Advances in Survivability-supported Framework for Traffic Engineering in Multi-service Backbone Networks

[1]VASILIOS PASIAS, [2]DIMITRIOS A. KARRAS, [3]RALLIS C. PAPADEMETRIOU
[1]ONEX COMPANY S.A., Athens, GREECE
[2]National and Kapodistrian University of Athens (NKUA), GREECE
and the Canadian Institute of Technology, Tirana, ALBANIA
[3]University of Portsmouth, School of Energy and Electronic Engineering, UK

Abstract: In this paper, optimisation models and heuristic algorithms that address the off-line survivability-supported Traffic Engineering (TE) problem in multi-service backbone networks are presented. In such networks traffic demands with different Quality of Service (QoS) and survivability requirements (e.g. existence of a node disjoint backup path for each primary path) inhere. The optimisation models for engineering the QoS traffic with different survivability prerequisites and the Best-Effort (BE) traffic are based on novel admission control/routing Integer Linear Programming (ILP) and Linear Programming (LP) optimisation sub-problems, which are solved sequentially. LP relaxations of the ILP sub-problems are also provided. The optimisation models and heuristic algorithms are tested on two different networks and their performance is compared.

Keywords: Traffic Engineering (TE), Survivability, Quality of Service (QoS), Best-Effort (BE), optimisation, heuristic algorithms, routing, protection, restoration.

Received: March 14, 2022. Revised: November 5, 2022. Accepted: December 7, 2022. Published: December 31, 2022.

1. Introduction

RECENT years have witnessed a tremendous increase in the carried traffic volume through the Internet and other networks. New high-revenue network-based services are introduced and some of them, unlike with others, require the provision of quality guarantees, such as *Internet telephony*. However, the trend is towards the development of hybrid *multi-service* networks capable of handling together voice, video and data traffic with different *Quality of Service* (QoS) and survivability requirements (*blocking ratio, packet loss ratio, maximum end-to-end delay, minimum available bandwidth* etc.).

In such networks, *Traffic Engineering* (TE) mechanisms can be applied for the efficient utilisation of the existing network resources (e.g. link capacity). In fact, TE is a major issue in the emerging multi-service networks supporting the *Multi-Protocol Label Switching* (MPLS) protocol as well as in *Asynchronous Transfer Mode* (ATM), *Differentiated Services* – DiffServ / non-DiffServ supporting *Internet Protocol* (IP) and *Frame Relay* (FR) networks [1]. This paper concentrates in MPLS-based and DiffServ-supported IP backbone networks interconnecting *Points-of-Presence* (POP) [2]. In these networks traffic flows (demands) with the same source-destination address having the same QoS requirements are aggregated forming what is called a *traffic trunk* in IP networks or a *Label Switched Path* (LSP) in MPLS networks. To the rest of the paper aggregated flows will be referred to as traffic trunks or simply trunks.

One of the main problems in TE is how to map traffic trunks in the network, while satisfying the QoS requirements of the trunks. In order to solve this problem a number of *on-line* [3-6] and *off-line* [1, 7-10] TE approaches have been already developed. In the on-line approaches, traffic trunks are mapped onto the network one at a time as soon as a new demand for a trunk emerges. On-line TE is state-dependent and applies on a short time-scale. The main objective of the on-line approaches is to allow the network to respond rapidly to any changes in traffic load or network topology. However, routing randomly emerging demands one at a time may cause an unfair utilisation of network resources. On the other hand off-line TE aims at establishing well-defined routes for traffic trunks in such a way that the utilisation of network resources is globally optimised. Specifically, instead of focusing on instantaneous network states and individual connections, the latter mechanism considers statistical behavior of traffic trunks. Combining this information with a centralised view of network topology and link capacities, off-line TE selects the topology of routes for traffic trunks and provisions resources on the selected routes for carrying trunk traffic in an optimal manner.

Unfortunately, considering the widespread use of communication networks, a failure or a disastrous attack affecting one or more network facilities (devices/nodes or links) could cause catastrophic social-economic effects. In order to either minimise the effect of such damage to the delivery of services or even to avoid service disturbance the networks must be made survivable. For this purpose two main categories of techniques can be used: *network design/capacity planning* and also *protection/restoration* techniques. Practically, in most cases it is affordable to use techniques of the latter category only. Protection/restoration methods can be well incorporated in the *traffic management* mechanisms of networks. Two very common protection schemes, particularly in *virtual circuit* based networks [11], [12], are *1+1* and *1:1 protection*. In the case that 1+1 protection is applied, fixed network resource reservations are made for both a primary and a node/link disjoint backup path for each demand. When 1:1 protection is applied fixed resource reservations are made for the primary path only; the predetermined resources for the node/link disjoint backup path will be used for routing traffic as soon as the primary path fails.

In this paper, three novel optimisation methods and two heuristic algorithms for off-line TE in survivable multi-service backbone networks, in the case of single or multiple node and/or link failure(s), are presented. All methods and algorithms support 1+1/1:1 protection. Survivability against multiple node or link failures in multi-service backbone networks, has not adequately addressed so far to the best of our knowledge. The remaining of the paper is organised as follows: in Section II the optimisation TE problems are presented, while in Section III the heuristic algorithms are described; the simulation results concerning the comparison between the off-line TE methods are illustrated in Section IV; finally Section V concludes the paper.

2. Optimisation Models Framework

Consider a weighted undirected graph G = (N, L) where N denotes the set of nodes and L denotes the set of edges (links). The graph is 2-connected, i.e. even if one link or node fails a path between each source-destination pair can still be found. Each link $l \in L$ has capacity C_l, propagation delay D_l, processing delay Dp_l associated with the nodes located at the two sides of the link, length L_l and cost K_l.

Four service (priority) classes based on QoS and survivability requirements are defined in this paper: the *High* class t' with the greater priority and the most strict QoS requirements (e.g. maximum end-to-end delay) requiring protection against node and or link failure(s) (e.g. voice traffic), the *Medium* class also being delay sensitive and requiring protection (e.g. *Virtual Private Network* (VPN) traffic), the *Low* class which is delay insensitive (e.g. *World-Wide-Web* (WWW) traffic) and the *Best-Effort* (BE) class with the lowest priority, involving BE traffic. For brevity to the rest of this section High, Medium and Low class trunks will be referred to as *QoS trunks* and the BE class trunks will be referred to as simply *BE trunks*.

