Mechanism for per-Class QoS Monitoring in IP/MPLS transport networks

Joan A. García Espín

Department of Telematics Engineering, Technical University of Catalonia (UPC)
joan.antoni.garcia@upc.edu

Abstract— This paper presents a study on the usage of MPLS as the technology for transport networks which have to be sensitive to IP Differentiated Services. Advantages of MPLS for implementing Traffic Engineering and the capabilities that lead it towards integration with IP-DiffServ are discussed. Several solutions for DS-TE MPLS are exposed. On the other hand, a mechanism for monitoring classes of service per node in DS-TE MPLS networks is presented.

Index Terms— Differentiated Services (DiffServ), dynamic resource allocation, Multi-Protocol Label Switching (MPLS), quality of service (QoS), traffic classification.

I. INTRODUCTION

Next Generation Internet is envisioned nowadays as a cloud of geographically disperse services, location-independent resources and high networking performance integrating security and quality of service/experience (QoS/QoE) for users end-to-end. In order to achieve this challenge, current network operators have migrated from ATM to Multi-Protocol Label Switching (MPLS) technology, mainly due to the difficulties ATM shows in large-network management, QoS scalability (using Integrated Services – IntServ– model) and integration with IP.

For the previous reasons, this paper introduces the problem of QoS provisioning in MPLS networks using native IP Differentiated Services –DiffServ– and the existing techniques for implementing it [1] [2] and moreover proposes a solution for bandwidth monitoring in DiffServ-MPLS networks. Therefore, two essential schemes are considered: (i) IP Differentiated Services [6] and (ii) MPLS with Traffic Engineering (MPLS-TE) capabilities [7].

Currently, MPLS is considered one of the technologies that better performs network service convergence for voice, video and data. One of the major reasons is the Traffic Engineering functionalities it offers, which helps providing class of service (CoS) differentiation, traffic prioritization, network resilience, path redundancy and resource optimization. However, in order to support the trend of IP towards universal transport network, new mechanisms for assuring QoS are needed, since these networks have been so far designed for best effort traffic, which means that no guarantees are provided to the data flow/aggregate.

DiffServ-MPLS architecture is able to provide service differentiation with a given QoS degree, less strict than IntServ does. If end-to-end, guaranteed QoS service provisioning is desired, current IP networks need to be enhanced in terms of service availability, efficient QoS provisioning and lower routing complexity. On the contrary, DiffServ on IP networks offer flexibility, scalability, robustness and widely adopted. In order to guarantee user requests and to leverage network utilization at maximum, a good monitoring/management of DiffServ-MPLS is fundamental. Therefore, there is a need to enhance MPLS network functionalities in order to fully support IP DiffServ mechanisms with, for example, automatic resource provisioning based on DiffServ CoS, enhanced scheduling based on service policies, optimized queue dimensioning or specific DiffServ to MPLS CoS mapping.

II. QOS SERVICE MODELS IN IP NETWORKS

A. Best Effort

The traditional datagram model is the so-called Best Effort model. Although categorized as a QoS service model, it is not providing QoS to traffic. By contrast, Best Effort is precisely based on the application of no guarantees of complying the service (lost packets, delayed, etc.). Moreover, Best Effort model is not handling classes of traffic; every single packet is not different from other packets from other users/services. Hence, it is neither possible to define classes of service nor assuring a given treatment to packets of a given flow.

B. Integrated Services (IntServ)

Integrated Services architecture was defined in RFC 1633. This service model is based on guaranteeing a given QoS level per-flow using strict bandwidth reservations. Indeed, it is far from Best Effort model, since it is providing a hard QoS with high granularity.

The former characteristics of IntServ model are:

1) Strict bandwidth reservations. IntServ model uses strict allocation of network resources (bandwidth) per-flow in order to satisfactorily achieve the desired QoS level.

2) Call Admission Control (CAC). The strict bandwidth reservation feature requires a mechanism to determine whether a new connection request can be accepted or not.
CAC performs this role considering current (present) occupation of each node, that is, each node performs CAC.

