Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Information Processing Letters 90 (2004) 73–80

Information Processing Letters

www.elsevier.com/locate/ipl

Packet loss probability for DiffServ over IP and MPLS reliable homogeneous multicast networks ☆

Abdullah AlWehaibi [a,*], Michael Kadoch [b], Anjali Agarwal [a], Ahmed ElHakeem [a]

[a] Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada
[b] Department de Genie Electrique, Ecole de Technologie Supérieure, Université du Quebec, Montreal, Canada

Received 9 May 2003

Communicated by F. Dehne

Abstract

Multicasting has become increasingly important with the emergence of Internet-based applications such as 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 order to provide QoS in group communications for real time applications such as video conferencing, reliable multicasting is used. Miscellaneous efforts have been undertaken to provide reliability on top of IP multicast. Two error control strategies have been popular in practice. These are the FEC (Forward Error Correction) strategy, which uses error correction alone, and the ARQ (Automatic Repeat Request) strategy, which uses error detection, combined with retransmission of data.

In this paper, we present a new fair share policy (FSP) that utilizes Differentiated Services to solve the problems of QoS and congestion control when reliable ARQ multicast is used. The results should provide insight into the comparisons of the residual packet loss probability between IP multicast in MPLS networks using FSP and plain IP multicasting using the same policy when DiffServ are adopted and when reliable ARQ multicast is considered.

Keywords: Interconnection networks; Packet loss; Multicast; IP; MPLS

1. Introduction

Multicasting has been at the center of interest in the area of Internet activities and has already attained 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 boost the packet forwarding performance.

Recent developments in Multiprotocol Label Switching (MPLS) open new possibilities to address some of the limitations of IP systems. MPLS is an In-

☆ Work supported by NSERC/BELL Canada, Joint CRD grant program.

* Corresponding author.
E-mail addresses: aa_alweh@ece.concordia.ca (A. AlWehaibi), ahmed@ece.concordia.ca (A. ElHakeem).

0020-0190/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2004.01.008

ternet 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 [2–4].

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

Reliable multicasting is used to provide QoS in group communications for real time multimedia applications such as video conferencing. Two main error control strategies are well known. These are the FEC (Forward Error Correction) strategy, which uses error correction alone, and the ARQ (Automatic Repeat Request) strategy, which uses error detection, combined with retransmission of repair data [7–9].

In ARQ strategy, when an error is detected at the receiver, a request (NAK) is transmitted to the sender to repeat the incorrect message, and this continues until the message is received correctly. ARQ can be divided into two types: stop-and-wait ARQ and Continuous ARQ which can be further divided into two subtypes: go-back-N ARQ and selective-repeat ARQ. In our work, we will use selective repeat ARQ. When reliable multicasting is used, there is a scalability problem to accommodate arbitrarily large groups of receivers where each receiver would be sending an acknowledgment to the sender, which in case of large group could easily result in *feedback implosion* problem. This problem can be solved by allowing the receivers to send NAKs only in case of errors or lost data. There are two methods to send the repair packets from the sender to the receiver or group of receivers:

(1) *Multicast repairs*: In case of receiving a NAK from one or more receivers the sender multicast again the repair packet to all receivers.
(2) *Unicast repairs*: With unicast repairs, if the sender received a NAK from one or more receivers, it resends the repair packet to only the receivers who did not receive the packet correctly in a unicast manner.

The multicast repairs method is simpler than the unicast repairs method and requires less overhead; however the multicast repairs methods consumes much more bandwidth. In our work we will evaluate the performance of the ARQ with multicast repairs only.

In this paper, we compare QoS performance of IP and MPLS multicasting, given their particular constraints [10]. 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. We present a new fair share policy and by taking the above constraints into consideration, we evaluate the QoS performance in terms of residual packet loss probability for a typical binary tree in the two cases of IP and MPLS multicasting. We also consider Differentiated Services; i.e., traffics with different priority classes when reliable ARQ multicast is used. Analysis tools will be used to evaluate our fair share policy (FSP) for different homogeneous network scenarios.

