

DiffServ Aware MPLS Traffic Engineering for ISP Networks: State of the Art and New Trends

Armand Toguyéni and Ouajdi Korbaa

Abstract—In the recent ten years, with the development of new applications through Internet such as multimedia or networked control applications, users need more and more quality of service (QoS). However, the requested QoS is not the same depending on the application. Most of the new models to manage internet traffic are based on specific QoS criteria which should be optimized. This paper presents main multiprotocol label switching (MPLS) approaches such as MPLS adaptive traffic engineering (MATE), load distribution in MPLS (LDM) and load balancing over widest disjoint paths (LBWDP) that are new models for traffic engineering. It also introduces periodic multi-step (PEMS) algorithm that adapts the offered quality depending on the class of the routed traffic.

Keywords—differentiated service, multipath routing, QoS routing, quality of service, traffic engineering.

1. Introduction

The growth of multimedia applications over wide area networks has increased research interest in quality of service (QoS). The communication delay and synchronization needed for voice, data and images are major concerns. Internet telephony (voice over IP) and other multimedia applications such as video conferencing, video on demand and media streaming require service guarantees and have strict timing requirements. The size and quality of display devices, and resources such as central processing unit, battery power and bandwidth (BW) are always limited.

Quality of service can be parameterized as throughput, delay, delay variation (jitter), loss and error rates, security guarantees and so on, that are acceptable in an application. As such, QoS depends on characteristics of applications. For instance, the variation in delay, the difference between the largest and the smallest delay, is called delay jitter and jitter is an important quality for Internet protocol (IP) telephony, which can tolerate a certain percentage of packet loss without any degradation of quality. For data transfer, loss is a crucial QoS parameter.

Internet is also more frequently used to control real time industrial system such as power plants or car production chains. All these applications should guarantee some features of the network with regard to the quality of transmission flows but with different criteria.

Quality of service control requires an understanding of the quantitative parameters at the application, system and network layers. This paper concerns the way we can achieve

QoS at network layer and more precisely in an Internet service provider (ISP) network. The ISP networks are essential for QoS because they assume the transit of flows at the network core. The problem is that very often the ISP must increase the capacity of its network resources because of the increase of users' flows. The ISP also notices that some parts of their networks are often congested while other parts are less. The idea developed here is to propose load balancing approaches to allow better performances of ISP networks.

The rest of the paper is organized as follows. Section 2 presents a state of the art of QoS in Internet. Section 3 concerns more particularly traffic engineering (TE) and illustrates this technique to improve QoS by examples of the models based on multiprotocol label switching (MPLS). Section 4 shows periodic multi-step algorithm (PEMS) – a new model to ingrate differentiated service (DiffServ) and traffic engineering. Finally, Section 5 presents conclusions.

2. Quality of Service in Internet: State of the Art

The convergence of networks and telecommunications networks has resulted in new requirements in terms of quality of service for networks. In this new framework, services based on networks are diverse and therefore have different requirements. One can easily understand that the requirements are different between a telerobotics application and an application on video on demand. As an example, let us consider the case where a cardiologist needs to control a remote robot to perform a heart surgery. We understand that in this context, the network must guarantee a continuous control flow which meets the requirements of real time.

In recent years, a network such as asynchronous transfer mode (ATM) has been designed for this purpose [1] but it was not imposed as architecture to replace transmission control protocol/Internet protocol (TCP/IP) model. ATM is often limited to function as a lower layer of the Internet. As ATM is not used as an end-to-end protocol, the Internet still works in best effort manner. This model does not meet the requirements of service quality for all applications. Indeed, the main difficulty in achieving this objective is the bottleneck limiting the services provided by Internet routers. A main reason lies in the functioning of interior gateway protocols of Internet. These protocols tend to route

packets according to one privileged path regardless the load. As a consequence, it is the unbalanced distribution of the load on the networks of Internet service providers. They try to solve problems of congestion through the regular adding of new resources to increase the bandwidth offered by the most congested roads. But this is a short-term solution that is quickly inadequate and costly.

In recent years several studies were interested in providing more robust answers to this problem. We can classify them into two main categories. The first category is the work aiming to accommodate the phenomena of congestion. The second category concerns efforts to develop models to better distribute the flow in a network. This is called traffic engineering.

The general idea of work to accommodate the phenomenon of congestion is to define classes of traffic, so that each router handles a flow of each class according to their respective priority rules. So it breaks with the usual first in first out technique, and a flow of a priority class may be sent before the other, even if received last. The implementation of this approach also relies on the use of appropriate scheduling techniques implementing the priority rules of each class. Among the principal techniques for scheduling, we can cite the generalized processor sharing (GPS), which is a theoretical ideal technique but impossible to implement in a network based on packet switching, because the emission of packets is not preemptive. Other sequencing techniques have been proposed to achieve results similar to those of GPS: weighted fair queue (WFQ) or W2FQ [2]. In this context, two main models have been tested by the Internet Engineering Task Force (IETF): the Intserv [3] and Diffserv model [4].

The Intserv is based on the definition of micro flow that crosses routers in a domain. The maintenance of a path requires the regular exchange of messages between pairs of routers to indicate that the path is still in service. Maintaining a soft state by micro flow in each crossed router, as well as the scheduling of these flows, creates a complexity that makes Intserv not scalable.

Diffserv (DS) is based on the aggregation of flows into a reduced number of classes divided into three categories of services: expedited forwarding (EF), assured forwarding (AF) and best effort (BE) service. The EF service meets the requirements of reliable and real-time traffics (low delay and low jitter). The AF service provides the bandwidth required for applications such as video over IP. The limited number of flow, the simplicity of scheduling algorithms and the limitation of the most complex mechanisms at ingress routers make Diffserv a scalable model.