The following notations are also used:

- Tqos: QoS class set.
- Σ: QoS traffic trunk set. Note that three traffic trunks for each source-destination pair are defined, each corresponding to a specific QoS class.
- Σb: BE traffic trunk set. One BE traffic trunk for each source-destination pair is normally defined.
- S: set of the normal operating state and the failure states of the network. Failure states here correspond to node and/or link failure(s).
- $T_{\sigma t}$: bandwidth demand (traffic) of the trunk $\sigma \in \Sigma$, which belongs to service class $t \in$ Tqos.
- $T_{\sigma b}$: trunk $\sigma \in \Sigma b$ traffic, belonging to the BE class.
- Rs(t,σ): set of all candidate routes for the trunk $\sigma \in \Sigma$ belonging to the QoS class $t \in$ Tqos, containing only operating links (links interconnecting operating nodes), at the network state $s \in S$. Note that for the BE trunks no candidate routes are defined.
- Kp_i: additive cost of the candidate route $i \in Rs(t,\sigma)$ for the trunk $\sigma \in \Sigma$ of class $t \in$ Tqos. The proposed path cost definitions can be found in Appendix A.
- $Z_{\sigma t}$: earnings/protection/restoration weight for $T_{\sigma t}$. It indicates the priority that trunk $\sigma \in \Sigma$ of class $t \in$ Tqos has as regards admissibility/routing into the network and/or protection/restoration.
- $Z_{\sigma b}$: earnings/protection/restoration weight for $T_{\sigma b}$.
- Wp_t: admission priority of trunk traffic belonging to service class $t \in$ Tqos.
- B: maximum end-to-end delay for paths.
- H: maximum hop count for paths.
- cb_s: link utilisation bound for $s \in S$ (0 – 1).
- $p_{\sigma s}$: backup path *reservation* parameter ($0 \le p_{\sigma s} \le 1$) for trunk $\sigma \in \Sigma$ traffic that belongs to High class t′, at state $s \in S$. Specifically, $p_{\sigma s}$ is the maximum fraction of demand $T_{\sigma t'}$ that must be accommodated through the backup path $q \in Rs(t',\sigma)$ at $s \in S$.
- RevS: Revenue scaling factor (0 – 1).
- U: optimisation weighting factor.
- In(n): set of links $l \in L$ directed into node $n \in N$.
- Out(n): set of links $l \in L$ directed out of node $n \in N$.
- $d_{\sigma s}$: *diversification* parameter for $s \in S$ and for $\sigma \in \Sigma b$, $0 \le d_{\sigma s} \le 1$; $d_{\sigma s}$ is the maximum fraction of BE traffic $T_{\sigma b}$ allowed to flow through any link $l \in L$ at $s \in S$.

The decision variables for the QoS-related optimisation sub-problems are the following:

- X_{rs}: primary path routing integer variable, that is $X_{rs} = 1$ if trunk $\sigma \in \Sigma$ traffic belonging to class $t \in$ Tqos uses the service route $r \in Rs(t,\sigma)$ as primary at state $s \in S$, otherwise $X_{rs} = 0$;
- Y_{qs}: backup path routing integer variable, that is $Y_{qs} = 1$ if trunk $\sigma \in \Sigma$ traffic belonging to the High class t′ uses the service route $q \in Rs(t',\sigma)$ as backup at $s \in S$, otherwise $Y_{qs} = 0$.

The decision variables for the BE-related optimisation sub-problems are the following:

- F_σ: total carried BE traffic for trunk $\sigma \in \Sigma b$.
- $X_{l\sigma}$: component of F_σ ($\sigma \in \Sigma b$) carried on link $l \in L$.

2.1. Model for the Survivable Admission control/Routing sub-problem for the QoS traffic (SARQOS)

The objective is to maximise the network revenue obtained from the QoS traffic admission. The first term denotes the revenue obtained by the traffic demands when routed on the admitted primary paths, while the second term denotes the revenue obtained by routing the traffic demands on the admitted backup paths when the corresponding primary paths are unavailable due to failure(s).

$$Max:\left(\sum_{t\in Tqos}\sum_{\sigma\in\Sigma}\sum_{r\in Rs(t,\sigma)} Wp_t \bullet Z_{\sigma t} \bullet X_{rs}\right)+\left(\sum_{\sigma\in\Sigma}\sum_{q\in Rs(t',\sigma)} Z_{\sigma t'} \bullet Y_{qs}\right)$$

The following constraint indicates that trunk $\sigma \in \Sigma$ traffic belonging to service class $t \in$ Tqos at $s \in S$, must use at most one primary route $r \in Rs(t,\sigma)$.

$$\sum_{r\in Rs(t,\sigma)} X_{rs} \le 1, \forall \sigma \in \Sigma, \forall t \in Tqos, s \in S \qquad (1)$$

The next constraint indicates that trunk σ ∈ Σ traffic belonging to the High class t', must use at most one backup route q ∈ Rs(t',σ).

$$\sum_{q\in Rs(t',\sigma)} Y_{qs} \le 1, \forall\sigma\in\Sigma, s\in S \qquad (2)$$

The next constraint is the link capacity constraint.

$$\left(\sum_{t\in Tqos}\sum_{\sigma\in\Sigma}\sum_{r\in Rs(t,\sigma):l\in r} T_{\sigma t}\bullet X_{rs}\right) + \left(\sum_{\sigma\in\Sigma}\sum_{q\in Rs(t',\sigma):l\in r}\rho_{\sigma s}\bullet T_{\sigma t'}\bullet Y_{qs}\right) \le cb_s\bullet C_l, \forall l\in L, s\in S \qquad (3)$$

Constraint (4) forces the primary and the backup paths for the trunk σ ∈ Σ traffic that belongs to class t′, at state s ∈ S, to be link disjoint.

$$\sum_{r\in Rs(t',\sigma):l\in r} X_{rs} + \sum_{q\in Rs(t',\sigma):l\in r} Y_{qs} \le 1, \forall\sigma\in\Sigma, \forall l\in L, s\in S \qquad (4)$$

Constraint (5) forces the primary and the backup paths for the trunk σ ∈ Σ traffic belonging to class t′, at state s ∈ S, to be node disjoint.

$$\sum_{r\in Rs(t',\sigma):n\in r} X_{rs} + \sum_{q\in Rs(t',\sigma):n\in q} Y_{qs} \le 1, \forall\sigma\in\Sigma, \qquad (5)$$
$$\forall n\in N-\{\sigma s,\sigma d\}, s\in S$$

, where σs is the source and σd the destination of the trunk σ ∈ Σ respectively.

2.2 Model for the Survivable Admission control/Routing sub-problem f or the BE traffic (SARBE)

The objective is to maximise the revenue obtained from the BE class traffic admission.

$$Max:\left(\sum_{\sigma\in\Sigma b} Z_{\sigma b}\bullet F_{\sigma}\right)$$

The following constraint indicates that the admitted bandwidth F_σ for σ ∈ Σb must not exceed the suggested traffic demand $T_{\sigma b}$.

$$0\le F\sigma\le T_{\sigma b}, \forall\sigma\in\Sigma b \qquad (6)$$

The next constraint indicates that node n ∈ N may be source, destination or relay for every σ ∈ Σb.

$$\sum_{l\in In(n)} X_{l\sigma} - \sum_{l\in Out(n)} X_{l\sigma} = F\sigma, \quad \text{n = source of trunk } \sigma\in\Sigma b$$

$$\sum_{l\in In(n)} X_{l\sigma} - \sum_{l\in Out(n)} X_{l\sigma} = -F\sigma, \quad \text{n = destination of trunk } \sigma\in\Sigma b$$

$$\sum_{l\in In(n)} X_{l\sigma} - \sum_{l\in Out(n)} X_{l\sigma} = 0, otherwise$$

∀ nodes n ∈ N, σ ∈ Σb.

