EVALUATION OF BANDWIDTH-DEPENDENT METRICS FOR TE LINKS IN A GMPLS PATH COMPUTATION SYSTEM.

Gino Carrozzo[1], Stefano Giordano[2], and Giodi Giorgi[1,2]

[1] *Consorzio Pisa Ricerche, Computer Science and Telecommunication Dept., c.so Italia 116, 56125 Pisa – ITALY, Telephone: +39 050 915 811, Fax: +39 050 915 823*

[2] *Information Engineering Dept., University of Pisa, via Caruso, 56122 Pisa – ITALY, Telephone: +39 050 2217 511, Fax: +39 050 2217 522*

Abstract: The GMPLS standardization is paving the way for the implementation of new configurable traffic engineering (TE) policies for transport networks. This paper takes aim at evaluating the effects of using bandwidth-dependent TE metrics in a centralized Path Computation System (PCS), suited for handling the routing requests in an operational transport network with a GMPLS control plane. The results of an intensive testing campaign show an evident improvement in the utilization of network resources when such TE metrics are enabled, whatever survivability requirement is imposed on the LSP (e.g. classical 1+1 protection, pre-planned or On-the-Fly restoration, etc.). Moreover, a simple policy function is suggested as a good trade-off between the achievable performance and the computing load on CPU.

1. INTRODUCTION

The international standardization committees (e.g. ITU-T, OIF and IETF) are all converging in the design of an integrated network with a common Generalized Multi-Protocol Label Switching (GMPLS) control plane. GMPLS will manage all the network data planes [1], providing the required automation in the computation, the setup and the recovery of circuits for next-generation Automatically Switched Transport Network (ASON). The core GMPLS architecture is based on a set of protocols for routing (e.g. G.OSPF-TE and G.ISIS-TE), signalling (e.g. G.RSVP-TE) and link management (e.g. LMP) in order to manage correctly the separation between the Data Plane and the Control Plane.

From a routing perspective, the GMPLS extensions provide new information for circuit computation and they enable configurable traffic engineering (TE) policies and new recovery strategies.

In such a context, this paper takes aim at evaluating the different bandwidth-dependent TE metrics in a centralized GMPLS Path Computation System (PCS). In details, Sec. 2 is focused on the requirements for the PCS in a centralized GMPLS network scenario. In Sec. 3 the implemented bandwidth-dependent metrics are defined, while some results of an intensive testing campaign are shown in Sec. 4, deriving conclusions in Sec. 5.

2. ROUTING REQUIREMENTS FOR A GMPLS PCS

Within the GMPLS standardization, traffic engineering (TE) and survivability are still discussed in terms of high level requirements, though they are fundamental for load balancing and traffic resiliency.

Path computation systems for standard IP networks are generally based on distributed, fast and simple routing algorithms (e.g. of the Shortest Path First class -SPF- [2]), integrated into the routing protocol module. These algorithms operate on a graph derived from the real network. The graph contains only the routing-capable nodes (a.k.a. vertices) and the links between them (a.k.a. edges) with an appropriate link metric. In the GMPLS context, a link connecting two ports of neighbouring nodes may consist of more than one consecutive physical resource (e.g. fibres), possibly crossing routing incapable devices (e.g. regenerators, optical amplifiers, optical mux/demux, etc.). For this reason a set of properties is assigned to each link for routing purposes (e.g. TE metric, available/used bandwidths, resource colours, SRLG list, inherent protections, etc.), transforming the traditional links in traffic engineering links (TE-links). A GMPLS path calculator is expected to return Label Switched Paths (LSPs), i.e. sequences of nodes, TE-links and labels, which try to match some constraints derived from the TE information above. Once an LSP is computed, it describes univocally a unidirectional or bi-directional connection between a source and a destination node. The standard SPF algorithms are not suited for such a computation, as they cumulate only the link metric along the graph. A modified SPF algorithm is needed, called Constraint-based Shortest Path First (CSPF), as routes should be the shortest among those which satisfy the required set of constraints [3]. In the GMPLS architecture no specification is available for the implementation of a PCS module and no preference can be derived by the standards on the choice of a centralized or distributed implementation, in spite of the intrinsic distributed approach of the GMPLS control plane.

