Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Communications xxx (2006) xxx–xxx
computer communications
www.elsevier.com/locate/comcom



# Traffic engineering with OSPF-TE and RSVP-TE: Flooding reduction techniques and evaluation of processing cost

Stefano Salsano [a,*], Alessio Botta [b], Paola Iovanna [c], Marco Intermite [b], Andrea Polidoro [a]

[a] *DIE – Università di Roma "Tor Vergata," Dip. Ingegneria Elettronica Via di Tor Vergata 110, 00133 Rome, Italy*
[b] *CoRiTeL – Consorzio di Ricerca sulle Telecomunicazioni*
[c] *Ericsson Lab Italy S.p.A.*

Received 11 July 2003; received in revised form 6 December 2005; accepted 7 December 2005

**Abstract**

This paper considers two important aspects related to the control plane of Traffic Engineered IP/MPLS networks: the "flooding reduction" mechanisms and the evaluation of processing cost for signaling and routing protocols. The flooding reduction mechanisms are needed to reduce the amount of information exchanged by Traffic Engineering enabled routing protocols. The trade-off between the amount of information exchanged and the network performance (connection blocking probability) is discussed in the light of specific aspects of OSPF-TE routing protocol and RSVP-TE signaling protocol. Different mechanisms are analyzed and a suggestion is given for the best one. The dynamic aspects related to the time needed to distribute the routing and signaling information are considered. Finally, the combined processing cost of routing and signaling is analyzed, and the possible bottlenecks of the architecture are discussed. It is worth mentioning that the discussed results have been derived not only with simulation/analysis but also with measurements coming from a testbed implementation.


*Keywords:* MPLS traffic engineering; OSPF-TE; RSVP-TE

## 1. Introduction

The so-called "new generation networks" handle a huge amount of IP traffic, a large portion of this traffic demands more than "best effort" service (for example QoS and reliability). Multi-Protocol Label Switching (MPLS) technology [1] can be useful to cope with these requirements. MPLS can enable smart Traffic Engineering (TE) [2,3] strategies, which handle in the most flexible way the network resources, and react dynamically to traffic changes. In this advanced scenario, paths for traffic flows can be chosen according to some optimality criteria by the so-called Constraint Based Routing (CBR) algorithm. The input to the CBR algorithm is the information about the status of the network that is distributed in real-time by the routing protocol. The paths are dynamically setup and released by means of a proper signaling protocol. Each MPLS-TE enabled node supports both a routing protocol and a label distribution protocol. The possible routing protocols are OSPF-TE [4] and ISIS-TE [5], which extend OSPF and IS-IS respectively. Specifically, the traditional routing protocols have been enhanced with the ability to carry information related to link attributes/states, to be used for explicit route calculation (e.g., available/reserved bandwidth). The label distribution protocol (or "signaling" protocol) is used to setup the so called Label Switched Paths (LSPs), supporting both explicit route indication and reservation of resources during dynamic LSP setup. RSVP-TE [6] and CR-LDP [7] are the two "TE-capable" label distribution protocols. In the following we will always consider OSPF-TE as the routing protocol and RSVP-TE as the signaling/label distribution protocol. This is consistent with the decisions in IETF to continue with the standardization of RSVP-TE rather than CR-LDP [8]. Fig. 1 provides a representation of the logical entities involved in the TE pro-

* Corresponding author. Tel.: +39 06 7259 7450; fax: +39 06 7259 7435.
*E-mail address:* stefano.salsano@uniroma2.it (S. Salsano).

0140-3664/$ - see front matter 
doi:10.1016/j.comcom.2005.12.007

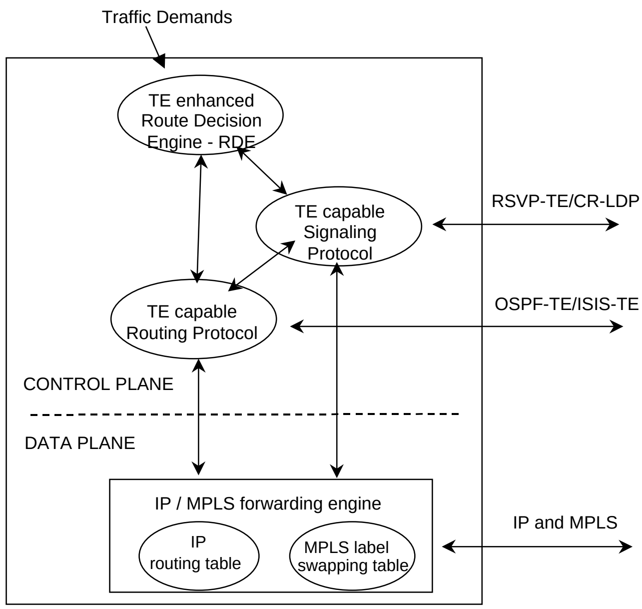


Fig. 1. Architecture of a TE enabled node (LER case).

cess and of their relationships (including the "data plane" elements).

We assume that Edge Nodes (LER – Label Edge Routers) receive the indication of the "Traffic Demands" to be supported, and that this is a dynamic process. Note that in this context a Traffic Demand (i.e., a *flow*) is typically an *aggregate* of several IP micro-flows. Once a request has been presented to an Edge Node, we assume that a logical entity, that will be referred to as "Route Decision Engine" (RDE), chooses the proper route within the network.[1] The RDE gathers the information related to the current topology and resource usage in the network by continuous interaction with the TE capable routing protocol (OSPF-TE in our assumption). When the RDE has chosen the route for a Traffic Demand, the corresponding LSP will be setup using RSVP-TE protocol, which will take care of performing node-by-node admission control and actual resource allocation. OSPF-TE advertises the change of local resource allocation status to all other LSRs by sending a Link State Update (LSU) message containing a special kind of Link State Advertisement (LSA) object called *opaque LSA* [8]. The object is called opaque because it is "hidden" to the basic OSPF routing logic, as it is only used by the TE logic. The LSU message is distributed to all LSRs using the OSPF "flooding" procedure. In order to avoid that the information flooding is executed for each minimal change, some "flooding reduction" mechanisms need to be used, so that the origination rate of OSPF-TE LSU messages can be reduced.

The basic method to address the signaling flooding problem is the distribution of a "coarser" link-state information. This can be accomplished either with a static set of thresholds or with "dynamic" thresholds, by considering the relative variation with respect to the older information. We compare these two approaches, showing that the dynamic approach performs slightly better that the fixed thresholds approach and it is much easier to manage and tune. We will show that these mechanisms can reduce the amount of flooding in a network by a large factor (e.g., by 5 or 10 times).

After presenting the network and traffic models in Section 2, in Section 3 we will analyze the performance in terms of call blocking probability covering the trade-off between signaling load and performance. Our results are consistent to those described in the literature ([9–11]) but we introduce noteworthy contributions:

- the analysis of why the dynamic thresholds are preferable to the static one and the refinements of the static thresholds to reach the performance of the dynamic ones
- results coming both from simulation and from a testbed implementation with real measurements.

We observe that the traffic engineering process described so far is a highly distributed process, which can suffer of inconsistent co-ordination between the various elements. There are two possible sources of inconsistency that should be taken into account: the "*Information Propagation Time*" and the "*Imprecise Information*".

