

Traffic Engineering of MPLS Backbone Networks in the Presence of Heterogeneous Streams ¹

Shekhar Srivastava², Appie van de Liefvoort, and Deep Medhi³

Computer Science & Electrical Engineering Department

University of Missouri–Kansas City

Kansas City, MO 64110 USA

Original submission: 13 January 2008

Revised: 3 September 2008

Revised (2nd): 3 February 2009

Final Version: 13 June 2009

Abstract

Heterogeneous streams (due to issues such as disparate traffic characteristics of each stream, or competing customers' traffic) raise the issue of whether to multiplex (some of) these streams. In an MPLS network, such multiplexing can be considered by putting different streams into a tunnel identified by a single label switched path (LSP), assuming that the different LSPs are assigned a reserved share of the resources. This issue becomes even more important in the traffic engineering of a backbone network when a decision needs to be made on which streams to multiplex when there are constraints on tunneling and capacity along with routing requirements for tunnels. In this paper, we introduce a distortion factor due to heterogeneous streams in traffic engineering of MPLS backbone networks in the presence of tunneling and capacity constraints by formulating a distortion-aware non-linear discrete optimization problem. Furthermore, we present a two phase heuristic approach to solve this formulation efficiently. In the first phase, the problem is decoupled into two subproblems and in the second phase we show how the non-linear problem (for one of the subproblems) can be simplified. We then present numerical results for both small and large networks to show where and how our approach helps to determine when and which streams to multiplex depending on whether the tunneling and/or capacity constraint is dominant; furthermore, by comparing our distortion-aware traffic engineering model with a distortion-ignorant traffic engineering model, we show the benefits of our approach.

Keywords:

MPLS, Traffic Engineering, Tunneling Constraint, Traffic Distortion, Quadratic/linear integer programming, Decomposition Algorithm.

¹ This work was supported in part by NSF grant # CNS-0106640.

² Current Affiliation: Huawei Technologies, Plano, TX USA.

³ Corresponding Author.

1 Introduction

Traffic, at the packet level for different applications, tends to have different characteristics. This fact has been observed for emerging applications such as video conferencing, peer-to-peer, and multimedia applications [1,4]. Moreover, heterogeneous traffic streams are multiplexed together to share the same link. When traffic of different characteristics are multiplexed together, traffic distortion occurs which can be significant depending on the characteristics of the streams [9, 18]. Specifically, such impacts can be fairly strong when one or more of them are correlated in nature. Besides distortion due to disparate traffic characteristics, a provider might have preference not to combine certain streams because they belong to traffics of competing customers and thus, may want to impose an induced distortion for the purpose of traffic engineering.

Consider now an MPLS (multi-protocol label switching) network with label switched paths, potentially for different heterogeneous streams (for ease of discussion, label switched paths and tunnels will be used interchangeably). We define *tunnel* as the unit of flow to which a certain bandwidth is allocated at each router that it traverses. If multiple paths are being aggregated to share bandwidth, then all of such paths together would be considered a tunnel. In order to minimize distortion, a possible approach would be to classify and multiplex appropriate streams into tunnels so that distortion is minimized within a tunnel. That is, “like-minded” streams can avoid (or minimize) distortion if the network has the capability to do so. For this purpose, one can consider the MPLS network in which multiple label switched path (LSP) tunnels can be set up between source and destination nodes; such tunnels can be used to ensure logical separation between streams in order to minimize distortions. Therefore, we require that the LSPs are setup with a proper amount of allocated bandwidth and that the routers enforce bandwidth separation among the LSPs.

There are a number of ways to decide how to take advantage of the tunneling idea. On one extreme, a separate tunnel for each possible traffic stream can be set up to avoid any distortion. The difficulty with this approach is that the number of tunnels can be prohibitively large. This then impacts packet processing and forwarding at an MPLS router [22] and furthermore, administratively, too many tunnels lead to higher network management costs. On the other extreme, all streams for the same source-destination node pairs can be multiplexed in the same tunnel. Such an allocation will result in streams having a considerably high amount of distortion, while administrative overhead on managing tunnels and lookup cost can be minimized.

1.1 Contributions of the Paper

In this work, we address the trade-off between distortion and manageability of label-switched paths in an MPLS network. The scope of our work resides in short- to medium-term traffic engineering planning where a network engineer needs to play with ‘what-if’ scenarios with newly anticipated demands and customers and what to mix while ensuring that the network has enough resources. manageability of label-switched paths in an MPLS network. Specifically, we address the following question: given information on the characteristics of individual streams and the number of tunnels (LSPs) that can be supported on each link (for example, to avoid impact on the lookup time), can we engineer a network so that the distortion level and the required number of tunnels are both addressed together?

In this paper, using a distortion measure [15] (see the Appendix for a brief summary), we construct an optimization formulation, which minimizes the distortion suffered by individual streams for different source-destination node pairs in a network environment. In addition to the capacity on a link, the formulation also incorporates restrictions in terms of the maximum number of allowed tunnels for manageability. The formulation is general in the sense that it is transparent to the method used to pre-compute the value of the distortion measure; rather, the distortion measure could be induced when considering competing customers. The formulation is a non-linear (quadratic) integer programming problem in nature and has a large number of binary variables and constraints. To solve this formulation, we propose a two-phase approach in which the first phase is solved by relaxing the binary variables and the second phase (non-linear objective function) is first linearized and solved using Lagrangian relaxation and subgradient optimization. Through computational results, we demonstrate the convergence of our approach.

Furthermore, we propose several distortion metrics to measure distortion at the network-wide and the stream level due to flow interactions. Through numerical results on these measures, we demonstrate the utility of our formulation (“Distortion-Aware Model”) towards allocating streams to tunnels while minimizing the distortion suffered by the individual streams, compared to the case when distortion is not explicitly incorporated (“Distortion-Ignorant Model”).

1.2 Related work

Network traffic engineering literature is rich; for example, see the recent book [14] and the references therein. In particular, there have been several works on MPLS traffic engineering [2, 7, 21]. At the same time, there is little work that accounts for distortions in a traffic engineering framework. The work that is perhaps closest to our work is [20], which studied the problem of routing streams to different virtual

paths (VPs), i.e., tunnels. This work addressed whether heterogeneous traffic with different quality-of-service requirements should be segregated on distinct tunnels or aggregated. They claimed that, in some cases, it is advantageous, but not always; they developed a criterion on how to make this decision for Gaussian models. They further commented that it is important to consider heterogeneous statistical multiplexing or traffic mixes in making routing decisions and suggested that multiservice networks may operate like multiple logical networks, which are segregated by service type. We, however, take a different view point. First, we address that it is important to consider tunneling restrictions. Second, we consider a network wide approach and also determine the routing/bandwidth allocation of tunnels for heterogeneous traffic streams through distortion while honoring the capacity and tunneling restrictions. Recently, the notion of limiting tunnels in a traffic engineering formulation for manageability has been addressed in a traffic engineering framework [16, 17]. The issue of tunnels' growth has recently received the attention from IETF [22]. To our knowledge, the distortion factor had not been considered before with the tunneling and capacity requirement in a traffic engineering framework.

The formulated traffic engineering problem is a quadratic integer programming problem. While there are solution approaches available to solve such problems [8, 13], we take advantage of the special structure of the objective function and the constraints to arrive at a simplified solution approach. We draw upon a decoupling heuristic proposed in [12] to decouple the main problem; however, our decoupled subproblems are not related to [12] and are solved exploiting the special structure.

1.3 Organization of the paper

The rest of the paper is organized as follows: In Section 2, we present the traffic engineering formulation as a quadratic integer programming problem. In Section 3, we present our solution approach, which decouples and decomposes the problem into smaller subproblems. In Section 4, we present and discuss numerical results, especially the benefit of distortion-aware models compared to distortion-ignorant models.

2 Traffic Engineering Modeling

2.1 Distortion

We start with our definition of distortion as it will be used for the purpose of traffic engineering modeling.

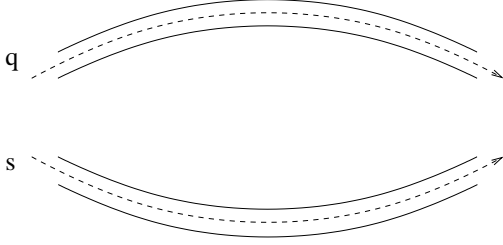


Fig. 1. Allocation of streams q and s only

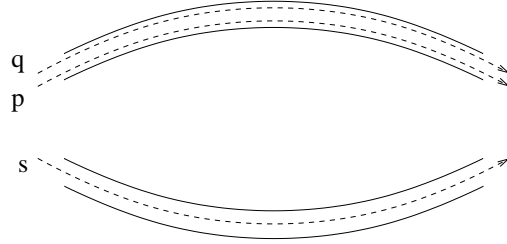


Fig. 2. Allocation of streams p , q , and s

Definition 1 Distortion measure Λ^{pq} is defined between two streams p and q , where $p \neq q$ as $\Lambda^{pq} \geq 0$, and captures the perceived negative impact of the streams on each other.

This means that Λ^{pq} is defined loosely as the extent of negative impact in the traffic streams p and q if they are multiplexed together. Suppose that we are required to route three streams p , q , and s with the same source and destination nodes, and we know the pair-wise relation among streams as $\Lambda^{pq} < \Lambda^{ps} < \Lambda^{qs}$. Suppose then that we only have two tunnels to route the packets from traffic streams p , q , and s . Since Λ^{qs} is the largest of them (i.e., q and s have the largest distortion), we want to route q and s on different tunnels as shown in Figure 1. Next, we want to decide on which tunnel to route the stream p . Since, $\Lambda^{pq} < \Lambda^{ps}$, we route p together with q as shown in Figure 2 and s is routed by itself on the other tunnel. Such a routing ensures minimal impact of distortion in all the streams p , q and s . Certainly, if there were no tunneling restrictions, then each stream would get its own tunnel.

Note that by definition, a network engineer would want to operate the network at a lower value of Λ^{pq} . The characterization of distortion measure can be based on many different requirements. In the Appendix, we present one possible approach of constructing the distortion measure. In the presented example, distortion is based on the departure distribution of the streams upon leaving the network, and it increases as the extent of corruption with regard to a Poisson source is increased. To see another application, consider a service provider supporting multiple customers on a common integrated MPLS/IP network. Some of these customers could be competitors of each other and hence would object to sharing a tunnel with each other, while others, such as banks providing financial services, would not like to share the traffic with anyone else. In general, for increasing the degree of separation, some might be willing to pay extra charges while others may be willing to accept sharing to some extent but prefer separation. Developing such a distortion index would require closely understanding the relationships between the customers and the services that they are providing. Such an application, however interesting, is outside the scope of this paper.