The architectural choice we propose in this work is for a PCS module inside a centralized network manager (NM), since this solution promises to be the most effective for the full and flexible management of traffic engineering into

the network, as well as for survivability purposes. This feeling is easily sustainable above all when operating in a multi-area (or multi-domain) scenario. In our implementation the PCS acts as a path computation server for the GMPLS network, receiving from the GMPLS Network Elements (NE) the topology information and the requests for computation across a single- or multi-area/AS. The LSPs computed by PCS (if any) are communicated to the ingress NE and trigger a standard GMPLS signalling session (e.g. via G.RSVP-TE). The communications between the NM and the NEs are carried out by means of COPS protocol with proper extensions [4]. Focusing on LSP requests with survivability requirements [5], we identified three Classes of Recovery (CoR) for the LSP requests (e.g. Gold, Silver, Bronze), respectively related to the request for LSPs with path protection (e.g. SDH/SONET 1+1), Fast Restoration, or On-the-fly restoration [6]. In our PCS different algorithms are used for the different CoRs, ranging from optimal implementations (e.g. in case of Gold CoR) to the sub-optimal ones (e.g. in case of Silver and Bronze CoR). In details, for the Gold CoR we focused on the R.Bhandari's algorithms [8] for computing a pair of least cost maximally disjoint paths; these algorithms represent the optimal counterpart to the sub-optimality required for the Silver CoR, for which we chose the Two Step Approach (TSA) algorithm. TSA is based on a double Dijkstra SPF run and on a simple temporary network transformation for avoiding links/nodes of the worker path. The main advantages of such an algorithm are in the easiness of implementation and in the limited complexity both of the SPF algorithm (e.g. the Dijkstra complexity in our implementation) and of the network transformation. Further details on this issue are in [4] and [6, 7].

3. BANDWIDTH-DEPENDENT TE METRICS

The routing engine inside our PCS module is based on an implementation of the Dijkstra SPF algorithm. Constrained SPF computations are obtained by adding a TE constraints validation step to the well-known Dijkstra flow [2] (e.g. check on the bandwidth availability of the candidate link). Moreover, in order to let the algorithm converge towards an optimal SPF solution (e.g. between those which satisfy the required TE-constraints), we chose to make bandwidth-dependent the link metric minimized during computation, according to the equation [9]:

$$total\_cost = metric_{std} + metric_{TE} + policy_x(bw) \qquad (1)$$

where $metric_{std}$ is the standard OSPF link metric, $metric_{TE}$ is the GMPLS metric for TE purposes, bw is the allocated bandwidth on the TE-link and $policy_x$ ($1 \leq x \leq 2$) is a proper traffic engineering function designed to balance the traffic load (e.g. bandwidth consumption) in the topology.

The policy functions we define for this work are detailed in Eq. 4 and Eq. 5, where bw_{free} is the free bandwidth, while bw_{th} is a threshold with respect to the total bandwidth (bw_{tot}) of the TE-link. The constants K and t are respectively a resource cost per unit and a smoothing factor of the overall TE function.

$$f_1(x) = K \cdot \left(1 + \left(\frac{x}{bw_{tot}}\right)^2\right) \tag{2}$$

$$f_2(x) = K \cdot \left(1 - \frac{x}{1 + bw_{tot}}\right)^{-t} \tag{3}$$

$$policy_1 = K \cdot \frac{1 + bw_{tot}}{1 + bw_{free}} \tag{4}$$

$$policy_2 = \begin{cases} 5 \cdot K & \text{if} \quad bw = 0 \\ f_1(bw) & \text{if} \quad 1 \leq bw \leq \frac{bw_{th}}{2} \\ f_2(bw) - f_2(\frac{bw_{th}}{2}) + f_1(\frac{bw_{th}}{2}) & \text{if} \quad \frac{bw_{th}}{2} \leq bw \leq bw_{th} \\ 10 \cdot K & \text{if} \quad bw \geq bw_{th} \end{cases} \tag{5}$$

In Figure 1-a the two policy functions are plotted, both with $K = 2$ and, only for $policy_2$, $t = 1.5$ and $bw_{th} = 75\%$ of bw_{tot}.