(7)

The link capacity constraint is shown below.

$$\sum_{\sigma\in\Sigma b} X_{l\sigma} \le cb_s\bullet V, l\in L, s\in S \qquad (8)$$

The next constraint is the flow non-negativity constraint. In order to increase survivability traffic diversification is involved.

$$0\le X_{l\sigma}\le d_{\sigma s}\bullet T_{\sigma b}, \forall\sigma\in\Sigma b, \forall l\in L, s\in S \qquad (9)$$

2.3 Model for the Combined Survivable Admission control/Routing problem for the QoS and BE traffic (Method 1 – M1)

Method 1 consists of two sub-problems (phases), which are solved sequentially. The first is the SARQOS optimisation sub-problem (phase 1), whose formulation is presented in Section I-A. This is solved first. Then the second optimisation sub-problem (phase 2) is solved which is a hybrid problem and it is called *Survivable Admission control/Routing problem for the QoS and BE traffic* (SARQOSBE). Its formulation is a combination of the formulations of SARQOS and SARBE (see below).

The objective is to maximise the revenue obtained from the BE class traffic admission (1st term) and to minimise the network resources (link bandwidth) conservation by the QoS traffic (2nd and 3rd term), that is to optimise the selection of the primary and backup paths in terms of path cost based routing metrics.

$$Max:\left(\sum_{\sigma\in\Sigma b} Z_{\sigma b}\bullet F_{\sigma}\right) + \left(\sum_{t\in Tqos}\sum_{\sigma\in\Sigma}\sum_{r\in Rs(t,\sigma)} Wp_t\bullet\frac{Z_{\sigma t}}{K_r}\bullet X_{rs}\right) + \left(\sum_{\sigma\in\Sigma}\sum_{q\in Rs(t',\sigma)}\frac{Z_{\sigma t'}}{K_q}\bullet Y_{qs}\right)$$

The following constraint relates SARQOS to SARQOSBE. The revenue to be obtained from the solution of the SARQOSBE sub-problem must be at least equal to the product of the revenue Rev(SARQOS) obtained from the solution of the SARQOS sub-problem (being the optimally maximum revenue) and the revenue scaling factor RevS, which is used to adjust the value of the revenue. The above product is actually a lower bound to the value of the total revenue obtained from the solution of SARQOSBE.

$$\left(\sum_{t\in Tqos}\sum_{\sigma\in\Sigma}\sum_{r\in Rs(t,\sigma)} Wp_t\bullet Z_{\sigma t}\bullet X_{rs}\right) + \left(\sum_{\sigma\in\Sigma}\sum_{q\in Rs(t',\sigma)} Z_{\sigma t'}\bullet Y_{qs}\right) \ge Rev(SARQOS)\bullet RevS \qquad (10)$$

The link capacity constraint is shown next

$$\left(\sum_{t\in Tqos}\sum_{\sigma\in\Sigma}\sum_{r\in Rs(t,\sigma):l\in r} T_{\sigma t}\bullet X_{rs}\right) + \left(\sum_{\sigma\in\Sigma}\sum_{q\in Rs(t',\sigma):l\in r}\rho_{\sigma} s\bullet T_{\sigma t'}\bullet Y_{qs}\right) + \left(\sum_{\sigma\in\Sigma} X_{l\sigma}\right) \le cb_s\bullet C_l, \forall l\in L, s\in S \qquad (11)$$

Constraints (1), (2), (4) – (7) and (9) are also used in the SARQOSBE sub-problem formulation. Note that the only coupling between SARQOS and SARQOSBE occurs in

(10). The variables X_{rs} and Y_{qs}, common to both sub-problems, may though have different values.

2.4 Model for the two-layered decomposition of the Combined Survivable Admission control/Routing problem for the QoS and BE traffic (Method 2 ï M2)

Method 2 consists of two sub-problems. The first sub-problem is a modified version of SARQOS where the objective function is replaced by the following:

$$Max:\left(\sum_{t\in Tqos}\sum_{\sigma\in\Sigma}\sum_{r\in Rs(t,\sigma)} Wp_t \bullet U \bullet Z_{\sigma t} \bullet X_{rs}\right)$$
$$+\left(\sum_{\sigma\in\Sigma}\sum_{q\in Rs(t',\sigma)} U \bullet Z_{\sigma t'} \bullet Y_{qs}\right)$$
$$+\left(\sum_{t\in Tqos}\sum_{\sigma\in\Sigma}\sum_{r\in Rs(t,\sigma)} Wp_t \bullet \frac{Z_{\sigma t}}{U \bullet K_r} \bullet X_{rs}\right)$$
$$+\left(\sum_{\sigma\in\Sigma}\sum_{q\in Rs(t',\sigma)} \frac{Z_{\sigma t'}}{U \bullet K_q} \bullet Y_{qs}\right)$$

, which either maximises the revenue obtained from the QoS traffic admission (2 first terms) or minimises the cost of routing the primary and backup paths (next 2 terms), subject to the value of the optimisation weighting factor U. For large values of U, emphasis is placed to the revenue maximisation; while for small values of U emphasis is given to routing costs minimisation. This sub-problem is solved first (phase 1).

Then the second sub-problem is solved (phase 2) which is the SARBE model, in which the constraint (8) is modified as follows:

$$\sum_{\sigma\in\Sigma b} X_{l\sigma} \le ((cb_s \bullet C_l) - V_l), l \in L, s \in S \qquad (12)$$

, where V_l is the bandwidth reserved on link $l \in$ L in order to satisfy the solution of the sub-problem of phase 1. In the case that 1:1 protection is applied, the reserved link bandwidth for the backup paths is not subtracted from the product $cb_s \bullet C_l$.

2.5 Model for the three-layered decomposition of the Combined Survivable Admission control/Routing problem for the QoS and BE traffic (Method 3 ï M3)

Method 2 consists of three sub-problems, which are solved sequentially in the order given next. The first sub-problem is called *Admission control/Routing problem for the QoS traffic* (ARQOS). It is a modified version of SARQOS in which survivability is not considered. Specifically, no admission control/routing decisions about backup paths are made. See the ARQOS formulation next.