The *Propagation Time* problem is related to the time needed to propagate the information in the network via signaling and routing protocols. In the mean time when the information is not up-to-date, an Edge Node can take incorrect (or sub-optimal) route selection decision. Another similar problem is related to the race conditions between allocation requests coming from two different Edge Nodes and arriving to an internal node almost in the same time, when resources are not enough to accommodate both. Note that in the design of the control architecture the network architect has few chances to solve this kind of problems, which are inherent to the distributed approach. Nevertheless, it is important to evaluate their impact on the performance of the network.

The *Imprecise Information* problem is related to the "reduced" information that can be distributed using OSPF-TE. Due to the "flooding reduction", the information available in the Edge Nodes to take routing decisions will be an approximation of the actual resource status. The impact of this approximation on network performance (e.g., network utilization, call blocking probability) must be evaluated. Note that the network architect has greater control on these aspects, as there are several flooding reduction techniques that can be chosen (and then tuned). A trade-off can be envisaged between the signaling load to distribute the information and the performance in terms of network utilization and call blocking probability.

Some works in the literature describe the problem of Imprecise Information and analyze the network performance. The work in [9] focuses on the trade-off between

[1] Note that the Route Decision Engine (RDE) is a logical process, from the physical standpoint it can either run "on" the LER or it can run on a separate machine connected to the edge node.

the amount of flooding and the network performance in terms of utilization/blocking probability. The aspects of processing cost are not explicitly dealt with. In [10], a similar evaluation on the trade-off is given and some processing cost aspects are also considered ([11] further investigates on the processing cost aspect). The analysis of processing cost in these works is concentrated on the routing protocol aspects and on the calculation of CBR algorithms. The processing cost related to the signaling protocol for path setup is not considered. We believe that this cost cannot be neglected and an important contribution of our work is the combined evaluation of processing cost for routing and signaling protocols given in section 0. Note that the work in [9–11] was based on generic assumptions regarding TE-enhanced routing and signaling protocols, as the protocols were not yet defined. In this paper we could consider the actual behavior of OSPF-TE, RSVP-TE and their interaction and even provide results coming from a testbed implementation. To conclude the survey on relevant literature, a very detailed analysis of processing cost for OSPF-TE has been performed in [13], anyway the focus of that work was on the stability issues of OSPF and the results cannot be applied in our context.

To the best of our knowledge, the issue of Propagation Time, i.e., the impact of the short-term dynamics of OSPF-TE and RSVP-TE has not been thoroughly analyzed before, and this constitutes a second important novelty of our work, reported in Section 4. The goal is to define the operational range where there is no impact of this inconsistency on the network operations.

## 2. Network and traffic models

### 2.1. Network model

Two different network topologies have been considered for our study (Fig. 2). Table 1 reports the number of nodes $N$, the number of unidirectional links $L$, the hop count averaged among all node pairs $\bar{h}$ and the link capacity $C$ (Mb/s). The reason to have two different topologies is that the smaller *7nodes* topology could be implemented both in the simulation study and in a testbed (see II. C below), allowing to compare simulation results with real measurements. The *30nodes* topology (the same used in [12]) was used to have simulation results for a network size comparable with a real life scenario.

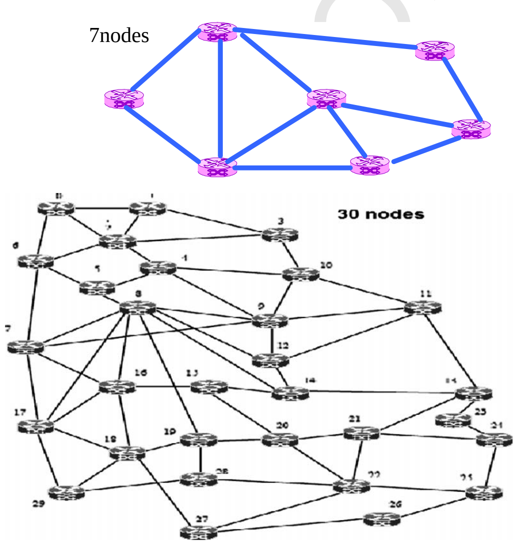


Fig. 2. Network topologies.

Table 1
Network topologies

| Topology | $N$ | $L$ | $\bar{h}$ | $C$ (Mb/s) |
|---|---|---|---|---|
| *7nodes* | 7 | 44 | 1.52 | 100 |
| *30nodes* | 30 | 118 | 3.96 | 635 |

### 2.2. Traffic model and CBR algorithm

In order to model the offered traffic, we considered two different traffic models, a "uniform" model and a "non-uniform" one.

We denote every (source, destination) couple as a Traffic Relation, the arrival rate of Traffic Demands within each Traffic Relation $i$ is denoted as $\lambda_i$ ($s^{-1}$). Under the uniform model, each node generates traffic requests directed to all other nodes of the network, according to a Poisson process, with uniform random selection of destination nodes, therefore $\lambda_i = \lambda\ \forall i$. The total arrival rate of Traffic Demands originating in each node is denoted as $\lambda_{node} = (N-1)\lambda$.

In the case of "non-uniform" model, the composition of two request arrival processes is considered. In addiction to a background uniform traffic, of rate $\lambda_{BG}$ ($s^{-1}$) per each traffic relation, we have a foreground traffic generated by a number of hot-spot pairs, with rate $\lambda_{FG}$ ($s^{-1}$). According to [10], we varied the amount of this foreground traffic in respect of total offered load up to 30%.

We model connection holding times using a negative exponential distribution where $T$ is the mean holding time. The bandwidth of each Traffic Demand is uniformly distributed between 0 and 2b of the capacity $C$ of a link. Therefore, the mean value of a single Traffic Demand is bC. The offered load for each traffic relation $i$ will be $R_o^i = \lambda_i TbC$ (bit/s). In the simulation scenario used in this paper we set $T = 200$ s (a relatively short flow duration in order to have a quite dynamic scenario).

In order to characterize the offered load to the network, we define a "normalized" offered load assuming that all the traffic demands are routed through a shortest path. We denote $h_i$ the shortest path length of the traffic relation $i$, hence the normalized offered load becomes ($N_{TR}$ is the number of Traffic Relations):

$$\rho_{SP} = \sum_{i=1}^{N_{TR}} R_O^i h_i \Bigg/ \sum_{j=1}^{L} C_j.$$

In the "uniform" traffic model the normalized traffic load becomes, as $N_{TR} = N(N-1)$:

$$\rho_{\mathrm{SP}} = \sum_{i=1}^{N_{\mathrm{TR}}} \lambda T b \bar{h} / L = N(N-1)\lambda T b \bar{h} / L.$$

where $\bar{h}$ is the mean distance (in number of hops) between nodes, averaged across all traffic relations (i.e., all pairs of Edge Nodes).

In the non-uniform model we can divide the total offered load in the two background and foreground components:

$$\rho_{\mathrm{SP}} = \sum_{i=1}^{N_{\mathrm{TR}}} \lambda_{\mathrm{BG}} T b \bar{h} \Big/ L + \sum_{i=1}^{N_{\mathrm{HOT-SPOT}}} \lambda_{\mathrm{FG}} T b h_i / L$$

We considered a CBR algorithms. that favors an evenly distribution of the traffic in the network even if it means considering longer path ("least resistance" [14]). The cost $S_i$ of each link $i$ is $S_i = B_T / B_i^A$ where $B^T$ is the maximum link bandwidth in the network, and $B_i^A$ is the available bandwidth in the link $i$. Links with not enough bandwidth are pruned as well.