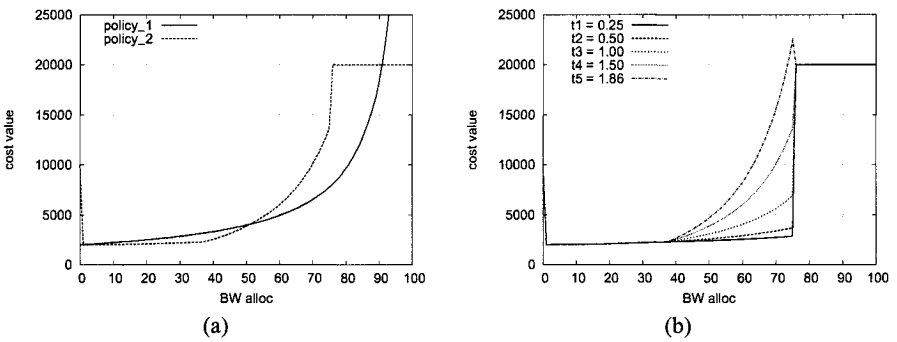

Figure 1. Policy functions (a) and effect of the smoothing factor t in $policy_2$ (b).

These policy functions increase the total cost of the link as allocated bandwidth increases, in order to avoid the overloading of the link and the resulting network congestion.

The $policy_1$ function has been designed in order to discourage the link picking as the available bandwidth decreases on it.

The $policy_2$ function has been conceived with the same aims, but with a further

requirement on the tunability of its behaviour when increasing the cost of the TE-link. Indeed, $policy_2$ increases the link cost less than $policy_1$ till the reference bandwidth of $bw_{th}/2$, encouraging LSP to pick resources on it; when the $bw_{th}/2$ is reached, a "tunable" trend is configurable (ref. Figure 1-b) depending on the value of the smoothing factor, encouraging or discouraging the link picking more or less than $policy_1$. Moreover, an extra cost is assigned to totally free TE-links (i.e. $5 \cdot K$) in order to discourage their utilization in presence of just used links: this allows to fill up the most of the TE-links in a balanced way, bounding the number of used links and the need for their installation (i.e. a kind of feedback to the network planning).

4. PERFORMANCE STUDIES

In this section the performance of the different bandwidth-dependent TE metrics proposed in Sec. 3 is evaluated. The computational environment for all the tests is based on an Intel Celeron 500MHz PC with Linux Slackware 8.1 OS. Measures have been collected on different topologies with increasing meshing degrees (ref. Figure 2 and 3).

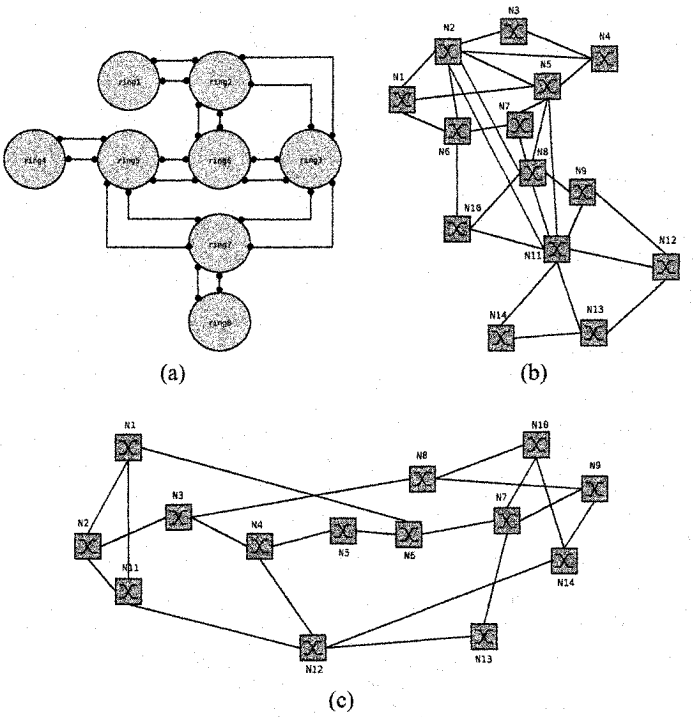

Figure 2. Test topologies: (a) interconnected rings; (b) Italian; (c) NSFNET.