Table 1
Notations used in Traffic Engineering Formulation

\mathcal{N}	:	Set of Nodes in the Network
\mathcal{K}	:	Set of demand pairs generating traffic in the network
\mathcal{L}	:	Set of links in the network
\mathcal{Q}_k	:	Set of streams for demand pair $k \in \mathcal{K}$
\mathcal{P}_k	:	Set of candidates paths for demand pair $k \in \mathcal{K}$
ζ_k	:	Revenue from carrying unit demand volume of demand pair $k \in \mathcal{K}$
λ_k^p	:	Average Arrival rate (in Mbps) of stream $p \in \mathcal{Q}_k$, $k \in \mathcal{K}$
Λ_{km}^{pq}	:	Distortion measure between stream $p, q \in \mathcal{Q}_k$ on path $m \in \mathcal{P}_k$ for $k \in \mathcal{K}$
T_ℓ	:	Maximum number of tunnels allowed on link $\ell \in \mathcal{L}$
C_ℓ	:	Capacity of link $\ell \in \mathcal{L}$ (in Mbps)
φ_k, θ_k	:	Normalization parameters
<i>Variables:</i>		
x_{km}^p	:	0/1 decision variable associated with stream $p \in \mathcal{Q}_k$, path $m \in \mathcal{P}_k$
w_{km}	:	Tunnel activity (binary) variable associated with pair $k \in \mathcal{K}$, path $m \in \mathcal{P}_k$

2.2 Formulation

We now consider a network where heterogenous streams arriving at a source for a specific destination need to be sent over one of the active tunnels (label switched paths) between the source and the destination (see Table 1 for notations) with a goal to minimize distortion. Each source-destination pair maintains its own set of tunnels.

The following parameters are assumed to be given: \mathcal{N} , \mathcal{K} , \mathcal{L} , C_ℓ , T_ℓ , \mathcal{Q}_k and ζ_k . For every stream $p \in \mathcal{Q}_k$, we assume that average arrival rate λ_k^p for stream p for pair k is available, and the distortion measure Λ_{km}^{pq} between streams p and q on path m for pair k is pre-computed (see the Appendix) or is available through other mechanisms for consideration in the traffic engineering formulation.

We also assume that a path generator (such as the k-shortest path algorithm) is used to generate the set of possible paths, \mathcal{P}_k , which will be candidates for tunnels. Let $|\mathcal{P}_k|$ be the number of candidate paths generated for demand $k \in \mathcal{K}$. We introduce the *decision variable* x_{km}^p associated with the stream $p \in \mathcal{Q}_k$, path $m \in \mathcal{P}_k$ of demand $k \in \mathcal{K}$. Its value is 1 if the traffic of stream p selects path m ; otherwise, its value is 0. Due to capacity restriction, it is quite possible that a demand may not be routed (see Section 2.3). Considering these aspects, we arrive at the following flow

constraints

$$\sum_{m \in \mathcal{P}_k} x_{km}^p \leq 1, \quad p \in \mathcal{Q}_k, k \in \mathcal{K} \quad (1a)$$

$$x_{km}^p \in \{0, 1\}, \quad p \in \mathcal{Q}_k, m \in \mathcal{P}_k, k \in \mathcal{K} \quad (1b)$$

To address the flow of the stream of a demand on each link (for each path), we introduce the indicator notation, δ , to map between the demand and the link, as they relate to the possible paths as follows:

$$\delta_{km}^\ell = \begin{cases} 1 & \text{if path } m \text{ of demand } k \text{ uses link } \ell \\ 0 & \text{otherwise.} \end{cases}$$

The bandwidth allocated to a tunnel is determined by the average arrival rate, λ (sometimes referred to as the sustained arrival rate), of the streams sharing the tunnel, measured in Mbps. Thus, the bandwidth needed on any link ℓ (denoted by F_ℓ) to carry flow for different demands can be captured by the amount

$$F_\ell = \sum_{k \in \mathcal{K}} \sum_{m \in \mathcal{P}_k} \sum_{p \in \mathcal{Q}_k} \delta_{km}^\ell \lambda_k^p x_{km}^p$$

For ease of representation, we will henceforth use the tuple (k, m, p) to mean $k \in \mathcal{K}, m \in \mathcal{P}_k, p \in \mathcal{Q}_k$, along with the compact summation notation $\sum_{(k,m,p)}$. Since each link ℓ has capacity C_ℓ , the following capacity constraints for each link $\ell \in \mathcal{L}$ must be satisfied:

$$\sum_{(k,m,p)} \delta_{km}^\ell \lambda_k^p x_{km}^p \leq C_\ell, \quad \ell \in \mathcal{L} \quad (1c)$$

To incorporate the restriction of limiting the number of active tunnels on any link ℓ to T_ℓ , we introduce a variable that captures the decision on whether any given path as a tunnel is being selected or not. The value taken by such a variable depends on the value taken by the corresponding flow variables. We define w_{km} as the tunnel activity variable; if it is one, then any of the streams can use the path and if it is zero, the path is not active and cannot be used. Such a dependency relation can be achieved through the following constraint:

$$x_{km}^p \leq w_{km}, \quad (k, m, p) \quad (1d)$$

Thus, when w_{km} is 0, constraint (1d) forces x_{km}^p to be 0 for all p ; on the other hand, when w_{km} is 1, x_{km}^p could be either 0 or 1. The following tunnel constraint limits the number of active tunnels on a link:

$$\sum_{(k,m)} \delta_{km}^\ell w_{km} \leq T_\ell, \quad \ell \in \mathcal{L}. \quad (1e)$$

$$w_{km} \in \{0, 1\}, \quad (k, m) \quad (1f)$$

Thus, constraints for the traffic engineering problems are as given from (1a) to (1f). There are, however, multiple objectives to be considered. One objective is to maximize the total revenue generated by the flow carried by the network while a second objective is to minimize the overall mismatch due to traffic streams sharing tunnels in the network. Furthermore, a third objective is to minimize the cost of allocation and maintenance of the tunnels. We show below how to construct a composite objective function based on these three objectives. The first objective, total revenue generated, can be written as:

$$f_r = \sum_{(k,m,p)} \zeta_k \lambda_k^p x_{km}^p. \quad (2)$$

Note that we encounter distortions only when for demand k there are any two disparate streams p and q , which share the same path m . In other words, only when $x_{km}^p = x_{km}^q = 1$, a cost is incurred due to the mismatch given by the distortion, Λ_{km}^{pq} . Summing it up across all the demands and all streams, we get

$$f_d = \sum_{(k,m,p,q)} x_{km}^p \Lambda_{km}^{pq} x_{km}^q \quad (3)$$

where $q \in \mathcal{Q}_k \setminus \{p\}$. The third objective accounts for the routing/maintenance cost of the flows. If c_{km} is the routing cost of the candidate tunnel w_{km} , $m \in \mathcal{P}_k$ and $k \in \mathcal{K}$, then, the total routing cost is

$$f_c = \sum_{k \in \mathcal{K}} \sum_{m \in \mathcal{P}_k} c_{km} w_{km} \quad (4)$$

The cost of routing a tunnel can be based on setup/maintenance costs as well as any forwarding required by the tunnel. Thus, the total routing cost would depend on the volume of flow on the tunnel and the number of hops taken by the tunnel and will be additive in both. Such a cost is captured as

$$c_{km} = \sum_{\ell \in \mathcal{L}} \sum_{p \in \mathcal{Q}_k} \delta_{km}^\ell \lambda_k^p x_{km}^p. \quad (5)$$

In order to accommodate all three objectives listed above, we combine them using normalization parameters as $f = f_r - \varphi f_d - \theta f_c$, where φ and θ (≥ 0) are normalization parameters. By substituting the value of c_{km} from (5) in the combined objective function, and considering the normalization parameters to be demand pair based, we can write the combined objective function as we get

$$f = \sum_{(k,m,p)} \zeta_k \lambda_k^p x_{km}^p - \sum_{(k,m,p,q)} \varphi_k x_{km}^p \Lambda_{km}^{pq} x_{km}^q - \sum_{(k,m,p)} \theta_k \sum_{\ell \in \mathcal{L}} \delta_{km}^\ell \lambda_k^p x_{km}^p w_{km} \quad (6)$$

Note the non-linearity in the third term due to $x_{km}^p w_{km}$. However, when coupled with constraints (1d), it can be reduced to $x_{km}^p w_{km} = x_{km}^p$. Therefore, the traffic

engineering problem, (\mathbf{P}) , for the MPLS backbone network can be written as:

$$\max_{\{\mathbf{x}, \mathbf{w}\}} f = \sum_{(k,m,p)} \left[\zeta_k \lambda_k^p x_{km}^p - \varphi_k \sum_{(q)} x_{km}^p \Lambda_{km}^{pq} x_{km}^q - \theta_k \sum_{(\ell)} \delta_{km}^\ell \lambda_k^p x_{km}^p \right] \quad (7)$$

subject to the set of constraints (1).

2.3 Remarks

It is worthwhile to make several remarks about the traffic engineering model we presented.

First, the incorporation of the tunneling restriction (in addition to the capacity constraint) forces some streams to be multiplexed. A goal of this formulation is to minimize distortion when like-minded streams are multiplexed on a single tunnel.

Second, our formulation is primarily for use in short- to medium-term traffic engineering planning where a network engineer needs to play with ‘what-if’ scenarios with newly anticipated demands and customers and what to mix while ensuring that the network has enough resources. Because of this, we address the possibility that all demands may not be met, which is indicated through the constraints (1a). Furthermore, we consider another goal in this context by addressing maximization of revenue and a third goal of minimizing any routing/maintenance cost; these are incorporated along with the goal to minimize distortion.

Third, our formulation does not address whether the quality-of-service is met in the real-time. With combinatorial requirement, we, however, indicate whether a stream is to be included or not (consider constraints (1b) along with (1a)) so that the quality-of-service is met from a planning point of view if a stream is included. If we relax constraints (1b) to fractional values, this would then mean that a stream with partial demand is allowed to be incorporated, which, in turn, means that some streams would be accommodated but the quality-of-service might suffer more.

Finally, while the revenue component of the objective function would seldom change, and that most such networks are engineered to carry all the available demands, it is also imperative to add the revenue in order to make sure that the formulation does not provide trivial solutions. Consider removing the revenue component from the objective function; then, the formulation would try to minimize just the distortion and tunnel cost subject to tunnel and capacity constraints. In this case, due to the inequality in constraints (1a)), the solution will always be to refuse all the demands, leading to zero distortion and zero tunnel cost, which is an undesirable outcome.

3 Solution Approach

Problem **(P)** is a discrete quadratic optimization problem. There are solution approaches available to solve such problems; for example, see [8, 13]. Although these approaches are useful, we base our solution approach on the special structure of the objective function and the constraints to arrive at a simplified solution approach. This is described next.

3.1 Decoupling into two Phases

Observe that the objective function of Problem **(P)** has two components. The first component needs to be maximized (total revenue) and is integer linear in nature. The second component needs to be minimized (the mismatch between the streams sharing a tunnel and the total tunnel cost) and is quadratic integer in nature. In most practical traffic engineering problems, it is uncommon to refuse a revenue generating demand. Therefore, the total generated revenue will seldom change. Based on this observation, we split the objective function into two parts and use a two-tier approach similar to the one presented in [12]. Here, the decoupled Phase I Problem **(P₁)** maximizes the function

$$f_1 = \sum_{(k,m,p)} \zeta_k \lambda_k^p x_{km}^p. \quad (8)$$

This objective function is considered with the set of constraints (1); the solution of this phase will be denoted by $f_1 = F^*$.

