2\cdot M\cdot|\beta|\cdot\widetilde{W^{\max}}$. Hence, since $\widetilde{W^{\max}}=\left\lceil\frac{W^{\max}}{\delta}\right\rceil$, it follows that this number of variables is at most $2\cdot M\cdot|\beta|\cdot\left(\frac{W^{\max}}{\delta}+1\right)$. From the assumption that $\frac{W^{(i_1,j_1)}}{W^{(i_2,j_2)}}=O(1)$, for each $(i_1,j_1),(i_2j_2)\in\beta$, it follows that $\frac{W^{\max}}{W^{\min}}=O(1)$. Therefore, since $\delta\triangleq\frac{W^{\min}\cdot\varepsilon}{N}$, the total number of variables that are used by Program RMP in order to solve instance $\left\langle \widetilde{G},\left\{\widetilde{c_e}\right\}_{e\in\widetilde{E}},\left\{\widetilde{w_e}\right\}_{e\in\widetilde{E}},\left\{\gamma^{(i,j)}\right\}_{(i,j)\in\beta},\left\{\widetilde{W^{(i,j)}}\right\}_{(i,j)\in\beta}\right\rangle$ of step (5) is at most $2\cdot M\cdot|\beta|\cdot\left(\frac{W^{\max}}{\delta}+1\right)=2\cdot M\cdot|\beta|\cdot\left(\frac{W^{\max}}{W^{\min}}\cdot\frac{N}{\varepsilon}+1\right)=O\left(\frac{N\cdot M\cdot|\beta|}{\varepsilon}\right)$. ■

4.4 Applications for Program RMP

Problem RMP may arise in several forms. In the single-commodity case, it adds an additive restriction to the well-known Maximum Flow Problem, which applies to paths that carry a positive flow. This restriction may be important in multipath routing schemes where additive QoS metrics, such as delay and jitter, are considered. In this section, we show that Program RMP can be used in order to support multipath routing with end-to-end reliability requirements, i.e., when we need multipath routing schemes that choose paths with a "good" probability of success.

The notion of reliability can be implemented by assigning to each link in the network a failure probability and restricting all paths that carry positive flow to have an end-to-end success probability that is larger than some given lower bound. This is formulated by the following problem.

Problem ReMP (Reliable Multipath) Given are a network $G(V,E)$, for each link $e\in E$, a failure probability $p_e\ge 0$ and a capacity $c_e>0$, and, for each commodity $(i,j)\in\beta$, a demand $\gamma^{(i,j)}>0$ and a success probability restriction $\Pi^{(i,j)}$. Find a path flow $f:P\to\mathbb{R}^+\cup\{0\}$ that minimizes the network congestion factor such that, if $p\in P^{(i,j)}$ and $f(p)>0$, then $\prod_{e\in p}(1-p_e)\ge\Pi^{(i,j)}$.

We establish an approximation scheme for Problem ReMP that reduces Problem ReMP into Problem RMP, as follows. The approximation scheme considers a set of classes $\left\{1,\left(1+\frac{\varepsilon}{N}\right)^{-1},\left(1+\frac{\varepsilon}{N}\right)^{-2},\cdots\right\}$ where $\varepsilon>0$ is a given approximation parameter. Then, each link with a success probability of $1-p_e$ is assigned with a weight i if $\left(1+\frac{\varepsilon}{N}\right)^{-(i+1)} \le 1-p_e \le \left(1+\frac{\varepsilon}{N}\right)^{-i}$. Thus, restricting the end-to-end weight of each simple path to *B* results with an end-to-end probability of at least $\left(1+\frac{\varepsilon}{N}\right)^{-(B+N)}$. Then, the value of *B* is determined from the success probability restriction $\Pi^{(i,j)}$. This value is set to relax the original instance of Problem ReMP in order to get a network congestion factor that is not larger than the optimal network congestion factor. Finally, the number of variables that are produced for the corresponding instance of Problem RMP is determined by the approximation parameter ε. Fig. 4.7 describes the approximation scheme for Problem ReMP.

ReMP Approximation Scheme $\left\langle G(V,E),\ \{p_e\}_{e\in E},\ \{c_e\}_{e\in E},\ \left\{\gamma^{(\mathrm{i,j})}\right\}_{(\mathrm{i,j})\in\beta},\ \left\{\Pi^{(\mathrm{i,j})}\right\}_{(\mathrm{i,j})\in\beta},\varepsilon\right\rangle$

Parameters:

$G(V,E)$ – network

$\{p_e\}_{e\in E}$ – failure probabilities

$\{c_e\}_{e\in E}$ – capacities

$\left\{\gamma^{(\mathrm{i,j})}\right\}_{(\mathrm{i,j})\in\beta}$ – demands

$\left\{\Pi^{(\mathrm{i,j})}\right\}_{(\mathrm{i,j})\in\beta}$ – probability restrictions

ε – approximation parameter

1. Delete all links with failure probability $p_e = 1$. Let $\widetilde{E}$ be the resulting set of links, and let $p_{\max}$ be the link with the largest failure probability that is smaller than 1.
2. Define the following set of classes: $\left\{1,\left(1+\frac{\varepsilon}{N}\right)^{-1},\left(1+\frac{\varepsilon}{N}\right)^{-2},\cdots,\left(1+\frac{\varepsilon}{N}\right)^{k},0\right\}$, where $k=\left\lceil \frac{\log(1-p_{\max})}{\log\left(1+\frac{\varepsilon}{N}\right)} \right\rceil$.
3. To each link $e\in\widetilde{E}$ that satisfies $\left(1+\frac{\varepsilon}{N}\right)^{-(i+1)} \le 1-p_e \le \left(1+\frac{\varepsilon}{N}\right)^{-i}$, assign a weight $w_e = i.$
4. For each probability restriction that satisfies $\left(1+\frac{\varepsilon}{N}\right)^{-(i+1)} \le \Pi^{(i,j)} \le \left(1+\frac{\varepsilon}{N}\right)^{-i}$, assign a weight restriction $W^{(i,j)} = i+1.$
5. Solve the instance $\left\langle G(V,\widetilde{E}),\ \{c_e\}_{e\in\widetilde{E}},\ \{w_e\}_{e\in\widetilde{E}},\ \left\{W^{(\mathrm{i,j})}\right\}_{(\mathrm{i,j})\in\beta},\left\{\gamma^{(\mathrm{i,j})}\right\}_{(\mathrm{i,j})\in\beta}\right\rangle$ of problem RMP using Algorithm RMP.
6. Construct a path flow f as follows:
   a. Let $h: P^{(i,j)} \to P^{(i,j)}_{\text{simple}}$ map each path $(v_0,v_1,\cdots v_{h-1},v_h,\cdots v_h,v_{h+1},\cdots v_n)\in P^{(i,j)}$ into a simple path $(v_0,v_1,\cdots v_{h-1},v_h,v_{h+1},\cdots v_n)\in P^{(i,j)}_{\text{simple}}$.
   b. For each $p\in P^{(i,j)}_{\text{simple}}$, $f(p) \triangleq \sum_{p'\mid h(p')=p} g(p').$
7. Return the path flow f.

Fig. 4.7 ReMP Approximation Scheme

Theorem 4.4 Given an instance $\left\langle G(V,E),\ \{p_e\}_{e\in E},\ \{c_e\}_{e\in E},\ \left\{\gamma^{(\mathrm{i,j})}\right\}_{(\mathrm{i,j})\in\beta},\ \left\{\Pi^{(\mathrm{i,j})}\right\}_{(\mathrm{i,j})\in\beta}\right\rangle$ of Problem ReMP and an approximation parameter ε, the *ReMP Approximation Scheme* satisfies the following:

a. For each commodity $(i,j)\in\beta$, $\sum_{p\in P^{(i,j)}} f(p)=\gamma^{(i,j)}$ i.e., the flow demand requirement is satisfied for all commodities.
b. The network congestion factor of the output is not larger than the network congestion factor of the optimal solution.
c. If the output assigns to a path $p\in P^{(i,j)}$ a positive flow then $\prod_{e\in p}(1-p_e)\geq\frac{\Pi^{(i,j)}}{(1+\varepsilon)}$; i.e. the restriction on the probability of success is relaxed by at most a factor of $(1+\varepsilon)$.
d. The scheme has a polynomial complexity with respect to the input and the given approximation parameter ε.

Proof

a. In step (5) of the algorithm, an instance $\left\langle G\left(V,\widetilde{E}\right),\ \{c_e\}_{e\in\widetilde{E}},\ \{w_e\}_{e\in\widetilde{E}},\ \left\{W^{(\mathrm{i,j})}\right\}_{(\mathrm{i,j})\in\beta},\left\{\gamma^{(\mathrm{i,j})}\right\}_{(\mathrm{i,j})\in\beta}\right\rangle$ of Problem RMP is solved by Algorithm RMP that satisfies the flow demand requirements for each commodity $(i,j)\in\beta$. Thus, the returned path flow f satisfies the flow demand requirements of all commodities.

b. Let α be the network congestion factor of the resulting path flow and let α^* be the network congestion factor of the optimal solution. We have to prove that $\alpha\leq\alpha^*$. To that end, consider the given instance of Problem ReMP and the corresponding instance of Problem RMP that was specified in step (5) of the ReMP Approximation Scheme. We will prove that the set of feasible paths of instance ReMP is contained in the set of feasible paths of instance RMP. Proving that will show that the constraints over the transformed instance of Problem RMP are relaxed with respect to the original constraints over the given instance of Problem ReMP. To that end, we consider a feasible path p with respect to the given instance of Problem ReMP. We will show that path p is also feasible with respect to the instance of Problem RMP in step (5). Since p is a feasible path with respect to the given instance of Problem ReMP, it satisfies $\prod_{e\in p}(1-p_e)\geq\Pi^{(i,j)}$. By the transformation to problem RMP, each link $e\in E$ with a probability p_e such that $\left(1+\frac{\varepsilon}{N}\right)^{-(i+1)}\leq 1-p_e\leq\left(1+\frac{\varepsilon}{N}\right)^{-i}$ is assigned with a weight value $w_e=i$. Therefore, $(1-p_e)\leq\left(1+\frac{\varepsilon}{N}\right)^{-w_e}$. In addition, each end-to-end reliability

constraint $\Pi^{(i,j)}$ such that $\left(1+\frac{\varepsilon}{N}\right)^{-(i+1)} \le \Pi^{(i,j)} \le \left(1+\frac{\varepsilon}{N}\right)^{-i}$ is transformed into a weight constraint $W^{(i,j)} = i+1$ that results with $\left(1+\frac{\varepsilon}{N}\right)^{-W^{(i,j)}} \le \Pi^{(i,j)}$. Therefore, it follows that $\left(1+\frac{\varepsilon}{N}\right)^{-W^{(i,j)}} \le \Pi^{(i,j)} \le \prod_{e\in p}(1-p_e) \le \left(1+\frac{\varepsilon}{N}\right)^{-\sum_{e\in p} w_e}$. Since $\left(1+\frac{\varepsilon}{N}\right)^{-W^{(i,j)}} \le \left(1+\frac{\varepsilon}{N}\right)^{-\sum_{e\in p} w_e}$, we conclude that $\sum_{e\in p} w_e \le W^{(i,j)}$. Thus, path p is also feasible with respect to the instance of Problem RMP that is specified in step (5).

c. We prove now that, if path $p \in P^{(i,j)}$ is assigned with a positive amount of flow by the ReMP Approximation Scheme, then $\prod_{e\in p}(1-p_e) \ge \frac{\Pi^{(i,j)}}{(1+\varepsilon)}$. Denote the success probability of path p as $\Psi(p)$. Then,

$$\Psi(p) = \prod_{e\in p}(1-p_e) \ge \prod_{e\in p}\left(1+\frac{\varepsilon}{N}\right)^{-(w_e+1)}.$$

Since the solution to Problem RMP consists of simple paths, and since each simple path contains at most $N-1$ links, we conclude that:

$$\Psi(p) \ge \prod_{e\in p}\left(1+\frac{\varepsilon}{N}\right)^{-(w_e+1)} \ge \left(1+\frac{\varepsilon}{N}\right)^{-(W(p)+(N-1))} \ge \left(1+\frac{\varepsilon}{N}\right)^{-W^{(i,j)}} \cdot \left(1+\frac{\varepsilon}{N}\right)^{-(N-1)} \quad (1)$$

.

From the way the algorithm transforms the probability restriction into a weight restriction, we conclude that

$$\left(1+\frac{\varepsilon}{N}\right)^{-W^{(i,j)}} \le \Pi^{(i,j)} \le \left(1+\frac{\varepsilon}{N}\right)^{-W^{(i,j)}+1}. \quad (2)$$

Therefore, from (1) and (2):

$$\Psi(p) \ge \left(1+\frac{\varepsilon}{N}\right)^{-W^{(i,j)}} \cdot \left(1+\frac{\varepsilon}{N}\right)^{-(N-1)} = \left(1+\frac{\varepsilon}{N}\right)^{-W^{(i,j)}+1} \cdot \left(1+\frac{\varepsilon}{N}\right)^{-(N-1)-1} \ge$$

$$\ge \Pi^{(i,j)}\left(1+\frac{\varepsilon}{N}\right)^{-N} = \frac{\Pi^{(i,j)}}{\left(1+\frac{\varepsilon}{N}\right)^{N}} \approx \frac{\Pi^{(i,j)}}{\left(1+N\cdot\frac{\varepsilon}{N}\right)} = \frac{\Pi^{(i,j)}}{(1+\varepsilon)}.$$

d. In order to prove that the complexity of the ReMP Approximation Scheme is polynomial, it suffices to show that the number of variables formulating the problem is polynomial with respect to the input and the approximation

parameter ε [24]. We saw in the proof for Theorem 4.3 that the number of variables formulating any instance of problem RMP is never larger than $2 \cdot M \cdot |\beta| \cdot W^{\max}$, where $W^{\max}$ is the largest weight restriction. Since, by construction, $W^{(i,j)} = i+1 \Leftrightarrow \left(1+\frac{\varepsilon}{N}\right)^{-(i+1)} \le \Pi^{(i,j)} \le \left(1+\frac{\varepsilon}{N}\right)^{-i}$, it follows that $\Pi^{(i,j)} \le \left(1+\frac{\varepsilon}{N}\right)^{-\left(W^{(i,j)}-1\right)}$. Thus, $\log_{1+\frac{\varepsilon}{N}} \Pi^{(i,j)} \le -\left(W^{(i,j)}-1\right)$ $\Rightarrow W^{(i,j)} \le 1 - \log_{1+\frac{\varepsilon}{N}} \Pi^{(i,j)}$. Let $\Pi^{\min} \triangleq \min_{(i,j)\in\beta} \left\{\Pi^{(i,j)}\right\}$. The largest weight restriction, namely $W^{\max}$, must therefore satisfy, $W^{\max} \le 1 - \log_{1+\frac{\varepsilon}{N}} \Pi^{\min} \approx 1 - \frac{N}{\varepsilon}\log\left(\Pi^{\min}\right)$. Therefore, the number of variables is no larger than $2 \cdot M \cdot |\beta| \cdot \left(1 - \frac{N}{\varepsilon}\log\left(\Pi^{\min}\right)\right) = O\left(\frac{N \cdot M \cdot |\beta| \cdot \left|\log\left(\Pi^{\min}\right)\right|}{\varepsilon}\right)$. ■

5. Solution of Problem RDJM

In this section we aim at solving problem RDJM, i.e. the problem of minimizing congestion while restricting the total weight of each path and the delay jitter among all paths. Since Problem RMP is a special case of Problem RDJM, it follows that Problem RDJM is NP-hard. Moreover, we show that even if we remove the weight restriction from Problem RDJM, it remains NP hard i.e., the problem of minimizing the congestion under just a delay jitter restriction is already intractable. To that end, we define Problem Relaxed_RDJM that is a special case of Problem RDJM that sets the weight restrictions of each commodity to infinite i.e., Problem Relaxed_RDJM is a special case of Problem RDJM that has no weight restrictions.