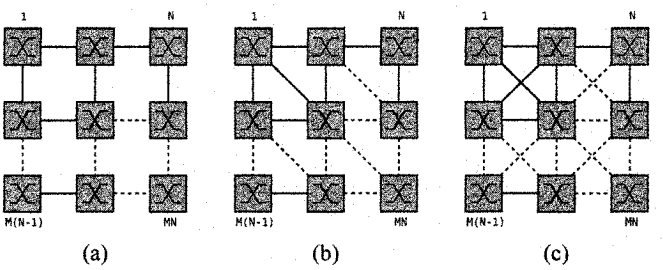

Figure 3. Manhattan topologies: (a) simple; (b) half-meshed; (c)meshed.

All these topologies have been modelled with generic nodes, configurable as SDH 4/4 Cross-Connects. Nodes have been assumed to be fully connectable, i.e. any of their ingress port may be cross-connected to an egress one. Adjacent nodes have been connected by a TE-link with random values for its TE-information (e.g. TE metric, available/used bandwidths, resource colours, SRLG list, etc.). All the TE-links have been assumed bi-directional and each configured with 4 STM-64 ports VC4-multiplexed. A large number of LSP computations (req_path) have been requested on each topology (e.g. up to a connection request from each node towards all the others), trying to establish an overloading condition for the algorithm operations. All the requests have been configured for bi-directional LSPs, with four possible values for the bandwidth:

- VC4 @ 139 Mbit/s ca. (e.g. 95.5% of req_path);
- VC4-4c @ 556 Mbit/s ca. (e.g. 3.0% of req_path);
- VC4-16c @ 2224 Mbit/s ca. (e.g. 1.0% of req_path);
- VC4-64c @ 8896 Mbit/s ca. (e.g. 0.5% of req_path).

For each topology have been observed: the number of computed LSPs ($comp\_path$), the mean TE-link usage percentage ($link\_utilization$) and the number of totally disjoint pairs of LSPs ($disjoint$) in case of LSP requests with recovery requirements.

Table 1. Gain in Link Utilization adopting the new TE policies.

	Bronze CoR	*Silver CoR*	*Gold CoR*
$policy_1$ vs. no_policy	+2.66%	+0.76%	+1.75%
$policy_2$ vs. no_policy	+2.75%	+0.80%	+2.01%

The results in Table 1 show that higher link utilization are achieved when $policy_1$ or $policy_2$ are used. This behaviour is much evident in topologies with lower meshing degrees, which are the common case for currently opera-

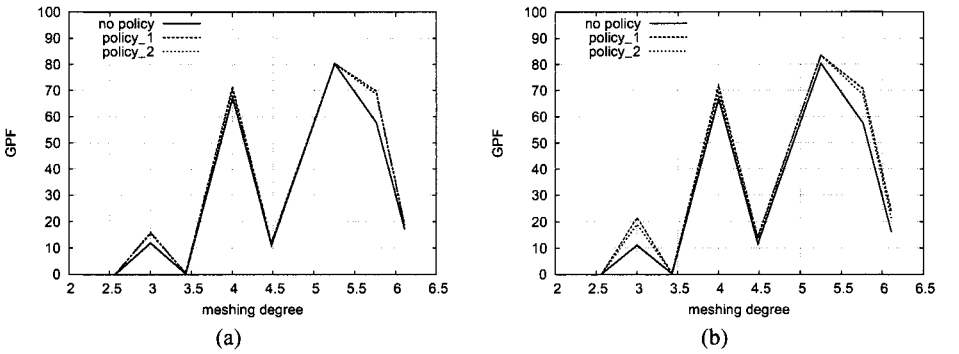

Figure 4. GPF at different TE policies (a) for Silver CoR; (b) for Gold CoR.