3) Signaling for reservation setup/teardown. IntServ model needs signaling for accomplishing Path Reservation task. For this purpose, Resource reSerVation Protocol (RSVP) was defined in RFC 2205. RSVP consists of a mechanism for sending an expedition packet (PATH request) from origin to destination nodes. If all nodes in the path can assure reservation constraints in terms of bandwidth (mainly), destination node generates a RESV message which is sent back to origin node through the same route. Every node receiving a RESV message, firmly reserves the resources associated. Once origin node gets RESV message, it assumes reservation has been successfully set up.

4) Node by node configuration. IntServ model must be configured node by node, that is, every router along the path must be set up to handle the same rules.

Last word on IntServ is related to scalability. This model works well for small-scale networks. Several problems appear when large number of flows has to be handled with IntServ. Moreover, this model requires network nodes to retain and store state information per flow, which again complicates the situation for large number of data flows.

C. Differentiated Services (DiffServ)

Scaling problems from IntServ derived into a less strict QoS service model, Differentiated Services. DiffServ is also called the loose or soft QoS model.

DiffServ architecture was specified in RFC 2475 and designed to scale well when dealing with large number of data flows. The keys of DiffServ success are the definition of a set of functions to be performed at the edge of the network domain, and moreover the use of data flow aggregation.

The main characteristics of DiffServ are:
1) Aggregates data flows.
2) Defines a set of traffic conditioning rules to be applied at the entrance of the network, that is, in the edge routers. Traffic conditioning consists of several processes such as classification, marking, policing or shaping.
3) Defines a set of Per-Hop-Behaviors (PHB) in the core routers of the network. PHBs are simple actions to perform in switching very quickly, such as packet with mark X entering interface IF1, must leave the node via IF5 with mark Y.

Unfortunately, DiffServ also has some drawbacks. On the one hand, when using a DiffServ model over an IP domain, it is difficult to known what the exact end-to-end delay will be. On the other hand, DiffServ does not assure 100% a given capacity for a given CoS, as it is not doing a hard reservation of the resources.

III. DiffSERV TECHNIQUES FOR QoS

A. DiffServ header and detailed model

DiffServ is based on a node taking a decision on what the next hop for every packet is by simply analyzing its DiffServ header. Originally, RFC 791 (Internet Protocol) reserved 8 bits from IP header to define a Type of Service (ToS) field of 3 bits (named Delay, Throughput and Reliability) and Precedence field of 3 bits –last 2 bits were set to zero. On a later stage, this field definition was revised in RFC 2474, where a joint field (Precedence plus ToS) called Differentiated Services Code Point (DSCP) with 6 bits was defined, preserving double zero termination. The first three bits of DSCP are called Class Selector Code Point and are directly mapped with legacy Precedence field for backward compatibility. Equivalence between IP Precedence and DSCP classes can be found in the following table:

<table>
<thead>
<tr>
<th>IP Precedence (RFC 791)</th>
<th>IP Legacy type</th>
<th>DSCP Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Routine</td>
<td>Best Effort</td>
</tr>
<tr>
<td>001</td>
<td>Priority</td>
<td>Assured Forwarding 1</td>
</tr>
<tr>
<td>010</td>
<td>Immediate</td>
<td>Assured Forwarding 2</td>
</tr>
<tr>
<td>011</td>
<td>Flash</td>
<td>Assured Forwarding 3</td>
</tr>
<tr>
<td>100</td>
<td>Flash Override</td>
<td>Assured Forwarding 4</td>
</tr>
<tr>
<td>101</td>
<td>Critical</td>
<td>Expedited Forwarding</td>
</tr>
<tr>
<td>110</td>
<td>Inter-network Control</td>
<td>Inter-network Control</td>
</tr>
<tr>
<td>111</td>
<td>Network Control</td>
<td>Network Control</td>
</tr>
</tbody>
</table>

Fig. 1. Evolution of ToS header in IP Protocol from RFC 791 to RFC 3168.