5.1 Intractability of Problem Relaxed_RDJM

We show that problem Relaxed_RDJM can be reduced to problem RMP, which was proven to be NP-hard (Section 4.1).

Theorem 5.1 Problem Relaxed_RDJM is NP hard.

Proof First, consider the single commodity instance of problem Relaxed_RDJM as a decision problem.

Given are a network $G(V,E)$, for each link $e \in E$ a weight $w_e > 0$ and a capacity $c_e > 0$, and, for a commodity $(s,t) \in V \times V$, a demand $\gamma > 0$ and a delay-jitter requirement J. Is there a path flow with network congestion factor of at most α such that, if p_1 and p_2 transfer positive amounts of flow, then $\left|W(p_1) - W(p_2)\right| \le J$?

Let $\left\langle \widetilde{G}\left(\widetilde{V},\widetilde{E}\right),\ \left\{\widetilde{c_e}\right\}_{e\in\widetilde{E}},\ \left\{\widetilde{w_e}\right\}_{e\in\widetilde{E}},\ \left(\tilde{s},\tilde{t}\right),\ \widetilde{W},\ \tilde{\gamma},\tilde{\alpha}\ \right\rangle$ be a single commodity instance of Problem RMP and let $\widetilde{w_{\min}} \triangleq \min_{e\in\widetilde{E}}\left\{\widetilde{w_e}\right\}$. The single commodity instance of Problem RMP was formulated as a decision problem and proven to be NP-hard in section 4.1. We transform RMP to Relaxed_RDJM as follows.

Define a network $G(V,E)$ that has the same nodes, links, capacities and weights as the original network $\widetilde{G}\left(\widetilde{V},\widetilde{E}\right)$. In addition, add a new link $e' = (s,t)$ with a capacity $\sum_{e\in\widetilde{E}} \widetilde{c_e}$, and a weight $\widetilde{w_{\min}}$. The source and target nodes $\left(\tilde{s},\tilde{t}\right)$ as well as the network congestion factor $\tilde{\alpha}$ of RMP, remain unchanged in Relaxed_RDJM. Finally, set the delay-jitter constraint J to be $\widetilde{W} - \widetilde{w_{\min}}$ and the flow demand γ to be $\tilde{\gamma} + \tilde{\alpha}\sum_{e\in\widetilde{E}} \widetilde{c_e}$.

It is easy to see that the transformed Relaxed_RDJM instance can be constructed in polynomial time.

We now show that there is a path flow that satisfies the given instance of problem RMP if and only if there is a path flow that satisfies the transformed instance to problem Relaxed_RDJM.

$\Leftarrow$: Suppose that there is a path flow for the transformed instance $\left\langle G(V,E),\ \{c_e\}_{e\in E},\ \{w_e\}_{e\in E},\ (s,t),\ J,\ \gamma,\alpha \right\rangle$ of problem Relaxed_RDJM and consider the network $G(V,E)$. By construction, removing the link e' from E yields a network $G(V,E-\{e'\})$ that is identical to the original network $\widetilde{G}(\widetilde{V},\widetilde{E})$. We now show that removing link e' and the flow traversing through it, yields a path flow that satisfies the instance $\left\langle \widetilde{G}(\widetilde{V},\widetilde{E}),\ \{\widetilde{c_e}\}_{e\in\widetilde{E}},\ \{\widetilde{w_e}\}_{e\in\widetilde{E}},\ (\tilde{s},\tilde{t}),\ \widetilde{W},\ \tilde{\gamma},\widetilde{\alpha} \right\rangle$ of problem RMP.

Since the capacity of link e' is $\sum_{e\in E-\{e'\}} c_e$ and the network congestion factor is at most α, it follows that link e' could transfer at most $\alpha\cdot\sum_{e\in E-\{e'\}} c_e = \widetilde{\alpha}\cdot\sum_{e\in\widetilde{E}}\widetilde{c_e}$ flow units. Therefore, since the demand of Relaxed_RDJM is $\gamma=\tilde{\gamma}+\widetilde{\alpha}\cdot\sum_{e\in\widetilde{E}}\widetilde{c_e}$, after removing link e' and the flow that was traversing through it, the total flow that traverses through the network is at least $\gamma-\widetilde{\alpha}\cdot\sum_{e\in\widetilde{E}}\widetilde{c_e}=\left(\tilde{\gamma}+\widetilde{\alpha}\cdot\sum_{e\in\widetilde{E}}\widetilde{c_e}\right)-\widetilde{\alpha}\cdot\sum_{e\in\widetilde{E}}\widetilde{c_e}=\tilde{\gamma}$ flow units.

After removing link e', it is obvious that the maximum flow that can traverse through the network without violating the capacity constraints is not larger than $\sum_{e\in E-\{e'\}} c_e$. However, as the network congestion factor is α, the maximum flow that could traverse through the network is not larger than $\alpha\cdot\sum_{e\in E-\{e'\}} c_e$. Since the demand of the instance of Relaxed_RDJM is $\gamma=\tilde{\gamma}+\widetilde{\alpha}\cdot\sum_{e\in\widetilde{E}}\widetilde{c_e}$, and since at most $\alpha\cdot\sum_{e\in E-\{e'\}} c_e$ flow units can traverse through the links in $E-\{e'\}$, at least $\gamma-\alpha\cdot\sum_{e\in E-\{e'\}} c_e$ flow units have traversed through link e'. Since $\gamma-\alpha\cdot\sum_{e\in E-\{e'\}} c_e=\tilde{\gamma}+\widetilde{\alpha}\cdot\sum_{e\in\widetilde{E}}\widetilde{c_e}-\alpha\cdot\sum_{e\in E-\{e'\}} c_e=\tilde{\gamma}+\widetilde{\alpha}\cdot\sum_{e\in\widetilde{E}}\widetilde{c_e}-\widetilde{\alpha}\cdot\sum_{e\in\widetilde{E}}\widetilde{c_e}=\tilde{\gamma}$, and since $\tilde{\gamma}>0$, we conclude that the link e' has transferred a positive amount of flow. Since, by construction, the weight of link e' equals $w_{\min}$, and since each pair of paths that carry a positive amount of flow has a delay jitter of at most J, the maximum weight of a path that carries a positive flow is $J+w_{\min}=\widetilde{W}$.

Therefore, for the network $\widetilde{G}(\widetilde{V},\widetilde{E})$ with weights $\{\widetilde{w_e}\}_{e\in\widetilde{E}}$ and capacities $\{\widetilde{c_e}\}_{e\in\widetilde{E}}$, we identified a path flow of at least $\tilde{\gamma}$ units such that, for each path $p\in P^{(\tilde{s},\tilde{t})}$ that

transfers a positive amount of flow, it holds that $W(p) \le \widetilde{W}$, as required by Problem RMP.

$\Rightarrow$: Suppose that there is a path flow for instance $\left\langle \widetilde{G}(\widetilde{V},\widetilde{E}),\ \{\widetilde{c_e}\}_{e\in\widetilde{E}},\ \{\widetilde{w_e}\}_{e\in\widetilde{E}},\ (\tilde{s},\tilde{t}),\ \widetilde{W},\ \tilde{\gamma},\tilde{\alpha} \right\rangle$ of problem RMP. We have to show that there is a path flow for instance $\left\langle G(V,E),\ \{c_e\}_{e\in E},\ \{w_e\}_{e\in E},\ (s,t),\ J,\ \gamma,\alpha \right\rangle$ of problem Relaxed_RDJM. We construct it as follows:

- Every path p in the network $G(V,E-\{e'\})$ is assigned with the flow of the corresponding path $\tilde{p}$ in network $\widetilde{G}(\widetilde{V},\widetilde{E})$.
- Link e' is assigned with flow $\alpha \cdot \sum_{e\in E-\{e'\}} c_e$.

We now show that the constructed path flow satisfies the requirements of the instance $\left\langle G(V,E),\ \{c_e\}_{e\in E},\ \{w_e\}_{e\in E},\ (s,t),\ J,\ \gamma,\alpha \right\rangle$ of Problem RDJM.

Clearly, the total flow transferred through network $\widetilde{G}(\widetilde{V},\widetilde{E})$ is now $\tilde{\gamma} + \alpha \sum_{e\in E-\{e'\}} c_e = \tilde{\gamma} + \tilde{\alpha}\sum_{e\in\widetilde{E}} \widetilde{c_e} = \gamma$ flow units.

Since the capacity of link e' is $\sum_{e\in E-\{e'\}} c_e$, then the link congestion factor over e' is $\frac{f_{e'}}{c_{e'}} = \frac{\alpha \cdot \sum_{e\in E-\{e'\}} c_e}{\sum_{e\in E-\{e'\}} c_e} = \alpha$. In addition, since the flow and the capacity of each link $e \in E-\{e'\}$ is equal to the flow and the capacity of the corresponding link $\tilde{e} \in \widetilde{E}$ in network $\widetilde{G}(\widetilde{V},\widetilde{E})$, the link congestion factor of every link $e \in E-\{e'\}$ is not larger than $\tilde{\alpha}$ (which equals to α). Therefore, the resulting network congestion factor is at most α.

It is only left to be shown that the constructed path flow satisfies the delay jitter constraints. Since e' carries a positive amount of flow and its weight is $w_{\min}$, the shortest path that transfers a positive amount of flow has a weight of $w_{\min}$. Therefore, as the weight restriction is $\widetilde{W}$, the difference between any two paths that carry positive amounts of flow is no more than $\widetilde{W} - w_{\min}$, which is precisely the delay jitter constraint J.

Thus, Problem Relaxed_RDJM is NP-hard. ■

5.2 Pseudo polynomial algorithm for Problem RDJM

In this section we present a pseudo-polynomial algorithm for the single commodity case of problem RDJM. Based on this result, a practical routing scheme is derived in Sections 5.3 and 5.4.

The pseudo-polynomial algorithm is based on solving the following variant of Problem RMP.

5.2.1 Problem RRMP (Restricted RMP)

Given are a network $G(V,E)$, two nodes $s,t \in V$, for each link $e \in E$ a weight $w_e > 0$ and a capacity $c_e > 0$, a demand $\gamma > 0$, a hop-count restriction $1 \le H \le N-1$ and upper and lower weight restrictions U,L respectively. Find a path flow that minimizes the network congestion factor such that, if $P \subseteq P^{(s,t)}$ is the set of paths in $P^{(s,t)}$ that are assigned with a positive flow, then, for each $p \in P$, it holds that $|p| \le H$ and $L \le D(p) \le U$.

Although Problem RMP doesn't restrict the paths to have an hop count of at most $N-1$, it is obvious that for any given instance of Problem RMP there exists an optimal solution to Problem RMP that consists of only simple paths (indeed, Algorithm RMP returns only solutions that consist of simple paths). Therefore, Problem RMP remains NP-hard even if we add a new restriction that enforces to return only paths with hop count of at most $N-1$. Thus, since this restricted version of Problem RMP is derived from Problem RRMP by setting to zero the lower weight restriction and by setting to $N-1$ the hop count restriction, it follows that Problem RRMP is *NP hard*.

5.2.2 Solving Problem RRMP

In this section we present a linear program for Problem RRMP. The program uses some of the notations introduced in section 4.2. For convenience, we repeat these definitions.

Let α be the network congestion factor. Define $f_e^{\omega,h}$ to be the link flow over link $e=(u,v) \in E$ that has traversed through paths $p \in P^{(s,u)}$ such that $\sum_{e \in p} w_e = \omega$ and $|p| = h$. Finally, for each $v \in V$, denote by $O(v)$ the set of links that emanate from v, and by $I(v)$ the set of links that enter that node, namely $O(v) = \{(v,l) | (v,l) \in E\}$ and $I(v) = \{(w,v) | (w,v) \in E\}$. Then, Problem RRMP can be formulated as the following linear program (Fig. 5.1).

Minimize α

s.t

$$\sum_{e\in O(v)} f_e^{\omega,h} - \sum_{e\in I(v)} f_e^{\omega-w_e,h-1} = 0 \qquad ,\forall v \in V-\{s,t\},\ \forall \omega \in [0,U],\ \forall h \in [0,H] \quad (1)$$

$$\sum_{e\in O(s)} f_e^{\omega,h} - \sum_{e\in I(s)} f_e^{\omega-w_e,h-1} = 0 \qquad ,\forall \omega \in [1,U] \quad \forall h \in [1,H] \quad (2)$$

$$\sum_{e\in O(s)} f_e^{0,0} = \gamma \quad (3)$$

$$\sum_{\omega=0}^{U}\sum_{h=0}^{l} f_e^{\omega,h} \le c_e \cdot \alpha \qquad ,\forall e \in E \quad (4)$$

$$f_e^{\omega,h} = 0 \qquad ,\forall e \in E,\ \omega < 0 \text{ or } h < 0 \quad (5)$$

$$f_e^{\omega,h} \ge 0 \qquad ,\forall \omega \in [0,U],\ \forall h \in [0,H],\ \forall e \in E \quad (6)$$

$$f_e^{\omega,h} = 0 \qquad ,\forall e \in I(t),\ \forall \omega \notin [L,U],\ \forall h \in [0,H] \quad (7)$$

$$\alpha \ge 0 \quad (8)$$

Fig. 5.1: Program RRMP

Except for constraint (7), Program RRMP is quite similar to the one presented for Problem RMP in section 4.2. Constraint (7) restricts the traffic that enters the target node *t* to traverse only paths of total weight not larger than U and not smaller than L.