The objective is to either maximise the revenue obtained from the QoS traffic admission (first term) or minimise the routing costs for the primary paths (second term), depending on the value of the factor U.

$$Max:\left(\sum_{t\in Tqos}\sum_{\sigma\in\Sigma}\sum_{r\in Rs(t,\sigma)} Wp_t \bullet U \bullet Z_{\sigma t} \bullet X_{rs}\right)$$
$$+\left(\sum_{t\in Tqos}\sum_{\sigma\in\Sigma}\sum_{r\in Rs(t,\sigma)} Wp_t \bullet \frac{Z_{\sigma t}}{U \bullet K_r} \bullet X_{rs}\right)$$

The link capacity constraint is shown below.

$$\sum_{t\in Tqos}\sum_{\sigma\in\Sigma}\sum_{r\in Rs(t,\sigma):l\in r} T_{\sigma t} \bullet X_{rs} \le cb_s \bullet C_l, \forall l \in L, s \in S \qquad (13)$$

The constraints (1) and (4) are also used to the ARQOS sub-problem formulation.

The second sub-problem (phase 2) is a modified version of the SARQOS sub-problem called *path pre-selection* SARQOS or *pp-SARQOS*. In this problem the obtained solution from phase 1 is considered. Let R`s(t,σ) be the set of all admitted paths at phase 1. In this set there is at most one path for each σ ∈ Σ and class t ∈ Tqos at s ∈ S. It is R`s(t,σ) ⊆ Rs(t,σ). Also let R``s(t',σ) be the set of all candidate backup paths, for trunk σ ∈ Σ and the High class t' at s ∈ S. Note that the paths in R`s(t',σ) are not included in R``s(t',σ). It is R``s(t',σ) ⊆ Rs(t,σ). Using the previous notation the pp-SARQOS sub-problem is formulated next.

The objective is either to maximise the revenue obtained by routing survivability-supported QoS traffic on the admitted backup paths when the corresponding primary paths fail due to node/link failure(s) (1st term), or to minimise the cost for routing the backup paths (2nd term), subject to the value of U.

$$Max:\left(\sum_{\sigma\in\Sigma}\sum_{q\in R``s(t',\sigma)} U \bullet Z_{\sigma t'} \bullet Y_{qs}\right)$$
$$+\left(\sum_{\sigma\in\Sigma}\sum_{q\in R``s(t',\sigma)} \frac{Z_{\sigma t'}}{U \bullet K_q} \bullet Y_{qs}\right)$$
$$X_{rs} = 1, \forall \mathrm{r} \in \mathrm{R`}(\mathrm{t}, \sigma), \forall \sigma \in \Sigma, \forall \mathrm{t} \in \mathrm{T}qos, s \in S \qquad (14)$$

The constraint (14) indicates that traffic belonging to service class t ∈ Tqos for source-destination pair σ ∈ Σ and for network state s ∈ S, uses the path r ∈ R`s(t,σ).

The next constraint indicates that traffic belonging to High class t' for pair σ ∈ Σ and for state s ∈ S, must use at most one backup route q ∈ R``s(t',σ).

$$\sum_{q\in R``s(t',\sigma)} Y_{qs} \le 1, \forall \sigma \in \Sigma, s \in S \qquad (15)$$

The link capacity constraint is shown below

$$\left(\sum_{t\in Tqos}\sum_{\sigma\in\Sigma}\sum_{r\in R`s(t,\sigma):l\in r} T_{\sigma t} \bullet X_{rs}\right) + \left(\sum_{\sigma\in\Sigma}\sum_{q\in R``s(t',\sigma):l\in r} \rho_{\sigma s} \bullet T_{\sigma t'} \bullet Y_{qs}\right)$$
$$\le cb_s \bullet C_l, \forall l \in L, s \in S \qquad (16)$$

The following constraint forces the primary and the backup paths for the High class t' and the trunk σ ∈ Σ at network state s ∈ S to be link disjoint.

$$\sum_{r\in R`s(t',\sigma):l\in r}^{`} X_{rs} + \sum_{q\in R``s(t',\sigma):l\in r}^{`} Y_{qs} \leq 1, \forall \sigma \in \Sigma, \forall l \in L, s \in S \qquad (17)$$

The next constraint forces the primary and the backup paths for class t' and trunk $\sigma \in \Sigma$ at state $s \in S$ to be node disjoint.

$$\sum_{r\in R`s(t',\sigma):n\in r}^{`} X_{rs} + \sum_{q\in R``s(t',\sigma):n\in r}^{`} Y_{qs} \leq 1, \qquad (18)$$

$$\forall \sigma \in \Sigma, \forall n \in N - \{\sigma s, \sigma d\}, s \in S$$

Note that for the solution of the pp-SARQOS sub-problem, QoS traffic demands with survivability guarantees (backup path definition) are required. Otherwise the objective function of the sub-problem is undefined. The third sub-problem (phase 3) is the SARBE in which constraint (8) takes the form of the constraint (12).

In the case that an ILP problem (Exact problem) is infeasible then an *LP-relaxed* version of it is solved. For the LP-relaxation of the ILP TE problems it is assumed that X_{rs} and Y_{qs} can take any value in the interval [0,1], i.e. the variables may take non-integer values. However, only integer (0 or 1) results are finally considered.

3. Heuristic Algorithms

In the context of the present work, two heuristic algorithms have been developed in order to address the survivability-supported TE problem in multi-service backbone networks. These are the *Survivable Traffic Engineering algorithm 1* (STE_1) and the *Survivable Traffic Engineering algorithm 2* (STE_2). These algorithms are based on the *Traffic Engineering Algorithm 1* (TEA_1) and the *Traffic Engineering Algorithm 2* (TEA_2).

Both TEA_1 and TEA_2 are based on the *Dijkstra shortest path algorithm* [13]. TEA_1 is used for admission control/routing of QoS traffic trunks. It involves the following steps (consider the aforementioned terminology):

1. Sort the QoS traffic trunks in decreasing order according to their priority, i.e. the higher priority trunks are placed first. The sorting takes place in two phases. Specifically, at the first phase, considering the priority classes previously defined, the trunks belonging to the High service class are put first followed by the trunks belonging to the Medium class etc. Then at the second phase the trunks of each class are re-arranged according to the their earnings rates, that is the trunks with the higher earnings rates are placed first followed by the trunks with the lower earnings rates. When two or more trunks have the same earnings rate then these trunks are placed in decreasing order of bandwidth demand, that is the trunks with the higher bandwidth demand are placed first followed by the trunks with the lower bandwidth demand.
2. Consider the higher priority trunk with demand $T_{\sigma t.}$
3. Create a sub-graph G′ where all the links with residual bandwidth less than the traffic demand $T_{\sigma t}$ are removed. This ensures that all the remaining links have bandwidth greater than or equal to $T_{\sigma t}$.
4. To sub-graph G′ use Dijkstra's algorithm to determine the minimum link weight path between the source and the destination of the trunk, considering special link weights based on the metrics presented in [14].
5. If a path exists and the number of intermediate hops along the path is less than the max hop bound H then establish the path and deduct the resources (e.g. link capacity) used by the path.
6. For each of the next trunks, placed in descending order of priority, repeat steps 3 to 5.

