

International Journal of Computer Networks (IJCN)



ISSN : 1985 - 4129

Volume 2 - Issue 2
Number of issues per year: 6

Copyright © 2010 Computer Science Journals. All rights reserved.

Editor in Chief Associate Professor Min Song

International Journal of Computer Network (IJCN)

Book: 2010 Volume 1, Issue 2

Publishing Date: 31-05-2010

Proceedings

ISSN (Online): 1985-4129

This work is subjected to copyright. All rights are reserved whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication of parts thereof is permitted only under the provision of the copyright law 1965, in its current version, and permission of use must always be obtained from CSC Publishers. Violations are liable to prosecution under the copyright law.

IJCN Journal is a part of CSC Publishers

<http://www.cscjournals.org>

© IJCN Journal

Published in Malaysia

Typesetting: Camera-ready by author, data conversion by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Table of Contents

Volume 2, Issue 2, May 2010.

Pages

- 77 - 100 A New Location Caching with Fixed Local Anchor for Reducing Overall Location Management Cost in Wireless Mobile Networks
Md. Mohsin Ali, G. M. Mashrur-E-Elahi, Md. Asadul Islam
- 115 - 131 Dynamic Hybrid Topology Design for Integrated Traffic Support in WDM Mesh Networks (Invited Paper)
Mina Youssef, Baek-Young Choiz, Caterina Scoglio, Eun Kyo Park
- 132 - 139 Efficient and Fair Bandwidth Allocation AQM Scheme for Wireless Networks
Rafe Alasem
- 140 - 151 A Stochastic Model Approach for Reaching Probabilities of Message Flow in Space-Division Switches
D. Shukla, Rahul Singhai, Surendra Gadewar , Saurabh Jain, Shewta Ojha, P.P. Mishra
- 101 - 114 Traffic Engineering in Metro Ethernet
Padmaraj Nair, Suku Nair, Girish Chiruvolu

A New Location Caching with Fixed Local Anchor for Reducing Overall Location Management Cost in Wireless Mobile Networks

Md. Mohsin Ali

*Department of Computer Science and Engineering (CSE)
Khulna University of Engineering & Technology (KUET)
Khulna, 9203, Bangladesh*

mohsin_kuet_cse@yahoo.com

G. M. Mashrur-E-Elahi

*Department of Computer Science and Engineering (CSE)
Khulna University of Engineering & Technology (KUET)
Khulna, 9203, Bangladesh*

ranju2k4cse_kuet@yahoo.com

Md. Asadul Islam

*Department of Computer Science and Engineering (CSE)
Khulna University of Engineering & Technology (KUET)
Khulna, 9203, Bangladesh*

asad_kuet@yahoo.com

Abstract

The proposed approach in this paper selects a fixed Visitor Location Register (VLR) as a Fixed Local Anchor (FLA) for each group of Registration Areas (RAs). During call delivery process, the calling VLR/FLA caches are updated with the called Mobile Terminal's (MT's) location information and the called VLR and FLA caches are updated with the calling MT's location information. Furthermore, the FLA and the old VLR caches are updated with MT's new location information during inter-RA handoff as a part of informing this to the FLA of that region. But for another case, it updates the new FLA, old FLA, and old VLR caches with new location information together with directly informing this to the Home Location Register (HLR). This location caching policy in local anchor strategy maximizes the probability of finding MTs' location information in caches. As a result, it minimizes the total number of HLR access for finding MT's location information prior to deliver a call. So, it significantly reduces the total location management cost in terms of location registration cost and call delivery cost. The analytical and experimental results also demonstrate that the proposed method outperforms all other previous methods regardless of the MT's calling and mobility pattern.

Keywords: location management, location registration, call delivery, cache, fixed local anchor.

1. INTRODUCTION

Wireless mobile networks and personal communication networks provide services to its subscribers that travel within the network coverage area. In order to correctly deliver incoming calls to MTs within this area, the up-to-date location information of each MT needs to be identified. Therefore, a location management strategy is necessary to effectively keep track of the MTs before initiating the call setup procedure. The basic operations of this strategy are location

registration, call delivery, and paging or searching. The first one is the process of informing the network about the MT's current location information; the second one is the connection establishment between the caller and called MTs, and the third one is the operation of determining the location of the MT. It is also observed that there are some trade-offs among the location registration, paging, and call arrival rate. If the MT registers its location during RA crossing, the network can precisely maintain its location and prevents the need for paging. However, in the case of lower call arrival rate, the network resources are wasted for processing frequent update information and the MT wastes its power by transmitting the update signal. On the other hand, a large coverage area has to be searched during call arrival process if the MT performs infrequent location registration - which eventually wastes the radio bandwidth [1]. Therefore, the central problem of location management is to develop algorithms and architectures with a view to minimizing the overall cost for location registration, call delivery, and paging.

Two standard architectures currently exist in wireless mobile networks are IS-41 [2] and GSM [3]. Both these architectures are based on a two-level database hierarchy. Two types of database called HLR and VLR are used to store the MTs' location information. Figure 1 shows the basic architecture of the wireless mobile networks under this two-level database hierarchy. The whole network coverage area is divided into cells having same size and shape. Each cell has a Base Transceiver Station (BTS) to communicate with the network through wireless link. The cells are grouped together to form larger areas called Registration Areas (RAs). All the BTS belonging to a given RA are wired to a Mobile Switching Center (MSC) which plays the role of the interface between the wireless and the wired portions of the network. In Figure 1, it is assumed that each VLR co-locates with the MSC and a group of RAs are interfaced with the Local Signaling Transfer Point (LSTP) following HLR. There may exist one or more HLRs in the network depending on its configuration. An HLR is the centralized database that contains the records of all users' profiles together with MTs' location information for the entire network. On the other hand, each VLR stores replications of the user profiles of the subscribers currently residing in its corresponding RA. Whenever an MT enters into a new RA, it needs to report its new location information to the MSC. The MSC updates its associated VLR and transmits this new information to the HLR. The HLR acknowledges the MSC about this successful registration and also sends back a location deregistration message to the MT's old VLR in the corresponding RA. In order to deliver a call to a target MT, the HLR is queried to determine the serving MSC of that target MT. The HLR then sends a message to this MSC with a view to determining the serving BTS of the target MT by searching all cells within the corresponding RA [5].

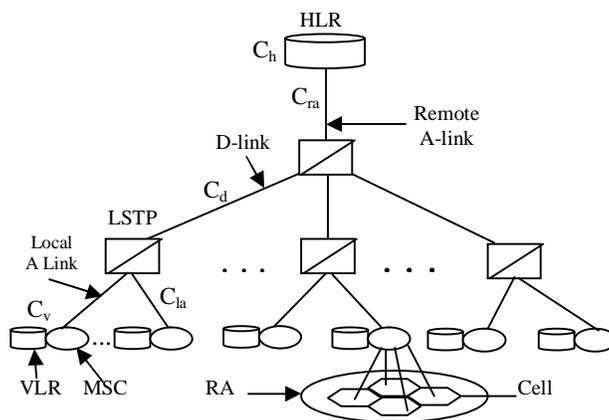


FIGURE 1: Signaling System No. 7 architecture.

As the number of MTs within the network is exploding day by day, location management under the IS-41 has suffered various critical problems like increasing signaling traffic in the network and the HLR bottleneck. As a result, a number of efforts have been reported to overcome these

problems [4], [5], [6], [7], [8], and [9]. However, all of these approaches have some specific working criteria. As for example, the location caching strategy [4] or replication strategy [8] effectively works for larger call-to-mobility ratio (CMR), while the local anchor strategy [5], [6] or forwarding-pointer strategy works better for small CMR values. A profile-based location caching with fixed local anchor strategy is proposed in [9], but suffers signaling overheads throughout the network during the MT's inter-LSTP movement.

Symbol	Description
()	Corresponding message number
[]	Cost for the particular signaling exchange
{ }	Cost for accessing the particular database
→	Exchange of the particular signaling message
←	Acknowledgement of the corresponding signaling message

TABLE 1: Description of Symbols Shown in Figure 2–Figure 10.

In this paper, a new location caching strategy is proposed by effectively using the MTs' calling and mobility pattern and combined with the local anchor strategy [5], [6]. Simply, it effectively exploits the concepts of both the location caching strategy [4] and the local anchor strategy [5], [6]. It also relieves the network from signaling overheads during inter-LSTP movement unlike the profile-based location caching with fixed local anchor strategy [9]. In the general location caching strategy [4], the called MT's location information is updated only in the call originating VLR cache at call originating time. This updating strategy can also be applied to the local anchor strategy [5], [6]. But, there is a scope of updating the called MT's VLR and FLA caches with the calling MT's location information for the same call too. On the other hand, location deregistration messages are sent to the old VLR and FLA when an MT performs an inter-RA or inter-LSTP movement. The old VLR and FLA caches can also be updated with MT's new location information together with the deregistration message. Moreover, there is another scope of updating new FLA cache with MT's new location information during inter-LSTP movement. This enhanced cache updating policy prepares the cache with up-to-date information frequently as it updates more than one cache for each call delivery and even updates them during location registration. So, the probability of searching the HLR decreases for call delivery and there is also no location update to the HLR for inter-RA handoff. As a result, the total location management cost in terms of location registration cost and call delivery cost decreases.

The rest of the paper is organized as follows. Section 2 provides an overview of the related recent research work. Proposed approach is described in Section 3. Section 4 gives the analytical modeling. Numerical results and comparison among different methods based on some experimental results are described in Section 5. Section 6 provides a concluding remark of this paper.

2. EXISTING LOCATION MANAGEMENT STRATEGIES

An extensive work has been done on location management to reduce the overall location management cost in terms of location registration cost and call delivery cost [2], [4], [5], [6], [7], [8], and [9]. Some of them are basic scheme which are generally used to manage the location irrespective of all the wireless networks. Some others are based on reusing the user location information obtained during the previous call to the user. This effectively reduces the call delivery cost. While some others are based on managing the local handoff locally instead of informing the centralized HLR. This reduces the location registration cost. There are also some methods which

use the MT's calling statistics from the HLR and replicate its location information to these calling VLR cache. These also manage the local handoff locally instead of accessing the heavily congested HLR. This reduces both the location registration and call delivery cost. The existing location management strategies are shown in Figure 2– Figure 8 and the symbols used in these figures are described in Table 1. The location management procedures of these strategies are described in the following subsections.

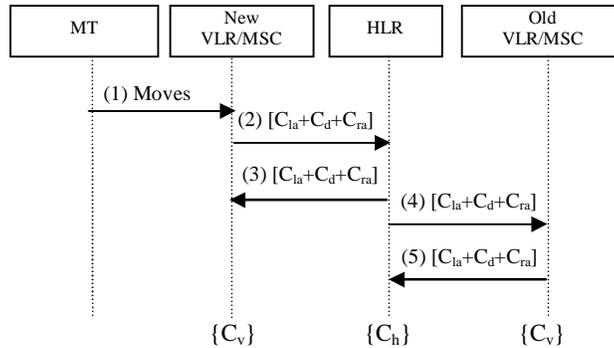


FIGURE 2: Location Registration under Both the IS-41 and LC Strategies.

2.1. IS-41 Strategy

The basic IS-41 standard is proposed in [2] where each MT informs its location information to the HLR during all type of handoff procedures. The call delivery procedure is performed by searching the MT's location information in the HLR prior to setup a call.

The location registration procedure of this strategy is described as follows (see Figure 2).

- (1) An MT handoffs into a new RA and informs its new location to the new MSC through the nearby BTS.
- (2) The MSC updates its associated VLR about this MT and sends a location registration to the HLR.
- (3) The HLR updates the MT's record and sends back a registration acknowledgement message to the new VLR.
- (4) It also sends a registration cancellation message to the old VLR.
- (5) The old VLR removes the record of the MT and sends back a cancellation acknowledgement message to the HLR.

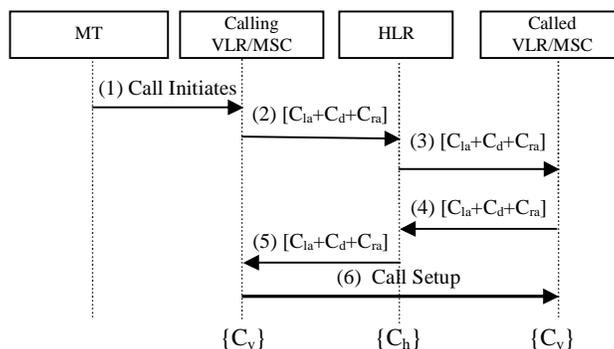


FIGURE 3: Call Delivery under the IS-41 Strategy.

On the other hand, the call delivery procedure under this strategy is described as follows (see Figure 3).

- (1) The calling MT initiates a call and sends a message to its serving MSC through a nearby BTS.
- (2) The calling MSC sends a message to the HLR with a request of the called MT's location information.
- (3) The HLR determines the called MT's current serving MSC and sends a location request message to that MSC.
- (4) The MSC allocates a Temporary Local Directory Number (TLDN) [9] to the MT and sends back a reply to the HLR together with the TLDN.
- (5) The HLR sends this information back to the calling MSC
- (6) The MSC sends a request message of call setup to the called MSC through the network shown in Figure 1.

2.2. Location Caching (LC) Strategy

A per-user location caching strategy is proposed in [4] where the called MT's location information is stored in the calling MT's VLR cache prior to call setup during each call delivery process. So to deliver a call, the called MT's location information is first searched at VLR cache instead of directly going to the HLR. This reduces the frequency of HLR access for delivering calls and eventually reduces the location management cost.

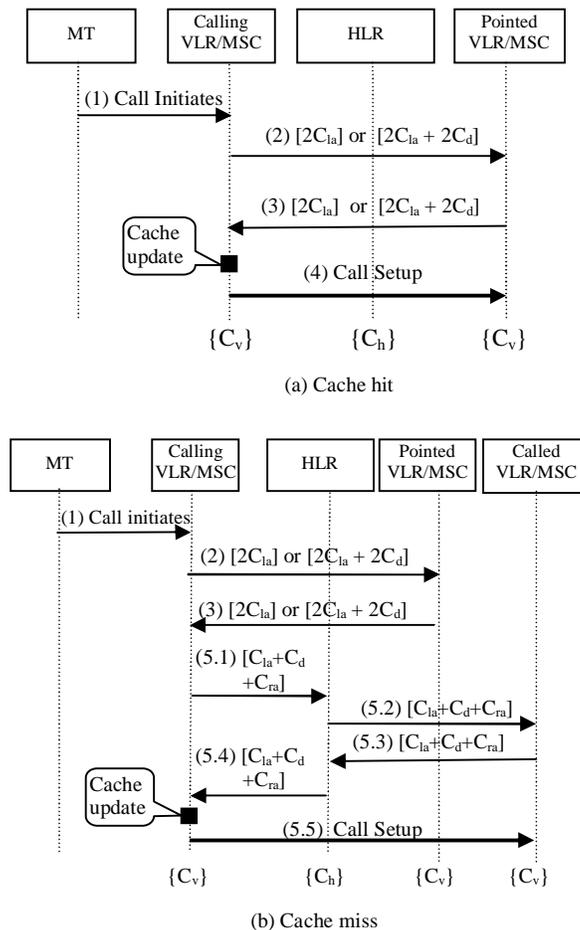


FIGURE 4: Call Delivery under the LC Strategy.

The location registration procedure of this strategy is the same as that of the IS-41 and described in Section 2.1 (see Figure 2). The call delivery procedure under this strategy is described as follows (see Figure 4).

- (1) The MT sends a call initiation message to the calling MSC through the nearby BTS.
- (2) The called MT's location information is searched in the calling VLR cache and it is assumed that there will be such an entry there. Then it sends a route request message to the pointed VLR/MS.
- (3) The pointed VLR verifies whether this information is exact or obsolete. If exact, then sends back an acknowledgement message stating cache hit with a TLDN to the calling MSC. Otherwise, sends a negative acknowledgement message stating cache miss to the calling MSC and go to step 5.
- (4) The calling VLR updates its cache with the called MT's location information. Following this, the MSC sends a call setup message to the called MSC through the network shown in Figure 1 (Call delivery procedure is complete. Do not proceed to the next step).
- (5) If the location information found in the pointed VLR is obsolete (cache miss), then follow the steps.
 - (5.1) The calling MSC sends a location request message to the HLR.
 - (5.2) The HLR determines the current serving MSC of the called MT and sends a location request message to the called MSC.
 - (5.3) The called MSC allocates a TLDN to the MT and sends it back to the HLR.
 - (5.4) The HLR forwards this message back to the calling MSC.
 - (5.5) The calling VLR updates its cache with the called MT's location information and the calling MSC sends a call setup message to the called MSC (Call delivery is complete. Do not proceed to the next step).

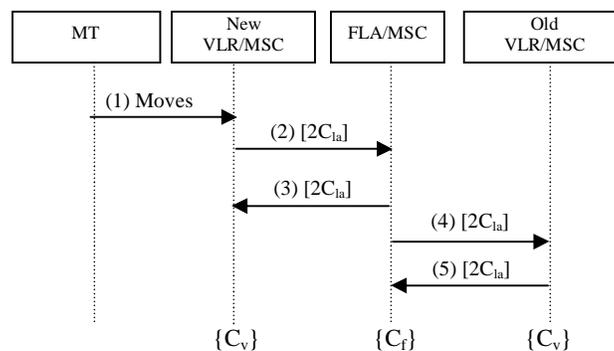


FIGURE 5: Intra-LSTP Movement Location Registration under Both the FLA and PCFLA Strategies.

2.3. Fixed Local Anchor (FLA) Strategy

The Fixed Local Anchor strategy is proposed in [5], [6] where local handoffs are managed locally without overwhelming the centralized HLR. An LSTP group is defined with some of the RAs and a fixed VLR plays the role of the Fixed Local Anchor (FLA) within this group. This FLA handles all the intra-LSTP handoff within this group and HLR tracks the location information of these FLAs.

The location registration procedure is divided into two categories: intra-LSTP movement and inter-LSTP movement. The intra-LSTP movement registration of this strategy is given in Figure 5 and inter-LSTP of that is given in Figure 6. These are described as follows.

- (1) An MT handoffs into a new RA and informs its new location to the new MSC through the nearby BTS.
- (2) The new MSC forwards a location registration message to its associated FLA within its region.
- (3) The FLA verifies the MT's profile. If there is an MT's record, then update it with the new location information and sends back an acknowledgement message together with a copy of the MT's profile to the new MSC. Otherwise go to step 6.
- (4) The FLA sends a deregistration message to the old MSC.

- (5) The old MSC clears the record for that MT from the VLR and replies a confirmation message to the FLA (Location registration by intra-LSTP movement is complete. Do not proceed to the next step).
- (6) If the FLA does not have the MT's record, the followings are performed.
 - (6.1) The serving MSC of the MT's new FLA sends a location registration message to the HLR.
 - (6.2) The HLR updates the MT's record in terms of MT's new FLA and sends back a copy of the MT's profile to that new FLA.
 - (6.3) The FLA updates the MT's record in terms of new VLR and acknowledges the new VLR with a copy of the MT's profile.
 - (6.4) The HLR sends a deregistration message to the MT's old FLA.
 - (6.5) The old FLA removes the MT's record from it and sends a deregistration acknowledgement message back to the HLR.
 - (6.6) The old FLA sends a deregistration message to the MT's old VLR.
 - (6.7) The old VLR removes the MT's record and sends a deregistration acknowledgement message back to the old FLA (Location registration by inter-LSTP movement is complete).

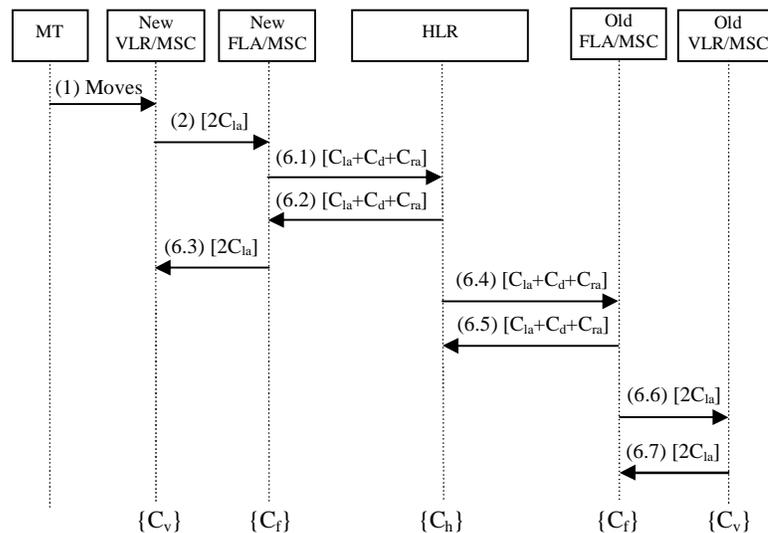


FIGURE 6: Inter-LSTP Movement Location Registration under the FLA Strategy.

On the other hand, the call delivery procedure under this strategy is described as follows (see Figure 8(b)).

- (1) The MT sends a call initiation message to the calling MSC through the nearby BTS.
 - (6.1) The calling MSC sends a location request message to the HLR.
 - (6.2) The HLR sends a message requesting the called MT's location information to the called FLA.
 - (6.3) The called FLA forwards this message to the called VLR/MSC.
 - (6.4) The called VLR/MSC sends a location route back to the HLR together with a TLDN to the MT.
 - (6.5) The HLR forwards this message back to the calling MSC.
 - (6.6) The calling MSC sends a call setup request message to the called MSC through the network shown in Figure 1 (Call delivery is complete).

2.4. Profile-based Location Caching with Fixed Local Anchor (PCFLA) Strategy

A Profile-based Location Caching with Fixed Local Anchor strategy is proposed in [9] where user profiles are effectively utilized to determine at which sites throughout the networks user's location information should be cached. In this approach, these site lists are prepared based on the long-term call related statistics maintained by the HLR from the callee's user profile. These lists are used to store the callees' location information to some of the most frequently calling VLR caches of the corresponding callees. On the other hand, an FLA is used to manage the intra-LSTP handoff locally like FLA strategy described in Section 2.3. The intra-LSTP movement location registration under this strategy is the same as that of the FLA strategy and step 1 to 5 of Section 2.3 describes that (see Figure 5).

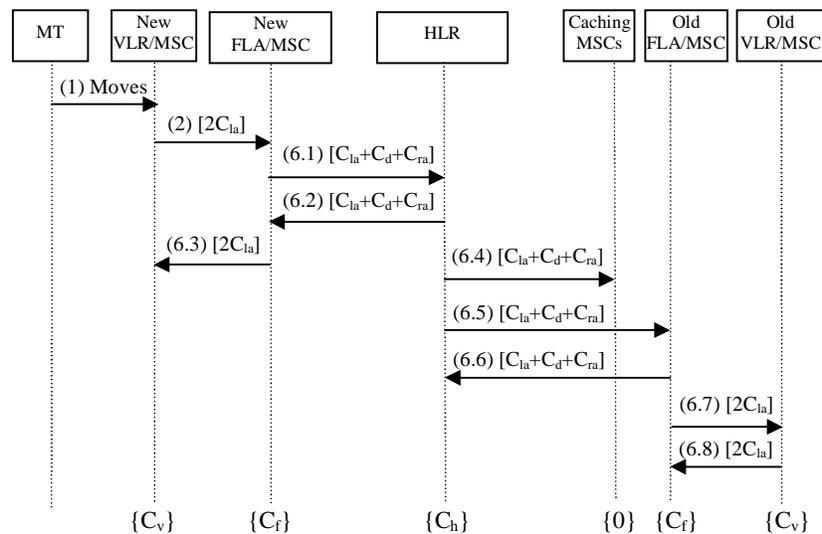


FIGURE 7: Inter-LSTP Movement Location Registration under the PCFLA Strategy.

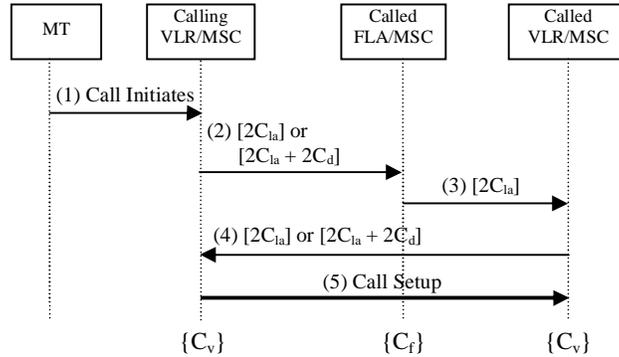
The inter-LSTP movement location registration following the intra-LSTP location registration (step 1 to 5 of Section 2.3) is described as follows (see Figure 7).

- (6) If the FLA does not have the MT's record, the followings are performed.
 - (6.1) The serving MSC of the MT's new FLA sends a location registration message to the HLR.
 - (6.2) The HLR updates the MT's record in terms of MT's new FLA and sends back a copy of the MT's profile to that new FLA.
 - (6.3) The FLA updates the MT's record in terms of new VLR and acknowledges the new VLR with a copy of the MT's profile.
 - (6.4) The HLR sends location cache update messages to MSCs selected by the long-term calling statistics maintained by the HLR which have location caches for that MT.
 - (6.5) The HLR sends a deregistration message to the MT's old FLA.
 - (6.6) The old FLA removes the MT's record from it and sends a deregistration acknowledgement message back to the HLR.
 - (6.7) The old FLA sends a deregistration message to the MT's old VLR.
 - (6.8) The old VLR removes the MT's record and sends a deregistration acknowledgement message back to the old FLA (Location registration by inter-LSTP movement is complete).

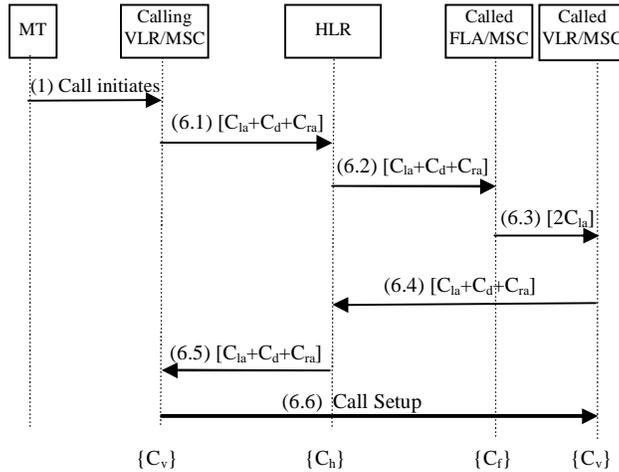
On the other hand, the call delivery procedure under this strategy is described as follows (see Figures 8(a) and 8(b)).

- (1) The MT sends a call initiation message to the calling MSC through the nearby BTS.

- (2) The calling MSC verifies if it has the called MT's location information in the VLR cache. If yes, then it sends a called MT's location request message to the called FLA. Otherwise go to step 6.
- (3) The called FLA forwards this message to the called MSC.
- (4) The called MSC sends an acknowledgement message to the calling MSC together with a TLDN to the MT.
- (5) The calling MSC sends a call setup request message to the called MSC through the network shown in Figure 1 (Call delivery is complete. Do not proceed to the next step).
- (6) If the calling MSC does not contain location cache for the called FLA, then it follows the call delivery steps 6.1 through 6.6 of the FLA strategy described in Section 2.3.



(a) When the calling MSC has location cache for the called FLA



(b) When the calling MSC does not have location cache for the called FLA

FIGURE 8: Call Delivery under the PCFLA (a and b) and FLA (b) Strategies.