Next, we minimize the second order mismatch between flows sharing a tunnel and the overall tunnel cost while keeping the generated revenue at its previously derived maximum value. The function to be minimized in the Phase II Problem **(P₂)** is

$$f_2 = \sum_{(k,m,p)} \left[\varphi_k \sum_q x_{km}^p \Lambda_{km}^{pq} x_{km}^q + \theta_k \sum_{(\ell)} \delta_{km}^\ell \lambda_k^p x_{km}^p \right] \quad (9)$$

This objective function is considered under the additional restriction

$$\sum_{(k,m,p)} \zeta_k \lambda_k^p x_{km}^p \geq (F^* - X) \quad (10)$$

along with the set of constraints (1).

Table 2
Mapping between variables x and z

x^p	x^q	Cons. on z^{pq}	z^{pq} (value)
0	0	$z \geq -1$	0
0	1	$z \geq 0$	0
1	0	$z \geq 0$	0
1	1	$z \geq 1$	1

3.2 Solving Phase I

The Phase I Problem, \mathbf{P}_1 , is an integer linear programming problem. In our case, observe that Phase I Problem (\mathbf{P}_1) generates the value, F^* , which acts as a bound and creates a constraint for the subsequent Phase II Problem, \mathbf{P}_2 . The generated solutions of x and w from Phase I Problem (\mathbf{P}_1) do not have any implications since they are really meaningful from the solution of Phase II Problem (\mathbf{P}_2). Thus, we take the continuous relaxation of Phase I Problem (\mathbf{P}_1) which is then a linear programming problem and obtain the maximum value F^* ; this value upper bounds the integer version of Phase I Problem (\mathbf{P}_1). Since, the integer solution in the Phase II problem might not exist with the revenue value of F^* . Therefore, we have introduced X , in order to make sure that the problem Phase II remains feasible.

3.3 Linearizing Phase II

The Phase II Problem, \mathbf{P}_2 , is a quadratic integer programming problem. The additional constraint (10) ensures that the values of variables x are only adjusted so that the weighted accepted flow does not decrease from F^* by more than X .

In Phase II Problem (\mathbf{P}_2), the product of x_{km}^p and x_{km}^q takes only two possible values 0, 1; this makes it possible to capture the same behavior using a linear function by using an extra set of variables (z). In fact, this approach builds upon the special structure of the problem—it not only avoids non-linearity but the newly added variable is continuous in nature as well. For brevity, we momentarily drop the subscripts k, m and retain the superscripts for identifying two streams p, q . If we, now, introduce the constraint

$$x^p + x^q - 1 \leq z^{pq}, \quad (11)$$

for every product term $x^p x^q$ and minimize the sum of $\Lambda^{pq} z^{pq}$, then non-linearity can be eliminated. In Table 2, we show the transformation between the non-linear term $x^p x^q$ and the constraint on the value of z^{pq} and the eventual value of variable z^{pq} due to minimizing nature of the problem.

Using this mapping and observing that the value of normalization parameter φ_k does not impact the solution, we build the modified Problem (\mathbf{P}'_2) which minimizes the linear objective function

$$f_2 = \sum_{(k,m,p)} \left[\varphi_k \sum_{(q)} \Lambda_{km}^{pq} z_{km}^{pq} + \theta_k \sum_{(\ell)} \delta_{km}^{\ell} \lambda_k^p x_{km}^p \right] \quad (12)$$

subject to constraints

$$x_{km}^p + x_{km}^q - 1 \leq z_{km}^{pq}, \quad (p, q, m, k) \quad (13a)$$

and

$$0 \leq z_{km}^{pq} \leq 1, \quad (p, q, m, k) \quad (13b)$$

in addition to the set of constraints (1) and (10).

The newly constructed Phase II Problem (\mathbf{P}'_2) is an integer linear programming problem. Consider the size of this problem. Let $m_k = |\mathcal{P}_k|$, $n_\ell = |\mathcal{L}|$ and $n_k = |\mathcal{Q}_k|$. The Phase II Problem (\mathbf{P}'_2) has \mathbf{x} and \mathbf{w} as binary variables and \mathbf{z} as the continuous variables. Thus, the total number of binary variables are $\sum_{(k)} m_k$ (for \mathbf{w}) and $\sum_{(k)} m_k n_k$ (for \mathbf{x}), and the number of continuous variables are $\sum_{(k)} m_k n_k (n_k - 1)$ (for \mathbf{z}). The total number of constraints are $2 n_\ell + \sum_{(k)} m_k n_k n_k$.

In order to effectively solve Formulation (\mathbf{P}'_2), we developed a decomposition algorithm using Lagrangian relaxation [3] with duality and subgradient optimization [5]. This is described next.

3.4 Decomposing Phase II

We observe that constraint (1d) is a coupling constraint between the variables, \mathbf{x} and \mathbf{w} . We thus take Lagrangian relaxation around this constraint using dual variable \mathbf{u} . Here, the Lagrangian function

$$L(\mathbf{x}, \mathbf{w}, \mathbf{z}; \mathbf{u}) = \sum_{(k,m,p)} \left[\varphi_k \sum_{(q)} \Lambda_{km}^{pq} z_{km}^{pq} + \theta_k \sum_{(\ell)} \delta_{km}^{\ell} \lambda_k^p x_{km}^p + u_{km}^p (x_{km}^p - w_{km}) \right]. \quad (14)$$

Rearranging the Lagrangian, we get

$$L(\mathbf{x}, \mathbf{w}, \mathbf{z}; \mathbf{u}) = \sum_{(k,m,p)} \left[\varphi_k \sum_{(q)} \Lambda_{km}^{pq} z_{km}^{pq} + \theta_k \sum_{(\ell)} \delta_{km}^{\ell} \lambda_k^p x_{km}^p + u_{km}^p x_{km}^p - u_{km}^p w_{km} \right] \quad (15)$$

That is, the Lagrangian can be re-written as

$$L(\mathbf{x}, \mathbf{z}, \mathbf{w}; \mathbf{u}) = L_{x,z}(\mathbf{x}, \mathbf{z}; \mathbf{u}) + L_w(\mathbf{w}, \mathbf{u}). \quad (16)$$

The dual Problem (D) is

$$s_D = \max_{\{\mathbf{u} \geq 0\}} g(\mathbf{u}), \quad (17)$$

where

$$g(\mathbf{u}) = \min_{\{\mathbf{x}, \mathbf{z}, \mathbf{w}\}} L(\mathbf{x}, \mathbf{z}, \mathbf{w}; \mathbf{u}). \quad (18)$$

Note that for a given \mathbf{u} , the Lagrangian L is separable in \mathbf{x} , \mathbf{z} on one side and \mathbf{w} on the other, reduces (18) to solving two independent subproblems.

$$\min_{\{\mathbf{x}, \mathbf{z}, \mathbf{w}\}} L = \begin{cases} \min_{\{\mathbf{x}, \mathbf{z}\}} L_{x,z}(\mathbf{x}, \mathbf{z}; \mathbf{u}) + \min_{\{\mathbf{w}\}} L_w(\mathbf{w}; \mathbf{u}) \\ g_{x,z}(\mathbf{u}) \quad + \quad g_w(\mathbf{u}) \end{cases} \quad (19)$$

where,

$$g_{x,z}(\mathbf{u}) = \min_{\{\mathbf{x}, \mathbf{z}\}} \sum_{(k,m,p)} [\varphi_k \sum_{(q)} \Lambda_{km}^{pq} z_{km}^{pq} + \theta_k \sum_{(\ell)} \delta_{km}^{\ell} \lambda_k^p x_{km}^p + u_{km}^p x_{km}^p], \quad (20)$$

which is subject to (1a), (1b), (1c), (10), (13a) and (13b), and

$$g_w(\mathbf{u}) = \min_{\{\mathbf{w}\}} \sum_{(k,m)} \left(- \sum_{(p)} u_{km}^p \right) w_{km}, \quad (21)$$

which is subject to (1e) and (1f).

The above requires solving three smaller problems, namely: s_D , $g_{x,z}(\mathbf{u})$ and $g_w(\mathbf{u})$. In the following subsections, we present our approach to solve these three separate problems, first starting with $g_w(\mathbf{u})$.

3.4.1 Solving $g_w(\mathbf{u})$

Problem $g_w(\mathbf{u})$ is a special case of Multiple Knapsack Problem (MKP). MKP is known to be NP complete. Thus, solving the General MKP problem by direct methods imposes a severe constraint on the scalability of the solution approach. In our case, the volume of each item is equal to 1 or 0 in all the knapsacks and the size of each knapsack is integral, making this special case effectively solvable by faster approaches. There always exists an integral solution for the LP version of MKP, which is based on the result stated below. Moreover, at least one feasible optimum solution of the relaxed $g_w(\mathbf{u})$ is integral and hence the use of the Simplex algorithm

will give us the integral solutions. *RMKP* (Relaxed MKP) can be written as:

$$(RMKP) \left\{ \begin{array}{l} \bar{F} = \max F(\mathbf{w}) = \sum_{j=1}^n c_j w_j \\ \text{subject to} \\ \sum_{j=1}^n \mathbf{A}_j w_j \leq \mathbf{b} \\ 0 \leq w_j \leq 1, \quad (j = 1, 2, \dots, n) \end{array} \right. \quad (22)$$

where \mathbf{A} is an $m \times n$ matrix with binary entries (\mathbf{A}_j is the j^{th} column of \mathbf{A} capturing the a_{ij} 's, the amount of resources required by object j in sack i), \mathbf{w} is an n -vector of variables (w_1, w_2, \dots, w_n), \mathbf{b} an m -vector with nonnegative integer components.

Theorem 1 *Relaxed Multiple Knapsack Problem (RMKP) with $a_{ij} = \{0, 1\}$ and integer b has at least one binary solution in the set of all optimal solutions. (See [16] for proof)*

Intuitive reasoning is to consider each constraint $\sum_{j=1}^n a_{ij} w_j \leq b_i$ for $i = 1, \dots, m$ as a hyperplane being placed in the n dimensional space. Since all $a_{ij} \in \{0, 1\}$, these hyperplanes intersect with the axes and with each other at points of type $w = (w_j, j = 1, \dots, n)$ only where $w_j \in \{0, 1\}$. Hence, all the extreme points (points of intersection of n planes) have binary values. For the same reason, we can effectively find solutions using a simplex approach, which checks the extreme points only.

3.4.2 Solving $g_{x,z}(\mathbf{u})$

Observe that $g_{x,z}(\mathbf{u})$ is a Mixed Integer Program which needs to be solved for each iteration of the algorithm. The number of steps required for convergence could be high and hence a fast approach needs to be used to ensure solutions for practical problems. Since at each step of the iteration, we need suboptimal solutions, we use an iterative heuristic $g_{x,z}(\mathbf{u}, X)$ (as **Algorithm 1**) based on the approach presented in [19] to solve $g_{x,z}(\mathbf{u})$. The heuristic requires the current values of \mathbf{u} . It also needs X value at which the previous iteration was completed. The heuristic is fairly simple and requires solving multiple continuous relaxations of the program to arrive at a sub-optimal solution of the problem.

