

Flow Allocation with Path Protection in Next Generation Internet Networks

Samer Lahoud, Géraldine Texier, Laurent Toutain

► **To cite this version:**

Samer Lahoud, Géraldine Texier, Laurent Toutain. Flow Allocation with Path Protection in Next Generation Internet Networks. IEEE International Conference on Communications, ICC '06, 2006, Istanbul, Turkey. 2, pp.860 -865, 2006, <10.1109/ICC.2006.254815>. <inria-00535998>

HAL Id: inria-00535998

<https://hal.inria.fr/inria-00535998>

Submitted on 15 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Flow Allocation with Path Protection in Next Generation Internet Networks

Samer Lahoud, Géraldine Texier and Laurent Toutain

GET/ENST Bretagne

Département Réseaux et Sécurité Multimédia

2 rue de la Châtaigneraie - CS 17607 - 35576 Cesson Sévigné Cedex - France

samer.lahoud@enst-bretagne.fr, geraldine.texier@enst-bretagne.fr, laurent.toutain@irisa.fr

Abstract—In this work, we study the flow allocation problem in Next Generation Internet (NGI) networks. Given a capacitated network, flow allocation consists of a simultaneous routing and bandwidth allocation problem for flows realizing a set of source-destination demands. We consider the case where demands are elastic and the flow realizing one demand can be arbitrary split over multiple paths. Moreover, we ensure that the allocated flows are survivable to network failures using path protection. We investigate the possibility of sharing backup capacity, thus enabling more efficient resource consumption. Depending on the optimization objective, we provide two different approaches to the problem. The first approach maximizes the total allocated volume and the second one maximizes the minimum allocated volume. We build mathematical formulations based on generalizations of the maximum multicommodity flow and the maximum concurrent flow problems. Moreover, we provide new approximation algorithms based on a primal-dual approach. These algorithms compute a solution that is within a guaranteed factor of the optimal. Numerical results enable to get deeper insights on the quality and compared performance of the different approaches.

I. INTRODUCTION

Once capacities have been assigned to the Next Generation Internet (NGI) core links in the capacity design phase, network engineers can tackle the flow allocation problem. The challenge is to determine how much traffic of each demand should be admitted in the network and how this traffic should be routed in order to satisfy the requirement of high network utilization while guaranteeing fairness to the users. Thus, the optimization objective of the flow allocation problem depends on the provisioning strategy chosen by network engineers. Moreover, survivability requirements are introduced at this level of the network optimization. Backup capacity is provisioned in order to provide protection for allocated flows against network failures. Hence, considering both the provisioning strategy and the survivability mechanism, we end up with various formulations for the flow allocation problem.

In this work, we study the survivable flow allocation problem with dedicated and shared path protection. We consider the case where demands are elastic, in the sense that they can employ any volume of resources they are assigned, and the flow realizing one demand can be arbitrary split over multiple paths. The main contribution of this paper is to provide two different formulations illustrating different provisioning strategies. The first approach maximizes the total allocated volume, given by the sum of allocated flows for all demands. The second

approach maximizes the minimum allocated volume. We build mathematical formulations based on generalizations of the maximum multicommodity flow and the maximum concurrent flow problems. Moreover, we provide new approximation algorithms based on a primal-dual approach. These algorithms compute a solution that is within a guaranteed factor of the optimal.

Our paper is organized as the following: in Section II, we introduce the context of our work, emphasizing the provisioning strategies and the survivability requirements. Then, in Section III, we present the mathematical formulation and the corresponding notation of the survivable flow allocation problems. In Section IV, we provide the approximation algorithms that compute efficient solutions to the problems presented in Section III. Moreover, we present an overview over the related work in section V. Finally, in Section VI, we present relevant simulation results, and conclude in Section VII with insights on our future work.

