ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Int. J. Electron. Commun. (AEÜ) III (III) III–III

International Journal AEÜ of Electronics and Communications

www.elsevier.de/aeue

On efficient traffic engineering with DV-based routing protocols in DiffServ-aware IP networks

Mirjana D. Stojanovic[a,*], Vladanka S. Acimovic-Raspopovic[b]

[a]*Mihailo Pupin Institute, Volgina 15, 11060 Belgrade, Serbia and Montenegro*
[b]*University of Belgrade, Faculty of Transport and Traffic Engineering,Vojvode Stepe 305, 11000 Belgrade, Serbia and Montenegro*

Received 27 October 2004

Abstract

In this paper, an approach to efficient traffic engineering in the DiffServ-aware network environment is proposed. We focus to distance vector-based routing protocols, considering both modifications of routing protocols needed to support path differentiation and traffic engineering methods relied on adjusting multiple per-link costs to particular network conditions. Further, a method for determining link cost of particular traffic class, as a unique generic function of the single generalized performance metric has been proposed. In order to achieve efficient traffic engineering, possible approximations of generic cost function and mappings of generalized to particular metrics have been proposed. Finally, prerequisites for implementing proposed approach have been discussed in the context of different administrative policies and time scales of their application.

Keywords: Differentiated services; Distance vector routing; Internet protocol; Quality of service; Traffic engineering

1. Introduction

The Internet Protocol (IP) technology has been widely accepted as a basis for service integration in the next generation of multiservice telecommunication networks. One of the key issues in multiservice IP networks concerns resolving problems of providing different quality of service (QoS) levels, in accordance with specific requirements of different applications and users. Considering scalability requirements, the approach to QoS provisioning comprises standardized [1], or proprietary architectures with differentiated services, i.e. DiffServ-aware architectures. QoS-aware routing refers to traffic aggregates, instead of individual flows.

Traffic engineering (TE) involves adapting of routing to network conditions in order to improve the overall network performance in the sense of increasing availability and throughput, minimization of packet loss and optimization of resource utilization. In this paper, an approach to efficient TE in the DiffServ-aware network environment is proposed. We focus to distance vector (DV)-based routing and propose several approaches to perform efficient TE based on link costs, expressed as a function of the single generalized performance metric (PM). The main objective is to provide a trade-off between achieving the required network performance and the implementation complexity.

The rest of the paper is organized as follows: Section 2 contains an overview of related work. The network management architecture is described in Section 3. Section 4 comprises proposals for TE methods with DV-based routing protocols in the DiffServ environment. In Section 5, link cost has been considered as a function of the single generalized PM. Simulation experiments and the obtained results have been presented in Section 6. TE policies concerning

* Corresponding author. Tel.: +381 11 775 460; fax: +381 11 2755 978.
E-mail address: stojmir@kondor.imp.bg.ac.yu (M.D. Stojanovic).

1434-8411/$ - see front matter © 2005 Elsevier GmbH. All rights reserved.
doi:10.1016/j.aeue.2005.08.001

implementation of proposed TE methods have been discussed in Section 7. Section 8 contains concluding remarks.

2. Motivation and related work

In the past few years, TE in IP-based networks has been widely addressed, including both intra-domain and inter-domain aspects. One approach concerns upgrading traditional IP routing protocols to support TE. A comprehensive research work is focussed to extensions of intra-domain link state routing protocols, like OSPF (Open Shortest Path First) and IS–IS (Intermediate System–Intermediate System). Recently proposed OSPF extension for intra-domain TE (OSPF-TE [2]) seems to provide more powerful and robust routing capabilities, on the count of acceptable additional protocol overhead [3].

Experimental studies have shown that traditional shortest path routing protocols such as OSPF and IS–IS can perform quite well, without any modifications, in combination with the TE [4]. This approach includes permanent monitoring of the traffic and topology, optimizing the set of the static link weights, and reconfiguring the routers statically, with new weight settings as needed.

Inter-domain TE has been explored more recently, in the context of BGP (Border Gateway Protocol) and its extensions to support TE, e.g. [5]. A heuristic algorithm for selection of the egress router that satisfies end-to-end bandwidth requirements, while optimizing the network resource utilization has been suggested in [6].

Alternative approach to TE is related with various propositions of QoS and constraint-based routing protocols [7,8]. In spite of significant advantages in terms of QoS provisioning and TE, the main drawbacks concern complex routing algorithms, and consequently implementation, as well as low scalability of the proposed protocols.

Another research topic is technology-specific and concerns networks based on the MPLS (Multi-Protocol Label Switching) and the emerging GMPLS (Generalized MPLS) [9,10]. They comprise explicit routing, based on pre-computed paths for specific types of traffic, with respect to their QoS requirements. In addition, proposals for TE in MPLS-based networks with DiffServ QoS architecture address an integrated QoS management based on both service demands and network conditions [11,12].

Following the basic idea presented in [4], which assumes Internet TE with link state protocols by adjusting link costs, our motivation for this work was to explore possibilities for TE based on similar principles, but in the DiffServ-aware environment. We suggest novel TE approaches that make a trade-off between fulfilling QoS requirements and the implementation complexity. We particularly focus to TE with DV-based routing protocols assuming calculating of DVs according to distributed Bellman–Ford's algorithm. This algorithm is applied in several well-known IP routing protocols including intra-domain protocols like RIP (Routing Information Protocol) as well as inter-domain protocols like BGP. BGP is frequently denoted as a path-vector protocol because its routing information includes the corresponding path. We will use the term "DV-based" protocols to denote a common class of protocols that rely on similar routing algorithms.

3. The network management architecture

Relationship of TE and network management process will be explained relying on the concept of QoS management suggested in [13] and extended in [14], where we have introduced a notion of the entity responsible for dynamic QoS management–QoS Manager (QM). TE methods proposed in this paper are also applicable to other similar architectures, e.g. [11,12]. QM entity encompasses the following functional components (see Fig. 1):

- *Service Level Agreement Manager* (*SLAM*): An entity that negotiates QoS with end users and other domains, by means of QoS signaling protocol, e.g. different variants of RSVP (Resource reSerVation Protocol) or some proprietary access signaling protocol.
- *Network Resource Manager* (*NRM*): An entity that decides about admission of new traffic flows to the network and resource allocation. NRM maintains a global view about the state of network resources, through the resource state table.
- *QoS Configuration Manager* (*QCM*): An entity that configures network elements for the new traffic flow and its associated traffic class.

Relationship of the TE process and QoS management is depicted in Fig. 1. TE performs adaptive management of traffic aggregates that are already present in the network, which involves short-term (seconds and minutes) controls

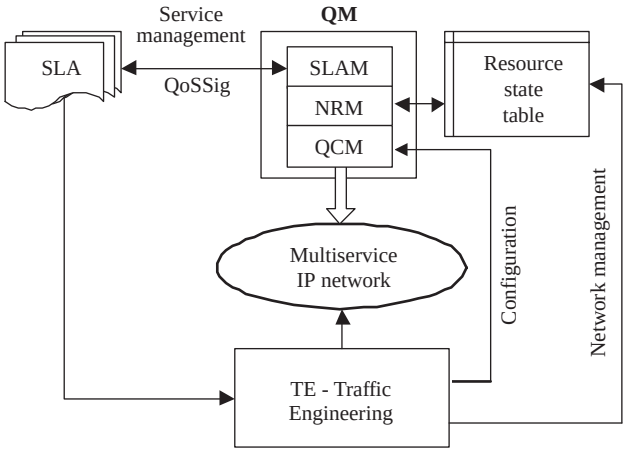


Fig. 1. Relationship of TE process and QoS management.

and medium-term (hours, days and weeks) updates of configurable parameters at network elements, e.g. routing tables, link costs, queuing and scheduling parameters, etc.

