Accommodations of QoS DiffServ Over IP and MPLS Networks*

Abdullah AlWehaibi[1], Anjali Agarwal[1], Michael Kadoch[2] and Ahmed ElHakeem[1]

[1]Department of Electrical and Computer Engineering
Concordia University
1455 de Maisonneuve Blvd. W.
MONTREAL, Canada (H3G 1M8)

[2] Department de Genie Electrique
Ecole de Technologie Superieure
Universite du Quebec
MONTREAL, CANADA

{aa_alweh@ece.concordia.ca;ahmed@ece.concordia.ca}

Abstract:- Multicasting has become increasingly important with the emergence of Internet-based applications such IP telephony, audio/video conferencing, distributed databases and software upgrading. IP Multicasting is an efficient way to distribute information from a single source to multiple destinations at different locations. In practice IP is considered as a layer 3 protocol. Multiprotocol label Switching (MPLS) replaces the IP forwarding by a simple label lookup. MPLS combines the flexibility of layer 3 routing and layer 2 switching.

In this paper, we present a new fair share policy (FSP) that implements Differentiated Services to solve the problems of QoS and congestion control when multicasting is used. Analysis tools are used to evaluate our new fair share policy (FSP) for different scenarios. The results should provide insights for the comparisons between IP multicast in MPLS networks using FSP and plain IP multicasting using the same policy when DiffServ are adopted.

Key-Words: Multicast, IP, MPLS, DiffServ, QoS.

1 Introduction

Multicasting has been at the center of interest in the area of Internet activities and has already contributed to some major successes. IP Multicast supports group communications by enabling sources to send a single copy of a message to multiple recipients at different locations who explicitly want to receive the information [1]. With the huge increase demand for bandwidth, one of the challenges the Internet is facing today is to keep the packet forwarding performance up.

Recent developments in Multiprotocol label Switching (MPLS) open new possibilities to address some of limitations of IP systems. MPLS is an Internet Engineering Task Force (IETF) standard [2]. It replaces the IP forwarding by a simple label lookup mechanism. MPLS combines the flexibility of layer 3 (L3) routing and layer 2 (L2) switching, which enhances network performance in terms of scalability, computational complexity, latency and control message overhead. Besides this, MPLS offers a vehicle for enhanced network services such as Quality of Services (QoS)/ Class of Service (CoS), Traffic Engineering and Virtual Private Networks (VPNs). IP multicast in MPLS networks is still an open issue [3-4].

On the other hand, the IETF DiffServ working group is looking at a more scalable model and more likely to be easier to implement than IntServ/RSVP model [5]. In the DiffServ architecture, traffic that requires the same Per-Hop-Behavior (PHB) is aggregated into a single queue. The DiffServ architecture focuses on the use of DiffServ (DS) byte, which is the redefined 8-bit Type of Service (TOS) field in the IPv4 header or the IPv6 Traffic Class octet as a QoS mechanism. Packets are classified into the corresponding queues using their DiffServ Code Points (DSCP). Packets use DSCP bits in order to receive a particular PHB, or forwarding treatment. Marking, classification, traffic conditioning or policing are done at network

* This work was supported by a CRD grant from NSERC/BELL Canada

boundaries (first router for example) and packet treatment and handling is carried on each network node [6].

It would be interesting to compare QoS performance of IP and MPLS multicasting, given their particular constraints. In regular IP multicasting only overhead pertaining to IP multicast tree should be established, while in MPLS multicasting we have to add also the corresponding MPLS multicast tree establishment times and control packets. In this paper and taking all the above constraints into consideration, we evaluate the QoS performance for a typical router in the two cases of IP and MPLS multicasting. We also consider Differentiated Services; i.e. traffics with different priority classes.

2 Fair Share Policy (FSP)

FSP is not a call admission rather it is a traffic policing mechanism. In FSP, packets are discarded in case of congestion differently at each queue according to source priority and the maximum number in the queue; i.e. the source with higher priority will experience less packet discarding than sources with lower priorities.