### 2.3. Simulation environment and testbed

We implemented a "custom" event-based simulator for the OSPF-TE/RSVP-TE environment. The simulator is developed in C++ under the Linux OS, and is available at [15]. The simulator is able to consider two different scenarios. In the first one there is the assumption of "ideal" (e.g., instantaneous) propagation of RSVP-TE and OSPF-TE information (see results in section 3). In the second scenario the real propagation of OSPF-TE and RSVP-TE information (see results in Section 4) is considered in the simulation by taking into account the processing and transmission time of RSVP-TE and OSPF-TE messages.

The testbed is composed of 7 PCs with a Linux Operating System (RedHat 7.1), which are interconnected by point-to-point Ethernet links (100 Mb/s) according to the topology shown in Fig. 2 (*7nodes* topology). Each PC represents a network node with a fully functional implementation of the MPLS-TE control plane (including OSPF-TE and RSVP-TE daemons, Route Decision Engine, Traffic Request Generator). The software packages installed and active on the test bed are: MPLS provided by Sourceforge [16], RSVP-TE daemon from TEQUILA project [17] and OSPF daemon by GNU Zebra software, version 0.92 [18] patched with TE extensions. It implements OSPF v.2 according to [19] with Opaque LSA capabilities [20]. Additional details on the testbed can be found in [21,22].

## 3. "Resource thresholds" mechanisms

The idea of resource threshold mechanisms is to advertise only significant changes of link state information. Therefore, a single advertisement is typically performed after a number of LSP setups and releases, instead of communicating the change of network status for each setup (release) of an LSP. The threshold mechanisms can be classified in static and dynamic ones.

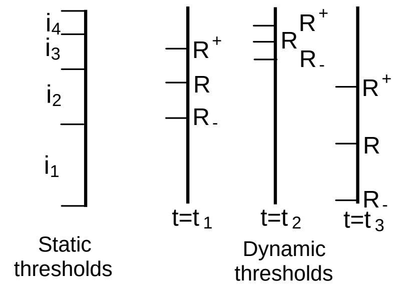


Fig. 3. Static and dynamic thresholds.

Using static thresholds, the link capacity is divided in intervals, limited by upper and lower threshold levels. In order to limit the effect of the inaccuracy introduced by the thresholds, it is sensible to fix just a few threshold levels in the lower part of link bandwidth occupancy and much more levels in the higher part of link bandwidth occupancy (near congestion). There is a large degree of freedom in the choice of the number and of the values of the threshold levels. In order to experiment with the different choices it is reasonable to define families of static threshold mechanisms that can be characterized by few parameters. The two families of threshold mechanisms ("logarithmic" and "3-piece-linear") that we have considered are described in Appendix A. Additional details about the use of threshold values are given in Appendix B.

The dynamic threshold approach assigns an initial threshold level on the empty link and calculates next upper and lower levels as functions of currently advertised reservation amount. Let $C$ be the link capacity and $R$ the currently advertised reserved bandwidth, the upper and lower thresholds are calculated, respectively, as

$$R^+ = R + F \cdot (C - R); \quad R^- = R - F \cdot (C - R).$$

Note that, as desired, the difference between upper and lower thresholds becomes narrower when the available bandwidth decreases. Note also that a larger value of $F (0 < F < 1)$ means more spaced dynamic threshold levels and a coarser vision of network status in the RDEs. Fig. 3 provides a sketch of the two mechanism.

### 3.1. Results and discussion

Let us analyze the trade-off between the amount of flooding and the network performance in terms of connection blocking. We started with a simulation analysis, in the scenario of "ideal" (e.g., instantaneous) propagation of RSVP-TE and OSPF-TE information.

The main results are reported in Figs. 3–6.[2] The leftmost value of the curves represents the network behavior with no threshold mechanisms (perfect vision). When we have a coarser information (smaller number of thresholds in the

[2] *30nodes* topology, $b = 0.05$; for the static thresholds: logarithmic function $\alpha = 10^4$. The figures are obtained under the uniform traffic model, but no difference can be noticed under the non-uniform traffic model.

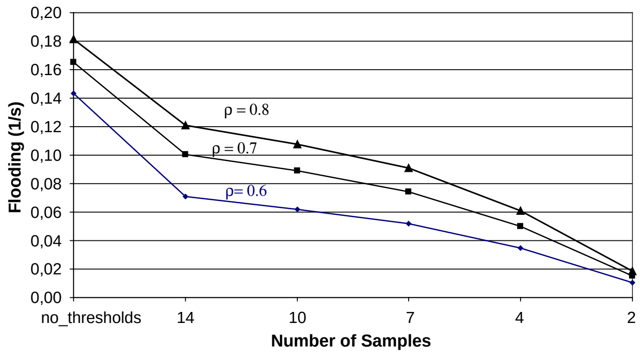


Fig. 4. Static thresholds: flooding.

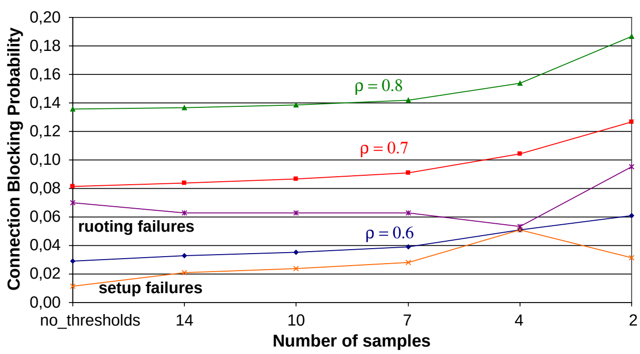


Fig. 5. Static thresholds: connection blocking probability.

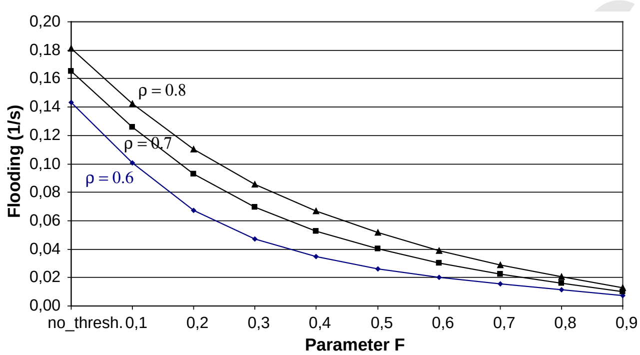


Fig. 6. Dynamic thresholds: flooding.

static scenario, larger $F$ in the dynamic one) we can drastically reduce the amount of flooding (the number of LSU messages originated per link per second is shown). On the other hand, blocking probability starts to increase when the information is too coarse. The analysis is reported for three different values of the "conventional" offered load $\rho_{SP}$ from 0.6 up to 0.8. The typical operating point should be $\rho_{SP} = 0.6$ or less, where the blocking probability is around 2%, while $\rho_{SP} = 0.7$ and $\rho_{SP} = 0.8$ can be already considered overload conditions, considering that the blocking probability is respectively in the order of 8% and 14%. Note that we will not show 95% confidence intervals of simulation results, however results are averages over long runs and such confidence intervals are always smaller than 3% of the value.

Looking at Figs. 3–6, we observe that there is a region (starting from the left) where the blocking probability does not increase significantly while the OSPF-TE message flooding is greatly reduced. This suggests that the optimal working point is where the blocking probability start to increase: in the given scenarios 7 thresholds for the static thresholds or $F = 0.7$ for the dynamic ones.