When solving the $g_{x,z}(\mathbf{u})$ problem, we define sets X_f and X_v containing the variables \mathbf{x} which have been fixed (their values are already decided) and the ones that are still variables, respectively. Initially, X_f is empty and X_v contains all the \mathbf{x} variables.

Since we take the linear relaxation of (\mathbf{P}_1) to arrive at the value of F^* , we cannot be

Algorithm 1 Successive Approximation Approach: $g_{x,z}(\mathbf{u}, X)$

```
set  $X_v = \{x_{km}^p, p \in \mathcal{Q}_k, m \in \mathcal{P}_k, k \in \mathcal{K}\}$ , set  $X_f = \{\phi\}$ 
 $X \leftarrow 0.5 X$ , done  $\leftarrow 0$ , change  $\leftarrow 1$ 
while (done = 0 AND change = 1) do
   $\mathbf{x} \leftarrow$  solve Relaxed- $g_{x,z}(\mathbf{u}, X_v, X)$ 
  if ( $\mathbf{x} = \text{NULL}$ ) then
     $X \leftarrow \alpha \times X$ 
  else
    done  $\leftarrow 1$ , change  $\leftarrow 0$ 
    for all  $p \in \mathcal{Q}_k, m \in \mathcal{P}_k, k \in \mathcal{K}$  do
      if ( $x_{km}^p \leq \eta$ ) then
         $x_{km}^p = 0, X_f = X_f \cup x_{km}^p, X_v = X_v \setminus x_{km}^p$ , change = 1
      else if ( $x_{km}^p \geq 1.0 - \eta$ ) then
         $x_{km}^p = 1, X_f = X_f \cup \{x_{km}^p\}$ 
         $X_v = X_v \setminus \{x_{km}^p\}$ , change = 1
      else
        done = 0
      end if
    end for
  end if
end while
if (done = 0 OR change = 1) then
  done = 0
  while (done = 0) do
     $\mathbf{x} \leftarrow$  solve MIP- $g_{x,z}(\mathbf{u}, X_v, X)$ 
    if ( $\mathbf{x} = \text{NULL}$ ) then
       $X \leftarrow \alpha \times X$ 
    else
      done  $\leftarrow 1$ 
    end if
  end while
end if
```

assured of the existence of a binary solution with the same revenue. Hence, requiring small values of X (around zero), might lead to the infeasibility of Problem (\mathbf{P}'_2) . In fact, we use X as the cushioning space to ensure its feasibility at every step of the iteration. Therefore, at the beginning of each iteration, we start with the initial value of X equal to half of the value reached at the completion of the previous iteration. Using the values of X and X_v , we solve the Problem *Relaxed- $g_{x,z}(\mathbf{u}, X_v, X)$* , which is the relaxed version of Problem $g_{x,z}(\mathbf{u})$ with the given value of X , members of the set X_v as continuous variables and elements of X_f as fixed to the already decided values. *Relaxed- $g_{x,z}(\mathbf{u}, X_v, X)$* is a continuous linear programming problem which can be solved effectively by the simplex method even for a fairly large number of variables and constraints.

If the Problem $Relaxed_{g_{x,z}}(\mathbf{u}, X_v, X)$ is infeasible (it returns NULL), then we increase the value of X by multiplying it with α and solve the problem again. We go on increasing the value of X until we get a solution of $Relaxed_{g_{x,z}}(\mathbf{u}, X_v, X)$. Upon obtaining the solution, we first inspect the solution for the values of variable $x_{km}^p \in X_v$. The values obtained could be (a) $x_{km}^p \geq 1.0 - \eta$, (b) $x_{km}^p \leq \eta$, (c) $\eta < x_{km}^p < 1.0 - \eta$, where η is the error margin (we assumed 0.05).

The values of x_{km}^p 's of type (a) are set to 1, type (b) are set to 0 and sets X_v and X_f are updated as, $X_v \leftarrow X_v \setminus \{x_{km}^p\}$, $X_f \leftarrow X_f \cup \{x_{km}^p\}$. The variables of type (c) are left as variables. We then delete capacities and tunnels on each link based on the values of x_{km}^p which are in X_f . We then solve the reduced Problem $Relaxed_{g_{x,z}}(\mathbf{u}, X_v, X)$. We repeat the procedure until we find variables of types (a) and (b) i.e., the reduction in the problem size is achieved.

Note that it is possible that when the value of x_{km}^p of type (a) is set to 1 and the problem is rerun, the solution might be infeasible because the capacity constraint (1c) is violated (due to the increase of the value of x_{km}^p to 1). For such a scenario, increasing the value of X will not make the problem feasible. To avoid such a pitfall, we keep track of the current state of the allocation on the link capacities and tunnels (the resources). While deciding the value of x_{km}^p of type (a), we increase the value only if none of the links are getting violated, and accordingly update the sets X_v and X_f . Otherwise, we do not update the sets and let x_{km}^p still be a variable. Such a complication does not arise when deciding for types (b) and (c).

Finally, when it is not possible to obtain any more reductions in the size of the problem, we solve the MIP problem using direct branch-and-bound methods. During experiments, it was observed that the size of such a reduced problem is fairly small and hence, does not prove to be a limitation. Therefore, we have a feasible solution with all $x \in \{0, 1\}$.

3.4.3 Solving s_D

Observe that the Problem s_D is an unconstrained optimization problem with variable \mathbf{u} . The function to be minimized is non-smooth and thus, we use the sub-gradient approach to solve the dual Problem s_D . This method iterates on the dual variable \mathbf{u} . Therefore, given \mathbf{u} , once the solutions to the subproblems $g_{x,z}(\mathbf{u})$ and $g_w(\mathbf{u})$ are obtained, a dual subgradient, $\pi = (\pi_{km}^p)$, for $g(\cdot)$ is computed using

$$\pi_{km}^p = (x_{km}^p - w_{km}), p \in \mathcal{Q}_k, m \in \mathcal{P}_k, k \in \mathcal{K} \quad (23)$$

Then the dual multiplier \mathbf{u} is updated using

$$u_{km}^q = \max\{0, u_{km}^q + \mu_u \pi_{km}^{q(u)}\} \quad (24)$$

where the step sizes, μ_u , is given by

$$\mu_u = \rho \frac{g^\# - g(u)}{\|\pi\|^2} \quad (25)$$

where, $g^\#$ is the minimal primal solution over the iterations. We initialize ρ to 2.0, and it is subsequently halved whenever u does not change for consecutive 40 iterations.

We estimate the proximity of the current point with the optimal solution based on the values of gap ($g^\# - g(\pi)$) and step size μ_u . As the values of $g^\#$ and $g(\pi)$ move closer, the gap and the step size both become smaller. Hence, we assume that the iterations have converged when

$$\epsilon_i = \min\left(\frac{g^\# - g(\pi)}{g^\#}, \mu_u\right) \leq \epsilon, \quad (26)$$

for a chosen value of ϵ .

Regarding the generation of a feasible solution, it is known that the final solution derived from the algorithm presented above might not be feasible (some of the constraints might be violated). Hence, a feasible solution needs to be derived from the given values of x and w . We take the values of w as given and adjust the values of x so as to make the solution feasible and recompute the value of objective function f . We store the values of x and w corresponding to the maximum f encountered during all the iterations. We put the maximum iteration bound as 1000 and if this bound is reached, we accept the stored solution as the optimal solution.

4 Results and Discussion

We have implemented our approach in C++ while using CPLEX callable libraries [6] to solve the phased subproblems. The results presented were obtained on a stand alone Linux machine with a 1 GHz processor. The purpose of this section is two-fold: first, we comment on the convergence behavior of the decomposition algorithm presented in Section 3.4, and second, we discuss the benefit of distortion-aware traffic engineering model, which is assessed by introducing a set of distortion metrics.

In the rest of the paper, we base our results and analysis on the considered distortion measure, which is presented in the Appendix. Each source is also characterized by the squared coefficient of variation c_{kp}^2 and correlation coefficient γ_{kp} .



Fig. 3. Small Networks: SN1 and SN2

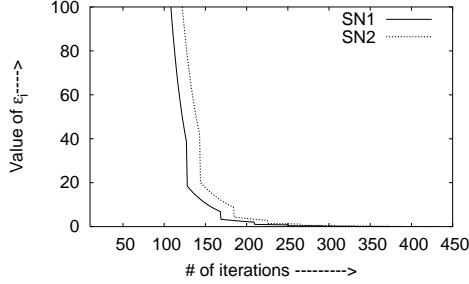


Fig. 4. ϵ_i for SNs with $T_\ell = 3$ and 100% capacity

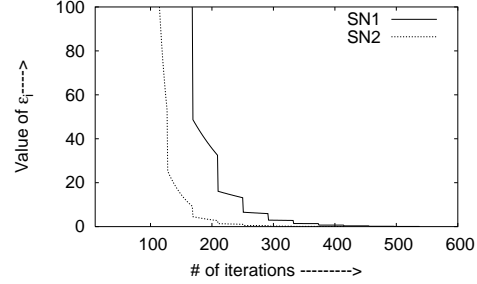


Fig. 5. ϵ_i for SNs with $T_\ell = 7$ and 100% capacity

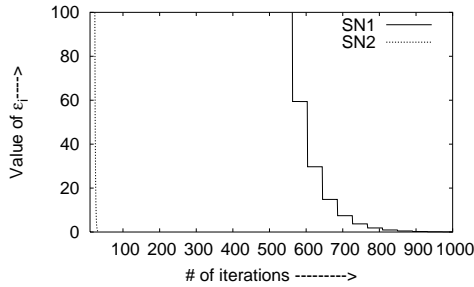


Fig. 6. ϵ_i for SNs with $T_\ell = 3$ and 300% capacity

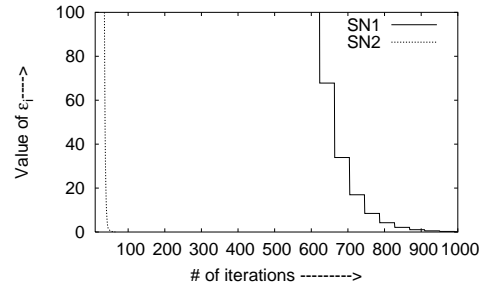


Fig. 7. ϵ_i for SNs with $T_\ell = 7$ and 300% capacity

4.1 Convergence Results

We present convergence results of the decomposition algorithm presented in Section 3.4 for two test networks (Figure 3). For both these networks, we assume that each link has a capacity of 622 Mbps, referred to as baseline capacity. We assume that every demand pair is comprised of 20 streams. These streams are varied in nature in terms of their first and second order characteristics: four streams with average rates of 10, 15, 20 and 25 Mbps with $c^2 = 6$ with no correlation; four streams with average rates of 10, 15, 20 and 25 Mbps with $c^2 = 1.5$ and $\gamma = 0.9$; four streams with average rates 10, 15, 20 and 25 with $c^2 = 2.5$ and $\gamma = 0.95$; four streams with average rates of 10, 15, 20 and 25 Mbps with $c^2 = 3.5$ and $\gamma = 0.85$; and finally, four streams with average rates of 10, 15, 20 and 25 Mbps with $c^2 = 4.5$ and $\gamma = 0.99$. Given these parameters, the distortion measure Λ is first pre-computed using the approximation, summarized in the Appendix. For our study, we generated 5 possible candidate paths for each node pair.