FSP guarantees fairness among flows having the same priority (i.e., required QoS) in two respects: Firstly the buffer space allocated to lower priority traffic is larger; thus leading to less packet discard. Secondly by selective packet discarding on packets from the same flow and making sure that the total number of packets discarded per flow is the same for all flows with the same QOS requirement. In this paper, we only explore the first fairness mechanism.

3 The Analytical Model

Our analytical model is shown in Fig.1. In this model, a typical IP or MPLS router and our FSP traffic policing mechanism process three independent sources corresponding to different input traffic classes. Source 1 is assigned the highest priority, then source 2 and finally source 3. For this model, the enforcement is assumed to occur at the router (node) according to Fair Share Policy. The following assumptions are used:

1- Assume a Bernoulli arrival for all sources; in order to be short and discrete inter-arrivals.

2- FSP uses non pre-emptive priority queuing.

3- The arrival probabilities are α_1, α_2 and α_3 for each source respectively. Note that α represents the probability of receiving a packet while one packet is served on the channel.

4- Service disciplines for different queues are β_1, β_2 and β_3 for each source respectively.

5- Average queue sizes are $\overline{n_1}$, $\overline{n_2}$ and $\overline{n_3}$ for each source respectively.

6- Maximum buffer sizes are n_{1max}, n_{2max} and n_{3max} for each source respectively.

7- Total system buffer size:

$$B = n_{1\,max} + n_{2\,max} + n_{3\,max}$$, where

n_{imax} and $i = 1,2,3$ is calculated as:

$$n_{1max} = \left(\left(\frac{\overline{n1}}{\overline{n_1} + \overline{n_2} + \overline{n_3}}\right) * B\right),\quad n_{2max} = \left(\left(\frac{\overline{n_2}}{\overline{n_1} + \overline{n_2} + \overline{n_3}} * B\right)\right)$$

and $$n_{3\,max} = \left(\left(\frac{\overline{n3}}{\overline{n1} + \overline{n2} + \overline{n3}}\right) * B\right)$$

8- All of MPLS or IP routers on the subject Internet are homogeneous in providing resource and traffic conditions, so we take one of them as a representive for IP routers and the other one as a representaive for MPLS routers.

9- All packets are of the same length.

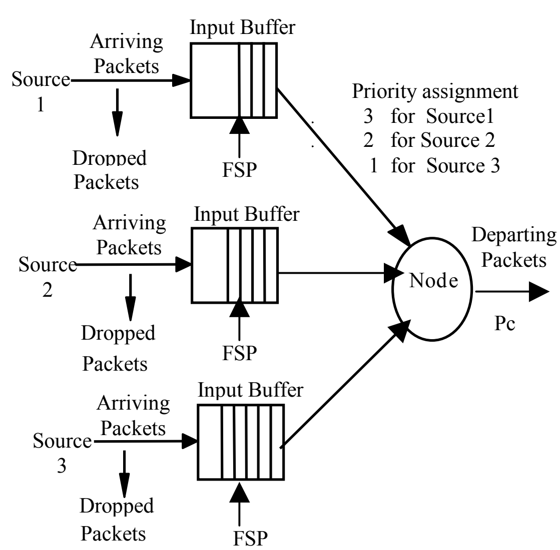

Fig.1 The analytical model

3.1 The Coupled State Diagrams

The coupled state diagrams for the analytical model in Fig.1 are shown in Fig. 2.

This diagram represents a typical router with 3 priority classes. The solution of the number in every class depends on the solutions of the other classes.; where $\beta_1 = 1$ always in order to give source with highest priority the best service probability, $\beta_2 = P_0^1$; i.e. packets from source 2 will be served only when the buffer corresponding to source 1 (which has higher priority) is empty and finally $\beta_3 = P_0^1 P_0^2$; i.e. packets from source 3