The path flow is then constructed from the output $\{f_e^{\omega,h}\}$ by employing Algorithm PFC_RDJM that is similar to Algorithm PFC that was described in Section 4. The only difference between the algorithms is the specification of the procedure that constructs in each iteration a path from *s* to *t*. This procedure incorporates to Procedure Path Construction the hop-count restriction. Algorithm PFC_RDJM is defined in the Appendix.

Algorithm RRMP$(\mathrm{G}(\mathrm{V},\mathrm{E}),\{s,t\},\{w_e\},\{c_e\},\gamma,L,U,H)$

1. $\widetilde{E} \leftarrow \{e \in E \,|\, w_e \le U\}$.
2. $\{f_e^{\omega,h}\} \leftarrow$ **RRMP** $(\mathrm{G}(\mathrm{V},\widetilde{\mathrm{E}}),\{s,t\},\{w_e\},\{c_e\},\gamma,L,U,H)$
3. $f \leftarrow$ **PFC_RDJM** $(G(V,\widetilde{E}),\{s,t\},\{f_e^{\omega,h}\},\gamma)$
4. **Return** path flow *f*.

Fig. 5.2: Algorithm RRMP

5.2.3 Solving Problem RDJM

We use Algorithm RRMP in order to establish an algorithm for Problem RDJM as follows. Given an instance $\langle G(V,E),\{s,t\},\{w_e\},\{c_e\},\gamma,J,H \rangle$ of problem RDJM,

we solve all the corresponding instances of Problem RRMP that have an upper weight restriction U and a lower weight restriction L such that $U - L = J$. More precisely, we solve for each $L \in \left[0, \min\{H \cdot w_{\max}, W\} - J\right]$ an instance $\left(\mathrm{G}(\mathrm{V},\mathrm{E}), \{s,t\}, \{w_e\}, \{c_e\}, \gamma, L, L+J, H\right)$ of Problem RRMP using Algorihm RRMP. Finally, the output of the algorithm is determined to be the solution of the instance that has the minimal network congestion factor. The formal description of the algorithm is described in Fig. 5.3.

Algorithm RDJM $\left(G(V,E), \{s,t\}, \{c_e\}, \{w_e\}, J, W, H, \gamma\right)$

Parameters :

$G(V,E)$ – network

$\{c_e\}_{e \in E}$ – capacities

$\{w_e\}_{e \in E}$ – weights

(s,t) – commodity

J – delay jitter restriction

W – weight restriction

γ – demand

Variables

$\alpha_{\min}$ – network congestion factor

g – path flow

Initialization:

$\alpha_{\min} \leftarrow \infty$.

$g \leftarrow NIL$.

IF $\min\{H \cdot w_{\max}, W\} < J$ **Then**

- Apply Algorithm RRMP in order to solve the instance $\left(\mathrm{G}(\mathrm{V},\mathrm{E}), \{s,t\}, \{w_e\}, \{c_e\}, \gamma, L = 0, U = \min\{H \cdot w_{\max}, W\}, H\right)$ of problem RRMP. Let α be the network congestion factor of path flow f.

ELSE

For each $L \in \left[0, \min\{H \cdot w_{\max}, W\} - J\right]$:

- Apply Algorithm RRMP in order to solve the instance $\left(\mathrm{G}(\mathrm{V},\mathrm{E}), \{s,t\}, \{w_e\}, \{c_e\}, \gamma, L, L+J, H\right)$ of problem RRMP. Let α be the network congestion factor of path flow f.

- If α is smaller than $\alpha_{\min}$, perform:
  - $\alpha_{\min} \leftarrow \alpha$.
  - $f \leftarrow g$.

Return path flow g.

Fig. 5.3: Algorithm RDJM

Theorem 5.2 Given an instance $\langle G(V,E),\{s,t\},\{c_e\},\{w_e\},J,W,H,\gamma\rangle$ of Problem RDJM, Algorithm RDJM finds a path flow g that minimizes the network congestion factor such that, if $P \subseteq P^{(s,t)}$ is the set of paths in $P^{(s,t)}$ that are assigned with a positive flow, then, for each $p_1, p_2 \in P$, it holds that $|p_1|,|p_2| \le H$, $W(p_1),W(p_2) \le W$, and $|W(p_1)-W(p_2)| \le J$.

Proof It is clear that, on every iteration of Algorithm RDJM, the solution to instance $(\mathrm{G}(\mathrm{V},\mathrm{E}),\{s,t\},\{w_e\},\{c_e\},\gamma,L,L+J,H)$ of problem RRMP transfers γ flow units from s to t while satisfying the delay jitter constraint *J*, the hop count restriction H, and the weight restriction *W*. In addition, since $L \in [0, \min\{H \cdot w_{\max}, W\} - J]$ it follows that the upper weight restriction $U = L + J$ is at most *W*. We need only prove that the path flow g minimizes the network congestion factor. This is quite immediate, but we present the details, for completeness. Algorithm RDJM examines all possible instances of problem RRMP whose solution is feasible with respect to the given instance of Problem RDJM. Since both optimization problems have the same objective function, and since we output the path flow that results with the *smallest* network congestion factor among all examined instances of problem RRMP, we get a feasible path flow for instance $\langle G(V,E),\{s,t\},\{c_e\},\{w_e\},J,W,\gamma\rangle$ that has a minimal network congestion factor. ■

5.3 Approximation Scheme for Problem RDJM

We now present an approximation scheme for problem RDJM for the case that $w_{\max} = O(J)$, where J is the restriction on the delay jitter. We note that this assumption is usually not restrictive. Indeed, in practice the bound on the delay jitter for multipath routing schemes is set to three times the serialization time, which is usually in the order of milliseconds [6]. Since the total delay of a link is also in the order of milliseconds, the assumption that both delays have the same order of magnitude is reasonable.

Since the complexity of Algorithm RDJM depends on the weight of the links, it is possible to reduce its complexity by scaling the weight of each link down into smaller integral values. Obviously, the delay jitter restriction must be scaled as well. However, in addition to its scaling, it is also incremented by the value of the hop count restriction *H*. This is done in order to ensure that the new delay jitter restriction will be relaxed with respect to the original delay jitter restriction. Thus, the resulting network congestion factor will not be larger then the optimal network congestion factor, and the delay jitter restriction will be broken by at most a factor of $(1+\varepsilon)$. Fig. 5.4 describes the approximation scheme for Problem RDJM.

RDJM Approximation Scheme $\langle G(V,E),\{s,t\},\{c_e\},\{w_e\},J,W,H,\gamma,\varepsilon\rangle$

Parameters:

$G(V,E)$ – network

$\{c_e\}_{e\in E}$ – capacities

$\{w_e\}_{e\in E}$ – weights

J – delay jitter restriction

W – weight restriction

H – hop count

γ – demand

ε – approximation parameter

1 $\delta \leftarrow \dfrac{\min\{J,W\}\cdot\varepsilon}{2\cdot N}$

2 $\widetilde{W} \leftarrow \left\lceil \dfrac{W}{\delta} \right\rceil$, $\widetilde{J} \leftarrow \begin{cases} \left\lceil \dfrac{J}{\delta} \right\rceil + H & J \le W \\ \left\lceil \dfrac{W}{\delta} \right\rceil + H & J > W \end{cases}$

3 For each $e \in E$:

$\widetilde{w_e} \leftarrow \left\lfloor \dfrac{w_e}{\delta} \right\rfloor$.

4 Solve instance $\left\langle G(V,E),\{s,t\},\{c_e\}_{e\in E},\{\widetilde{w_e}\}_{e\in E},\widetilde{J},\widetilde{W},H,\gamma \right\rangle$ of problem RDJM using Algorithm RDJM, and return the resulting path flow.

Fig 5.4: RDJM Approximation Scheme

Theorem 5.3: Given are an instance $\langle G(V,E),\{s,t\},\{c_e\},\{w_e\},J,W,H,\gamma\rangle$ of problem RDJM and an approximation parameter ε. The output of the *RDJM Approximation Scheme* is a path flow *f* that satisfies the following:

a. $\sum_{p\in P^{(s,t)}} f(p) = \gamma$, i.e., the flow demand requirement is satisfied.

b. If α^* is the network congestion factor of the optimal solution then, for each $e\in E$, it holds that $\sum_{p\in P^{(s,t)}} \Delta_e(p)\cdot f(p) \le \alpha^*\cdot c_e$ i.e., the network congestion factor is at most α^*.

c. For each pair of paths $p_1,p_2\in P^{(s,t)}$, that transfer a positive amount of flow, it holds that $|p_1|,|p_2|\le H$, $D(p_1),D(p_2)\le D\cdot(1+\varepsilon)$ and $|D(p_1)-D(p_2)|\le J\cdot(1+\varepsilon)$ i.e., the hop count restriction is satisfied and both the end-to-end delay and the delay-jitter restrictions are violated by a factor of at most $(1+\varepsilon)$.

Proof

a. By construction, the output of the RDJM Approximation Scheme (Fig. 5.4) is the output of Algorithm RDJM over an instance $\left\langle G(V,E),\{s,t\},\{c_e\},\{\widetilde{w}_e\},\widetilde{J},\widetilde{W},H,\gamma\right\rangle$. Since Algorithm RDJM returns a path flow *f* that transfer γ flow units from *s* to *t*, it follows that $\sum_{p\in P^{(s,t)}} f(p)=\gamma$.

b. Assume that $W\geq J$. Let α be the network congestion factor of the output and let α^* be the network congestion factor of the optimal solution. We have to show that $\alpha\leq\alpha^*$. To that end, we prove that the restrictions over the instance $\left\langle G(V,E),\{s,t\},\{c_e\},\{\widetilde{w}_e\},\widetilde{J},\widetilde{W},H,\gamma\right\rangle$ of line (4), are *relaxed* with respect to the original restrictions of the given instance $\left\langle G(V,E),\{s,t\},\{c_e\},\{w_e\},J,W,H,\gamma\right\rangle$ of Problem RDJM. Since the Algorithm rounds down the weight of each link, and round up the weight restriction, it is easy to see that this restriction is relaxed with respect to the original restriction. We turn to consider the delay jitter restrictions. To that end, consider a pair of paths $p_1,p_2\subseteq E$ that satisfy $\left|W(p_1)-W(p_2)\right|\leq J$ with respect to the weights $\{w_e\}$. We will prove that $\left|\widetilde{W(p_1)}-\widetilde{W(p_2)}\right|\leq\widetilde{J}$ with respect to the weights $\{\widetilde{w}_e\}$. In other words, we have to prove that $-\widetilde{J}\leq\widetilde{W(p_1)}-\widetilde{W(p_2)}\leq\widetilde{J}$ for the delays $\{\widetilde{w}_e\}$.Without loss of generality, assume that $W(p_1)\geq W(p_2)$. Therefore, (i) $0\leq W(p_1)-W(p_2)\leq J$. In order to prove that $\left|\widetilde{W(p_1)}-\widetilde{W(p_2)}\right|\leq\widetilde{J}$ we first use the left hand side of (i) in order to verify that $-\widetilde{J}\leq\widetilde{W(p_1)}-\widetilde{W(p_2)}$.

$$W(p_1)-W(p_2)\geq 0 \;\Rightarrow\; \frac{1}{\delta}\left(W(p_1)-W(p_2)\right)\geq 0 \;\Rightarrow\; \frac{1}{\delta}\left(\sum_{e\in p_1}w_e-\sum_{e\in p_2}w_e\right)\geq 0 \;\Rightarrow$$

$$\Rightarrow\; \sum_{e\in p_1}\frac{w_e}{\delta}-\sum_{e\in p_2}\frac{w_e}{\delta}\geq 0 \;\Rightarrow\; \left\lfloor\sum_{e\in p_1}\frac{w_e}{\delta}-\sum_{e\in p_2}\frac{w_e}{\delta}\right\rfloor\geq 0 \;\Rightarrow\; \left\lceil\sum_{e\in p_1}\frac{w_e}{\delta}\right\rceil-\left\lfloor\sum_{e\in p_2}\frac{w_e}{\delta}\right\rfloor\geq 0 \;\Rightarrow$$

$$\Rightarrow\; \sum_{e\in p_1}\left\lceil\frac{w_e}{\delta}\right\rceil-\sum_{e\in p_2}\left\lfloor\frac{w_e}{\delta}\right\rfloor\geq 0 \;\Rightarrow\; \sum_{e\in p_1}\left(\left\lfloor\frac{w_e}{\delta}\right\rfloor+1\right)-\sum_{e\in p_2}\left\lfloor\frac{w_e}{\delta}\right\rfloor\geq 0 \;\Rightarrow$$

$$\Rightarrow\; \sum_{e\in p_1}\left\lfloor\frac{w_e}{\delta}\right\rfloor+|p_1|-\sum_{e\in p_2}\left\lfloor\frac{w_e}{\delta}\right\rfloor\geq 0 \;\Rightarrow\; \sum_{e\in p_1}\left\lfloor\frac{w_e}{\delta}\right\rfloor+H-\sum_{e\in p_2}\left\lfloor\frac{w_e}{\delta}\right\rfloor\geq 0 \;\Rightarrow$$

$$\Rightarrow\; \sum_{e\in p_1}\left\lfloor\frac{w_e}{\delta}\right\rfloor-\sum_{e\in p_2}\left\lfloor\frac{w_e}{\delta}\right\rfloor\geq -H \;\Rightarrow\; \sum_{e\in p_1}\left\lfloor\frac{w_e}{\delta}\right\rfloor-\sum_{e\in p_2}\left\lfloor\frac{w_e}{\delta}\right\rfloor\geq -H-\left\lceil\frac{J}{\delta}\right\rceil \;\Rightarrow$$

$$\Rightarrow\; \widetilde{W(p_1)}-\widetilde{W(p_2)}\geq -\left(\left\lceil\frac{J}{\delta}\right\rceil+H\right) \;\Rightarrow\; \widetilde{W(p_1)}-\widetilde{W(p_2)}\geq -\widetilde{J}.$$

We now use the right hand side of (i) namely $W(p_1)-W(p_2)\le J$, in order to prove that $\widetilde{W(p_1)}-\widetilde{W(p_2)}\le \tilde{J}$.