In DiffServ, the treatment of traffic flows in each node follows the same pattern with four differentiated blocks: classifier, marker, meter and shaper/dropper. In brief, each block performs the following tasks:

-- Classifier: evaluates packet header and decides what traffic class it belongs to.
-- Marker: alters or not DSCP field in packet header if required. This process is also called coloring.
-- Meter: checks whether the packet (and flow) is respecting traffic profile agreed (transmission rate, data peaks, etc.) and communicates with Marker or Shaper/dropper for them to act accordingly (class type degradation, profile shaping, excess packets discarding, etc.).
-- Shaper/dropper: act over outgoing packet either delaying it for better fitting a traffic profile (shaping) or discarding it (dropping).
The following image represents in detail how blocks described above inter-work in the DiffServ model:

![Block diagram of the DiffServ model](image)

**Fig. 2.** Block diagram of the DiffServ model. After metering, P1 is complying traffic profile but P2 is not, hence, P1 is directly passed to shaper and P2 is re-marked for later dropping. Meanwhile P3 was in-profile but its DSCP was to be changed at this node, before passing it to shaping process (Source: Cisco Systems, Inc.).

As a result of the new 6 bit coding for DSCP, up to 64 traffic classes can be identified in the network. Hence, traffic classification is now an important process and, as most of the processing intelligence is performed at the entrance of the network (ingress nodes); a special care must be taken in queue dimensioning.

**B. Per-Hop Behavior**

DiffServ model allows performing resource allocation per-flow; since each packet is identified and processed according to the CoS it belongs to (DSCP). This relationship is called Per-Hop Behavior (PHB) and refers not only to CoS but also to aggregate CoS flows, the so-called Forwarding Equivalence Class or FEC. Basic PHB definitions (already introduced in Table I) are Expedited Forwarding, Assured Forwarding and Best Effort. The first guarantees QoS to traffic flow, while the second one assures QoS but adding a discard probability in case of network congestion.

PHB describes, per node and per CoS/FEC, all needed information about resource allocation (bandwidth), queuing/scheduling mechanisms (out of the scope of this document), and packet shaping dropping policies. Moreover, PHB can include congestion avoidance schemes. Typically, Random Early Detection and Weighted Random Early Detection are considered as congestion avoidance techniques.

In IP networks, the whole DiffServ process requires that every node on the path analyses IP headers, decoding IP addresses and ToS/DSCP fields for applying PHB. This issue is indeed time-consuming, as IP address decoding and route lookup load node’s CPU with avoidable work, at least in the core of the network. MPLS technology is able to perform quick packet switching and can be aware of DiffServ model, as it is explained on the coming section.

For example, UMTS networks support four traffic classes: Conversational, Streaming, Interactive and Background, each of them requiring different QoS constraints. Conversational and streaming require end-to-end QoS due to their need for bandwidth, delay and jitter guarantees. Interactive and Background classes are less strict for QoS parameters but require service differentiation. Therefore, the monitoring mechanism presented in this paper would help tracking QoS guarantees for UMTS services on MPLS networks, which fit the needs of most of mobile operators [8] [9].

**IV. QoS in MPLS Networks Integrating IP DiffServ and Traffic Engineering**

This section explains what features of MPLS and MPLS with Traffic Engineering (TE) make this technology suitable for integration with IP DiffServ schemes in order to provide QoS in the Next Generation Internet. A special notice must be done on the scope of this section: this paper only considers intra-domain QoS provisioning using DiffServ and MPLS (with or without Traffic Engineering). Inter-MPLS-domain issues are out of scope of this document.

It is also important to remark that MPLS (RFC 3031) does not define a new QoS architecture or model, because MPLS technology itself is not able to provide any given service quality to network traffic [5]. Instead, MPLS uses the standard definition of IP DiffServ (RFC 2475) for provisioning QoS. MPLS-DiffServ integrated model is defined in RFC 3270.