In terms of traffic engineering, there are two scopes: one corresponding to pure IP networks [5] and another based on the use of multiprotocol label switching. The MPLS is suitable in the networks of Internet service providers because it allows establishing paths in architecture that basically operate in disconnected mode. In this context, the works that are generally developed propose models to select a set of candidate paths (CPs) that meet specific criteria

of QoS. The combination of criteria is generally a NP-complete problem. This leads to propose heuristics such as MPLS adaptive traffic engineering (MATE) or load distribution in MPLS (LDM) that will we describe in Section 3.

To reconcile the advantages of Diffserv and TE, one looks now to their integration: it is the DS-TE model. The objective of DS-TE is to ensure an end-to-end QoS meeting the requirements of a given flow. The approach does not consist to define paths with the same quality as in the case of conventional traffic engineering. It has also different QoS routing that proceeds hop by hop. The idea of DS-TE is to define traffic classes of which are allocated priorities to be assigned to a layered service providers (LSPs). These traffic classes can share same links in a network using different modes of bandwidth management such as max allocation with reservation bandwidth constraints [6], [7] or "Russian doll" management [8]. This requires the development of techniques allowing a preemption flows belonging to a higher-priority class to assure LSP meets their requirements instead of a stream belonging to a lower-priority class [9].

The reader will find in [10] a more complete survey of the state of the art, in the integration of traffic engineering and Diffserv for DS-TE.

3. Illustration of Traffic Engineering in a MPLS Network

Several models are proposed in the literature to perform traffic engineering based on MPLS. In this section, we consider particularly three models: MATE, LDM and LBWDP (load balancing over widest disjoint paths). These models will be compared with traffic bifurcation (TB) that is a mathematical formulation of route optimization problem [10], [11]. It is a theoretical model that cannot be implemented online because it requires knowing a priori all flows that must be routed. So it gives a reference to compare the different propositions.

3.1. MPLS Adaptive Traffic Engineering

The main goal of MATE [12] is to avoid network congestion by adaptively balancing the load among multiple paths based on measurement and analysis of path congestion. This approach uses a constant monitoring of the links using probe packets to evaluate link properties such as packet delay and packet loss. Using these statistics the MATE algorithm is able to optimize packets repartition among multiple paths to avoid link congestion.

Formally a MATE network is modeled by a set L of unidirectional links. It is shared by a set S of ingress-egress (IE) node pairs, indexed $1, 2, 3, \dots, S$. Each of these IE pairs s has a set $P_s \subseteq 2^L$ of LSPs available to it. The P_s are

disjoint sets. An IE pair s has a total input traffic of rate r_s and routes x_{sp} amount of it on LSP $p \in P_s$ such that

$$\sum_{p \in P_s} x_{sp} = r_s, \quad \text{for all } s. \quad (1)$$

Let $x_s = (x_{sp}, p \in P_s)$ be the rate vector of s , and $x = (x_{sp}, p \in P_s, s \in S)$ the vector of all rates. The flow on a link $l \in L$ has a rate that is the sum of source rates on all LSPs that traverse link l :

$$x^l = \sum_{s \in S} \sum_{l \in P, p \in P_s} x_{sp}. \quad (2)$$

Associated with each link l is a cost $C_l(x^l)$ as a function of the link flow x^l . We assume that, for all l , $C^l(\cdot)$ is convex. Its objective is like this:

$$\min_x C(x) = \sum_l C_l(x^l) \quad (3)$$

$$\text{subject to } \sum_{p \in P_s} x_{sp} = r_s, \quad \text{for all } s \in S \quad (4)$$

$$x_{sp} \geq 0, \quad \text{for all } p \in P_s, s \in S. \quad (5)$$

A vector x is called a feasible rate if it satisfies Eqs. (4) and (5). A feasible rate x is called optimal if it is a minimum of the problem Eqs. (3)–(5). A standard technique to solve the constrained optimization problem, Eqs. (3)–(5) is the gradient projection algorithm. In such an algorithm routing is iteratively adjusted in opposite direction of the gradient and projected onto the feasible space defined by Eqs. (4) and (5). The complexity of this algorithm is $O(n^2)$. The designers of MATE have proved in [12] that it converges to an optimal routing when specific conditions are verified (see Theorem 2, page 4 in [12]).

3.2. Load Distribution in MPLS Network

Depending on the dynamic network status, LDM [13] selects a subset of the LSPs (candidate path set) for an ingress-egress pair, and distributes traffic load among those LSPs. Let L_{ij} denotes the set of all LSPs set up between an ingress node i and an egress node j , and let A_{ij} the corresponding candidate LSPs, then $A_{ij} \subseteq L_{ij}$. Initially, A_{ij} is set as follows:

$$A_{ij} = \{\text{LSPs from } i \text{ to } j \text{ with the smallest hop count and with the utilization rate lower than } \eta_0\}.$$

The utilization rate of an LSP, $u(l)$, is defined as the maximum of the utilization value of the links along the LSP l , and let $h(l)$ denotes the hop count of LSP l . The utilization rate of a candidate paths set A_{ij} is defined as following:

$$U(A_{ij}) = \min[u(l), \forall l \in A_{ij}]. \quad (6)$$

The LDM decides whether to expand the candidate LSP set based on the congestion level of candidate paths set. If $U(A_{ij}) \geq \rho$, then LDM further expands A_{ij} . The expansion of A_{ij} continues, considering LSPs in L_{ij} in the increasing order of hop count until $U(A_{ij}) < \rho$ or there is no LSP left in L_{ij} for further consideration.

Generally, an LSP $l \in L_{ij}$ with $h(l) = (h(\text{shortest LSP}) + m)$ should satisfy the following two conditions to be eligible for A_{ij} :

1. $u(l) < \max[u(k), \forall k \in A_{ij}]$,
2. $u(l) < \eta_m$, where $\eta_m < \eta_n$ for $m > n$.