$$W(p_1)-W(p_2)\le J \Rightarrow \frac{1}{\delta}\left(W(p_1)-W(p_2)\right)\le \frac{J}{\delta} \Rightarrow \frac{1}{\delta}\left(\sum_{e\in p_1} w_e-\sum_{e\in p_2} w_e\right)\le \frac{J}{\delta}\Rightarrow$$

$$\Rightarrow \sum_{e\in p_1}\frac{w_e}{\delta}\le \frac{J}{\delta}+\sum_{e\in p_2}\frac{w_e}{\delta} \Rightarrow \left\lfloor \sum_{e\in p_1}\frac{w_e}{\delta}\right\rfloor \le \left\lfloor \frac{J}{\delta}+\sum_{e\in p_2}\frac{w_e}{\delta}\right\rfloor \le \left\lfloor \left\lceil \frac{J}{\delta}\right\rceil +\sum_{e\in p_2}\frac{w_e}{\delta}\right\rfloor =$$

$$=\left\lceil \frac{J}{\delta}\right\rceil +\left\lfloor \sum_{e\in p_2}\frac{w_e}{\delta}\right\rfloor \Rightarrow \left\lfloor \sum_{e\in p_1}\frac{w_e}{\delta}\right\rfloor -\left\lfloor \sum_{e\in p_2}\frac{w_e}{\delta}\right\rfloor \le \left\lceil \frac{J}{\delta}\right\rceil \Rightarrow$$

$$\Rightarrow \sum_{e\in p_1}\left\lfloor \frac{w_e}{\delta}\right\rfloor -\left\lfloor \sum_{e\in p_2}\frac{w_e}{\delta}\right\rfloor \le \left\lceil \frac{J}{\delta}\right\rceil \Rightarrow \sum_{e\in p_1}\left\lfloor \frac{w_e}{\delta}\right\rfloor -\sum_{e\in p_2}\frac{w_e}{\delta}\le \left\lceil \frac{J}{\delta}\right\rceil \Rightarrow$$

$$\Rightarrow \sum_{e\in p_1}\left\lfloor \frac{w_e}{\delta}\right\rfloor -\sum_{e\in p_2}\left(\left\lfloor \frac{w_e}{\delta}\right\rfloor +1\right)\le \left\lceil \frac{J}{\delta}\right\rceil \Rightarrow \sum_{e\in p_1}\left\lfloor \frac{w_e}{\delta}\right\rfloor -\sum_{e\in p_2}\left\lfloor \frac{w_e}{\delta}\right\rfloor -|p_2|\le \left\lceil \frac{J}{\delta}\right\rceil \Rightarrow$$

$$\Rightarrow \sum_{e\in p_1}\left\lfloor \frac{w_e}{\delta}\right\rfloor -\sum_{e\in p_2}\left\lfloor \frac{w_e}{\delta}\right\rfloor \le \left\lceil \frac{J}{\delta}\right\rceil +|p_2| \Rightarrow \sum_{e\in p_1}\left\lfloor \frac{w_e}{\delta}\right\rfloor -\sum_{e\in p_2}\left\lfloor \frac{w_e}{\delta}\right\rfloor \le \left\lceil \frac{J}{\delta}\right\rceil +H \Rightarrow$$

$$\Rightarrow \sum_{e\in p_1}\widetilde{w_e}-\sum_{e\in p_2}\widetilde{w_e}\le \tilde{J} \Rightarrow \widetilde{W(p_1)}-\widetilde{W(p_2)}\le \tilde{J}.$$

Thus, $\alpha\le\alpha^*$ for the case that $W\ge J$.

Assume now that $W<J$. Since all paths have positive weights, it follows that all feasible paths that satisfy the weight and the delay jitter restrictions must have weight in the range [0,*W*]. Thus, the optimal network congestion factor of the given instance $\langle G(V,E),\{s,t\},\{c_e\},\{w_e\},J,W,H,\gamma\rangle$ is equal to the optimal network congestion factor of the instance $\langle G(V,E),\{s,t\},\{c_e\},\{w_e\},\tilde{J},W,H,\gamma\rangle$, where $\tilde{J}=W$. Since by construction the algorithm solves such an instance, and since we have just proved that the for the case that $\tilde{J}=W$ the algorithm outputs a path flow with a network congestion factor that is at most the optimum, it follows that the algorithm outputs a path flow with a network congestion factor at most α^* for the case that $W<J$.

c. Since the instance $\langle G(V,E),\{s,t\},\{c_e\}_{e\in E},\{\widetilde{w_e}\}_{e\in E},\tilde{J},\widetilde{W},H,\gamma\rangle$ that was specified in line (4) has the same hop count restrictions as the original instance $\langle G(V,E),\{s,t\},\{c_e\},\{w_e\},J,W,H,\gamma\rangle$, it is obvious that $|p_1|\le H$, $|p_2|\le H$. It is left to be shown that $|W(p_1)-W(p_2)|\le J\cdot(1+\varepsilon)$ and $W(p_1),W(p_2)\le W\cdot(1+\varepsilon)$.

Without loss of generality, assume that $W(p_1) \geq W(p_2)$. Therefore,

$$W(p_1) - W(p_2) = \sum_{e \in p_1} w_e - \sum_{e \in p_2} w_e \leq \sum_{e \in p_1} \left\lceil \frac{w_e}{\delta} \right\rceil \cdot \delta - \sum_{e \in p_2} \left\lfloor \frac{w_e}{\delta} \right\rfloor \cdot \delta \leq$$

$$\leq \sum_{e \in p_1} \left(\left\lfloor \frac{w_e}{\delta} \right\rfloor + 1 \right) \cdot \delta - \sum_{e \in p_2} \left\lfloor \frac{w_e}{\delta} \right\rfloor \cdot \delta = \sum_{e \in p_1} \left(\widetilde{w_e} + 1 \right) \cdot \delta - \sum_{e \in p_2} \widetilde{w_e} \cdot \delta =$$

$$= \left(\sum_{e \in p_1} \widetilde{w_e} - \sum_{e \in p_2} \widetilde{w_e} \right) \cdot \delta + |p_1| \cdot \delta.$$

Since p_1 and p_2 transfer positive amounts of flow, they satisfy the delay jitter constraint of instance $\left\langle G(V,E), \{s,t\}, \{c_e\}_{e \in E}, \left\{\widetilde{w_e}\right\}_{e \in E}, \widetilde{J}, \widetilde{W}, H, \gamma \right\rangle$ of line (4). Therefore,

$$W(p_1) - W(p_2) \leq \left(\sum_{e \in p_1} \widetilde{w_e} - \sum_{e \in p_2} \widetilde{w_e} \right) \cdot \delta + |p_1| \cdot \delta \leq \widetilde{J} \cdot \delta + |p_1| \cdot \delta \leq \widetilde{J} \cdot \delta + H \cdot \delta.$$

Since $\delta = \dfrac{\min\{J, W\} \cdot \varepsilon}{2 \cdot N}$ and Problem RDJM defines the hop count restriction H to be in the range $1 \leq H \leq N-1$, we conclude that

$$W(p_1) - W(p_2) \leq \widetilde{J} \cdot \delta + H \cdot \delta \leq \left(\left\lceil \frac{J}{\delta} \right\rceil + H \right) \cdot \delta + (N-1) \cdot \delta \leq$$

$$\leq \left(\frac{J}{\delta} + 1 + H \right) \cdot \delta + (N-1) \cdot \delta = J + (N+H) \cdot \delta \leq J + 2 \cdot N \cdot \frac{J \cdot \varepsilon}{2 \cdot N} = J(1+\varepsilon). \quad \blacksquare$$

Theorem 5.4: Given are an instance $\left\langle G(V,E), \{s,t\}, \{c_e\}, \{w_e\}, J, W, H, \gamma \right\rangle$ of problem RDJM and an approximation parameter ε. For $w_{\max} = O(J)$, the RDJM Approximation Scheme has a polynomial complexity with respect to the input and the approximation parameter ε

Proof The complexity of the RDJM Approximation Scheme is determined by step (4). The complexity of step (4) is determined by Algorithm RRMP that executes Program RRMP for each $L \in \left[0, \min\left\{ H \cdot \widetilde{w_{\max}}, \widetilde{W} \right\} - \widetilde{J} \right]$. Therefore we only have to prove that the complexity of Program RRMP is polynomial, and that the total number of iterations in Algorithm RRMP is polynomial (i.e. $\min\left\{ H \cdot \widetilde{w_{\max}}, \widetilde{W} \right\} - \widetilde{J}$ is polynomial with respect to the input and the approximation parameter ε).

In order to show that Program RRMP is polynomial, we need to prove that the program contains a polynomial number of variables [24]. From similar considerations to those brought in the proof to Theorem 4.3, it is easy to see that the number of variables $\left\{ f_e^{\omega,h} \right\}$ formulating Program RRMP in each iteration of Algorithm RRMP is at most $2 \cdot M \cdot U \cdot H$. In addition, Since for the instance $\left\langle G(V,E), \{s,t\}, \{c_e\}_{e \in E}, \left\{\widetilde{w_e}\right\}_{e \in E}, \widetilde{J}, \widetilde{W}, H, \gamma \right\rangle$ of step (4), Algorithm RDJM set a value

for *U*, that satisfies $U \le \min\left\{H \cdot \widetilde{w_{\max}}, \widetilde{W}\right\}$, it follows that the total number of variables in Program RRMP is at most (i) $2 \cdot M \cdot \left(H \cdot \widetilde{w_{\max}}\right) \cdot H \le 2 \cdot M \cdot H^2 \cdot \widetilde{w_{\max}}$. Thus, we conclude that the total number of variables is at most $2 \cdot M \cdot H^2 \cdot \widetilde{w_{\max}} \le 2 \cdot M \cdot N^2 \cdot \widetilde{w_{\max}} = 2 \cdot M \cdot N^2 \cdot \left\lfloor \frac{w_{\max}}{\delta} \right\rfloor = 2 \cdot M \cdot N^2 \cdot \left\lfloor \frac{w_{\max}}{\frac{\min\{J,W\} \cdot \varepsilon}{2 \cdot N}} \right\rfloor$.

Consider the case of $W \ge J$. Since $w_{\max} = O(J)$, we conclude that the number of variables that are formulating Program RRMP is at most

$$2 \cdot M \cdot N^2 \cdot \left\lfloor \frac{w_{\max}}{\frac{\min\{J,W\} \cdot \varepsilon}{2 \cdot N}} \right\rfloor = 2 \cdot M \cdot N^2 \left\lfloor \frac{w_{\max}}{\frac{J \cdot \varepsilon}{2 \cdot N}} \right\rfloor \le \frac{4 \cdot M \cdot N^3 \cdot w_{\max}}{J \cdot \varepsilon} = O\left(\frac{M \cdot N^3}{\varepsilon}\right).$$

Consider the case of $W < J$. Since Algorithm RRMP executes Program RRMP over graph $\mathrm{G}\left(\mathrm{V}, \widetilde{\mathrm{E}}\right)$ that contains only links $e \in E$ that satisfy $w_e \le W$ it follows that

$$2 \cdot M \cdot N^2 \cdot \left\lfloor \frac{w_{\max}}{\frac{\min\{J,W\} \cdot \varepsilon}{2 \cdot N}} \right\rfloor = 2 \cdot M \cdot N^2 \left\lfloor \frac{w_{\max}}{\frac{W \cdot \varepsilon}{2 \cdot N}} \right\rfloor \le 2 \cdot M \cdot N^2 \left\lfloor \frac{w_{\max}}{\frac{w_{\max} \cdot \varepsilon}{2 \cdot N}} \right\rfloor \le \frac{4 \cdot M \cdot N^3}{\varepsilon} = O\left(\frac{M \cdot N^3}{\varepsilon}\right)$$

Thus, Program RRMP is polynomial.

In order to show that $\min\left\{H \cdot \widetilde{w_{\max}}, \widetilde{W}\right\} - \widetilde{J}$ is polynomial with respect to the input and the approximation parameter ε, it is sufficient to prove that $H \cdot \widetilde{w_{\max}} - \widetilde{J}$ is polynomial. Therefore it is sufficient to show that $H \cdot \widetilde{w_{\max}}$ is polynomial with respect to the input and ε. However since we have already established in (i) that the number of variables is at most $2 \cdot M \cdot H^2 \cdot \widetilde{w_{\max}}$, and we show that this number is polynomial, it follows that $H \cdot \widetilde{w_{\max}}$ must also be polynomial in the input and the approximation parameter ε.

Therefore, the RDJM Approximation Scheme has a polynomial complexity with respect to the input and the given approximation parameter ε. ■

5.4 Converting Non-Simple Paths into Simple Paths

As was already explained in Sec. 2 the optimal solution to Problem RDJM may use non-simple paths in order to transfer the flow demand from the source *s* to the destination *t*. In the definition of Problem RDJM we used the hop count restriction in order to control this phenomenon. In this section we present a simple approach that completely avoids this phenomenon using a bounded buffer at the source node. To that end, we transform each non-simple path in the solution of Problem RDJM into a simple path. Then, we identify the delay difference δ between the delay of the non-simple path and the delay of the simple path. Finally, we assign the traffic to the simple path but first delay it for δ units of time using a buffer at the source node.

Clearly, for each path in the solution to Problem RDJM, the delay difference δ is always smaller than the weight (delay) restriction *W*. In addition, each path that transfers $f(p)$ flow units and is delayed by δ time units consumes $f(p) \cdot \delta$ units of

buffer space. Thus the buffer size at the source node is at most $\sum_{p \in P} f(p) \cdot \delta \leq \sum_{p \in P} f(p) \cdot W = W \cdot \sum_{p \in P} f(p) = \gamma \cdot W$. As TCP requires *at least* one bandwidth-delay product of buffer space at each end of the connection [42], such buffer space requirements are reasonable.

6. Solution of Problem KPR

In this section we aim at solving Problem KPR, i.e. the problem of minimizing congestion subject to a restriction on the number of routing paths. In [10] the problem of maximizing the flow subject to a restriction on the number of routing paths was proven to be NP-hard. Since we establish in Theorem 2.1 the equivalence between these two objective functions it follows that Problem KPR is NP-hard. Therefore, we present an efficient approximation scheme for the problem. We begin by introducing the following auxiliary problem.

6.1 Problem σ-IR (σ- Integral Routing)

In this section we formulate Problem σ-IR and solve it. We begin with the following definition.

Definition 6.1 Given are a network $G(V,E)$, for each link $e \in E$ a capacity $c_e > 0$, and, for each commodity $(i,j) \in V \times V$, a demand $\gamma^{(i,j)}$ and a value $\sigma^{(i,j)} \in \mathbb{R}^+$. The path flow of commodity (i,j) is said to be $\sigma^{(i,j)}$*-integral*, if every path $p \in P^{(i,j)}$ is assigned with a flow value that is integral in $\sigma^{(i,j)}$.

Problem σ-IR (σ- Integral Routing) Given are a network $G(V,E)$, for each link $e \in E$ a capacity $c_e > 0$ and for each commodity $(i,j) \in V \times V$, a demand $\gamma^{(i,j)}$ and a value $\sigma^{(i,j)} \in \mathbb{R}^+$. For each commodity $(i,j) \in V \times V$, find a $\sigma^{(i,j)}$-integral path flow that satisfies the demand $\gamma^{(i,j)}$ and minimizes the network congestion factor.