**A. MPLS-DiffServ: Label Switched Path Modes**

With respect to traffic handling, MPLS uses traffic flows aggregation at the edge nodes (ingress nodes) in order to achieve higher scalability. As commented before, several traffic flows related to similar applications or which may receive the same treatment inside the MPLS domain are aggregated into the Forwarding Equivalence Class. This FEC will receive a given PHB along the domain, that is, there is an aggregated processing of FEC in the code nodes and forwarding is based on MPLS label. At this point is where Label Switched Path (LSP) in MPLS acquires relevance for DiffServ integration.

MPLS-DiffServ allows mapping DSCP fields into MPLS header (LABEL and/or EXP fields). Label Switch Routers in MPLS (core nodes of MPLS domain) can only access MPLS header for fast switching purposes and, thus, no access to IP header is permitted. However, DSCP needs 6 bits and EXP can only offer 3. The problem of mapping Precedence/DSCP fields into MPLS header has two solutions:

1) If 8 or less PHBs are needed in the core, Precedence can be mapped directly to EXP field.

2) If more than 8 PHBs are needed in the core, DSCP can be mapped into LABEL field.

First situation generates a solution using the so-called E-LSPs (EXP-inferred-LSPs) whereas the second one creates the so-called L-LSPs (Label-inferred-LSPs). The second solution, L-LSPs, is intended to carry only one CoS into the LSP, which increases substantially the number of LSPs needed in the domain. By contrast, E-LSPs allow up to 8 CoS/PHBs per LSP, which reduces the overall number of LSPs in the domain and makes easier to deploy end-to-end services [10].
Another appropriate feature of MPLS for its integration with IP DiffServ is the possibility of defining different LSP modes. If E-LSP solution is considered, these modes depend on how user and provider agree on label/DSCP mapping at the egress node of the MPLS domain. MPLS allows three LSP modes: uniform, pipe and short-pipe:

-- Uniform LSP mode assumes the entire domain under administrative right of the Service Provider (SP). Therefore, SP will replace user’s original Precedence/DSCP value with the topmost EXP field before leaving the domain. Outgoing packet will be policed according to the new DSCP.

-- Pipe LSP mode assumes both SP and user implementing (different) DiffServ policies for queuing and scheduling. SP keeps user’s Precedence/DSCP in the packet at the egress node but outgoing policing is made following EXP value of the last label the packet had inside the MPLS domain (SP controls the policy).

-- Short-pipe LSP mode assumes the same situation as Pipe mode but outgoing policing at the egress node is made following original user’s Precedence/DSCP value (user controls the policy).

Both SP and user should agree on the most suitable mode for them.

B. Traffic Engineering over MPLS (MPLS-TE)

Apart from MPLS integration with IP DiffServ, MPLS also offers the possibility of using an enhanced intelligence for routing and traffic handling, also known as Traffic Engineering (TE). The application of TE over MPLS networks is specified in RFC 2702.

MPLS-TE introduces a set of new characteristics which reinforce MPLS-DiffServ integration when combined. MPLS-TE main characteristics are:

-- Use of constraint-based, explicit or implicit routing.

-- Admission control capabilities.

-- Path protection and restoration capabilities.

-- Use of signaling with traffic extensions for TE such as RSVP-TE.

-- Use of routing protocols with traffic extensions for TE such as OSPF-TE or IS-IS.

To sum up, MPLS-TE advantages can be concentrated on three main features:

1) Possibility of spreading traffic flows around the network in order to achieve more flexibility than an Interior Gateway Protocol could offer. This is especially important when a link is suddenly disconnected and traffic flows have to be re-routed in very little time.

2) Feasibility of reserving network resources (mainly bandwidth) per link and PHB, using MPLS header fields such as label or EXP.

3) Manageability of service unfairness during periods of network congestion. MPLS-TE offers the possibility of statically or dynamically configuring back-up paths and assigning a certain pool of resources to them.

C. DiffServ-aware TE over MPLS (DS-TE MPLS)

Standalone MPLS-TE needs some minor adaptations to be aware of IP/MPLS DiffServ operation and characteristics (RFC 4124). The idea behind DS-TE MPLS is to bring MPLS-TE to a dimension where traffic classes can be assigned pools of resources. This way, while MPLS-TE only allows a single pool of bookable bandwidth per link, DS-TE MPLS performs a partitioning of this global pool into sub-pools. This feature allows creating, for example, a sub-pool for one CoS/PHB and ensuring this CoS a guaranteed set of resources in the node.