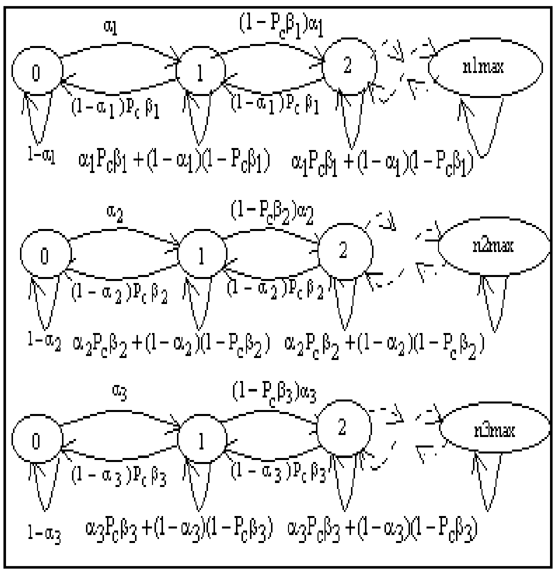

Fig.2 The Coupled State Diagram

will be served only when the buffers corresponding to source 1 and source 2 (which have higher priority) are all empty. P_c is the probability that reflects the comulative effects of channel error and congestion of next router (i.e. the probability of no loss or errors on the channel).

Packet loss probability for each source can be obtained by calculating the probability to be in last stage in the state diagram P_{n1max}, P_{n2max} and P_{n3max} respectively.

For IP based networks, the source arrival probability α is actually a compsite one; for instance α_1 (for source 1) can be written as:

$$\alpha_1 = \tau\alpha_1^1 + \alpha_1^2 \; , \; \tau = \frac{\Delta_1 + \Delta_2}{\Delta_1}$$

Where Δ_1 is the processing time at lower layers (for example MAC layer) and Δ_2 is the processing time at IP layer, α_1^1 is the intrinsic arrival probability at the application layer (on top of IP layer), α_1^2 is the extra arrival probability due to IP control overhead used to establish the IP multicast tree. The above equation can be rewritten in terms of α_1^1 as:

$$\alpha_1 = \tau\alpha_1^1 + \xi_1\alpha_1^1 \qquad \xi_1 = \frac{\alpha_1^2}{\alpha_1^1}$$

where ξ_1 is where is a factor.

Similarly for MPLS based networks, α_1 can be written as: $\alpha_1 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3$

Where α_1^1 and α_1^2 are the same as in the case of IP networks; α_1^3 is the extra arrival probability due MPLS control overhead used to establish MPLS multicast paths or tree.

α_1 can be rewritten in terms of α_1^1 as:

$$\alpha_1 = (1+\xi_1+\xi_2)\alpha_1^1 \qquad \xi_2 = \frac{\alpha_1^3}{\alpha_1^1}$$

where ξ_2 is a factor.

3.2 Solution of the Analytical Model

By writing the balance equations for the state diagram in Fig.2 [7], and solving these equations by iteration to find the probabilities. In order to write the equations in simpler forms we define:

$$\lambda = (1-P_c\beta)\alpha \; , \; \mu = (1-\alpha)P_c\beta$$

$$\text{and } \sigma = \alpha P_c\beta + (1-\alpha)(1-P_c\beta) \qquad (1)$$

To find a specific probability P_i :

$$P_i = \frac{1-\sigma}{\mu}P_{i-1} - \frac{\lambda}{\mu}P_{i-2} \qquad (2)$$

Where i=3,4…nmax.

$$\text{Notice that } P_1 = (\alpha/\mu)P_0 \qquad (3)$$

$$\text{and } P_2 = \frac{1-\sigma}{\mu}P_1 - \frac{\alpha}{\mu}P_0 \qquad (4)$$

P_i can be rewritten in the following form:

$$P_i = (a_i + b_i)P_0 \qquad (5)$$

where i=3,4…nmax; and nmax is a specific source maximum buffer size.