We shall establish a solution to Problem σ-IR for the case of a single commodity. To that end, we first show that the optimal network congestion factor is always an element of a predefined finite set.

6.1.1 What are the possible network congestion factors?

In this subsection we observe that the optimal network congestion factor must belong to a set that consist of at most $M \cdot \frac{\gamma}{\sigma}$ elements.

Definition 6.2 Given an instance $\langle G(V,E), \{s,t\}, \{c_e\}, \gamma, \sigma \rangle$ of Problem σ-IR, We say that the given instance has a *feasible solution*, if network $G(V,E)$ consists of a directed path from source *s* to target *t*.

We note that a given instance may have a feasible solution even in the case where it is not possible to transfer the entire flow demand without violating the capacities constraints. In that case the optimal network congestion factor will have a value that is *larger than* 1.

Theorem 6.1 Given an instance $\langle G(V,E),\{s,t\},\{c_e\},\gamma,\sigma\rangle$ of Problem σ-IR that has a feasible solution. The optimal network congestion factor must be a member in set $\left\{\frac{i\cdot\sigma}{c_e}\middle| e\in E, i\in\left[0,\frac{\gamma}{\sigma}\right], i\in\mathbb{Z}\right\}$.

Proof Let *F* be the set of all σ-integral flow vectors that transfer γ flow units from *s* to *t*, and let $\overline{\alpha}$ be its corresponding set of network congestion factors i.e., $\overline{\alpha}=\left\{\max_{e\in E}\left\{\frac{f_e}{c_e}\right\}\middle|\{f_e\}\in F\right\}$. Since the given instance has a feasible solution, it follows that $F\neq\phi$, and therefore $\overline{\alpha}\neq\phi$ .By the definition of $\overline{\alpha}$, it follows that for each $\alpha\in\overline{\alpha}$, there is a flow vector $\{f_e\}\in F$ and a link $e\in E$ such that $f_e=\alpha\cdot c_e$. On the other hand, since all flow vectors in *F* are σ-integral and transfer at most γ flow units over each link, it follows that for each $\{f_e\}\in F$ and for each $e\in E$, $\frac{f_e}{\sigma}\in\mathbb{Z}$ and $f_e\leq\gamma$. Thus, combining both arguments, it follows that, for each $\alpha\in\overline{\alpha}$ there is a link $e\in E$ such that $\frac{\alpha\cdot c_e}{\sigma}\in\mathbb{Z}$ and $\alpha\cdot c_e\leq\gamma$. In other words, for each $\alpha\in\overline{\alpha}$ there is a link $e\in E$, and an integer $i\in\mathbb{Z}$ such that $\alpha=\frac{i\cdot\sigma}{c_e}$ and $\alpha\leq\frac{\gamma}{c_e}$. Finally, since $\alpha\leq\frac{\gamma}{c_e}$, it follows that $\frac{i\cdot\sigma}{c_e}\leq\frac{\gamma}{c_e}$ and therefore $i\leq\frac{\gamma}{\sigma}$. Thus, we conclude that for each $\alpha\in\overline{\alpha}$ there is a link $e\in E$, and an integer $i\leq\frac{\gamma}{\sigma}$ such that $\alpha=\frac{i\cdot\sigma}{c_e}$. Therefore, $\overline{\alpha}=\left\{\frac{i\cdot\sigma}{c_e}\middle| e\in E, i\in\left[0,\frac{\gamma}{\sigma}\right], i\in\mathbb{Z}\right\}$.

Since the set $\overline{\alpha}$ contains the network congestion factors that corresponds to all σ-integral flow vectors that transfer γ flow units from *s* to *t*, it follows that the optimal network congestion factor, denoted by α^*, must also be an element in that set, i.e., $\alpha^*\in\overline{\alpha}$. ■

We henceforth use $\overline{\alpha}$ in order to denote the set $\left\{\frac{i\cdot\sigma}{c_e}\middle| e\in E, i\in\left[0,\frac{\gamma}{\sigma}\right], i\in\mathbb{Z}\right\}$. Observe that $\left|\overline{\alpha}\right|\leq M\cdot\frac{\gamma}{\sigma}$. In addition, note that the set $\overline{\alpha}$ may contain values that are larger than 1. However, since the optimal solution of Problem σ-IR has the minimum network congestion factor, these infeasible values will be considered only when there are no feasible solutions for the given instance.

6.1.2 Procedure Test

Before solving Problem σ-IR, we introduce *Procedure Test* that is specified in Fig. 6.1. This procedure is given an instance $\langle G(V,E),\{s,t\},\{c_e\},\gamma,\sigma\rangle$ of Problem σ-IR and a congestion restriction α. If there exists a solution to the given instance such that the network congestion factor is at most α, then the procedure returns it. Otherwise, the procedure returns Fail.

We explain now the main idea behind Procedure Test. Initially, the procedure multiplies all link capacities by a factor of α in order to satisfy the restriction on the network congestion factor. Then it rounds down the capacity of each link to a multiple of σ, and applies any standard Maximum Flow Algorithm over the resulting network. Since all link capacities are σ-integral, the maximum flow algorithm determines a σ-integral link flow that transfers a maximum flow demand. If this link flow transfers from *s* to *t* at least γ flow units, then the procedure returns it. Otherwise the procedure fails. The formal description of the procedure is specified in Fig. 6.1.

Procedure Test $(G,\{s,t\},\{c_e\},\gamma,\sigma,\alpha)$

Parameters :

G – network

(s,t) – commodity

$\{c_e\}$ – capacities

γ – demand

σ – integrality restriction

α – network congestion factor

1. For each $e \in E$

   $\tilde{c}_e \leftarrow \left\lfloor \frac{\alpha \cdot c_e}{\sigma} \right\rfloor$.

2. Let $\langle G,(s,t),\{\tilde{c}_e\}\rangle$ be an instance of the Maximum Flow Problem.

   Solve this instance using the *push – relabel* method [25]. Denote by $\{f_e\}$ the respective solution, and by F the total transferred flow from *s* to *t*.

3. If $F \geq \frac{\gamma}{\sigma}$

   Return the link flow $\{\sigma \cdot f_e\}$

   Else

   Return Fail

Fig. 6.1: Procedure Test

Definition 6.3 We say that Procedure Test *succeeds* whenever it does not return Fail.

Lemma 6.1 If Procedure Test succeeds for an input $\langle G,(s,t),\{c_e\},\gamma,\sigma,\alpha\rangle$, then the returned link flow $\{f_e\}$ can be decomposed into a σ-integral path flow that transfers at least γ flow units with a network congestion factor of at most α.

Proof Suppose that Procedure Test succeeds with an input $\langle G,(s,t),\{c_e\},\gamma,\sigma,\alpha\rangle$. Therefore, by construction, the solution $\{f_e\}$ to the instance $\langle G,(s,t),\{\widetilde{c_e}\}\rangle$ of the Maximum Flow Problem transfers at least $\frac{\gamma}{\sigma}$ flow units. Thus, it is easy to see that the returned link flow $\{\sigma\cdot f_e\}$, transfers a total flow of at least $\sigma\cdot\frac{\gamma}{\sigma}=\gamma$ units. In addition, since the solution to the instance $\langle G,(s,t),\{\widetilde{c_e}\}\rangle$ satisfies the capacity constraint $f_e\le\widetilde{c_e}$, for each link $e\in E$, it follows that $\sigma\cdot f_e\le\sigma\cdot\widetilde{c_e}$ holds for each $e\in E$ and therefore the link flow $\{\sigma\cdot f_e\}$ satisfies, $\sigma\cdot f_e\le\sigma\cdot\widetilde{c_e}=\sigma\cdot\left\lfloor\frac{\alpha\cdot c_e}{\sigma}\right\rfloor\le\alpha\cdot c_e$, for each link $e\in E$. Hence, the returned link flow transfers at least γ flow units for the instance $\langle G,(s,t),\{c_e\}\rangle$, such that the network congestion factor is at most α. It remains to be shown that link flow $\{\sigma\cdot f_e\}$ can be decomposed into a σ-integral path flow. To that end, note that the capacities of the instance $\langle G,(s,t),\{\widetilde{c_e}\}\rangle$ are integral. Therefore, the *Push -Relabel* method returns an *integral* link flow [25]. Since the link flow $\{f_e\}$ has only integral values, the returned link flow $\{\sigma\cdot f_e\}$ is σ-integral. Thus, by employing the Flow Decomposition Algorithm [12], a σ-integral path flow can be decomposed out of the returned σ-integral link flow.

Theorem 6.2 Given are an instance $\langle G,(s,t),\{c_e\},\gamma,\sigma\rangle$ of Problem σ-IR and an input $\langle G,(s,t),\{c_e\},\gamma,\sigma,\alpha\rangle$ for Procedure Test. Denote by α^* the optimal network congestion factor corresponding to $\langle G,(s,t),\{c_e\},\gamma,\sigma\rangle$. Then, Procedure Test succeeds *iff* $\alpha\ge\alpha^*$.

Proof

$\Rightarrow$: Suppose that Procedure Test succeeds for the input $\langle G,\{s,t\},\{c_e\},\gamma,\sigma,\alpha\rangle$. Therefore, by Lemma 6.1, there exists a solution for the instance $\langle G,\{s,t\},\{c_e\},\gamma,\sigma\rangle$ of Problem σ-IR, such that the network congestion factor is at most α. Therefore, $\alpha\ge\alpha^*$.

$\Leftarrow$: Suppose that $\alpha\ge\alpha^*$. Therefore, there exists a solution for the instance $\langle G,(s,t),\{c_e\},\gamma,\sigma\rangle$ of Problem σ-IR with a network congestion factor of at most α.

In other words, there exists a σ-integral path flow that transfers at least γ units of flow over the instance $\langle G,\{s,t\},\{\alpha\cdot c_e\}\rangle$ of the Maximum Flow Problem. Since the path flow is σ-integral, we can reduce the capacities $\{\alpha\cdot c_e\}_{e\in E}$ to become multiples of σ without affecting the capability to transfer the flow demand γ. Therefore, it is possible to transfer γ units of flow over the instance $\left\langle G,(s,t),\left\{\left\lfloor\frac{\alpha\cdot c_e}{\sigma}\right\rfloor\cdot\sigma\right\}\right\rangle$ of the Maximum Flow Problem. Hence, it is possible to transfer $\frac{\gamma}{\sigma}$ units of flow over the instance $\left\langle G,(s,t),\left\{\left\lfloor\frac{\alpha\cdot c_e}{\sigma}\right\rfloor\right\}\right\rangle=\left\langle G,(s,t),\{\widetilde{c_e}\}\right\rangle$ of the Maximum Flow Problem. Thus, by construction, Procedure Test does not return Fail, i.e., it succeeds. ■

6.1.3 Solving Problem σ-IR

We now solve problem σ-IR for the single commodity case. To that end, we introduce *Algorithm σ-IR*, specified in Fig. 6.2. Given an instance $\langle G,\{s,t\},\{c_e\},\gamma,\sigma\rangle$ of Problem σ-IR, Algorithm σ-IR finds the smallest $\alpha\in\overline{\alpha}$ (where the set $\overline{\alpha}$ is defined in subsection 6.1.1) such that Procedure Test succeeds for the input $\langle G,\{s,t\},\{c_e\},\gamma,\sigma,\alpha\rangle$. Since Theorem 6.1 establishes that the set $\overline{\alpha}$ must contain the optimal network congestion factor, Algorithm σ-IR discovers the optimal network congestion factor. Finally, as Procedure Test succeeds with $\alpha_1\in\overline{\alpha}$, it must succeed with every $\alpha\geq\alpha_1$, Algorithm σ-IR employs a *binary search* in order to efficiently discover the smallest $\alpha\in\overline{\alpha}$ for which Procedure Test succeeds. Algorithm σ-IR is formally described as follows.

Algorithm σ - IR $\left(G,\{s,t\},\{c_e\},\gamma,\sigma\right)$

Parameters :

G – network

$\{s,t\}$ – commodity

$\{c_e\}_{e\in E}$ – capacities

γ – demand

σ – integrality restriction

Variables:

α – network congestion factor

$\{f_e\}_{e\in E}$ – link flow

1. Perform a binary search over the set $\overline{\alpha}$ in order to find the smallest $\alpha\in\overline{\alpha}$ such that $\textbf{Test}\left(G,\{s,t\},\{c_e\},\gamma,\sigma,\alpha\right)\neq\textbf{FAIL}$.

2. If the binary search has failed i.e., there is no $\alpha\in\overline{\alpha}$ such that $\textbf{Test}\left(G,\{s,t\},\{c_e\},\gamma,\sigma,\alpha\right)\neq\textbf{FAIL}$, **Return Fail**.

Else

Denote by $\alpha_{\min}$ the network congestion factor that was output in the search process.

3. Perform, $\{f_e\}\leftarrow\textbf{Test}\left(G,\{s,t\},\{c_e\},\gamma,\sigma,\alpha_{\min}\right)$.

4. Execute the *Flow Decomposition Algorithm* [12] over the link flow $\{f_e\}$, in order to obtain a path flow $f:P\rightarrow\mathbb{R}^{+}\cup\{0\}$. Let $\left(p_i\right)_{i=1}^{\xi}$ be the resulting set of paths that transfer a positive flow.

5. Return $\left(p_i,f\left(p_i\right)\right)_{i=1}^{\xi}$.

Fig. 6.2 Algorithm σ-IR

In order to prove that Algorithm σ-IR identifies an optimal solution for Problem σ-IR, we first establish that, if Algorithm σ-IR fails, then there is no feasible solution for the given instance. Then we prove that, if it succeeds, then it identifies an optimal solution.

Theorem 6.3 Given is an instance $\left\langle G,\{s,t\},\{c_e\},\gamma,\sigma\right\rangle$ of Problem σ-IR. If Algorithm σ-IR returns Fail, then there is no feasible solution for the given instance.

Proof By construction, if Algorithm σ-IR fails then there is no $\alpha\in\overline{\alpha}$ such that $\textbf{Test}\left(G,\{s,t\},\{c_e\},\gamma,\sigma,\alpha\right)\neq\text{FAIL}$. Thus, according to Theorem 6.2, the optimal network congestion factor, denoted by α^{*}, is larger then the largest element in set $\overline{\alpha}$.