In order to fully evaluate the convergence, we studied the solution algorithm for both the sample networks by considering four scenarios. We describe here the convergence based on the value of ϵ_k which captures the gap between the primal and dual solution values. We observed the convergence for the tunnel and capacity constrained scenario. That is, we restricted the number of active tunnels to 3 and use a baseline capacity. The convergence behavior is presented in Figure 4 for SN1 and SN2. Next, we increased the number of permitted tunnels to 7 and still used the baseline capacity and present the results in Figure 5. This scenario was constrained in tunnels but well provisioned in capacity. We also increased the capacity to 300% of the baseline capacity and present results with active tunnels equal to 3 and 7 in Figure 6 and Figure 7, respectively.

From these results, we observed that for various constrained and well-provisioned scenarios, the convergence is achieved. Convergence of capacity constrained scenarios is particularly faster and typically require about 300-400 iterations. However, for capacity well provisioned scenarios, the convergence is somewhat slower for SN1 and relatively fast for SN2; this means that the tunneling constrained traffic engineering problem is somewhat more difficult to solve than capacity constrained traffic engineering problem. Observe that the values of ϵ_i decrease in steps which can be attributed to the approach of halving the parameter ρ which directly affects the step sizes.

4.2 Distortion-Aware Traffic Engineering

4.2.1 Distortion Metrics

To compare Formulation **(P)**, referred henceforth as the *Distortion-Aware-Model* (DAM), against a naïve allocation strategy *without* distortion, referred henceforth as the *Distortion-Ignorant-Model* (DIM), we consider the DIM to be an integer linear programming model, without the distortion measure term (quadratic term) in the objective function. For comparison between the DAM and DIM approaches, we have developed the following four metrics to capture the measured distortion and flow allocation:

- **Metric A:** Captures overall network level distortion:

$$A = \frac{\sum_{(k,m,p,q)} x_{km}^p \Lambda^{pq} x_{km}^q}{\sum_{(k,m,p)} x_{km}^p \lambda_k^p} \quad (27)$$

Table 3
Value of Metric A for Small Network for multiple scenarios

Capacity	$T_\ell = 15$				$T_\ell = 25$			
	DAM			DIM	DAM			DIM
	$\theta = 0$	$\theta = 0.1$	$\theta = 0.25$		$\theta = 0$	$\theta = 0.1$	$\theta = 0.25$	
100%	2.45 (58)	3.55 (41)	3.55 (40)	5.41 (38)	1.80 (75)	3.54 (42)	3.54 (42)	4.22 (59)
150%	2.42 (60)	3.47 (41)	4.04 (40)	6.07 (37)	1.42 (75)	3.67 (42)	3.74 (42)	6.31 (46)
200%	2.54 (59)	3.59 (41)	3.58 (40)	8.35 (25)	1.48 (75)	3.62 (42)	3.56 (42)	7.74 (35)

- **Metric B:** Captures individual demand level distortion:

$$B_k = \frac{\sum_{(m,p,q)} x_{km}^p \Lambda^{pq} x_{km}^q}{\sum_{(m,p)} x_{km}^p \lambda_k^p} \quad (28)$$

- **Metric C:** Captures individual active path level distortion:

$$C_{km} = \frac{\sum_{(p,q)} x_{km}^p \Lambda^{pq} x_{km}^q}{\sum_{(p)} x_{km}^p \lambda_k^p} \quad (29)$$

- **Metric D:** Compares the distortion variation in the DAM and the DIM approaches. Denote N as the number of active tunnels; then,

$$D = \frac{A^{DIM} N^{DIM} - A^{DAM} N^{DAM}}{N^{DAM} - N^{DIM}} \quad (30)$$

As mentioned in Section 2, distortions could be constructed based on many different requirements; one such example is presented in the Appendix. These metrics capture the general increase and decrease in the value of distortion from a network-wide point of view.

4.2.2 Results for Small Network

The first study reported is for SN1 (a six-node network). The traffic demands and streams were the same as discussed in Section 4.1. Besides the baseline capacity, we also consider cases with 150% and 200% of the baseline capacity. To restrict tunnels on a link, we consider two tunnel limit values: 15 and 25. For comparison, we consider three different values of the tunnel cost normalization parameter, θ , at 0, 0.1, and 0.25, where $\theta = 0$ means that the tunnel cost is ignored. In all studies, the normalization parameter, φ , with distortion measure is set to 1 for the DAM; note that the DIM can also be thought of as the DAM when $\varphi = 0$.

We present results for Metric A in Table 3. We also include in parenthesis the total number of active tunnels in the overall network for the specified scenario. Clearly, network level distortion is much less with the DIM than the DAM, regardless of the tunnel size. Furthermore, observe that the presence of the tunnel cost parameter ($\theta > 0$) leads to a significant reduction in the number of used tunnels while the value of Metric-A goes up. Note that the increase of θ from 0.1 to 0.25, does not have appreciable impact on the distortion. The result that provides the most strength to the DAM approach, is that even when the number of used tunnels are equal to or less than the DIM, the value of Metric A is less for the DAM approach.

Next we consider Metric B. In this case, we present graphs where the demand pair numbers are sorted based on the value of Metric B and plotted with demands having an increasing value of Metric B on the y -axis. In Figure 8, we present Metric B for the DIM and the DAM for $\theta = 0, 0.1, 0.25$ when $T_\ell = 15$ and $T_\ell = 25$ for the baseline capacity case. Observe that with higher values of T_ℓ , the distortion for each demand pair decreases for both the DAM and the DIM. Moreover, the values of Metric B for DIM is high and as expected, the DAM succeeds in decreasing the distortion for each demand. Interestingly, for $T_\ell = 25$, the differences between the DAM and the DIM for some demand pairs is less. This can be attributed to the capacity constrained nature of the scenario. Models (DAM or DIM) do not have many choices to allocate the streams, rather capacity determines the available tunnels and the allocated stream. Moreover, the value of Metric B for $\theta = 0.1$ and $\theta = 0.25$ is similar.

In Figure 9, we list Metric B for the DIM and the DAM with $\theta = 0, 0.1, 0.25$ for $T_\ell = 15$ and $T_\ell = 25$; this time for the 200% of the baseline capacity. Observe that the DAM provides significant benefits as compared to the DIM. When we compare results with the previous scenarios, we note that the gap between the behavior of the DAM and the DIM has increased considerably. The observation suggests that the presence of higher amounts of capacity and tunnels makes the DAM approach even more lucrative as compared to the DIM.

Next, consider Metric C which captures the distortion on each active tunnel (with positive allocated flow) for the DAM and the DIM. For these graphs, we filter the active tunnels (from all the tunnels) and compute Metric C for the tunnel. Then we sort the tunnels based on the value of Metric C and present the tunnels on the x -axis and the corresponding value of Metric C on the y -axis. Obviously, the plots are non-decreasing in nature.

In Figure 10, Metric C is shown for the DIM and the DAM with $\theta = 0, 0.1, 0.25$ for $T_\ell = 15$ and $T_\ell = 25$ for the baseline capacity. Observe that the distortion per unit of carried flow for tunnels is much higher for the DIM than for the DAM. This

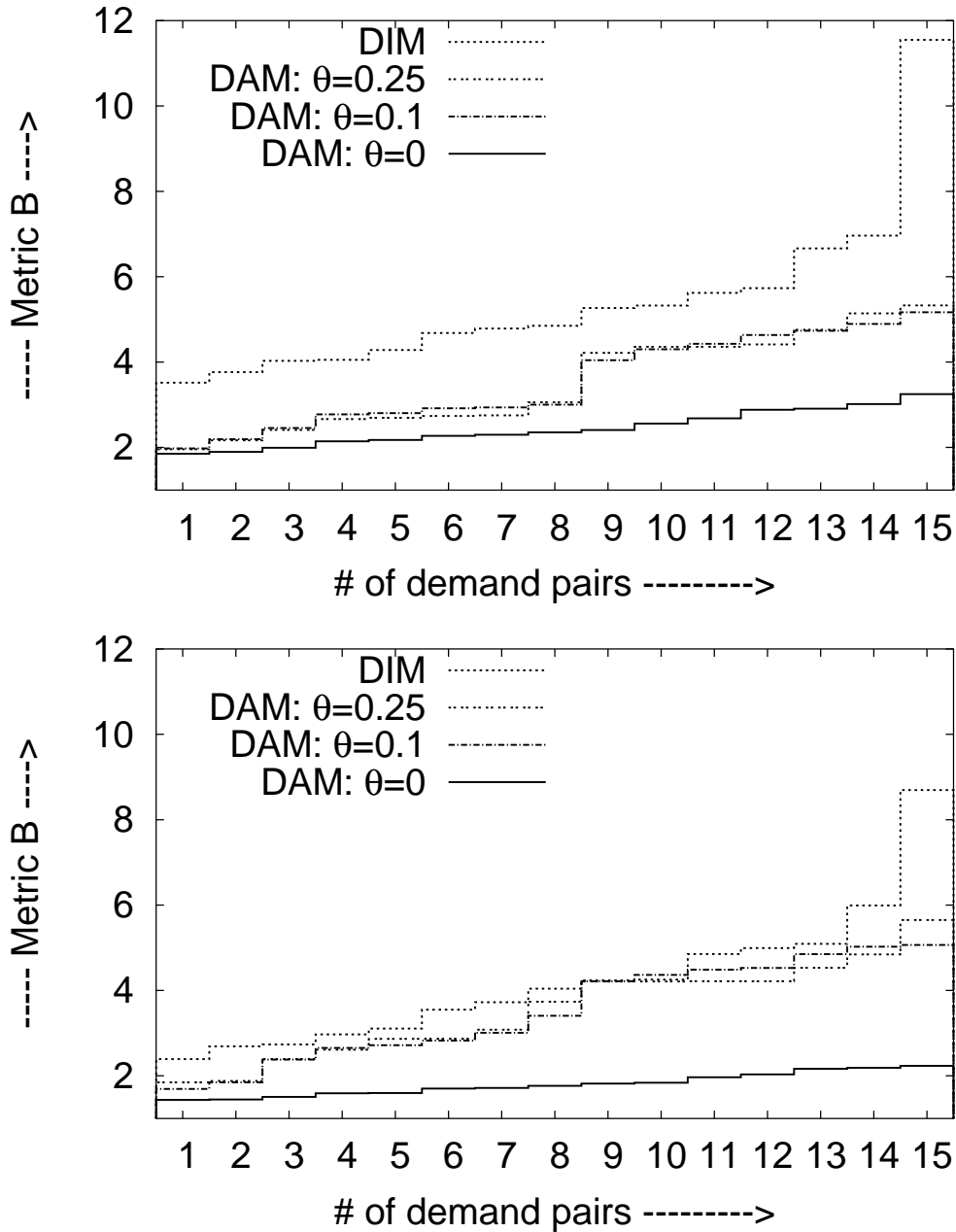


Fig. 8. Metric B with $T_\ell = 15$ and $T_\ell = 25$ and 100% capacity for Small Network

leads to a sense of disparity in terms of performance as seen by the carried traffic. On one hand, there are tunnels which have the value of Metric C as zero; on the other hand, many tunnels have Metric C as high as 12. The DAM not only ensures that the distortion is less in the network, but that the tunnels have relatively similar performance. The value of Metric C only varies from zero to four (for 15 tunnels on links) and from zero to two (for 25 tunnels on link). In general, the DIM allocates fewer tunnels than the DAM approach. However, such an affect is avoided by the presence of tunnel cost. The DAM approach with tunnel cost, not only allocates a