$$a_i = (\frac{1-\sigma}{\mu}a_{i-1} - \frac{\lambda}{\mu}a_{i-2}) \qquad (6)$$

Where $a_0 = a_1 = 0$ and $a_2 = \frac{(1-\sigma)\alpha - \mu\alpha}{\mu^2}$

$$b_i = (\frac{1-\sigma}{\mu}b_{i-1} - \frac{\lambda}{\mu}b_{i-2}) \qquad (7)$$

Where $b_0 = b_1 = b_2 = 0$ and $b_3 = \frac{-\lambda\alpha}{\mu^2}$

Taking into account that: $\sum_{i=0}^{nmax} P_i = 1$

P_0 can be found using the following:

$$P_0 = \frac{1}{1 + \sum_{i=1}^{nmax} (a_i + b_i)} \qquad (8)$$

then any probability can be found in terms of P_0 as in equation (5). The average number of packets in the buffer for a specific source can be found as:

$$\bar{n} = \sum_{i=0}^{nmax} i * P_i \qquad (9)$$

Notice that the loss probability is equal to the probability to be in last stage of state diagram; for example the loss probability for source 1 is: $P_{L1} = P_{n1max}$.

4. Analysis Results

Fig. 3 shows the average number of packets in the system buffer versus ξ_1 for all sources for both IP and MPLS. The figure shows that IP and MPLS will have very similar average number of packets especially for low priority traffic and when the intrinsic arrival rates are relatively high. Note that the value of τ is relatively small in which we assumed that the difference in packet processing between IP and MPLS is small.

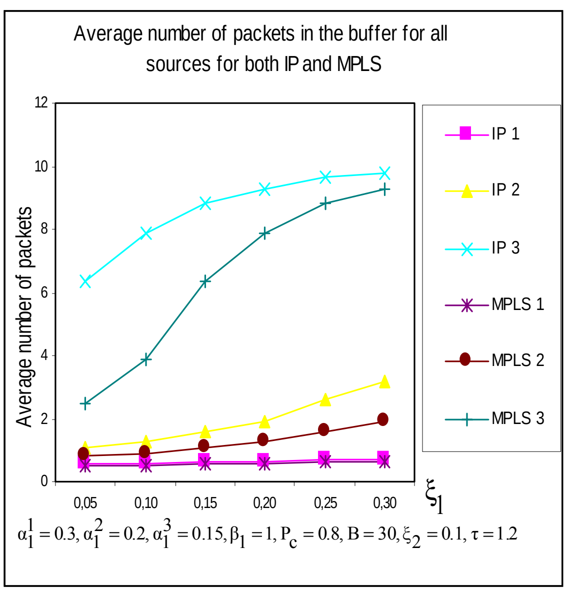

Fig.3 Average number of packets in the buffer for all sources for both IP and MPLS (small τ)

Fig.4 shows the packet loss probability for all sources for both IP and MPLS versus ξ_1 for relatively high intrinsic arrival rates and small τ. It shows that IP and MPLS have almost the same loss probability, except a small difference for source 3; and as ξ_1 increases the difference becomes even smaller.

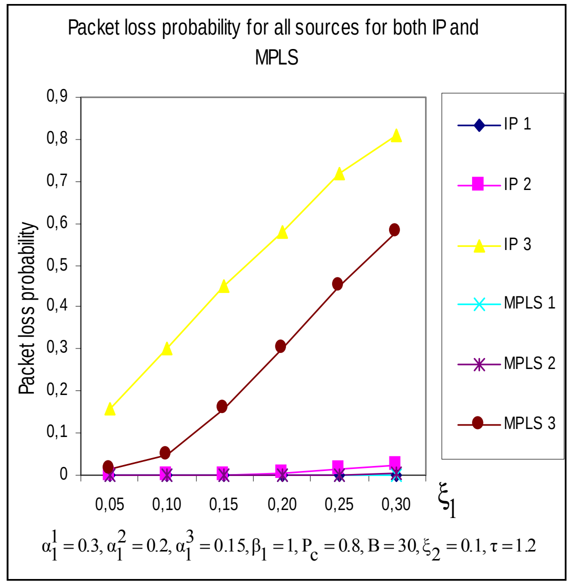

Fig.4 Packet loss probability for all sources for both IP and MPLS (small τ)

However, Figs.5 and 6 show that when τ increases MPLS will have superiority over IP in terms of average number of packets in the system buffer and packet loss probability. As shown in Fig. 5, the average number of packets in the system buffer in the case of MPLS is less than IP for all sources and this difference is clear for low priority sources 2 and 3. Fig. 6 shows that the packet loss probability in the case of MPLS is less than IP for all sources. This means when the difference in packet processing τ between MPLS and IP increases, MPLS will be better.