TE is also coupled with the traffic prediction process, which can be performed according to negotiated SLAs for individual traffic flows, metering and historical data. Traffic distribution matrix is created according to predicted traffic and the network is dimensioned based on that matrix. TE processes the results of long-term (months and years) performance monitoring, which may be used for network redesign in the sense of removing bottleneck links, adding new network elements, upgrades of software and hardware at the existing equipment, etc.

4. Proposals for TE methods with DV-based routing protocols

DV-based routing comprises a decentralized routing scheme, in which individual nodes have no information on the entire network topology. Each node knows only its direct neighbors, exchanges its DVs with them and keeps a table containing distances to all other nodes via each neighbor. DV calculation typically relies on the Bellman–Ford's algorithm, which functions as follows. Optimal path between a source X and destination Y is chosen from the set of all available paths, based on the minimum distance criterion. The minimum distance, $D(Y, X)$, represents a path with minimum sum of link costs, which is calculated from the expression

$$D(Y, X) = \min_{X_i} D^X(Y, X_i), \tag{1}$$

where X_i denotes each neighbor of node X, while $D^X(Y, X_i)$ represents DV from the node X to the node Y, across the node X_i, which is calculated as follows:

$$D^X(Y, X_i) = c(X, X_i) + \min_{X_j} D^{X_i}(Y, X_j), \quad X_j \neq X, \tag{2}$$

where X_j denotes each neighbor of the node X_i, $c(X, X_i)$ is the cost of link (X, X_i), while $D^{X_j}(Y, X_i)$ represents distance from the node X_i to the node Y, across the node X_j. The calculation is iterated until the node Y is reached. DV-based protocols typically have the ability of dynamic traffic rerouting, by means of different variants of topology update algorithms, in the case of link failure or due to change of link cost. In that case, control packets with routing information are propagated through network only if changes of link cost do affect optimal paths.

If link cost reflects one or more relevant PMs, it is possible to adapt traffic routing dynamically, according to network conditions. We will denote TE approach from [4] with TES (Traffic Engineering with Shared link costs). TES deals with one cost per each link and does not assume any changes of the basic routing protocol. Recalculation of single link costs is performed on the basis of the overall measured or estimated link utilization.

4.1. *TEnE – Traffic Engineering with n-level cost differentiation on Each link*

The generic approach for TE in the DiffServ-aware environment relies on quite simple idea: if n is the number of traffic classes, n different costs should be defined for each link – one cost for each traffic class. We adopt the convention that class n corresponds to the best effort service, which is network default service. Each cost is calculated as a function of relevant PM, for the observed traffic class p. In this case, Eqs. (1) and (2) transform as follows:

$$D(p, Y, X) = \min_{X_i} D^X(p, Y, X_i), \quad p = 1, 2, \ldots, n, \tag{3}$$

$$D^X(p, Y, X_i) = c(p, X, X_i) + \min_{X_j} D^{X_i}(p, Y, X_j), \quad X_j \neq X. \tag{4}$$

The main advantage of TEnE approach refers to the ability of finding optimal paths for each traffic class, in the sense of particular PM. Basic routing algorithms should not be changed – they rely on the minimum distance criterion. However, generic approach suffers from drawbacks that concern scalability issues and processing requirements, due to enlargement of routing tables and additional computational and implementation complexity. We further propose three heuristic approaches that reduce complexity of the generic approach. They rely on reducing the number of routing differentiation levels and/or reducing the number of links with multiple costs.

4.2. *TE2E – Traffic Engineering with 2-level cost differentiation on Each link*

In a large number of networks, path differentiation can be performed only with two levels, where the first level generally refers to m classes ($1 \leqslant m < n$) and the second level refers to the rest of $n-m$ traffic classes. This approach should be typically applicable for guaranteeing delay performance at the first level and bandwidth performance at the second level. Eqs. (3) and (4) stand for calculating DVs, with $p \in \{1, 2\}$.

4.3. *TEnC – Traffic Engineering with n-level cost differentiation on Critical links*

In a properly designed network, the number of critical links concerning any particular PM is reasonably low. Besides, such links usually do not behave badly all the time, but only under certain worst-case conditions [8]. The set of potentially "bad" links can be identified and upgraded in the TE process. Hence, different costs should be applied for

different traffic classes only to paths that include potentially "bad" links. If n-level cost differentiation is applied only to critical links, then Eqs. (3) and (4) stand for the default class, n, with $p = n$. For any other class p, $1 \leqslant p < n$, we have

$$D(p, Y, X) = \begin{cases} \min_{X_i} D^X(p, Y, X_i), & \text{if there is at least one critical link from } X \text{ to } Y, \\ \min_{X_i} D^X(n, Y, X_i), & \text{otherwise,} \end{cases} \quad (5)$$

$$D^X(p, Y, X_i) = \begin{cases} c(p, X, X_i) + \min_{X_j} D^{X_i}(p, Y, X_j), & X - X_i \text{ is critical, with at least one critical link from } X_j \text{ to } Y, \\ c(p, X, X_i) + \min_{X_j} D^{X_i}(n, Y, X_j), & X - X_i \text{ is critical, without critical links from } X_j \text{ to } Y, \\ c(n, X, X_i) + \min_{X_j} D^{X_i}(p, Y, X_j), & X - X_i \text{ is critical, with at least one critical link from } X_j \text{ to } Y \end{cases} \quad (6)$$

with $X_j \neq X$.

In this case, the overall path distance could be a sum of costs representing one PM for "good" links and costs representing some other PM for "bad" links. This approach provides feasible path selection assuming two conditions: one concerning proper definitions of PMs and cost functions, as will be explained in Section 5 and the other concerning choice of appropriate PMs.

4.4. *TE2C – Traffic Engineering with 2-level cost differentiation on Critical links*

Further reduction of implementation complexity should be achieved by performing routing differentiation on two levels, with presumptions as in the case of the TE2E method, but only on critical links. This means that only a small number of paths will contain links with two different costs: one for the first routing level, typically representing delay performance and another for the second routing level, typically representing link utilization. Eqs. (5) and (6) stand for calculating DVs, with $p = 1$ and $n = 2$.

4.5. Implementation issues

Implementation issues concern size of routing tables, structure and throughput of routing control packets and complexity (algorithmic, computational, and operational). Qualitative comparison of different TE methods with respect to performance and implementation issues is supplied in Table 1. Routing tables should be enlarged exactly n times in the case of TEnE method and twice if TE2E is applied. With TE2C and TEnC, table size may be implementation-dependent; however reducing table size may increase processing requirements at network nodes.

With all TE methods, additional protocol overhead, due to information about the routing differentiation level is proportionally low, as indicated in Table 1.

High complexity of the generic method refers to multiple computational and processing requirements that may be significantly reduced by performing routing differentiation with only two levels (TE2E). Reducing complexity of TEnE with TEnC strongly depends on percentage of critical links in the network. Similar observations stand for TE2C in comparison with TE2E.