The first condition means LDM utilizes the LSPs with more extra hops if they have lower utilization than the LSP that has the highest utilization among the LSPs in the current A_{ij} .

The second condition implies links with an utilization rate higher than η_m can only be used by the LSPs with less than m extra hops.

The candidate path set could either be pre-computed when there are some significant changes in the dynamic network status or be computed on demand for a new arriving user flow request. This is done in a $O(n^2)$ time in the worst case, and n refers here to the number of available paths between the ingress-egress pair of routers. For each incoming traffic flow, LDM randomly selects an LSP from the candidate LSP set according to a probability distribution function. This probability is inversely proportional to number of hops in the path. At the opposite, it is proportional to the utilization rate of the LSP. The complexity of the LDM splitting procedure is $O(n)$. Here n refers to the number of candidate paths selected at the end of the previous step and belonging to the set A_{ij} .

Let us notice here that instability can affect LDM because of oscillations due to candidate path selection. This oscillation problem can be solved using two thresholds. In [14] the authors propose a new version of LDM that corrects the instability of the original model. One of the disadvantages of LDM is to ignore the residual capacity of a path before assigning it a new traffic.

3.3. Load Balancing over Widest Disjoints Paths Algorithm

This model uses the selection path algorithm proposed by widest disjoint paths (WDP) algorithm [15] and a splitting algorithm called prediction of effective repartitioning (PER).

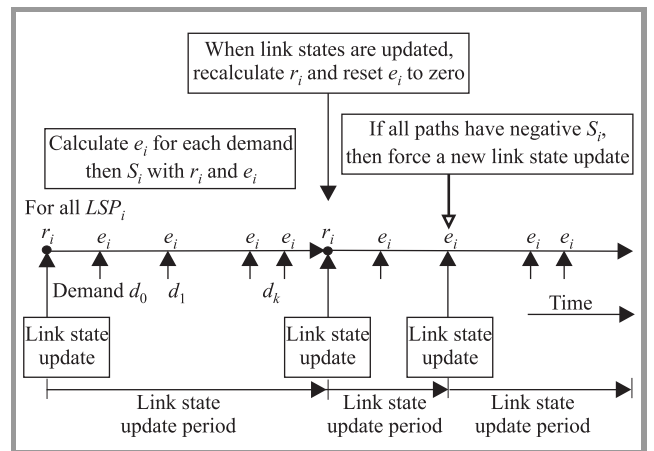


Fig. 1. Illustration of the principle of PER.

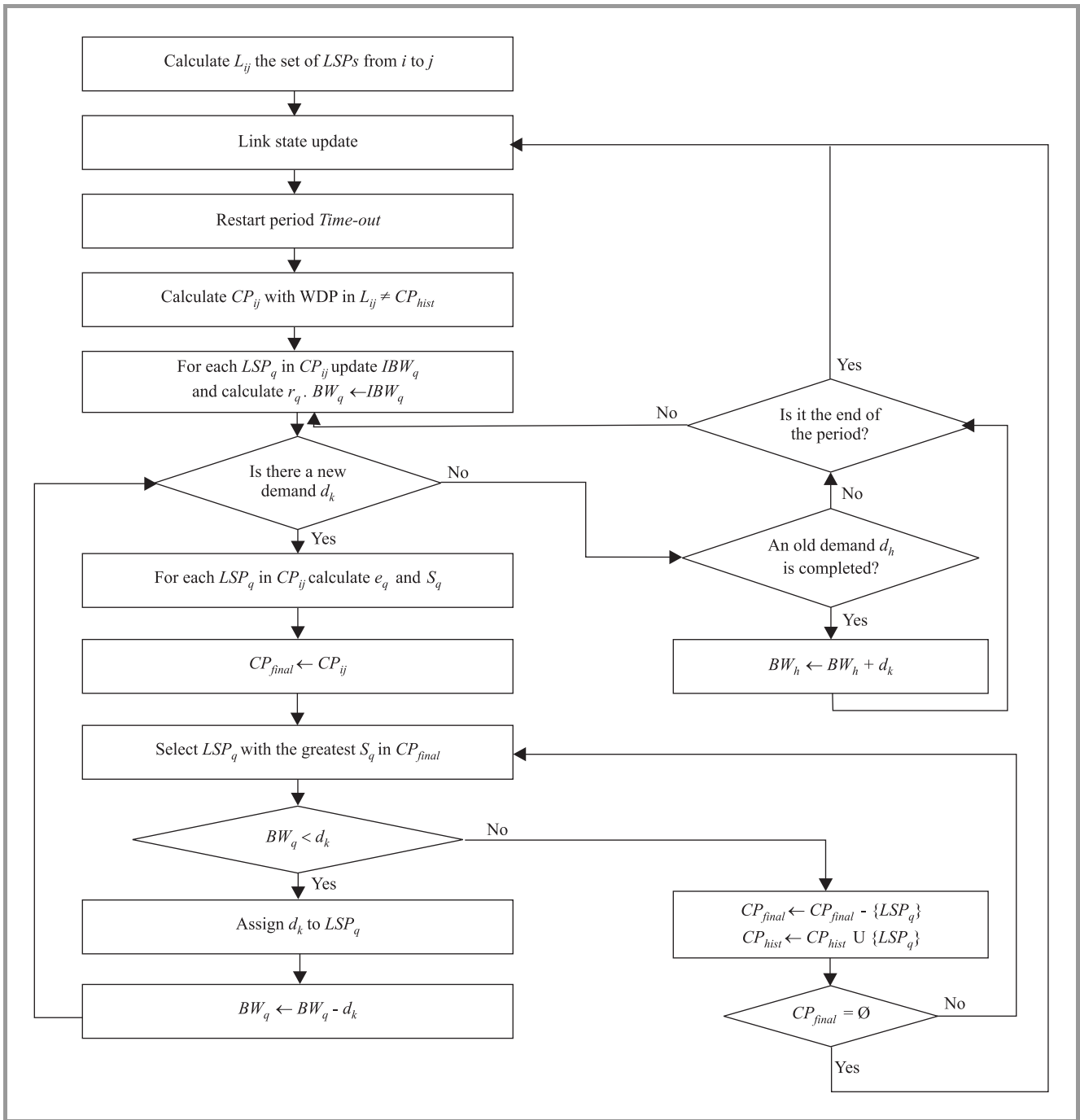


Fig. 2. Flowchart of PER algorithm.