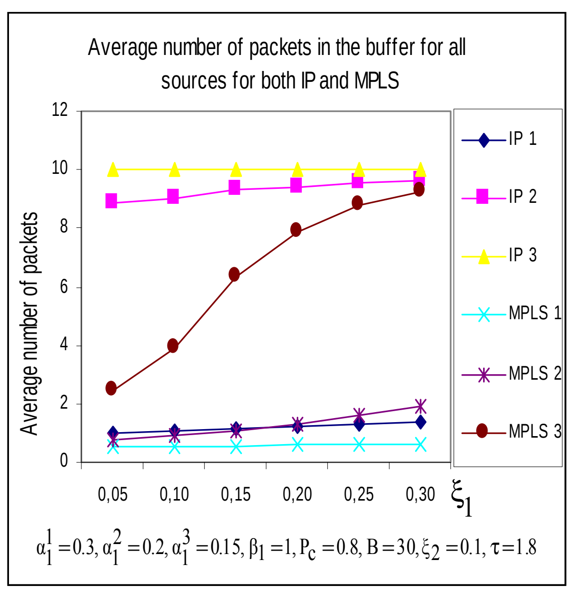

Fig.5 Average number of packets in the buffer for all sources for both IP and MPLS (large τ)

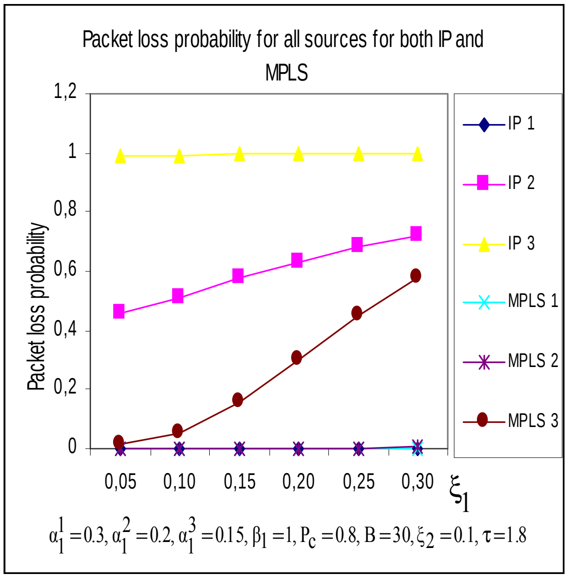

Fig.6 Packet loss probability for all sources for both IP and MPLS (large τ)

In the previous figures 3, 4, 5 and 6 ξ_2 was constant and relatively small; that's why MPLS performance was better or very similar to IP performance. However, in the next figures we will study the effects of ξ_2 on MPLS performance. Figs. 7 and 8 show that IP will have superiority over MPLS when ξ_2 increases especially for sources 2 and 3. As shown in Fig.7, the average number of packets in the system buffer in the case of IP (which is constant) is less than MPLS.

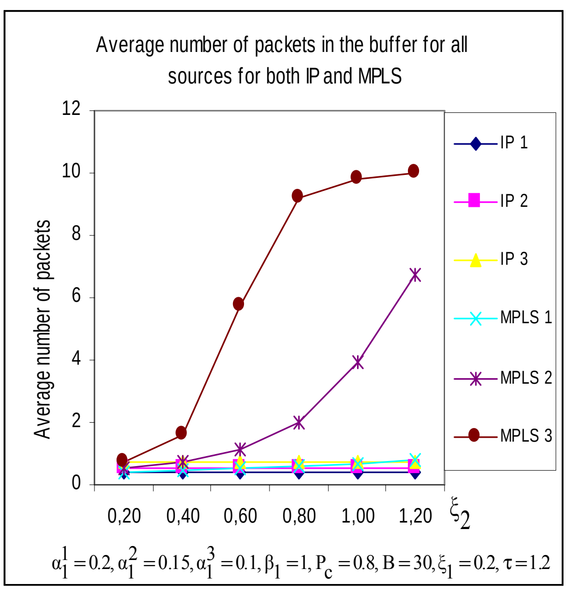

Fig.7 Average number of packets for all sources for both IP and MPLS (Effect of ξ_2)

Similarly, Fig. 8 shows that packet loss probability in the case of IP (which is constant) is less than MPLS. This means when the extra arrival rate due MPLS control overhead used to establish MPLS multicast paths or tree increases, IP will be perform better.