Finally, since proposed approaches rely on using single PM, there is no additional slowdown of the protocol convergence, which may be the problem if DV algorithm is extended with multiple metrics [15].

5. Cost functions

Let μ_p be a non-negative link metric that quantifies performance of the traffic class p, traversing any link in the network. Let M_p be maximum allowed value of the observed metric μ_p, hence $v_p = \mu_p/M_p$ represents the normalized value of μ_p. Suppose that the cost of any link $X_i - X_j$, for traffic class p, $c(p, X_i, X_j)$, can be expressed by generic function of normalized PM, i.e. $c(p, X_i, X_j) = c(v_p)$, for $v_p \geqslant 0$. Function $c(v_p)$ is positive, increasing and convex, for values $0 \leqslant p \leqslant 1$, with $c(0) = C_0$ and $c(1) = C_1$. For values $v_p > 1$, link cost is infinite, i.e. $c(v_p) = +\infty$. Thus, generic function $c(v_p)$ uniquely determines link cost for traffic class p, for any measured or estimated value of v_p. Regarding DV protocol, condition $c(v_p) = +\infty$ for $v_p > 1$ is equivalent with marking link out of order for class p. Concerning QoS routing, the above condition corresponds to constraint-based approach, meaning that each path that contains even one link with infinite cost is eliminated from the set of available paths.

In practice, a very small change of v_p in most cases causes also a small change of cost $c(v_p)$, which does not significantly influence the overall path cost. This conclusion leads to a more suitable, practical approach to TE, based on operating with a finite set of pre-computed discrete cost values. The objective is to avoid dynamic computations of costs as well as too frequent reconfiguration of routers and, consequently, increase of the control traffic intensity. Further, we want to distinguish the case of performance deterioration from link failure, therefore the infinite value of $c(v_p)$ should be replaced with the finite value $C_{\max}$ in the approximated function. Although this approach is not constraint-based, the overall cost of the path containing even one link with cost $C_{\max}$ should be so high that practically it will never be selected.

Such approach leads to approximation of the generic function $c(v_p)$ with a staircase-like function, $c_s(v_p)$. Function $c_s(v_p)$ takes $m + 1$ discrete values from the set $\{C_0, C_0 + \Delta C_1, C_0 + \sum_{j=1}^{m-1} \Delta C_j, \ldots, C_{\max}\}$, where $m \in \{1, 2, 3, \ldots\}$, $C_0 = c(0)$, ΔC_j represents the difference between two consecutive values of $c_s(v_p)$, $C_0 + \sum_{j=1}^{m-1} \Delta C_j < c(1) = C_1$, and

Table 1. Comparison of TE methods

TE	Routing differentiation field	Size of routing table	Exchange of control packets
TE*n*E	$\lceil \log_2 n \rceil$-bit	n times larger than in basic DV protocol	Up to n times more frequent than in TES
TE2E	1-bit	2 times larger than in basic DV protocol	Up to 2 times more frequent than in TES
TE*n*C	$\lceil \log_2 n \rceil$-bit or $\lceil \log_2 n \rceil + 1$-bit	Up to n times larger than in basic DV protocol	Less frequent than TE*n*E
TE2C	2-bit	Up to 2 times larger than in basic DV protocol	Less frequent than TE2E

$C_{\max} >> C_1$. Staircase-like function is defined as follows:

$$c_s(v_p) = \begin{cases} C_0, & 0 \leqslant v_p < v_{p,1}, \\ C_0 + \sum_{j=1}^{i} \Delta C_j, & v_{p,i} \leqslant v_p < v_{p,i+1}, \\ \quad \Delta C_j > 0, \quad i = 1, 2, \ldots, m-1, & \\ C_{\max}, & v_p \geqslant v_{p,m}, \quad v_{p,m} < 1, \end{cases} \tag{7}$$

where $\{v_{p,1}, \ldots, v_{p,m}\}$ represents set of the values of variable v_p in which function $c_s(v_p)$ changes its value. The set $\{v_{p,1}, \ldots, v_{p,m}\}$ should be determined with regards to $c(v_p)$, from the following conditions: (1) $c_s(v_{p,i}) = c(v_{p,i}) = C_0 + \sum_{j=1}^{i-1} \Delta C_j + \Delta C_i/2$, for $i = 1, 2, \ldots, m-1$; and (2) $c_s(v_{p,m}) = c(v_{p,m}) = C_0 + \sum_{j=1}^{m-1} \Delta C_j + (C_1 - C_0 - \sum_{j=1}^{m-1} \Delta C_j)/2$, i.e.

$$v_{p,i} = \begin{cases} c^{-1}(C_0 + \sum_{j=1}^{i-1} \Delta C_j + \Delta C_i/2), & \\ \quad i = 1, 2, \ldots, m-1, & \\ c^{-1}((C_0 + C_1 + \sum_{j=1}^{i-1} \Delta C_j)/2), & \\ \quad i = m, & \end{cases} \tag{8}$$

where notation c^{-1} stands for the inverse function of generic cost function c.

Let us now explore a particular class of generic cost functions defined as follows:

$$c(v_p) = \begin{cases} C_0 + (C_1 - C_0) v_p^a, & 0 \leqslant v_p \leqslant 1, \quad a \geqslant 1, \\ \quad c(0) = C_0 \geqslant 0, \quad c(1) = C_1 > C_0, & \\ +\infty, & v_p > 1, \end{cases} \tag{9}$$

which, for $0 \leqslant v_p \leqslant 1$ gives a family of strictly convex power functions for all $a > 1$ and a linear function for the boundary value $a = 1$. When approximating $c(v_p)$ with $c_s(v_p)$, we suppose a particular important case of Eqs. (7) and (8), which assumes equidistant intervals of cost change such that $\Delta C_j = (C_1 - C_0)/m$, for $j = 1, 2, \ldots, m-1$. In this case, Eq. (7) is transformed as follows:

$$c_s(v_p) = \begin{cases} C_0, & 0 \leqslant v_p < (1/2m)^{1/a}, \\ C_0 + (i/m)(C_1 - C_0), & ((2i-1)/2m)^{1/a} \\ \quad \leqslant v_p < ((2i+1)/2m)^{1/a}, & \\ \quad i = 1, 2, \ldots, m-1, & \\ C_{\max}, & v_p \geqslant (1 - 1/2m)^{1/a}. \end{cases} \tag{10}$$

According to Eq. (10) the set of link costs can be precalculated using the set of only five parameters, i.e. $\{a, m, C_0, C_1, C_{\max}\}$. $C_{\max}/C_1$ should be larger at least an order of magnitude than maximum number of hops between any pair {source, destination}, to assure exclusion of any path containing even one link with $v_p \geqslant v_{p,m}$. Examples of generic cost functions defined with Eq. (9) and their approximations according to Eq. (10), with different values of parameter a, are illustrated in Fig. 2.