II. CONTEXT AND MOTIVATION

In this section, we review the major key elements involved in the survivable flow allocation problem. We briefly present the survivability mechanisms and the provisioning strategies in use.

a) Survivability: NGI network operators are more and more solicited to ensure a minimal level of service disruption during network failures. This requirement can be specified as part of an extended Service Level Agreement (SLA). Moreover, many mechanisms have been developed in order to satisfy the survivability requirement. Generally speaking, protection mechanisms consist in provisioning backup capacity that can be used by the failed traffic in order to survive the failure. These mechanisms can be broadly classified into link protection and path protection. In the link protection mechanism, the protected entity is one single link. A backup path is provided to recover the traffic on a failed link. In the path protection mechanism, used in this work, the protected entity, called primary path, spans from the source node to the destination node. A backup path is provided to recover the failed traffic on the primary path, when any of its links fails. Primary and backup paths must be link disjoint, so that no single link failure can affect both paths. Moreover, node failures can be taken into account by requiring node disjointness.

Path protection provides good utilization of backup resources compared to link protection where the protected entity is a single link. However, timeliness of path protection can become a real issue as the switching between primary and backup resources is done at the source node.

Protection mechanisms can also be classified according to the allocation type of the backup resources. Therefore, two types of protection are defined: dedicated and shared protection. The difference between the two lies in whether or not resource sharing is possible between different protection paths. The idea is to share the backup resources between protected entities that are not affected in the same failure situation. In dedicated protection, a protecting entity is dedicated to one single protected entity. This results in a very fast restoration service while the ratio of redundancy (i.e., the ratio of capacity taken by protection and working paths in the network) usually reaches 100% [1]. On the other hand, shared protection can substantially reduce the ratio of redundancy required to achieve 100% restorability at the expense of a little longer restoration time.

b) Provisioning Strategies: With the rapid growth of Internet traffic, the simple over-provisioning has become less economical. Therefore, one big challenge for network engineers is to determine how much traffic of each demand should be admitted to the network and how this traffic should be routed in order to satisfy the requirement of high network utilization and to guarantee fairness to the users. One simple strategy consists in maximizing the total admitted traffic. This approach ensures maximum resource utilization. However, it does not guarantee any fairness between different demands. This becomes a real issue when demands belong to different users and allocated volumes can influence their satisfaction. Hence, another strategy consists in maximizing the minimum allocated volume for all demands, thus ensuring simple fairness between the demands.

III. NOTATION AND PROBLEM FORMULATION

In the following, we consider a capacitated network represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with \mathcal{V} being the set of vertices ($|\mathcal{V}| = V$) and \mathcal{E} the set of links ($|\mathcal{E}| = E$). The capacity for each $e \in \mathcal{E}$ is denoted by C_e . This is the total bandwidth that can be routed on link e . Moreover, we denote by \mathcal{D} the set of directed demands ($|\mathcal{D}| = D$). The allocated volume for demand d is realized by means of a survivable flow allocated to cycles joining the source and the destination node of demand d . The cycle set corresponding to demand d is denoted by \mathcal{K}_d and includes K_d pairs of link-disjoint paths. This ensures the flow survivability under single link failures. Particularly, the cycle incidence relation is represented by the constant α_{edk} that equals 1 if link e belongs to cycle $k \in \mathcal{K}_d$ realizing demand d and 0, otherwise. Moreover, the flow realizing demand d and allocated to cycle k is denoted by f_{dk} .

A. Flow Allocation with Dedicated Path Protection

In this section, we provide two different formulations for the flow allocation problem with Dedicated Path Protection (DPP) each one realizing a different optimization objective.

LP 1 MS-DPP

$$\text{Maximize} \quad \sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} f_{dk} \quad (1)$$

Subject to:

$$\sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} \alpha_{edk} f_{dk} \leq C_e, \quad \forall e \in \mathcal{E} \quad (2)$$

$$f_{dk} \geq 0, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K}_d \quad (3)$$

1) Maximizing the Total Allocated Volume: The first approach, referred to as MS-DPP, maximizes the total allocated volume. The correspondent mathematical problem is presented in LP 1. This problem is a generalization of the maximum multicommodity flow. Using the notation introduced above, the objective of the problem expressed in (1) consists in maximizing the sum of the cycle flow values f_{dk} , for all the demands and all the cycles realizing these demands respectively. Moreover, constraint (2) ensures that the allocated traffic on link e ($\forall e \in \mathcal{E}$) does not exceed its capacity.