TEA_2 is used for admission control/routing of BE trunks. It involves the same basic procedure and number of steps as TEA_1. However, at step 1 the BE traffic trunks are sorted in decreasing order according to their earnings rates, i.e. the trunks with the higher earnings rates are placed first. When two or more BE trunks have the same earnings rate then these trunks are placed in decreasing order of bandwidth demand. Furthermore, at step 4 Dijkstra's algorithm determines the minimum hop path between the source and the destination of the considered trunk. In step 5 no hop count test is made.

Note that in both TEA_1 and TEA_2 the complexity of step 3 is O(L) and the complexity of step 4 is $O(N^2)$. Since connected networks are considered it is $L \leq N^2$. So, the overall complexity of the steps 3 and 4 is $O(N^2)$.

The STE_1 algorithm (*Method 4 – M4*) establishes a primary path and where required a backup path for each QoS and BE traffic trunk. STE_1 is outlined next.

1. Sort the QoS traffic trunks in decreasing order according to their priority (see TEA_1 above).
2. Consider the higher priority trunk with demand $T_{\sigma t.}$
3. Execute steps 3 to 5 of the TEA_1 algorithm in order to find a primary route r for the trunk. Store the route r.
4. Exclude from graph G the nodes (links) belonging to the primary route r except source and destination.
5. To the remaining graph execute steps 3 to 5 of TEA_1 to find a node (link) disjoint backup route for r.
6. Repeat steps 3 – 5 for each of the High and Medium class traffic trunks, considering that the trunks are in descending order of priority.
7. For each of the lower QoS class trunks repeat step 3, having in mind that the trunks are placed in descending order of priority.
8. Execute TEA_2 for each BE trunk placed in descending order of priority.

The STE_2 algorithm (*Method 5 – M5*) also establishes a primary path and where required a backup path for each QoS and BE traffic trunk. STE_2 is illustrated below.

1. Sort the QoS traffic trunks in decreasing order according to their priority (see TEA_1 above).
2. Consider the higher priority trunk with demand $T_{\sigma t.}$
3. Execute steps 3 to 5 of the TEA_1 algorithm in order to find a primary route r for the trunk. Store the route r.
4. For each of the next trunks repeat step 3, having in mind that the trunks are placed in descending order of priority.
5. Consider the higher priority High class trunk requiring a backup route.
6. Exclude from graph G the nodes (links) belonging to the corresponding primary route r except source and destination.

7. To the remaining graph execute steps 3 to 5 of TEA_1 to find a node (link) disjoint backup route for r.
8. Repeat steps 6 and 7 for each of the High and Medium class traffic trunks, considering that the trunks are in descending order of priority.
9. Execute TEA_2 for each BE trunk placed in descending order of priority.

4. Simulations and Results

For the tests the *NetLab* software package [15] was used. It was developed using the *Tcl/Tk* scripting language [16]. NetLab is used for network topological design and simulation, incorporating a *Graphical User Interface* (GUI). NetLab is capable of solving ILP and LP optimisation problems using the *lp_solve 4.0* software [17], which is a freely available LP and ILP solver.

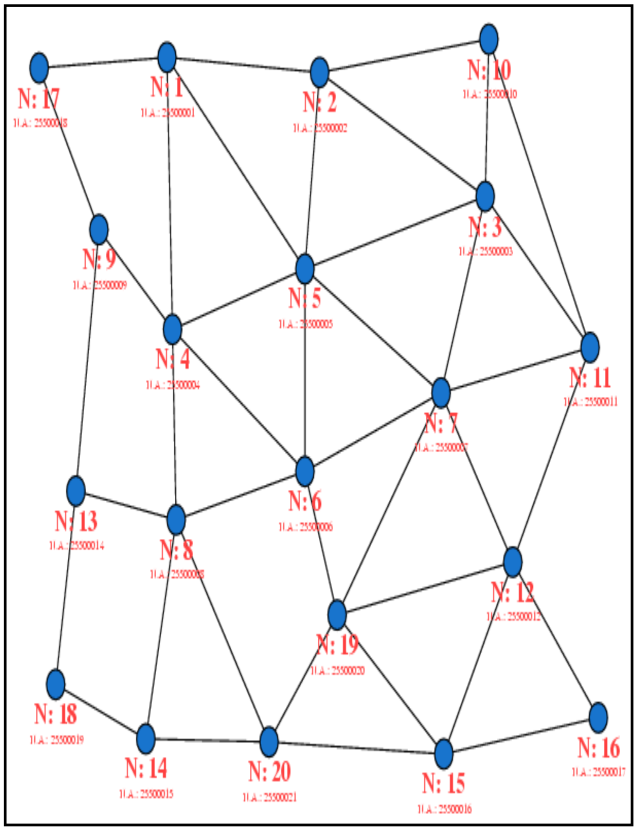

Figure 1. Network A.

The experiments were run on a PC equipped with a Pentium III 638 MHz CPU and 448 MB RAM. Note that the solution of the optimisation problems and the heuristic algorithms is both CPU and memory intensive.

The objective of the tests was to find the blocking ratios of trunks in two typical networks. Specifically, for each method the ratio of the number of admitted primary routes to the number of expected primary routes for the QoS trunks, the ratio of the number of admitted backup routes to the number of expected backup routes for the survivability-supported QoS trunks and also the ratio of the number of admitted routes to the number of expected routes for the BE trunks must be obtained.

For the tests the network A (Figure 1) with 20 nodes and 40 links and the network B (Figure 2) with 15 nodes and 40 links were used. Each link of A was assigned a capacity equal to 8 Mbps and each link of B was assigned a capacity equal to 4 Mbps. In network A two sets of 10 tests each were executed. The first test involved 1520 and the second one 1140 demands for primary QoS trunks, backup paths for the survivability-supported QoS trunks and BE trunks. Here, these demands are characterised as *commodities*. In network B two sets of 10 tests each were performed with 1260 and 840 commodities respectively. Note that the primary and the corresponding backup path must be node disjoint. 1-and-1 protection was used for the tests. Also the following assumptions were made: $Wp_t = 2$ for all QoS classes, $Z_{\sigma t} = Z_{\sigma b} = 1$ for all demands, $U = 1$, $p_{\sigma s} = 1$ for all High class demands, $d_{\sigma s} = 1$ for all BE demands, $RevS = 1$, $H = 6$ for all candidate paths for the High class, $H = 10$ for all candidate paths for the other QoS classes and $cb_s = 0.95$.