3. PROPOSED NEW LOCATION CACHING WITH FIXED LOCAL ANCHOR (NLCFLA) STRATEGY

In the following, a simple but effective location management strategy is proposed for the next-generation wireless mobile networks. In this strategy, FLA strategy is used for location registration and users' calling and mobility patterns are used to update the MTs' location information in the caches of the calling MTs' VLR, called MTs' VLR, calling MTs' FLA, called MTs' FLA, old FLA, and old VLR. However, this strategy effectively exploits the advantages of both the LC and FLA strategies and does not make the centralized HLR congested with enormous

signaling messages during inter-LSTP handoff like PCFLA. In the LC strategy, the calling MT's VLR cache is updated with the called MT's location information just prior to setup each call between the calling and called MTs. But in the NLCFLA strategy, caches of the calling MT's VLR are updated with the called MT's location information, and caches of the pointed/called MT's FLA and VLR are updated with the calling MT's location information during each call setup procedure. No separate signaling messages are needed to update these caches. These are updated during the regular signaling message exchange. Moreover, it also updates the caches of the new FLA,

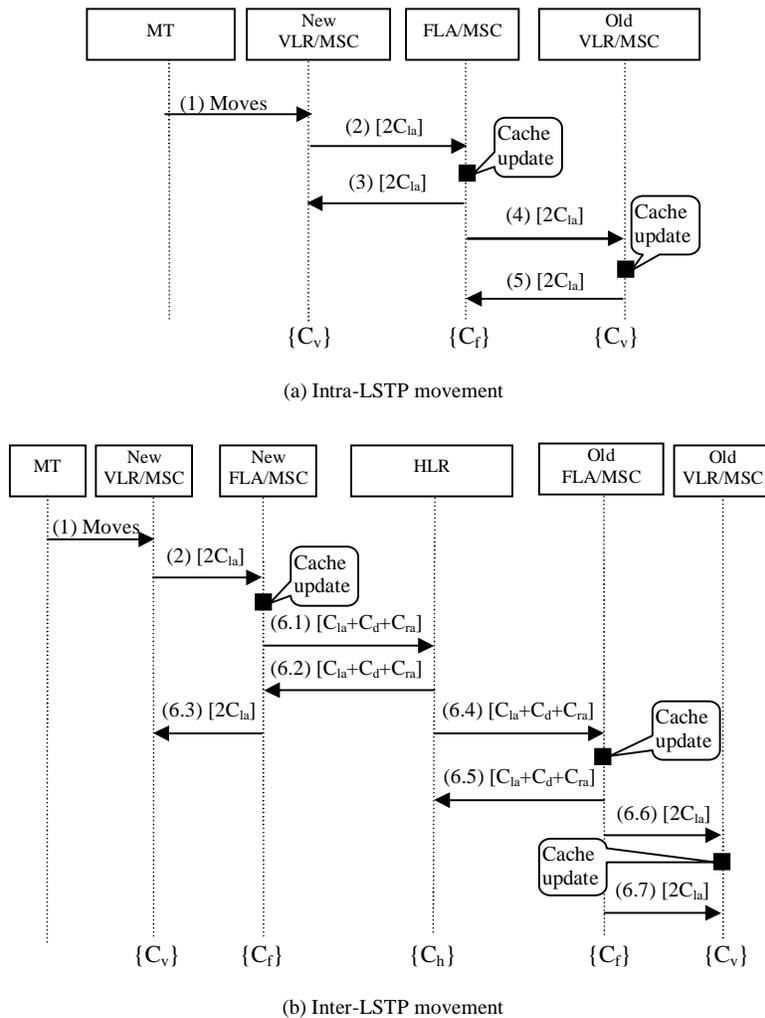


FIGURE 9: Location Registration under the NLCFLA Strategy.

old FLA, and old VLR during intra-LSTP or inter-LSTP handoffs. This also does not need to separate signaling message. When an MT handoffs to a new VLR/MS, it sends its location information to its designated FLA. This designated FLA cache is updated with the MT's new location information together with the same regular registration message. On the other hand, registration cancellation messages are sent to the old FLA and VLR at the time of normal registration procedure. The caches of these old FLA and VLR also updated with the MT's new location information together with the regular registration cancellation message. But in the PCFLA, the MT's new location information is sent to the frequently calling VLR sites using separate explicit signaling messages throughout the networks. During each call delivery process, the calling MT's VLR cache is first searched for the called MT's location information. If that location information is found, it initiates the call setup procedure directly to that called MT without

accessing the HLR for location query. This pointed VLR cache information may be obsolete. In that case, it directly communicates to the HLR for querying location information. But, the probability of having obsolete information in these caches is much lower as many caches are updated for each call delivery as well as handoff rather than only one location update in LC strategy. So, the overall location management cost under this strategy outperforms all other previous strategy. The location management procedure under this strategy is shown in Figure 9 and Figure 10 and the symbols used in these figures are described in Table 1.

The location registration under this strategy is divided into two categories: intra-LSTP movement and inter-LSTP movement. These strategies are described as follows (see Figure 9).

- (1) An MT handoffs into a new RA and informs its new location to the new MSC through the nearby BTS.
- (2) The new MSC forwards a location registration message to its associated FLA within its region.
- (3) The FLA updates its cache with MT's new location information. Following this, the FLA verifies the MT's profile. If there is an MT's record, then update it with the new location information and sends back an acknowledgement message together with a copy of the MT's profile to the new MSC. Otherwise go to step 6.
- (4) The FLA sends a deregistration message to the old MSC.
- (5) The old VLR updates its cache with MT's new location information. Following this, the old MSC clears the record for that MT from the VLR and replies a confirmation message to the FLA (Location registration by intra-LSTP movement is complete. Do not proceed to the next step).
- (6) If the FLA does not have the MT's record, the followings are performed.
 - (6.1) The serving MSC of the MT's new FLA sends a location registration message to the HLR.
 - (6.2) The HLR updates the MT's record in terms of MT's new FLA and sends back a copy of the MT's profile to that new FLA.
 - (6.3) The FLA updates the MT's record in terms of new VLR and acknowledges the new VLR with a copy of the MT's profile.
 - (6.4) The HLR sends a deregistration message to the MT's old FLA.
 - (6.5) The old FLA updates its cache with MT's new location information. Following this, the old FLA removes the MT's record from it and sends a deregistration acknowledgement message back to the HLR.
 - (6.6) The old FLA sends a deregistration message to the MT's old VLR.
 - (6.7) The old VLR updates its cache with MT's new location information. Following this, the old VLR removes the MT's record and sends a deregistration acknowledgement message back to the old FLA (Location registration by inter-LSTP movement is complete).

On the other hand, the call delivery procedure under this strategy is described as follows (see Figures 10(a) and 10(b)).

- (1) The MT sends a call initiation message to the calling MSC through the nearby BTS.
- (2) The called MT's location information is searched in the calling VLR cache and it is assumed that there will be such an entry there. Then it sends a route request message to the pointed FLA/MSC.
- (3) The pointed FLA cache is updated with the calling MT's location information. Following this, the pointed FLA searches the entry for the called MT's location information and forwards this message to the pointed VLR/MSC.
- (4) The pointed VLR cache is updated with the calling MT's location information. Following this, the pointed VLR/MSC verifies whether the location information for the called MT is exact or obsolete. If exact (cache hit), then it sends an acknowledgement message to the calling

MSC together with a TLDN to the MT. Otherwise, it sends a negative acknowledgement message (cache miss) to the calling MSC and go to step 6.

- (5) The calling VLR cache is updated with the called MT's location information. Following this, the calling MSC sends a call setup request message to the called MSC through the network shown in Figure 1 (Call delivery is complete. Do not proceed to the next step).
- (6) If the pointed VLR/MSC contain obsolete location information (cache miss), then it follows the steps below.

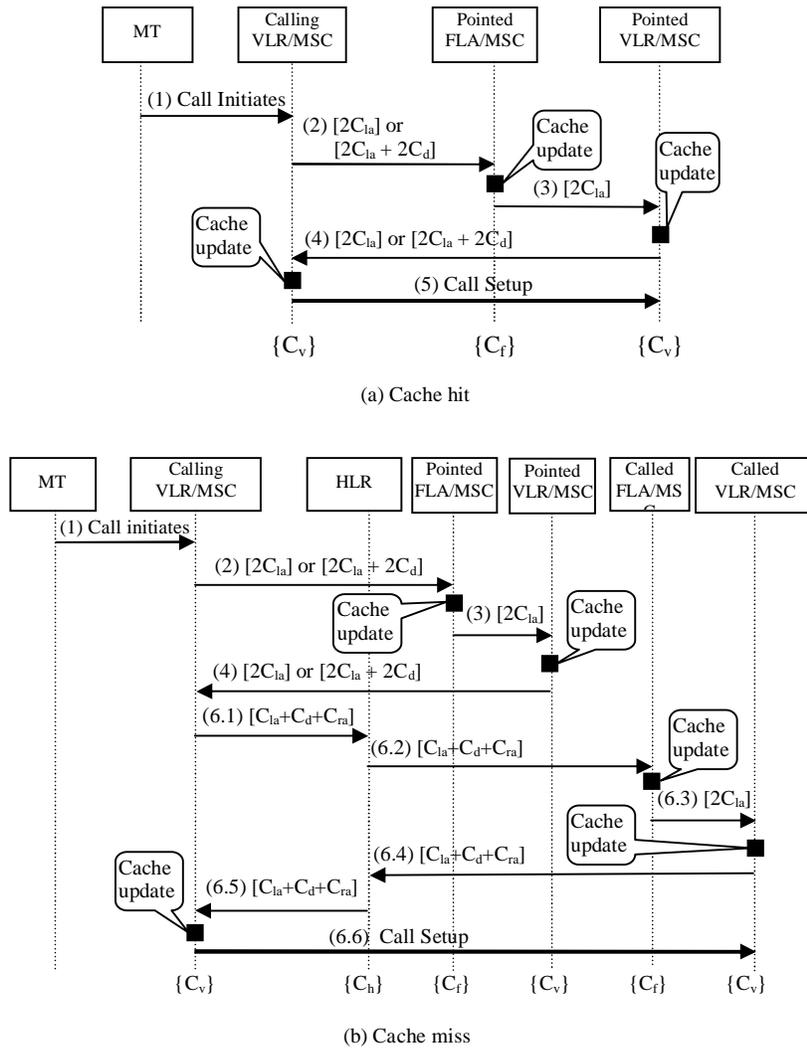


FIGURE 10: Call Delivery under the NLCFLA Strategy.

- (6.1) The calling MSC sends a location request message to the HLR.
- (6.2) The HLR sends a message requesting the called MT's location information to the called FLA.
- (6.3) The called FLA cache is updated with the calling MT's location information and the called FLA forwards the location request message to the called VLR/MSC.
- (6.4) The called VLR cache is updated with the calling MT's location information. Following this, the called VLR/MSC sends a location route back to the HLR together with a TLDN to the MT.
- (6.5) The HLR forwards this message back to the calling MSC.

(6.6) The calling VLR cache is updated with the called MT's location information. Following this, the calling MSC sends a call setup request message to the called MSC through the network shown in Figure 1 (Call delivery is complete).

4. ANALYTICAL MODELING

A fluid flow mobility model [10] is considered in order to evaluate the performance of the proposed and the related approaches. It is assumed that MTs are moving at an average speed of v in uniformly distributed direction over $[0, 2\pi]$ with a view to crossing the LSTP region composed of N equal rectangular-shaped and sized RAs [9]. The parameters used for the MTs' movement rates analysis are defined as follows.

- γ : the MT's movement rate out of an RA
- μ : the MT's movement rate out of an LSTP region
- λ : the MT's movement rate to an adjacent RA within a given LSTP region

According to [11], these parameters are calculated as follows.

$$\gamma = \frac{4v}{\pi\sqrt{S}} \quad (1)$$

$$\mu = \frac{4v}{\pi\sqrt{NS}} \quad (2)$$

$$\lambda = \gamma - \mu = \left(1 - \frac{1}{\sqrt{N}}\right)\gamma \quad (3)$$

Where v is the average moving speed of an MT, S is the size of the RA, and N is the number of RAs within a LSTP region.

A continuous-time Markov Chain state transition diagram is used in Figure 11 to show an MT's RA movement representing the fluid flow mobility model. Each state i ($i \geq 0$) defines the RA number of a given LSTP region where an MT can stay and state 0 means the MT stays outside of this region. The state transition $a_{i,i+1}$ ($i \geq 1$) represents an MT's movement rate to an adjacent RA within a given LSTP region, and $a_{0,1}$ represents an MT's movement rate to an RA of that region from another one. On the other hand, $b_{i,0}$ ($i \geq 1$) represents an MT's inter-LSTP region movement rate and it is assumed that there are maximum K number of such movements.

Therefore, from Figure 11, it is obtained that $a_{i,i+1}$ ($i \geq 1$) = λ and $a_{0,1} = b_{i,0} = \mu$, respectively.

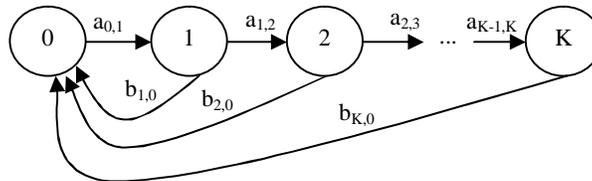


FIGURE 11: State Transition Diagram of an MT's RA Movement.

On the other hand, if π_i is the equilibrium probability of state i , the following equations can be obtained from a continuous-time Markov Chain given in Figure 11.

$$\mu\pi_0 = \mu \sum_{i=1}^K \pi_i \quad (4)$$

$$\mu\pi_{i-1} = (\lambda + \mu)\pi_i, \quad i = 1 \tag{5}$$

$$\lambda\pi_{i-1} = (\lambda + \mu)\pi_i, \quad 2 \leq i \leq K - 1 \tag{6}$$

$$\lambda\pi_{i-1} = \mu\pi_i, \quad i = K \tag{7}$$

Additionally, the sum of the probabilities of all states is 1. So,

$$\pi_0 + \pi_1 + \pi_2 + \dots + \pi_K = \sum_{i=0}^K \pi_i = 1 \tag{8}$$

By substituting (8) into (4), it can be obtained the equilibrium probability of state 0, π_0 . So,

$$\pi_0 = \frac{1}{2} \tag{9}$$

Finally, from (5), (6), (7) and (9), π_i is obtained as follows.

$$\pi_i = \begin{cases} \frac{1}{2} & \text{if } i = 0 \\ \frac{1}{2} \left(\frac{\mu}{\lambda + \mu} \right) \left(\frac{\lambda}{\lambda + \mu} \right)^{i-1} & \text{if } 1 \leq i \leq K - 1 \\ \frac{1}{2} \left(\frac{\lambda}{\lambda + \mu} \right)^{i-1} & \text{if } i = K \end{cases} \tag{10}$$

Parameter	Description
C_{la}	Cost of sending a signaling message through the local A-link
C_d	Cost of sending a signaling message through the D-link
C_{ra}	Cost of sending a signaling message through the remote A-link
C_v	Cost of a query or an update of the VLR
C_f	Cost of a query or an update of the FLA
C_h	Cost of a query or an update of the HLR

TABLE 2: Description of Cost Parameters Shown in Figure 1.

4.1. Analysis of Location Management Costs

In order to analyze the location registration cost, call delivery cost, and total location management cost of the IS-41, LC, FLA, PCFLA, and NLCFLA strategies, different parameters shown in Table 2 and 3 are considered. The following notations are also used to represent the cost of each strategy [9].

- U_X : the average location registration cost of the X strategy for an MT staying in an LSTP region
- S_X : the average call delivery cost of the X strategy for an MT staying in an LSTP region

- T_X : the average total location management cost of the X strategy for an MT staying in an LSTP region
- U_X^Y : the average location registration cost of the X strategy generated by movement type Y for an MT staying in an LSTP region

Parameter	Description
P_l	Probability of locating caller and callee within the same LSTP region
P_n	Probability of playing a new VLR as the role of an FLA
P_o	Probability of playing an old VLR as the role of an FLA
P_f	Probability of locating callee in the FLA area
P_c	Probability of calling MSC having location cache for the called FLA
q	The MT's call-to-mobility ratio (CMR)
τ	The MT's cache hit ratio under the LC and NLCFLA strategies
α	The MT's call arrival rate through MSC
m	Number of location servers (HLRs) in the system

TABLE 3: Parameters Used for the Cost Analysis.

Furthermore, the average number of unique RAs that an MT visits within a given LSTP for K movements can be calculated from Figure 11 and represented by the following equation.

$$\Phi(K) = \pi_1 + 2\pi_2 + 3\pi_3 + \dots + K\pi_K = \sum_{i=1}^K i\pi_i \quad (11)$$

The location management cost functions of the IS-41, LC, FLA, PCFLA, and NLCFLA strategies can be derived from Figure 2 – 10.

1) IS-41 Strategy

The average location registration cost of the IS-41 strategy is derived as follows.

$$\begin{aligned} U_{IS-41} &= \pi_0 U_{IS-41}^{inter} + (\Phi(K)-1)U_{IS-41}^{intra} \\ &= \pi_0 \times \{4(C_{la} + C_d + C_{ra}) + (2C_v + C_h)\} + (\Phi(K)-1) \times \{4(C_{la} + C_d + C_{ra}) + (2C_v + C_h)\} \\ &= \{\pi_0 + (\Phi(K)-1)\} \{4(C_{la} + C_d + C_{ra}) + (2C_v + C_h)\} \end{aligned} \quad (12)$$

The average call delivery cost of the IS-41 is defined as follows (see Figure 3).

$$S_{IS-41} = 4(C_{la} + C_d + C_{ra}) + (2C_v + C_h) \quad (13)$$

So, the average total cost of the IS-41 strategy is expressed as follows.

$$T_{IS-41} = U_{IS-41} + qS_{IS-41} \quad (14)$$

2) LC Strategy

The average location registration cost of the LC strategy is defined as follows.

$$U_{LC} = U_{IS-41}$$

$$= \{\pi_0 + (\Phi(K) - 1)\} \{4(C_{la} + C_d + C_{ra}) + (2C_v + C_h)\} \quad (15)$$

The average call delivery cost of the LC strategy is expressed as follows.

$$S_{LC} = \tau C_{LC}^{hit} + (1 - \tau) C_{LC}^{miss} \quad (16)$$

Where C_{LC}^{hit} and C_{LC}^{miss} represents the call delivery cost of the LC strategy when location information pointed by cache for the called MT is correct and obsolete, respectively (see Figures 4(a) and 4(b)). τ and these two parameters are calculated by the following equations.

$$\begin{aligned} \tau &= \frac{\tau \times \alpha + (1 - \tau) \times \alpha}{\tau \times \alpha + (1 - \tau) \times \alpha + \lambda} \\ &= \frac{\alpha}{\alpha + \lambda} \end{aligned} \quad (17)$$

Since the caches in LC strategy are updated only at call arrival time.

According to [12],

$$q = \frac{\alpha}{\lambda} \quad (18)$$

So, (17) is expressed as follows.

$$\tau = \frac{q}{1 + q} \quad (19)$$

Again,

$$C_{LC}^{hit} = P_l(4C_{la} + 2C_v) + (1 - P_l)(4C_{la} + 4C_d + 2C_v) \quad (20)$$

$$\begin{aligned} C_{LC}^{miss} &= C_{LC}^{hit} + S_{IS-41} \\ &= \{P_l(4C_{la} + 2C_v) + (1 - P_l)(4C_{la} + 4C_d + 2C_v)\} + \{4(C_{la} + C_d + C_{ra}) + (2C_v + C_h)\} \end{aligned} \quad (21)$$

So, the average total cost of the LC strategy is expressed as follows.

$$T_{LC} = U_{LC} + qS_{LC} \quad (22)$$

3) FLA Strategy

The average location registration cost of the FLA strategy is defined as follows.

$$U_{FLA} = \pi_0 U_{FLA}^{inter} + (\Phi(K) - 1) U_{FLA}^{intra} \quad (23)$$

Three possible cases occur in intra-LSTP movement: (a) new VLR is the FLA and old VLR is not the FLA, (b) old VLR is the FLA and new VLR is not the FLA, and (c) neither new VLR nor old VLR is the FLA. On the other hand, four possible cases occur in inter-LSTP movement: (a) new VLR is the FLA and old VLR is not the FLA, (b) old VLR is the FLA and new VLR is not the FLA, (c) both new VLR and old VLR are the FLA, and (d) neither new VLR nor old VLR is the FLA.

According to these possible cases, U_{FLA}^{intra} and U_{FLA}^{inter} are calculated as follows.

$$U_{FLA}^{intra} = P_n(1 - P_0)F_1 + P_0(1 - P_n)F_2 + (1 - P_n)(1 - P_0)F_4 \quad (24)$$

$$U_{FLA}^{inter} = P_n(1 - P_0)F_5 + P_0(1 - P_n)F_6 + P_n P_0 F_7 + (1 - P_n)(1 - P_0)F_8 \quad (25)$$

Where $F_1, F_2, F_4, F_5, F_6, F_7,$ and F_8 are expressed as follows (see Figure 5 and Figure 6).

$$F_1 = F_2 = 2 \times 2C_{la} + C_v + C_f \quad (26)$$

$$F_4 = 4 \times 2C_{la} + 2 \times C_v + C_f \quad (27)$$

$$F_5 = F_6 = 4(C_{la} + C_d + C_{ra}) + 4C_{la} + C_v + 2C_f + C_h \quad (28)$$

$$F_7 = 4(C_{la} + C_d + C_{ra}) + 2C_f + C_h \quad (29)$$

$$F_8 = 4(C_{la} + C_d + C_{ra}) + 8C_{la} + 2C_v + 2C_f + C_h \quad (30)$$

The average call delivery cost under this strategy depends on whether the called MT located in the same FLA area or in different FLA area. Thus this cost is expressed as follows.

$$S_{FLA} = P_f D_1 + (1 - P_f) D_2 \quad (31)$$

Where D_1 and D_2 are expressed as follows (see Figure 8(b)).

$$D_1 = 4(C_{la} + C_d + C_{ra}) + (C_v + C_f + C_h) \quad (32)$$

$$D_2 = 4(C_{la} + C_d + C_{ra}) + (2C_v + C_f + C_h) + 2C_{la} \quad (33)$$

So, the average total cost under this strategy is expressed as follows.

$$T_{FLA} = U_{FLA} + qS_{FLA} \quad (34)$$

4) PCFLA Strategy

The average location registration cost under the PCFLA strategy is defined as follows.

$$U_{PCFLA} = \pi_0 U_{PCFLA}^{inter} + (\Phi(K) - 1) U_{PCFLA}^{intra} \quad (35)$$

The three possible cases that occur in intra-LSTP movement and four possible cases that occur in inter-LSTP movement are the same as that of the FLA strategy. According to these possible cases, U_{PCFLA}^{intra} and U_{PCFLA}^{inter} are calculated as follows.

$$U_{PCFLA}^{intra} = P_n(1 - P_0)C_1 + P_0(1 - P_n)C_2 + (1 - P_n)(1 - P_0)C_4 \quad (36)$$

$$U_{PCFLA}^{inter} = P_n(1 - P_0)C_5 + P_0(1 - P_n)C_6 + P_n P_0 C_7 + (1 - P_n)(1 - P_0)C_8 \quad (37)$$

Where $C_1, C_2, C_4, C_5, C_6, C_7,$ and C_8 are expressed as follows (see Figure 5 and Figure 7).

$$C_1 = C_2 = 2 \times 2C_{la} + C_v + C_f \quad (38)$$

$$C_4 = 4 \times 2C_{la} + 2 \times C_v + C_f \quad (39)$$

$$C_5 = C_6 = 4(C_{la} + C_d + C_{ra}) + 4C_{la} + C_v + 2C_f + C_h + m(C_{la} + C_d + C_{ra}) \quad (40)$$

$$C_7 = 4(C_{la} + C_d + C_{ra}) + 2C_f + C_h + m(C_{la} + C_d + C_{ra}) \quad (41)$$

$$C_8 = 4(C_{la} + C_d + C_{ra}) + 8C_{la} + 2C_v + 2C_f + C_h + m(C_{la} + C_d + C_{ra}) \quad (42)$$

On the other hand, the average call delivery cost under this strategy is calculated as follows

$$S_{PCFLA} = P_c C_{PCFLA}^{cache} + (1 - P_c) C_{PCFLA}^{nocache} \quad (43)$$

Where C_{PCFLA}^{cache} and $C_{PCFLA}^{nocache}$ are the call delivery costs under the PCFLA strategy when the calling MT's MSC contain location cache for the called FLA and when the calling MT's MSC does not have location cache for the called FLA, respectively.

There are four possible cases occur for delivering a call under this strategy: (a) the calling and called MTs are located within the same LSTP region and the called MT is resided in the FLA area, (b) the calling and called MTs are located within the same LSTP region and the called MT is resided in the other VLR area, (c) the calling and called MTs are located in the different LSTP regions and the called MT is resided in the FLA area, and (d) the calling and called MTs are located in the different LSTP regions and the called MT is resided in the other VLR area.

According to these possible cases, C_{PCFLA}^{cache} and $C_{PCFLA}^{nocache}$ is calculated as follows (see Figures 8(a) and 8(b)).

$$C_{PCFLA}^{cache} = P_l\{P_f N_1 + (1 - P_f)N_2\} + (1 - P_l)\{P_f N_3 + (1 - P_f)N_4\} \quad (44)$$

$$C_{PCFLA}^{nocache} = S_{FLA} = P_f D_1 + (1 - P_f)D_2 \quad (45)$$

Where $N_1, N_2, N_3,$ and N_4 are defined as follows (see Figures 8(a) and 8(b)).

$$N_1 = 2 \times 2C_{la} + C_v + C_f \quad (46)$$

$$N_2 = 3 \times 2C_{la} + 2 \times C_v + C_f \quad (47)$$

$$N_3 = 2 \times (2C_{la} + 2C_d) + C_v + C_f \quad (48)$$

$$N_4 = 2 \times (2C_{la} + 2C_d) + 2 \times C_v + C_f + 2C_{la} \quad (49)$$

As a result, the average total cost of this strategy is expressed as follows.

$$T_{PCFLA} = U_{PCFLA} + qS_{PCFLA} \quad (50)$$

Criteria		Condition	No. of cache update(s)	Average No. of cache update(s)	
Cache hit		Pointed VLR is found in the FLA area	2	2.5	
		Pointed VLR is not found in the FLA area	3		
Cache miss		Pointed VLR is found in the FLA area and the called VLR is found in the FLA area	3	4	
		Pointed VLR is found in the FLA area and the called VLR is not found in the FLA area	4		
		Pointed VLR is not found in the FLA area and the called VLR is found in the FLA area	4		
		Pointed VLR is not found in the FLA area and the called VLR is not found in the FLA area	5		
Location registration	Intra-LSTP movement	New VLR is the FLA and old VLR is not the FLA	1	1.33	1.67
		New VLR is not the FLA and old VLR is the FLA	1		
		New VLR is not the FLA and old VLR is not the FLA	2		
	Inter-LSTP movement	New VLR is the FLA and old VLR is not the FLA	2	2	
		New VLR is not the FLA and old VLR is the FLA	2		
		New VLR is not the FLA and old VLR is not the FLA	3		
		New VLR is the FLA and old VLR is the FLA	1		

TABLE 4: Average No. of Cache Updates per Call.

5) NLCFLA Strategy

The average location registration cost under the NLCFLA strategy is defined as follows (see Figure 9).

$$U_{NLCFLA} = U_{FLA} = \pi_0 U_{FLA}^{inter} + (\Phi(K) - 1)U_{FLA}^{intra} \quad (51)$$

Where U_{FLA}^{inter} and U_{FLA}^{intra} are the same as that of the FLA strategy and calculated in Section 4.1(3).

On the other hand, the average call delivery cost under the NLCFLA strategy is expressed as follows (see Figure 10).

$$S_{NLCFLA} = \tau C_{PCFLA}^{cache} + (1-\tau)(C_{PCFLA}^{cache} + C_{PCFLA}^{nocache}) \quad (52)$$

Where C_{PCFLA}^{cache} and $C_{PCFLA}^{nocache}$ are the same as that of the PCFLA strategy and these are calculated in Section 4.1(4). Moreover, τ for the NLCFLA strategy is calculated as follows (see Table 4).

$$\tau = \frac{\tau \times 2.5\alpha + (1-\tau) \times 4\alpha + 1.67\lambda}{\tau \times 2.5\alpha + (1-\tau) \times 4\alpha + 1.67\lambda + \lambda} \quad (53)$$

By solving this, the following result for τ is obtained as a function of α and λ .

$$\tau = \frac{5.5\alpha + 2.67\lambda - \sqrt{6.25\alpha^2 + 19.35\alpha\lambda + 7.13\lambda^2}}{3\alpha} \quad (54)$$

By solving (53) after applying (18) to it, the following result for τ is obtained as a function of q .

$$\tau = \frac{5.5q + 2.67 - \sqrt{6.25q^2 + 19.35q + 7.13}}{3q} \quad (55)$$

As a result, the average total cost of this strategy is expressed as follows.

$$T_{NLCFLA} = U_{NLCFLA} + qS_{NLCFLA} \quad (56)$$

Set	C_{la}	C_d	C_{ra}
1	1	1	1
2	1	3	3
3	1	3	6
4	1	5	10

TABLE 5: Signaling Costs Parameter Set.