Thus, $\alpha^* \notin \overline{\alpha}$. Finally, according to Theorem 6.1, it follows that there is no feasible solution for the instance $\langle G, \{s,t\}, \{c_e\}, \gamma, \sigma \rangle$. ■

Theorem 6.4 Given is an instance $\langle G, \{s,t\}, \{c_e\}, \gamma, \sigma \rangle$ of Problem σ-IR. If Algorithm σ-IR does not fail, then it returns a σ-integral path flow that transfers at least γ flow units from *s* to *t*, such that the network congestion factor is minimized. Thus, Algorithm σ-IR identifies an optimal solution for Problem σ-IR.

Proof Since Algorithm σ-IR does not fail, it follows by construction that the value $\alpha_{\min}$, identified in step (2) of the algorithm, satisfies $\textbf{Test}(G, \{s,t\}, \{c_e\}, \gamma, \sigma, \alpha_{\min}) \neq \text{FAIL}$. Thus, by Lemma 6.1, the returned link flow $\{f_e\}$ can be decomposed into a σ-integral path flow that transfers at least γ flow units, such that the network congestion factor is at most $\alpha_{\min}$.Finally, this path flow is constructed using the Flow Decomposition Algorithm (step 3).

It remains to be shown that $\alpha_{\min}$ is the optimal network congestion factor for the given instance. However, this is obvious, since $\alpha_{\min}$ is defined to be the smallest $\alpha \in \overline{\alpha}$ such that $\textbf{Test}(G, \{s,t\}, \{c_e\}, \gamma, \sigma, \alpha) \neq \text{FAIL}$. Therefore, it follows by Theorem 6.2, that $\alpha_{\min}$ equals to the smallest $\alpha \in \overline{\alpha}$ such that $\alpha \geq \alpha^*$, where α^* denotes the optimal network congestion factor for the given instance. However, since we established in Theorem 6.1 that $\alpha^* \in \overline{\alpha}$, it follows that $\alpha_{\min} = \alpha^*$. ■

Theorem 6.5 The complexity of Algorithm σ-IR is $O\left(\left(\log N + \log\frac{\gamma}{\sigma}\right) \cdot T(N,M)\right)$, where $T(N,M)$ is the running time of the Push-Relabel Algorithm.

Proof We saw in section 6.1.1 that $\left|\overline{\alpha}\right| \leq M \cdot \frac{\gamma}{\sigma}$. Since Algorithm σ-IR executes a binary search over the set $\overline{\alpha}$ in order to find the smallest $\alpha \in \overline{\alpha}$ such that $\textbf{Test}(G, \{s,t\}, \{c_e\}, \gamma, \sigma, \alpha) \neq \textbf{FAIL}$, it follows that Procedure Test is executed at most $O\left(\log\left(\left|\overline{\alpha}\right|\right)\right)$ times. Note that $\log\left(\left|\overline{\alpha}\right|\right) \leq \log\left(M \cdot \frac{\gamma}{\sigma}\right) =$ $= \log M + \log\frac{\gamma}{\sigma} \leq 2 \cdot \log N + \log\frac{\gamma}{\sigma}$. Since the most time consuming part of Procedure Test is the execution of the Push-Relabel Algorithm, the procedure consumes $O(T(N,M))$ operations. Thus the time complexity of Algorithm σ-IR is $O\left(\left(\log N + \log\frac{\gamma}{\sigma}\right) \cdot T(N,M)\right)$. ■

6.2 Approximation Scheme for Problem KPR

We now employ the solution to Problem σ-IR in order to derive an efficient approximation scheme for Problem KPR. This scheme is based on the following results.

Theorem 6.6 Given are an instance $\left\langle G(V,E),\{c_e\},\left\{\gamma^{(i,j)}\right\},\left\{\mathrm{K}^{(i,j)}\right\}\right\rangle$ of Problem KPR and a value $r \geq 1$. If the optimal network congestion factor of the given instance is α^*, then, for $\sigma^{(i,j)} = \dfrac{\gamma^{(i,j)}}{\mathrm{K}^{(i,j)} \cdot r}$, the optimal network congestion factor for the instance $\left\langle G,\{c_e\},\left\{\gamma^{(i,j)}\right\},\left\{\sigma^{(i,j)}\right\}\right\rangle$ of Problem σ-IR is at most $\left(1+\dfrac{1}{r}\right)\cdot\alpha^*$.

Proof Consider the optimal solution of the instance $\left\langle G(V,E),\{c_e\},\left\{\left(1+\dfrac{1}{r}\right)\cdot\gamma^{(i,j)}\right\},\left\{\mathrm{K}^{(i,j)}\right\}\right\rangle$ of problem KPR. Obviously, the optimal network congestion factor is $\left(1+\dfrac{1}{r}\right)\alpha^*$. Round down the path flow of the considered optimal solution to be a multiple of $\dfrac{\gamma^{(i,j)}}{r\cdot\mathrm{K}^{(i,j)}}$ for each path $p \in P^{(i,j)}$. Therefore, the resulting path flow is $\dfrac{\gamma^{(i,j)}}{r\cdot\mathrm{K}^{(i,j)}}$-integral for each $p \in P^{(i,j)}$. In this process, the flow over each path is reduced by at most $\dfrac{\gamma^{(i,j)}}{r\cdot\mathrm{K}^{(i,j)}}$ flow units. Since there are no more than $\mathrm{K}^{(i,j)}$ paths, the demand of each commodity is reduced by at most $\dfrac{\gamma^{(i,j)}}{r}$ flow units. Since initially each commodity transferred $\left(1+\dfrac{1}{r}\right)\cdot\gamma^{(i,j)}$ flow units, it follows that, after the rounding, each commodity transfers at least $\gamma^{(i,j)}$ flow units.

Thus, we identified a path flow that satisfies the instance $\left\langle G,\{c_e\},\left\{\gamma^{(i,j)}\right\},\left\{\sigma^{(i,j)}\right\}\right\rangle$ of Problem σ-IR, where $\sigma^{(i,j)} = \dfrac{\gamma^{(i,j)}}{\mathrm{K}^{(i,j)}\cdot r}$ for each $(i,j) \in V \times V$. In addition, since in this process we only reduce flow, the network congestion factor is at most $\left(1+\dfrac{1}{r}\right)\alpha^*$. ■

Theorem 6.7 Given an instance $\left\langle G,\{c_e\},\left\{\gamma^{(i,j)}\right\},\left\{\sigma^{(i,j)}\right\}\right\rangle$ of Problem σ-IR, for each commodity $(i,j) \in V \times V$, the total number of paths $p \in P^{(i,j)}$ that transfer a positive amount of flow is at most $\left\lceil \dfrac{\gamma^{(i,j)}}{\sigma^{(i,j)}} \right\rceil$.

Proof Since, for each commodity $(i,j) \in V \times V$ **,** the flow is $\sigma^{(i,j)}$-integral, each path $p \in P^{(i,j)}$ that transfers a positive amount of flow must transfer at least $\sigma^{(i,j)}$ flow units. Therefore, for each commodity $(i,j) \in V \times V$, there are no more than $\left\lceil \frac{\gamma^{(i,j)}}{\sigma^{(i,j)}} \right\rceil$ different paths. ■

We are now ready to define an approximation scheme for Problem KPR, and evaluate its performance. To that end, we employ Algorithm σ-IR that solves Problem σ-IR for the single commodity case.

KPR Approximation Scheme Given a single commodity instance $\langle G, \{s,t\}\{c_e\}, \gamma, \mathrm{K} \rangle$ of Problem KPR and a real number $r \geq 1$, define a single commodity instance $\langle G, \{s,t\}\{c_e\}, \gamma, \sigma \rangle$ of Problem σ-IR, with $\sigma = \frac{\gamma}{\mathrm{K} \cdot r}$, and solve it through Algorithm σ-IR.

Given an instance of Problem KPR, it follows from Theorem 6.6 that the *KPR Approximation Scheme* returns a solution with a network congestion factor that is larger by a factor of at most $\left(1 + \frac{1}{r}\right)$ than the optimal network congestion factor. In addition, by Theorem 6.7, it follows that the returned solution consists of at most $\left\lceil \frac{\gamma}{\sigma} \right\rceil = \left\lceil \frac{\gamma}{\frac{\gamma}{\mathrm{K} \cdot r}} \right\rceil = \lceil \mathrm{K} \cdot r \rceil$ different paths with positive flow. We summarize the above discussion as follows.

Corollary 6.1 Given are a single commodity instance $\langle G, (s,t)\{c_e\}, \gamma, \mathrm{K} \rangle$ of Problem KPR and a real number $r \geq 1$. The KPR Approximation Scheme produces at most $\lceil \mathrm{K} \cdot r \rceil$ paths with a network congestion factor that is at most $\left(1 + \frac{1}{r}\right)$ times larger than the optimal network congestion factor of the given instance. ■

Since we proved that the complexity of Algorithm σ-IR is $O\left(\left(\log N + \log \frac{\gamma}{\sigma}\right) \cdot T(N,M)\right)$, and since $\sigma = \frac{\gamma}{\mathrm{K} \cdot r}$, it follows that the complexity of the KPR Approximation scheme is $O\left(\left(\log(N) + \log(\mathrm{K} \cdot r)\right) \cdot T(N,M)\right) = O\left(\left(\log(N) + \log(\mathrm{K}) + \log(r)\right) \cdot T(N,M)\right)$, i.e., polynomial in the input size.

In [10], a 0.5-approximation scheme is presented for the problem of maximizing flow under a restriction on the number of paths. In Theorem 2.1 we establish a reduction between the minimization of the network congestion factor and the maximization of the flow. In the following we prove that this reduction is an approximate preserving

reduction. More specifically, we prove that the reduction of Theorem 2.1 maps *r*-approximates solutions to the congestion minimization problem into $\frac{1}{r}-$approximate solutions to the flow maximization problem. Hence, our solution to Problem KPR for $r=1$ provides an alternative solution to the problem of [10], with equal performance guarantees. Moreover, our approach generalizes the result of [10] to any $r\geq 1$.

Theorem 6.8 Given are a network $G(V,E)$ two nodes $\{s,t\}$ capacities $\{c_e\}$ and a demand γ. If $\{f_e\}$ is a link flow that transfers γ flow units from *s* to *t* such that the network congestion factor α is larger by a factor of at most *r* than the optimal network congestion factor, then $\left\{\frac{1}{\alpha}\cdot f_e\right\}$ is $\frac{1}{r}$ approximation to the instance $\langle G(V,E),\{c_e\},\{s,t\}\rangle$ of the maximum flow problem.

The proof to the Theorem appears in the Appendix

For completeness we provide an approximate preserving reduction that maps $\frac{1}{r}-$approximate solutions of the flow maximization problem into *r*-approximates solutions of the congestion minimization problem.

Theorem 6.9 Given are a network $G(V,E)$ two nodes $\{s,t\}$ and capacities $\{c_e\}$. If $\{f_e\}$ is a link flow that transfers F' flow units from *s* to *t* such that $\{f_e\}$ is an $\frac{1}{r}-$approximation to the instance $\langle G(V,E),\{c_e\},\{s,t\}\rangle$ of the Maximum Flow Problem then $\left\{f_e\cdot\frac{\gamma}{F'}\right\}$ is a link flow that transfer γ flow units from *s* to *t* with a network congestion factor larger by a factor of at most *r* than the optimal network congestion factor.

The proof to the Theorem appears in the Appendix

6.3 Extensions to Problem KPR

In this section we briefly describe some additional results that are related to Problem KPR. More specifically, one result deals with the dual problem, which restricts the network congestion factor and minimizes the number of paths for routing and another result adds a cost restriction.

6.3.1. Problem MPR

Problem MPR is defined as follows:

Problem MPR (Minimum Paths for Routing) Given are a network $G(V,E)$, two nodes $s,t\in V$, A capacity $c_e>0$ for each link $e\in E$, a demand $\gamma>0$ and a restriction on the network congestion factor α. Find a path flow with a network congestion factor of at

most α, such that, if $P \subseteq P^{(s,t)}$ is the set of all paths in $P^{(s,t)}$ that are assigned with a positive flow, then $|P|$ is minimized.

A solution to Problem MPR can be derived by employing the KPR Approximation Scheme (specified in section 6.2) as follows.

Given an instance of Problem MPR, denote by K^* the number of different paths that carry a positive flow in the optimal solution. Observe that K^* must take values only in the range $[1, M]$ (since every link flow can be decomposed into the sum of at most M paths and cycles that carry a positive flow). Therefore, given an instance $\langle G,(s,t),\{c_e\},\gamma,\alpha\rangle$ of Problem MPR a real number $r \geq 1$, one can define, for each $\mathrm{K} \in [1, M]$, an instance $\langle G,(s,t),\{c_e\},\gamma,\mathrm{K}\rangle$ of Problem KPR, and solve the resulting instance using the KPR Approximation Scheme for the given r. If the resulting network congestion factor is larger than $\left(1+\frac{1}{r}\right)\cdot\alpha$, a smaller value of K shall be chosen, otherwise a larger one shall be chosen. We terminate when we get the smallest K such that the network congestion factor is at most $\left(1+\frac{1}{r}\right)\cdot\alpha$. This approximation scheme shall be termed as the MPR Approximation Scheme.

Theorem 6.10 Given an instance $\langle G,(s,t),\{c_e\},\gamma,\alpha\rangle$ of Problem MPR, the MPR Approximation Scheme finds a path flow with at most K^* paths that transfer at least γ units of flow, such that the network congestion factor is at most $\left(1+\frac{1}{r}\right)\cdot\alpha$.

Proof By construction, the output of the MPR Approximation Scheme is a path flow that transfers at least γ flow units such that the network congestion factor is at most $\left(1+\frac{1}{r}\right)\cdot\alpha$. Therefore, we only need to prove that the number of defined paths is indeed K^*. Denote by K^{**} the number of different paths that the MPR Approximation Scheme calculated for the instance $\langle G,(s,t),\{c_e\},\gamma,\alpha\rangle$. We have to prove that $\mathrm{K}^* \geq \mathrm{K}^{**}$. By construction, K^{**} is the smallest $\mathrm{K} \in [1, M]$ such that the optimal network congestion factor for the instance $\langle G,(s,t),\{c_e\},\gamma,\mathrm{K}\rangle$ of Problem KPR is smaller than $\left(1+\frac{1}{r}\right)\cdot\alpha$. Therefore, for the instance $\langle G,(s,t),\{c_e\},\gamma,\mathrm{K}^{**}-1\rangle$, the KPR Approximation Scheme returns a network congestion factor that is larger than $\left(1+\frac{1}{r}\right)\cdot\alpha$. Since, for a given instance $\langle G,(s,t),\{c_e\},\gamma,\mathrm{K}\rangle$ of Problem KPR and a real number $r \geq 1$, the KPR Approximation Scheme finds a path flow with a network congestion factor that is at most $\left(1+\frac{1}{r}\right)$ times larger than the optimum, the optimal network congestion factor for the instance $\langle G,(s,t),\{c_e\},\gamma,\mathrm{K}^{**}-1\rangle$ of Problem KPR

is larger than α. Therefore, since the optimal solution for the given instance of Problem MPR produces a network congestion factor of at most α we conclude that $\mathrm{K}^{*} > \mathrm{K}^{**} - 1$, hence $\mathrm{K}^{*} \geq \mathrm{K}^{**}$. ■

We note that, by employing Procedure Test (introduced in section 6.1) directly, a *better* time complexity approximation scheme can be derived. We omit the detailed description of this approximation scheme, and only note that, instead of solving an instance $\langle G,(s,t),\{c_e\},\gamma,\mathrm{K}\rangle$ of Problem KPR for each $\mathrm{K} \in [1,M]$, we execute Procedure Test directly, for each $\mathrm{K} \in [1,M]$, with an input $\left\langle G,(s,t),\{c_e\},\gamma,\sigma,\left(1+\frac{1}{r}\right)\cdot\alpha\right\rangle$ such that $\sigma = \frac{\gamma}{\mathrm{K}\cdot r}$.