tion (PER) [16]. PER is an improvement of LDM splitting algorithm. PER is designed to take into account the capacity of the selected path when it assigns a new traffic. The basic idea is that each ingress node takes into account its previous assignments of traffics to the different paths it manages. At each time, it must know the residual capacity of each of its paths to reach a given destination. However, local management made by each ingress node is necessarily partial. Indeed, an ingress node A is in competition with other nodes that can handle paths sharing links with the paths from A . Therefore the vision of the node A must

reflect the actual state of the paths it manages. To do this, the idea developed by PER is to establish a periodic routing plan. Before beginning a given period, the node uses link state update to obtain the residual bandwidth of each path it manages. At the beginning of the period it has a perfect vision of the state of those paths. Then, during the period the state of its paths is updated in terms of assignments done. Knowing that there will be drifts, at the end of each period it performs a new update to prepare the routing plan of the next period. Figure 1 gives an illustration of the principle of PER.

During each period, the routing for a given destination is based on the calculation of the theoretical distribution of each path managed by the ingress router for a given destination. This calculation is based on Eq. (7). Let $A_{ij} = \{l_1, l_2, \dots, l_n\}$ be the set of candidate paths from ingress node I to egress node J . It takes into account criteria like the hop count $h(k)$ of each path and the residual bandwidth capacity $b(k)$, where k is the index of a LSP in A_{ij} :

$$r_k = p_0 \frac{H}{h(k)} + p_1 \frac{b(k)}{B} \quad \text{with } p_0 + p_1 = 1, \quad (7)$$

where: H is the constant to make the sum of the probabilities that are inversely proportional to the hop count of an LSP:

$$H = \frac{1}{\sum_{k=1}^n \frac{1}{h(k)}}, \quad (8)$$

coefficient B is the sum of residual bandwidth of all the LSPs in A_{ij} :

$$B = \sum_{k=1}^n b(k), \quad (9)$$

p_0, p_1 are parameters of the model fixed by the network manager depending on its requirements.

During a period after each new request assignment, the ingress router computes the effective repartition rate e_k of each path using Eq. (10). This rate is calculated simply by considering the amount of traffic requests assigned to a path compared with the sum of all the requests routed to a destination by all paths in A_{ij} during the period:

$$e_k = \frac{\sum_{p=1}^{m(k)} d_p^k}{\sum_{q=1}^m d_q}, \quad \text{where } \sum_{k=1}^n \sum_{p=1}^{m(k)} d_p^k = \sum_{q=1}^m d_q, \quad (10)$$

where: $m(k)$ is the number of flow traffics assigned to LSP number k between the n LSPs of set A_{ij} ,

m is the total number of flow traffics the considered ingress router has to route to router J : $m = \sum_{i=1}^k m(k)$,

d_p^k is the traffic amount of the p demand assigned to LSP number k ,

d_q is the q traffic flow routed by the ingress node with a LSP of the set A_{ij} .

For each incoming flow, the ingress router calculates a relative distribution rate S_k for each k :

$$S_k = \frac{r_k - e_k}{r_k}. \quad (11)$$

This relative distribution rate enables selecting effectively the LSP which is assigned the flow. This LSP must verify the following 3 conditions:

1. S_k must be positive. This means that the effective distribution rate is below its theoretical rate. Therefore it is possible to increase its load.
2. The requested bandwidth $BW(d_k)$ must be less than $b(k)$ the residual bandwidth of the LSP.
3. There is no LSP verifying the conditions 1 and 2 with a greater S_k .

If there is no LSP to verify the conditions for delivering the demand then the router must force the update of data of path before the end of the period. This forced update enables to build a new set of candidate paths and consequently to establish a new routing plan based on this set. In case of failure, the demand must be distributed over several LSPs. Figure 2 summarizes how PER works.

3.4. Evaluation of Different TE Models Based on MPLS

In literature each of the presented models is said by the authors as being the best. Also to get an idea of the quality of the different models presented in this section, we have evaluated them by simulations. All simulations have been performed on the same architecture. For the sake of simplicity, let us consider the architecture given by Fig. 3 to compare their relative performances. The simulations have been conducted with the simulator NS2.

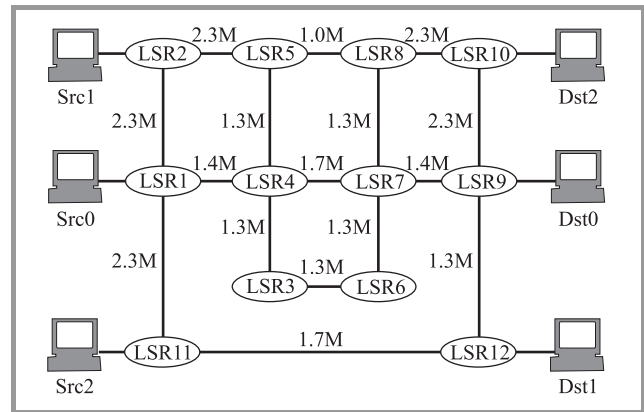


Fig. 3. Simulation topology. Explanations: LSR – label switch router, Src – source router, Dst – destination router.