2. The analytical model underlying 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

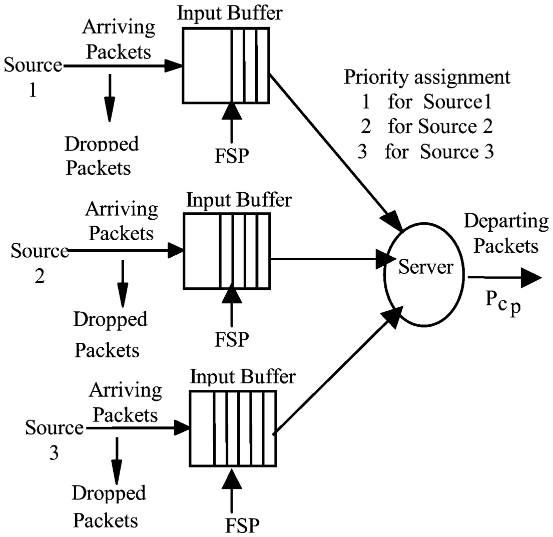


Fig. 1. The analytical model.

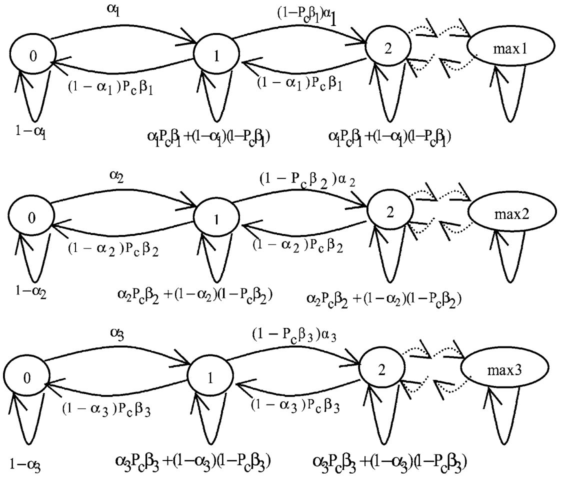


Fig. 2. The coupled state diagrams.

priorities. Moreover, FSP guarantees fairness among flows having the same priority (i.e., required QoS) in buffer space allocated to lower priority traffic is larger; thus leading to less packet discard [10]. 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 interarrivals.
(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 $\overline{E_1}(n)$, $\overline{E_2}(n)$ and $\overline{E_3}(n)$ for each source respectively.
(6) Maximum buffer sizes are $\max_1$, $\max_2$ and $\max_3$ for each source respectively.
(7) Total system buffer size:

$$B = \max_1 + \max_2 + \max_3,$$

where $\max_p$, $p = 1, 2, 3$, is calculated as:

$$\max_p = \frac{\Pr_p}{\sum_p \Pr_p} * B,$$

where $\Pr_p$ is source p priority.
(8) All of MPLS or IP routers on the subject Internet are homogeneous in providing resource and traffic conditions.
(9) All packets are of the same length.
(10) Steady state conditions prevail such that the distribution of the number of packets in the queue will not change with time and hence $E_1(n)$ for source 1 for example will be taken as a representative figure of the actual number in the queue n_1.

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 1 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 will be served only when the buffers corresponding to source 1 and source 2 (which have higher priority) are all empty. Pc_p is the probability of successful delivery to next router for priority p traffic ($p = 1, 2, 3$).

Packet loss probability for each source can be obtained by calculating the probability to be in last stage in the state diagram $P_{\max_1}$, $P_{\max_2}$ and $P_{\max_3}$ respectively.

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

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

where Δ_1 is the processing time at lower layers (for example MAC layer), Δ_2 is the processing time at IP layer and τ is the IP processing time factor (or processing factor).

α_1^1 is the intrinsic arrival probability, 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:

$$\alpha_1 = \tau\alpha_1^1 + \xi_1\alpha_1^1, \quad \xi_1 = \frac{\alpha_1^2}{\alpha_1^1}, \tag{2}$$

where ξ_1 is the IP control overhead factor (or IP factor).