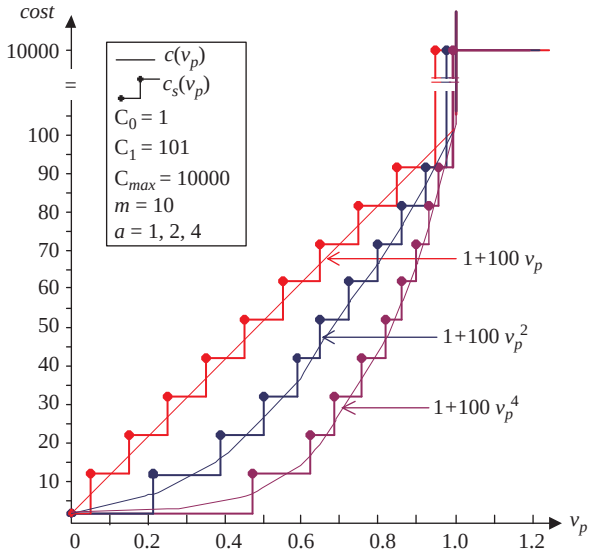


Fig. 2. Examples of generic cost functions and their approximations with staircase-like functions.

We will further discuss particular PMs, supposing that edge-to-edge (E2E) network performance bounds are known. Link cost should regularly be a function of bandwidth consumption, if bandwidth for the observed traffic class is not reserved. In order to adjust to function $c_s(v_p)$, defined by Eq. (7), metric v_p should be expressed through the offered traffic load L_p normalized to bandwidth, B_p, i.e. $v_p = L_p/B_p$, meaning that link cost will become extremely high if the link bandwidth is almost exhausted. This strategy should usually be applied for the best effort service or for several classes with lower priorities, when link cost expresses a measure of common available bandwidth shared between those classes.

If the bandwidth for particular class is reserved, cost function can express another PM, e.g. delay, which is typical for premium service. Measured or estimated link delay, d_p, should be normalized to the upper delay bound of aggregated flow that traverses particular link. If h is the maximum number of hops between any source and destination and $D_{p,\mathrm{E2E}}$ is the maximum E2E network delay for premium service, then $v_p = hd_p/D_{p,\mathrm{E2E}}$. Similar considerations stand for delay variation (jitter).

Another important case refers to loss-sensitive traffic. Considering reserved bandwidth for traffic class p, metric v_p should be expressed as a ratio of the link packet loss probability for that class, P_{lp}, and the maximum allowed link packet loss probability. If h is the maximum number of hops and $P_{lp,\mathrm{E2E}}$ represents maximum E2E packet loss probability for class p, then $v_p = P_{lp}/[1-(1-P_{lp,\mathrm{E2E}})^{1/h}]$.

Variable v_p can also express normalized value of a single mixed metric (SMM) [15], if defined as a suitable mathematical function of two or more single PMs. Adjusting weighting factors of each metric may increase the influence of particular QoS requirement. For example, additive PMs like delay d_p and jitter j_p can be mixed, such that $v_p = h(w_1 d_p + w_2 j_p)/(w_1 D_{p,\mathrm{E2E}} + w_2 J_{p,\mathrm{E2E}})$, where weighting factors w_1 and w_2 determine the importance of particular PM in the overall cost, for traffic class p.

6. Simulation and results

Extensive simulations have been carried out to explore the effectiveness of our approach. The network simulator ns-2 [16] has been used together with the Trace Graph analyzer of ns-2 trace files [17]. The original ns-2 DV routing modules have been extended to support proposed TE methods. Packet header and link state structures have also been modified to allow multiple traffic classes and link costs, respectively. TE process has been settled in the simulation script. Input parameters for calculating costs in the TE process have been obtained by the Trace Graph.

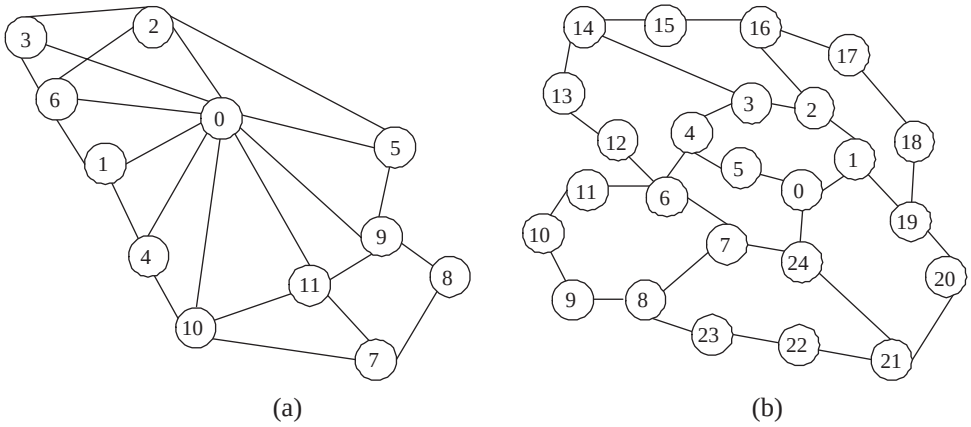
Simulated topologies are presented in Fig. 3. All nodes in NET1 and NET2 are interconnected by duplex 34 Mb/s links, with delays in the interval [5 ms, 10 ms], including propagation and processing at network nodes. Packet loss probabilities of individual links in both networks vary in the interval $[10^{-7}, 5{*}10^{-7}]$. The default ns-2 FCFS (First-Come First-Served) scheduling and drop-tail queue management has been applied in all nodes, with queue size 50 packets. Costs have been modeled by the set of functions defined by Eq. (10) with $C_0 = 1$, $C_1 = 101$, $C_{\max} = 10\,000$ and $a \in \{1, 2, 4\}$. Specification of traffic distribution matrices for all experiments is provided in the Appendix.

In the first set of experiments, handling of traffic concentration over critical links has been compared for different TE methods. Approximately 10% of potentially critical links, with respect to bandwidth utilization, have been assumed. IP traffic has been simulated by FTP sources, attached to the TCP agents, with packet size 1000 Byte. Four service classes have been supposed – premium service (PS) for delay-sensitive traffic, gold service (GS) and silver service (SS) for loss-sensitive traffic and best effort service (BE). Equal number of FTP sources per each class, between each pair {source, destination}, has been activated successively, with ingress rates 0.5 Mb/s for PS, GS and SS and 1.5 Mb/s for BE traffic. Pairs {source, destination} have been determined in such fashion that traffic concentration affects critical links, while the average utilization with respect to all links remains proportionally low. The number of per-class sources has been varied from 10 to 30 to simulate different offered loads on critical links, assuming equal link costs, set to minimum value 1, without TE. Thus, the load factor LF, which represents the ratio of offered load on critical link and link capacity, has been varied in the interval [0.5, 1.5].

Duration of each experiment is 25 s. For both topologies, recalculation of link costs has been performed at the equidistant time intervals, $T_{\mathrm{calc}} = 5$ s. Note that frequency of cost computation for simulation purposes has been chosen much higher than the one normally expected in a real network. Discussion on TE policies that determine interval of cost recalculation is supplied in Section 7.