Set	C_v	C_f	C_h
5	1	1	1
6	1	2	3
7	1	2	3
8	1	3	5

TABLE 6: Database Access Costs Parameter Set.

5. NUMERICAL RESULTS AND COMPARISONS

We have implemented the proposed and all other approaches containing equations (14), (22), (34), (50), and (56) using C programming language and execute them on a Windows XP Operating System installed computer having configuration of Pentium 4 processor with 3.00 GHz speed and 512 MB of RAM. We assume $N = 55$, $v = 5.6$ km/h, $S = 20$ km², $P_l = 0.043$, $K = 55$, and $m = 5$ [9] for generating various numerical results of these approaches. In addition to these fixed parameters, we compare the performance of the LC, FLA, PCFLA, and NLCFLA strategies with that of the IS-41 under various conditions given in Table 5 and 6. These are compared in terms of

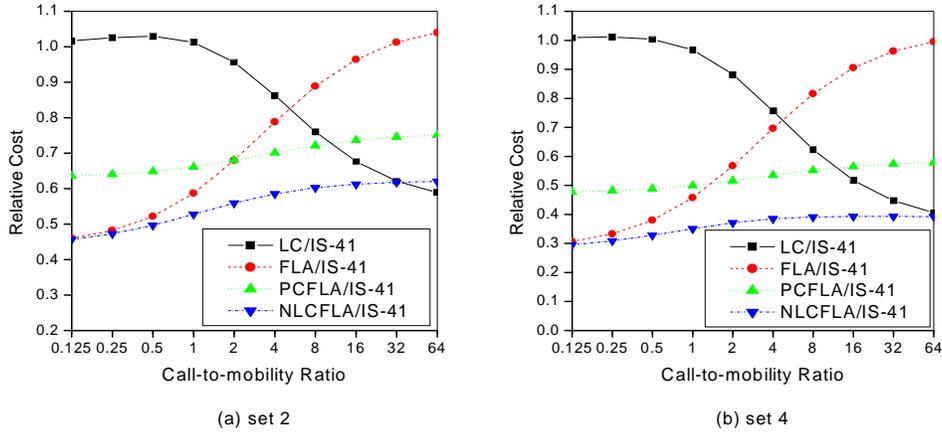


FIGURE 12: Relative Signaling Cost.

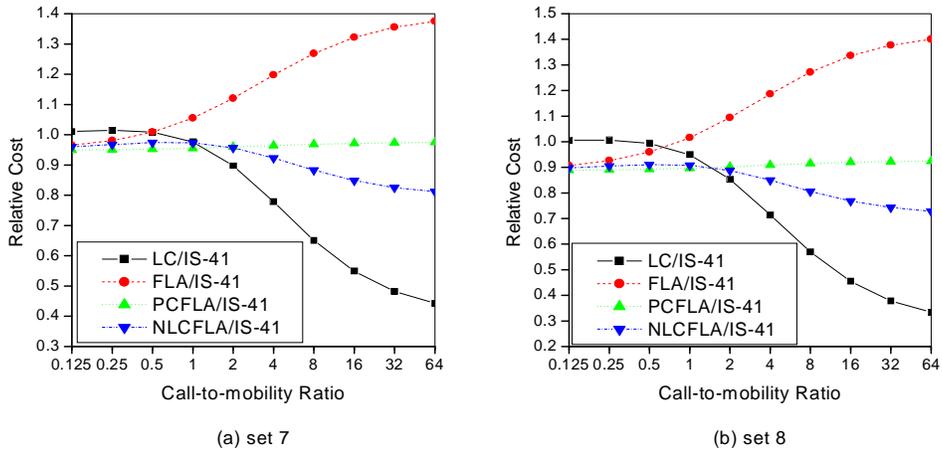


FIGURE 13: Relative Database Access Cost.

$$Relative\ cost = \frac{T_X}{T_{IS-41}} \tag{57}$$

relative cost where its value 1 means that the costs under both strategies are exactly the same. The relative cost of the X strategy can be defined as the ratio of the average total cost of the X strategy to that of the IS-41 strategy using the following equation.

5.1 Signaling Costs, Database Access Costs, and Total Costs

Figures 12, 13, and 14 show the relative signaling costs, relative database access costs, and relative total costs of the LC, FLA, PCFLA, and NLCFLA strategies with respect to CMR for different parameter sets given in Table 5 and 6, respectively. In Figure 12, signaling cost dominates the database access cost by setting the cost parameters C_v, C_f and C_h to 0, whereas in Figure 13, database access cost dominates the signaling cost by setting the cost parameters C_{la}, C_d , and C_{ra} to 0. On the other hand, all the cost parameters have the same domination effect in Figure 14.

We see from the graphs that the cost of the LC strategy gets lower as CMR increases, while the cost of the FLA strategy gets lower as CMR decreases. These trends are expected and easily

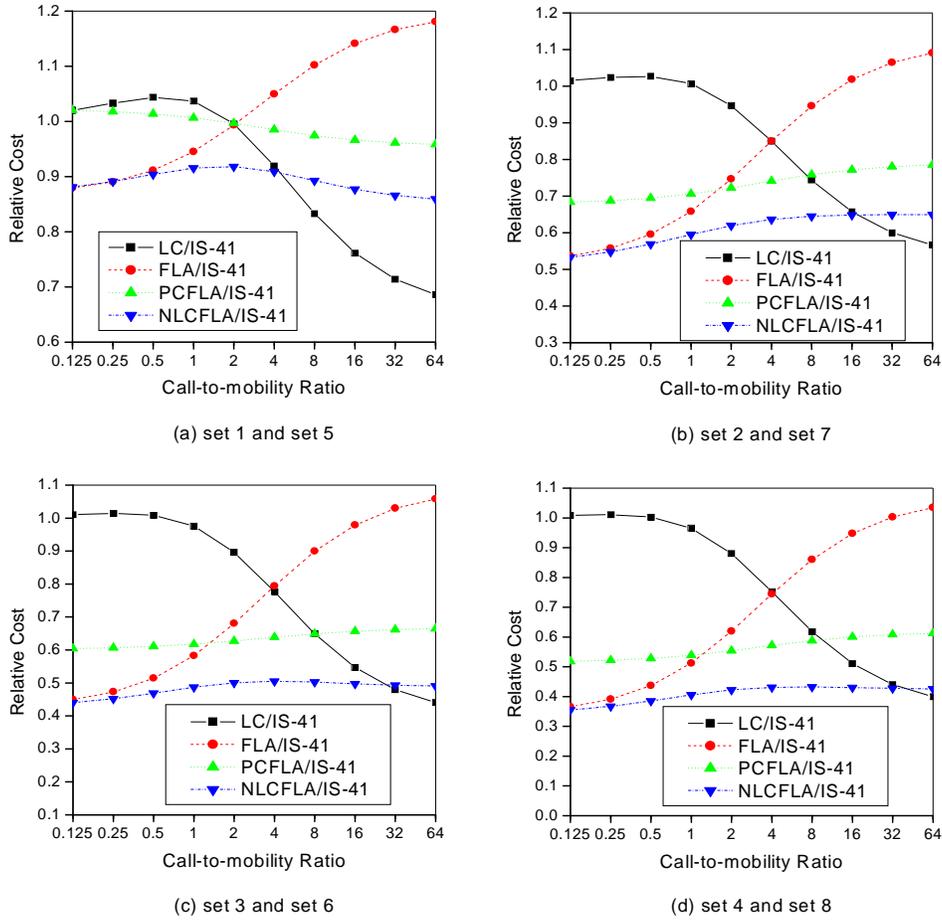


FIGURE 14: Relative Total Cost.

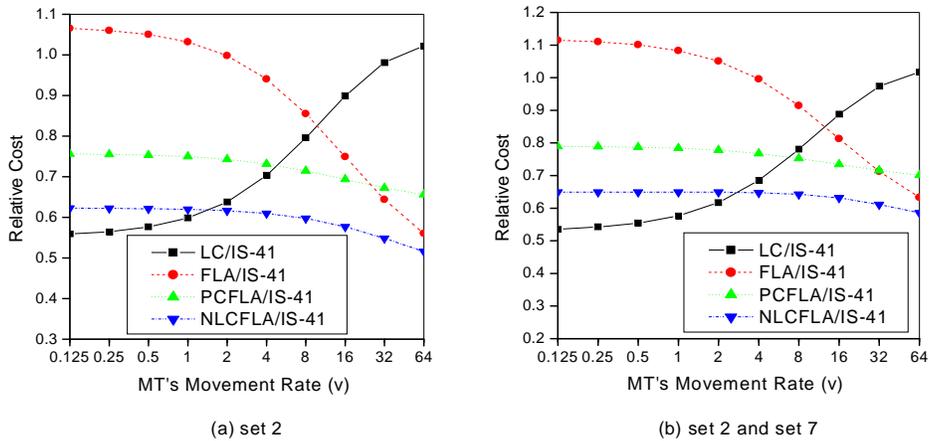


FIGURE 15: Relative Signaling Cost (a) and Total Cost (b).

explainable from the working principle of these two strategies. As CMR increases, the call delivery becomes dominated, and thus the cost of the LC strategy gets lower. Alternatively, as CMR decreases, the location registration becomes dominated, and thus the cost of the FLA

strategy gets lower. Moreover, it is observed from the graphs that the costs of both the PCFLA and NLCFLA strategies are not much sensitive to the change of CMR and the cost of the latter strategy is always lower than that of the former one. These trends are also expected and easily explainable from the working procedure of these two strategies. These two are the combination of the FLA strategy and a modified form of the LC strategy. The FLA strategy works better for lower CMR as location registration dominates for this condition. On the other hand, the modified form of the LC strategy works better for increased CMR as call delivery dominates for this condition. As a result, the combination of these two makes the PCFLA and NLCFLA strategies less sensitive to the CMR. But, the cost of the latter strategy is always less than that of the former strategy. The reason behind this trend is that the PCFLA strategy sends separate excessive location caching message to the MSCs during inter-LSTP movement registration, whereas the NLCFLA strategy performs location caching during the exchange of regular messages. It also does not send separate location caching message and these are performed in call delivery time as well as location registration time. So the cost of this strategy always gets lower.

The relative signaling and total costs of the LC, FLA, PCFLA, and NLCFLA strategies with respect to MT's mobility rate (v) are shown in Figure 15. This graph is generated by assuming $\alpha = 50$ and replacing λ of (17) and (54) by v . It is observed from this graph that the cost of the LC strategy increases as v increases and the FLA strategy shows increased cost as v decreases. Moreover, the PCFLA and NLCFLA strategies are less sensitive to the change of v and the latter strategy always outperforms the former one irrespective of the values of v . These are expected and easily explainable with the same dominating principle as described in the previous paragraph.

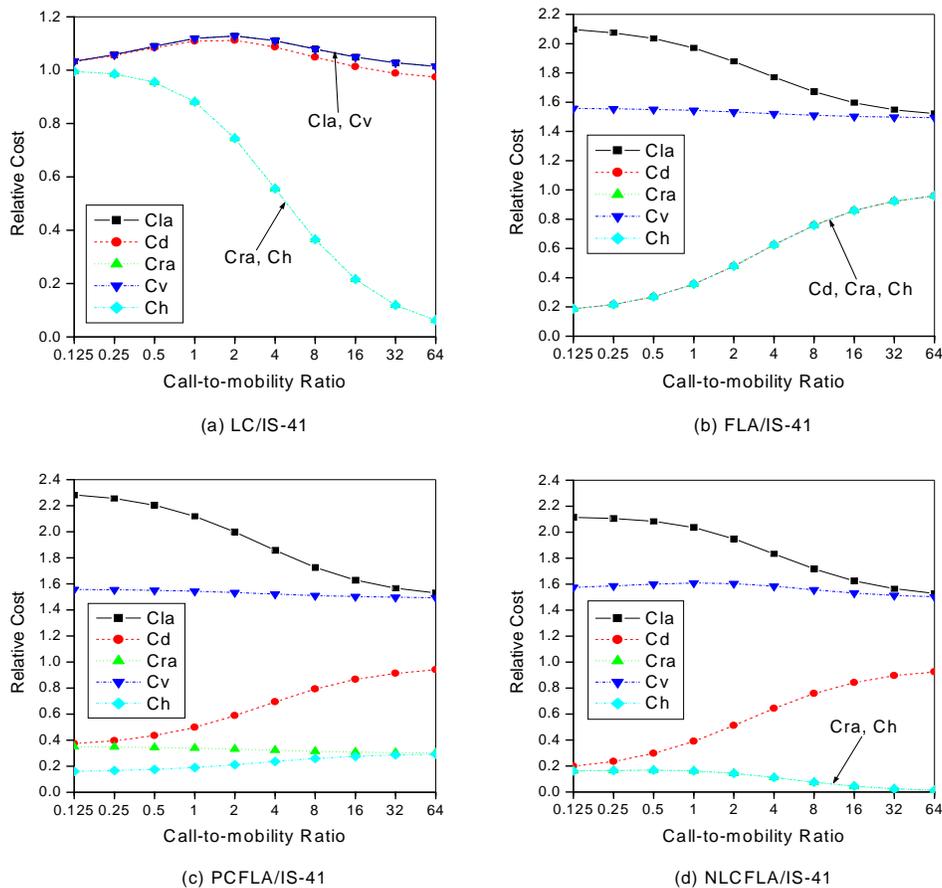


FIGURE 16: Impact of C_{la} , C_d , C_{ra} , C_v , and C_h .

5.2 Impact of C_{la} , C_d , C_{ra} , C_v , and C_h .

Figure 16 shows how the individual cost terms C_{la} , C_d , C_{ra} , C_v , and C_h affect on the overall relative costs of the LC, FLA, PCFLA, and NLCFLA strategies with respect to CMR for set 2 and 7 data given in Table 5 and 6. The values of C_v and C_f are assumed to be equal for the analysis as C_f is one of the form of C_v . As CMR increases, the cost of the LC strategy gets lower, whereas the cost of the FLA strategy gets higher for dominant cost terms C_{ra} and C_h . But, the cost of both the PCFLA and NLCFLA strategies remain almost same with a little decrease for the latter strategy with the increase of CMR for these dominant cost terms. These are easily explainable from the working principle of these strategies. The LC strategy shows this behavior since it needs small access to the HLR for increased CMR due to its availability of MTs' location information in the caches. On the other hand, FLA needs more access to the HLR for increased CMR since there are no caches maintained here. Alternatively, the PCFLA and NLCFLA strategies use both caches and FLA for optimizing call delivery and location registration. But, the NLCFLA strategy outperforms the PCFLA as it does not need to send separate location caching message during inter-LSTP movement. Moreover, it is observed from the graphs that the cost of the FLA, PCFLA, and NLCFLA strategies for the dominant cost terms C_{la} and C_v are higher than that of the IS-41 strategy. This is due to the fact that these strategies except IS-41 distribute the functionality of the HLR to the VLRs.

6. CONCLUSION

Location management in wireless mobile networks is one of the most important and challenging issues in the current world. To manage the location effectively and efficiently, a simple but efficient location management strategy called NLCFLA is proposed in this paper. This strategy uses FLA strategy which optimizes the location registration cost with a view to managing the local location registration locally instead of informing to the HLR. It also updates the MT's location information in VLR caches during the exchange of regular messages to the network at call origination time, call delivery time and also at location registration time. It does not use separate messages for location caching like previous strategy. This updating strategy eventually optimizes the call delivery cost. As a result, it minimizes the total location management cost in terms of location registration cost and call delivery cost regardless of the CMR. The various cost functions under the same analytical modeling for the proposed and related strategies are derived. It also investigates the effects of individual cost parameters for each strategy. The numerical results obtained from these cost functions show that the proposed NLCFLA strategy outperforms all the related previous strategies irrespective of the MT's calling and mobility pattern.

Currently, it is being observed that how this caching strategy can be combined with other location management strategies to get better result.

7. REFERENCES

- [1] W. Wong and V. Leung. "Location management for next generation personal communication networks". IEEE Network, 14(5):18–24, September 2000.
- [2] EIA/TIA. "Cellular radio-telecommunications intersystem operations". Tech. Rep. IS-41 Revision B, EIA/TIA, December 1991.
- [3] M. Mouly and M. B. Pautet. "The GSM system for mobile communications". Telecom Publishing, 49 rue Louise Bruneau, Palaiseau, France, January 1992.
- [4] R. Jain, Y.B. Lin, and S. Mohan. "A caching strategy to reduce network impacts of PCS". IEEE Journal on Selected Areas in Communications, 12(8):1434–1444, October 1994.
- [5] J. Ho and I. Akyildiz. "Local anchor scheme for reducing signaling costs in personal communications networks". IEEE/ACM Transactions on Networking, 4(5):709–725, October 1996.
- [6] E. Bae, M. Chung, and D. Sung. "Performance analysis of fixed local anchor scheme for supporting UPT services". Computer Communications, 23(18):1763–1776, December 2000.
- [7] R. Jain and Y. B. Lin. "An auxiliary user location strategy employing forwarding pointers to reduce network impacts of PCS". ACM-Baltzer Journal on Wireless Networks, 1(2):197–210, July 1995.

- [8] N. Shivakumar and J. Widom. "*User profile replication for faster location lookup in mobile environments*". In Proceedings of the 1st Annual International Conference on Mobile Computing and Networking, pp. 161–169, 1995.
- [9] K. Kong. "*Performance analysis of profile-based location caching with fixed local anchor for next-generation wireless networks*". IEICE transactions on communications, E91-B(11):3595–3607, November 2008.
- [10] F. Baumann and I. Niemegeers. "*An evaluation of location management procedures*". In Proceedings of the Third Annual International Conference on Universal Personal Communications, USA, pp. 359–364, September 1994.
- [11] K. Kong, M. Song, K. Park, and C. Hwang. "*A comparative analysis on the signaling load of Mobile IPv6 and Hierarchical Mobile IPv6: Analytical approach*". IEICE Transactions on Information and Systems, E89-D(1):139–149, Japan, January 2006.
- [12] E. Pitoura and G. Samaras. "*Data Management for Mobile Computing*". Kluwer Academic Publishers, vol. 10, 1998.

Traffic Engineering in Metro Ethernet

Padmaraj M. V. Nair

mpadmara@lyle.smu.edu

Suku V. S. Nair

nair@lyle.smu.edu

F. Marco Marchetti

marco@lyle.smu.edu

*HACNet Lab, Southern Methodist University,
Dallas, TX, USA*

Girish Chiruvolu

Maher Ali

*Alcatel-Lucent
Plano TX, USA*

Abstract

Traffic engineering is one of the major issues that has to be addressed in Metro Ethernet networks for quality of service and efficient resource utilization. This paper aims at understanding the relevant issues and outlines novel algorithms for multipoint traffic engineering in Metro Ethernet. We present an algorithmic solution for traffic engineering in Metro Ethernet using optimal multiple spanning trees. This iterative approach distributes traffic across the network uniformly without overloading network resources. We also introduce a new traffic specification model for Metro Ethernet, which is a hybrid of two widely used traffic specification models, the pipe and hose models.

Keywords: Metro Ethernet, Traffic Engineering, Multiple Spanning Trees

1. INTRODUCTION

Ethernet has evolved over the past decade from a simple CSMA/CD shared medium access protocol to a full-duplex, switched network. Ethernet dominates the current LAN realizations and it has been estimated that more than 90% of IP traffic originates from an Ethernet LAN. Inherent simplicity of Ethernet brings nice properties like cost effectiveness, rapid provisioning on demand, ease of interworking, simple packet based technology and ubiquitous adoption [6]. It promises relatively inexpensive high-speed access, which can then be combined with new networking services from the enterprise domain. With the latest enhancements, it has turned out to be a feasible technology even beyond the LAN. Switching, Fast Ethernet and Gigabit Ethernet have brought more bandwidth to the technology.

However Ethernet lacks end-to-end QoS guarantees, protection mechanisms and service performance monitoring [6]. Present Ethernet connectivity is largely restricted to the fast-growing but relatively small niche of the Internet access. In order to enter the metro domain, Ethernet must prove that it is a carrier-grade technology. Deploying Ethernet in the MAN will require many different upgrades. First, it requires the port-portfolio of Ethernet physical layers to be extended for the local loop and for interoffice connection within the MAN. Second, there is a need to upgrade

the Ethernet switching layer. It should be reliable and secure enough to handle mission critical corporate data.

Current Ethernet protocol relies on the IEEE spanning tree protocol (STP) [2], which provides a loop-free connectivity across various network nodes. But this protocol has slow re-convergence time in the case of a failure that typically is 50 seconds. Time critical applications on Ethernet can only afford a reconfiguration time less than few tens of milliseconds. Further, STP uses a single spanning tree to carry the entire network traffic, resulting in congestion and resource underutilization. To improve upon reconstruction time 802.1W [3] standardized its Rapid Spanning Tree Protocol (RSTP). In both RSTP and STP, after a re-convergence the MAC address might get associated with an altogether different switch port and makes it difficult to predetermine the path between a given pair of nodes after a spanning re-convergence.

The Multiple Spanning Tree Protocol (MSTP) [4] defined in IEEE 802.1s standard provides alternate paths between two nodes within an administrative region such that basic traffic engineering can be enabled. This standard provides guidelines for better resource utilization, localization of failures, and faster recovery. MSTP uses multiple spanning trees and VLANs [1] are mapped onto these trees.

Customers are demanding end-to-end Ethernet solutions in which geographically distant LANs are connected as if they were one single LAN. This customer virtual LAN(C-VLAN) provides full connectivity between customer sites (LANs), and allows a station in one LAN to engage in unicast, multicast, and broad-cast communication with any other station belonging to the C-VLAN [14]. In this paper we present an algorithmic approach for traffic engineering based on construction of multiple spanning trees in the metro do-main using customer traffic demands and given network topology.

Traffic specification is another important factor in traffic engineering. Current Metro Ethernet uses pipe and hose models. In our multiple spanning tree algorithm we use the pipe model. The pipe model provides the most efficient interface for the provider in terms of bandwidth allocation. However, it suffers from being complex and it requires pair-wise traffic information from the customer. The hose model, on the other hand, does not require knowledge of pair-wise traffic, but only aggregates. In addition, it is easy to specify by the customer. However, the lack of exact traffic distributions makes it least efficient in terms of bandwidth allocation. To take advantage of both these methods we introduce a hybrid model (Augmented Hose) in the future directions. The idea is to enhance our multiple spanning tree construction method with this new traffic specification model in the future.

The rest of the paper is organized as follows. Next section provides an overview of the IEEE Ethernet standards and the kind of network in consideration is described in section 3. Section 4 discusses the related work in this area. Section 5 proposes an algorithmic approach to multiple spanning tree construction. In section 6 we provide examples showing the performance of our approach. Future directions are discussed in section 7. Finally, section 8 concludes with a summary of our work.

2. OVERVIEW OF THE ETHERNET

This section elaborates on some of the relevant IEEE standards for Ethernet.

2.1 IEEE Spanning Tree

Ethernet traditionally relies on spanning tree defined in IEEE 802.1D, which inherently provides loop-free communication among all the nodes. Basically the construction of the spanning tree through the exchange of node ids and link costs (a.k.a. Port vectors). The algorithm terminates upon the discovery of the root node with the lowest id number and the lowest cost paths between every node and the root node. Each port of a switch assumes one of the three roles, namely, a)

Root port b) Designated port c) Disabled port. The port that leads to the Root Bridge with least path cost is the Root port for that non-root bridge. All the other ports (links) are either designated ports which forward frames on behalf of a LAN segment or blocked ports. Blocked ports lead to the underutilization of the links. The blocked ports/links serve as backups in the event of a change in topology due to link/node failure. However, the spanning tree reconstruction after a link failure has slow convergence as the default timers associated with STP are based on worst-case behavior of the network in terms of size (diameter) and response. Typically the worst-case reconvergence of the STP for a stable spanning tree after an event of link failure is around 50 seconds which is not acceptable for real time mission critical applications.

2.2 The Rapid Spanning Tree

The blocking port role in STP is split into two in RSTP, backup and alternate port roles. These ports do not forward user data. If the port is considered for immediate forwarding state in the event of root port failure then it is an alternate port. A port is backup port if it is for immediate forwarding in the case of a designated port failure. This is due to the fact that the alternate and the backup ports also do the learning even before they are promoted to the forwarding state. This significantly reduces the convergence time of RSTP, but still in the order of few seconds (~1-2 seconds) as the flush message (broadcast by the switch that gains in the subtree, rooted by itself, topology) takes time to propagate across the nodes as a result of (logical) topology change due to change in port status. Moreover, RSTP suffers from the same drawback as STP: both standards use a single tree to carry all the network traffic causing resource overloading and resource under utilization.

2.3 The Multiple Spanning Trees

IEEE 802.1s deals with defining Multiple Spanning Trees (MSTs) wherein each VLAN can be uniquely mapped onto a single spanning tree among the set of spanning trees. One can have a spanning tree per VLAN or multiple VLANs can be mapped onto a single spanning tree. Collection of MSTs in a VLAN-aware network is defined as a Spanning Forest (SF). The SF allows the utilization of all links that would otherwise be idled by the standard ST and thereby eliminating the wastage of bandwidth. However, the complexity of management increases with the number of spanning trees, switches must maintain port state information for each spanning tree of the SF, and the set of VLANs mapped on to them. The MSTP also allows a set of regions to be defined whose union spans the entire network. Within a region one or more MSTs can be defined. MSTP ensures that the frames of a given VLAN are assigned to only one of the spanning tree instances within a region. A common spanning tree (CST) spans across all the regions. The key advantage of such segmentation of the network into regions is the localization of any failure while having minimum/no impact on the non-local VLAN traffic. However, in order to have alternate paths between any pair of nodes with different VLAN segments attached, multiple regions have to be defined. This is due to the fact that each of the regions binds a given VLAN traffic with a unique instance of spanning tree, which in turn translates into a single but unique path. Moreover, spanning trees constructed in a region essentially follow RSTP and thus inherently carry the weakness of RSTP.

3. SYSTEM DESCRIPTION

Several architectural options have been proposed and deployed to carry Ethernet frames across metro area.

- Extending the native Ethernet protocol which is standardized under 802.1 committee.
- Using MPLS as the transport technology.
- Using SONET/SDH as the transport technology via General Frame via Generic Framing Procedure (GFP) encapsulation and Link Capacity Adjustment Scheme (LCAS) rate adaptation.
- Using generalized MPLS (GMPLS) to control Ethernet switches in the metro network.

Our research is based on the first architecture in conformance with the 802.1 standards. Fig 1 depicts the typical architecture for Metro Ethernet [14]. The metro domain consists of core switches/nodes and edge nodes (ingress/egress nodes). The goal is to transport end user Ethernet frames across the metro domain in a transparent manner. Since the Ethernet MAC addresses are flat, ingress nodes encapsulate the user frames with the proposed extension fields. The edge node identifiers are inserted appropriately such that the core nodes learn only about the edge nodes of the metro domain and not the end user MAC addresses. The edge nodes in turn learn about the MAC addresses of the end user frames belonging to the VLAN attached to them [10]. Another approach would be to use the labels that uniquely identify the edge nodes. Thus, encapsulation of end user frames can address the MAC address table explosion issue in the Metro domain.

The edge nodes of a metro region are access points (AP). Customer virtual LANs (C-VLAN) connect to different APs. A C-VLAN route is basically a tree that spans all locations of a given customer. A C-VLAN must use exactly one spanning tree for communication. A 12-bit VLAN identifier (VID), inserted into the end-user Ethernet frames, identifies each of the VLANs. The tagged frames belonging to a given VLAN will be forwarded across the network only to the end-hosts of the VLAN. This provides an implicit VLAN segregation and optimal network resource utilization. The traffic requirement of a customer is represented as a demand matrix (or access point matrix), which shows the traffic between AP pairs of a particular C-VLAN.