We define as "merit" factor the ratio between the amount of flooding with thresholds and without thresholds. For offered load 0.6, this factor is 3.1 for static-threshold and 10.6 for the dynamic thresholds, respectively at 7 thresholds and at $F = 0.7$ where the blocking probability is still under control. In Fig. 8 we compare 3-piece linear ($\beta = 0.75$, $\gamma = 0.95$) static thresholds with 14 and 7 levels, logarithmic ($\alpha = 10^4$) static thresholds with 14 and 7 levels and dynamic ($F = 0.7$) thresholds. The 3-piece linear and the logarithmic thresholds have the same merit factor (1.7) for 14 levels while the 3-piece linear yields a larger reduction (merit factor 3.3) than the logarithmic (2.2) for 7 levels. The dynamic thresholds have the larger merit factor (7.3). Note that the connection blocking probability using static mechanisms with 14 thresholds is unchanged with respect to the case without any threshold method, and only minimally increased using static mechanism with 7 thresholds or dynamic mechanisms with $F = 0.7$.

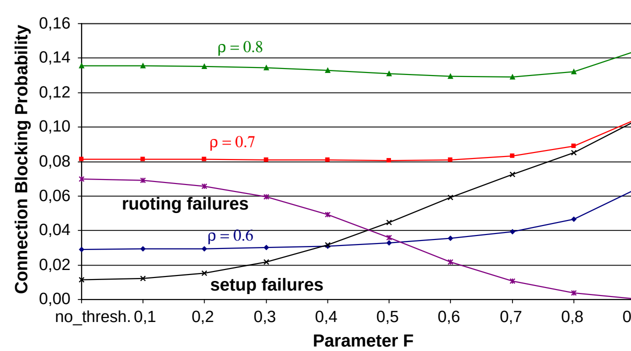


Fig. 7. Dynamic thresholds: connection blocking prob.

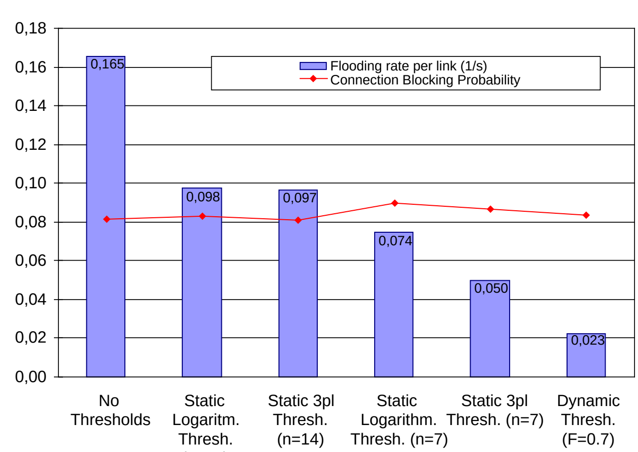


Fig. 8. Static vs. dynamic threshold.

Several simulations have been carried out for the two considered network topologies, under different load scenarios and different traffic models: using the dynamic thresholds with $F = 0.7$, we obtained a merit factor ranging from 8 to 15 without affecting in significant way the network performance (same blocking probability). The results with static thresholds are not equally stable. Comparing the static thresholds with the dynamic ones, we think that it is much easier to reduce OSPF-TE protocol message exchange with the dynamic ones. Moreover, we can say that the dynamic threshold mechanism is simpler to be configured because only the value of $F$ needs to be fixed. This means that one does not have to configure all the threshold values in the routers as in the static thresholds. The use of dynamic thresholds could represent an important improvement with respect to the currently used static thresholds.

In order to validate the simulation analysis, the dynamic threshold mechanism has been implemented in our testbed and various experiments have been carried out in parallel with the simulation environment with the *7nodes* topology (identical to the testbed topology). The main results are reported in Figs. 9 and 10. These two figures represent a comparison between the simulated scenario and the emulated one (testbed). An offered load $\rho_{SP} = 0.7$ is used. As can be seen from the figures we have obtained in the testbed the same behavior as in the simulation.

The final consideration in this section concerns the signaling load due to RSVP-TE. In Figs. 5 and 7 the blocking probability for an offered load $\rho_{SP} = 0.7$ is split into the two components of "routing" failures and "setup" failures. The former ones represent the connections rejected by the CBR algorithm in the ingress Edge Node, the latter ones the connections which are accepted by the CBR algorithm, but then rejected by the RSVP-TE setup procedure due to the local admission control in one of the crossed nodes. According also to [9], we note that the coarser the information, the larger the number of connections that are rejected during the setup phase, originating an unneeded signaling in the network. This suggests that a more detailed analysis should be performed to take into account also the signaling load in the definition of the optimal working point. This analysis will be carried out in Section 0.

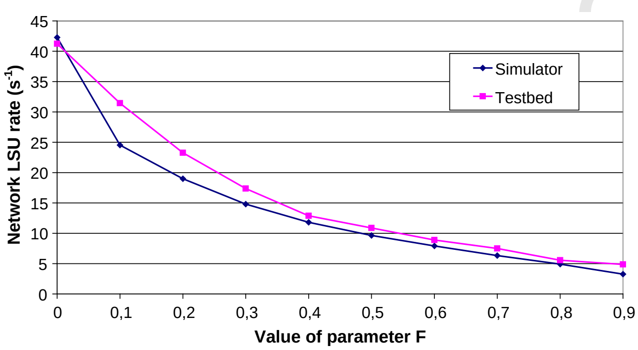


Fig. 9. Flooding reduction comparison.

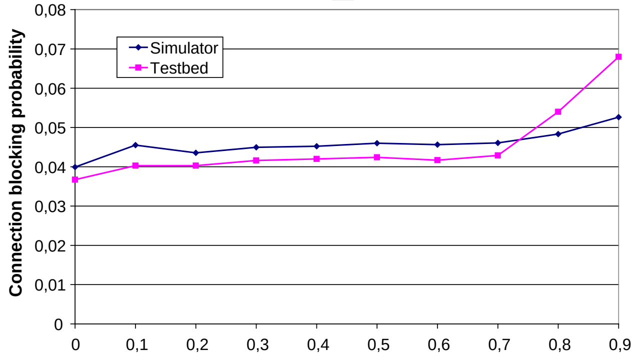


Fig. 10. Blocking probability comparison.

## 4. Impact of message processing/transmission time

As we have observed in the previous section, there is a good agreement between the results coming from the "ideal" simulator and from the testbed. We recall that in the simulations analyzed in the previous section, an ideal behavior for both reservation and routing protocol has been assumed. This means that all processing and propagation times of control plane messages were considered to be null.

The agreement between simulation and testbed results seems to imply that there is no impact of the RSVP-TE and OSPF-TE delays in propagating signaling messages. In this paragraph, we want to verify under which operating conditions this assumption is valid. To analyze the impact on network performance of RSVP-TE and OSPF-TE delays, as a function of the overall connection requests rate, we introduced the processing delays in the simulator and considered the actual behavior of RSVP-TE and OSPF-TE in propagating their messages.

As a preliminary step, we had to figure out the characteristic delays of RSVP-TE and OSPF-TE messages. The value for the processing/propagation time of an OSPF-TE LSU has been taken from [23]. Our simplifying hypothesis is that this delay remains constant from hop to hop and over time. Therefore, the propagation time of an LSU flooding procedure is linear with the number of hops crossed. The value of a single hop processing/propagation time has been set to 34 ms.