Depending on TE method, link costs are recalculated according to maximum value of the normalized relevant PM at the observed time interval, T_{calc}. For purpose of determining v_p we have assumed maximum allowed per-class link bandwidths and mappings of PMs as described in Section 5. If B is the overall link capacity, then we suppose $B_1 = B_2 = B_3 = 0.15B$, while $B_4 = 0.55B$, where indexes 1, 2, 3 and 4 denote PS, GS, SS and BE, respectively. For TES, common link cost has been determined according to overall link utilization. For generic method TE4E, link delay for PS has been estimated taking into account queue length and service rate equal to link capacity. Normalized metric v_1 has been obtained with respect to maximum allowed E2E delay and maximum possible number of hops in each network. Packet loss probabilities for GS and SS have been estimated according to overall link utilization, from the set of random values in the interval $[10^{-7}, 5 * 10^{-7}]$. Metrics v_2 and v_3 have been determined with respect to maximum allowed E2E packet loss probabilities for corresponding services. Normalized metric for BE, v_4, represents ratio of offered BE traffic and B_4. Same principle has been applied with TE4C, for estimating values $v_1 - v_4$ on critical links, while for other links common metric v_4 has been estimated based on the overall link utilization. With TE2E, v_1 has been determined as described above, while v_2 (common metric for GS, SS and BE) has been estimated based on utilization of available bandwidth for those three classes ($B_2 + B_3 + B_4$). With TE2C, such principle is applied on critical links, while for other links common metric v_2 is estimated based on the overall link utilization.

Relevant performance characteristics of PS, GS and SS are presented in Table 2. The values are normalized to required absolute values of particular metrics, i.e. maximum



(a) (b)

Fig. 3. Simulated topologies. (a) NET1 and (b) NET2.

Table 2. Maximum normalized E2E PM values

TE	PS delay		GS loss probability		SS loss probability	
	NET1	NET2	NET1	NET2	NET1	NET2
TES	0.731	1.420	0.599	1.799	0.059	0.179
TE4E	0.415	0.711	0.399	0.599	0.059	0.179
TE4C	0.508	0.820	0.499	0.699	0.059	0.179
TE2E	0.415	0.711	0.599	0.899	0.059	0.089
TE2C	0.508	0.820	0.599	0.899	0.059	0.089

allowed E2E delay for PS is 100 ms, while maximum allowed E2E packet loss probability equals 10^{-6} for GS and 10^{-5} for SS. When applied in NET2, TES exceeds allowed metrics boundaries for PS and GS, because link costs are determined only with respect to the overall link utilization. TE4E provides the best overall performance, with respect to particular QoS requirements. TE4C is slightly worse than TE4E, due to presence of paths that contain links with shared costs. TE2E provides the same delay performance as TE4E, while packet loss probability of GS is worse than TE4E, but still acceptable. Note that SS performs excellent in all cases, because of less rigid QoS requirements. In NET2, packet loss probability is even less with TE2E and TE2C, because of path sharing with more claimed GS.

Simulation results in Fig. 4 present maximum link utilization vs. load factor LF, when different TE methods are applied, considering all links in the network. Curve denoted by "Average" in Fig. 4 represents the mean of maximum utilizations with respect to all links that carry traffic, if TE is not applied. With all investigated methods, network congestion has been avoided, even in the case when critical link, without TE, should be overloaded 50% (LF = 1.5). TES provides most even link utilization, because it regularly changes shared costs only with respect to the overall link utilization. TE4E provides slightly higher maximum utilization than TES due to taking into account 4 different normalized PMs. Similar considerations stand for TE4C, for which higher maximal link utilization than in TE4E appears due to mixing different traffic classes on links with shared costs. In TE2E and TE2C, premium traffic has been associated with the first routing level, while other traffic classes have been associated with the second level. This is the worst case with regards to maximum link utilization. With TE2E, maximum utilization appears on links that carry mixed GS, SS and BE traffic. Finally, maximum utilization is the highest with TE2C, because of mixing traffic from all classes on "good" links, with shared costs between two routing levels. However, even with TE2C, maximum link utilization does not exceed 85.1% for NET1 and 77.5% for NET2. Compared to the average link utilization, larger network NET2 behaves worse. For example, although cost of link 0–1 rises, alternative path 0–5–4–3–2–1 may still have higher overall cost, hence traffic should be routed over link 0–1. The ratio of maximum link utilization with TE and the average link utilization varies in the interval [1.85, 2.55] for NET1 and [2.87, 4.45] for NET2.

For all TE methods, the best results in the sense of reducing maximum link utilization have been obtained for $a = 1$. This happens due to linear change of link costs, which enables more even distribution of the overall traffic. For $a = 4$, maximum link utilization is considerably higher for lower LF, than if $a = 2$ or 1, due to high values of initial threshold that triggers change of minimum cost. In this case, considering Eq. (10), link cost remains minimal (C_0) until PM