2) Maximizing the Minimum Allocated Volume: The second approach, referred to as MM-DPP, maximizes the minimum allocated volume. The correspondent mathematical problem is presented in LP 2. This problem is a generalization of the maximum concurrent flow. The objective, expressed in (4), consists in maximizing the minimum allocated volume for all demands. This minimum allocated volume is denoted by F and verifies constraint (5). Moreover, constraint (6) ensures that the allocated traffic on link e ($\forall e \in \mathcal{E}$) does not exceed its capacity.

LP 2 MM-DPP

$$\text{Maximize} \quad F \quad (4)$$

Subject to:

$$\sum_{k \in \mathcal{K}_d} f_{dk} \geq F, \quad \forall d \in \mathcal{D} \quad (5)$$

$$\sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} \alpha_{edk} f_{dk} \leq C_e, \quad \forall e \in \mathcal{E} \quad (6)$$

$$f_{dk} \geq 0, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K}_d \quad (7)$$

B. Flow Allocation with Shared Path Protection

In order to add sharing to the flow allocation problems, we consider S single failure scenarios where $s = 0$ is the non failure situation and $1 \leq s \leq S$ corresponds to the failure of link e where $e = s$ (we get $S = 1 + E$). Moreover, θ_{dks} denotes the availability of the primary path in cycle $k \in \mathcal{K}_d$

realizing demand d under situation s . Hence, θ_{dks} equals 1 if the primary path of cycle k is available in situation s (i.e. all the links in this path are available) and 0, otherwise. We introduce new constants: α_{edk}^p (resp. α_{edk}^b) that equals 1 if link e belongs to the primary (resp. backup) path of cycle $k \in \mathcal{K}_d$ realizing demand d and 0, otherwise. For a compact notation, we will use $\bar{\theta}_{dks}$ to denote the value of $1 - \theta_{dks}$.

LP 3 MS-SPP

$$\text{Maximize } \sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} f_{dk} \quad (8)$$

Subject to:

$$\sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} (\theta_{dks} \alpha_{edk}^p + \bar{\theta}_{dks} \alpha_{edk}^b) f_{dk} \leq C_e, \quad \forall e \in \mathcal{E}, \forall s \in \mathcal{S} \quad (9)$$

$$f_{dk} \geq 0, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K}_d \quad (10)$$

1) *Maximizing the Total Allocated Volume:* MS-SPP is obtained by modifying the capacity constraint (2) in the dedicated version of the problem. Now, under situation s , the total flow on link e is given by the sum of the working flows on link e that are available under situation s and the backup flows that are activated under the same situation (i.e. recovering the failing working flows). This total flow must not exceed the capacity of link e under any situation s . Hence the new capacity constraint is given in (9) and the problem formulation in LP 3.

LP 4 MM-SPP

$$\text{Maximize } F \quad (11)$$

Subject to:

$$\sum_{k \in \mathcal{K}_d} f_{dk} \geq F, \quad \forall d \in \mathcal{D} \quad (12)$$

$$\sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} (\theta_{dks} \alpha_{edk}^p + \bar{\theta}_{dks} \alpha_{edk}^b) f_{dk} \leq C_e, \quad \forall e \in \mathcal{E}, \forall s \in \mathcal{S} \quad (13)$$

$$f_{dk} \geq 0, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K}_d \quad (14)$$

2) *Maximizing the Minimum Allocated Volume:* Following the same reasoning, MM-SPP is obtained by replacing the capacity constraint (6) in LP 2 by constraint (9) in LP 3.