Most of the manufacturers have implemented solutions with eight sub-pools available per physical interface, one sub-pool per EXP value in MPLS, that is, one per CoS in Precedence/DSCP. Two models have been defined for bandwidth partitioning in sub-pools: (i) Maximum Allocation Model –MAM– and (ii) Russian Dolls Model –RDM. In (i) the main pool is divided in sub-pools such that the sum of all bandwidths allocated per pool is equal to the main pool, whereas in (ii), sub-pools are assigned per groups of classes. An example for RDM can consider groups of 2 consecutive-numbered classes, that is, sub-pool 1 is assigned to class 1; sub-pool 2 is assigned to class 1 plus 2; sub-pool 3 is assigned to classes 1 plus 2 plus 3; and so on up to the main pool.

In summary, DS-TE MPLS provides a seamless integration of IP DiffServ and MPLS-TE, combining their characteristics to get per-class traffic engineering capabilities over MPLS networks. This merging allows the provisioning of strict point-to-point QoS guarantees with much higher scalability than IntServ. However, a problem appears: DS-TE MPLS has no native mechanisms for checking whether QoS levels are being met or not. This issue has been traditionally faced by developing hardware-specific monitoring modules and using Simple Network Management Protocol (SNMP). On the next section, this paper proposes a basic monitoring mechanism per-class which could be easily integrated in the network management system by implementing a software module or even, due to its simplicity, could even be integrated in DS-TE MPLS protocol architecture.

V. A PROPOSAL FOR PER-CLASS QoS TRAFFIC MONITORING MECHANISM IN DS-TE MPLS NETWORKS

A monitoring system is responsible of receiving/requesting the different notifications and alarms sent by network nodes and applying some changes in the configuration of the nodes if necessary. However, quick responsiveness from network management system is needed for minimizing excessive data loss of network congestion, while network resource usage is maximized as decisions are taken quickly. In the case of provider’s IP/MPLS backbone networks, traffic profiles are almost flat and have soft bandwidth variations in short time scale (seconds order). Therefore, responsiveness is not such critical in this case, but it is highly recommended, although few changes are inferred in the network at the end. This allows a continuous and precise tune up of network devices without introducing nervousness in service patterns.
A. Scenario and considerations

Consider a DS-TE MPLS domain, which is accessed by a cloud of users which utilize marked traffic with IP Precedence set up from 0 to 7 (from now on IPPn, where n=0..7). In this case, IPP0 corresponds to Best Effort (BE) traffic and IPP7 to EF traffic; CoS ranging from 0 to 7 have increasing priority order.

MPLS domain is a cloud composed of Label Edge Routers in the limits and Label Switching Routers in the core. LERs perform label pushing or popping if they are ingress or egress, respectively. LSRs can either perform pushing, popping or swapping, but it is not crucial in the situation considered.

Users generate slow-moving traffic patterns, that is, no big variations are observed during short periods of time (tens of minutes or below). Hence, users can be other networks with regular, slow-changing data flows towards third parties’ domains.

B. Problem statement

MPLS domain administrator desires to monitor a set of CoS from different LSPs across the domain. An initial partition of main bandwidth pool following MAM criteria has been defined. However, domain administrator will not assign (configure in the network devices) the whole bandwidth sub-pool to each CoS unless measured traffic is close to upper bound of the reservation. For maximizing network usage purposes, unused bandwidth in sub-pools will be borrowed by other priority CoS which need more than their assigned bandwidth.

C. Proposed solution

The need for per-class monitoring is assumed to be feasible in network devices composing the DS-TE MPLS domain, as this problem is out of scope of this study.