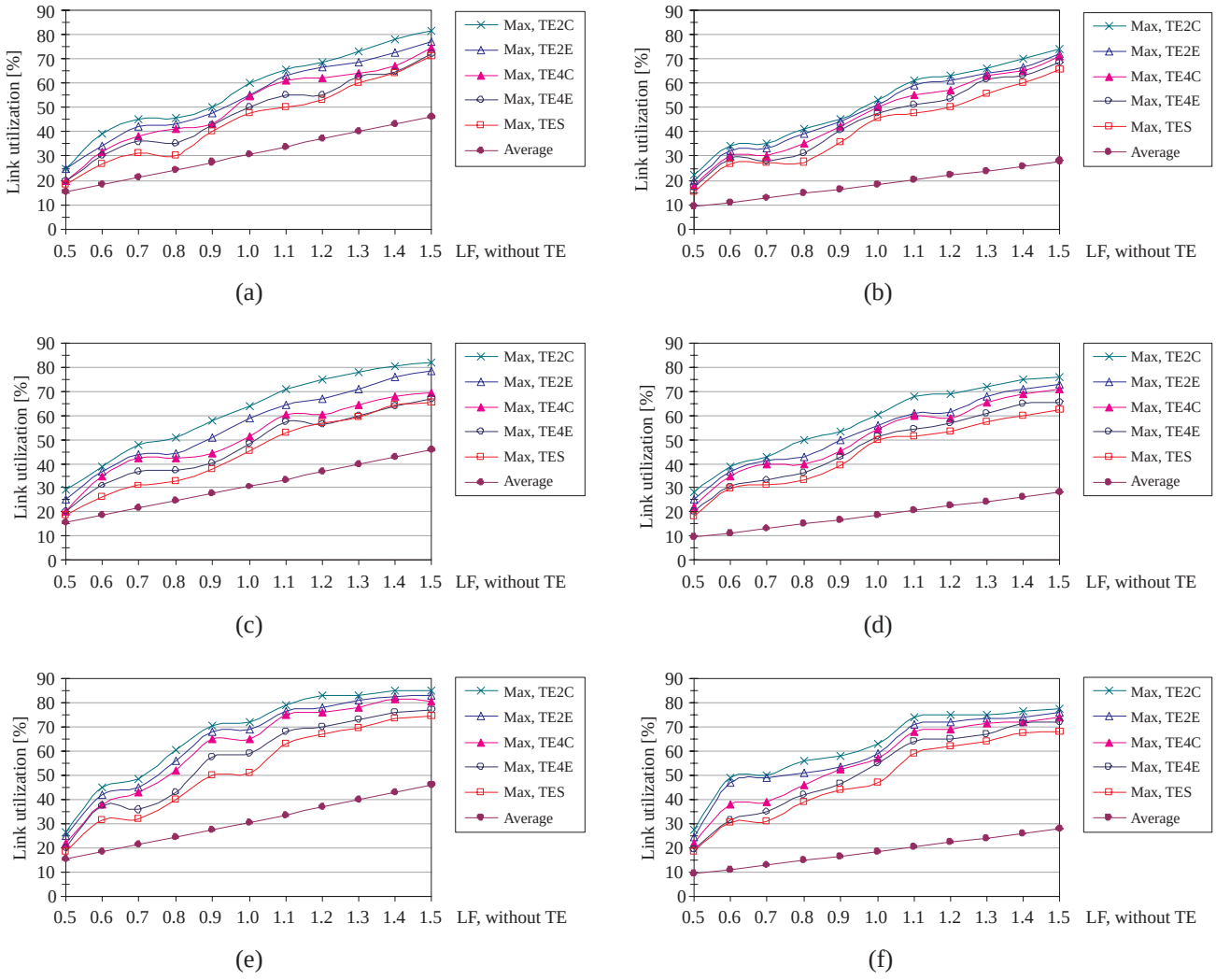


Fig. 4. Link utilization of NET1 and NET2 vs. ratio offered load/link capacity, without TE for different values of cost parameter a: (a) NET1, $a = 1$. (b) NET2, $a = 1$. (c) NET1, $a = 2$. (d) NET2, $a = 2$. (e) NET1, $a = 4$ and (f) NET2, $a = 4$.

v_p reaches $v_{p,1} = 0.473$, and after that it keeps the value $C_0 + \Delta C_1$ until v_p reaches $v_{p,2} = 0.622$. Besides, the "swing" effect appears which is manifested by successive fluctuation of certain amount of traffic between two links as their costs oscillate among the two lowest values, C_0 and $C_0 + \Delta C_1$. This phenomenon is more expressed in smaller and star-like NET1 topology, where "swing" effect occurs e.g. between links 0–9 and 0–10 or between links 0–11 and 0–10. With $a = 4$, the "saturation" effect appears for higher values of LF, which is manifested with almost constant maximum link utilization when overload factor LF changes its value from 1.1 to 1.5. This happens because link costs change rather frequently for higher values of v_p ($v_p > 0.7$), thus allowing more even traffic distribution among links. Maximum link utilization should be almost equal for LF = 1.4 or LF = 1.5, with all investigated values of a. This leads to a conclusion that choice of parameter a depends on the operator's policy. If the objective is to provide fair link utilization for all offered traffic loads, lower values of parameter a should be chosen. If the policy is to react only when a real threat of congestion appears, higher values of parameter a may be selected.

In the second set of experiments, we have investigated intensity of control traffic, for different TE methods and values of cost parameter a. We define normalized number of control packets, N_{cp}, as a ratio of the number of all generated control packets when TE is applied and the number of all generated control packets when TE is not applied, with respect to all T_{calc} intervals. When TE is not applied, cost of each link is common for all traffic classes and it does not change during simulation. The overall number of control packets generated in the interval T_{calc} then depends on the network topology and the period of DVs exchange between adjacent nodes, which is protocol implementation dependent, but typically equals couple of seconds (1–2 s). Assumed simulation conditions concerning network topologies, number of classes,

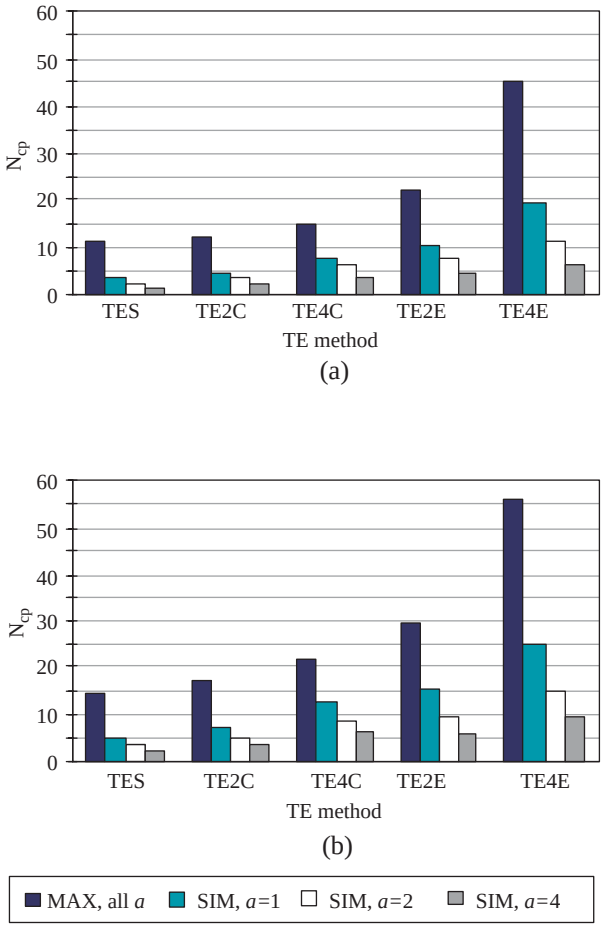


Fig. 5. Normalized number of overall generated control packets for different TE methods: (a) NET1 and (b) NET2.

traffic sources, percentage of critical links and cost functions are same as in the first set of experiments, with LF = 1.0. Simulation results are presented in Fig. 5. The values of N_{cp} denoted by "MAX" correspond to the theoretically worst case, when all link costs and all minimum DVs are changed. N_{cp} values denoted by "SIM" have been obtained by simulation. Lower values of N_{cp} for higher a appear as a result of less frequent alterations of DVs. Approximations of TE4E allow generation of lower number of control packets, and consequently less N_{cp}, with best results when both number of routing differentiation levels and number of links with multiple costs are reduced.

We have further investigated processing loads at the individual nodes, related with control packets. Processing load at each node has been estimated by means of the Trace Graph analyzer, on the basis of number of received and generated control packets during the worst interval T_{calc}, i.e. interval in which the highest number of control packets has been processed. Simulation results have been presented in Table 3, for LF = 1.0 and cost parameter $a = 2$. Maximum processing load refers to a node with the highest estimated load, while average

Table 3. Normalized processing loads at network nodes

TE method	Maximum		Average	
	NET1	NET2	NET1	NET2
TE4E	4.875	3.915	2.449	2.373
TE2E	2.925	2.421	1.518	1.417
TE4C	2.471	1.985	1.345	1.455
TE2C	1.725	1.422	1.181	1.241

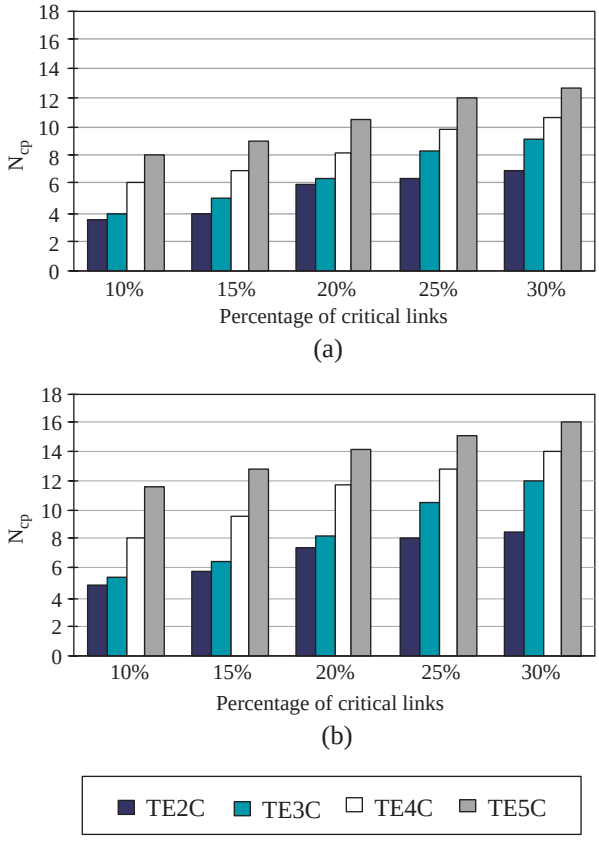


Fig. 6. Normalized number of overall generated control packets vs. percentage of critical links: (a) NET1 and (b) NET2.