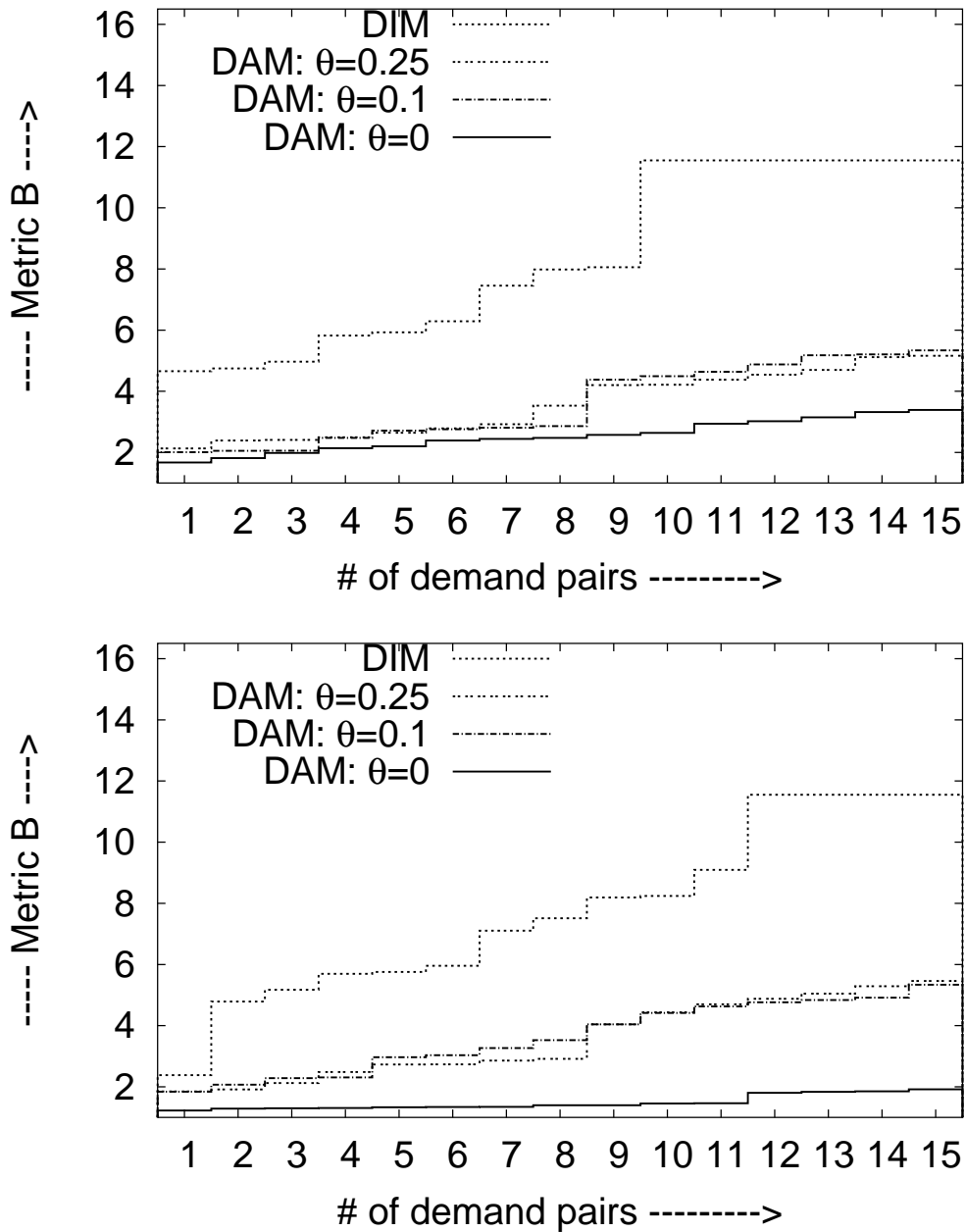


Fig. 9. Metric B with $T_\ell = 15$ and $T_\ell = 25$ and 200% capacity for Small Network

lesser number of tunnels, but it also still maintains smaller values of Metric C.

By increasing the baseline capacity to 200%, Metric C is shown for Figure 11 for DIM and DAM with $\theta = 0, 0.1, 0.25$ with $T_\ell = 15$ and $T_\ell = 25$. As before, the value of Metric B for the DIM is higher than the DAM for almost all tunnels. However, with more capacity, the DIM allocates even fewer tunnels than the previous scenario. But, the DAM approach maintains a similar number of active tunnels and continues to ensure that all tunnels get lower values of Metric C. Also observe that for some scenarios, the value of Metric C for the DAM with nonzero θ is higher

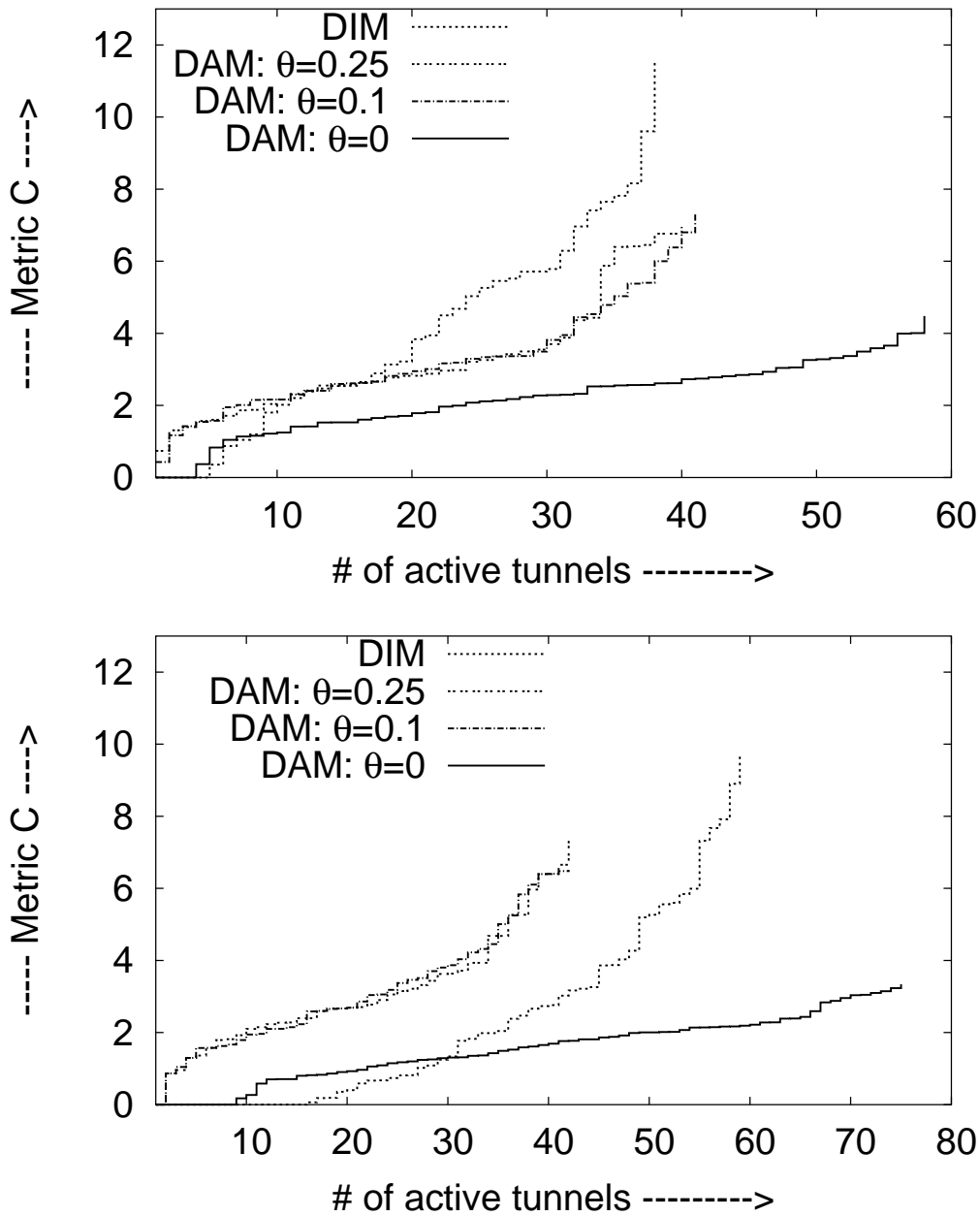


Fig. 10. Metric C with $T_\ell = 15$ and $T_\ell = 25$ and 100% capacity for Small Network

than the DIM approach. For these scenarios, we observe that the number of allocated tunnels are much lower than the DIM approach. Even for these scenarios, in general, the value of Metric A (presented before) is higher for the DIM approach (except in a few cases).

Observe that the presence of more capacity in the network leads to the DIM allocating fewer tunnels, each having a high value of Metric C. Furthermore, DAM without tunnel cost, uses many more tunnels (still conforming to the tunneling re-