IV. APPROXIMATION ALGORITHMS

Solving the linear programs presented in the previous section is not viable in practice. Particularly, when considering large topologies with a large number of demands, these linear programs can become very difficult to solve and solutions will heavily depend on the solver package. In this work, we develop approximation algorithms based on a primal-dual approach. This approach uses a basic property of the duality theory in

linear programming: considering a maximization problem, the objective value of any feasible solution to the primal problem is at most the objective value of any feasible solution of its dual. Therefore, by comparing the objective values of the primal and dual formulations, we can measure the closeness of a solution (for the primal problem) to the optimal one. By this, we obtain ρ -approximations to the problem: for a maximization problem, a ρ -approximation is an algorithm that computes a solution with an objective value $A(I)$ verifying $A(I) \geq \rho \text{OPT}(I)$, where $\text{OPT}(I)$ is the optimal objective value and ρ a real number from the interval $[0, 1]$. Sometimes ρ is given in the form $\rho = 1 - \epsilon$. Note that, there has been considerable work on approximation for multicommodity problems: the paper in [2] presents an excellent overview over the work. Particularly, our work will be mainly based on the algorithms and analysis in [3].

A. MS-DPP

In order to adapt the primal-dual approach for MS-DPP, we introduce in LP 5 the dual problem corresponding to LP 1. In this dual problem, ω_e denotes the dual variables associated with constraint (2) in LP 1. With some intuition, let us consider that ω_e is a weight associated with link e . ω_e represents also the marginal cost of using an additional unit of bandwidth from the link capacity. Similarly, we suppose that the weight of one cycle is the sum of the weights for its component links. Then, constraint (16) says that the weight for each cycle associated with demand d is at least 1. This observation will contribute in formulating the approximation for MS-DPP.

LP 5 MS-DPP: Dual problem

$$\text{Minimize } \sum_{e \in \mathcal{E}} C_e \omega_e \quad (15)$$

Subject to:

$$\sum_{e \in \mathcal{E}} \alpha_{edk} \omega_e \geq 1, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K}_d \quad (16)$$

$$\omega_e \geq 0, \quad \forall e \in \mathcal{E} \quad (17)$$

Algorithm 1 computes a ρ -approximation to the MS-DPP problem. The algorithm starts by assigning a weight $\omega_e = \delta$ for each link in the network (δ is chosen depending on ϵ and is defined later in Theorem 1). Then, we define in (18) the set \mathcal{K}^* of cycles with weight less than 1. The algorithm loops until \mathcal{K}^* becomes empty.

$$\mathcal{K}^* = \{k \in \mathcal{K}_d, \forall d \in \mathcal{D} / \omega_k = \sum_{e \in \mathcal{E}} \alpha_{edk} \omega_e < 1\} \quad (18)$$

At each iteration, the algorithm selects the shortest cycle k^* in \mathcal{K}^* (let us say k^* realizes demand d i.e. $k^* \in \mathcal{K}_d$) with respect to weights ω_e , and increases both the primal and the dual variables associated with this cycle: for the primal problem, the algorithm augments flow along this cycle. Note that shortest cycle computation is done using the algorithm introduced in

[4]. The amount of flow sent along cycle k^* is determined by the bottleneck capacity of the cycle, using the *original* capacities. The bottleneck capacity of a cycle is the capacity of the minimum capacity link in the primary and the backup paths. If we denote this capacity by c . The primal solution is updated by setting $f_{dk^*} = f_{dk^*} + c$ (this solution can be scaled to become feasible). Then, the dual variables are updated so that the weight of a link is exponential in the congestion of the link. The update function for each link e on k^* is defined as $\omega_e = \omega_e(1 + \epsilon \frac{c}{C_e})$. This update ensures that the weight of the bottleneck link on k^* increases by a factor of $(1 + \epsilon)$ (the dual solution can also be scaled to become feasible). Thus, at the end of every iteration we have implicit primal and dual feasible solutions. The algorithm terminates when all paths have a weight that is greater or equal to 1, i.e. \mathcal{K}^* is empty. Note that, the final solution may be infeasible since we are using the original capacities. Thus, the total allocated flow $f_t = \sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} f_{dk}$ is scaled according to Lemma 1 and yields a feasible solution to the MS-DPP problem. Moreover, the approximation guarantee and the corresponding running time of Algorithm 1 are given in Theorem 1.