Similarly for MPLS based networks, α_1 can be written as:

$$\alpha_1 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3, \tag{3}$$

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:

$$\alpha_1 = (1 + \xi_1 + \xi_2)\alpha_1^1, \qquad \xi_2 = \frac{\alpha_1^3}{\alpha_1^1}, \tag{4}$$

where ξ_2 is the MPLS control overhead factor (or MPLS factor).

By writing the balance equations for the state diagrams in Fig. 2 [11,12], and solving these equations to find the probabilities. In order to write the equations in simpler forms we define:

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

$$\sigma = \alpha P_c\beta + (1-\alpha)(1 - P_c\beta), \tag{5}$$

$$P_1 = \frac{\alpha}{\mu}P_0, \tag{6}$$

$$P_2 = \frac{1-\sigma}{\mu}P_1 - \frac{\alpha}{\mu}P_0 = \left[\frac{(1-6)\alpha}{\mu^2} - \frac{\alpha}{\mu}\right]P_0, \tag{7}$$

$$P_n = \frac{1-\sigma}{\mu}P_{n-1} - \frac{\lambda}{\mu}P_{n-2} \quad \text{for } n = 3, 4, \ldots, \max_p . \tag{8}$$

Eq. (8) can be rewritten as:

$$P_{n+2} = \frac{1-\sigma}{\mu}P_{n+1} - \frac{\lambda}{\mu}P_n \quad \text{for } n = 3, 4, \ldots, \max_p . \tag{9}$$

Define $m = (1-\sigma)/\mu$ and $q = \lambda/\mu$.

Eq. (9) is the 2nd order homogeneous difference equation [13], which has the general form:

$$P_{n+2} + 2aP_{n+1} + bP_n = 0, \tag{10}$$

where $a = -m/2$ and $b = q$, the general solution of Eq. (10) is of the form [13]:

$$P_n = Ar_1^n + Br_2^n, \tag{11}$$

where r_1 and r_2 are the distinct roots of the Eq. (10); A and B are constants. The characteristic equation of Eq. (10) is:

$$r^2 - mr + q = 0 \tag{12}$$

which has the solution:

$$r_1 = \frac{m + \sqrt{m^2 - 4q}}{2}, \qquad r_2 = \frac{m - \sqrt{m^2 - 4q}}{2}.$$

The initial conditions for the set of equations are P_1 and P_2. Using Eq. (11), we write:

$$P_1 = Ar_1 + Br_2 = kP_0, \tag{13}$$

$$P_2 = Ar_1^2 + Br_2^2 = \omega P_0, \tag{14}$$

where

$$\omega = \left(\frac{(1-\sigma)\alpha}{\mu^2} - \frac{\alpha}{\mu}\right) \quad \text{and} \quad k = \frac{\alpha}{\mu}.$$

Substituting for r_1 and r_2 and solving Eqs. (13) and (14) together to find A and B, we obtain:

$$B = \left(\frac{\omega - kr_1}{r_2^2 - r_1r_2}\right)P_0 \quad \text{and} \quad A = \left(\frac{\omega P_0 - Br_2^2}{r_1^2}\right).$$

In order to find nth probability P_n, our solution for Eq. (11) can be written as:

$$P_n = \left(\frac{\omega P_0 - Br_2^2}{r_1^2}\right)\left(\frac{m + \sqrt{m^2 - 4q}}{2}\right)^n + \left(\frac{\omega - kr_1}{r_2^2 - r_1r_2}\right)P_0\left(\frac{m - \sqrt{m^2 - 4q}}{2}\right)^n. \tag{15}$$

Taking into account that $\sum_{n=0}^{\max_p} P_n = 1$, P_0 can be found using the following equation:

$$\begin{aligned} P_0 &= \frac{1}{1+\sum_{n=1}^{\max_p} P_n} \\ &= \frac{1}{1+k+\omega+Ar_1^3+Br_2^3+\cdots+Ar_1^{\max_p}+Br_2^{\max_p}} \\ &= \frac{1}{1+k+\omega+Ar_1\frac{1-r_1^{\max_p -2}}{1-r_1}+Br_2\frac{1-r_2^{\max_p -2}}{1-r_2}}. \qquad (16) \end{aligned}$$