For each model for TE we have used the same profile of traffic. This profile has the following characteristics:

- the volume of each individual demand is 300 kbit/s;
- the source and destination pairs are Src0-Dst0, Src1-Dst1 and Src2-Dst2, selected randomly;
- one flow generated in certain time is stopped in a random time;

- we adapt a time based triggering as a triggering policy and update link state every 3 s;
- the delay of each simulation is 150 s.

Figure 4 presents the results obtained by the different models. In order to have a reference one has represented on the same graph the curve corresponding to the theoretical model TB. Note that the curve of TB was not obtained by simulation in NS2 but by calculation in Matlab. The goal is to have a reference to compare with the proposed heuristic models. As we can see, MATE and LBWDP have comparable results, close to TB. At the opposite, LDM presents a utilization rate that may exceed 100%. This reflects the fact that LDM does not verify that the selected LSP owns a capacity of residual bandwidth enough to support the demand.

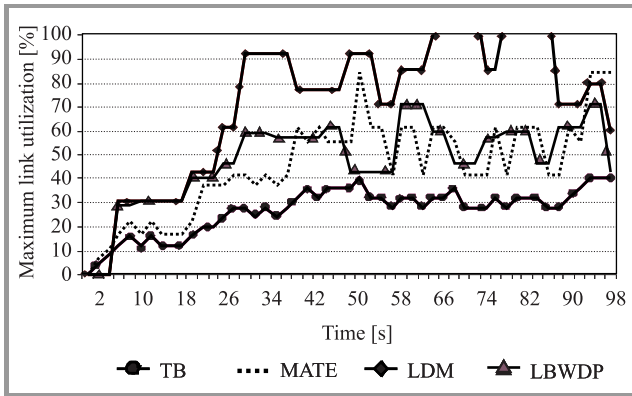


Fig. 4. Simulations results.

Our simulation results showed that LBWDP is one of the best algorithms for traffic engineering because with a priori decision its balance of flows is comparable with the results given by TB.

4. Periodic Multi-Step Routing Algorithm for DS-TE

In this section, we propose new DS-TE model for the intra-domain network, called PEMS [16], to give the differentiated services for the three classes defined in DiffServ. The PEMS is composed of three phases. The preprocessing phase is achieved off-line and extracts good paths of all possible paths which can include every link at least once within them for each source-destination pairs using only topology information. These paths are kept until the topology is changed.

When a traffic demand arrives, it uses PER algorithm to select one LSP to carry current flow. Many QoS metrics such as hop-count, available bandwidth and delay constraints are considered before the path selection assignment. In PEMS, hop-count and disjointness are used in the pre-processing phase together with available bandwidth and measured delay in the cost function to establish splitting ratios. PEMS basically aims to minimize the maximum link utilization

like LBWDP algorithm and additionally to give different service quality to each class, especially to guarantee the low delay to EF class. But it has two differences in that PEMS uses measured delay $de(i)$ instead of hop-count and that it adapts different p_0 , p_1 values according to the class, in contrast to LBWDP, which uses the same parameter values regardless of class. To establish the routing plan for each period, PEMS uses Eq. (12) that is an adaptation of Eq. (7) used by LBWDP:

$$r_i = p_0 \frac{D}{de(i)} + p_1 \frac{b(i)}{B} \quad \text{with} \quad p_0 + p_1 = 1. \quad (12)$$

In Eq. (12), D is a constant to make the sum of the probabilities that are inversely proportional to delay of an LSP_i , $de(i)$. Formally D is defined as it follows:

$$D = \frac{1}{\sum_{i=1}^k \frac{1}{de(i)}}. \quad (13)$$

In this model, bandwidth is associated with delay for differentiating the traffic at the flow level. Bandwidth has a bigger weight p_1 for AF class, while delay has a bigger weight p_0 for EF class. Adaptation of selective parameters is used to give different weight according to the metric important of each class. PEMS puts the weight parameters, p_0 and p_1 , of each class as follows.

In Table 1, for EF class, p_0 is bigger than p_1 in order to give preference to LSP in BP_{ij} that owns the best delay than residual bandwidth because this class is for delay-sensitive traffic. For AF class, the criterion is inversed and so parameter p_1 is greater to express the preference of LSPs with important residual bandwidth.

Table 1

Example of parameter values for the three classes of traffic

Class	EF	AF	BE
p_0	0.7	0.3	0.5
p_1	0.3	0.7	0.5

This stage can be ameliorated by adapting dynamically the parameters of the splitting ratio equation depending on the network state.

Figure 5 gives PEMS flowchart to summarize how it works. The meaning of notations are as follows:

- $de(i)$: delay of LSP_i ;
- $b(i)$: residual bandwidth of LSP_i ;
- CP_{EF} , CP_{AF} , CP_{BE} : candidate path set for EF class, AF class and BE class, respectively;
- d_{cc}^k : k th demand with class cc ;
- CP_{cc} : current class (one in CP_{EF} , CP_{AF} or CP_{BE});
- $CP_{potential}^{cc}$: subset of CP_{cc} corresponding to LSP_i that can process the requested demand d_{cc}^k .