load denotes mean of maximum loads with respect to all network nodes. Both maximum and average processing loads have been normalized to corresponding ones, obtained without applying TE. Higher maximum processing loads in NET1 appear due to node 0 (see Fig. 3), which is interconnected with nine neighbors and receives more control packets than nodes in NET2, which have up to four neighbors. However, both topologies experience similar average node processing loads for the same TE method.

In the third set of experiments, efficiency of TE2C and TE*n*C methods has been explored, depending on number of critical links. Fig. 6 presents N_{cp} as a function of percentage of critical links, considering LF = 1.0 and $a = 2$. In TE*n*C,

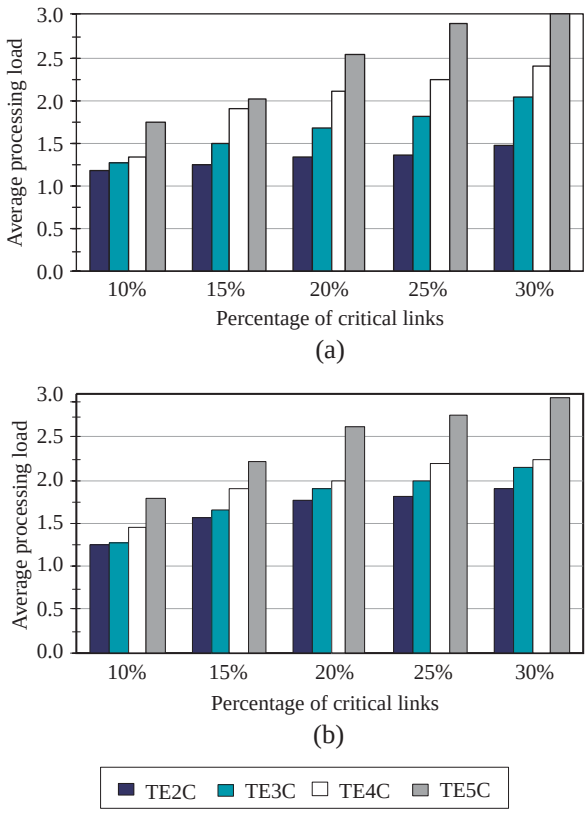


Fig. 7. Normalized average processing load vs. percentage of critical links: (a) NET1 and (b) NET2.

number of traffic classes, n, has been varied from 3 to 5. Results presented for TE2C method refer to approximation of TE4E. Number of control packets rises as number of path differentiation levels and number of critical links is being increased. Average processing loads at network nodes, normalized to ones experienced without TE, vs. different percentage of critical links, have been presented in Fig. 7. The obtained values are equivalent for both topologies, with remarkable advantages of TE2C and TEnC in comparison with TE2E and TEnE, for lower percentage of critical links. If percentage of critical links achieves 25% in NET1 and 30% in NET2, N_{cp} and processing loads with TE2C and TEnC become nearly same as with TE2E and TEnE,
respectively.

7. TE policies

In order to implement proposed approach, the following issues have to be addressed with respect to time-scales of their application: selection of suitable TE method, definition of cost function $c_s(v_p)$, specification of relevant PMs, computation of link costs and definition of critical links. Those items basically constitute an administrative TE policy, since they comprise a set of rules needed to manage TE process. An overview of TE policy elements and time scales of their application are presented in Table 4.

Routing protocol may optionally support all proposed TE methods, thus allowing network administrator to select the most appropriate one, on a long-term basis. Criteria for selecting parameters of cost function $c_s(v_p)$ as well as mappings of generalized to particular PMs have been addressed in Sections 5 and 6.

Cost recalculation frequency should be a trade-off between network stability and performance optimization objectives. DV-based protocols suffer from convergence problems that may jeopardize network stability. Duration of transition period, caused by propagation of DVs through network after change of link costs, strongly depends on network topology, number of changed costs and link delays. Transition periods with proposed TE methods do not considerably enlarge in comparison to the one experienced with TES, because basic routing algorithm does not change. In simulation experiments from Section 6, they were in the interval [0.55 s, 0.8 s] for 12-node topology NET1 and [0.7 s, 1.0 s] for 25-node topology NET2. We do not foresee short-term changes of link costs, except in the case of failures or heavy congestion.

Link costs need to be carefully determined with the objective to keep network performance stable, in spite of short-term variations of traffic intensity and offered link loads. Considering cost functions $c_s(v_p)$, the problem deduces to adequate estimation of relevant value of v_p on each link. Another important question concerns the number of costs (regarding all links in the network) that should really be changed after each recalculation [4]. In Section 6, we have shown the impact of cost parameter a to number of cost changes. In the DiffServ-aware network, TE policy might prefer rather rare changes of certain link cost related with particular class, thus keeping estimated v_p and its associated cost nearly constant in long-term period. Anyway, network management system must permanently have precise and upgraded information about the state of the overall network and its performance.

Criteria for definition of critical links in TEnC and TE2C methods may be related with embedded long-term link features. For example, backbone links with lower overall capacities potentially represent network bottlenecks. The objective of TE policy, realized through medium-term assignment of multiple costs to such links, should be to disburden them from certain amount of traffic, or to avoid forwarding of traffic with hard QoS requirements over them. In other cases, critical links may be defined on the basis of their experienced or predicted large utilization. There is no benefit of TEnC or TE2C if critical links have to be redefined almost each time when costs are recalculated. TE process is iterative and proper selection of critical links has to be learned through several steps performed in cooperation with the analysis of

Table 4. Elements of TE policy

Element	Time scale	Remark
TE method	Long-term	If protocol supports different TE methods
Cost function $c_s(v_p)$	Long-term	For example, set $\{a, m, C_0, C_1, C_{\max}\}$
Set of PM	Long-term	One PM per class or routing differentiation level
Cost recalculation	Medium-term or short-term	Short-term, occasionally (failures, heavy congestion)
Redefinition of critical links	Long-term or medium-term	Only with TE*n*C or TE2C