RSVP-TE messages (Path, Resv, PathTear, and ResvTear) processing/propagation times were taken from [24]. Again, we made the simplifying assumption that all these times remain constant during the evolution of a simulation, as if they were independent from the number of reservation sessions installed. We considered values of 14, 14, 6, and 20 ms respectively for Path, Resv, PathTear, and ResvTear processing/propagation times.

These delays add inaccuracy in the RDE vision of network status. Each router will have a different vision of the status of network occupation, and this vision in general is not aligned with the real one. Similarly to the effect of a threshold mechanism this will cause the RDE not to always select the optimal paths for LSPs.

By means of simulations, we analyzed the impact of the inaccuracy on network performance. A scenario with no thresholds is analyzed, in order to consider this phenomenon in isolation, the load $\rho_{SP}$ is 0.7. Under the typical scenario assumed so far, with the total requests arrival rate $\lambda_{node}$ of 0.07 $s^{-1}$, we noticed no impact of processing/transmission delays. Therefore, we started to increase the rate of incoming LSP requests in the network. To have a fair comparison, we kept the network load constant, therefore we reduced the connection holding time. We were able to understand when the considered delays start to be influent on network performance. Fig. 11 reports the connection blocking probability and setup failures versus the total arrival rate for the "ideal" system and the system with processing/transmission delays. The blocking probability of the ideal system is obviously not dependent on the arrival rate. It can be seen that RSVP-TE and OSPF-TE messages delays start to influence the connection blocking probability in the system with processing/transmission delays when the request rate is increased by a factor of 20. The degradation of connection blocking is relatively mild, considering that for an increase of request rate by a factor of 100, it goes from 8% to 9.5%. On the other hand, the inaccurate vision of network status causes a rapid growth of setup failures, which are almost null in our initial scenario with $\lambda_{node}$ of 0.07 $s^{-1}$. When $\lambda_{node}$ is 20 times higher ($\sim$1.4 $s^{-1}$), the setup failures are in the order of 3% of offered calls.

In order to understand the previous results, consider that a node is concerned by a connection when it is source, destination or in the path of an LSP. Let $f_{node}$ be the arrival rate of Traffic Demands that "concern" a node: $f_{node} = \lambda_{node} \cdot (\bar{h} + 1)$, where $\bar{h}$ is the mean length of LSPs that are setup (the blocking probability is neglected). $1/f_{node}$ will be the mean inter-arrival time of two connections that concern a node. Approximating $\bar{h}$ with the shortest path, we have that $1/f_{node} = 3.25$ s for $\lambda_{node} = 0.07\ s^{-1}$. According to the assumed values, the characteristic times of RSVP-TE and OSPF-TE procedures are in the order of 50–100 ms, that is 30–60 times smaller than the considered value of $1/f_{node}$. The impact on blocking probability starts when the inter-arrival time of calls concerning a node is in the order of the characteristic times of routing and signaling procedures.

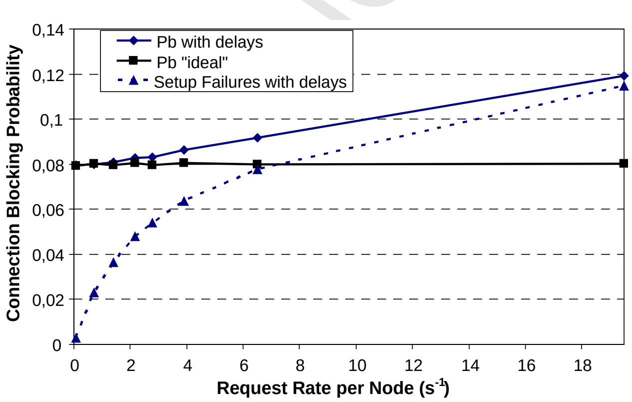


Fig. 11. Network performance vs. total request rate.

## 5. Combined routing/signaling processing cost

In this section, we evaluate the processing cost of the combined OSPF-TE/RSVP-TE architecture. We will show that threshold mechanisms are effective in decreasing the load component due to OSPF-TE, and that the RSVP-TE processing load must be carefully considered as it constitutes the system bottleneck.

The evaluation is based on the definition of a theoretical model of processing costs, combined with the simulator environment. Using our simulator, we can evaluate the number (and the rate) of OSPF-TE flooding procedures that are started by a node. We can also count the number of RSVP-TE messages (Path, Resv, PathTear, and ResvTear). Then we are able to evaluate the total processing cost by multiplying the processing cost of each message $w_{msg}$ for its rate $r_{msg}$.

We will also confirm the theoretic/simulation model results with measurements performed in the tested, related to message rates and to the CPU load.

### 5.1. Message processing cost

Let us consider the different components of processing cost in a TE enabled MPLS network. A component is related to the OSPF-TE messages due to the flooding of state information. Another component is the processing cost of the LSP setup (and release) messages via RSVP-TE protocol. Due to the soft state approach, the processing related to RSVP refresh messages must be also considered.

The processing cost for each message obviously depends on the specific implementation of OSPF-TE and RSVP-TE. In general it can be dependent on the network topology (e.g., on the size of the network) and on the network status (e.g., number of established LSPs). In order to perform our evaluation what we need is actually the relative processing cost of the messages, rather that their absolute values. For this purpose, we take as reference the processing cost of an OSPF Link Status Update (LSU) message containing the first copy of a Link State Advertisement (LSA) received by a router. We assume that one unit of processing cost is needed to check that the LSA is not yet "installed" in the database, to install it and to prepare a copy of it to be sent to all other interfaces but the receiving one. We can now in general define the processing cost of the other messages with reference to this processing unit, using a set of generic parameters as shown in the third column of Table 2. For example $a_1$ is the relative processing cost of a "Copy-LSA" message with respect to the "First-LSA message. The processing cost of RSVP-TE messages is actually split into two factors, $Q$ and $b_i$ $i = 1$–5 for the different RSVP-TE messages. $Q$ represents the relative processing

Table 2
Control plane messages

| Message | Notation | Processing unit | |
|---|---|---|---|
| | | Generic | Assumed |
| "First-LSA" | $w_{\text{firstLSA}}$ | 1 | 1 |
| "Copy-LSA" | $w_{\text{copyLSA}}$ | $a_1$ | 0.5 |
| Path | $w_{\text{Path}}$ | $Q$ | 5 |
| Resv | $w_{\text{Resv}}$ | $Qb_1$ | 6 |
| PathTear | $w_{\text{PathTear}}$ | $Qb_2$ | 3 |
| ResvTear | $w_{\text{ResvTear}}$ | $Qb_3$ | 7 |
| RefreshPath | $w_{\text{RefrPath}}$ | $Qb_4$ | 2.5 |
| RefreshResv | $w_{\text{RefrResv}}$ | $Qb_5$ | 2.5 |

cost of a Path message with respect to a First LSA message: $Q = w_{\text{Path}}/w_{\text{first LSA}}$. The factor $b_i$, for each RSVP-TE message represents its relative processing cost with respect to a Path message.

The exact parameter values are obviously dependent on the specific protocol implementations and also on the network operating point. For the purpose of this paper, we assumed reasonable values starting from the results available in the literature. In particular, [24] have been used to infer the relative processing costs of RSVP-TE messages. [24] has been compared to [23], where the processing cost of OSPF messages is discussed, in order to estimate the value of $Q$. The RSVP-TE processing in typical implementations is dependant on the number $n_{\text{link}}$ of active sessions per link, that can be evaluated as

$$n_{\text{link}} = \lambda_{\text{tot}}(1 - P_{\text{B}})T \cdot \bar{h}/L.$$

In our scenario we have a relatively low number of active sessions per links (in the order of 20), therefore we assumed a processing cost for RSVP-TE close to the minimum values reported in [24].