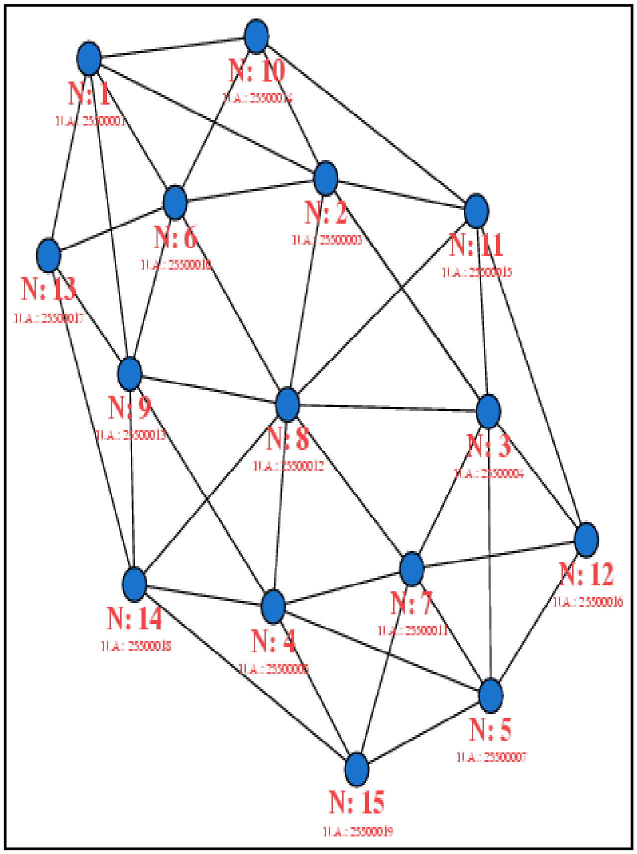

Figure 2. Network B.

In order to find a number of candidate *admissible* routes [1] for each QoS trunk, a special algorithm called *Path Finding Algorithm* (PFA), was initially utilised. For more on PFA see Appendix B. PFA found totally 561 and 1264 candidate admissible paths for networks A and B respectively.

The experimental results obtained from the solution of the Exact and the LP-relaxed optimisation models as well as from the heuristic algorithms are presented in Figures 3 - 6. However, due to the large number of results only average blocking ratios from each set of tests are illustrated in this paper. Note that in the 1st set of tests (Figure 3) no results were acquired from Method 1 because the SARQOSBE sub-problem was infeasible, probably due to the huge number of variables involved (34280). Besides, in the 3rd set of tests (Figure 5) no results were obtained from Method 3 because the pp-SARQOS sub-problem was infeasible.

From the tests, it was obtained that Method 2 out-performed all other optimisation methods as regards both QoS trunk admissibility and backup path establishment. On the other hand Method 1 out-performed all other optimisation models regarding BE trunk admissibility. However, Method 5 performed the best among all methods regarding QoS and BE trunk admissibility. On the contrary, Method 4 performed the best among all methods regarding backup path establishment. The Exact models performed better than the LP-relaxed models but the solution time for the LP-relaxed models was less. Also, the solution time for Methods 4 and 5 was much less than for the optimisation methods in all cases.

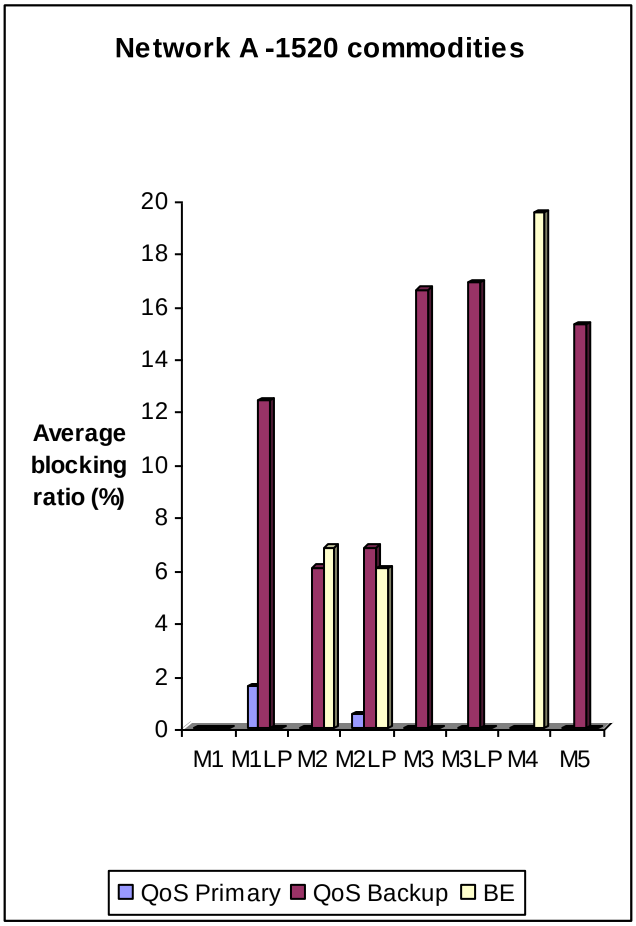

Figure 3. Results from the first set of tests for Network A.

5. Conclusions

Considerable literature exists on off-line TE techniques for use in multi-service networks. However, survivability-supported off-line TE methods have not yet studied thoroughly. This paper focuses on the formulation of novel optimisation models and heuristic algorithms for the solution of the survivability-supported off-line TE problem in multi-service backbone networks where demands with different QoS and survivability requirements inhere.

Extensive experimental work supports the theoretical work providing evidence on the performance of the proposed optimisation models and heuristic algorithms. Basically, the selection of the best optimisation model depends mainly on the required efficiency that the QoS and BE traffic admission control/routing and the backup path planning should have, always according to the employed network management strategy (e.g. find as many backup paths as possible) or TE policy (e.g. give priority to QoS traffic over BE traffic). However, STE_2 (Method 5) showed the best performance among all approaches in terms of both QoS and BE traffic admissibility and total solution time but not in terms of backup path planning where the STE_1 algorithm (Method 4) outperformed all other approaches. In every case the solution time for the greedy heuristics (STE_1 and STE_2) was much less than for the optimisation methods.