Table A1. Specification of traffic matrices in simulation experiments

Experiment	NET1		NET2	
	Traffic flow $S_{n,p,i} \rightarrow D_{n,p,j}$	Critical links	Traffic flow $S_{n,p,i} \rightarrow D_{n,p,j}$	Critical links
First set	$p = 1, 2, 3, 4$		$p = 1, 2, 3, 4$	
	$n = 1, 2, \ldots, 30$	0–9	$n = 1, 2, \ldots, 30$	1–19, 19–20
	$i \in \{0, 1, \ldots, 6\}$	0–11	$i \in \{0, 1, \ldots, 7, 11, 12, \ldots, 20, 21, 24\}$	21–22, 24–21
	$j \in \{7, 8, \ldots, 11\}$		$j \in \{19, 20, \ldots, 23\}$	
Second set	$p = 1, 2, 3, 4$		$p = 1, 2, 3, 4$	
	$n = 1, 2, \ldots, 20$	0–9	$n = 1, 2, \ldots, 20$	1–19, 19–20
	$i \in \{0, 1, \ldots, 6\}$	0–11	$i \in \{0, 1, \ldots, 7, 11, 12, \ldots, 20, 21, 24\}$	21–22, 24–21
	$j \in \{7, 8, \ldots, 11\}$		$j \in \{19, 20, \ldots, 23\}$	
Third set	$p = 1, 2, 3, 4$	0–9, 0–11	$p = 1, 2, 3, 4$	1–19, 19–20, 21–22,
	$n = 1, 2, \ldots, 20$	1–4, 5–9,	$n = 1, 2, \ldots, 20$	24–21, 18–19, 6–7,
	$i \in \{0, 1, \ldots, 6\}$	9–8, 10–11,	$i \in \{0, 1, \ldots, 7, 11, 12, \ldots, 20, 21, 24\}$	17–18, 4–5, 4–6,
	$j \in \{7, 8, \ldots, 11\}$	11–7	$j \in \{19, 20, \ldots, 23\}$	22–23, 2–1

the overall network behavior. Therefore, we rather foresee long-term redefinition of critical links. Medium-term redefinition of critical links may be needed occasionally, e.g. after network failures or under heavy congestion.

8. Conclusions

Different TE methods, relying of adaptation of multiple per-link costs, with DV-based routing protocols in DiffServ-aware IP networks have been proposed and examined. With the suggested approach, efficient TE can be achieved with slight modifications of the existing routing protocols. Complexity of the generic TE method with separate link costs for each traffic class may be reduced by decreasing number of routing differentiation levels and/or number of links with multiple costs. Implementation assumes definition of operator-specific TE policy which encompasses: selection of the appropriate TE method, definition of cost function, specification of PMs, recalculation of link costs and definition of critical links. All those elements are related with corresponding time scales.

Simulation results indicate that long-term selection of particular approximate TE method should depend on QoS requirements, number of critical links, number of traffic classes and scalability issues. Approximation of generic TE method by reducing number of links with multiple costs is effective only with up to 20% of critical links in the network.

The proposed method for determining link cost of particular traffic class relies on suitable staircase approximation of the generic cost function, which allows TE with a finite set of pre-calculated costs. A notion of the single generalized PM has been introduced as a basis for unified calculation of each cost by means of a suitable cost function. Generalized metric is normalized to maximum allowed value on each link and fits to the additive composition rule that stands for the overall path cost. Simulation results indicate that choice of particular cost function depends on long-term performance optimization objectives.

The most difficult task in specifying TE policy refers to appropriate estimation of relevant PMs, based on which costs are recalculated. Different policies related with medium term cost recalculation have been discussed. They may be preventive or reactive, while both of them have to deal with number of costs that should really be changed after each recalculation, in order to fulfill QoS requirements and preserve network stability.

Acknowledgements

The authors wish to thank the anonymous reviewers for their suggestions and comments that really helped them in improving contents and presentation of this paper.

Appendix

Specification of traffic matrices in simulation experiments is presented in Table A1. Traffic flow is specified by flow source $S_{n,p,i}$ and flow destination $D_{n,p,j}$, where n denotes ID number of traffic flow, p denotes ID number of traffic class, while i and j denote ingress and egress nodes, respectively, according to Fig. 3. Different combinations $(i, j), i \neq j$, were used to simulate different traffic concentration on individual links. In the third set of experiments, number of critical links has been varied in the intervals [2, 7] and [4, 11] in NET1 and NET2, respectively.

References

[1] Blake S, et al. An architecture for differentiated services. RFC 2475. IETF; 1998.
[2] Katz D, Kompella K, Yeung D. Traffic engineering extensions to OSPF version 2. RFC 3630. IETF; 2003.
[3] Alnuweiri H, Wong L-YK, Al-Khasib T. Performance of new link state advertisement mechanisms in routing protocols with traffic engineering extensions. IEEE Commun Mag 2004;42(5):151–62.
[4] Fortz B, Rexford J, Thorup M. Traffic engineering with traditional IP routing protocols. IEEE Commun Mag 2002;40(10):118–24.
[5] Xiao L, et al. QoS extension to BGP. In: Proceedings of the 10th international conference on network protocols, 2002. p. 100–9.
[6] Ho K-H. Multi-objective egress router selection policies for inter-domain traffic with bandwidth guarantees. Center for Communication Systems Research; 2004.
[7] Srivastava S. et al. Benefits of traffic engineering using QoS routing schemes and network controls. Comput Commun 2004;27(5):387–99.
[8] Xiao X. Providing quality of service in the Internet. Dissertation. Michigan, USA: Michigan State University; 2000.
[9] Awduche D, et al. Requirements for traffic engineering over MPLS. RFC 2702. IETF; 1999.
[10] Iovanna P, Sabello R, Settembre M. A traffic engineering system for multilayer networks based on the GMPLS paradigm. IEEE Network 2003;17(2):28–37.
[11] Trimintzios P. et al. A management and control architecture for providing IP differentiated services in MPLS-based networks. IEEE Commun Mag 2001;39(5):80–8.
[12] Trimintzios P. et al. Service-driven traffic engineering for intradomain quality of service management. IEEE Network 2003;17(3):29–36.
[13] Stojanovic M, Acimovic-Raspopovic V. A novel approach for providing quality of service in multiservice IP networks. Int J Facta Universitatis: Ser Electron Energetics 2004;17(2): 261–74.
[14] Stojanovic M, Acimovic-Raspopovic V. An approach to dynamic QoS management in multiservice IP networks. Telecommunications (CYPTT) 2004;49(2):9–15.
[15] Costa L, et al. Distance-vector QoS-based routing with three metrics. In: Proceedings of the IFIP networking, Paris; 2000.
[16] Network simulator ns-2 and Network Animator NAM. [Online]. Available: http://www.isi.edu/nam.
[17] Malek J. TraceGraph – network simulator ns trace files analyzer. [Online]. Available: http://www.geocities.com/tracegraph, 2003.

Mirjana Stojanovic received her B.Sc. (1985) and M.Sc. (1993) degrees in Electrical Engineering and her Ph.D. degree (2005) in Technical Sciences, all from the University of Belgrade. Currently, she is head of the Telecommunication Networks group at the Mihailo Pupin Institute, Belgrade. She managed or participated in several major national projects concerning telecommunication networks design and implementation. Her research interests include communication protocols, IP quality of service and traffic engineering as well as network management systems.

Vladanka Acimovic-Raspopovic received her B.Sc. (1976) and, M.Sc. (1984) degrees in Electrical Engineering and her Ph.D. (1995) degree in Technical Sciences, all from the University of Belgrade. She is an Associate Professor at the University of Belgrade, Faculty of Transport and Traffic Engineering (Telecommunication Networks Department). She managed or participated in 26 major national research projects and studies concerning radio and optical transmission systems and telecommunication networks design and implementation. Her research interests are in the field of performance evaluation, QoS routing and tele-traffic engineering in next generation broadband networks.