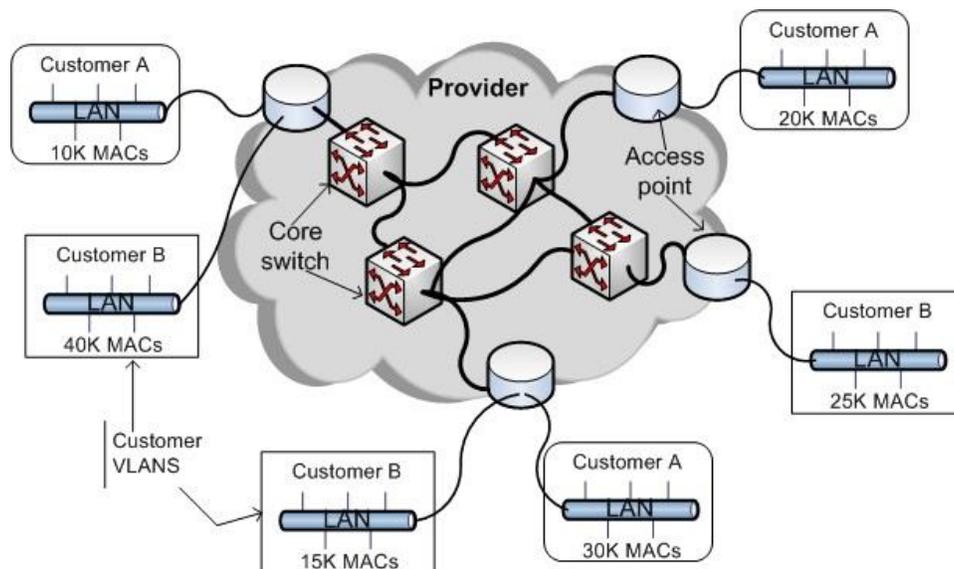


FIGURE 1: Metro Network.

4. RELATED WORK

Traffic engineering of Ethernet using spanning tree is a widely researched topic because of performance issues. QoS-aware Multiple Spanning Tree Mechanism [9] was proposed to address problems related to IEEE 802.1 standards and its extensions to the Spanning Tree protocol. They have proposed a simple and highly effective enhancement to the MST protocol to achieve high degree of QoS by keeping in perspective the different characteristics of the various traffic types. They use the traffic priorities defined in 802.1d standard for mapping VLANs to STs. For example the STs for best effort traffic is built based on the inter-face speed and end-to-end delay is the most important factor for building STs for multimedia traffic.

A fault tolerant multiple spanning tree protocol is proposed in Viking [12]. The Viking system provides at least two switching paths between any pair of end-nodes in two different STs, primary and backup switching paths. The path selection is based on cost assigned to each link. These

links are combined to form STs using path aggregation algorithm. The aggregation algorithm starts with an empty spanning tree set and select each path in the list to add to a spanning tree which does not form a loop. If there is no such tree then form a new spanning tree. One issue with this loop avoiding path selection procedure is that avoiding loops might leave out some paths which could have resulted in a better optimal tree. Viking relies on per-VLAN-spanning tree implementation of Cisco where there is a separate spanning tree running on every switch for every VLAN. This has the limitation on the number of VLANs the metro Ethernet can support due to the maximum VLAN tag size and the number of spanning trees a switch can support.

The article in [11] looks into the basic problems of traffic engineering in Metro Ethernet, such as load balancing, QoS-based protection, label-space management, evolving trends in traffic management at standard bodies and their implications. This proposed a grouping scheme that extends the current label space in the provider domain and allows for a large number of VLANs to be provisioned efficiently. Further, the issues of load balancing, multiple spanning trees, and interaction between grouping and bandwidth provisioning are discussed. It also addressed differentiated survivability in next-generation Ethernet and provided a novel scheme based on multiple spanning trees. The traffic engineering method we are proposing in this paper can use the C-VLAN grouping scheme described in this article to save VLAN tag space.

DiffPause [5] is a scheme for congestion control that is highly scalable, robust, and compatible with the assured forwarding of the differentiated services model for high-speed Metro Ethernet. The DiffPause introduces an Early Warning Threshold such that the upstream nodes can reduce the outgoing rate and throttle the aggressive traffic aggregates, thereby providing fair bandwidth allocation. This scheme takes advantage of per link-based back-pressure mechanism without any bandwidth reservation and is independent of the number of ongoing individual sessions, thus leading to high scalability.

In [13] we introduced a multiple spanning tree region construction algorithm which provides basic traffic engineering of Ethernet in the Enterprise domain. The proposed method supports dynamic nature of VLANs where the VLAN nodes can be added, moved, or removed without much effort. This method provides enough resources in the region to support protection from at least single failures. These multiple spanning tree regions are carefully engineered to provide better convergence time, reusability of VLAN tags, protection from failures, and optimal broadcast domain size.

A distributed scheme for fast restoration of Metro Ethernets, a fast failure recovery spanning tree scheme is proposed in [7]. This method restores lost facilities within 50 milliseconds irrespective of the network size.

5. TRAFFIC ENGINEERING USING OPTIMAL MULTIPLE SPANNING TREES

This optimal multiple spanning tree (OMST) method primarily uses customer traffic demands and network capacity to generate spanning trees. The approach complies with IEEE 802.1s MSTP and addresses six major components in constructing spanning trees. These are:

- How to assign weights to links in the network
- Root node selection for a spanning tree
- Parameters determining weight of the spanning tree
- How to map customers on to the tree
- Effectively distributing traffic across the network
- Avoid resource overloading

A C-VLAN is mapped to a spanning tree and frames forwarded to a tree can be identified by the VLAN ID. The C-VLAN grouping method suggested in [11] can used to save VLAN tags and limit the number of trees. Then the demand matrix for a group of C-VLANs will be the sum of all demand matrices. Thus, when spanning trees are constructed using our method, each customer

group is mapped to a single spanning tree. However, in this proposal we are not considering the grouping method and spanning trees are constructed for individual C-VLANs. The spanning trees are constructed such that the traffic is well distributed and no link or node is overloaded. Links for the spanning tree needs to be selected in such a way that it can handle the traffic demand generated by end nodes.

Dijkstra's algorithm can be a simple solution to this problem: build a spanning tree for each C-VLAN and then update weight on the links to reflect the new traffic. Since a customer is not using all the APs, we may not get the best path between C-VLAN's APs for the given traffic, even though the spanning tree is minimum weight. Further, the order in which links are selected, location of the root node and the order of spanning tree construction can affect the final solution. So our proposed solution considers all these requirements to build spanning trees. Though our approach might not deal with sudden load changes in the network, constant monitoring and simple reconfigurations can increase performance.

5.1 Dynamic Link Weight

Link weight is a function of the given link capacity and other parameters such as delay. To simplify the problem we are assuming that capacity and delay are the only parameters used in finding the link weights. However, this linear function can be extended with more parameters in the future. The delay itself can be divided into propagation delay and queuing delay. Queuing delay would increase as the traffic increases. Here we are using propagation delay which depends on the distance.

- Link weight is directly proportional to Delay.
- Link weight is inversely proportional to Capacity.

So a simple function is,

$$W_{ij} = D_{ij}/C_{ij}$$

Where W_{ij} is weight of the link between nodes i & j , C_{ij} is capacity of the link between node i & j , and D_{ij} is the delay between nodes i & j . Since we are trying to find the shortest possible paths (tree) between APs and equally utilize available resources (such as capacity), delay and link capacity are valid parameters. When the algorithm progresses available capacity on links will change, thus the link weight would also change. Hence adding more parameters to the function might need updating more variables during the course of algorithm execution.

5.2 Algorithm

The algorithm first sorts C-VLANs based on their traffic demand and builds a spanning tree for each customer starting from the customer with the highest demand. We use the highest-demand-first technique because inserting lower traffic demand into the network can be done without loss of much performance. Construction of each spanning tree is followed by calculating its aggregated weight. Aggregated weight is calculated using the formula given below, which is similar to the method suggested in [8].

$$ST_{AW} = \sum W_{i,j} \times T_{i,j}$$

Where, ST_{AW} = Aggregated weight of the spanning tree. $T_{i,j}$ = Traffic flowing through the link connecting nodes i and j . If this link is used in more than one path (that is more than one pair of access points are using the link) then $T_{i,j}$ is the sum of traffic on the link. $W_{i,j}$ = Weight of the link between nodes i & j .

Once the tree is constructed for a C-VLAN, the link weights on the graph are updated to reflect the new traffic. This process is continued for each C-VLAN in descending order of their demands. The solution steps are described in Fig. 2 and a detailed description follows it.

```

1. Sort C-VLANs in descending order, based on traffic demand.
2. For each C-VLAN in the sorted list do {
    a) Select an AP as the root node from the list of APs that this C-VLAN is using.
    b) Create the parent tree with the root.
    c) Calculate aggregate weight of the parent tree.
    d) For all APs the C-VLAN is using {
        i. Find the path with minimum weight between each pair of APs in the graph
        ii. Sort these paths in descending order of their aggregate weights
        iii. For each path do {
            1) Select the unprocessed path with highest weight and map on to the tree
            2) If (loops exist), then break loops
            }
        iv. Calculate the aggregated weight of the tree and we now have the final tree.
        v. Update the weights of links in master graph.
    }
}

```

FIGURE 2: Algorithm for constructing spanning trees

Sort the C-VLANs based on the demand and set the index to highest demand C-VLAN: C-VLANs are sorted in descending order of their demand. The aim is to construct spanning trees for C-VLANs in the order of their traffic demands. This way more demand would be satisfied using best optimal trees and the traffic would be well distributed on the network resulting in better performance.

Take next C-VLAN with the AP matrix and find a root node for the tree: The next selected C-VLAN will have the next highest demand requirement. After selecting a C-VLAN an access point is selected as the root for the tree. This can be done in three different ways:

- Select an AP from the list of APs the C-VLAN is using, which carries the lowest traffic.
- Select an AP from the list of APs the C-VLAN is using, which carries the highest traffic.
- Select a random AP.

We prefer the first option since there is a higher possibility that more traffic will flow through the root node of a spanning tree.

Create a spanning tree rooted at the root node: Create a minimum spanning tree rooted at the above selected node from the given graph. Let us call this the parent tree.

Calculate the aggregated weight of the tree: Aggregated weight of the tree is calculated using the previously defined function which considers both link weight and the traffic flowing on the link.

Find the shortest path between all APs of the C-VLAN and sort them in descending order of weights: Find the shortest path between each pair of APs (only for APs the C-VLAN is using) which satisfies the demand and assign weights to the path. A simple method like Dijkstra's algorithm can be used. Assign weights to each link in the path using the formula,

$$W_{i,j} \times T_{i,j}.$$

Where $T_{i,j}$ is the traffic flowing through the link and $W_{i,j}$ is the weight of the link. These paths represent the best routes the C-VLAN can have based on the traffic demand. Now list these paths in the descending order of their weights. For example:

AP1 $\leftarrow \rightarrow$ AP2, AP3 $\leftarrow \rightarrow$ AP1, AP2 $\leftarrow \rightarrow$ AP4

Take next path & map to the tree: Selection of the next path from the list can be done in two different ways:

- Select the path with the lowest weight first.
- Select the path with the highest weight first. This is a better choice because shorter paths would be applied towards the end of mapping.

Once we select a path, add its links to the tree. Since adding new links to the tree generate loops, the next task is to break loops.

Break loops: This step involves detecting the loop which is a very complex problem when looking for multiple loops. So to make this method simple, after adding each link from the above given path, algorithm breaks the loop. Adding one link to the spanning tree means a loop and this loop can be easily located. Finding multiple loops and breaking them in an optimal way is part of the future work. Now breaking loop can be done in two different ways. A simple approach would be to delete one of the links other than the newly added link. However this may not result in an optimal solution because some links may cause an increase in weight of the tree killing the whole purpose of mapping paths onto the tree. An optimal solution is described in Fig.3.

```

1) Delete the link which contributes more weight to the tree.
  a) For the loop {
    i) Determine the links which formed the loop.
    ii) For each link check {
        (1) If this link is removed, do other links in the loop have enough capacity to
            handle the excess traffic? Then this link is a candidate, otherwise this link
            cannot be removed.
    }
    iii) Once we get a list of candidate links for removal, remove the link which
        contributes the highest weight to the tree. In other words, after breaking each link
        in the loop we will get a different tree. Now keep the tree with the lowest weight.
  }
    
```

FIGURE 3: Algorithm for breaking loops.

Update weights on the graph: Once all the paths mapped on to the tree, the C-VLAN gets the final tree. Next step is to update weights on the links of the graph based on the traffic through the links of the tree. Now the capacity of each link in the graph will change to,

$$C_{ij} = C_{ij} - T_{ij}$$

Where, C_{ij} is the capacity of link (i, j) in the graph and T_{ij} is the traffic of link (i, j) in the tree. This update will occur only for the links shared between the tree and graph. Once the capacity is updated, we need to recalculate the weight on each link in the graph using the weight formula defined in subsection 5.1.

This algorithm completes with optimal multiple spanning trees for C-VLANs. Now the spanning tree assigned to each C-VLAN can be further tuned by blocking ports which are leading to access points that are not used by this C-VLAN. This avoids traffic leakage and thus prevents unnecessary wastage of resources.

6. PERFORMANCE ANALYSIS

Performance evaluation of this algorithm is based on complexity analysis and simulation studies. This section describes results of our experiments.

6.1 Complexity Analysis

Since Dijkstra's algorithm is used in OMST, the complexity analysis is basically based on the complexity of Dijkstra. The complexity of sorting C-VLANs is simply the complexity of sorting algorithm we use, which is $O(n \log n)$ for n C-VLANs. The part which generates the initial spanning tree, the parent tree, is the first complex operation on the graph. The source based spanning tree algorithm on graph with V vertices and E edge has complexity $O(E + V \log V)$ using

the Fibonacci heap. Calculating aggregate weight of the spanning tree is simple and involves a smaller part of the graph and hence does not contribute much to the complexity.

Finding shortest paths between access points of a C-VLAN in the original graph is the next complex part of the algorithm. Since mapping of paths onto a spanning tree is done one link at a time, this process is less complex compared to finding shortest paths. Shortest paths from one node to all other nodes in the graph can be derived using Dijkstra's method. This algorithm has the same complexity as the spanning tree construction method, which is $O(E + V \log V)$ using Fibonacci heap. Though we need only paths between access points, the algorithm may end up finding shortest paths to all nodes in the graph. Since we need to find the shortest path from each access point to all other access points, the total running time is $O((A-1)(E + V \log V))$, which is $O(A(E + V \log V))$. The number of access points used by the client is represented using the variable A . So the overall complexity of this algorithm to generate a spanning tree can be represented using $O(A(E + V \log V))$.

6.2 Experimental Results

Inputs to the simulation are the network topology in matrix format and traffic demands for each C-VLAN. Network topologies and traffic demands are randomly generated based on probability functions. For every topology we used, the bandwidth of links was set to 100 Mbps. Propagation delay is randomly selected between 0.2 and 1.0 milliseconds. Output of the simulation is in the form of spanning trees. Then we analyze the performance of these trees with trees constructed using Dijkstra's method.

Fig. 4 shows the performance of a 22 node network. Two parameters are selected to show the performance, aggregated weight (Ag_W) of spanning tree and average weighted length (Av_WL) of paths within the tree. To calculate average weighted length, we first find the weighted length of all paths between AP pairs for this particular C-VLAN. The average of these values gives us the final average weighted length. This parameter gives us a clear idea about the delay involved and broadcast domain size. These parameters are compared with spanning trees generated using simple tree construction methods. The resulting network has 13 access points and a performance matrix is collected for 5 C-VLANs. Fig. 5 shows the performance matrix for different networks. Parameters are selected as the average for each network type.

Fig. 6 shows the traffic distribution on a heavily connected 12 node (40 links) network with 100 Mbps bandwidth based on the following approaches:

1. Single spanning tree for connecting all the nodes in the network. This spanning tree will be used by all C-VLANs in the network.
2. MSTs for C-VLANs. However the link capacities are not updated after constructing trees, resulting in almost similar trees.
3. MSTs are constructed and link capacities are updated after building a tree for the C-VLAN. This is very much similar to our approach, except that finding shortest paths and mapping onto the tree is not applied. Even though we haven't seen this kind of an approach in any literature for Metro Ethernet, we used this to prove the efficiency of our technique.
4. The optimal multiple spanning tree approach.

From the graph it is clear that algorithm 1 and 2 caused some links to overload and used less than 30% of links. Even though we used multiple spanning trees in the second approach, the result is close to the single spanning tree. Since traffic is not reflected in the graph after building spanning trees, choosing different root nodes for each C-VLAN will not make much difference to the structure of spanning trees. However, the third technique and our approach are very much similar. In our algorithm the maximum traffic on a link was less than 50% of link capacity, and less than 10% of links with no traffic. In the third technique (MST with traffic updates) 25% of links were not used and up to 80% capacity of some links was used. In conclusion our approach

can indeed increase the performance of spanning trees and keep traffic distributed efficiently in a metro domain.

C-VLANs	% of decrease in	
	Ag_W	Av_WL
1	32	20
2	46	39
3	32	29
4	39	32
5	38	39
Average	37.4	31.8

FIGURE 4: Performance of a 22 node network.

# of nodes	# of access points	# of C-VLANs	Ag_W	Av_WL
10	6	4	24.5	21.5
15	10	6	32.8	27.2
20	12	8	30.3	24.4
25	14	10	32.1	26.9
30	17	13	34.4	27.9

FIGURE 5: Performance matrix for 5 different networks.

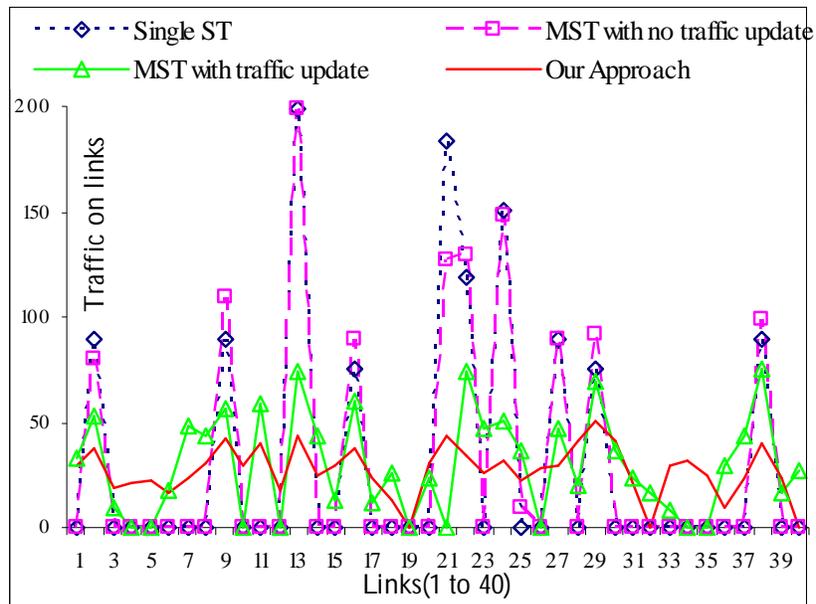


FIGURE 6: Traffic distribution in a 40 link network.

7. FUTURE DIRECTIONS

In this section we introduce a new traffic demand specification model that will be used in the future to evaluate the method presented in the previous section and other traffic engineering models for Metro Ethernet.

7.1 The Augmented Hose Model

This new model for traffic specification in Metro Ethernet is based on the observation that the customer sites for some of C-VLANs are usually clustered into components. As a result, the traffic can be classified as inter and/or intra-component traffic. The cases where the distribution of traffic is dominated by intra-component traffic are of interest for further optimization. In the case that there is no inter-component traffic, these components can be implemented as separate C-VLANs. However, if there is a percentage of the traffic that is to be delivered in the native technology (e.g., Ethernet) between these components, the hose model suffers from over-provisioning the bandwidth for the C-VLAN. Augmented hose model uses measurements obtained from the customer to greatly enhance bandwidth utilization in the provider network.

Consider for example the following hose model specification for the network in Fig. 7. Consider the bandwidth reservation for directed link (1, 2). Using the hose model, the bandwidth reserved is minimum value of: 1) the aggregate ingress bandwidth from A, B, and C, and 2) the aggregate egress bandwidth for D, E, and F. This value is computed to be $\min(222, 211) = 211$.

Suppose now that measurements at customer points A, B, and C indicate that:

- At site A 85% of traffic goes to {B, C}
- At site B 80% of traffic goes to {A, C}
- At site C 65% of traffic goes to {A, B}

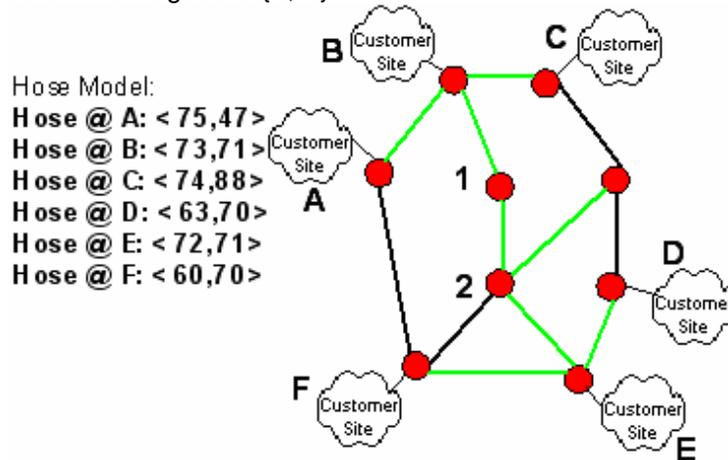


FIGURE 7: Sample network illustrating the benefits of hose augmentation.

This information can be communicated to the provider at the time of provisioning bandwidth for the C-VLAN in order to reduce the bandwidth reservation. For the directed link (1, 2), the bandwidth reserved is now equal to $(1-0.85)*75 + (1-0.8)*73 + (1-0.65)*74 = 51.75$.

Consider Fig. 8, where we show a C-VLAN $A = \{0, 1, 2, 3, 4, 5, 15, 16, 17, 18, 19, 20\}$. The C-VLAN is partitioned into two components: North and South.

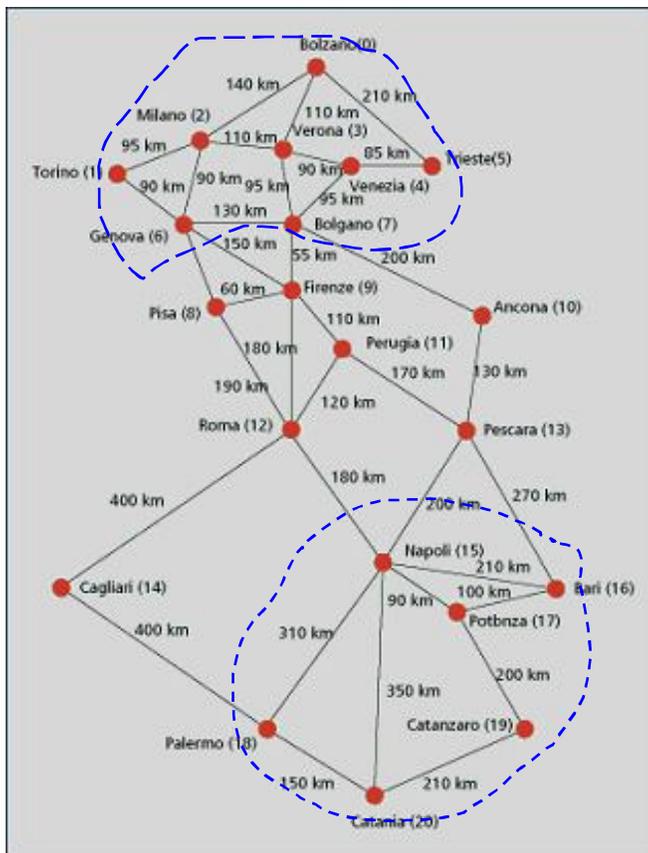


FIGURE 8: Network with a C-VALN composed of two components.

Fig. 9 shows the benefits of hose augmentation as a function of the inter-component traffic and using one ST. We notice that as the traffic increases between these components, the bandwidth savings using hose augmentation decreases. As expected, when the inter-component traffic reaches 50% of the traffic, the augmented hose model performance approaches to that of a regular hose model. However, under our assumption of non-zero inter-component traffic, we observe gains of about 25% in the case of 20% inter-component traffic. As Fig. 10 shows, these savings even increase when the number of STs is increased. The figure shows for example, that we have 36% savings in bandwidth in the case of 20% inter-component traffic and using 21STs.

The multiple spanning tree construction algorithm presented in section 5 will be analyzed in the future using this new traffic specification model. Further, in our subsequent papers we will study the performance of Augmented Hose and use this method for analysis of other traffic engineering methods.

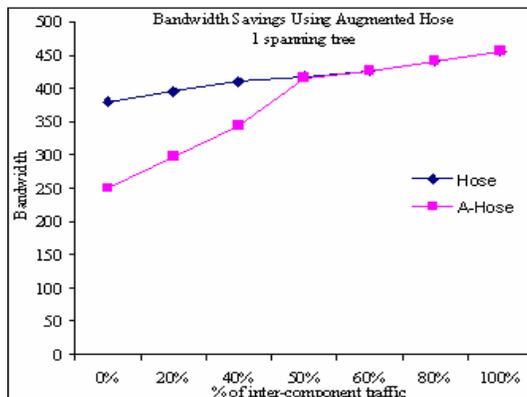


FIGURE 9: Enhancements to Bandwidth using 1 tree.

Fig. 9.

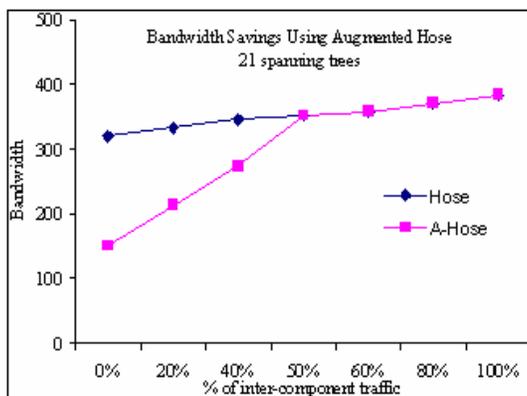


FIGURE 10: Enhancements to Bandwidth using 21 trees

8. CONCLUSIONS

Construction of spanning trees using well-known schemes such as Dijkstra’s algorithm do not specifically address issues such as load sharing, efficient utilization of resources, QoS and reliability. For example, using Dijkstra’s method could result in trees sharing a common pool of links. These links could get overloaded at the same time leaving some links idle. When considering a network topology such as Metro Ethernet, QoS is an important factor which depends on customers, their traffic demand and how well the given traffic can be distributed. In the paper we developed an iterative approach which uses traffic demand and dynamic link weight characteristics to fine tune the initial spanning tree to ensure that the resulting tree would satisfy the demand requirements and offer the best possible performance. Through a number of simulation experiments, we concluded that this approach can indeed increase the performance of spanning trees and keep traffic distributed efficiently in a metro domain. Further, we introduced a new traffic specification model, augmented hose, which uses measurements obtained from the customer to properly utilize bandwidth in the provider network. Through experimental studies we observed that this method performs better than hose model when inter-component traffic is below 50% of the traffic. In the future we will use this model to study performance of traffic engineering methods including the algorithm we suggested in this paper.