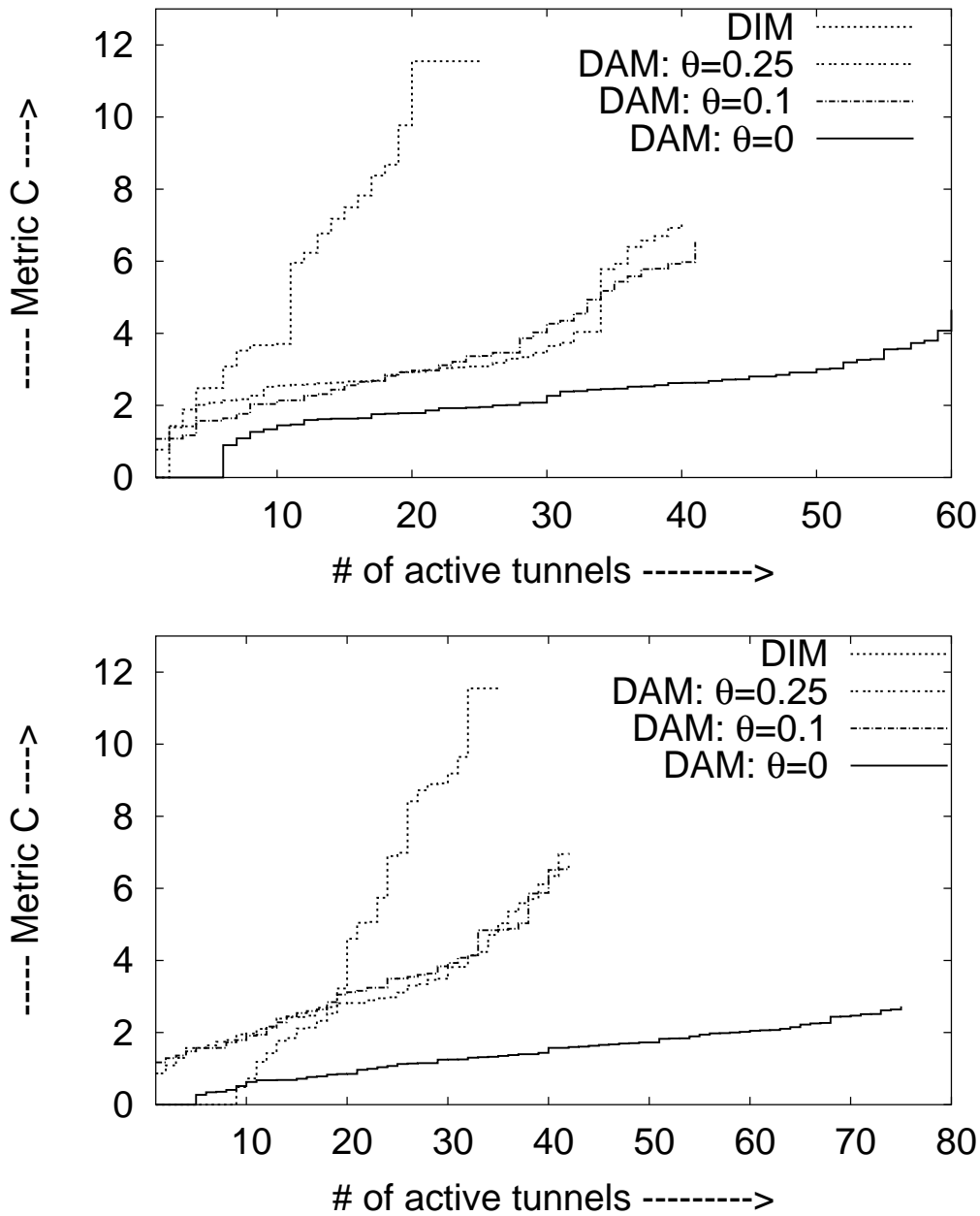


Fig. 11. Metric C with $T_\ell = 15$ and $T_\ell = 25$ and 200% capacity for Small Network

striction) and for them achieves much lower values of Metric C. When the DAM model has positive tunnel cost (nonzero θ), the number of used tunnels decreases drastically for the DAM but the value of Metric C is higher than the scenario without the tunnel cost. However, even with the decreased tunnels and increased value of Metric C, the DAM with tunnel cost performs better than the DIM model. In other words, the DAM approach fares better in terms of values of Metric C as compared to the DIM approach even when it has fewer active tunnels.

The results for Metric C indicate that the DAM achieves lower values of distortion

Table 4

Value of Metric D for the Small Network for various scenarios

Capacity	$T_\ell = 15$			$T_\ell = 25$		
	$\theta = 0$	$\theta = 0.1$	$\theta = 0.25$	$\theta = 0$	$\theta = 0.1$	$\theta = 0.25$
100%	3.17	20.00	31.70	7.12	-5.9	-5.9
150%	3.45	20.58	21.00	6.34	-34.0	-33.3
200%	1.72	3.85	4.37	4.00	17.0	17.34

by allocating more tunnels. It can be argued that such behavior is still beneficial since the routers are capable of handling such an increased number of tunnels. This brings out the original intention of our work “to trade-off the middle ground” and provide solutions which have lower distortion and also an acceptable number of tunnels. Using Metric D, we show that the DAM approach does more than that. Metric D compares the product of average distortion per unit flow and tunnels used to achieve the distortion for the DAM and the DIM. In other words, Metric D represents the differential in average distortion per unit flow between the DIM and the DAM normalized by the differential in the number of active tunnels.

In Table 4, we present values for Metric D for various scenarios for the small network. Results show that the DAM not only distributes the demands over more tunnels but also does intelligent allocation of streams to tunnels leading to a smaller value of distortion as compared to the DIM. Interestingly, the extent of benefits is varied based on the network topology, capacity and tunneling restrictions. Observe that θ amplifies the benefits of the DAM approach in terms of Metric A, vis-a-vis the used number of tunnels. The presence of tunnel cost leads to solutions which not only use fewer tunnels (than $\theta = 0$), but use these tunnels so that the distortion per unit flow is still low. When compared to the DIM, it translates into very high values of Metric D. Moreover, at times we find that the Metric D is negative, meaning that the DAM approach for a given θ uses fewer tunnels than the DIM and still has the smaller value of the product of Metric A and number of used tunnels.

4.2.3 Results for Large Network

In this section, we present results for a randomly generated network with 40 nodes and 77 links. We assumed that there are demands between every pair of nodes consisting of 5 streams each. These five streams have average arrival rates of 10, 15, 20, 25, 30, c^2 of 6, 1.5, 2.5, 3.5, 4.5 and γ of 0, 0.9, 0.95, 0.85, 0.99, respectively. Given these parameters, distortion measure Λ is first pre-computed using the approximation given in the Appendix. We assumed that each link has a capacity of 10 Gbps and can support 1000 tunnels. The value of Metric A (and active tunnels) for the DAM approach was found to be 0.034 (3681), 0.37 (2493) and 0.37 (2464) for θ equal to 0, 0.1, and 0.25, respectively. For the DIM approach, the corresponding

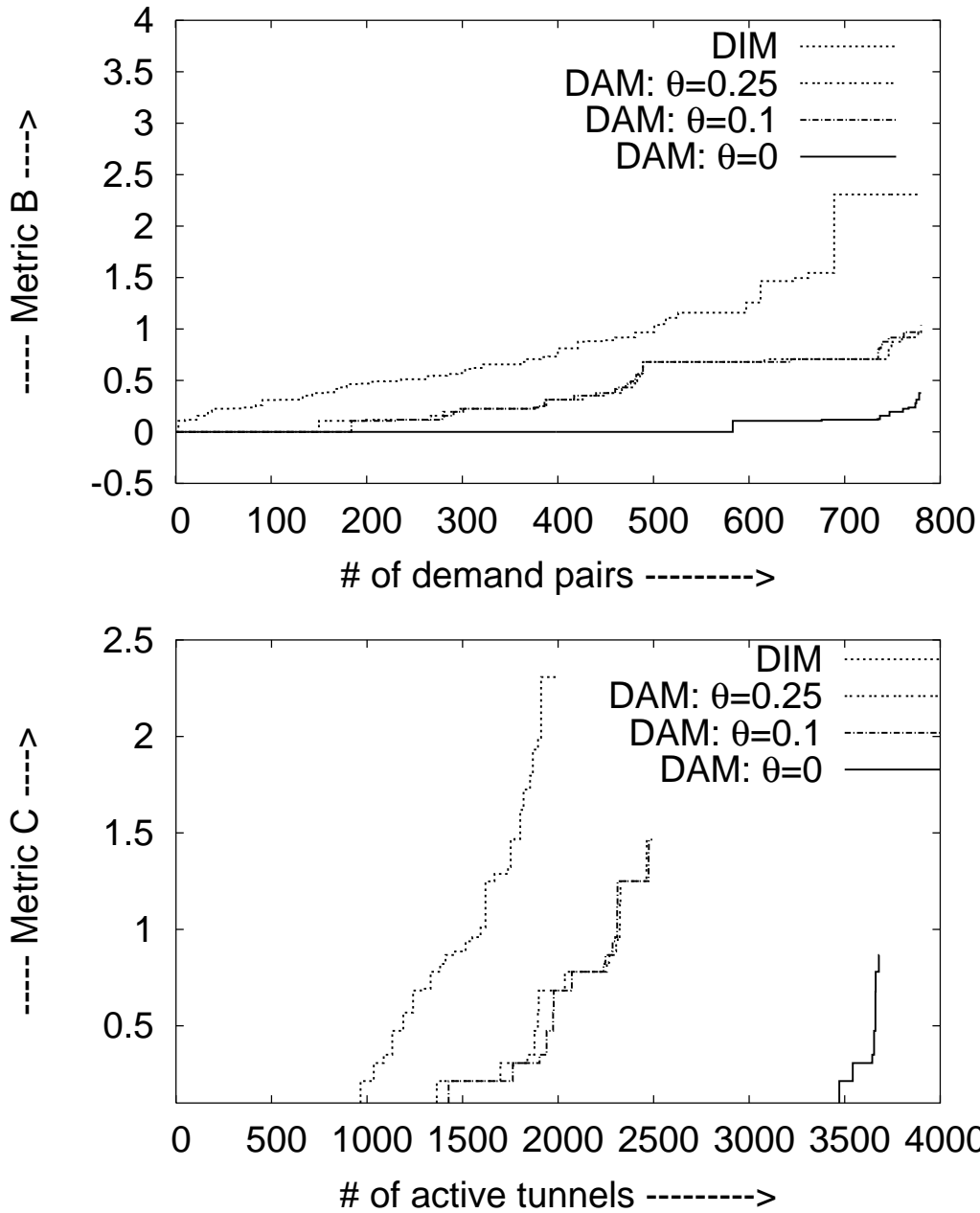


Fig. 12. Metric B and Metric C for Large Network

value is 0.93 (2002). In Figure 12, we present the values of Metric B and C. The value of Metric D for the DAM approach was found to be 1.03, 1.91, and 2.05 for θ equal to 0, 0.1, and 0.25, respectively. The results follow the same trend as was observed for the small network.

Convergence behavior for this large network was similar to the small networks discussed earlier. The computing time on a 1 GHz processor linux machine took less than a minute to complete for this large problem. The overall solutions approach is

fast because each iteration is inexpensive.

In summary, the numerical results show that the formulation is indeed powerful and that it captures the basic motive of minimizing distortion in the presence of tunneling and capacity constraints. More so, the formulation adequately responds to various possible scenarios of tunnel and capacity restrictions. Moreover, the solution also gives a novel way of determining the optimal mapping of traffic streams to tunnels. For the ease of forwarding, streams sharing the same tunnels can be grouped together as the same service class. Such a grouping ensures that no additional filtering needs to be done to incorporate the solution in a real world network. Observe that such a mapping is driven by better end-to-end performance as seen by individual streams.

5 Conclusion and Future Work

In this work, we presented an approach for effective traffic engineering of an MPLS network in the presence of heterogeneous streams, by incorporating distortion, tunneling limits, and capacity. Our approach is primarily applicable for ‘what-if’ studies for short- and medium-term traffic engineering planning, and is not geared for real-time traffic engineering. By taking distortion into account, we presented a quadratic integer programming formulation. We then showed how this model can be linearized. We developed an efficient two-phase algorithm to solve the traffic engineering formulation.

Our approach can be labeled as a “Distortion-Aware” traffic engineering approach. We developed several metrics to compare our approach against a distortion-ignorant approach. Through numerical studies, we have quantified this difference and shown that our approach is quite effective in allocating heterogeneous traffic streams to like-minded tunnels (LSPs) to reduce distortion in a network-wide setting.

It may be noted that our notion of distortion is broad-based and generic. For example, if a provider does not want two customers to be combined on the same tunnel, it can assign a high distortion value for this pair of customers; then, our model will identify if they are appropriate to be combined based on a set of constraints and for a given objective function. From our numerical results, it is evident that without incorporating the distortion measure, some unwanted traffic streams (or customers) may be combined.