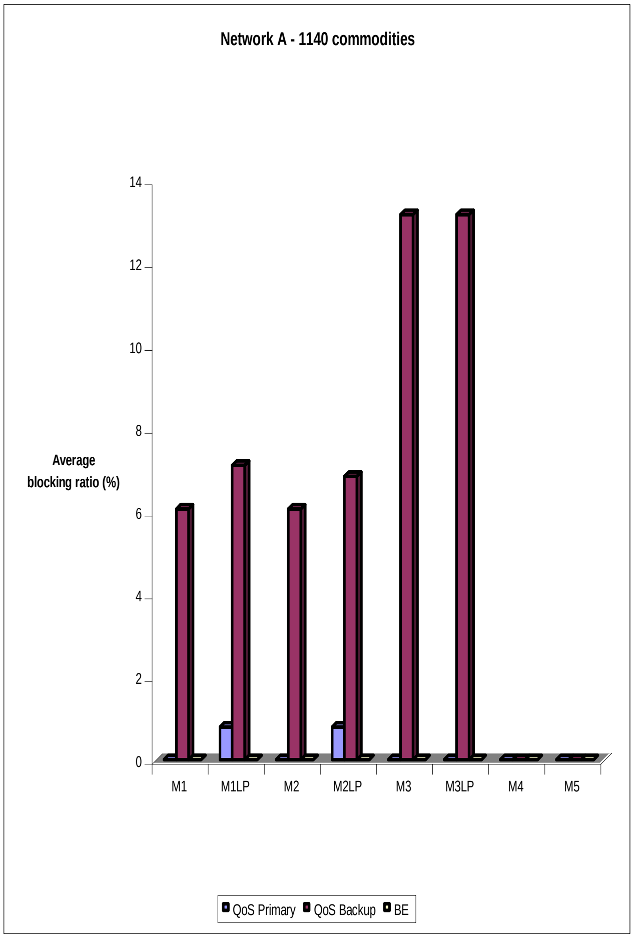

Figure 4. Results from the second set of tests for Network A.

Nevertheless, some characteristics of the off-line TE methods, especially regarding load balancing, are currently under study. In addition, more protection schemes, such as *M+N/M:N* and *shared path protection*, are investigated. A framework for off-line TE and on-line routing in survivable multi-service backbone networks is also under development.

References

[1] Girão-Silva, R., Craveirinha, J., Clímaco, J. et al. Multiobjective routing in multiservice MPLS networks with traffic splitting — A network flow approach. J. Syst. Sci. Syst. Eng. 24, 389–432 (2015). https://doi.org/10.1007/s11518-015-5262-4
[2] XiPeng Xiao, T. Telkamp, V. Fineberg, Cheng Chen, L.M. Ni, “A Practical Approach for Providing QoS in the Internet Backbone”, IEEE Communications Magazine, Vol. 40, No. 12, December 2002, pp. 56 - 62.
[3] Awduche, D. O., & Jabbari, B. (2002). Internet traffic engineering using multi-protocol label switching (MPLS). Computer Networks, 40(1), 111-129.
[4] K. Kar, M. Kodialam, and T.V. Lakshman, “Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications”, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 12, December 2000, pp. 2566 - 2579.
[5] Subhash Suri, Marcel Waldvogel, Daniel Bauer and Priyank Ramesh Warkhede, “Profile-Based Routing and Traffic Engineering”, IEEE Computer Communications, Vol. 25, 2002.
[6] Singh, Sandeep Kumar, Tamal Das, and Admela Jukan. "A survey on internet multipath routing and provisioning." IEEE Communications Surveys & Tutorials 17, no. 4 (2015): 2157-2175.

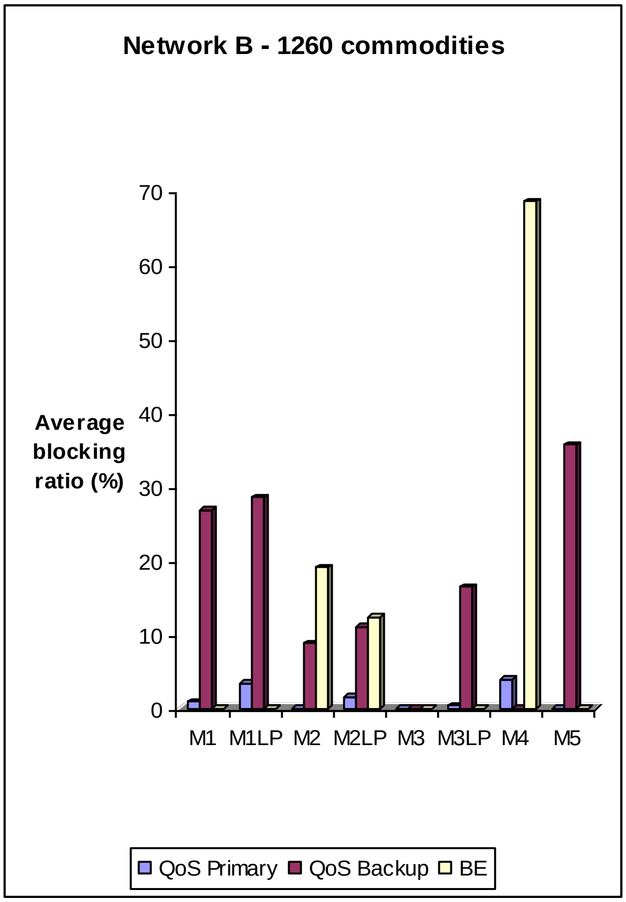

Figure 5. Results from the first set of tests for Network B.

[7] E. Karasan and E. Yetginer, “Robust Path Design Algorithms for Traffic Engineering with Restoration in MPLS Networks”, IEICE Transactions on Communications, Vol. E85–B, No. 9, September 2002.
[8] P. Trimintzios, T. Bauge, G. Pavlou, L. Georgiadis, P. Flegkas, R. Egan, “Quality of Service Provisioning for Supporting Premium Services in IP Networks”, Proceedings of IEEE Globecom 2002, Taipei, Taiwan, November 2002.
[9] M.K. Girish, B. Zhou, and J.Q. Hu, “Formulation of the Traffic Engineering Problems in MPLS Based IP Networks”, in Fifth IEEE Symp. Comp. Commun. (ISCC), 2000, pp. 214 - 219.
[10] S. R. Thirumalasetty and D. Medhi, “MPLS Traffic Engineering for Survivable Book-Ahead Guaranteed Services”, Technical Report, CST/UMKC, January 2001.
[11] Segovia, Juan, Pere Vila, Eusebi Calle, and Jose L. Marzo. "Improving the resilience of transport networks to large-scale failures." Journal of Networks 7, no. 1 (2012): 63.
[12] Kazutaka Murakami and Hyong S. Kim, “Comparative Study on Restoration Schemes of Survivable ATM Networks”, IEEE INFOCOM '97, Kobe Japan, April 1997, pp. 3C.1.1-8.
[13] Dijkstra, E. W., “A Note on Two Problems in Connexion with Graphs”, Numerische Mathematik, 1959, pp. 269 - 271.
[14] B. Fortz and M. Thorrup, “Optimizing OSPF/IS-IS Weights in a Changing World”, IEEE Journal on Selected Areas in Communications, No. 20, May 2002.
[15] R.C. Papademetriou and V. Pasias, “NetLab - An Interactive Simulation Environment for Communication Networks”, Proceedings of the 1st IEEE International Conference on Information Technology: Research and Education (ITRE 2003), NJ, USA, August 11 - 13, 2003, pp. 123 - 127.
[16] Welch, Brent B., Ken Jones, and Jeffrey Hobbs. Practical programming in Tcl and Tk. Prentice Hall Professional, 2003.
[17] Linderoth, Jeffrey T., and Andrea Lodi. "MILP software." Wiley encyclopedia of operations research and management science 5 (2010): 3239-3248.