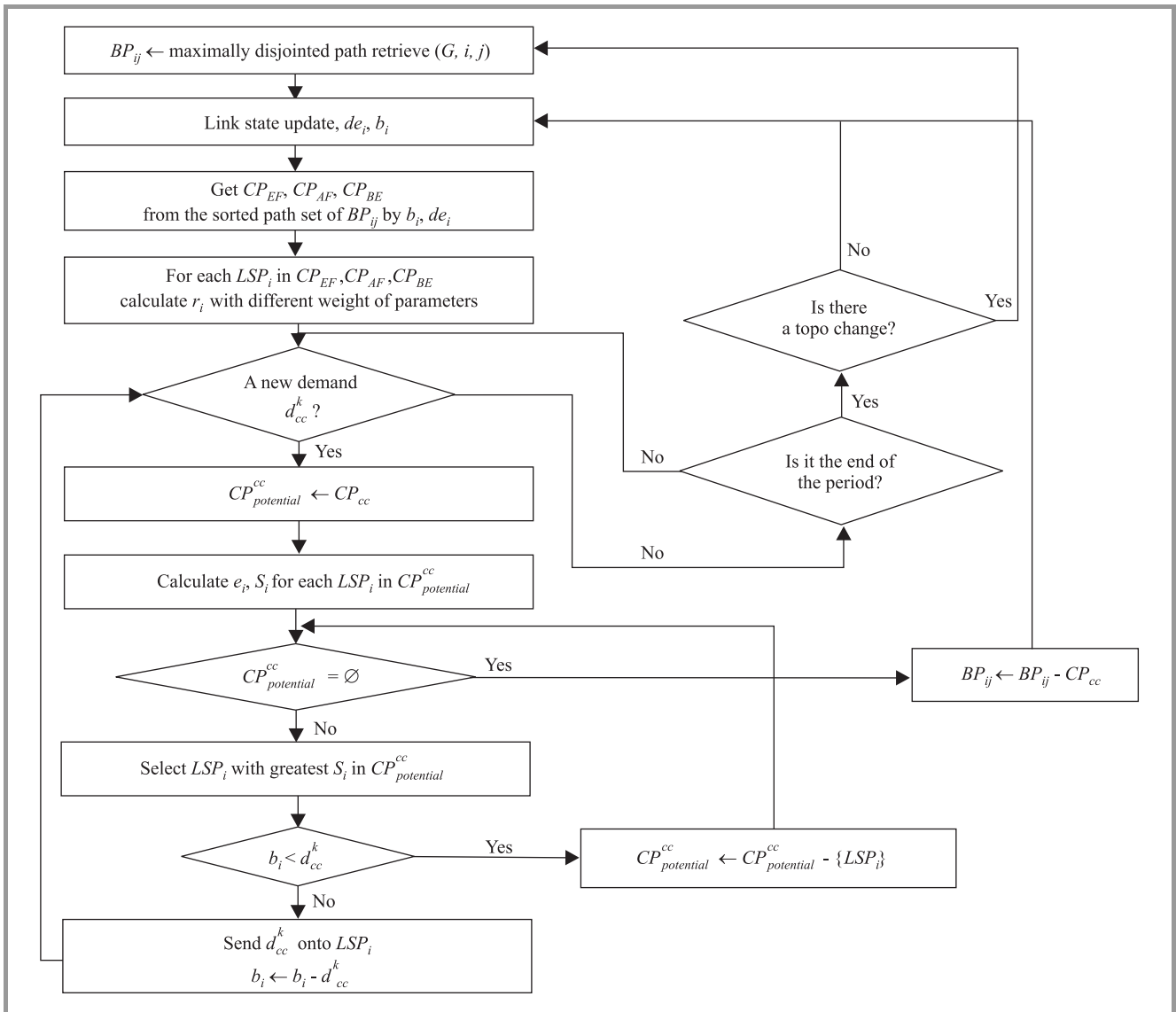


Fig. 5. PEMS flowchart.

In the online mode, when link state information are updated, new candidate paths for each class are calculated, based on updated information, such as measured delay and residual bandwidth. At this point, we use metric ordering by delay and residual bandwidth. This phase selects multiple low-delayed paths in the ordered paths set as candidate paths of delay-sensitive traffic and selects multiple paths having more residual capacity for the traffic to which the bandwidth is important for multipath routing to each traffic class.

Several simulations on multiple architectures have been done to assess PEMS in comparison with LBWDP. Different architectures have been generated using the generator BRITE, trying to be as close as possible to connectivity in a MPLS area. All simulations were conducted with MPLS network simulator for NS2 (MNS). In order to obtain comparable results for the two models, for each architecture we have defined traffic scenarios to apply to both models. In each simulation the goal is to transfer requested traffics

between pairs of routers. Requested traffics are generated every 2 s and are all at a rate of 500 kbit/s. They belong to one of the three differentiation class (EF, AF or BE). The class is selected randomly but is the same for both models. Each traffic is stopped after a delay common for the two models. Every 3 s, each router performs its link state update to refresh the routing model parameter.

The first simulations were based on architectures of tens nodes in order to simultaneously verify the correctness of PEMS model and to compare it with LBWDP. Figure 6 illustrates the type of architecture generated by BRITE for 31 nodes. Figure 7 shows the obtained results with regard to delay criteria. These results shows that PEMS deals differentiate the flows of the three classes. Indeed, for each architecture, the average delay obtained with PEMS for the class EF is smaller than for the delay of class AF traffic which is smaller than the delay experimented by class BE traffic. For LBWDP, one can see, for example, that for 10 or 20 nodes EF traffics results in a poorer delay.

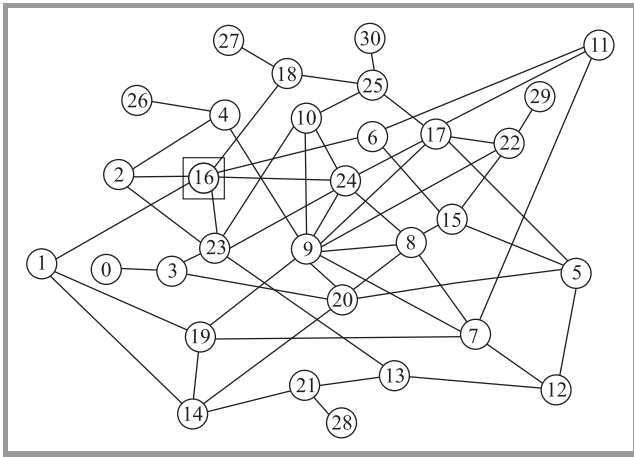


Fig. 6. Example of topology generated by BRITE for 31 nodes.