6.3.2 Problem BCKPR

Another restriction that can be added to Problem KPR is a "cost", as follows.

Problem BCKPR (Bounded Cost K-Path Routing) Given are a network $G(V,E)$, for each link $e \in E$, a capacity $c_e > 0$ and a weight (i.e., "cost") $w_e > 0$, and, for each commodity $(i,j) \in V \times V$, a demand $\gamma^{(i,j)}$, a split restriction $\mathrm{K}^{(i,j)}$ and a weight ("cost") restriction $C^{(i,j)}$. Find a path flow that minimizes the network congestion factor, subject to the following:

- $\sum_{e\in E} w_e \cdot f_e^{(i,j)} \leq C^{(i,j)}$.
- If $P_x^{(i,j)} \subseteq P^{(i,j)}$ is the collection of all paths in $P^{(i,j)}$ that are assigned with positive flow, then $\left|P_x^{(i,j)}\right| \leq \mathrm{K}^{(i,j)}$.

An approximation scheme for Problem BCKPR for the single commodity case can be derived by substituting the Maximum Flow algorithm in Procedure Test with a Minimum Cost Flow algorithm [12]. Then, if $\sum_{e\in E} w_e \cdot f_e \leq C$, Procedure Test will return Fail and a larger network congestion factor will be chosen by Algorithm σ-IR. We omit the precise description of the approximation scheme and its proof.

7. Future research

During this work we observed that multipath routing offers many advantages in contexts that are not necessarily related to congestion avoidance or load balancing. In the following we present a brief description of some ideas for future research.

Multipath routing and survivability

Multipath routing can be used in order to improve resilience and avoid congestion. The combination of both benefits can be obtained by employing the idea of *diversity coding* [3], which adds redundant information to the data stream, like error detection and correction codes. Then, in order to increase fault tolerance, the redundant information is routed along paths that are disjoint to the paths that are used to transfer the original data stream. Therefore, it is desired to develop new multipath routing schemes that also engage the diversity coding concept. For example, it is desired to develop schemes for multipath routing that maximize the total flow (or minimize the congestion) and satisfy a fundamental property that restricts each path that transfer positive data flow to have an adequate set of disjoint paths with enough bandwidth to protect this flow.

Multipath routing and security

Splitting the traffic among several node-disjoint paths has the advantage that no node (except the source and the target) gets to see the whole data stream. Therefore, in practice, reconstructing the data stream is possible only at the target node. This inherent advantage of multipath routing can be used in order to enhance security.

In order to exploit this idea, we should identify several node-disjoint paths, and transfer a *limited* portion of the total flow demand over each. This limitation is a key point, since otherwise a single path could transfer about all of the flow.

Accordingly, it is desirable to incorporate this notion into multipath routing schemes in order to satisfy security requirements.

Energy efficiency and multipath routing

In wireless networks, where nodes have limited power resources (batteries), energy conserving routing must be employed in order to maximize the network lifetime, i.e., the time until the first battery drains-out [29]. Since energy consumption is proportional to a node's transmission rate, it is easy to see that splitting the traffic among several paths can prolong the network lifetime, i.e., the time until the first battery drains out. Therefore, it is of interest to design and explore practical multipath routing schemes that incorporate energy conservation policies for such environments.

Recovery schemes for multipath routing

Multipath routing can provide additional benefits in networks where resource reservation must be made before data can be sent along a route (e.g. ATM). Consider for example a routing scheme that uses multipath routing in order to reduce congestion. Then, in case of a failure in one of the paths, we may split the data stream

that was traveling over the failed path among the remaining paths, *without* any additional path computation or resource reservation. This fast recovery property can be obtained in several ways. For example, it may be employed by a restriction that, upon a path failure, the sum of the *spare* capacities of the remaining paths is not smaller than the flow that was traveling over the failed path.

Distributed multipath routing schemes

So far, we considered multipath routing schemes that employed source routing; hence, all routing decisions were determined at the source node. Although source routing is simple and intuitive, there are many cases that the use of hop-by-hop routing is preferable. Hence, investigation of distributed multipath routing schemes is called for.

Designing practical distributed multipath routing schemes raises several challenges. One major challenge is to limit the buffer space usage at each node in the network. In other words, if $h_v(i)$ denotes the number of next hops that are allocated for a target i in the routing table of a node v, a distributed multipath routing scheme needs to cope with the restriction that $\sum_{i \in V} h_v(i)$ cannot exceed the available buffer space at node $v \in V$.

Dynamic multipath routing schemes

So far, we have investigated multipath routing schemes where all demands are known in advance. However, in practice, demands may arrive one at a time without any a-priory knowledge regarding future demands. Therefore, it is important to investigate on-line multipath routing schemes that don't rely on such a-priory knowledge.

We note that we are currently working on this issue and have obtained some important results.

Appendix

This Appendix contains the proofs of Theorems 2.1, 6.8 and 6.9. In addition, it provides a formal specification of Algorithm RDJM_PFC.

Theorem 2.1 Given a network $G(V,E)$ two nodes *s*,*t* capacities $\{c_e\}$ and a demand γ. $\{f_e\}$ is a solution to the instance $\langle G(V,E),\{c_e\},\{s,t\}\rangle$ of the Maximum Flow Problem that transfers $F \geq \gamma$ flow units from *s* to *t iff* $\left\{f_e \cdot \frac{\gamma}{F}\right\}$ is a link flow that transfer γ flow units from *s* to *t* such that the network congestion factor is minimized .

Proof

$\Rightarrow$

It is easy to see that, since $\{f_e\}$ is a vector that satisfies the capacity and flow conservation constraints, then the vector $\left\{f_e \cdot \frac{\gamma}{F}\right\}$ must also satisfy these constraints. In addition, since $\{f_e\}$ transfers *F* flow units from *s* to *t* i.e., $\sum_{e\in O(s)} f_e - \sum_{e\in I(s)} f_e = F$ then it follows that $\sum_{e\in O(s)} \frac{\gamma}{F} f_e - \sum_{e\in I(s)} \frac{\gamma}{F} f_e = \frac{\gamma}{F}\cdot\left(\sum_{e\in O(s)} f_e - \sum_{e\in I(s)} f_e\right) = \frac{\gamma}{F}\cdot F = \gamma$ flow units. Thus, $\left\{f_e \cdot \frac{\gamma}{F}\right\}$ is a link flow that transfers γ flow units from *s* to *t*. It is left to be shown that the value $\alpha \triangleq \max_{e\in E}\left\{\frac{f_e}{c_e}\cdot\frac{\gamma}{F}\right\}$ is the minimum of all link flows that transfer γ flow units from *s* to *t*. Since there must be a link that satisfies $f_e = c_e$, it follows that $\alpha = \frac{\gamma}{F}$. By way of contradiction, suppose that there exists a link flow $\{\tilde{f}_e\}$ that transfers γ flow units from *s* to *t* such that $\tilde{\alpha} = \max_{e\in E}\left\{\frac{\tilde{f}_e}{c_e}\right\} < \frac{\gamma}{F}$. Consider the vector $\left\{\frac{\tilde{f}_e}{\tilde{\alpha}}\right\}$. Obviously, the network congestion factor is 1 and therefore the capacity constraints are satisfied. Since the flow conservation constraints are also satisfied, the vector $\left\{\frac{\tilde{f}_e}{\tilde{\alpha}}\right\}$ is a link flow. It is given that $\sum_{e\in O(s)} \tilde{f}_e - \sum_{e\in I(s)} \tilde{f}_e = \gamma$. Therefore, $\sum_{e\in O(s)} \frac{\tilde{f}_e}{\tilde{\alpha}} - \sum_{e\in I(s)} \frac{\tilde{f}_e}{\tilde{\alpha}} = \frac{1}{\tilde{\alpha}}\cdot\left(\sum_{e\in O(s)} \tilde{f}_e - \sum_{e\in I(s)} \tilde{f}_e\right) = \frac{\gamma}{\tilde{\alpha}} > \frac{\gamma}{\gamma/F} = F$. Thus, link flow $\left\{\frac{\tilde{f}_e}{\tilde{\alpha}}\right\}$ transfers more than *F* flow units from *s* to *t*.

$\Leftarrow$

Suppose that $\{\tilde{f}_e\}$ is a link flow that transfers γ flow units from *s* to *t* such that the network congestion factor is minimized. We have to show that $\left\{\tilde{f}_e \cdot \frac{F}{\gamma}\right\}$ is a link flow

that transfers F flow units from s to t. Since $\{\widetilde{f}_e\}$ satisfies the flow conservation constraint it follows that $\left\{\widetilde{f}_e \cdot \frac{F}{\gamma}\right\}$ also satisfies the flow conservation constraints. We will prove that link flow $\left\{\widetilde{f}_e \cdot \frac{F}{\gamma}\right\}$ satisfies the capacity constraints. To that end, we observe that the network congestion factor of $\{\widetilde{f}_e\}$ is at most $\frac{\gamma}{F}$. Proving that will show that the maximum network congestion factor is not larger than 1 and thus the capacity constraints are satisfied.

Suppose that the link flow $\{\widetilde{f}_e\}$ has a network congestion factor that is larger than $\frac{\gamma}{F}$ and consider the link flow $\{f_e\}$ that was defined to be the solution to the instance $\langle G(V,E),\{c_e\},\{s,t\}\rangle$ of the Maximum Flow Problem that transfers $F \geq \gamma$ flow units from s to t. It is easy to see that the link flow $\left\{\frac{\gamma}{F} \cdot f_e\right\}$ transfers γ flow units from s to t. In addition since the link flow $\{f_e\}$ is a solution to the Maximum Flow Problem, it follows that there exists one link $e \in E$ such that $f_e = c_e$. Thus, the network congestion factor of link flow $\left\{\frac{\gamma}{F} \cdot f_e\right\}$ has a network congestion factor that is equal to $\frac{\gamma}{F}$. Obviously, this contradicts the optimality of $\{\widetilde{f}_e\}$.

It is left to be shown that the link flow $\left\{\widetilde{f}_e \cdot \frac{F}{\gamma}\right\}$ transfers F flow units from s to t. To that end, we employ the fact that $\{\widetilde{f}_e\}$ transfers γ flow units from s to t as follows.

$$\sum_{e \in O(s)} \frac{F}{\gamma} \widetilde{f}_e - \sum_{e \in I(s)} \frac{F}{\gamma} \widetilde{f}_e = \frac{F}{\gamma} \cdot \left(\sum_{e \in O(s)} f_e - \sum_{e \in I(s)} f_e \right) = \frac{F}{\gamma} \cdot \gamma = F \ . \qquad \blacksquare$$

Theorem 6.8 Given are a network $G(V,E)$, two nodes $\{s,t\}$, capacities $\{c_e\}$ and a demand γ. If $\{f_e\}$ is a link flow that transfers γ flow units from s to t such that the network congestion factor α is larger by a factor of at most r than the optimal network congestion factor, then $\left\{\frac{1}{\alpha} \cdot f_e\right\}$ is a $\frac{1}{r}$ approximation to the instance $\langle G(V,E),\{c_e\},\{s,t\}\rangle$ of the maximum flow problem.

Proof

Since $\{f_e\}$ is a link flow, it follows that the vector $\left\{\frac{1}{\alpha} \cdot f_e\right\}$ satisfies the conservation constraints. In addition since the network congestion factor of $\{f_e\}$ is α, it follows that the vector $\left\{\frac{1}{\alpha} \cdot f_e\right\}$ has a network congestion factor of 1. Thus, the capacity

constraints are satisfied. Thus, $\left\{\frac{1}{\alpha}\cdot f_e\right\}$ is a link flow. We will show now that if F is the total traffic that the optimal solution to the instance $\langle G(V,E),\{c_e\},\{s,t\}\rangle$ of the maximum flow problem transfers from *s* to *t*, then $\left\{\frac{1}{\alpha}\cdot f_e\right\}$ transfers at least $\frac{F}{r}$.

To that end, denote by $\{f_e^*\}$ the optimal solution that transfers γ flow units from *s* to *t* such that the network congestion factor is minimized. In addition, let α^* be the network congestion factor of link flow $\{f_e^*\}$. It follows from Theorem 2.1 that $\left\{f_e^*\cdot\frac{F}{\gamma}\right\}$ is the optimal solution to the instance $\langle G(V,E),\{c_e\},\{s,t\}\rangle$ of the maximum flow problem. Therefore, there exists a link $e\in E$ such that $f_e^*\cdot\frac{F}{\gamma}=c_e$. Thus, there exists a link $e\in E$ such that $\frac{f_e^*}{c_e}=\frac{\gamma}{F}$. Thus, $\alpha^*=\frac{\gamma}{F}\Rightarrow F=\frac{\gamma}{\alpha^*}\Rightarrow\frac{F}{r}=\frac{\gamma}{r\cdot\alpha^*}$. Therefore, we will only prove that link flow $\left\{\frac{1}{\alpha}\cdot f_e\right\}$ transfers at least $\frac{\gamma}{r\cdot\alpha^*}$ flow units.