### 5.2. OSPF-TE and RSVP-TE message rates

According to the OSPF behavior, each flooding procedure results in the exchange of a number of LSU messages that depends on the topology of the network. For a given topology (only point-to-point links are considered) with $N$ nodes and average degree $D$, the number of messages that are generated by each flooding procedure is $N\cdot(D-1)+1$ (see Appendix C). These messages may correspond to two different processing costs in the node. If an (Opaque) LSA is received from a router for the first time, it has to store it and to send it to all the interfaces. When further copies of the same (Opaque) LSA are received, the node simply discards them, resulting in a lower processing cost. In particular in a flooding procedure there will be $N-1$ "first-LSA"s and $N\cdot(D-2)+2$ "copy-LSA"s.

Each successful LSP setup will generate a number $h_{(x)}$ of Path and Resv messages, where $h_{(x)}$ is the number of hops of the LSP $x$. The release of the same LSP will generate a number $h_{(x)}$ of PathTear and ResvTear messages. During the lifetime of the flow the soft state nature of the LSPs will originate $h_{(x)}$ Path and Resv messages with a rate corresponding the refresh rate RR ($s^{-1}$). In the following, we will denote $h_{\text{LSP}}$ the average number of hops of an LSP, leaving out the dependence on the specific LSP $x$. A failed setup of an LSP (see Fig. 13) will generate $h_{(y)}$ Path messages, $r_{(y)}$ Resv messages up to the node where the reservation fails, $r_{(y)}$ ResvError and ResvTear to tear down the part of the LSP attempted to set up, and $h_{(y)} - r_{(y)}$ PathError to advertise source node about the setup failure.

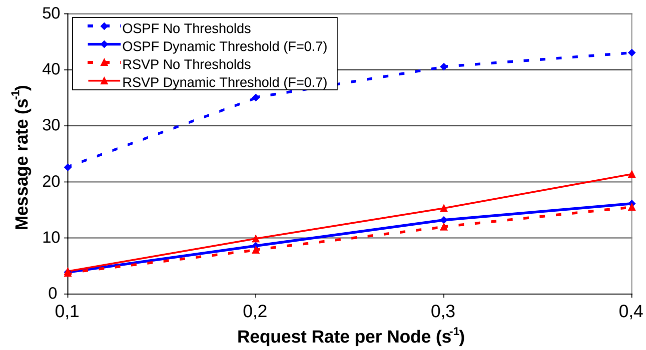


Fig. 12. Number of messages per second.

Utilizing our testbed implementation we measured the exact number of messages exchanged among the nodes. We studied the behavior of the whole architecture in term of packets exchanged by the two protocols, OSPF-TE and RSVP-TE, comparing a scenario without any threshold mechanism with the one utilizing the Dynamic Thresholds with parameter $F$ set to 0.7. Fig. 12 reports the results of these measures representing the message rate for each protocol, in both scenarios, versus the request rate per node $\lambda_{\text{node}}$. We can see that introducing an efficient threshold mechanism, OSPF-TE flooding is enormously reduced, while the number of RSVP messages exchanged are "lightly" increased, by the presence of the Setup Failures.

### 5.3. Definition of processing cost model and results

We started by considering the scenario where no flooding reduction techniques are used: a flooding procedure is executed for each state change. We consider the ideal case, where there is no delay in transmission and processing of OSPF-TE and RSVP-TE messages. Under these assumptions, the Edge Nodes have a perfect vision of the network status and there will be no blocking at the RSVP-TE level. Let $\lambda_{\text{tot}}$ be the total arrival rate of traffic demand to the network, $P_{\text{B}}^{\text{CBR}}$ the blocking rate due to refusals of the CBR algorithm in the originating Edge Node and $n_{\text{LSP}}$ the mean number of active LSP. The processing cost for this scenario is

$$\begin{aligned}P_{\text{tot}} = {} & 2\lambda_{\text{tot}}(1 - P_{\text{B}}^{\text{CBR}})h_{\text{LSP}} \cdot (N-1)w_{\text{firstLSA}} + 2\lambda_{\text{tot}} \\ & \times (1 - P_{\text{B}}^{\text{CBR}})h_{\text{LSP}} \cdot [N(D-2)+2]w_{\text{copyLSA}} + \lambda_{\text{tot}} \\ & \times (1 - P_{\text{B}}^{\text{CBR}})h_{\text{LSP}} \cdot (w_{\text{Path}} + w_{\text{Resv}} + w_{\text{Path-Tear}} + w_{\text{Resv-Tear}}) \\ & + n_{\text{LSP}} \cdot \text{RR} \cdot h_{\text{LSP}} \cdot (w_{\text{Refr-Path}} w_{\text{Refr-Resv}}).\end{aligned}$$

The first two terms represent the processing load for OSPF-TE messages: each call setup that is accepted spans on average $h_{\mathrm{LSP}}$ links and on each links it triggers one flooding procedure for the setup and one for the release; the flooding procedure in turn generates $(N-1)$ "first" LSA messages and $N(D-2)+2$ "copy" LSA messages. The third term represents the RSVP-TE messages that are exchanged during the successful setup and release of the LSP. The fourth term takes into account the RSVP-TE messages related to the maintenance of RSVP soft state: RR is the refresh rate ($\mathrm{s}^{-1}$).

If we consider the scenario with flooding reduction techniques and real processing and transmission times of OSPF-TE and RSVP-TE messages, the setup of an LSP may fail with a probability $P_{\mathrm{B}}^{\mathrm{RSVP}}$. The processing cost can be represented by

$$\begin{aligned}P_{\mathrm{tot}} &= 2\lambda_{\mathrm{tot}}(1-P_{\mathrm{B}}^{\mathrm{CBR}})h_{\mathrm{LSP}}\cdot\frac{1}{M}\cdot(N-1)w_{\mathrm{firstLSA}}+2\lambda_{\mathrm{tot}}\\ &\quad\times(1-P_{\mathrm{B}}^{\mathrm{CBR}})h_{\mathrm{LSP}}\cdot\frac{1}{M}\cdot[N(D-2)+2]w_{\mathrm{copyLSA}}+\lambda_{\mathrm{tot}}\\ &\quad\times(1-P_{\mathrm{B}}^{\mathrm{CBR}})(1-P_{\mathrm{B}}^{\mathrm{RSVP}})h_{\mathrm{LSP}}\cdot(w_{\mathrm{Path}}+w_{\mathrm{Resv}}+w_{\mathrm{PathTear}}\\ &\quad+w_{\mathrm{ResvTear}})+\lambda_{\mathrm{tot}}(1-P_{\mathrm{B}}^{\mathrm{CBR}})(P_{\mathrm{B}}^{\mathrm{RSVP}})h'_{\mathrm{LSP}}\cdot(w_{\mathrm{Path}}\\ &\quad+xw_{\mathrm{Resv}}+xw_{\mathrm{ResvErr}}+xw_{\mathrm{ResvTear}}+(1-x)w_{\mathrm{PathErrr}})\\ &\quad+n_{\mathrm{LSP}}\cdot RR\cdot h_{\mathrm{LSP}}\cdot(w_{\mathrm{Refr-Path}}+w_{\mathrm{Refr-Resv}}).\end{aligned}$$

We notice that the first two terms are reduced by the merit factor $M$ of the flooding reduction technique. The term related to the RSVP load has been split into two terms that take into account the LSPs that are successfully setup and the LSPs that are rejected by RSVP. $h'_{\mathrm{LSP}}$ is the mean length of LSPs that experience a setup failure. The parameter $x$ takes into account the number of hops of the LSP that can be setup before finding a node that rejects the request (see Fig. 13).