Therefore, the solution of probability of steady state of the number of packets in the buffer is now given by Eq. (15). The expected number of packets in the buffer for a specific source p can be found as:

$$\begin{aligned} E_p(n) &= \sum_{n=0}^{\max_p} n * P_n \\ &= 1 * k * P_0 + 2 * \omega * P_0 \\ &\quad + \sum_{n=3}^{\max_p} n * \left(Ar_1^n + Br_2^n\right). \qquad (17) \end{aligned}$$

Notice that the packet loss probability for source p is equal to the probability to be in last stage of the state diagram:

$$P_{Lp} = P_{\max_p} \quad (p = 1, 2, 3). \qquad (18)$$

The same solution above applies to state diagrams in Figs. 2(middle) and 2(bottom) as well except that in Fig. 2(middle) $\alpha = \alpha_2$, $\beta = \beta_2$ and $\max = \max_2$ and that in Fig. 2(bottom) $\alpha = \alpha_3$, $\beta = \beta_3$ and $\max = \max_3$.

One more assumption is added for reliable multicast that we use a complete binary tree, where each parent router has two children routers until we reach leafs. Fig. 3 shows an example of a complete binary tree with the root, which is the nearest router to the sender or the rendezvous point, and the leafs, which are the routers with receivers underneath them. As shown in the figure the depth of this tree is 4 and the total number of routers is 15.

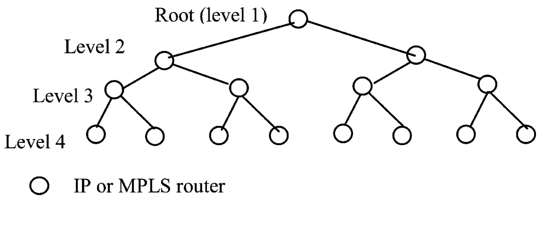


Fig. 3. A complete homogeneous binary multicast tree.

3. Reliable ARQ multicast repairs

Pc_p which is the probability of successful delivery to next router for certain priority traffic, would be given as:

$$\mathrm{Pc}_p = (1 - \mathrm{Po}_p - \mathrm{Pe}_p)^L, \quad p = 1, 2, 3. \qquad (19)$$

Po_p is the byte overflow for a certain priority traffic which can be obtained by dividing the packet overflow probability in Eq. (18) by packet length (L). Pe_p is the byte error probability for a certain priority traffic p and it is assumed to be equal to Po_p.

In Eq. (19), two assumptions are made:

(1) Packet loss of source packet is caused by consecutive byte losses at the intermediate routers.
(2) Interleaving is used in order to break byte burst losses and efficiently turn them independent random byte losses at the source and destination [9].

Probability of no packet loss for certain priority traffic is given by:

$$\begin{aligned} P_{\text{no packet loss}_p} &= (1 - \mathrm{Po}_p)^L \\ &\cong 1 - \mathrm{LPo}_p \\ &\text{for small values of } \mathrm{Po}_p,\ p = 1, 2, 3. \end{aligned}$$

Therefore, probability of packet loss for small values of Po_p and for certain priority traffic p can be expressed as:

$$P_{\text{packet loss}_p} = 1 - P_{\text{no packet loss}_p} = \mathrm{LPo}_p,$$

$$\mathrm{Po}_p = \frac{P_{\text{packet loss}_p}}{L}, \quad p = 1, 2, 3.$$

In this case upon the receipt of a NAK from one or more receivers, the sender multicast again the repair packet to all receivers. Due to the use of ARQ multicast repairs, the intrinsic arrival probability α_p^1 for certain priority traffic p would increase according to:

$$\alpha_p^{1'} = \alpha_p^1(1 + F_p), \quad p = 1, 2, 3. \qquad (20)$$

F_p is the number of failures for certain priority traffic p and N is the total number of routers in the multicast tree. This increase in the intrinsic arrival probability is due to that every router in the whole network receives a copy of each repair packet. The Probability of success for worst case scenario for certain priority traffic p is given as:

$$\begin{aligned}\mathrm{Ps}_p w &= \text{Probability of success}\\ &= \mathrm{Pc}_p^N \text{ worst case.}\end{aligned} \tag{21}$$