9. REFERENCES

1. IEEE 802.1q, “Virtual Bridged Local Area Networks”, 1998.

2. IEEE 802.1d, "Media access control bridges", 1998.
3. IEEE 802.1w, "Rapid spanning tree configuration", 2002.
4. IEEE 802.1s, "Standards for Local and metropolitan area networks", 2002.
5. A. Ge and G. Chiruvolu: "DiffServ-Compatible Fair Congestion Control through Extended Pause (DiffPause) for Metro-Ethernet", IEEE ICC, 2004.
6. "Metro Ethernet networks - A technical overview". Metro Ethernet Forum white paper, December 2003.
7. M. Padmaraj, S. Nair, M. Marchetti, G. Chiruvolu, and M. Ali, "Bandwidth sensitive fast failure recovery scheme for Metro Ethernet", Computer Networks, Volume 52, Issue 8, 12 June 2008, pp. 1603-1616
8. Y. Li , Y. Bouchebaba, "A New Genetic Algorithm for the Optimal Communication Spanning Tree Problem", Selected Papers from the 4th European Conference on Artificial Evolution, November, 1999, pp.162-173
9. Y. Lim, H. Yu, S. Das, S. Lee, M. Gerla. "QoS-aware multiple spanning tree mechanism over a bridged LAN environment". IEEE Global Telecommunications Conference, 2003. pp. 3068-3072.
10. I. Hadzic,"Hierarchical MAC address Space in Public Ethernet Networks", IEEE Globecom, 2001, pp.1563-1569.
11. M. Ali, G. Chiruvolu, and A. Ge, "Traffic Engineering in Metro Ethernet", IEEE Network, vol. 19, no. 2, March 2005.
12. S. Sharama, K. Gopalan, S. Nanda, and T. Chiueh, "Viking: A multi-spanning-tree Ethernet architecture for metropolitan area and cluster networks", IEEE INFOCOM 2004, March 2004.
13. M. Padmaraj, S. Nair, M. Marchetti, G. Chiruvolu, and M. Ali, "Traffic Engineering in Enterprise Ethernet with Multiple Spanning Tree Regions", In proceedings of IEEE/IEE International Conference on High Speed Networks, August 2005.
14. M. Padmaraj, "Quality of Service in Metro Ethernet", PhD Thesis, Southern Methodist University, Dallas, Texas, May 2006

Dynamic Hybrid Topology Design for Integrated Traffic Support in WDM Mesh Networks¹ (Invited Paper)

Mina Youssef

*Electrical and Computer Engineering
Kansas State University
Manhattan KS, 66502 USA*

mkamel@ksu.edu

Baek-Young Choi

*Computer Science and Electrical Engineering
University of Missouri
Kansas City MO, 64110 USA*

choiby@umkc.edu

Caterina Scoglio

*Electrical and Computer Engineering
Kansas State University
Manhattan KS, 66502 USA*

caterina@ksu.edu

Eun Kyo Park

*City University of New York
Staten Island NY, 10314 USA*

eun.park@csi.cuny.edu

Abstract

The future Internet will require the transport of a wide range of services including high bandwidth one-to-many applications, with a dynamic interconnection of devices. WDM layer support realizes such services in a transparent, reliable and efficient way. Most of the recent studies have been focused on efficiently building and configuring light-paths for unicast or light-trees for multicast in isolation, and do not take existing traffic demands and configuration into consideration. In this paper we consider a dynamic design problem of integrated traffic in a realistic WDM mesh network. In such a network, new traffic demands of either multicast and/or unicast are supported dynamically in the presence of an existing mixture of traffic. The amount of bandwidth per wavelength is abundant, while the wavelengths and light splitting capabilities on WDM switches are limited. Using subwavelength sharing among traffic demands of unicast and multicast, we build a hybrid virtual topology that exploits both existing light-trees and light-paths. By optimizing WDM resources in addition to resource sharing with existing unicast and multicast demands, we truly maximize the WDM layer capability and efficiently support more traffic demands. We validate the efficiency of our approach with extensive simulations on various network topologies.

Keywords: WDM Networks, Dynamic Topology Design, Unicast, Multicast.

¹ An earlier version of this work has appeared in the Proceedings of International Conference on Computer Communications and Networks (ICCCN)'08.

1. INTRODUCTION

As the Internet traffic continues to grow exponentially and Wavelength Division Multiplexing (WDM) technology matures, the WDM network, with tera-bits per second bandwidth links, becomes a dominant backbone for IP networks. Continuously emerging bandwidth-intensive applications in current and future Internet, however, present the need of efficient and scalable support in an underlying network. Particularly, it is increasingly important for the WDM layer to facilitate, in an efficient and scalable manner, high bandwidth one-to-many applications such as web cache updating, transfer of software upgrade, transfer of video, audio, and text data of a live lecture to a group of distributed participants, whiteboard and teleconferencing [1], [2].

In-network replication or branching of multicast traffic may be done in either an optical WDM domain or an electronic IP domain. In IP over WDM networks, mere IP layer multicasting is not efficient enough without the support of the WDM layer. Enabling multicasting at the WDM layer has clear advantages. First, with the available optical layer resources (e.g. light splitters, wavelength converters and wavelengths) we can utilize a more efficient in-network replication via an optical layer multicast tree than an IP layer multicast tree created without understanding of underlying physical network. With the inherent light splitting capability of optical switches, it is more efficient to do light splitting than copying IP datagrams in an electronic domain. IP multicasting creates copies of data packets at intermediate routers from an optical into an electronic domain and then converts them into an optical signal, called O/E/O conversion. On the intermediate non-member nodes, this process introduces extra-delays and consumes IP resources unnecessarily. Second, performing multicast in optics is desirable and secure, as it provides consistent support of format and bit-rate transparencies across both unicast and multicast transmissions without requiring the data format to be known to the upper layer.

With no WDM layer multicast support, IP multicast sessions can be realized by having an IP router on a multicast tree make copies of a data packet in an electronic domain and transmit another copy to the downstream routers. However, this requires O/E/O conversion of every data packet at intermediate routers on the tree incurs extra-latency and requires the data format to be known to the upper layer. When IP multicast is supported via light-paths (i.e., WDM multiple unicasts), it avoids the delay of O/E/O conversion at intermediate nodes. However, this scheme is not scalable with a large number of multicast members and number of groups in the network. An ideal approach to supporting multicasting at the WDM layer is to create multicast trees in the optical layer directly. This can be achieved by light-tree [3], which uses optical light splitting at intermediate nodes as needed in order to replicate an optical signal to multiple downstream paths. It minimizes the use of wavelengths and bandwidth as well as O/E/O conversion. The minimal use of physical resources enables us to support larger sessions as compared to other approaches.

In supporting large multicast groups and members, however, the WDM layer multicast using wavelengths only may not be feasible, especially due to the limited number of wavelengths. Based on the observation that a high bandwidth capacity per wavelength is abundant enough to be shared by multiple traffic demands, [4] exploited existing unicast light-path to build light-trees for the WDM layer multicast with the subwavelength sharing approach. However it assumed that there was no multicast session set up initially and only existing unicast light-paths and available physical wavelength were considered to build light-trees.

In this work, we extend the idea of subwavelength sharing to existing multicast demands. We provide a more flexible and practical framework where existing light-trees could be shared as well. A technical challenge in subwavelength sharing of light-trees is that a light-tree is designed for a specific group and a wavelength is not desirable to be shared by different groups with different set of members. It is because the sharing of a light-tree for other multicast groups will cause all the destinations on the tree to receive data packets sent on the tree unnecessarily. We minimize the excess traffic due to light-tree sharing while optimizing all the resources. We bound the degree of

sharing in order to meet the QoS of existing traffic. Our solution is aimed for a *dynamic* environment where new traffic demands need to be satisfied without disturbing the service of existing traffic. Our main contribution is that when new traffic demands, either unicast or/and multicast, arrive, they can be supported incrementally, taking available resources and existing traffic demands into consideration. If desirable or physical resources are lacking, we allow the use of existing light-paths or light-trees in conjunction with a possibly new (partial) light-tree. A challenging issue when light-trees are shared, is to minimize excess traffic incurred by the different multicast demand from the existing tree(s). In order to address that, we formulate an optimization problem that includes the overhead of excess traffic. We find that this hybrid (light-trees and light-paths) virtual topology design enables us to establish multicast trees when it would otherwise be impossible. Thus, more traffic demands are supported under practical network condition of limited wavelength constraints. Furthermore, our solution maximally utilizes the available resources of existing light-paths whose traffic demand does not reach full wavelength capacity, but has bounded a degree of sharing for QoS of existing traffic.

The idea of using existing multicast light-trees and/or unicast light-paths gives enormous flexibility in terms of a dynamically integrated future Internet traffic environment, compared to a pure light-tree approach. By optimizing the WDM layer multicast as well as resource sharing with existing mixture of unicast and multicast demands, we truly maximize the WDM layer capability under a practical environment.

The remainder of this paper is organized as follows. In Section 2 we provide the background of our study. We summarize related works in Section 3. In Section 4, we formally state the problem and discuss our approach. The evaluation and validation of our scheme is presented in Section 5. We conclude the paper in Section 6.

2. Background

In Wavelength Division Multiplexing (WDM) networks, each directional fiber optical link is partitioned into multiple data channels, each of which operates on a separate wavelength, permitting high bandwidths. Routers or switches are connected via semi-permanent optical pipes called 'light-paths' that may extend over several physical channels via wavelength routing. At intermediate nodes, incoming channels belonging to in-transit light-paths are transparently coupled to outgoing channels through a passive wavelength router, avoiding the unnecessary IP layer interruption with O/E/O conversion. Meanwhile, at a node terminating light-path, the incoming signal from the channel is converted to the electronic domain so that packets can be extracted and processed and may be retransmitted on an outgoing light-path(s) after electronic IP routing. The concept of a light-tree can be extended using optical light splitters, in order to replicate an optical signal to multiple downstream paths. The light-paths and/or light-trees establish a virtual topology on top of a physical topology made of optical fibers and switches/routers. A virtual topology configuration is constrained by a number of physical resource limitations: 1) The establishment of each light-path requires the reservation of WDM channel on the physical links along the paths and the number of available the WDM channels are limited on a link. 2) The number of transmitters and receivers at each node limits the number of light-path initiating and terminating on the node. 3) The maximum length of a light-path without signal regeneration may be limited by the signal attenuation along the light-path. Therefore, optimizing the use of WDM network resources is a crucial task in order to process traffic demand efficiently.

The concept of wavelength sharing has been proposed before in the context of unicast or multicast individually. The work in [4] was the first that proposed the sub-wavelength resource sharing among unicast and multicast traffic demands and provided a general solution under practical constraints. Traffic grooming concerns grouping of small flows into a single wavelength, that can be processed and routed as one entity. Our work addresses the issue of sub-wavelength sharing with or without traffic grooming. The future Internet will involve interconnections of large number of devices that are

aggregated at access networks. High bandwidth WDM mesh networks at the backbone can be better utilized with sub-wavelength sharing in the wavelength assignment and routing. Sub-wavelength sharing can be used in various network business models. Different network business models can be considered as below [5]:

- Model A: An ISP that owns the network from the "ground up" (i.e., to the duct) and only delivers IP-based services.
- Model B: The business owns the layer-one infrastructure and sells services to customers who may themselves resell to others. It serves as the carriers' carrier and offers wholesale services to ISPs
- Model C: An ISP that leases fiber or transport capacity from a third part, and only delivers IP-based services.
- Model D: The business is a bandwidth broker. It provides "match-making" by enabling a variety of ISPs (model 3) to lease bandwidth from a variety of network operators (model 2).

Models B and C are complementary whereas both are integrated in case of model A. Our work can be considered as the issues of model A, B or D.

3. Related work

In recent years, many studies have been conducted in regards to the problem of designing virtual (or logical) topology for WDM networks. The problem of multicasting for IP over WDM networks can be decomposed into two subproblems, namely multicast-tree design, and routing and wavelength assignment (RWA) for the designed multicast-tree. As to the problem of multicast-tree design, two classes of approaches have been taken; namely, optimization and heuristics.

The multicast-tree design problem has been modeled often as a linear optimization problem to minimize the O/E/O conversions, as it is the main bottleneck in utilizing the true potential of optical networks. Other objectives, such as minimizing the average hop count or average number of transceivers used in the network [3] and the total link weight of the light tree [6] have also been used. In [7], the problem of optimal virtual topology design for multicast traffic is studied using light-paths. The authors aim to minimize the maximum traffic flowing on any light-path in the network while designing the logical topology for the multicast traffic. In the same work, the authors have presented several heuristics for topology design such as Tabu search, simulated annealing. Linear optimization techniques have been also used for the unicast single shortest virtual topology design [8], RWA [9], [10], restoration and reconfiguration [11], [12], [13] problems in WDM networks. Several heuristics have been proposed to design the multicast tree in WDM networks. Although a minimum Steiner tree [14] (which is obtained by solving Integer Linear Programming ILP) is more desirable, finding one for an arbitrary network topology is an NP-complete problem [15], thus heuristics are often used to obtain a near-minimum cost multicast tree. Authors in [16] have presented four heuristic algorithms: namely, Re-route-to-Source & Re-route-to-Any, Member-First, and Member-Only, for designing a multicast forest for a given multicast group. The minimum spanning tree (MST) heuristic or the shortest-path tree (SPT) heuristic [14] are also commonly used for designing the multicast tree. Authors in [17] have also presented two such algorithms, Breadth First Search (BFS) and a dynamic and incremental tree construction algorithm. Given an existing multicast tree with a large number of members, new member nodes perform an operation of join/graft similar to CBT [18] and DVMRP [19]. Once a multicast tree is designed, it can be implemented with either wavelength-routing [20], [3] or Optical Burst/Label Switching (OBS/OLS) [21], [22]. In the former case, multicast data will be switched to one or more outgoing wavelengths according to the incoming wavelength that carries it. That is a wavelength needs to be reserved on each branch of a multicast tree. In IP over WDM multicast using label switching, multicast label switched paths are set up first. Afterwards, only the optical labels carried by the bursts need O/E conversions for electronic processing, whereas the burst payload always remains in the optical domain at intermediate nodes. The major disadvantage of the wavelength routing approach is that it may not utilize the bandwidth efficiently in case traffic demand is not up to wavelength capacity. It also has large setup latency and it is not efficient under bursty traffic conditions. Meanwhile, with OBS/OLS, a burst dropping (loss) probability may be potentially significant in a highly loaded OBS

network which can lead to heavy overheads such as a large number of duplicate retransmissions in IP layer. In addition, it may take a longer time for an end host to detect and then recover from burst dropping (loss) [23]. In [24], the authors introduced a light-hierarchy graph renewal and distance priority light-tree algorithm (GRDP-LT), which was proposed to improve the light-trees quality for any multicast under light splitting constraints. In a light-hierarchy, cycles are allowed, which is different from the light-tree where no cycle exists in the structure.

In a closely related work [4], a hybrid multicast topology was first designed given only light-paths and physical links. The work is to support better utilize the bandwidths of wavelengths for new multicast traffic; assuming only unicast traffic was supported initially. In this work, we extend the concept of sub-wavelength sharing to existing light-trees as well, in addition to light-paths, and build a hybrid virtual topology which exploits both light-trees and light-paths. We optimize WDM resources for new traffic while keeping the a priori configurations for the existing traffic, so that their performances are not disturbed.

4. Problem formulation

In this section, we formally state the problem of designing hybrid virtual topology for given set of multicast demands using available physical, light-path, and light-tree topologies. The network channel resources are the available wavelengths on the physical links, the available degree of sharing of light-paths, C^v and the available degree of sharing of light-trees, C^t for other traffic demands. The number of available wavelengths is bounded by the physical resources, and the degrees of sharing of light-paths and light-trees are bounded for the QoS of existing traffic. Next, we formulate our objective function and discuss the constraints required for the design problem.

4.1 Objective function

$$\min \sum_s \sum_m \sum_n \left(w_{m,n} M_{s,m,n} + \alpha_{m,n} Y_{s,m,n} + \sum_i \beta_i T_{s,i,m,n} \right) \quad (1)$$

The objective is to minimize the cost of the selected hybrid topology components, namely physical wavelengths (M), light-paths (Y) and light-trees (T). In the objective function, m and n indicate a source and a destination of the corresponding link, respectively, and s indicates the corresponding multicast session. The cost components, M, Y, and T are binary variables and are weighted by parameters of w , α , and β . New unicast traffic demand can be considered a special type of multicast where it has only one destination. Thus our objective function adds all the cost of physical links, light-paths, and light-trees for each link and each session, for satisfying either unicast or multicast traffic demands. The weights can be assigned depending on the deployment or operational costs. For example, in our evaluations later, we have used the weights of the physical links to reflect the preference to a path with minimum hops. That is, all physical links have the same weights. The weight of a light-path is set to the sum of weights of physical links that were used to design the light-path. Then the weight of a light-tree is the sum of physical links building the tree. The objective function with those weights indirectly minimizes the number of non-destination intermediate nodes between different physical links and light-paths. It also selects light-trees that have a minimum number of fortuitous nodes [25], where a node in a light-tree is a fortuitous destination if it is not a member in that session but receives an excess copy of multicast session packets [26] due to the configuration. Our formulation minimizes the unnecessary excess traffic to fortuitous nodes, since the cost of a light-tree includes the cost of all individual links.

Data Input	Definition
$p_{m,n}^{adjacent}$	Boolean matrix. Represents the adjacency of the physical topology.
$V_{m,n}^{adjacent}$	Boolean matrix. Represents the adjacency of the existing virtual light-path topology.
$tree_{i,m,n}$	Boolean matrix. Represents the adjacency of the existing virtual light-tree topology i .
$source_s$	A source node number for the new multicast session s .
$session_{s,m}$	Boolean value. Represents if node member m belongs to the multicast session s .
$w_{m,n}$	Cost of using a physical link m, n .
$\alpha_{m,n}$	Cost of using an existing light-path m, n .
β_i	Cost of using a existing light-tree for multicast session i .
$C_{m,n}^{pavailable}$	The number of available wavelengths on the physical link between node m and n .
C^v	Degree of sharing on an existing light-path.
C^t	Degree of sharing on an existing light-tree.

Decision variable	Definition
$member_{s,m}$	Boolean value. $member_{s,m} = 1$ if node m is a session member or an existing light-tree member, for new session s .
$M_{s,m,n}$	Boolean value. $M_{s,m,n} = 1$ if the physical link m, n is used for the multicast session s .
$Y_{s,m,n}$	Boolean value. $Y_{s,m,n} = 1$ if the light-path m, n is used for the multicast session s .
$T_{s,i,m,n}$	Boolean value. $T_{s,i,m,n} = 1$ if the physical link m, n in an existing light-tree i is used for the new multicast session s .
$\gamma_{s,i}$	Boolean value. $\gamma_{s,i} = 1$ if an existing light-tree i is selected to satisfy a new multicast session s .
$f_{s,m,n}$	Flow accommodations ($\sum_m member_{s,m}$) from the source node of session s over different physical links m, n .
$y_{s,m,n}$	Flow accommodations ($\sum_m member_{s,m}$) from the source node of session s over different light-paths m, n .
$t_{s,i,m,n}$	Flow accommodations ($\sum_m member_{s,m}$) over light-tree i to satisfy the multicast demand s .

TABLE 1: Data input and decision variable definitions

4.2 Constraints

We discuss a number of constraints to create hybrid topologies in this subsection. We carefully set the constraints so that the number of variables and equations necessary to be minimized, and be solved for a relatively large networks. The constraints can be of three major types, namely, constraints for light-tree design, flow conservation, and resource bounds. Data input and decision variables are defined in Table 1.

1) Constraints for light-tree generation:

The following set of equations are to choose light-tree(s) i to satisfy a multicast session s . They ensure that the variable $T_{s,i,m,n}$ will maintain the topology of the selected light-tree with the

necessary nodes. These equations will also ensure that routing of any traffic flowing on the light-tree i , $T_{s,i,m,n}$, is feasible.

$$T_{s,i,m,n} \leq tree_{i,m,n} \quad \forall s, i, m, n \quad (2)$$

$$\sum_m \sum_n T_{s,i,m,n} = \sum_m \sum_n v_{s,i} tree_{i,m,n} \quad \forall s, i \quad (3)$$

$$T_{s,i,m,n} + T_{s,i,k,n} \leq 1 \quad \forall s, i, m, n, k, k \neq m \quad (4)$$

$$T_{s,i,m,n} + T_{s,i,k,n} + M_{s,m,n} + M_{s,n,m} + Y_{s,m,n} + Y_{s,n,m} \leq 1 \quad \forall s, i, m, n \quad (5)$$

$$\sum_i \sum_n (T_{s,i,m,n} + T_{s,i,n,m}) + session_{s,m} \leq member_{s,m} C \quad \forall m, s \quad (6)$$

Eq. (2) is to ensure that if a link m, n is part of the selected existing light-tree i to satisfy the new multicast demand s , all links in the existing tree should be part of the new multicast tree. Eq. (3) with Eq. (2) guarantees that the variable $T_{s,i,m,n}$ will include all the links of the selected light-tree i to support the multicast session s . Eq. (4) is to avoid the situation that a node in the light-tree i receives more than a packet for the same multicast session s . Eq. (5) ensures that different resources (channels) cannot be used more than one time for the same session s on the same link m, n (or light-path m, n). It also eliminates the case where the traffic flows on the same link m, n in different directions (m, n and n, m) for the same session s . Eq. (6) enables that the members of the new light-tree to include the destinations of used existing light-trees as well as the original destinations of multicast session s . C is a big positive number.

2) Constraints for light-tree generation:

$$\sum_n \left(y_{s,source_s,n} + f_{s,source_s,n} + \sum_i t_{s,i,source_s,n} \right) \leq \sum_k member_{s,k} - 1 \quad \forall s, n \neq source_s \quad (7)$$

$$\sum_n \left(y_{s,m,n} - y_{s,n,m} + f_{s,m,n} - f_{s,n,m} + \sum_i (t_{s,m,n} - t_{s,n,m}) \right) \leq member_{s,n} \quad \forall s, n \neq source_s \quad (8)$$

$$y_{s,m,source_s} + M_{s,m,source_s} + T_{s,i,m,source_s} = 0 \quad \forall s, i, m \quad (9)$$

$$t_{s,i,m,source_s} = 0 \quad \forall s, i, m \quad (10)$$

First, Eq. (7) makes sure that the source node of each session $source_s$ sends the traffic demand to all the destinations using the virtual and physical links and the light-trees attached with the source node. The destinations are the multicast session s members, intermediate nodes and the light-tree members if an existing light-tree is selected to satisfy the demand for the session s . Eq. (8) represents the flow balance equation. Eq. (9) ensures that the source node of session s will not receive a multicast packet from the same session. The source node has no traffic demand for each multicast session for the light-tree flow as shown in Eq. (10).

3) Constraints for resource bounds:

$$f_{s,m,n} \leq M_{s,m,n} C \quad \forall s, m, n \quad (11)$$

$$\sum_s M_{s,m,n} \leq C_{m,n}^{pavailable} \quad \forall m, n \quad (12)$$

$$y_{s,m,n} \leq Y_{s,m,n} C \quad \forall s, m, n \quad (13)$$

$$\sum_s Y_{s,m,n} \leq C^v \quad \forall m, n \quad (14)$$

$$t_{s,i,m,n} \leq T_{s,i,m,n} C \quad \forall s, i, m, n \quad (15)$$

$$t_{s,i,m,n} \geq T_{s,i,m,n} \quad \forall s, i, m, n \quad (16)$$

$$\sum_s v_{s,i} \leq C^t \quad \forall i \quad (17)$$

$$M_{s,m,n} \leq P_{m,n}^{adjacent} \quad \forall s, m, n \quad (18)$$

$$Y_{s,m,n} \leq V_{m,n}^{adjacent} \quad \forall s, m, n \quad (19)$$

$$\sum_s (Y_{s,m,n} + M_{s,m,n}) \leq member_{s,m} C \quad \forall s, m \quad (20)$$

We assume that the maximum number of physical wavelengths of an optical link is C . We also assume the degree of sharing of a light-path, and the degree of sharing of a light-tree are limited by C^v and C^t , respectively, in order to ensure the quality of service of traffic performance.

In Eq. (11), the physical link $M_{s,m,n}$ is used to support the multicast session s , multicast traffic can be sent over it. Eq. (12) constrains the number of multicast sessions to number of available channels on the physical link m, n . Eq. (13) is similar to Eq. (11) but used for light-paths. The number of multicast sessions that can use light-path m, n is constrained according to the degree of sharing C^v in Eq. (14). Eqs. (15) and (16) force the traffic to flow on the links m, n of the selected light-tree i and to be in the proper direction. Eq. (17) constrains number of multicast sessions that can use light-tree i to C^t where C^t is the degree of sharing the light-tree. Eqs. (18) and (19) constrain the selection of a physical link and a light-path between the existing ones. Eq. (20) guarantees that all the intermediate nodes of the selected physical links and light-paths are included in $member_{s,m}$.

In summary, the above constraints of light-tree generation, flow-conservation and resource bounds enable us to use light-trees and light-paths within the resources available and to meet the given multicast demands. Note that the composite objective function of physical and hybrid virtual topology resources given in Eq. (1) provides a generic abstraction for capturing a wide variety of

resource and performance optimization such as wavelength, hop count and delays, by controlling the weights of the objective.

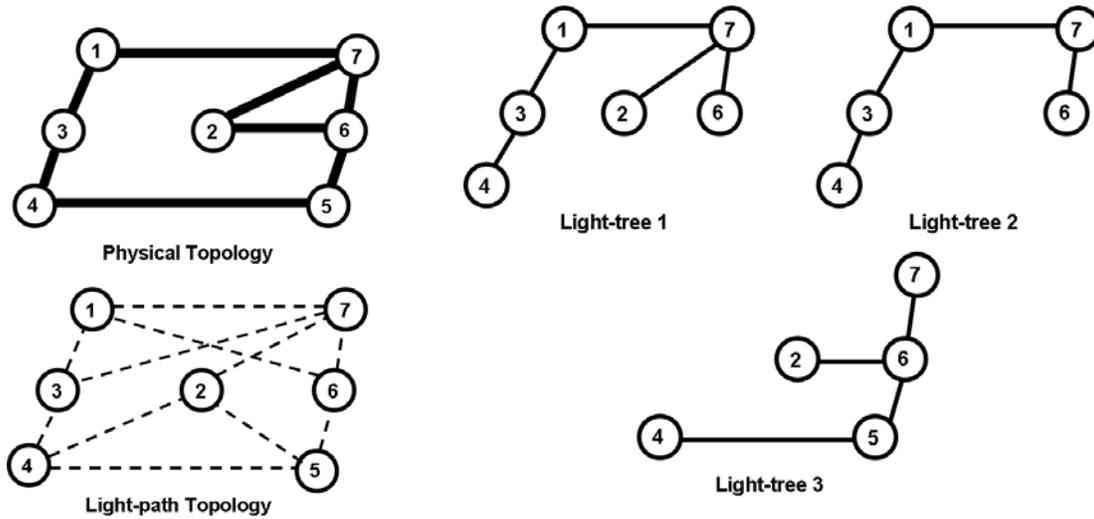


FIGURE 1: A simple network topology (7-node).

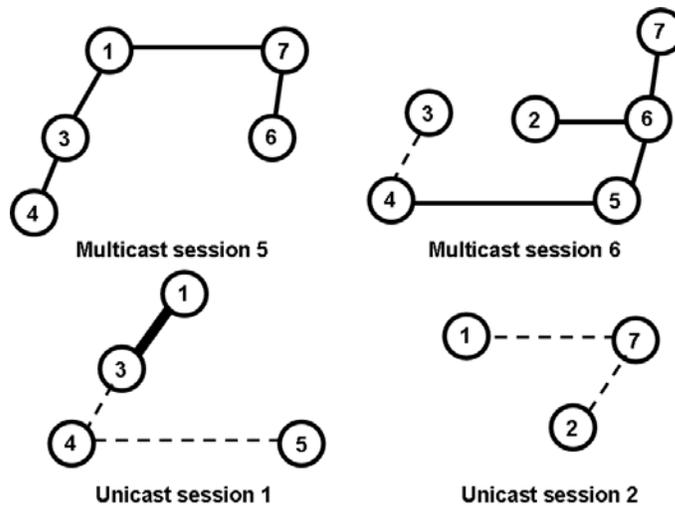


FIGURE 2: Designed hybrid topologies for new unicast and multicast demands (7-node network). (Thick black line: physical link, dashed line: existing light-path, and black line: existing light-tree).

5. Simulation results

Extensive simulations have been conducted to investigate the validity, feasibility and efficiency of our solution. First, we use a simple network topology to explain our solution in detail and validate it. We then apply our solution to a larger network to evaluate the feasibility and efficiency in a real scale network. A simple network topology with 7-nodes is illustrated in Figure 1. Figure 1(a) shows the physical and virtual light-path topologies of the network, and Figure 1(b) depicts the virtual topologies of existing light-trees prior to the arrival of new traffic demands. Each physical link carries a limited number of wavelengths. We assume that the number of available wavelengths on each physical link is $C^p = 5$. First, in order to validate our solution, we suppose the integrated demands of six new multicast sessions and two new unicast demands have been requested. The multicast sessions are $S_1 = \{1, 2, 3, 4, 6, 7\}$, $S_2 = \{1, 2, 3, 4, 6, 7\}$, $S_3 = \{1, 2, 3, 4, 7\}$, $S_4 = \{2, 3, 4, 6,$

7}, $S_5 = \{1, 3, 4, 6, 7\}$ and $S_6 = \{3, 4, 5, 6\}$ with source nodes $\{4, 6, 1, 7, 1, 5\}$, respectively. Note that S_1 and S_2 have the same set of destinations but different source nodes. Two new unicast traffic demands are requested additionally, and they are $U_1 = \{1, 5\}$ and $U_2 = \{1, 2\}$ with source nodes $\{1, 2\}$ respectively.