In this situation, a proactive monitoring system based on device polling is proposed. Periodically, a software module can poll for current recorded bandwidth usage per-class inside LSP. There is no need to monitor IPP0, since it is assigned a BE service model and it is only allowed to use remaining (not reserved) bandwidth in links.

Polling Algorithm

The polling mechanism consists of periodically taking samples of the bandwidth usage per class. Simple comparison (subtraction) of two consecutive samples is a good approximation, since traffic patterns are not due to change excessively if the sampling period is low enough, by definition of the scenario. However, this algorithm is poor in terms of memory and gives the same importance to old and current samples. Hence, I propose an Exponentially Weighted Moving Average (formerly EWMA), since the weighting for each older data point decreases exponentially and, thus, newer samples have more importance than older ones, but memory is preserved. This way, present traffic trends are highlighted.

Formerly, calculation of EWMA should be done following the expression from Roberts (1959) [12], where current observation is considered. Let $S_t$ be the calculation of EWMA for time $t$ (present sample time), $Y_t$ the sample taken for time $t$ (present time also) and $\alpha$ the smoothing factor (constant, between 0 and 1):

$$S_t = \alpha Y_t - (1 - \alpha) S_{t-1}$$

(1)

When choosing the smoothing factor, the decision depends on operator’s managerial needs and the degree of nervousness operator wants to provide its monitoring system. Moreover, the time to configure network nodes must be considered when choosing the smoothing factor together with the sampling period. A smoothing factor of $2/(N+1)$, where $N$ is the number of samples to be considered in the calculation of the average, would provide, for example, 75% of importance to present sample with $N=3$.

Maximum deviation thresholds

Although EWMA is able to control the nervousness of the management system, it is also proposed to set up a threshold for the monitoring system to interact with the network (reconfiguration) once over-passed. That is to say, unless present EWMA surpasses a given threshold, the monitoring system will not react recalculating bandwidth assignments to be configured on egress routers for CoS under monitoring.

Monitoring initialization

Once monitoring process started, EWMA can be initialized in several manners (assuming the smoothing factor is constant):

-- Setting the N-1 old EWMA calculations to the rate a given CoS has been assigned for calculating present EWMA. This solution does not provide trusty results in the transitory but it assumes the CoS is being transmitted at the committed rate.

-- Setting the N-1 old EWMA calculations to zero. This solution is slower than the previous one in converging to the real value and moreover, the first calculations must be obviated by the management system.

-- Other initial values for EWMA can be considered basing on operator’s needs, either theoretical or as a result of testing.

D. Advantages and disadvantages of the solution

A clear advantage, as stated before, is that operator’s management system has a close estimation on how many bandwidth users are consuming per class. This lets operator “play” with unused bandwidth in network nodes in order to perform over-booking of sub-pools already assigned which are not being used (for example, for BE traffic). This assigned-but-not-used resources can even be used by other CoS. Operator will not reconfigure network nodes, but will have to keep track of what resources are being borrowed by which CoS. Thus, network resources would increase its occupation ratio and operator would be able to obtain a higher overall
yield of the network.

A clear disadvantage of this system is that, if the network size increases, the complexity of keeping track of all CoS in all nodes increases. Therefore, scalability is envisioned as a disadvantage of the mechanism proposed.

Another drawback that should be studied is that EWMA tune up would depend on every operator and even could depend on temporal variations of the smoothing factor or the number of samples.

VI. CONCLUSION

In this paper, a revision of DS-TE MPLS usage and a mechanism for per-class QoS monitoring on these networks has been introduced and described. This system provides the operators with an end-to-end QoS monitoring mechanism across an MPLS-DiffServ transport network, as a solution to support new service requirements demanded latest multimedia applications.

Future work would go for formally describing the monitoring mechanism and analytically describing the co-allocation of resources per CoS. Moreover, a simulator and a first prototype should be implemented following guidelines stated in [11] and analysis over the prototype would show the performance and the reliability of the mechanism working real scenarios.

Other point that must be considered in the future is to fine-tune the numerous variable values that have been introduced in the EWMA calculation algorithm so performance of the mechanism can be optimized within the different scenarios.

REFERENCES