Algorithm 1 Approximation Algorithm for MS-DPP

```

1:  $t = 0$ ;  $\omega_e = \delta$ ,  $\forall e \in \mathcal{E}$ 
2: while  $\mathcal{K}^*$ , given by (18), is not empty do
3:   Let  $k^* \in \mathcal{K}_d$  be the shortest cycle in  $\mathcal{K}^*$ 
4:    $c = \min\{C_e : \forall e \in k^*\}$ 
5:    $f_{dk^*} = f_{dk^*} + c$  and  $\omega_e = \omega_e(1 + \epsilon \frac{c}{C_e})$ ,  $\forall e \in k^*$ 
6:    $t = t + 1$ 
7: end while

```

Lemma 1 *At the end of Algorithm 1, the total allocated flow $f_t = \sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} f_{dk}$ need to be scaled by $\log_{1+\epsilon}(\frac{1+\epsilon}{\delta})$ in order to become feasible.*

Theorem 1 *Assuming that $\epsilon = 1 - \rho^{\frac{1}{2}}$ and $\delta = (1 + \epsilon)((1 + \epsilon)E)^{-1/\epsilon}$, Algorithm 1 computes a ρ -approximation to MS-DPP in time $O(EDT_{mwc} \log_{1+\epsilon} \frac{1+\epsilon}{\delta})$ where T_{mwc} is the time required to compute a shortest cycle.*

Full proofs for the above lemma and theorem are reported in [5] and are omitted in this work due to page limitations.

B. MM-DPP

As we proceeded for the MS-DPP, we formulate the dual of LP 2 in LP 6. In this dual problem, π_d denotes the dual variables associated with constraint (5) and ω_e the dual variables associated with constraint (6) in LP 2. Constraint (21) says that π_d is the minimum weight for the cycles associated with demand d . Technically, the primal-dual algorithm, presented in Algorithm 2 starts by assigning a weight of $\frac{\delta}{C_e}$ to link e . The algorithm alternates between primal flow variables and dual weight variables to satisfy the capacity constraint in (6) and the weight constraint in (21). Particularly, the algorithm proceeds in phases. In each phase and for each demand d , the

LP 6 MM-DPP: dual problem

$$\text{Minimize } \Omega = \sum_{e \in \mathcal{E}} C_e \omega_e \quad (19)$$

Subject to:

$$\sum_{d \in \mathcal{D}} \pi_d \geq 1 \quad (20)$$

$$\sum_{e \in \mathcal{E}} \alpha_{edk} \omega_e \geq \pi_d, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K}_d \quad (21)$$

$$\omega_e \geq 0, \quad \forall e \in \mathcal{E} \quad (22)$$

shortest cycle $k^* \in \mathcal{K}_d$ is calculated using the last values of weights ω_e . Shortest cycle computation is also done using the algorithm introduced in [4]. The algorithm allocates 1 unit of flow between source and destination. Once the flow is sent, the weight of the links that carry the flow is increased, based on an exponential function. Therefore, the algorithm alternates between sending flow along shortest cycle and adjusting the weight of the links along which flow has been sent until the optimal solution is reached. The entire procedure ends as soon as $\Omega > 1$, where Ω denotes the objective of the dual problem LP 6. Finally, Theorem 2 states the polynomial performance of the above algorithm. Note that, at the end of the iterations, the allocated flow is infeasible for the primal problem and thus need to be scaled down according to Lemma 2.