Our approach takes an end-to-end view in addressing distortion with traffic engineering. Several future directions may be pursued. A future direction would be to allow the ability to incorporate link level distortion. Another direction would be to consider how distortion is impacted when there is a network failure and traffic

streams are to be re-routed. It is to be noted that the notion of distortion considered here is broad based. A third future work would be if typically parameters of interests such as loss, delay, and jitter could be mapped to a measure of distortion when traffic streams are combined in an MPLS tunnel. Finally, how to incorporate any additional restrictions due to MPLS tunnel implementation because of queues at routers would be worthwhile to address.

Acknowledgement

We thank the anonymous reviewers for their constructive comments that have helped improve the presentation of the paper considerably. This work was supported in part by NSF grant # CNS-0106640.

Appendix

A method (and the motivation behind this method) to determine pair-wise distortion measure Λ^{pq} for distortion between two streams p and q is detailed in [15]. We briefly summarize how it is used as input to the traffic engineering problem. Given the average arrival rate (λ), the squared coefficient of variation (c^2) and the coefficient of correlation (γ), the mismatch between streams p and q is approximated as

$$\Lambda^{pq} = \Lambda_r^{pq} + \Lambda_c^{pq} + \Lambda_\gamma^{pq}, \quad (31)$$

where Λ_r^{pq} is the contribution of rates to the mismatch between the streams, Λ_c^{pq} is the contribution due to c^2 , and Λ_γ^{pq} due to the coefficient of correlation. These, in turn, are estimated by

$$\Lambda_r^{pq} = |\lambda_p - \lambda_q|, \quad (32a)$$

$$\Lambda_c^{pq} = |(c_p^2 - 1) + (c_q^2 - 1)|, \quad (32b)$$

and

$$\Lambda_\gamma^{pq} = |(\lambda_p \gamma_q) + (\lambda_q \gamma_p)|. \quad (32c)$$

A special case is when both the streams are Poisson, in which we set $\Lambda^{pq} = 0$, since no distortion occurs.

In the remainder of this appendix, we explain the reason for choosing such a distortion measure. Consider that the above mentioned streams p, q , share an exponential server S with an infinite queue size, and under a heavy load. Upon being served by the server S , the stream p is separated and passed through another exponential server S_p , and we compute the value of $P[Q_p > 0]$. Similarly, we compute the value of $P[Q_q > 0]$ for stream q . In order to achieve this, we built upon the linear algebraic queueing theory (LAQT) based model presented in [18]. We now show that higher (smaller) values of Λ^{pq} would translate into higher (smaller) values of

metric $P[Q > 0]$ at the servers. Moreover, higher values of $P[Q > 0]$ is undesirable for any stream passing through a network since it negatively impacts the observed waiting time for an arriving packet.

In Figures 13- 15, we plot values of $P[Q > 0]$ for increasing Λ^{pq} (on x-axis) where stream p has different distribution characteristics. We consider three types arrival processes distributions of stream p : (1) Poisson ($c^2 = 1$), (2) high c^2 , but uncorrelated, and (3) high c^2 and correlated. Similarly, stream q varies from Poisson to uncorrelated (but with high c^2) to correlated and high c^2 . In our case, a correlated stream is approximated using [10, 11] and is referred to as *Marie-Mitchell* with parameters λ , c^2 , and γ .

For the case when streams p is Poisson (λ_p) and q has high c^2 and is also correlated (i.e., $q \sim \text{Marie-Mitchell}(\lambda_q, c_q^2, \gamma_q)$), we present results in Figure 13; note that $\tilde{\Lambda}^{pq}$ is determined based on (32) and we used $\lambda_p = \lambda_q = 5$ for this example. It may be noted that we observed distortions. Since, $P[Q > 0]$ is minimal (for both streams) for lower values of $\tilde{\Lambda}^{pq}$, metric (32) is a meaningful metric to be minimized in order to ensure that the distortions in the streams are minimal. Similar patterns were observed for other values of λ_p and λ_q .

For the case when stream p is not Poisson and has a high c^2 but is uncorrelated, and q is same as the previous case, i.e., $\sim \text{Marie-Mitchell}(\lambda_q, c_q^2, \gamma_q)$, we present results in Figure 14;. Here, we used $p \sim \text{Marie-Mitchell}(5, 4, 0)$ and for stream q , we used $\lambda_q = 5$ for illustration. Observe that, for this case, the suggested measure $\tilde{\Lambda}^{pq}$ can also adequately capture the trend of distortion in both the streams. Similar results were obtained for streams p with different λ and c^2 , and stream q with other

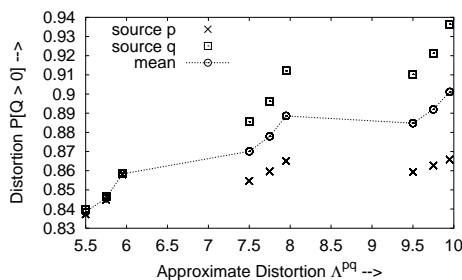


Fig. 13. Stream p is Poisson

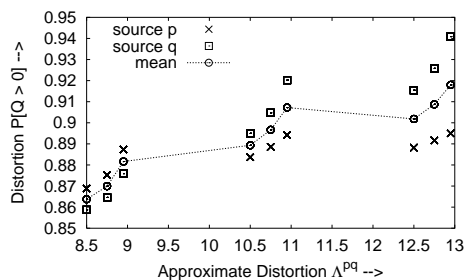


Fig. 14. Stream p is uncorrelated

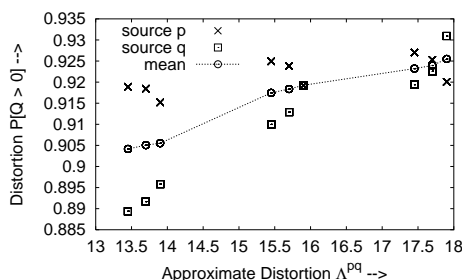


Fig. 15. Stream p is correlated

values of λ .

In the final case, both streams have high c^2 and are also correlated ($\gamma > 0$). In Figure 15, we present results for the $P[QueueLength > 0]$ for both streams p and q as the approximate metric increases. For this illustration, we have used stream p with $\lambda = 5$, $c^2 = 4$ and $\gamma = 0.99$, and stream q with $\lambda = 5$.

Similar behaviors were observed for different values of λ , c^2 and γ of the stream p and λ for stream q . To summarize, the chosen form of Λ^{pq} adequately captures the extent of distortion in the streams p and q . Thus, the presented distortion metric (Λ) is readily available for various types of streams under consideration to be multiplexed, and it reflects the negative impact of sharing of streams upon their packet arrival characteristics.

It may be noted that the traffic engineering formulation presented in this paper is independent of the distortion measure described above. Other methods can be developed and used while retaining the traffic engineering formulation presented in this paper.

References

- [1] A. L. Corte, A. Lombardo, and G. Schembra, "Modeling superposition of on-off correlated traffic sources in multimedia applications," in *Proceedings of IEEE INFOCOM*, 1995, pp. 993–1000.
- [2] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS adaptive traffic engineering," in *Proc. IEEE INFOCOM 2001*, April 2001, pp. 1300–1309.
- [3] A. M. Geoffrion, "Lagrangian relaxation for integer programming," *Mathematical Programming Study*, vol. 2, pp. 82–114, 1974.
- [4] Q. He, M. Ammar, G. Riley, H. Raj, and R. Fujimoto, "Mapping peer behavior to packet-level details: A framework for packet-level simulation of peer-to-peer systems," in *Proceedings of MASCOTS*. ACM, October 2003, pp. 71–78.
- [5] M. Held, P. Wolfe, and H. Crowder, "Validation of sub-gradient optimization," *Mathematical Programming*, pp. 62–88, 1974.
- [6] *Manual of Cplex Callable Libraries*, ILOG Corporation, USA.
- [7] K. Kar, M. Kodialam, and T. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 2566–2579, 2000.
- [8] Y. Li, P. M. Pardalos, K. G. Ramakrishnan, and M. G. C. Resende, "Lower bounds for the quadratic assignment problem," *Annals of Operations Research*, vol. 50, pp. 387–411, 1994.

- [9] L. Lipsky, P. Fiorini, W. Hsin, and A. van de Liefvoort, "Auto-correlation of lag- k for customers departing from semi-Markov processes," Technical University of Munich, Tech. Rep. TUM-19506, 1995.
- [10] R. Marie, "Méthodes itératives de résolution de modèles mathématiques de systèmes informatiques," *R.A.I.R.O. Informatique/Computing Science*, vol. 12, pp. 107–122, 1978.
- [11] K. Mitchell, "Constructing correlated sequence of matrix exponentials with invariant first-order properties," *Operations Research Letters*, vol. 28, pp. 27–34, 2001.
- [12] D. Mitra and K. G. Ramakrishnan, "Techniques for traffic engineering of multiservice, multipriority networks," *Bell Labs Technical Journal*, vol. 6, pp. 139–151, 2001.
- [13] P. M. Pardalos, Y. Ye, and C.-G. Han, "Algorithms for the solution of quadratic knapsack problems," *Linear Algebra and its Applications*, vol. 152, pp. 69–91, July 1991.
- [14] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers, 2004.
- [15] S. Srivastava, "Models and algorithms for effective traffic engineering of tunnel-based backbone networks," Ph.D. dissertation, Computer Science & Electrical Engineering Department, University of Missouri–Kansas City, 2004.
- [16] S. Srivastava, B. Krithikaivasan, D. Medhi, and M. Pióro, "Traffic engineering in the presence of tunneling and diversity constraints: Formulation and Lagrangean decomposition approach," in *Proceedings of 18th International Teletraffic Congress*. Berlin: Elsevier Science, 2003, pp. 461–470.
- [17] S. Srivastava and D. Medhi, "Traffic engineering of tunnel-based networks with class specific diversity requirements," *Journal of Combinatorial Optimization*, vol. 12, pp. 97–125, September 2006.
- [18] S. Srivastava, K. Mitchell, and A. van de Liefvoort, "Correlations induced in a packet stream by background traffic in a multiplexed environment," in *Proceedings of 18th International Teletraffic Congress (ITC)*. Berlin: Elsevier Science, 2003, pp. 511–520.
- [19] S. Srivastava, S. R. Thirumalasetty, and D. Medhi, "Network traffic engineering with varied levels of protection in the next generation Internet," in *Performance Evaluations and Planning Methods for the Next Generation Internet (edited by A. Girard, B. Sansò, and F. Vazquez-Abad)*. Springer, 2005, pp. 99–124.
- [20] C.-F. Su and G. de Veciana, "Statistical multiplexing and mix-dependent alternate routing in multiservice networks," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 99–108, 2000.
- [21] X. Xiao, A. Hannan, B. Bailey, and L. Ni, "Traffic engineering with MPLS in the Internet," *IEEE Network*, vol. 14, no. 2, pp. 28–33, March/April 2000.
- [22] S. Yasukawa, A. Farrel, and O. Komolafe, "An analysis of scaling issues in MPLS-TE core networks," *IETF RFC 5439*, February 2009. <http://www.rfc-editor.org/rfc/rfc5439.txt>