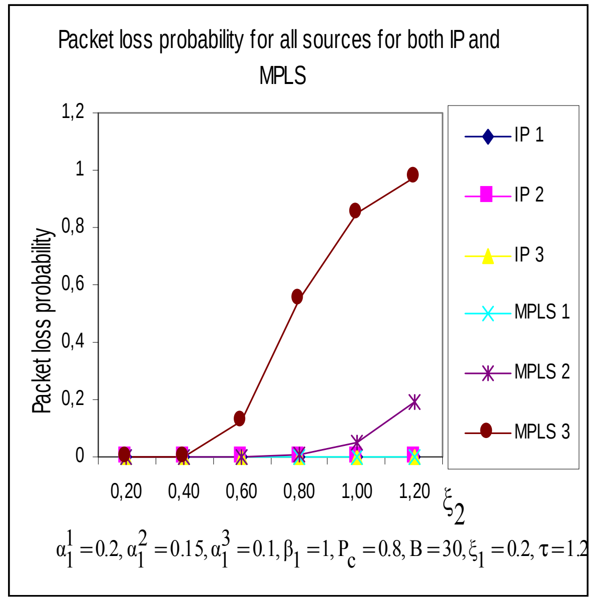

Fig.8 Packet loss probability for all sources for both IP and MPLS (effect of ξ_2)

5. Conclusions and Future Work

In this paper, a comparison between IP multicast sessions and MPLS multicast sessions is carried using analysis tools. In addition to that a new Fair Share Policy (FSP), which is a traffic policing mechanism is proposed to ensure proper QoS. Also, Differentiated Services are used in this comparison. In this paper, we found that when the difference in packet processing time between IP and MPLS is high, IP multicast will perform less efficiently than MPLS in terms of average number of packets in the system buffer and loss probability. However, when this difference is small, IP performs very similar to MPLS. In addition to that when MPLS have higher arrival rate due MPLS trees establishment control overhead, it would perform worst than IP.

Taking the same values of ξ_1 and ξ_2 for all priority classes; implicitly assumes a shared tree. In the near future different values for ξ_1 and ξ_2 for different priority classes (source trees) can be assumed. Also, in the coming future, ARQ/FEC can be implemented in our analytical model or similar model to ensure a guaranteed delivery of multicast packets.

References:

[1] B. Quinn et all, " IP Multicast Applications: Challenges and Solutions ", *RFC 3170*, September 2001; http://www.ietf.org/rfc/rfc3170.txt .

[2] E. Rosen, A. Viswanathan and R. Callon," Multiprotocol Label Switching Architecture ", *RFC 3031*, January 2001; http://www.ietf.org/rfc/rfc3031.txt

[3] D. Ooms et all, " Framework for IP multicast in MPLS ", *IETF Draft, draft-ietf-mpls-multicast-07.txt*, January 2002.

[4] D. Ooms and W. Livens, " IP Multicast in MPLS Networks ", *Proceedings of the IEEE Conference on High Performance Switching and Routing*, June 2000, pp. 301 – 305.

[5] J. Wrocklawski, " The Use of RSVP with IETF Integrated Services ", RFC 2210, September 1997; http://www.ietf.org/rfc/rfc2210.txt

[6] M. Carlson et al., " An Architecure for Differentiated Services ", *RFC 2475*, December 1998; http://www.ietf.org/rfc/rfc2475.txt

[7] T. Saadawi, M. Ammar and A. Elhakeem, " *Fundamentals of Telecommunication Networks* ", New York, NY: John Wiley & Sons Inc