Algorithm 2 Approximation algorithm for MM-DPP

```

1:  $t = 0$ ;  $\omega_e = \frac{\delta}{C_e}$ ,  $\forall e \in \mathcal{E}$ 
2: while  $\Omega < 1$  do
3:   for  $d = 1, \dots, D$  do
4:     Let  $k^*$  be the shortest cycle in  $\mathcal{K}_d$ 
5:      $f_{dk^*} = f_{dk^*} + 1$ 
6:      $f_e = f_e + 1$  and  $\omega_e = \omega_e(1 + \frac{\epsilon}{C_e})$ ,  $\forall e \in k^*$ 
7:   end for
8:    $t = t + 1$ 
9: end while

```

Lemma 2 *At the end of Algorithm 2, $F = \frac{t-1}{\log_{1+\epsilon}(\frac{1}{\delta})}$.*

Theorem 2 *Assuming that $\epsilon = 1 - \rho^{\frac{1}{3}}$ and $\delta = (\frac{E}{1-\epsilon})^{-1/\epsilon}$, Algorithm 2 computes a ρ -approximation to MM-DPP in time $O(2D \log D \left\lceil \frac{1}{\epsilon} \log_{1+\epsilon} \frac{E}{1-\epsilon} \right\rceil T_{mwc})$, where T_{mwc} is the time required to compute a minimum weight cycle.*

Full proofs and additional remarks for the above lemma and theorem are reported in [5] and are omitted in this work due to page limitations.

C. Sharing Backup Bandwidth: MS-SPP and MM-SPP

When backup bandwidth is shared, the results obtained for the dedicated case could not be directly extended. Let

us first present the approximation algorithm for MM-SPP. In order to formulate the corresponding approximation, we introduce the dual problem in LP 7 where ω_{es} denotes the dual variable associated with constraint (13). With some care while extending the results, the same analysis can be carried for this problem. The approximation, presented in Algorithm 3 follows the same sketch as for the dedicated protection case. It alternates between allocating flows to shortest cycles and updating the weights of the links on these cycles. However, two main differences are worth to be mentioned.

- 1) The shortest cycle computation in step 4 requires additional attention. Considering that ω_{es} is a weight associated with link e in situation s , the weight of one cycle is expressed as the sum of two components: a first component that adds up the weights for the links in the primary path, whenever this path is available, and a second component that adds up the weights for the links in the backup path, whenever the corresponding primary path is failing. Then, finding a shortest cycle with respect to the above mentioned weighting is not an easy task: a detailed review of the field is provided in [5]. Thus, in order to remedy to this problem, we use a *brute-force* approach that assumes that the sets of cycles for all demands are precomputed. Particularly, this cycle set (of exponential size) can be reduced by limiting the hop number, following, for instance, some quality of service constraints. Then, the shortest cycle that minimizes (25) is found by simple comparison over the computed set. Note that the cycle set computation time is not accounted in our algorithm.
- 2) The weight update is done as in the following: the weight, ω_{es} , corresponding to link e in situation s , is updated if e belongs to the primary path (resp. the backup path) of the shortest cycle, and the primary path is available (resp. failing) in situation s .

Lemma 3 *At the end of Algorithm 3, $F = \frac{t-1}{\log_{1+\epsilon}(\frac{1}{\delta})}$.*

Theorem 3 *Assuming that $\epsilon = 1 - \rho^{\frac{1}{3}}$ and $\delta = (\frac{E(E+1)}{1-\epsilon})^{-1/\epsilon}$, Algorithm 3 computes a ρ -approximation to the MM-SPP in time $O\left(2D \log D \left\lceil \frac{1}{\epsilon} \log_{1+\epsilon} \frac{E(E+1)}{1-\epsilon} \right\rceil T_{mwc}\right)$, where T_{mwc} is the time required to compute a shortest cycle.*

In Lemma 3, we report the scaling that yields a feasible primal solution, and in Theorem 3, we report the performance of Algorithm 3. In Theorem 3, the time required to compute a minimum weight cycle is dominated by a linear search procedure on the precomputed set of cycles. Let us denote K the maximum number of cycles realizing a demand $d \in \mathcal{D}$, i.e. $K = \max_{d \in \mathcal{D}} K_d$. Then the linear search procedure that computes the minimum weight cycle runs in $O(K)$ and the global running time of Algorithm 3 is $O\left(2D \log D \left\lceil \frac{1}{\epsilon} \log_{1+\epsilon} \frac{E(E+1)}{1-\epsilon} \right\rceil K\right)$. The same observations apply to the MS-SPP problem, and an extended presentation is provided in [5]. Moreover, in [5],

we introduce an efficient procedure that enables to tackle the computational complexity of the linear search procedure.