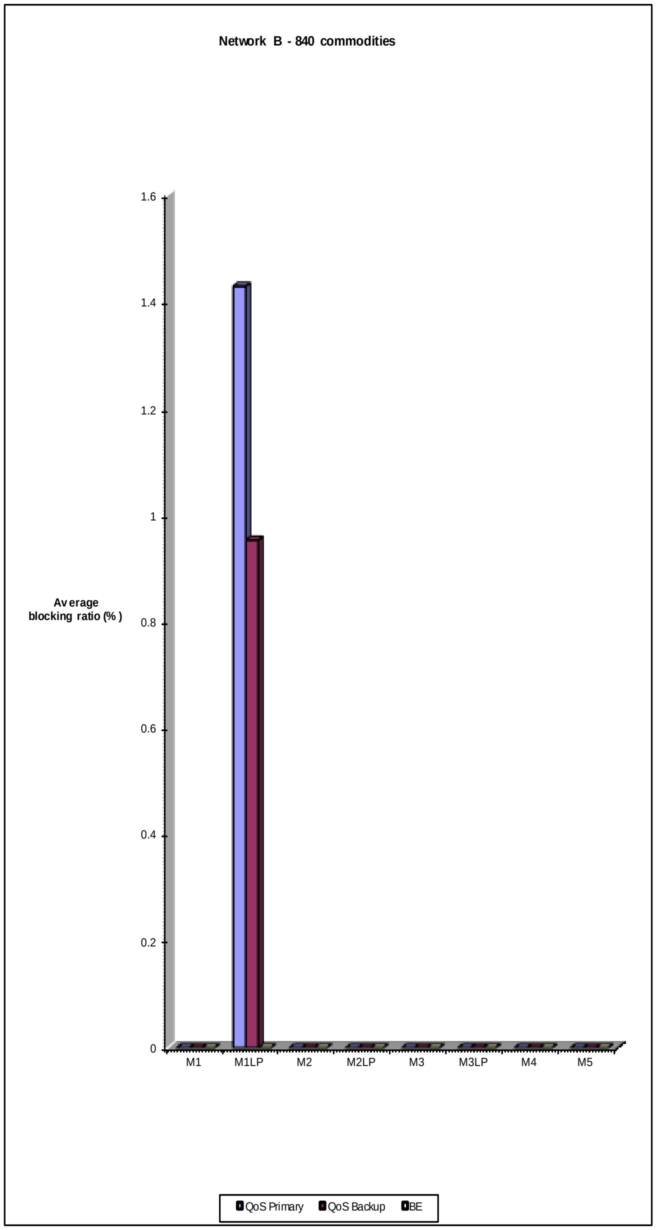

Figure 6. Results from the second set of tests for Network B.

[18] Thong Fan and E.S. Lee, "Multiple QoS constrained routing with inaccurate information", Electronic Letters, Vol. 35, No.21, 14th October 1999, pp. 1807 - 1808.

[19] Younis, Ossama, and Sonia Fahmy. "Constraint-based routing in the internet: Basic principles and recent research." IEEE Communications Surveys & Tutorials 5, no. 1 (2003): 2-13.

[20] Bertsekas, Dimitri, and Robert Gallager. Data networks. Athena Scientific, 2021.

[21] Azar, Erik, and Mario Eguiluz Alebicto. Swift data structure and algorithms. Packt Publishing Ltd, 2016.

APPENDIX A

Considering the aforementioned terminology, the total path cost K_r is given by the relation:

$$K_r = \sum_{l \in r} K_l, \; l \in L \qquad (19)$$

The value of the link cost K_l depends on the QoS class of the traffic demand that is to be routed.

For constant-bit-rate services, such as voice, the main consideration regarding routing is to minimize total service delay and delay variation (*jitter*), while preserving specific end-to-end delay and jitter bounds. Note that the total service delay consists of propagation, queuing and node processing delays. Therefore, service paths with bounded minimum end-to-end total delay and jitter must be selected. For this kind of services the following link cost metrics can be used:

- Considering the M/M/1 queue type model for links, it is: $K_l = (C_l/(V_l)^2) + D_l + Dp_l$ (20)
  , where V_l is the residual bandwidth on link $l \in L$.
  If $V_l = cb_s \bullet C_l$, it is: $K_l = (1/(cb^2 \bullet C_l)) + D_l + Dp_l$ (21)
- The link length could be used for the link cost definition. Long paths introduce large propagation delays.
- The next hop-based link metric can be used:
  $K_l = 1$ (22)
  The less hops a path has the less delay it introduces (see Appendix B).

For the variable-bit-rate services all the above link costs can be used, provided that the average traffic rate does not exceed specific bounds. However, especially for variable-bit-rate services with QoS guarantees, like real-time video, either the link cost in (20)/(21) or the link length would be adequate. Besides, considering *Weighted Fair Queuing* (WFQ) scheduling [18] in the network, it is:

- $K_l = (y_{so}/T_{\sigma t}) + (Pmax/T_{\sigma t}) + (Pmax/V_l) + D_l + Dp_l$, if the link is adjacent to the traffic source

- $K_l = (Pmax/T_{\sigma t}) + (Pmax/V_l) + D_l + Dp_l$, otherwise (23)

, where y_{so} is the *bucket size* corresponding to the traffic source and Pmax is the *maximum packet size* in the network.

For the traffic classes resembling to the BE service class the link cost $K_l = (1/V_l)$ and the link costs in (22) and (23) can be used. For more details on link costs, taking into account QoS, see [19] and [20].

APPENDIX B

PFA finds a set of candidate admissible routes with or without QoS guarantees for each QoS traffic trunk. It is a step-by-step procedure based on the iterative execution of a modified version of the *Floyd-Warshall* (FW) *all-pairs shortest path algorithm* [21]. FW finds the shortest path between each source and destination in a network. PFA is described below.

1. Execute FW to the initial network topology.
2. Exclude a node or a link from the standard topology (as if it has failed) and execute the FW algorithm for the new topology.
3. Store the established source-destination paths by the FW algorithm.
4. Compare the established paths with the paths already included to the current set of candidate paths (if any exists).
5. The paths not previously established are included to the set of candidate paths.

6. Repeat steps 2 to 5, each time excluding a different node or link from the network, for a specified number of iterations.

FW estimates the total path delay summing together the link propagation delays along the path or equivalently the total path length, summing together the link lengths along the path, since basically link length accounts for link propagation delay. It also estimates the number of intermediate hops a candidate path has. Each hop is associated with a node and consequently with additional processing delay and jitter. Therefore, the smaller the number of hops of a path is, the less jitter and delay the traffic trunk(s) that use this path will experience. Besides, using a smaller number of hops increases the transmission reliability of the traffic trunk(s), since the probability of a failure on the path decreases.