Figure 2 shows the created hybrid optical topologies with the given new traffic demands, using our ILP formulation. The parameters used are $w = 1$, $\beta = 0.01$, $C^v = 2$, $C^t = 2$, and the value of α is proportional to the number of used physical links to implement the light-path. Note that for a light-tree, the value of β corresponds to each individual link in the tree. We used the homogeneous weights for concise discussions. However, the weights can vary for each link as discussed in the previous section. For example, the weights may be proportional to the actual length of the links, so that it would reflect the delays. The values of w , α , and β indicate the relative preference of resource components for the new hybrid topology design. Small value of β increases the preference of using the light-trees over the light-paths and physical links. In the example scenario, light-trees are weighted least, so that they would be preferred. The result shows that they are all indeed the minimum cost trees that partially exploit existing light-paths, light-trees as well as physical link wavelengths. In Figure 2, which represents session topologies, we can observe that the light-trees are first exploited entirely according to the degree of sharing C^t . The light-paths and particularly physical links are not used extensively due to their high cost coefficients w and α with respect to the cost coefficient of the light-tree β .

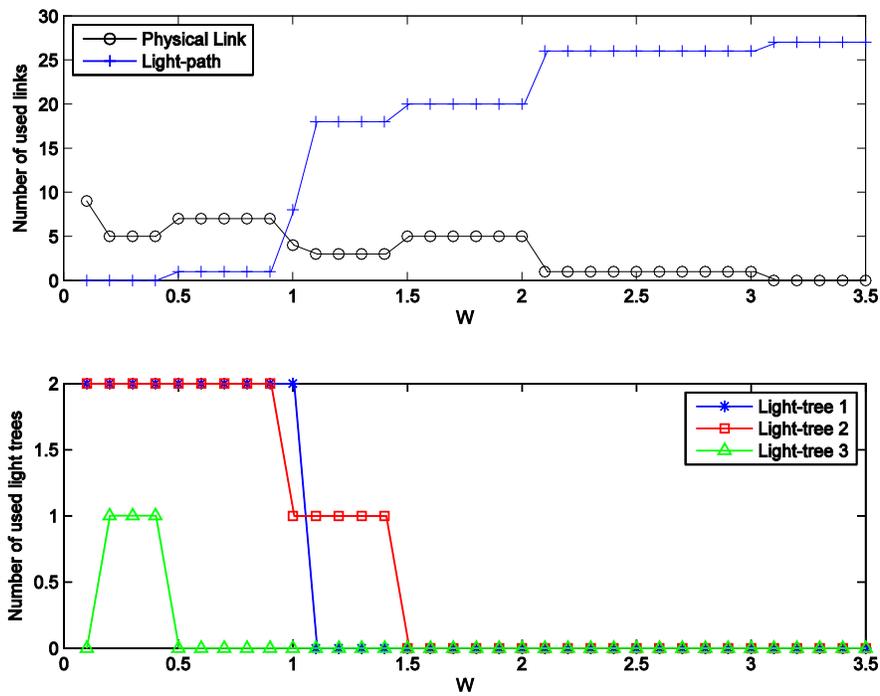


FIGURE 3: w vs. the number of links (top) and light-trees (bottom) used (7-node network).

Next, we investigate the impact of optimization cost weights as shown in Figures 3 and 4. Figure 3 shows the variation of the hybrid topologies when w is changed, while β is fixed to be 1 and $C^v = C^t = 2$. The figure shows that the number of used physical links changes according to the number of used light-trees. The number of used light-paths increases when w becomes more expensive to equalize the decrease in number of light-trees and physical links. Similarly, Figure 4 shows the variation of the hybrid topologies when β is changed, while w is constant to be 1 and $C^v = C^t = 2$.

As β increases, the preference of using the light-trees decreases and the number of used light-paths and physical links increases. When a light-tree is no longer used, light-paths and physical links are used to overcome the shortage of resources to satisfy the traffic demands. We did not vary α , as we set the parameter α to be the sum of physical link weights used for the light-path.

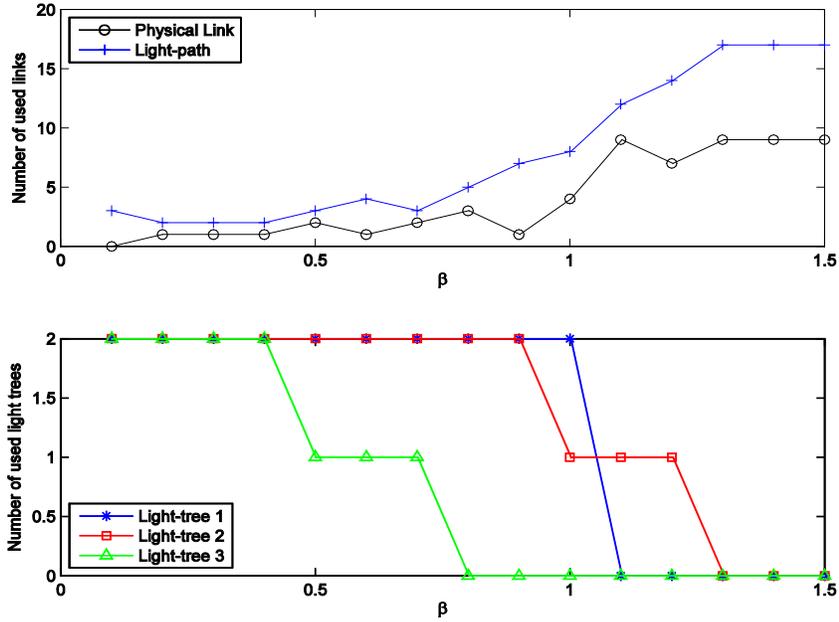


FIGURE 4: β vs. the number of links (top) and light-trees (bottom) used (7-node network).

We now consider a larger network with 14-nodes as illustrated in Figure 5. Figure 5(a) shows the physical and existing light-paths, and Figure 5(b) depicts the existing light-tree topologies. The physical cost coefficient w is equal to 1, while the light-tree cost coefficient β is set to 0.01. The light-path cost coefficient α is proportional to the number of used physical links to implement each light-path. Degree of sharing a light-path C^v and a light-tree C^t are set to 2. The new multicast demands used to evaluate the formulation are $S_1 = \{1, 3, 4, 6, 9\}$, $S_2 = \{5, 7, 8, 10, 13, 14\}$, $S_3 = \{6, 7, 11, 12, 13\}$, $S_4 = \{9, 10, 11, 12, 14\}$, $S_5 = \{2, 3, 7, 8, 11, 12, 14\}$, $S_6 = \{1, 2, 13, 14\}$, $S_7 = \{1, 3, 7, 11\}$, and $S_8 = \{3, 4, 7, 14\}$, with source nodes to be $\{1, 13, 6, 9, 2, 2, 11, 14\}$, respectively. In addition, two unicast demands, $U_1 = \{1, 8\}$ and $U_2 = \{3, 10\}$ are requested with source nodes $\{1, 10\}$.

The solution was found successfully for the larger network, and we show the created hybrid topologies for the unicast traffic in Figure 6, and for the multicast traffic in Figures 7. We depict the topologies for the multicast traffic only for the first five demands, for a concise illustration. Figure 7 shows the hybrid topology solutions for individual multicast sessions. Multicast session 1 uses the existing light-tree 1 in addition to two physical links between nodes 3 and 4, and 6 and 9. The new multicast sessions 2 and 3 use the existing light-trees 2 and 4, respectively, in addition to other light-paths, to satisfy their multicast session demand. The new multicast session 4 uses light-trees 3 and 4, and the new multicast session 5 uses a physical link, an existing light-path and the existing light-tree 5.

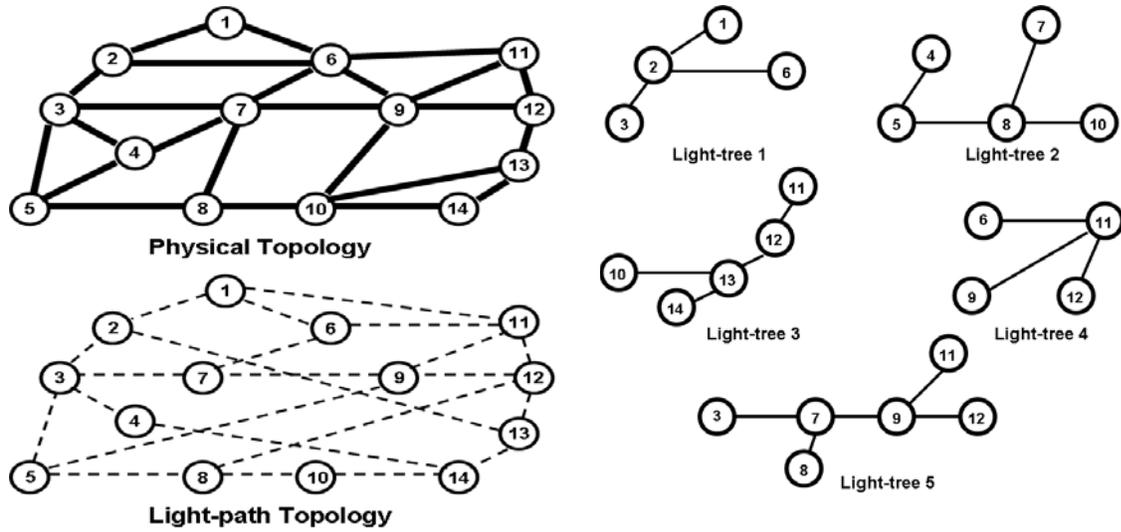


FIGURE 5: A Larger Network Topology (14-nodes).

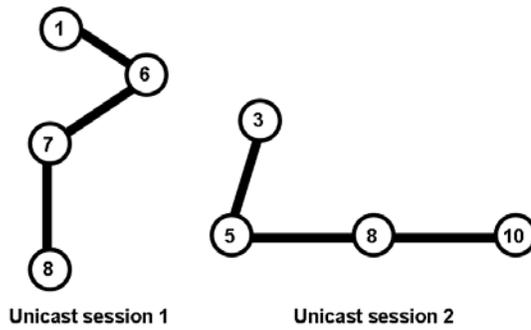


FIGURE 6: Designed hybrid topologies for new unicast demands (14-node network). (Thick black line: physical link)

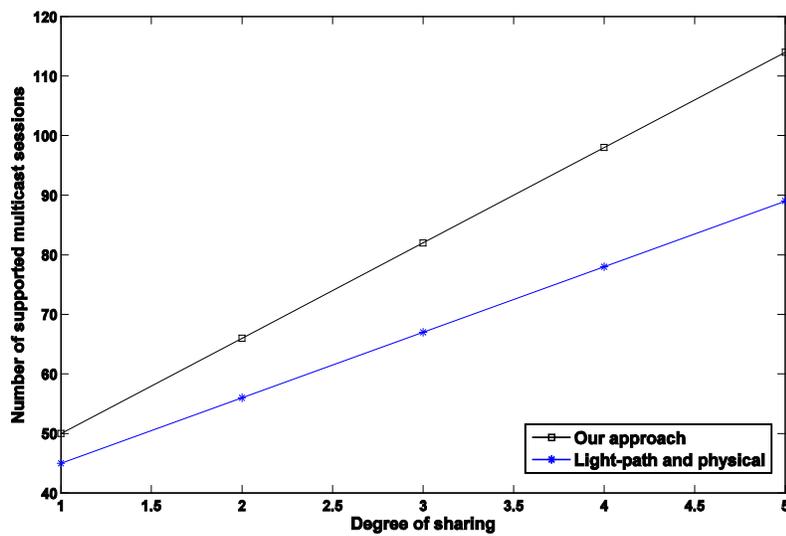


FIGURE 7: Designed hybrid topologies for new multicast demands (14-node network). (Thick black line: physical link, dashed line: existing light-path, and black line: existing light-tree)

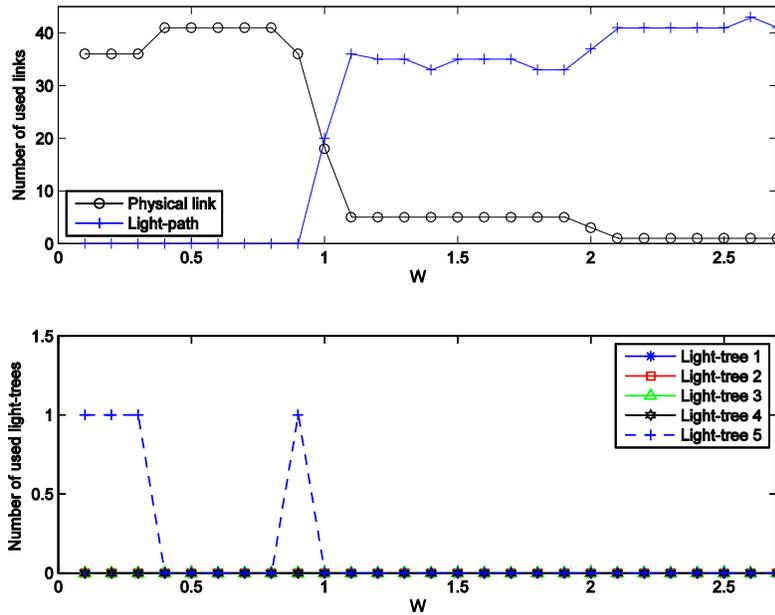


FIGURE 8: Number of links (top) and light-trees (bottom) vs. w (14-node network).

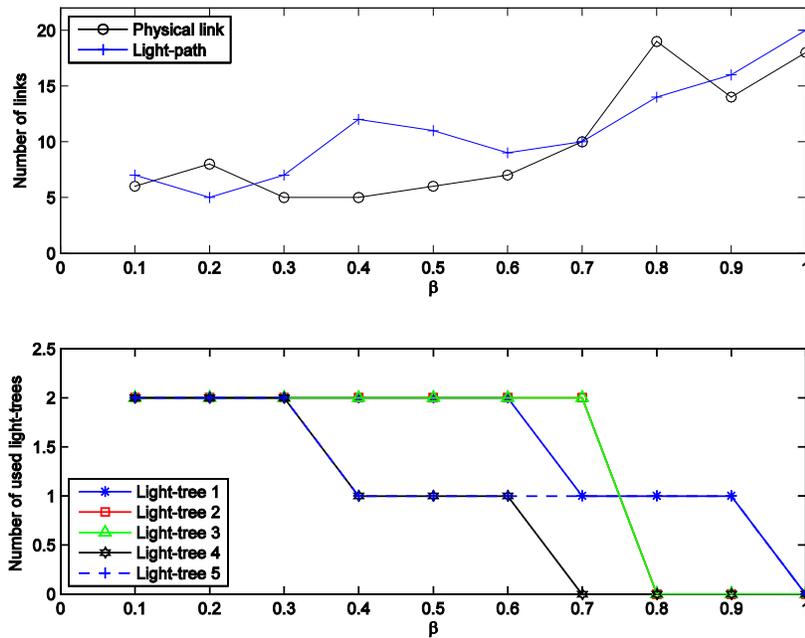


FIGURE 9: Number of links (top) and light-trees (bottom) vs. β (14-node network)

Figures 8 and 9 illustrate the impact of optimization cost weights for the 14-node network. Figure 8 shows the variation of the hybrid topologies when β is changed and w is constant to be 1 where $C^v = C^t = 2$. For small value of β , all light-trees are used to create hybrid multicasts in addition to some physical links and light-paths. As β increases, more physical links and light-paths are mainly utilized to keep the cost of creating the hybrid multicast topologies low. Similarly, Figure 9 shows the variation of the hybrid topologies when w is changed and β is fixed to 1 where $C^v = C^t = 2$. The figure shows that the number of light-paths increases while the number of physical links used is

decreasing, because physical links become more expensive to transport the multicast traffic demand. In addition, both Figures 3 and 8 show a similar structure of the created hybrid multicast topologies. The figures show that there is a value of w at which the light-trees are no longer used to create the hybrid topologies. This value depends on the network structure as well as the light-trees structure. Similarly, Figures 4 and 9 show similar structure of the created hybrid multicast topologies, and the existence of a value for β at which light-trees are not used to create the hybrid topologies.

We next evaluate the number of supported multicast sessions while varying the degree of sharing of light-paths and light-trees, for this large network. We assumed that C^v and C^t are the same.

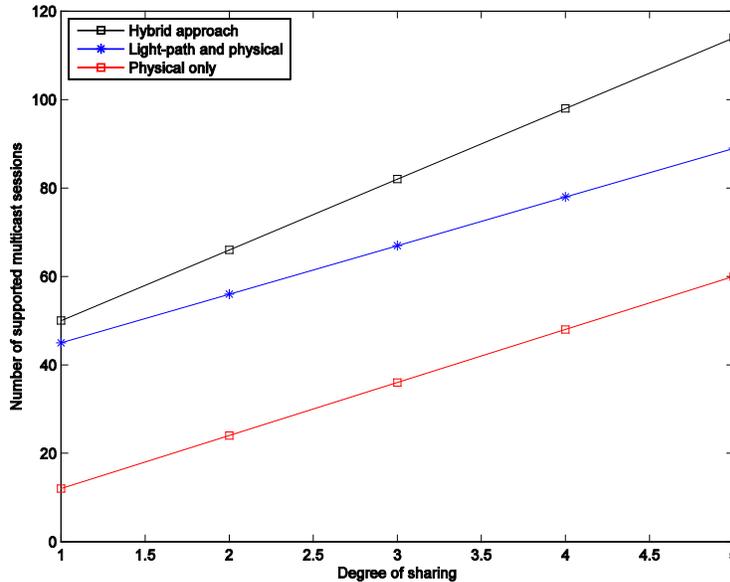


FIGURE 10: The number of supported multicast sessions vs. degree of sharing of the light-paths and the light-trees (14-node network)

Figure 10 shows that the number of supported multicast sessions increases as the degree of sharing increases. Results are compared with no hybrid approach that does not use sub-wavelength sharing. It also compares with the previous work [4], that was the first work that proposed sub-wavelength sharing with existing light-paths but the sharing of multicast trees were not allowed. Figure 10 shows that the number of multicast sessions linearly increases as the degree of sharing increases. The proposed approach clearly satisfies more multicast demands with the sharing of light-trees as well as light-paths.

Sharing degree	7-node network		14-node network	
	# Multicast	% increase	# Multicast	% increase
1	22	0	50	0
2	36	63	66	32
3	50	127	82	64
4	61	177	114	128

TABLE 2: Effect of sharing the wavelengths

Table 2 compares the small 7-node and large 14-node networks to evaluate the impact of the sharing. It shows the percentage of the increase in the number of supported multicast sessions, with the degree of sharing. The percentage increase is computed comparing with the case of no sub-wavelength sharing. Number of supported sessions increases almost linearly with the degree of sharing for both 7-node and 14-node networks, respectively.

Sharing degree(C^t)	7-node network		14-node network	
	# Multicast	% increase	# Multicast	% increase
1	22	0	50	0
2	28	27	55	10
3	31	40	60	20
4	34	54	65	30

TABLE 3: Sharing degree (C^t) vs. number of supported multicast sessions ($C^v=1$)

Sharing degree(C^v)	7-node network		14-node network	
	# Multicast	% increase	# Multicast	% increase
1	22	0	50	0
2	29	31	61	22
3	39	77	72	44
4	49	122	83	66

TABLE 4: Sharing degree (C^v) vs. number of supported multicast sessions ($C^t=1$)

Tables 3 and 4 show the effect of degree of sharing light-paths and light-trees for both 7-node and 14-node networks. The number of supported multicast sessions increases linearly with the degree of sharing light-paths and light-trees. The degree of sharing light-paths gives more freedom to support more sessions than degree of sharing light-trees. The extensive simulations with small and large mesh networks shown in this section illustrate that the proposed solution creates hybrid topologies for more traffic demands than pure wavelength assignment in an efficient and scalable manner.

6. Conclusions

The future Internet will require the transport of a wide range of services including high bandwidth one-to-many applications, with dynamic interconnection of devices and services. Due to limited wavelengths of WDM networks, an optimal resource management is important for a new set of traffic service demands while keeping services for the existing traffic. We proposed a hybrid optical topology design over constrained WDM mesh network, where both light-paths and light-trees are built. Particularly, existing light-trees as well as light-paths are re-used to create new hybrid multicast virtual topology. It is to increase the number of supported integrated traffic demands of both unicast and multicast for the future Internet, using the excess bandwidth of a wavelength. This sub-wavelength sharing is done within a degree of sharing of light-trees and light-paths. The degree of sharing allows and also bounds the amount of sharing so as to maintain QoS of existing traffic. The problem of creating a hybrid optical topology is formulated using ILP approach, given existing physical, light-path, and light-tree topologies. We formulated the ILP in a compact manner, and the solutions can be reached for a relatively large network in a reasonable time.

Our approach shows how the existing physical, light-path and light-tree topologies are exploited for the newly arriving demands optimally without re-designing all the topologies from the scratch. This approach maximally utilizes the available bandwidth resources from existing light-paths as well as light-trees whose traffic demand does not reach full wavelength capacity. We show this hybrid virtual topology design enables us to establish multicast trees when it would otherwise be impossible with a pure light-tree approach. The proposed solution can be used in a real practical environment where both unicast and multicast demands are supported, and new multicast demands can be realized incrementally and optimally. Extensive simulations are performed over various WDM mesh networks, to show the validity as well as feasibility with relatively large networks. As for a future work, an efficient heuristic approach can be made to speed up a solution.

7. REFERENCES

1. K. Hastings and N. Nechita, "Challenges and opportunities of delivering IP-based residential television service". IEEE Communications Magazine, vol. 38, no. 11, pp. 86–92, November 2000
2. R. K. Pankaj, "Wavelength requirements for multicasting in all-optical networks". IEEE/ACM Transactions on Networking, vol. 7, pp. 414–424, 1999.
3. L. Sahasrabudde and B. Mukherjee, "Light-trees: Optical multicasting for improved performance in wavelength-routed networks." IEEE Communications Magazine, vol. 37, no. 2, pp. 67–73, February 1999.
4. S. Bhandari, B.-Y. Choi, and E. K. Park, "Hybrid topology for multicast support in constrained WDM networks". In Proceedings of 20th International Teletraffic Congress, Ottawa Canada, Jun. 2007.
5. E. L. V. et. al., "Architecturing the services in an optical network". IEEE Communications Magazine, vol. 39, no. 9, pp. 80–89, Sep. 2001.
6. N. K. Singhal and B. Mukherjee, "Protecting Multicast Sessions in WDM Optical Mesh Networks". Journal of Lightwave Technology, vol. 21, no. 4, April 2003.
7. M. Mellia, A. Nucci, A. Grosso, E. Leonardi, and M. A. Marsan, "Optimal Design of Logical Topologies in Wavelength-Routed Optical Networks with Multicast Traffic". in IEEE Globecom, vol. 3, 2001, pp. 1520–1525.
8. G. Agrawal and D. Medhi, "Single Shortest Path-based Logical Topologies for Grooming IP Traffic over Wavelength-Routed Networks". In Proceedings of 2nd IEEE/Create-Net International Workshop on Traffic Grooming, 2005.
9. D.-N. Yang and W. Liao, "Design of Light-Tree Based Logical Topologies for Multicast Streams in Wavelength Routed Optical Networks". In IEEE INFOCOM, 2003.
10. D. Cavendish and B. Sengupta, "Routing and wavelength assignment in WDM rings with heterogeneous wavelength conversion capabilities". In IEEE Infocom, 2002.
11. D. Banerjee and B. Mukherjee, "Wavelength-Routed Optical Networks: Linear Formulation, Resource Budget Tradeoffs and a Reconfiguration Study," IEEE ACM Transactions on Networking, vol. 8, no. 5, pp. 598–607, October 2000.
12. A. E. Gencata and B. Mukherjee, "Virtual-topology adaptation for WDM mesh networks under dynamic traffic," In IEEE INFOCOM, June 2002.
13. B. Ramamurthy and A. Ramakrishnan, "Virtual topology reconfiguration of wavelength-routed optical WDM networks". In Global Telecommunications Conference (GLOBECOM), vol. 2, 2000.
14. F. K. Hwang, D. S. Richards, and P. Winter, "The Steiner Tree Problem". New York: Elsevier, 1992.
15. R. Karp, "Reducibility among combinatorial problems". Complexity of Computer Computations, 1972.

16. X. Zhang, J. Wei, and C. Qiao, "Constrained multicast routing in WDM networks with sparse light splitting". *Journal of Lightwave Technology*, vol. 18, pp. 1917–1927, 2000.
17. M. Jeong, Y. Xiong, H. C. Cankaya, M. Vandenhoute, and C. Qiao, "Efficient Multicast Schemes for Optical Burst-Switched WDM Networks". In *IEEE ICC*, pp. 1289–1294, 2000.
18. T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing". in *ACM SIGCOMM*, pp. 85–95, October 1993.
19. T. Pusateri, "DVMRP version 3. draft-ietf-idmr-dvmrp-v3-07". IETF, August 1998.
20. R. Malli, X. Zhang, and C. Qiao, "Benefit of Multicasting in All-Optical Networks". In *SPIE Conf. All-Optical Networks*, pp. 196–208, 1998.
21. C. Qiao, "Labeled optical burst switching for IP-over-WDM integration". *IEEE Communications Magazine*, vol. 38, no. 9, pp. 104–114, 2000.
22. X. Zhang, J. Wei, and C. Qiao, "On Fundamental Issues in IP over WDM Multicast". In *IEEE International Conference on Computer Communications and Networks*, October 1999.
23. M. Jeong, C. Qiao, and Y. Xiong, "Reliable WDM Multicast in Optical Burst-Switched Networks". *Opticomm*, pp. 153–166, October 2000.
24. F. Zhou, M. Molnar, and B. Cousin, "Is light-tree structure optimal for multicast routing in sparse light splitting wdm networks?". In *18th International Conference on Computer Communications and Networks*, 2009.
25. B. Mukherjee, "Optical Communication Networks: WDM, Broadcast/Multicast and Wavelength-Routing". *Mc Graw Hill*, 1997.
- 26 T. Stern and K. Bala, "Multiwavelength Optical Networks". *Addison Wesley*, 1999.

Efficient and Fair Bandwidth Allocation AQM Scheme for Wireless Networks

Rafe Alasem

*Computer Science College
Computer Science Department
Imam Muhammed Ibn Saud Islamic University
PO Box 5701, Riyadh 11432, KSA*

rafe.alasem@gmail.com

Abstract

Heterogeneous Wireless Networks are considered nowadays as one of the potential areas in research and development. The traffic management's schemes that have been used at the fusion points between the different wireless networks are classical and conventional. This paper is focused on developing a novel scheme to overcome the problem of traffic congestion in the fusion point router interconnected the heterogeneous wireless networks. The paper proposed an EF-AQM algorithm which provides an efficient and fair allocation of bandwidth among different established flows.

Finally, the proposed scheme developed, tested and validated through a set of experiments to demonstrate the relative merits and capabilities of a proposed scheme

Keywords: Wireless Network, Congestion Control, Active Queue Management, Random Early Detection.

1. INTRODUCTION

Wireless communication technology is playing an increasingly important role in data networks. Wireless networks are usually connected to the internet via backbone gateway routers. The packet loss may occur at fusion points that connect the backbone network to the wireless networks.

However, in a wireless heterogeneous network, the loss occurs due to the channel nature and characteristics. For example, in IEEE 802.11 wireless networks, congestion may be defined as a state where the shared wireless medium is completely occupied by the nodes because of given channel characteristics in addition to external interference. The shared environment of the wireless medium causes a node to share the transmission channel not just with other nodes in the network, but also with external interference resources [1].

In reality that a wireless channel is shared by challenging neighbor nodes and the number of nodes sharing this channel may change all the time [2]. An additional reason is that the wireless link bandwidth is affected by many changing physical conditions, such as signal strength, propagation distance, and transmitter power. For example, an IEEE 802.11 node can modify its MAC-layer data rate dynamically according to different situations, which means the output bandwidth of this node and other neighbor nodes may also change .

In wireless networks throughput degradation can occur due to the sharing the lossy channel and packets collision. Slotted CSMA/CD is used to overcome a bit the collision occurrence at the lossy channel. Sending the frames at the slot start reduces the chance of frames' collision.

In addition, the using of a conventional mechanism in managing the traffic in inside the wireless gateway node causes an additional loss to truly arrived frames. Throughput degradation can occur also due to improper use of traffic management schemes at the fusion points of heterogeneous wireless networks [6]. For these reasons, an efficient mechanism for congestion control should be applied at the bottleneck nodes to overcome the additional loss in an accurate data.