LP 7 MM-SPP: Dual problem

$$\text{Minimize } \Omega = \sum_{e \in \mathcal{E}} \sum_{s \in \mathcal{S}} C_e \omega_{es} \quad (23)$$

Subject to:

$$\sum_{d \in \mathcal{D}} \pi_d \geq 1 \quad (24)$$

$$\sum_{e \in \mathcal{E}} \sum_{s \in \mathcal{S}} (\theta_{dks} \alpha_{edk}^p + \bar{\theta}_{dks} \alpha_{edk}^b) \omega_{es} \geq \pi_d, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K}_d \quad (25)$$

$$\omega_{es} \geq 0, \quad \forall e \in \mathcal{E}, \forall s \in \mathcal{S} \quad (26)$$

Algorithm 3 Approximation algorithm for MM-SPP

- 1: $t = 0$; $\omega_{es} = \frac{\delta}{C_e}, \quad \forall e \in \mathcal{E} \forall s \in \mathcal{S}$
 - 2: **while** $\Omega < 1$ **do**
 - 3: **for** $d = 1, \dots, D$ **do**
 - 4: Let $k^* = (p^*, b^*)$ be the shortest cycle in \mathcal{K}_d
 - 5: **for** $s = 1, \dots, S$ **do**
 - 6: **if** $\theta_{dk^*s} = 1$ **then**
 - 7: $f_{es} = f_{es} + 1$ and $\omega_{es} = \omega_{es}(1 + \frac{\epsilon}{C_e})$, $\forall e \in p^*$
 - 8: **else**
 - 9: $f_{es} = f_{es} + 1$ and $\omega_{es} = \omega_{es}(1 + \frac{\epsilon}{C_e})$, $\forall e \in b^*$
 - 10: **end if**
 - 11: **end for**
 - 12: **end for**
 - 13: $t = t + 1$
 - 14: **end while**
-

V. RELATED WORK

Flow allocation with path protection mechanisms has been investigated in a number of recent papers. Particularly, three relevant papers [6], [7] and [8] are similar to our work. First, in [6], the authors introduce algorithms for capacity dimensioning for optical networks. The objective is to size the demand volume in order to satisfy the capacity constraints. The authors use a path protection mechanism. However, sharing is only enabled between the flows realizing the same demand. Based on a maximum concurrent flow formulation, they provide a ρ -approximation algorithm that runs in time polynomial in the problem size. Second, in [7], the authors introduce a model for the computation of a minimum cost multicommodity flow with a path protection mechanism. Full sharing is supported but the primary flows are allocated on a set of disjoint paths that are precomputed for each demand. The authors provide a ρ -approximation algorithm and the running time depends on the number of disjoint paths. Third, [8] investigated very recently the problem of survivable flow allocation with full sharing between the demands. A preliminary step computes the set

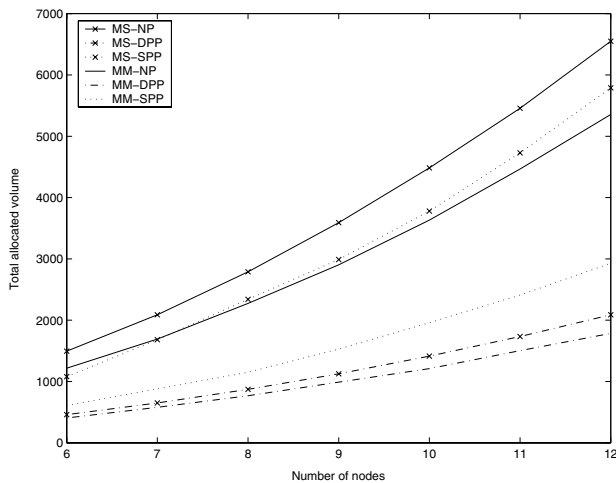


Fig. 1. Comparison of bandwidth consumption

of failure disjoint paths and feed the approximation algorithm (note that the corresponding problem for the precomputation step is \mathcal{NP} -hard). The authors provide a $\rho/2$ -approximation algorithm with the running time depending on the number disjoint pair of paths in the network. They suggest reducing the running time by reducing the precomputed set of paths. A broader view of the state of the art, that also cover general approximations for multicommodity flow problems is presented in [5].