The second category of simulations were based on architectures of several hundred nodes. In this case, our main goal was to verify capacity to optimize traffic splitting in a dense architecture. Another goal was to verify the scalability of models, but this problem is out of the scope of this paper. For traffic splitting, simulations do not take care of

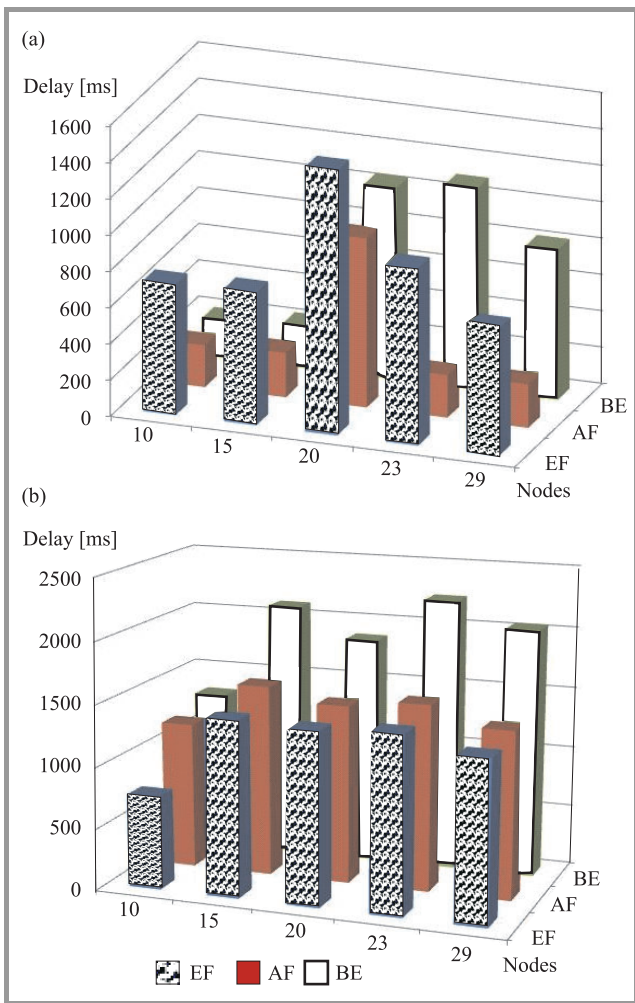


Fig. 7. Performance benchmarking between (a) LBWDP and (b) PEMS with regard to average delay.

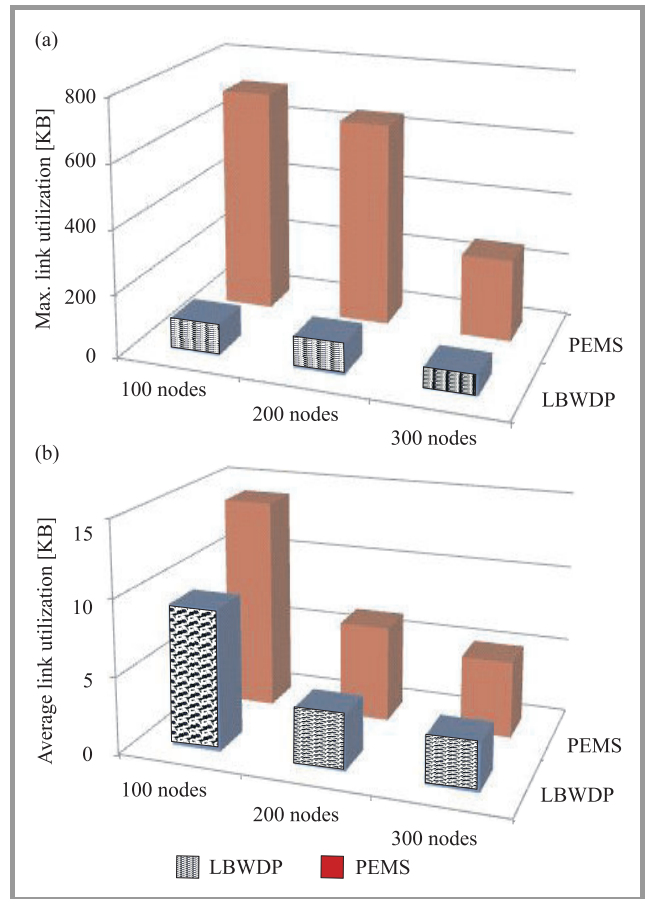


Fig. 8. Performance benchmarking between LBWDP and PEMS: (a) maximum and (b) average link utilization.

the class of the traffic. In this case the comparison criterion is link utilization. The simulations give both maximum link utilization and average link utilization. Indeed, maximum link utilization indicates if a model privileges some paths. The average link utilization measures the average of utilization rate of all the links used in architecture. Thus if this average is low, many more links of the architecture have been used.

The results illustrated in Fig. 8 prove that LBWDP better balances the traffic in the network as it does not take account of each traffic class to route.

5. Conclusions

Multiprotocol label switching offers many advantages to service providers. In order to support today's various kinds of applications, the system needs to guarantee the quality of service. However, MPLS is incapable of providing differentiated service levels in a single flow. Hence MPLS and DiffServ seem to be a perfect match and if they can be combined in such a way to utilize strong points of each technology it can lead to a symbiotic association that can make the goal of end to end QoS feasible. DiffServ aware traffic engineering mechanisms operate on the basis of different DiffServ classes of traffic to improve network per-

formance and extend the base capabilities of TE to allow route computation and admission control to be performed separately for different classes of service. Algorithms like PEMS seem to be a good compromise between improvement of resource utilization and the QoS required by end users.