Eq. (21) represents an upper bound for worst case scenario of probability of success when ARQ multicast repairs method is used. However, using ARQ multicast repairs have a better chance of success with each trial since the number of receivers who did not receive the packet correctly decreases with each trial. Therefore, the average probability of success for a certain priority p packet in a typical transmission multicast trial from sender can be calculated as:

$$\mathrm{Ps}_p \,\mathrm{avg} = \frac{\mathrm{Pc}_p^N + \mathrm{Pc}_p^{(N/2)+1} + \cdots + \mathrm{Pc}_p^{(N/2^D)+D-1}}{D}, \tag{22}$$

where D is the network depth. If the packet does not suffer loss or error on any of the N routers of the multicast tree, with probability Pc_p^N no further repair is needed, this explains the first term of Eq. (22). However, if there has been an error or loss which located at level 1 (see Fig. 3), then the repair packet, then the repair packet would be sent from sender to the router at level 1, and then the repair packet will flow to $N/2$ routers under level 1. All such $(N/2)+1$ transmissions of repair packet have to be correct, otherwise further repair is needed and so on. The probability of these $(N/2)+1$ correct transmissions of subject repair packet is given $\mathrm{Pc}_p^{(N/2)+1}$ and so on for the remaining terms in Eq. (22).

We divide by D (network depth) because we assume that errors are equally likely to occur on different levels of the tree giving rise to the addition of different terms (Eq. (22)) and the division by the depth D where: $D = \log_2(N+1)$.

A more accurate expression for Ps_p was evaluated in [14]. However, results of [14] shows that of Eq. (22) is very close to the exact value over a wide range of Pc_p and N.

The total number of ARQ trials T_p for specific priority traffic p can be expressed as:

$$T_p = \mathrm{Ps}_p + 2\mathrm{Ps}_p(1-\mathrm{Ps}_p) + 3\mathrm{Ps}_p(1-\mathrm{Ps}_p)^2 + \cdots. \tag{23}$$

Therefore, the number of failures (or retransmissions only) for certain priority traffic p can be given as:

$$F_p = T_p - 1, \tag{24}$$

where Ps_p is the average probability of packet success for priority p traffic corresponding to one ARQ trial.

Defining Ps'_p as the final probability of success for priority p traffic:

$$\begin{aligned}\mathrm{Ps}'_p &= \mathrm{Ps}_p + (1-\mathrm{Ps}_p)\mathrm{Ps}_p + \cdots + \mathrm{Ps}_p(1-\mathrm{Ps}_p)^{T_p-1}\\ &= 1-(1-\mathrm{Ps}_p)^{T_p}.\end{aligned} \tag{25}$$

Eq. (25) is for (T_p) trials of a typical packet to the multicast tree; Where we note that for one trial $\mathrm{Ps}'_p = \mathrm{Ps}_p$ and for infinite retransmission trials $\mathrm{Ps}_p = 1$ as it should be. Therefore, the residual loss (after all ARQ trials) is given by:

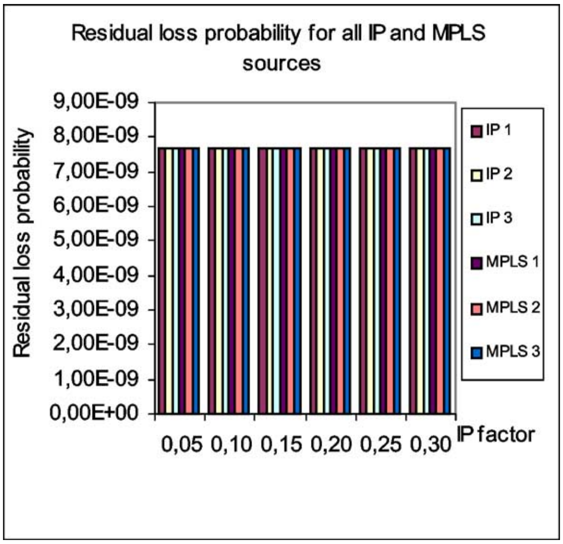
$$\mathrm{Ploss}_p = 1 - \mathrm{Ps}'_p. \tag{26}$$