Table 2. GPF gain adopting the new TE policies.

	Silver CoR	*Gold CoR*
$policy_1$ vs. no_policy	+2.86%	+5.28%
$policy_2$ vs. no_policy	+2.49%	+4.01%

tive networks. However, the advanteges of using TE policies are related to the optimality of the algorithm adopted for path computation: indeed, when a sub-optimal strategy is used (e.g. TSA in case of Silver CoR), TE enhancements are lower, due to the worst resource picking; this behaviour is not observed in case of optimal algorithms (e.g. Bhandari's for Gold CoR). In order to summarize the performance for the different CoRs, a Global Performance Factor (GPF) has been defined as:

$$GPF = \frac{disjoint}{req\_path} \cdot \frac{comp\_path}{req\_path} \tag{6}$$

in which the first term is related to the algorithm's effectiveness in creating maximally disjoint paths, while the latter term represents the algorithm's effectiveness in computing valid paths, according to the resource availability on TE links. In Figure 4 the GPF is drawn at the different meshing degrees, showing a mean higher performance in case of utilization of the TE policies, both for Gold and for Silver CoRs (ref. Table 2 for numerical details). However, the advantages of defining complex TE policies (ref. Eq. 5) do not pay for the introduced load on CPU, as demonstrated by the improvement in link utilization of less than 0.25% and by the worsening of the GPF value obtained when using $policy_2$ w.r.t. $policy_1$. Therefore, a simple policy function such

as the one described in Eq. 4 is suitable and effective for a good tradeoff between the achievable performance (i.e. in terms of load balancing and blocking probability of the restoration) and the computational load on the CPU.

5. CONCLUDING REMARKS

In this paper the effects of using different bandwidth-dependent TE metrics in a centralized GMPLS PCS have been evaluated. The results shown above highlight how an improvement in the utilization of the network resources is always achievable when applying TE policies in path computation, whatever CoR is required. However, from the same tests we derive as a general result for different topologies at different meshing degrees, that the definition of fine-grain TE policies do not pay for the achieved performace enhancment, degraded also by the additional CPU load. Further investigations are ongoing to study the performance of a fine TE tuning in a fixed GMPLS topology with real expected traffic loads.

ACKNOWLEDGMENTS

This work has been supported in part by the Italian Ministry of Education, University and Research through the project TANGO (MIUR Protocol No. RBNE01BNL5).

REFERENCES

[1] E. Mannie (Editor) et al., *Generalized Multi-Protocol Label Switching (GMPLS) Architecture*, draft-ietf-ccamp-gmpls-architecture-07.txt, Internet Draft, Work in progress, May 2003.

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows - Theory, Algorithms and Applications*, Prentice Hall, 1993.

[3] J. Strand et al., *Issues for Routing in the Optical Layer*, IEEE Communications Magazine, pages 81-87, Feb. 2001.

[4] *Traffic models and Algorithms for Next Generation IP networks Optimization (TANGO) Project*, http://tango.isti.cnr.it/.

[5] P. Lang (Editor) et al., *Generalized MPLS Recovery Functional Specification*, draft-ietf-ccamp-gmpls-recovery-functional-00.txt, Internet Draft, Work in progress, Jan. 2003.

[6] G. Carrozzo, *Algorithms and Engines for On line Routing in Generalized MPLS Networks*, PhD dissertation, Dept. Information Eng., University of Pisa, ITALY, 2004.

[7] G. Carrozzo et al., *A Pre-planned Local Repair Restoration Strategy for Failure Handling in Optical Transport Networks*", Photonic Network Communication, vol. 4 (no. 3-4), pages 345-355, Jul./Dec. 2002.

[8] R. Bhandari, *Survivable Networks - Algorithms for Diverse Routing*, Kluwer Academic Publisher, 1999

[9] B. Awerbuch et al., *Throughput-Competitive On-Line Routing*, IEEE Symposium on Foundations of Computer Science(FOCS), 1993, pp. 32-40.