A problem not addressed here is the comparison of PEMS to other models in terms of scalability. Our actual simulations results suggest that PEMS is scalable. This must be verified by simulations confirming a polynomial complexity of its algorithms. We think that PEMS must be scalable since this complexity concerns only the edge router of a MPLS network.

These models have yet to be assessed on real hardware architecture in order to confirm the performance illustrated by the simulations. Another important perspective is the ability to adapt models such as PEMS to the guarantee of quality of service of end-to-end communications. This poses the problem of application of DS-TE routing to inter-domains.

References

- [1] A. Alles, "ATM internetworking", in *Proc. Eng. InterOp Conf.*, Las Vegas, USA, 1995.
- [2] J. C. R Bennett and H. Zhang, "WF2Q: worst-case fair weighted fair queueing", in *Proc. IEEE Infocom'96 Conf. Comput. Commun.*, San Francisco, USA, 1996.
- [3] R. Braden, D. Clark, and S. Shenker, "Integrated service in the Internet architecture: an overview", RFC 1633, June 1994.
- [4] D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated service", RFC 2475, Dec. 1998.
- [5] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols", *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 118–124, 2002.
- [6] J. Ash and W. S. Lai, "Max allocation with reservation bandwidth constraints model for Diffserv-aware MPLS traffic engineering and performance comparisons", RFC 4126, June 2005.
- [7] F. Le Faucheur, "Maximum allocation bandwidth constraints model for Diffserv-aware MPLS traffic engineering", RFC 4125, June 2005.
- [8] F. Le Faucheur, J. Boyle, W. Townsend, D. Skalecki, K. Kompella, and T. D. Nadeau, "Russian dolls model for DS-TE", RFC 4127, June 2005.
- [9] F. Le Faucheur, J. Boyle, W. Townsend, D. Skalecki, K. Kompella, and T. D. Nadeau, "Protocol extensions for support of Diffserv-aware MPLS traffic engineering", RFC 4124, June 2005.
- [10] K. Lee, "Modèle global pour la qualité de service dans les réseaux de FAI: intégration de DiffServ et de l'ingénierie de trafic basée sur MPLS", Ph.D. thesis, Ecole Centrale de Lille, France, 2006.
- [11] Y. Lee, Y. Seok, Y. Choi, and C. Kim, "A constrained multipath traffic engineering scheme for MPLS networks", in *Proc. IEEE ICC'2002 Conf.*, New York, USA, 2002.
- [12] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS adaptive traffic engineering", in *Proc. Infocom'2001 Conf.*, Anchorage, USA, 2001, pp. 1300–1309.
- [13] J. Song, S. Kim, and M. Lee, "Dynamic load distribution in MPLS networks", in *Proceedings of ICOIN 2003*, Lecture Notes in Computer Science, vol. 2662. Heidelberg: Springer, 2003, pp. 989–999.
- [14] K. Lee, A. Toguyéni, and A. Rahmani, "Hybrid multipath routing algorithms for load balancing in MPLS based IP network", in *Proc. AINA'2006 Conf.*, Vienna, Austria, 2006, pp. 165–170.
- [15] N. Srihari and Z. Zhi-Li, "On selection of paths for multipath routing", in *Proc. IWQoS'01 Conf.*, Karlsruhe, Germany, 2001.
- [16] K. Lee, A. Toguyéni, and A. Rahmani, "Periodic multi-step routing algorithm for DS-TE: PEMS", in *Proceedings of the CAiSE07 Workshops and Doctoral Consortium*, Lecture Series on Computer and Computational Sciences, vol. 2. Berlin/Heidelberg: Springer, 2006, pp. 1–4.



Armand Toguyéni was born in Dakar, Senegal, in 1964. He obtained in 1988 the engineer Diploma of the Institut Industriel du Nord (French Grande Ecole) and the same year his Master degree in computer sciences. He obtained a Ph.D. in automatic control for manufacturing and discrete events systems in 1992 and his Habilitation à Diriger des Recherches in 2001. He is a Professor of computer sciences and computer networks at the Ecole Centrale de Lille, France. He has in charge the Department of Computer Sciences of the Institut de Génie Informatique et Industriel de Lens. His research interests is the quality of service of D.E.S. More particularly one of his topic research is the design of new model to improve QoS in Internet. His research also concerns the development of networked control systems.

e-mail: Armand.Toguyeni@ec-lille.fr

LAGIS, Ecole Centrale de Lille, Cité scientifique – BP 48 59651 Villeneuve d'Ascq Cedex, France



Ouajdi Korbaa obtained in 1995 the engineering Diploma from the Ecole Centrale de Lille, France, and in the same year, the Master degree in production engineering and computer sciences. He is Ph.D. in production management, automatic control and computer sciences, of the University of Sciences and Technologies of Lille, France, since 1998. He also obtained, from the same university, the Habilitation à Diriger des Recherches degree in computer sciences in 2003. He is full Professor in the University of Sousse, Tunisia, and Director of the E-learning Department. He published around 70 papers (journals, plenary sessions, books chapters and conferences) on scheduling, performance evaluation, optimization, design, and monitoring. His current research field is the discrete optimization and more particularly cyclic scheduling, production planning, networks QoS optimization. He is reviewer for different journals and conferences. He is references by the "Who'sWho" in science and engineering editions of 2004, 2006, and 2008.

e-mail: ouajdi_korbaa@yahoo.fr

ISITC, Laboratory LI3, University of Sousse
5 bis, rue du 1^{er} Juin 1955
Hammam Sousse 4011, Tunisia