4. Analysis results

Figs. 4 to 6 show the performance comparisons between IP sources and MPLS sources in the multicast tree when ARQ multicast repairs mechanism is applied. Fig. 4 shows the residual packet loss probability for all sources for both IP and MPLS versus IP factor (ξ_1) for small processing factor (τ). It shows that IP and MPLS sources have very same residual packet loss probability (almost zero).

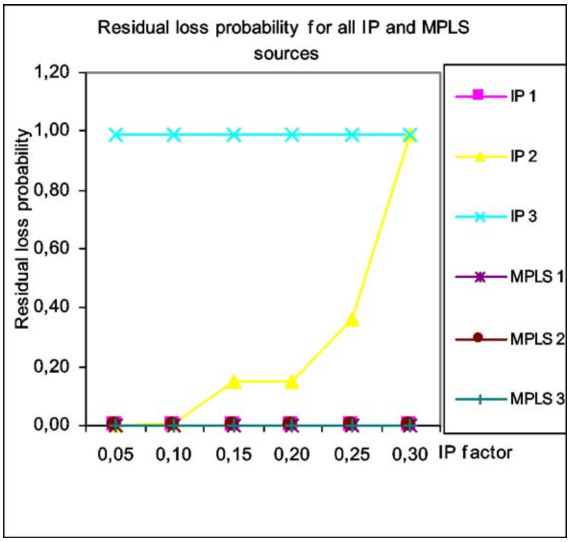
However, Fig. 5 shows that when the processing factor (τ) increases MPLS will have superiority over IP in terms of the residual packet loss probability. As shown in Fig. 5 the residual packet loss probability in case of MPLS (which is zero) is less than IP for all sources and this difference is clear for low priority sources 2 and 3.

In Figs. 4 and 5 MPLS factor was constant and relatively small; explaining why MPLS performance was better or very similar to IP performance. However, in the following figure we will study the effect of MPLS factor (ξ_2) on MPLS performance. Fig. 6 shows



$\alpha_1^1 = 0.2,\ \alpha_1^2 = 0.15,\ \alpha_1^3 = 0.1,\ \beta_1 = 1,\ T_p = 2,\ D = 4,$
$B = 30,\ \xi_2 = 0.1,\ \tau = 1.2,\ L = 500$

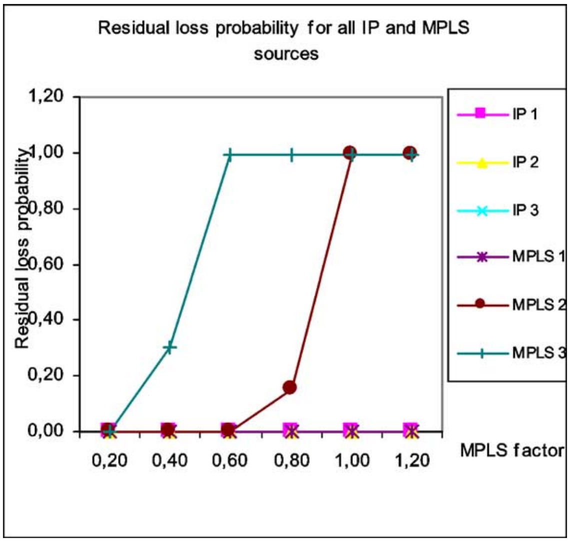
Fig. 4. Residual loss probability versus IP factor (small τ).



$\alpha_1^1 = 0.25,\ \alpha_1^2 = 0.2,\ \alpha_1^3 = 0.15,\ \beta_1 = 1,\ T_p = 2,\ D = 4,$
$B = 30,\ \xi_2 = 0.1,\ \tau = 1.8,\ L = 500$

Fig. 5. Residual loss probability versus IP factor (large τ).

that IP will be superior over MPLS when MPLS factor increases. As shown in Fig. 6 the residual packet loss probability in the case of IP (which is zero) is less than MPLS for all sources and this difference is clear for low priority sources 2 and 3. 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 especially for low priority traffics and when the intrinsic traffics increase.