In an IEEE 802.11 wireless network, the occurrence of high density nodes in a single collision domain can cause congestion, in consequence causing a substantial bottleneck gateway router [5]. Packet dropping, packet delays and session disruptions are a consequence of congested network.

Congestion control problem occurs when the demand on the network resources is greater than the available resources and due to increasing mismatch in link speeds caused by intermixing of heterogeneous network technologies. This congestion problem cannot be solved with a large buffer space. Clearly too much traffic will lead to a buffer overflow, high packet loss and large queuing delay. Furthermore, congestion problem cannot be solved by high-speed links or with high-speed processor, because the high-speed link connected via the high-speed switch with the low-speed links will cause congestion at the wireless fusion point of interconnection.

Drop Tail has been proposed in [4]. The most operational routers currently use Drop Tail coupled with FIFO (First in first out) scheduling scheme. In Drop Tail, all packets are accepted until the maximum length of the queue is reached and then dropping subsequent incoming packets until space becomes available in the queue.

Drop Tail is not appropriate as a feedback control system for high-speed networks because it sustains full queues and this may increase the average queuing delay in the network. More importantly, Drop Tail can cause a lockout due to traffic phase effects and the global synchronization, and thus results in low throughput. The lost packet from a Drop Tail queue will usually be retransmitted by TCP protocol via its retransmission timer. No congestion is detected until the buffer becomes full and the maximum congestion indicator is generated because all arriving packets are dropped. Then each source detects lost packets it will slow down the arrival rate of the sending packets until the queue will be less than the capacity of the link. No congestion indicator will be generated when the queue is not full, each source will increase until overflow happens again

In the recent years, Active queue management (AQM) mechanisms have been proposed to provide an efficient queue management by selectively dropping/marketing packets when congestion is anticipated so that TCP senders can reduce their transmission rate before an overflow occurs. AQM mechanisms are employed in the Internet by the routers to provide better stability, fairness, and responsiveness to dynamic variations in computer networks. Using queue management mechanisms in an efficient way will avoid the congestion collapse and lead to high link utilization.

In this paper, we present a novel buffer management approach for congestion control in a wireless network. This approach achieves both efficient and fair allocation of bandwidth among flows by randomly dropping frames and increases data throughput to the next hop.

The rest of this paper is organized as follows: In section II, the network model is presented. The proposed scheme is developed in section III. Extensive simulations and results are investigated in section IV. Section V concludes this paper..

2. NETWORK MODEL

The network model considered in this paper will be explained in this section. IEEE 802.11 based wireless LAN networks have been chosen as a reference model of this investigation. A bottleneck wireless gateway router has been taken into account in this topology (Figure 1). A wireless channel L is shared by N challenging neighbor nodes and the number of nodes sharing this channel may change all the time. We suppose that all nodes use the same power and modulation methods. Figure 1 illustrates the wireless network model. The traffic model is shaped as follows. We assume that each node is transmitting HTTP packets of size S bits, and they are generated according to Markovian-Modulated Poisson Process MMPP with arrival rate λ .

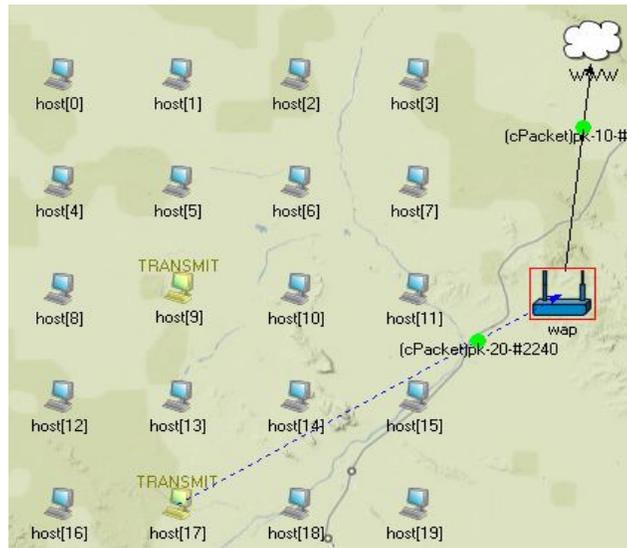


FIGURE 1: IEEE 802.11 Wireless network

The fusion point that connects the wireless network with wired network has a finite amount of buffer space, and this buffer is managed via an adaptive queue management scheme. In addition, the output link has a fixed bandwidth and connects the wireless network with Internet. But for a wireless network, for example in 802.11b, the node can dynamically change its MAC-layer data rate to 1, 2, 5.5, or 11Mbps. Consequently, when congestion is occurred the TCP is unable to maintain fairness and stability with improper estimation of the link capacity parameter. For this reason, the need for an adaptive and intelligent AQM algorithm to be implemented at the fusion point of a wireless network is critical and crucial.

The congestion sliding widow w is increased by one every round trip time if no congestion is detected, and it is reduced by half if a congestion is detected. This is called an additive-increase multiplicative decrease (AIMD) mechanism that represents the behavior of TCP flows, and it is described by the following nonlinear differential equation [9]:

$$\dot{w} = \frac{1}{r(t)} - \frac{w(t-1)}{2r(t-1)}p(t-1) \quad (1)$$

Where R corresponds to the round-trip time (seconds) and p is representing the probability of packet drop/mark. In addition, the instantaneous queue can be expressed in the following equation:

$$q(t) = \frac{w(t)}{r(t)} N - C \tag{2}$$

Where C is the link speed (packets/sec) and N load factor (number of TCP connections).

3. EF-AQM SCHEME

EF-AQM scheme has been designed and analyzed in terms of feedback control theory (Figure 2). The advantages of using control theory are to increase the speed of response and to bring further improvement to the system robustness and stability. These advantages can be achieved by regulating the output queue length around a target value Q_{ref} . An important goal of the AQM design is to stabilize the queue length $q(t)$ at a given target Q_{ref} , so that the magnitude of the error signal.

$$e(t) = Q_{ref} - q(t) \tag{3}$$

is kept as small as possible.

The output of the EF-AQM controller represents the dropping probability and is simplified as:

$$y(k) = y(k-1) + K_0(e(k) - e(k-1)) + K_1Te(k-1) \tag{4}$$

where T is the sampling period time, K_1 and K_2 represent the tuning parameters of the controller.

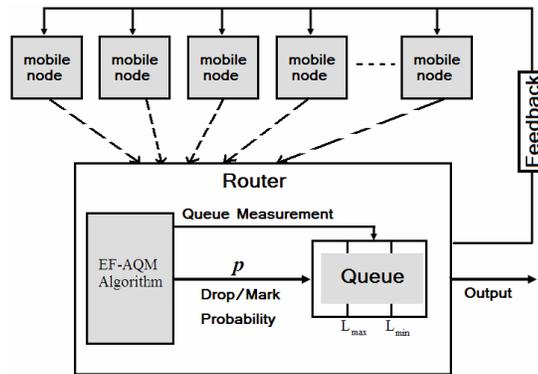


FIGURE 2: EF-AQM as a feedback control system

The dropping probability is calculated according to the intelligent controller, and it is considered as a function of the difference between the current value of the queue length and the reference queue length. Our aim is to compute this dropping probability $p_d(t)$ such that it will keep the instantaneous queue length close to the target queue. Therefore, the dropping probability $p_d(k)$ can be achieved as:

$$p_d(k) = \begin{cases} 0 & y(k) < 0 \\ y(k) & 0 < y(k) < 1 \\ 1 & y(k) > 1 \end{cases} \quad (5)$$

The dropping probability is calculated at every packet arrival. The reason for randomizing the packet drops is the hypothesis that users generating more traffic would have a greater number of packets dropped. The same as in Random Early Detection (RED) [7], if the average queue length exceeds a minimum threshold L_{min} , incoming packets are dropped/marked with a probability that is a linearly increasing function of the average queue length. When the average queue size exceeds a maximum threshold L_{max} , the router is likely to incur congestion, and all incoming packets are dropped/marked. When it is between, a packet is dropped with a probability p which represents the output of EF-AQM controller.

4. PERFORMANCE EVALUATION

To validate the performance and the robustness of the proposed EF-AQM algorithm for wireless network, we simulated it using OMNET++ platform [10] with highly bursty traffic. Different scenarios have been chosen to validate the proposed algorithm with different number of flows. The parameters used in EF-AQM simulation are: $N = 10, 20, 30$ and 40 flows, $L_{max} = 500$ packets, $L_{min} = 100$ packets, packet length = 100 bytes, IEEE 802.11 propagation delay = 10 ms. Probability based dropping $p_{max} = 0.2$ and the target queue length is $(L_{max} + L_{min})/2$.

4.1. Throughput

As shown in Figure 3, EF-AQM has offered higher throughput as compared to the classical algorithm Drop Tail and RED for the 10, 20, 30 and 40 flows respectively. It is observed that although the number of TCP flows has increased, EF-AQM has offered higher throughput for different control approaches and reach (100%) for the smaller flows. This due to the stability of EF-AQM in maintaining the queue length which makes it more stable around the target queue and shrunk in width; thereby packet dropping is less despite of increasing the number of TCP flows.

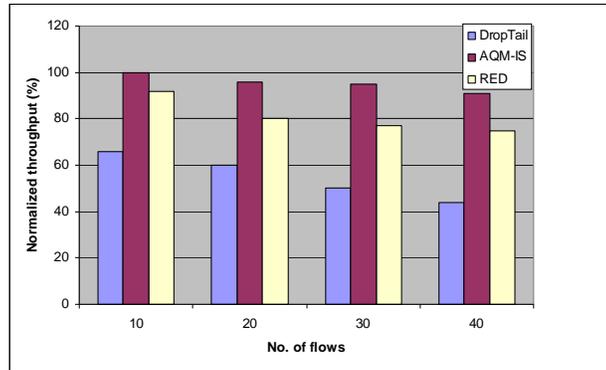


FIGURE 3: Throughput comparison for different flows

4.2. Queuing Delay

The queuing delay is considered as one of the important metrics in performance evaluation of any AQM controller. Figure 4 shows that the queuing delay of the proposed EF-AQM is very small compared other approaches for a variety of flows. It is noted that the queuing delay becomes constant for EF-AQM controller despite increasing traffic load or changing the type of data traffic. For other schemes, the delay is continued to increase when changes occur in any of the network parameters.

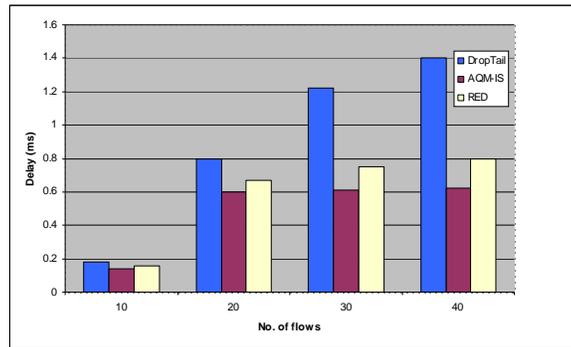


FIGURE 4: Delay comparison for different flows

4.3 Queue length

Figures 5 Shows the instantaneous queue length evolution comparison under the EF-AQM approach compared with RED for the number of TCP flows equal to 20, 30 and 40 respectively. It can be seen that the instantaneous queue length of the EF-AQM is stable and oscillate around the target queue length. While the instantaneous queue length of the RED algorithm is still fluctuated away from the target queue length as the number of TCP connection increased due to the sensitivity of RED to any change in its parameters. It is worth noting that the EF-AQM scheme with adaptive tuning parameters is the most stable control scheme as compared to the others; demonstrates steadiness mode despite higher number of TCP flows.

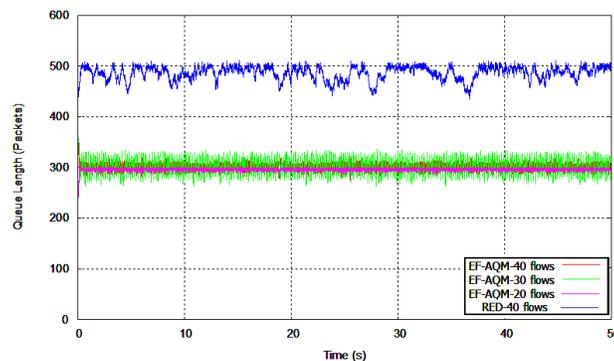


FIGURE 5: Instantaneous queue comparison for different flows

5. PERFORMANCE EVALUATION

This section demonstrates merits and capabilities of the proposed scheme. Some of the critical issues are further discussed below:

Convergence: It is noted that the speed of convergence of the proposed congestion control mechanisms to a stable operating point is independent of the number of TCP sources and connections. The output queue length converges rapidly to the reference queue as the sources start transmission their data. It is also observed that the EF-AQM scheme has the optimum convergence time rather than RED.

Fairness: One of the important goals of proposed congestion control mechanisms is the contiguity between the link utilization and the queuing delay in the bottleneck router. Fair dropping of packets and fair sharing of bandwidth for all connections is achieved. Also it is worth noting that EF-AQM offers an optimum round trip time fairness metric in comparison to RED scheme due to its minimum queuing delay.

Robustness and Stability: One goal in this investigation is to explore the robustness and stability, in terms of minimizing oscillations of queuing delay or of throughput. In practice, stability is frequently associated with rate fluctuations or variance. It is observed that EF-AQM scheme has offered an optimum queuing length and queuing delay oscillations. It is also noted that RED has offered higher queuing delay and queue length oscillations.

Scalability: It is worth mentioning that global implementation of the proposed EF-AQM congestion control mechanism in a decentralized form is expected to offer high network utilization. Scaling the local stability and low queuing delay for individual gateways in a large network will sufficiently present a global implementation of a congestion control mechanism over a scalable network.

Efficiency: One of the key concerns in the design of congestion control mechanisms has been the CPU usage time and maximizing bandwidth utilization. It is observed that the proposed mechanism provide high throughput in comparison to the standard RED. It is worth noting that EF-AQM has a moderate CPU usage compared to RED algorithm.

6. CONCLUSION

This paper presented an efficient and fair bandwidth allocation AQM algorithm to overcome the problem of congestion control in heterogeneous wireless network. It has been demonstrated that the new EF-AQM has achieved desirable properties such as robustness and fast system response, as compared to the traditional DropTail and RED. Finally, a set of experiments has been provided to demonstrate the efficiency of the proposed design approach. It is noted that the

proposed EF-AQM design approach performs significantly better than many well-known schemes, and guarantees the robustness of the controller.

7. REFERENCES

- [1] Prashanth A., Ashish S., Elizabeth B., Kevin A. and Konstantina P. " Congestion-Aware Rate Adaptation in Wireless Networks: A Measurement-Driven Approach", *IEEE SECON*, San Francisco, CA, June 2008.
- [2] Jian Pu; Hamdi, M., "Enhancements on Router-Assisted Congestion Control for Wireless Networks," *Wireless Communications, IEEE Transactions on* , vol.7, no.6, pp.2253-2260, June 2008.
- [3] Acharya, P.A.K.; Sharma, A.; Belding, E.M.; Almeroth, K.C.; Papagiannaki, K., "Congestion-Aware Rate Adaptation in Wireless Networks: A Measurement-Driven Approach," *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on* , vol., no., pp.1-9, 16-20 June 2008.
- [4] Jacobson V., (1988), "Congestion avoidance and control", *ACM SIGCOMM Computer Communication Review*, Vol. 18(4), p.314-329.
- [5] Dong, Y.; Makrakis, D.; Sullivan, T., "Network congestion control in ad hoc IEEE 802.11 wireless LAN," *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on* , vol.3, no., pp. 1667-1670 vol.3, 4-7 May 2003.
- [6] Xiao Laisheng; Peng Xiaohong; Wang Zhengxia; Xu Bing; Hong Pengzhi, "Research on Traffic Monitoring Network and Its Traffic Flow Forecast and Congestion Control Model Based on Wireless Sensor Networks," *Measuring Technology and Mechatronics Automation, 2009. ICMTMA '09. International Conference on* , vol.1, no., pp.142-147, 11-12 April 2009.
- [7] Floyd, S, Jacobson, V., (1993), "Random early detection gateways for congestion avoidance". *IEEE/ACM Trans. on Networking*, Vol. 1(4), pp.397-413.
- [8] Alasem, R, Hossain, M. A. and Awan, I., (2007), "Intelligent Active Queue Management Predictive Controller using Neural Networks", *IEEE Computer Society (The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies)*, ISBN 0769528783, Cardiff, UK, pp. 205-210.
- [9] C. Hollot, V. Misra, D. Towsley ,W. Gong. "A Control Theoretic Analysis of RED". *Proc. INFOCOM* Vol. 3, April 2001,pp. 1510 – 1519.
- [10] Alasem, R., Hossain, M. A., Awan, I., (2007), "Active Queue Management Controller using Smith Predictor for Time Delay Networks," *IEEE International Conference on Networking, Sensing and Control*, London, UK , Vol.1, pp.568-573.
- [11] Varga, A. (2010), *Omnet++: user manual*, [Online : <http://www.omnetpp.org>]

A Stochastic Model Approach for Reaching Probabilities of Message Flow in Space-Division Switches

D. Shukla

*Deptt. of Mathematics and Statistics,
Dr. H.S.G. Central University, Sagar (M.P.), India.*

diwakarshukla@rediffmail.com

Rahul Singhai

*International Institute of Professional Studies,
Devi Ahilya Vishwavidyalaya, Indore (M.P.) India.*

singhai_rahul@hotmail.com

Surendra Gadewar

Govt. Excellence School, Sagar (M.P.), 470003

skgadewar@gmail.com

Saurabh jain

*Deptt. of Computer Sc. and Applications,
H.S. Gour Sagar University, Sagar (M.P.), 470003*

iamsaurabh_4@yahoo.co.in

Shewta Ojha

*Deptt. of Computer Sc. and Applications,
H.S. Gour Sagar University, Sagar (M.P.), 470003*

P.P. Mishra

Chhatrasal Government College, Panna, (M.P.)

mishrapp@yahoo.com

Abstract

In computer networks messages are transmitted through switches in order to reach destinations. Cross point switches are the simplest technique which requires one-to-one connection between input and output. The limitation of a cross bar switch technology is that if numbers of lines are large then cross point increases in huge amount consequently increasing the dimension of the switch. Space division switches are used to route the calls, which implies betterment over the cross bar technology. Actually it is built up of several smaller rectangular crossbars. This takes the advantage over the crossbar switches that less cross points are needed in the space division switches. The limitations of space division switches are that if more the number of cross points then there is increase the outgoing reaching probabilities of messages but the cost and overheads is also high with lesser cross points the congestion increases. In this paper we considered the architecture of a three crossbar space-division switch and assumed a Markov chain model for the transitional analysis of message flow. With the help of simulation study it is concluded that the impact of reaching probabilities for the different values of parameters must be kept in mind while designing this switch by designers.

Keywords: Space-division switch, Cross-Bar Technology, Markov chain model, Reaching probabilities, Transition Probability matrix, Simulation study, Message flow .

1. REVIEW OF LITERATURE

Ko and Davis [3] proposed a protocol known as space-division multiple access (SDMA) which is useful for a satellite switched communication network. Abott [1] discussed a new technique for switching system using digital Space-Division concept for dealing with high-speed data signals. Yamada et al. [16] derived the high-speed digital switching technology with the help of space-division switches. Karol et al. [8] presented an input versus output analysis of queuing on a space-division packet switching. In a contribution Li [5] performed analysis for non-uniform traffic in the setup of Space-Division switches. Yamanka et al. [17] expanded space-division (SD) switch architecture and suggested a bipolar circuit design for gigabit-per-second cross-point switch LSIs. Lee and Li [4] have studied the performance of a non blocking space-division packet switch using finite-state Markov chain model, given the traffic intensities changes as a function of time. Li [6] derived the performance of a non blocking space-division packet switch in a correlated input traffic environment. Wang and Tobagi [14] suggested a self-routing space-division fast packet switch architecture achieving output queuing with a reduced number of internal path. Cao [2] derived a discrete-time queuing network model for space-division packet switches. Pao and Leung [10] used space-division approach to implement a shared buffer in an ATM switch which does not require scaling up the bandwidth of the shared memory. Shukla, Singhai & Gadewar [12] presented Markov Chain analysis for reaching probabilities of message flow in space division switches.

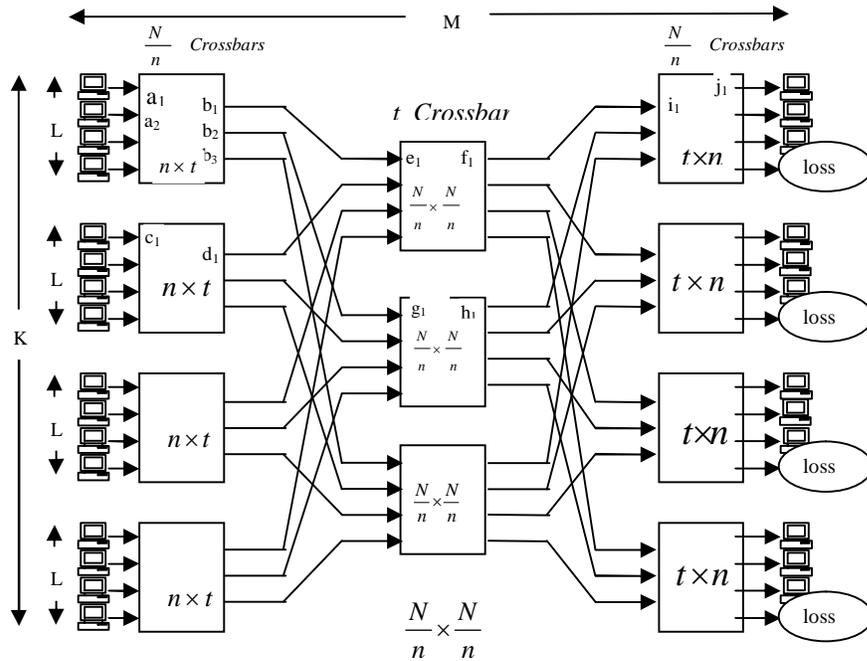


FIGURE 1: Three cross bar space division switch

2. MOTIVATION

Shukla and Gadewar [11] have suggested a Markov chain model for the transitional analysis of message flow in a two crossbar space division switches. We extend this model, in this paper,

from two-crossbars to three crossbar setup and with the help of a simulation study, the impact on reaching probabilities of message is analyzed.

3. INTRODUCTION AND ASSUMPTIONS

In what follows, we consider a space-division switch [11],[12] with parameters $N = 16, n = 4, t = 3$ shown in fig. 1 and assume the followings:

- The left side of switches is input and the flow of information is from left to right.
- Each input line, on left side, is attached with a computer having different initial probabilities of selection by users. This level is the stage 1.
- The middle crossbars are stage 2 containing three crossbars with each having four inputs and four output lines.
- The third stage contains four crossbars, each with three inputs and four output lines. At this, three output lines are with computers and the fourth one, in each crossbar, is a loss state.
- The term $I(M,K,L)$ denotes an input state at M^{th} stage in K^{th} crossbar and at L^{th} input line where $M=1,2,3; K=1,2,3,4; L=1,2,3,4$. For example, in fig. 1 the term a_1 is input state $I(1,1,1)$, a_2 is state $I(1,1,2)$, c_1 is $I(1,2,1)$, e_1 is $I(2,1,1)$, g_1 is $I(2,2,1)$, and i_1 is $I(3,1,1)$.
- The term $O(M,K,L)$ denotes output state at M^{th} stage, in K^{th} crossbar and L^{th} output line like the term b_1 is output state $O(1,1,1)$, b_2 is $O(1,1,2)$, b_3 is $O(1,1,3)$, d_1 is $O(1,2,1)$, f_1 is $O(2,1,1)$, h_1 is $O(2,2,1)$ and j_1 is $O(3,1,1)$.

As special, the output states $O(3,1,4)$, $O(3,2,4)$, $O(3,3,4)$ and $O(3,4,4)$ are loss states and when a message reaches to them, it is assumed lost or reached to the known destinations.

3.1. Markov Chain Model

Let $\{X_n, n = 0,1,2,3,\dots\}$ be a Markov chain with state space $I(M,K,L)$ and $O(M,K,L)$, $M=1,2,3$ and $K,L = 1,2,3,4$. The X_n denotes the state of message at the n^{th} step transition over states $I(M,K,L)$ and $O(M,K,L)$. The unit-step transition probabilities over states are:

$$\left. \begin{aligned}
 P[X_{n+1} = O(1,K,1) / X_n = I(1,K,L)] &= L_K \\
 P[X_{n+1} = O(1,K,2) / X_n = I(1,K,L)] &= L_{2K} \\
 P[X_{n+1} = O(1,K,3) / X_n = I(1,K,L)] &= 1 - (L_K + L_{2K}) \\
 P[X_{n+1} = I(1,K,L) / X_n = O(1,K,J)] &= P_{LK}, \\
 &\text{when } J = 1,2,3 \\
 P[X_{n+1} = I(1,K,L) / X_n = O(1,K,J)] \\
 &= 1 - \sum_{i=1}^3 P_{iK} \text{ when } L = 4; K = 1,2,3,4; J = 1,2,3
 \end{aligned} \right\} \text{When } L=1,2,3; K=1,2,3,4$$

$$\begin{aligned}
 P[X_{n+2} = O(2,K,L) / X_{n+1} = I(2,K,L)] \\
 &= P[X_{n+2} = I(2,K,L) / X_{n+1} = O(2,K,L)] \\
 &= Q_{LK}, \text{ when } L=1,2,3; K=1,2,3 \\
 P[X_{n+2} = O(2,K,L) / X_{n+1} = I(2,K,L)] \\
 &= P[X_{n+2} = I(2,k,L) / X_{n+1} = O(2,k,L)] = 1 - \sum_{i=1}^3 Q_{iK}, \\
 &\text{when } L = 4; K = 1,2,3, \\
 P[X_{n+2} = O(2,K,L) / X_{n+1} = I(2,K,L)] \\
 &= P[X_{n+2} = I(2,K,L) / X_{n+1} = O(2,K,L)]
 \end{aligned}$$

$$= Q_{LK}, \text{ when } L = 1,2,3; K = 1,2$$

$$P[X_{n+3} = O(3, K, L) / X_{n+2} = I(3, K, J)] = R_{LK},$$

$$\text{when } L = 1,2,3; K = 1,2,3,4; J = 1,2,3$$

$$P[X_{n+3} = O(3, K, 4) / X_{n+2} = I(3, K, J)]$$

$$= 1 - \sum_{i=1}^3 R_{iK}, \text{ when } L = 4; K = 1,2,3,4; J = 1,2,3$$

$$P[X_{n+3} = I(3, K, 1) / X_{n+2} = O(3, K, L)] = S_{1K}$$

$$P[X_{n+3} = I(3, K, 2) / X_{n+2} = O(3, K, L)] = S_{2K}$$

$$P[X_{n+3} = I(3, K, 3) / X_{n+2} = O(3, K, L)] = 1 - \sum_{i=1}^2 S_{iK}$$

$$\text{when } L = 1,2,3; K = 1,2,3,4$$

The terms $L_{ik}, S_{ik}, P_{ik}, Q_{ik}, R_{ik}$ ($i=1,2,3$) are the probabilities of transition lying between 0 and 1 and placed as elements of transition probability matrix given below.