To that end, recall that $\{f_e\}$ transfers γ flow units from *s* to *t* i.e., $\sum_{e\in O(s)}f_e-\sum_{e\in I(s)}f_e=\gamma$. Therefore, $\sum_{e\in O(s)}\frac{1}{\alpha}f_e-\sum_{e\in I(s)}\frac{1}{\alpha}f_e=$ $=\frac{1}{\alpha}\cdot\left(\sum_{e\in O(s)}f_e-\sum_{e\in I(s)}f_e\right)=\frac{\gamma}{\alpha}$. Finally, since $\alpha\le r\cdot\alpha^*$, it follows that link flow $\left\{\frac{1}{\alpha}\cdot f_e\right\}$ transfers at least $\frac{\gamma}{\alpha}\ge\frac{\gamma}{r\cdot\alpha^*}$ flow units. ■

Theorem 6.9 Given are a network $G(V,E)$ two nodes $\{s,t\}$ and capacities $\{c_e\}$. If $\{f_e\}$ is a link flow that transfers F' flow units from *s* to *t* such that $\{f_e\}$ is a $\frac{1}{r}-$approximation to the instance $\langle G(V,E),\{c_e\},\{s,t\}\rangle$ of the Maximum Flow Problem, then $\left\{f_e\cdot\frac{\gamma}{F'}\right\}$ is a link flow that transfer γ flow units from *s* to *t* with a network congestion factor larger by a factor of at most *r* than the optimal network congestion factor.

Proof

It is easy to see that, since $\{f_e\}$ is a vector that satisfies the capacity and the flow conservation constraints then the vector $\left\{f_e\cdot\frac{\gamma}{F'}\right\}$ must also satisfy these constraints. In addition, since $\{f_e\}$ transfers *F'* flow units from *s* to *t* i.e., $\sum_{e\in O(s)}f_e-\sum_{e\in I(s)}f_e=F'$,

then it follows that $\sum_{e\in O(s)}\frac{\gamma}{F'}f_e-\sum_{e\in I(s)}\frac{\gamma}{F'}f_e=\frac{\gamma}{F'}\cdot\left(\sum_{e\in O(s)}f_e-\sum_{e\in I(s)}f_e\right)=\frac{\gamma}{F'}\cdot F'=\gamma$. Thus, $\left\{f_e\cdot\frac{\gamma}{F'}\right\}$ is a link flow that transfers γ flow units from *s* to *t*. It is left to be shown that, if α^* is the optimal network congestion factor, then it follows that the network congestion factor of link flow $\left\{f_e\cdot\frac{\gamma}{F'}\right\}$ is at most $r\cdot\alpha^*$. To that end, denote by $\{f_e^*\}$ the optimal solution to the instance $\langle G(V,E),\{c_e\},\{s,t\}\rangle$ of the Maximum Flow Problem and by *F* the total flow that link flow $\{f_e^*\}$ transfers from *s* to *t*. We established (Theorem 2.1) that the link flow $\left\{\frac{\gamma}{F}\cdot f_e^*\right\}$ transfers γ flow units from *s* to *t* while minimizing the network congestion factor. Thus, $\alpha^*=\max_{e\in E}\left\{\frac{f_e^*}{c_e}\cdot\frac{\gamma}{F}\right\}$. Since $\{f_e^*\}$ is a maximum flow, it follows that there exists a link $e\in E$ such that $f_e^*=c_e$. Therefore, $\alpha^*=\max_{e\in E}\left\{\frac{f_e^*}{c_e}\cdot\frac{\gamma}{F}\right\}=\frac{\gamma}{F}$. In order to complete the proof we have to show that $\alpha\triangleq\max_{e\in E}\left\{\frac{f_e}{c_e}\cdot\frac{\gamma}{F'}\right\}\le r\cdot\frac{\gamma}{F}$. However, Since $\{f_e\}$ is a $\frac{1}{r}-$approximation to the instance $\langle G(V,E),\{c_e\},\{s,t\}\rangle$ of the Maximum Flow Problem, it follows that $F'\ge\frac{1}{r}\cdot F$. Thus, $\alpha=\max_{e\in E}\left\{\frac{f_e}{c_e}\cdot\frac{\gamma}{F'}\right\}\le\max_{e\in E}\left\{\frac{f_e}{c_e}\cdot\frac{\gamma}{\frac{1}{r}\cdot F}\right\}=r\cdot\max_{e\in E}\left\{\frac{f_e}{c_e}\cdot\frac{\gamma}{F}\right\}$. Finally, since $f_e\le c_e$ for each $e\in E$, it follows that $\alpha=\max_{e\in E}\left\{\frac{f_e}{c_e}\cdot\frac{\gamma}{F'}\right\}\le r\cdot\max_{e\in E}\left\{\frac{f_e}{c_e}\cdot\frac{\gamma}{F}\right\}\le r\cdot\max_{e\in E}\left\{\frac{\gamma}{F}\right\}=r\cdot\frac{\gamma}{F}=r\cdot\alpha^*$.

■

We provide now a specification for Algorithm RDJM_PFC, which was informally described in Section 5.2.2. The algorithm is used by Algorithm RRMP that is specified in Fig 5.2. As was already explained, the algorithm constructs a feasible path flow out of the given link flow $\left\{f_e^{\omega,h}\right\}$. The algorithm employs Procedure Path_Construction_RDJM, specified in Fig (A.2). Algorithm RDJM_PFC is specified as follows (Fig. A.1)

Algorithm PFC_RDJM $\left(G(V,E),\{s,t\}\left\{f_e^{\omega,h}\right\},\{\gamma\}\right)$

Initialization:

For each path $p \in P^{(s,t)}$: $f(p) \leftarrow 0$.

While $\gamma > 0$ do:

1. $S \leftarrow$ Path_Construction_RDJM$\left(G(V,E),\{s,t\}\left\{f_e^{\omega.h}\right\}\right)$.
2. $f_{e_k}^{\Delta_k,\eta} \leftarrow f_{e_k}^{\Delta_k,\eta_k} - \min_{(e_k,\Delta_k,\eta)\in S}\left\{f_{e_k}^{\Delta_k,\eta_k}\right\}$, for each $(e_k,\Delta_k,\eta_k)\in S$.
3. $\gamma \leftarrow \gamma - \min_{(e_k,\Delta_k,\eta_k)\in S}\left\{f_{e_k}^{\Delta_k,\eta_k}\right\}$.
4. Denote the path $s \xrightarrow{e_1} v_1 \xrightarrow{e_2} v_2 \cdots v_{|S|} \xrightarrow{e_{|S|}} t$ as *p*, where e_k corresponds to the triplet $(e_k,\Delta_k,\eta_k)\in S$.
5. $f(p) \leftarrow \min_{(e_k,\Delta_k,\eta_k)\in S}\left\{f_{e_k}^{\Delta_k,\eta_k}\right\}$.

Return the path flow *f*.

Fig. A.1: Algorithm PFC_RDJM

Procedure Path_Construction_RDJM$\left(G(V,E),\{s,t\},\left\{f_e^{\omega,h}\right\}\right)$

Initialization

$S \leftarrow \phi$, $u_0 \leftarrow s$, $\Delta_0 \leftarrow 0$, $\eta_0 \leftarrow 0$, $k \leftarrow 0$

While $u_k \neq t$ do

1. Select a positive variable $f_{e_k}^{\Delta_k,\eta_k}$ such that $e_k \triangleq (u_k,u_{k+1}) \in E$.
2. $S \leftarrow S \cup (e_k,\Delta_k)$, $\Delta_{k+1} \leftarrow \Delta_k + w_{e_k}$, $\eta_{k+1} \leftarrow \eta_k + 1$, $k \leftarrow k+1$.

Return *S*

Fig. A.2: Procedure Path_Construction_RDJM

References

[1] S. Iyer, S. Bhattacharyya, N. Taft, N. McKeoen, C. Diot, “A measurement Based Study of Load Balancing in an IP Backbone”, Sprint ATL Technical Report, TR02-ATL-051027, May 2002.

[2] D. Bertsekas and R. Gallager, “Data networks”, Prentice-Hall, 1992.

[3] A. Tsirigos and Z. J. Haas, “Multipath Routing in Mobile Ad Hoc Networks or How to Route in the Presence of Topological Changes”, In Proceedings of IEEE MILCOM 2001, pages 878-883, October 2001.

[4] M. Kodialam and T. V. Lakshman, “Restorable Dynamic Quality of Service Routing”, IEEE Communications Magazine, vol. 40, no. 6, June 2002.

[5] D. Thaler and C. Hopps, “Multipath Issues in Unicast and Multicast Next-Hop Selection”, IETF RFC 2991, 2000.

[6] C. Villamizar, “OSPF Optimised Multipath (OSPF-OMP)”, Internet Draft <draft-ietf-ospf-omp-02.txt>, 1998.

[7] S. Nelakuditi and Zhi-Li Zhang, “On Selection of Paths for Multipath Routing”, In Proc. IWQoS, 2001, Karlsruhe, Germany.

[8] I. Cidon, R. Rom and Y. Shavitt, “Analysis of Multi-Path Routing”, IEEE Transactions on Networking, 7:885–896, 1999.

[9] W. Lai, Ed. and D. McDysan, Ed.,” Network Hierarchy and Multilayer Survivability”, IETF RFC 3386, November 2002.

[10] G. Baier, E. Koehler and M. Skutella, “On the *k* -Splittable Flow Problem”, In Proceedings of the 10th Annual European Symposium on Algorithms (ESA), 2002.

[11] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin and O. Waarts, “On-Line Routing of Virtual Circuits with Applications to Load Balancing and Machine Scheduling”, Journal of the ACM, vol. 44, no. 3, pages 486-504, May 1997.

[12] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, "Network Flows: Theory, Algorithm, and Applications", Prentice Hall, 1993.

[13] D.D. Sleator and R.E. Tarjan, "Amortized efficiency of list update and paging rules", Communications of the ACM, vol. 28, no. 2, pages 202-208, February 1985.

[14] J.Moy, ”OSPF Version 2”, IETF RFC 2328, April 1998.

[15] Y. Wang and Z. Wang, “Explicit Routing Algorithms For Internet Traffic Engineering”, In Proceedings of ICCN'99, Boston, October 1999.

[16] Yanxia Jia, Ioanis Nikolaidis and P. Gburzynski, "Multiple Path QoS Routing", Proceedings of ICC'01 Finland, pages 2583-2587, June 2001.

[17] S. Chen and K. Nahrstedt, “An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions”, IEEE network Magazine, Special Issue on Transmission and Distribution of Digital Video, vol. 12, num. 6, pages 64-79, December 1998.

[18] J.Kleinberg, “Single-Source Unsplittable Flow”, in proceeding of the 37th annual symposium on foundations of computer science, October 1996 pages 23-25.

[19] Y. Dinitz, N. Garg and M. X. Goemans, “On the Single –Source Unsplittable Flow Problem”, Combinatorica , 19, pages 68-77.

[20] F. Shahrokhi and D. Matula, “The Maximum Concurrent Flow Problem” Journal of ACM, vol. 37, no. 2, pages 318-334, 1990.

[21] O. Hauser, M. Kodialam and T.V. Lakshman “Capacity Design of Fast Path Restorable Optical Networks“, IEEE/ACM Transactions on Networking, October 2000.

[22] A. Pitsillides, S. Nikolopoulos and D. Tipper,”Addressing Network Survivability Issues by Finding the K-best Paths Through a Trellis Graph", Proceedings of IEEE INFOCOM *'97*, Kobe, Japan, April 1997.

[23] M. R. Garey and D. S. Johnson, "Computers and Intractability", W.H. Freeman and Co., 1979.

[24] N. Karmarkar, "A New Polynomial-Time Algorithm For Linear Programming", Combinatorica, vol. 4, pages 373-395, 1984.

[25] A. V. Goldberg and R. E. Tarjan, "A New Approach to The Maximum Flow Problem", Journal of ACM, vol.35, no. 4, pages 921-940, 1988.

[26] A. Bley, "On the Complexity of Vertex-Disjoint Length-Restricted Path Problems", Konrad-Zuse-Zentrum fur Informationstechnik Berlin Tech, Report SC-98-20, 1998.

[27] G. Ausiello, P.Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela and M. Protasi, "Complexity and Approximation: Combinatorial Optimization Problems and Their Approximation Properties", Springer-Verlag, 1999.

[28] E. Comer, "Internetworking with TCP/IP Volume I: Principles, Protocols and Architecture", Prentice Hall, 3rd edition, 1995.

[29] G. Zussman and A. Segall, "Energy Efficient Routing in Ad Hoc Disaster Recovery Networks", In Proceedings of IEEE INFOCOM 2003, April 2003.

[30] D. Awduche, J. Malcolm, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS" , Internet Draft <draft-awduche-mpls-traffic-eng-00.txt>, April, 1998.

[31] A. E. I. Widjaja, "Mate: MPLS Adaptive Traffic Engineering", Internet Draft <draft-widjaja-mpls-mate-00.txt>, August, 1998.

[32] S. Plotkin, "Competitive Routing of Virtual Circuits in ATM Networks", IEEE J. Selected Areas in Communications, vol. 13, pages 1128-1136, August 1995.

[33] J.M. Akinpelu, "The Overload Performance of Engineered Networks With Non-Hierarchical and Hierarchical Routing", AT&T Bell Laboratories Technical Journal, vol. 63, no. 7, September 1984.

[34] G.R. Ash, "Dynamic Network Evolution, with Examples from AT&T's Evolving Dynamic Network", IEEE Communications Magazine, pages 26-39, July 1995.

[35] B.R. Hurley, C.J.R. Seidl and W.F. Sewell, "A Survey of Dynamic Routing Methods for Circuit-Switched Traffic", IEEE Communications Magazine, vol. 25, no. 9, September 1987.

[36] Y. Azar, B. Kalyanasundaram, S. Plotkin, K. Pruhs, and O. Waarts, "On-line Load Balancing of Temporary Tasks", In Proceedings of Workshop on Algorithms and Data Structures, pages 119-130, August 1993.

[37] R. Ogier, B. Bellur, and N. Taft-Plotkin, "An Efficient Algorithm for Computing Shortest and Widest Maximally Disjoint Paths", SRI International Technical Report ITAD-1616-TR-170, November 1998.

[38] N. Taft-Plotkin, B. Bellur, and R. Ogier, "Quality-of –Service Routing Using Maximally Disjoint Paths", In Proceedings of IEEE IWQoS99, pages 119-128, London, UK, January 1999.

[39] M. Kodialam and T. V. Lakshman, "Restorable Dynamic Quality of Service Routing", IEEE Communication Magazine, vol. 40, no. 6, June 2002.

[40] A. Ouorou, P. Mahey, and J.-Ph. Vial, "A Survey of Algorithms For Convex Multicommodity Flow Problems", Management Science, vol. 46, pages 126-147, January 2000.

[41] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell and J. McManus, "Requirements for Traffic Engineering Over MPLS", IETF RFC 2702, September 1999.

[42] J. Semke, J. Mahdavi, and M. Mathis, "Automatic TCP buffer tuning", In Proc. ACM SIGCOMM'98 Conference, 1998.

[43] V. Paxson, "End-to-End Routing Behavior in the Internet", in proc. ACM SIGCOM, 1996.