$\alpha_1^1 = 0.2,\ \alpha_1^2 = 0.15,\ \alpha_1^3 = 0.10,\ \beta_1 = 1,\ T_p = 2,\ D = 4,$
$B = 30,\ \xi_1 = 0.2,\ \tau = 1.2,\ L = 500$

Fig. 6. Residual loss probability versus MPLS factor.

5. Conclusions and future work

In this paper, a performance comparison between IP multicast trees and MPLS multicast trees 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 and ARQ reliable multicasting are used in this comparison. In this paper, we found that when the difference in packet processing time (τ) between IP and MPLS is high and when MPLS factor is small, IP multicast will perform less efficiently than MPLS in terms of residual packet loss probability. However, when this difference in packet processing time is small IP performs very similar to MPLS. In addition to that when MPLS has higher arrival rate due to MPLS trees establishment control overhead and when the processing factor is small, IP would perform better than MPLS.

Analysis results revealed that there is a noticeable improvement in QoS defined as the residual packet loss probability for a higher priority traffic when MPLS multicasting replaces IP multicasting especially if MPLS factor is small and processing factor is large.

In addition to that, the study finds that when applying the ARQ multicast repairs mechanism, there would be a noticeable improvement in terms of the residual packet loss probability which enhances the reliability of multicasting for both IP and MPLS trees.

The routers in the network could be identical in their capabilities (homogeneous network) or different (heterogeneous network).

Each router may have different capabilities; for example one router could have the ability to correct errors (FEC) and use ARQ, one may use only ARQ but cannot correct errors, a third one may not have MPLS capability. In this paper, the study carried only homogeneous networks. In the near future, heterogeneous networks would be considered.

References

[1] B. Quinn, et al., IP multicast applications: challenges and solutions, RFC 3170, 2001, http://www.ietf.org/rfc/rfc3170.txt.

[2] E. Rosen, A. Viswanathan, R. Gallon, Multiprotocol label switching architecture, RFC 3031, 2001, http://www.ietf.org/rfc/rfc3031.txt.

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

[4] D. Ooms, W. Livens, IP multicast in MPLS networks, in: Proceedings of the IEEE Conference on High Performance Switching and Routing, 2000, pp. 301–305.

[5] J. Wrocklawski, The use of RSVP with IETF integrated services, RFC 2210, 1997, http://www.ietf.org/rfc/rfc2210.txt.

[6] M. Carlson, et al., An architecture for differentiated services, RFC 2475, 1998, http://www.ietf.org/rfc/rfc2475.txt.

[7] S. Lin, D. Costello, Error Control Coding: Fundamentals and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[8] B. Li, Reliable multicast transmissions using forward error correction and automatic retransmission requests, in: Canadian Conference on Electrical and Computer Engineering, 2001, vol. 2, 2001, pp. 1145–1150.

[9] S. Wicker, Error Control Strategies—for Digital Communication and Storage, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[10] A. AlWehaibi, A. Agarwal, M. Kadoch, A. ElHakeem, Performance comparison between MPLS multicast and IP multicast using fair share policy and priority buffers, in: IASTED CSN 2002, Spain, 2002.

[11] T. Saadawi, M. Ammar, A. Elhakeem, Fundamentals of Telecommunication Networks, Wiley, New York, 1994.

[12] L. Kleinrock, Queuing Systems, vols. I and II, Wiley, New York, 1975.

[13] D. Jordan, P. Smith, Mathematical Techniques, Oxford University Press, Oxford, 1994.

[14] A. AlWehaibi, M. Kadoch, A. ElHakeem, Computation of the residual packet loss probability in a binary multicast tree, in: IEEE Canadian Conference on Electrical and Computer Engineering, CCECE 03, Montreal, Canada, 2003.