		States						
		$I(1, K, 1)$	$I(1, K, 2)$	$I(1, K, 3)$	$I(1, K, 4)$	$O(1, K, 1)$	$O(1, K, 2)$	$O(1, K, 3)$
States	$I(1, K, 1)$	0	0	0	0	L_1k	L_2k	$\{1 - (L_1k + L_2k)\}$
	$I(1, K, 2)$	0	0	0	0	L_1k	L_2k	$\{1 - (L_1k + L_2k)\}$
	$I(1, K, 3)$	0	0	0	0	L_1k	L_2k	$\{1 - (L_1k + L_2k)\}$
	$I(1, K, 4)$	0	0	0	0	L_1k	L_2k	$\{1 - (L_1k + L_2k)\}$
	$O(1, K, 1)$	P_1k	P_2k	P_3k	$\{1 - (P_1k + P_2k + P_3k)\}$	0	0	0
	$O(1, K, 2)$	P_1k	P_2k	P_3k	$\{1 - (P_1k + P_2k + P_3k)\}$	0	0	0
	$O(1, K, 3)$	P_1k	P_2k	P_3k	$\{1 - (P_1k + P_2k + P_3k)\}$	0	0	0

TABLE 1: Transition probability matrix (t. p. m.) for stage 1

		States						
		$I(1, K, 1)$	$I(1, K, 2)$	$I(1, K, 3)$	$I(1, K, 4)$	$O(1, K, 1)$	$O(1, K, 2)$	$O(1, K, 3)$
States	$I(1, K, 1)$	0	0	0	0	L_1k	L_2k	$\{1 - (L_1k + L_2k)\}$
	$I(1, K, 2)$	0	0	0	0	L_1k	L_2k	$\{1 - (L_1k + L_2k)\}$
	$I(1, K, 3)$	0	0	0	0	L_1k	L_2k	$\{1 - (L_1k + L_2k)\}$
	$I(1, K, 4)$	0	0	0	0	L_1k	L_2k	$\{1 - (L_1k + L_2k)\}$
	$O(1, K, 1)$	P_1k	P_2k	P_3k	$\{1 - (P_1k + P_2k + P_3k)\}$	0	0	0
	$O(1, K, 2)$	P_1k	P_2k	P_3k	$\{1 - (P_1k + P_2k + P_3k)\}$	0	0	0
	$O(1, K, 3)$	P_1k	P_2k	P_3k	$\{1 - (P_1k + P_2k + P_3k)\}$	0	0	0

TABLE 2: Transition probability matrix (t. p. m.) for stage 2

← States →

	$I(3, K, 1)$	$I(3, K, 2)$	$I(3, K, 3)$	$O(3, K, 1)$	$O(3, K, 2)$	$O(3, K, 3)$	$O(3, K, 4)$
$I(3, K, 1)$	0	0	0	R_{1k}	R_{2k}	R_{3k}	$\{1 - (R_{1k} + R_{2k} + R_{3k})\}$
$I(3, K, 2)$	0	0	0	R_{1k}	R_{2k}	R_{3k}	$\{1 - (R_{1k} + R_{2k} + R_{3k})\}$
$I(3, K, 3)$	0	0	0	R_{1k}	R_{2k}	R_{3k}	$\{1 - (R_{1k} + R_{2k} + R_{3k})\}$
$O(3, K, 1)$	S_{1k}	S_{2k}	$\{1 - (S_{1k} + S_{2k})\}$	0	0	0	0
$O(3, K, 2)$	S_{1k}	S_{2k}	$\{1 - (S_{1k} + S_{2k})\}$	0	0	0	0
$O(3, K, 3)$	S_{1k}	S_{2k}	$\{1 - (S_{1k} + S_{2k})\}$	0	0	0	0
$O(3, K, 4)$	S_{1k}	S_{2k}	$\{1 - (S_{1k} + S_{2k})\}$	0	0	0	0

↑ States ↓

TABLE 3: Transition probability matrix (t. p. m.) for stage 3

3.2. Model Classification

The probabilities L_{ik} , P_{ik} , Q_{ik} , R_{ik} and S_{ik} may be functions of M , K and L parameters and on this basis the classification of Markov chain models be as below:

- M-Dependent model- where probabilities L_{ik} , P_{ik} , Q_{ik} , R_{ik} and S_{ik} are only functions of M .
- K-Dependent model- where probabilities L_{ik} , P_{ik} , Q_{ik} , R_{ik} and S_{ik} are only functions of K .
- L-Dependent model- where probabilities are functions of K and L parameters both.

4. CALCULATION OF REACHING (INITIAL) PROBABILITIES

Let P_{ik} ($i = 1, 2, 3$) be the probability of choosing the i th input line in K th switching element of the space division switch configuration given in fig. 1 of the section 1.0. For $i = 4$, the probability is

$$\left\{ 1 - \sum_{i=1}^3 p_{ik} \right\}.$$

For the Markov chain $\{X_n, n = 0, 1, 2, 3, \dots\}$ over the states $I(M, K, L)$, the initial probabilities of choosing a connecting path is

$$P[X_0 = I(1, K, 1)] = p_{1k}, P[X_0 = I(1, K, 2)] = p_{2k}$$

$$P[X_0 = I(1, K, 3)] = p_{3k},$$

$$P[X_0 = I(1, K, 4)] = p_{4k} = 1 - (p_{1k} + p_{2k} + p_{3k}) = \left\{ 1 - \sum_{i=1}^3 p_{ik} \right\}$$

4.1. Outgoing Probabilities At Stage 1, 2 and 3

The $O(1, K, L)$ over varying K and L are the outgoing states, for the stage 1, where the message is ready to route into for the next stage.

$$P[X_1 = O(1, K, L)]$$

= P[message reaches to the state $O(1, K, L)$ at the first step]

The general form for $M = 1$ (stage-1) is

$$P[x = O(1, k, L)] = L_{1k}$$

when $L = 1; K = 1, 2, 3, 4$ }>

$$= L_{1^*k} \text{ when } L = 2$$

$$= \{1 - L_{1k} + L_{2k}\} \text{ when } L = 3$$

The general form for $M = 2$ (stage-2) is

$$P[x = O(2, k, L)] = Q_{LK} \sum_{i=1}^4 L_{1i} \text{ when } k = 1, L = 1, 2, 3$$

$$P[x=O(2,k,L)] = \{1 - \sum_{i1}^3 Q_{ik}\} \sum_{i=1}^4 L_{1i} \text{ when } k = 1, L = 4$$

$$P[x= O(2, k, L)] = Q_{LK} \sum_{i=1}^4 L_{2i} \text{ when } k = 2, L = 1, 2, 3$$

$$P[x= O(2,k,L)] = \{1 - \sum_{i1}^3 Q_{ik}\} \sum_{i=1}^4 L_{2i} \text{ when } k = 2, L = 4$$

$$P[x=O(2,k,L)] = Q_{LK} \{4 - \sum_{i=1}^4 L_{1i} - \sum_{i=1}^4 L_{2i}\} \text{ when } k=3, L = 1, 2, 3$$

$$P[x = O(2, k, L)] = \{1 - \sum_{i1}^3 Q_{ik}\} \{4 - \sum_{i=1}^4 L_{1i} - \sum_{i=1}^4 L_{2i}\} \text{ when } k = 3, L = 4$$

The general form for $m = 3$ (stage-3) is

$$P[x = O(3, k, L)] = R_{LK} * \left[Q_{k1} \sum_{i=1}^4 L_{1i} + Q_{k2} \sum_{i=1}^4 L_{2i} + Q_{k3} \left\{ 4 - \sum_{i=1}^4 L_{1i} - \sum_{i=1}^4 L_{2i} \right\} \right]$$

when $k = 1, 2, 3 ; L = 1, 2, 3$

$$P[x=O(3,k, L)] = R_{LK} * \left\{ 1 - \sum_{i=1}^3 R_{ik} \right\} * \left[Q_{k1} \sum_{i=1}^4 L_{1i} + Q_{k2} \sum_{i=1}^4 L_{2i} + Q_{k3} \left\{ 4 - \sum_{i=1}^4 L_{1i} - \sum_{i=1}^4 L_{2i} \right\} \right]$$

when $k = 1, 2, 3, L = 4$

$$P[x=O(3,k,L)] = R_{LK} \left[\left\{ 1 - \sum_{i=1}^3 Q_{i1} \right\} \sum_{i=1}^4 L_{1i} + \left\{ 1 - \sum_{i=1}^3 Q_{i2} \right\} \sum_{i=1}^4 L_{2i} \right. \\ \left. + \left\{ 1 - \sum_{i=1}^3 Q_{i3} \right\} \left\{ 4 - \sum_{i=1}^4 L_{1i} - \sum_{i=1}^4 L_{2i} \right\} \right]$$

when $K = 4, L = 1, 2, 3$

$$P[x=O(3,k,L)] = \left\{ 1 - \sum_{i=1}^3 R_{ik} \right\} * \left[\left\{ 1 - \sum_{i=1}^3 Q_{i1} \right\} \sum_{i=1}^4 L_{1i} + \left\{ 1 - \sum_{i=1}^3 Q_{i2} \right\} \sum_{i=1}^4 L_{2i} \right. \\ \left. + \left\{ 1 - \sum_{i=1}^3 Q_{i3} \right\} \left\{ 4 - \sum_{i=1}^4 L_{1i} - \sum_{i=1}^4 L_{2i} \right\} \right]$$

when $k = 4, L = 4$

5. K-Dependent Model and Simulation Study

Consider the following K-dependent Markov chain model with unit-step transition probabilities

$$P[X_1 = O(1, K, J) / X_0 = I(1, K, L)] = K(a)^K \text{ when } J = 1, 2 \\ P[X_1 = O(1, K, J) / X_0 = I(1, K, L)] = \{1 - 2K(a)^K\} \text{ when } J = 3 \Bigg\} L = 1, 2, 3, 4$$

$$\begin{aligned}
 &P[X_1 = I(1, K, J)/X_0 = O(1, K, L)] = K(b)^K \quad \text{when } J = 1,2,3 \\
 &P[X_1 = I(1, K, J)/X_0 = O(1, K, L)] = \{1 - 3K(b)^K\} \quad \text{when } J = 4 \quad \left. \vphantom{P[X_1 = I(1, K, J)/X_0 = O(1, K, L)]} \right\} L = 1,2,3 \\
 &P[X_2 = O(2, K, J)/X_1 = I(2, K, L)] = K(c)^K \quad \text{when } J = 1,2,3 \\
 &P[X_2 = O(2, K, J)/X_1 = I(2, K, L)] = \{1 - 3K(c)^K\} \quad \text{when } J = 4 \quad \left. \vphantom{P[X_2 = O(2, K, J)/X_1 = I(2, K, L)]} \right\} L = 1,2,3,4 \\
 &P[X_2 = I(2, K, J)/X_1 = O(2, K, L)] = K(c)^K \quad \text{when } J = 1,2,3 \\
 &P[X_2 = I(2, K, J)/X_1 = O(2, K, L)] = \{1 - 3K(c)^K\} \quad \text{when } J = 4 \quad \left. \vphantom{P[X_2 = I(2, K, J)/X_1 = O(2, K, L)]} \right\} L = 1,2,3,4 \\
 &P[X_3 = O(3, K, J)/X_2 = I(3, K, L)] = K(d)^K \quad \text{when } J = 1,2,3 \\
 &P[X_3 = O(3, K, J)/X_2 = I(3, K, L)] = \{1 - 3K(d)^K\} \quad \text{when } J = 4 \quad \left. \vphantom{P[X_3 = O(3, K, J)/X_2 = I(3, K, L)]} \right\} L = 1,2,3 \\
 &P[X_3 = I(3, K, J)/X_2 = O(3, K, L)] = K(e)^K \quad \text{when } J = 1,2 \\
 &P[X_3 = I(3, K, J)/X_2 = O(3, K, L)] = \{1 - 2K(e)^K\} \quad \text{when } J = 3 \quad \left. \vphantom{P[X_3 = I(3, K, J)/X_2 = O(3, K, L)]} \right\} L = 1,2,3,4
 \end{aligned}$$

Where a,b,c,d and e are constants having values in between 0.00 to 0.5. These transition probabilities are not by the variation in L. Because of being K-dependent model, the idea for this form of probability is to consider probabilities in power function of K.

5.1 Reaching Probabilities Over Stages The Fig. 5.1 to fig. 5.4 shows the variation of reaching probabilities $P[M,K,1]$ over K according the values of constants $a=0.1, c=0.1, 0.1 < d <= 0.5$.

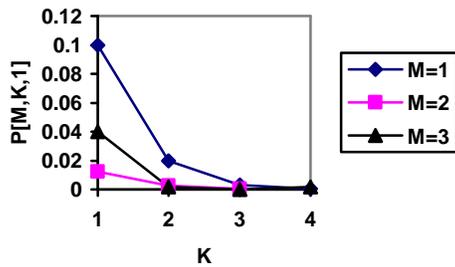


Fig. 5.1

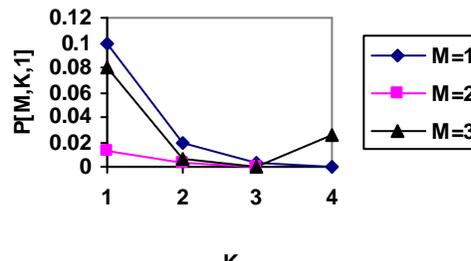


Fig. 5.2

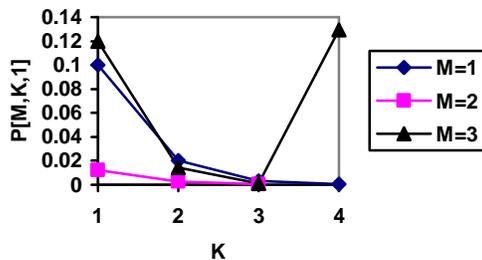


Fig. 5.3

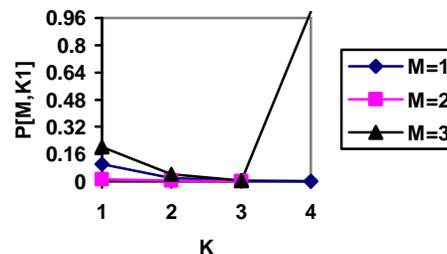


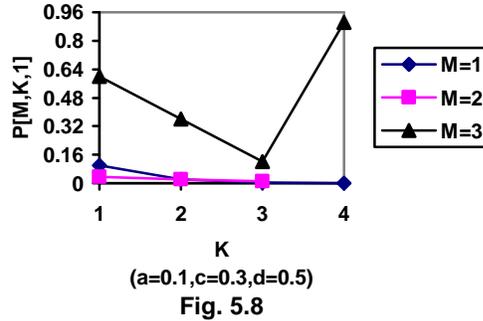
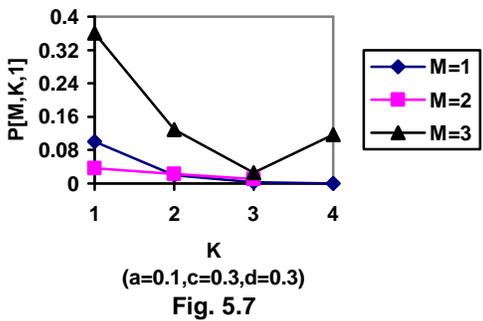
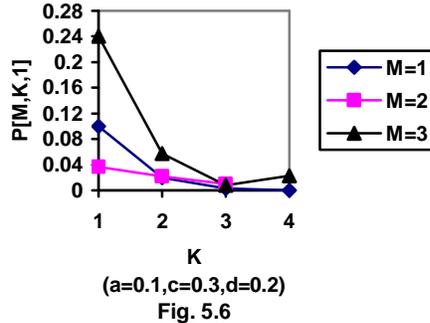
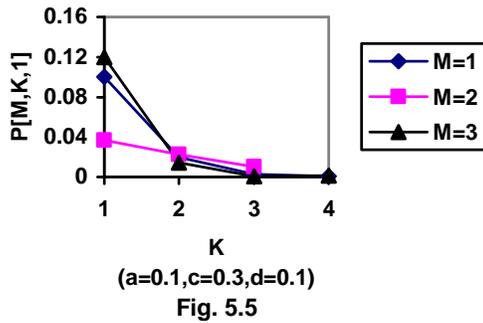
Fig. 5.4

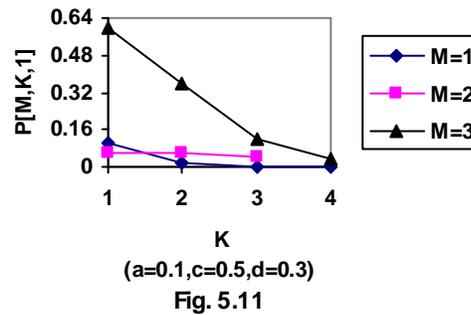
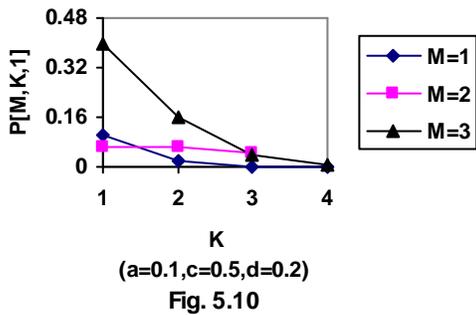
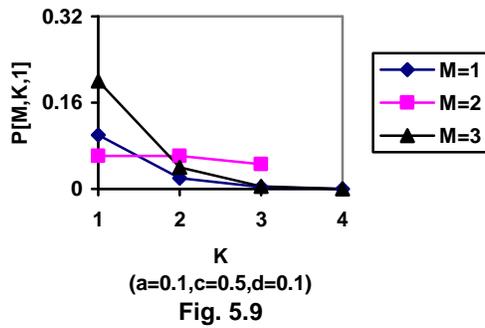
In light of assumed unit-step transition probability model, for the constant value of $k=1$, the connecting probability $P[M,K,1]$ decreases at second stage ($M=2$) with respect to first stage

(M=2), but increases for(M=3). At K=2 and K=3 the similar pattern has observed with relatively closer probability difference.

With the small values of d i.e. for $d = 0.1$ or $d = 0.2$, a sudden increase of probability $P[M,K,1]$ has observed at $K=4$ for $M=3$ than compare to $M=1$. The outgoing probability at first stage ($M=1$) is highest, followed by third stage $M=3$ but congestion occurs at intermediate crossbar $M=3$. It is because of the fact that several output lines reaches to the middle crossbars. For fixed value of M , the increase in value of K has the most significant impact in reducing the probability $P[M,K,1]$. But, exceptionally at $K=4$, a sudden increase has observed for large value of d say $d > 0.4$. One more thing is observed that the reaching probability $P[M,K,1]$ increases for $M=3$ With respect to increase value of d .

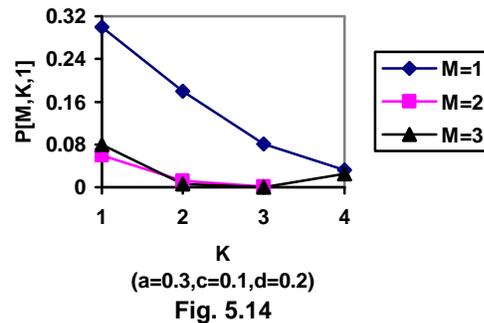
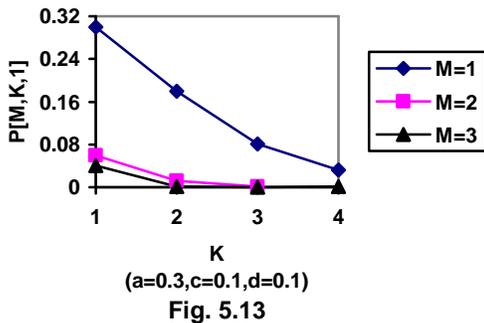
According to fig.5.5 to 5.9 we observe that the variation over parameter c along with d also produces a change in the probability level. The third stage has constantly high probability than others. At $K=1$, the third stage probability $p[3,1,1]$ is higher than others. But on $K=2$, we have $P=1,2,1]=p[2,2,1]$ and a decreasing pattern for $M=3$. The decrease over K continues up to $K=3$, but a sudden increase has found thereafter which is drastic when d is large. As shown in fig. 5.10 to 5.12, the pattern of probability distribution differs much at the third stage for large values of c and d .

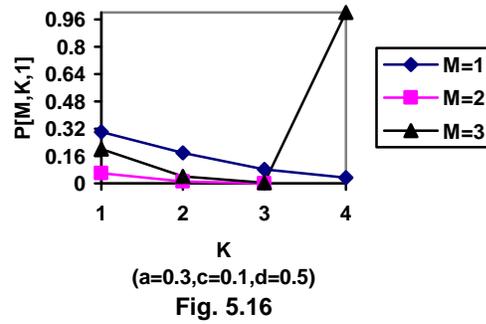
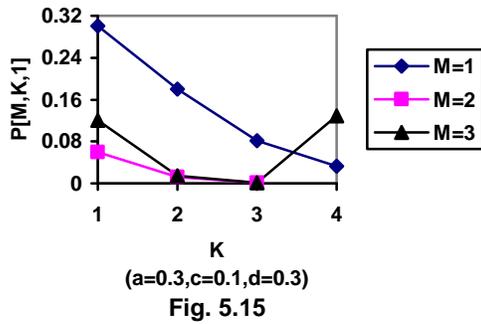




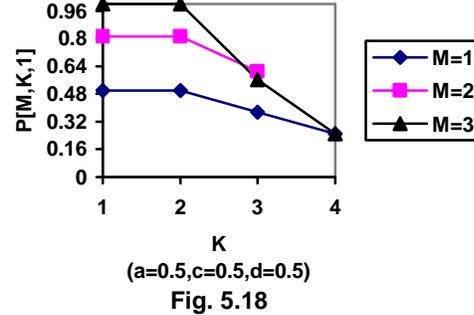
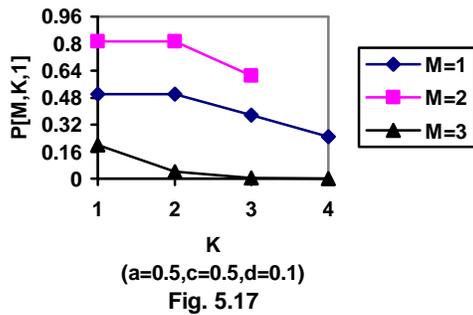
At the third stage, the probability $P[M, K, 1]$ becomes high and decreases gradually over increasing value of K . As shown in fig. 1.15, the higher value of parameters c and d both has a significant effect on reaching probabilities at the third stage.

In light of fig. 5.13 to 5.16, it is observe that these parameters drastically changes the probability pattern more than the variation of c and d . At $a = 0.3$, the first stage has observed the higher probability than the others. The increase in values primarily changes the outgoing probability.





While looking in fig. 5.17 to 5.18, it is found that for higher value of a and c both with respect to different values of d, a sudden variation arises at the third stage of probability and the reaching probability reduces constantly over the increases in K. But at this stage, the entire pattern of variation seems to get stabilize over the variation of K and probabilities are nearly parallel to the X-Axis.



The reaching probabilities at stage 2 increases than other stages when a is higher than c & d both. But with increasing values of c and d, when a=0.5 creates a decreasing pattern of probability over increasing values of K. However, the probability M=3 reduces than M=1 and M=2 at many occasions for large values of a. The uppermost element of third stage has highest chance of outgoing probability as shown in fig. 5.18, when a=c=d=0.5.

6. CONSLUSION

Some interesting highlights of the simulation study are concluded bellow.

- (i) Under the assumed transition probability model the outgoing probabilities from crossbars reduces at the intermediate links(crossbar at M=2)for small values of d(d=0.1 or d=0.2). The higher values of d increases the reaching probabilities P[M,K,1] for M=3. This seems at stage 3, the higher choice of value d i.e. $d \in (0.3,0.5)$ is recommended for better chances connectivity (when a=0.1 and c=0.1)).
- (ii) The simultaneous increase in c and d values i.e. $c,d \in (0.3,0.5)$ also increases the outgoing chances of passing the message at the third stage. The unequal probability distributions between switching elements at the second and third stage increases the outgoing probability at the last stage. It means that unequal probability of outgoing message through three pins of crossbar plays a significant role. The unequal probability allocation to switching pins improves the message passing chances.
- (iii) The increment in values of a, produces high outgoing probabilities at the first stage but relatively low outgoing at the third stage P[3,1,1]. Therefore, smaller values of a, $a \in (0.1,0.2)$ is recommended in order to get high outgoing probabilities at the third stage.

- The unequal transition probabilities over K and M definitely affects the outgoing probabilities at the third stage. However, for $k > 2$, a slight downfall in the message passing probability is observed.
- (iv) As a special case, when $a=0.5, c=0.5$ and $d=0.5$, the outgoing probability becomes independent of K for $K=1$ and $K=2$ and depends for $K > 2$. Further, this reveals a special feature that outgoing probability at the third stage is constantly higher than any other stages. Equal values of parameters a, c, d generates higher chance of passing the message through the third stage.
 - (v) While looking into the variation of d , the probability $P[3, K, 1]$ increases as d increases for $K=4$ only up to the stage where $\epsilon (0.1, 0.3)$. When $c > 0.3$ the outgoing probability $P[3, K, 1]$ becomes high for $K=1$. This seems if message is to pass from the first element of the third stage ($M=3, K=1$) the higher values of c and d are suitable (e.g. $c=0.5, d=0.5$) and if the same is to pass through fourth element of third stage ($M=3, K=4$) the small c and large d (e.g. $c=0.1, d=0.5$) is required. This reveals that the choice of c and d highly affect the outgoing probabilities but parameter a does not have so.
 - (vi) In K -dependent model, the increase in parameter a , has very important role in deciding about the probability pattern of outgoing message.
 - (vii) One interesting observation in three pin case found as for $L=4$. The reaching probability is much higher. Moreover, on more specific observation is that at $K=1$, a linear trend is found for increasing values of d when a, c are fixed.

In all, in space-division switches, the outgoing probability at the third stage under K -dependent Markov chain model is highly dependent on the appropriate choice of parameters a, c and d . If the transition inside the switching elements are preset as per model probability then the passing of message through certain connecting lines shall be easy in terms of chances. This recommends to the switch designers to construct space-division switches with unequal transition probabilities within elements and between elements. So we can conclude that the hardware designers of space division switches must keep in mind the recommended values of different parameter respectively for getting better chance of connectivity.

7. REFERENCES

1. Abott, G.F. "digital space division –a technique for switching high-speed data signals", IEEE communications magazine, vol. 22, no. 4, pp. 32-38 (1984).
2. Cao, X.-R. "The maximum throughput of a nonblocking space-division packet switch with correlated destinations", IEEE Transactions on Communications, Vol. 43, No.5, pp.1898-1901, 1995.
3. K.-T. Ko and B.R. Davis, "A space-division multiple-access protocol for spot-beam antenna and satellite switched communication network" IEEE Journal on Selected Areas in Communication 1(1), 126–132, 1983.
4. Lee, M.J. and Li, S.-Q. "Performance of a nonblocking space-division packet switch in a time variant nonuniform traffic environment", IEEE Transactions on Communications, Vol. 39, No. 10, pp. 1515-1524, 1991.
5. Li, S.-Q "Nonuniform traffic analysis on a nonblocking space-division packet switch", IEEE Transactions on Communications, vol. 38, no.7, pp. 1085-1096, 1990.
6. Li, S.-Q "Performance of a nonblocking space-division packet switch with correlated input traffic", IEEE Transactions on Communications, vol. 40, no.1, pp. 97-108, 1992.
7. M.J. Karol, M.G. Hluchyi and S.P. Morgan, "Input versus output queuing on a space-division packet switch", IEEE Transaction on Communications 35 (12), 1347–1356, 1987.
8. Medhi, J. "Stochastic Processes", Ed 4, Wiley Eastern Limited (Fourth reprint), New Delhi, (1991).
9. Naldi, M. "Internet access traffic sharing in a multi operator environment", Computer Network, vol. 38, pp. 809-824, 2002.
10. Pao, D.C.W. and Leug, S.C. "Space division-approach to implement a shared buffer in an ATM switch", Computer Communications, vol. 20, Issue 1, pp. 29-37, 1997.

11. Shukla, D., Gadewar, S. and Pathak, R.K. "A Stochastic model for space-division switches in computer networks", Applied Mathematics and Computation (Elsevier Journal), Vol. 184, Issue 2, pp.. 235-269, 2007.
12. Shukla, D., Singhai R. and Gadewar S.K., "Markov Chain Analysis for Reaching Probabilities of Message Flow In Space-Division Switches", In electronic proceedings of ICMCS-08, Loyola College, Chennai, India , 2008.
13. Tanenbaum, A.S. "Computer Network", 3rd Ed., Prentice-Hall, Inc., USA(25th Indian reprint), (1996).
14. Wang. W. and Tobagi, F. A. "The Christmas-tree switch: an output queuing space-division fast packet switch based on interleaving distribution and concentration functions", Computer Networks and ISDN Systems, vol. 25, Issue 6, pp. 631-644,1993.
15. Yamada, H., Kataoka, H., Sampei, T. and Yano, T. "High-speed digital switching technology using space-division switch LSI'S", IEEE Selected Areas in Communications, vol. 4, no. 4, pp. 529-535, 1986.
16. Yamanka, N, Kikuchi, S., Suzuki, M. and Yoshioka, Y. "A 2 Gb/s expandable space-division switching LSI network architecture for gigabit-rate broad-band circuit switching", IEEE Selected Areas in Communications, vol. 18, no. 8, pp. 1543-1550,1990.