Fig. 14 reports the total processing cost versus the parameter $F$ of dynamic thresholds (offered load $\rho_{\mathrm{SP}}=0.7$, $b=0.05$, NSFNET topology,). The total processing cost is split among the routing component, the RSVP-TE (setup and release) and the RSVP-TE refresh. The processing cost of each message is as shown in the last column of Table 2.

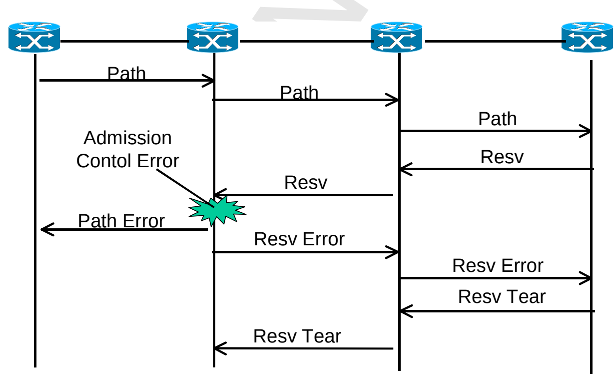


Fig. 13. Failed setup procedure.

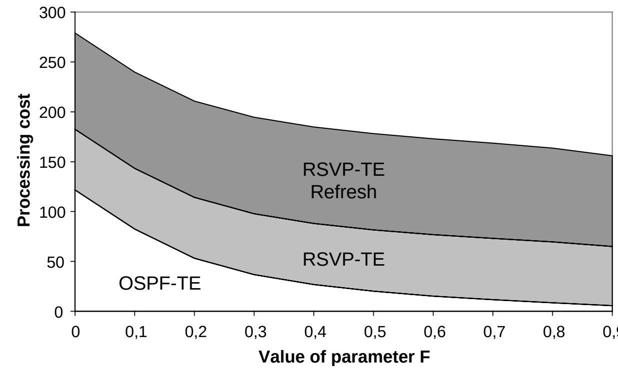


Fig. 14. Total processing cost and its components.

To confirm these theoretic values, we performed some similar measurements in the testbed. We measured processing load in each node in terms of percentage of CPU usage in the two different scenarios: the first one without any threshold mechanism (upper part of Fig. 15) and the second one where the Dynamic Threshold mechanism is implemented with factor $F$ set to 0.7 (bottom part of Fig. 15). The figures show the measured CPU processing loads related to the two protocols (averaged on all the network nodes) versus the requests arrival rate. All measurements were been taken in the testbed during simulations with network load $\rho_{\mathrm{SP}}=0.7$. The reduction of OSPF flooding by means of Dynamic Threshold mechanism significantly reduces the total processing load while the increase of RSVP-TE load due to the presence of setup failures is negligible.

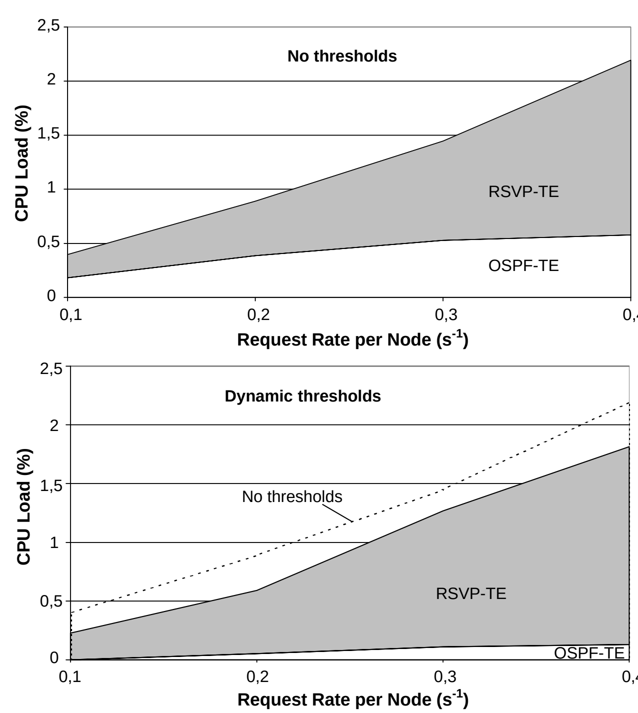


Fig. 15. Processing load.

The first important result is that the use of dynamic thresholds is effective in reducing the overall processing cost: RSVP-TE processing does not increase in a significant way due to setup failures when the network vision become coarser. On the other hand, the overall reduction is less than it was expected considering the large reduction of OSPF-TE flooding. The RSVP-TE cost component, which is basically independent of the flooding reduction technique (see Fig. 14), accounts for the most part of the total processing cost in the region where these flooding reduction techniques are effective. In particular, the RSVP-TE refresh component has a great impact on the total processing (see Fig. 14), suggesting that attention should be paid to reduce it. In particular, aggregate refresh mechanisms, as well as the reduction of refresh rate (we have considered the default refresh rate of $1/30\ s^{-1}$) could be considered. Our analysis suggests that while total OSPF-TE processing cost can be controlled with dynamic threshold mechanisms, the total RSVP-TE processing cost represents a potential bottleneck.

## 6. Conclusions

In this work, we first analyzed the effectiveness of the flooding reduction techniques for OSPF-TE in a MPLS-TE network. The trade-off between the amount of flooding and the connection blocking probability has been analyzed for different mechanisms. The result is the selection of the dynamic threshold mechanism as the most efficient and simplest one.

This analysis has been first performed assuming an instantaneous propagation of the signaling/routing information. Then, the transmission and processing delays of OSPF-TE and RSVP-TE have been considered. This second analysis was able to identify the operating conditions under which these transmission/processing delays do not impact on the network performance.

Finally, the aspects of combined processing cost for routing and signaling have been analyzed. It is shown that the signaling processing cost does not increase significantly when the flooding reduction mechanism are used, therefore the goal to reduce the overall processing cost is met. On the other hand, the analysis showed that the processing cost of signaling represents the largest part of processing cost and may constitute the system bottleneck.

## Appendix A. Families of static threshold mechanisms

Each family can be represented by an increasing function $F(x)$ defined in the interval $0 < x < 1$, with range from 0 to 1 and that is sampled at $M$ equally spaced intervals where $M$ is the number of threshold levels. The threshold values are equal to $C \cdot F(k/M)$ where $1 < k < M - 1$ and $C$ is the link capacity. For example a linear function $F(x) = x$ will define $M$ equally spaced threshold level.