VI. NUMERICAL RESULTS

To illustrate the performance of our algorithms, we performed a series of experiments over simulation topologies. We consider complete graph topologies. In graph theory, a complete graph is a simple graph where an link connects every pair of vertices. Certainly, complete graphs were not chosen for simulation as realistic topologies. Nevertheless, as complete graphs are maximally connected (i.e. the only vertex cut which disconnects the graph is the complete set of vertices or roughly saying, they have many alternate paths), they will illustrate the extreme performance points in our experiments. In all our simulations, each link is assumed to have a capacity of 100 units and demands are considered between all the nodes in the graph. Figure 1 illustrates the compared performance for the different algorithms presented in section IV. It expresses the total allocated volume for each algorithm as a function of the node number of the complete graph. Note that, results concerning provisioning for non protected flows are included for reference, although not directly introduced in this paper. These are annotated MS-NP and MM-NP for the solution that maximizes the sum of the allocated flows and the minimum allocated demand respectively, both with no protection requirements. Figure 1 shows that the gap between the total allocated volume for MS-DPP and for MS-SPP is very large: around 1800 bandwidth units for the 9-node topology. This is not the case when we compare the allocation for MM-DPP and MM-SPP. Intuitively,

this is due to the fact that sharing becomes less efficient when the objective is to guarantee some fairness between demands (as for MM problems). However, we note that for the two couples of problems, the gap keeps on increasing when the number of nodes increases. This is a normal result of large highly-connected topologies that offer more paths for allocation and thus more possibilities to share backup flows. Finally, the no-protection problems achieve, as expected, the highest allocated volume at the expense of high vulnerability to network failures.

VII. CONCLUSION

In this work, we provide ρ -approximations for flow allocation problems with end-to-end protection. While a large number of papers have provided *efficient* solutions to similar problems, this paper concentrates on guaranteed approximations. Moreover, the originality is increased by comparing for the first time two different approaches for network provisioning and evaluating the impact of fairness. Therefore, future work will concentrate on providing an extension to the max-min fair problems that enable to approximate the global optimal solution. Moreover, we will study, in a similar work, algorithms for flow allocation problems with local protection. An extended version of this work is provided in [5] containing additional simulation results and numerical evaluation as well as the full mathematical proofs. Moreover, in a recent work, we provide efficient methods that tackle the computational complexity of the shared path protection versions.

REFERENCES

- [1] P.-H. Ho, J. Topolcai, and H. Mouftah, "On achieving optimal survivable routing for shared protection in survivable next-generation internet," *Reliability, IEEE Transactions on*, vol. 53, no. 2, pp. 216–225, 2004.
- [2] L. Fleischer, "Approximating fractional multicommodity flow independent of the number of commodities," *SIAM J. Discret. Math.*, vol. 13, no. 4, pp. 505–520, 2000.
- [3] N. Garg and J. Koenemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems." in *FOCS '98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 1998, p. 300.
- [4] J. W. Suurballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, pp. 325–336, 1986.
- [5] S. Lahoud, "Survivable routing and flow allocation in next generation internet networks," Ph.D. dissertation, GET/ENST Bretagne, 2006.
- [6] O. Hauser, M. Kodialam, and T. Lakshman, "Capacity design of fast path restorable optical networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2002, pp. 817–826 vol.2.
- [7] L. Fleischer, A. Meyerson, I. Saniee, B. Sheperd, and A. Srinivasan, "A scalable algorithm for the minimum expected cost restorable flow problem," CORC, Tech. Rep. TR-2003-10, 2004.
- [8] A. Todimala and B. Ramamurthy, "Approximation algorithms for survivable multicommodity flow problems with applications to network design," in *Infocom 2006*, to appear.