The first family that we have considered is a generalization of the default threshold levels assumed in [25]. According to [25], the threshold levels can be arbitrarily fixed while the default is set to 14 levels. These 14 default levels actually define a 3-piece-linear function (see Fig. 16). We generalize this function, assuming that each linear piece will cover one third of the definition interval and considering two parameters $\beta$ and $\gamma$ such that $F(1/3) = \beta$ and $F(2/3) = \gamma$ $(0 < \beta < \gamma < 1)$. A specific threshold setting for this family is identified by ($M$, $\beta$, and $\gamma$). Therefore, there are two degrees of freedom in adjusting the shape of the function to be sampled. The second family we considered is based on a logarithmic function: $F(x) = \ln(\alpha x)/\ln(\alpha)$, with $\alpha \gg M$. The parameter $\alpha$ defines the shape of the sampled function, with small $\alpha$ (e.g., $\alpha = 10^3$) the function will be more similar to a linear function. For higher $\alpha$ (e.g., $\alpha = 10^6$) there will be less detailed information when the link is not loaded and much more precise information when the link is heavily loaded. Using this "logarithmic" mechanism, a specific choice of thresholds is identified by ($M$, $\alpha$), i.e., we have a single parameter to change.

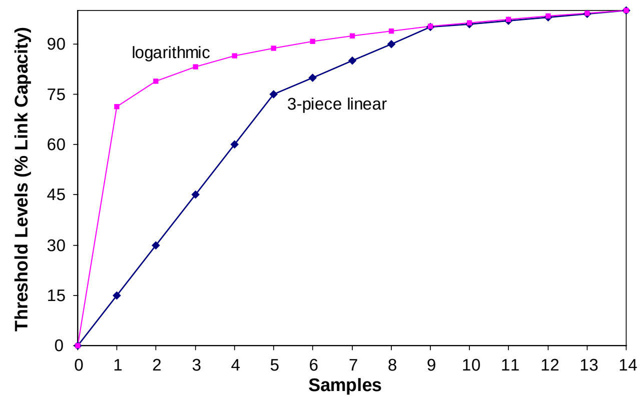


Fig. 16. Static threshold levels.

## Appendix B. Avoiding oscillations with static thresholds

The basic approach is to communicate the middle value of an interval when a threshold is crossed [10]: $L(k) = (F(k/N) + F((k+1)/N))/2$. This may lead to unneeded flooding when the bandwidth oscillates around a threshold level. In [25] it is suggested to use different increase and decrease thresholds to notify the increase and the decrease of bandwidth occupancy, trying to avoid this oscillation. The "increase" threshold $F^{+}(k/N)$ and the "decrease" threshold $F^{-}(k/N)$ can be defined starting from $F(k/N)$ as follows:

$$F^{+}(k/N) = F(k/N),$$

$$F^{-}(k/N) = \frac{F(k/N) + F((k-1)/N)}{2}.$$

On the other hand [25], considers to advertise the actual value instead of a conventional value when a threshold is crossed. When oscillating around a threshold value, for example an increase threshold, a different status will be communicated each time that the threshold is crossed in the increase direction. Therefore, we decided to use the different increase and decrease thresholds and to communicate the middle value as follow:

$$L^{+}(k) = (F^{+}((k+1)/N) + F^{-}(k/N))/2,$$

$$L^{-}(k) = (F^{-}(k/N) + F^{+}((k-1)/N))/2,$$

where $L^{+}(k)$ and $L^{-}(k)$ are the advertised level when the increase threshold $F^{+}(k/N)$ and the decrease threshold $F^{-}(k/N)$ are crossed.

## Appendix C. Number of messages for a flooding procedure

Let $d_i$ be the degree of node $i$, $N$ be the number of nodes, $D$ be the average degree of a node; assume that originating node is $n_1$. The originating node will send $d_1$ copies of the message. Each other node $i$ will send $d_i - 1$ copies (the node will not send the message on the receiving interface). Then:

$$\begin{aligned} \text{NumOfMsg} &= d_1 + \sum_{i=2}^{N}(d_i - 1) = 1 + \sum_{i=1}^{N}(d_i - 1) \\ &= 1 + \sum_{i=1}^{N} d_i - N = 1 + ND - N \\ &= N(D-1) + 1. \end{aligned}$$

## References

[1] E. Rosen A. Viswanathan, R. Callon., Multiprotocol Label Switching Architecture, IETF RFC 3031, January 2001.
[2] X. Xiao, A. Hannan, B. Bailey, L. M. Ni, Traffic Engineering with MPLS in the Internet", IEEE Network, March/April 2000.
[3] D. Awduche et al., Requirements for Traffic Engineering Over MPLS, IETF RFC 2702, September 1999.
[4] D. Katz, D. Yeung, K. Kompella, Traffic Engineering Extensions to OSPF Version 2, IETF RFC 3630, September 2003.
[5] H. Smith, T. Li, IS–IS extensions for Traffic Engineering, IETF <draft-ietf-isis-traffic-04.txt>, IETF RFC 3784, June 2004.
[6] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, RSVP-TE: Extensions to RSVP for LSP Tunnels, IETF RFC 3209, December 2001.
[7] B. Jamoussi et al., Constraint-Based LSP Setup using LDP, IETF RFC 3212, January 2002.
[8] L. Andersson, G. Swallow, The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols, IETF RFC 3468, February 2003.
[9] A. Shainkh, J. Rexford, K.G. Shin, Evaluating the Overheads of Source-Directed Quality-of-Service Routing", International Conference on Network Protocols (ICNP), 1998.
[10] G. Apostolopoulos, R. Guerin, S. Kamat, S.K.Tripathi, Quality of Service Based Routing: A Performance Perspective, SIGCOMM 1998.
[11] G. Apostolopoulos, R. Guerin, S. Kamat, Implementation and Performance Measurements of QoS Routing Extensions to OSPF, Infocom, 1999.
[12] R.R. Irashko, W.D. Grover, M.H. MacGregor, Optimal capacity placement for path restoration in STM or ATM mesh-survivable networks, IEEE/ACT Trans. on Networking, June 1998.
[13] A. Basu, J.G. Riecke, Stability Issues in OSPF Routing", SIGCOMM, 2001.
[14] G. Conte, P. Iovanna, R. Sabella, M. Settembre, L. Valentini, A Traffic Engineering Solution for GMPLS Network: A Hybrid Approach Based on Off-line and On-line Routing Methods, ONDM 2003 Conference, February 4–6, 2003 Budapest, Hungary.
[15] MPLS-TE control plane simulator <http://www.coritel.it/download.html>.
[16] Sourceforge MPLS home page, <http://mpls-linux.sourceforge.net/>.
[17] IBCN testlab Home Page <http://dsmpls.atlantis.rug.ac.be/>.
[18] Zebra Home Page, <http://www.zebra.org/>.
[19] J. Moy, OSPF Version 2, IETF RFC 2328, April 1998.
[20] R. Coltun, The OSPF Opaque LSA option, IETF RFC 2370, July 1998.
[21] A. Bosco, A. Botta, M. Intermite, P. Iovanna, S. Salsano, Distributed Implementation of a Pre-Emption and Re-routing Mechanisms for a Network Control Based on IP/MPLS Paradigm, ONDM 2003 Conference, February 4–6, 2003 Budapest, Hungary.
[22] A. Bosco, A. Botta, G. Conte, P. Iovanna, R. Sabella, S. Salsano, Internet like control for MPLS based traffic engineering: performance evaluation, Performance Evaluation, vol. 59/2–3, February 2005, Elsevier Science, pp 121–136.
[23] A. Shaikh, A. Greenberg, Experience in Black-box OSPF Measurement, ACM SIGCOMM Internet Measurement Workshop (IMW), November 2001.
[24] K. Nemeth, G. Feher, I. Cseleny, Benchmarking of Signaling Based Resource Reservation in the Internet", Networking 2000.
[25] CISCO on line documentation: "MPLS Traffic Engineering".