J. Basic. Appl. Sci. Res., 2(11)11626-11643, 2012
© 2012, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Understanding Pull-based Method Efficiency in Peer-to-Peer Live Video Streaming over Mesh Networks

Behzad Soleimani Neysiani[1*], Naser Neematbakhsh[2], Behrang Barekatain[3], Mohd Aizaini Maarof[3], Kamran Zamanifar[2]

[1*] Faculty of Computer Engineering, NajafAbad Islamic Azad University, NajafAbad, Iran
[2]Faculty of Computer Engineering, Isfahan University, Isfahan, Iran
[3]Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Johor Bahru, Malaysia

ABSTRACT

In the recent years, multimedia streaming has been really popular among end users, especially live video streaming over the Internet. However, video streaming applications need efficient and reliable methods for disseminating video frames in the networks. Although Mesh-based Peer-to-Peer (P2P) networks make the system more robustness against link failures, the performance of the system can be degraded considerably if a peer joins or leaves the network. Consequently, the necessity of an efficient exchange method for degrading the side effects of these problems is inevitable. Contrary to the push-based exchange method, which is more suitable for tree-based P2P systems, the pull-based technique is used in many previous mesh-based systems. To best of our knowledge, no previous study evaluated the efficiency of this exchange method in various peer churning rates and network conditions. In this work, the efficiency of the pull-based exchange method in mesh-based P2P live video streaming is analyzed and examined for different peer churning rates, network sizes, initial buffer times, BMS (Buffer Map Status) exchange interval times and heterogeneous bandwidth. Moreover, a trade-off between the BMS message interval time and the network overhead in pull-based method is evaluated in order to clearly show why most of the previous studies used one second BMS interval time. The obtained results are discussed based on five important performance metrics including playback skip, End-to-End delay, dissemination time, initial startup delay and network overhead to provide a comprehensive evaluation of pull-based method.

KEYWORDS: Pull, Peer-to-Peer Networking, Mesh Networks, Peer Churning, Performance Evaluation

1. INTRODUCTION

Video streaming will be the most important traffic in the Internet in near future [1]. Video-on-Demand (VoD) and live video are two classifications of video streams [2]. In live video streaming, a source, such as a live camera, disseminates live video frames to destinations. Video playback is synchronized among receivers and some operations such as backward and pause cannot be performed due to the live nature of the video stream. Contrary to live video streaming, all receivers watch the video stream from the first frame irrespective of the time that they join the network in VoD-based systems. It means that VoD stream is not synchronized among nodes [3]. This allows VoD-based systems to be more flexible in comparison with live video streaming applications. YouTube [4] and PPLive [5] are two VoD and live systems, respectively. Interactive and non-interactive are two types of video applications which use live and VoD streaming techniques, respectively [6]. Large bandwidth, efficient network infrastructure and high performance frame exchange method are three important factors for providing smooth video playback on receivers. Node churning is another factor which has enormous effects on the perceived video quality. Churning will be occurred if a node moves in the network so that it loses its current neighbors or leaves the network [7]. As a result, a flexible infrastructure for video dissemination is needed to support this event efficiently. Peer-to-Peer networks provide this infrastructure [8]. However, providing high video quality over time-varying channels is one of the most important problems in P2P video streaming, especially in wireless networks. Constant Bit rate (CBR) [9] is used in ISDN (Integrated Service Digital Network), because it does not need variable bit rate for data transmission. On the other hand, in some networks, encoding the video stream with same bit rate during the transmission is very difficult; due to delay, available bandwidth and some other parameters which are not clearly specified before and during each transmission. Therefore, Variable Bit Rate (VBR) [9] technique can be used to achieve desired video quality on receivers. As mentioned before, video application systems introduce high video quality if they can support three important requirements [10]. First of all, an efficient video compression method so that it not only increases bandwidth utilization, but also provides acceptable video quality for peers with low available bandwidths. Moreover, a flexible and reliable infrastructure in order to decrease the side effects of existing challenges such as node and link failures. Finally, although the two first requirements increases video quality, existence of an effective exchange method is really important for providing smooth video playback on end receivers, especially if these nodes can join or leave the network repeatedly.

The rest of this study is organized as follows. Video compression standards and Peer-to-Peer networks will be discussed in sections II and III, respectively. Section IV explains existing exchange methods in P2P networks for video streaming. The main research question of this study and related simulation results will be propounded and presented in section V. Finally, the paper will be concluded in section VI.

***Corresponding Author:** Behzad Soleimani Neysiani, Faculty of Computer Engineering, NajafAbad Islamic Azad University, NajafAbad, Iran, Email: Sabarg@sco.iaun.ac.ir

2. VIDEO COMPRESSION STANDRDS

Large bandwidth and enough buffers are the two most important requirements for providing smooth video playback on end receivers. However, in many P2P networks, it is difficult to provide enough download/upload bandwidth for all nodes, especially mobile gadgets such as mobile phones or iPods. Therefore, employing efficient video compression techniques is really necessary. This not only decreases the required bandwidth for video dissemination, but also relieves nodes from large buffers. In the last decade, many different video compression standers such as MPEG-4 are introduced. Among all existing video compression standards, the H.264/AVC (Advanced Video Coding) [11] introduces high performance methods in compressing and encoding the video frames [12]. In these standards, as is illustrated in figure 1, frames are classified into many GoPs (Group-of-Pictures). In the H.264/AVC standard, depending on different profiles, different order of frames in a GOP can be encoded by the encoder. In fact, each GOP consists of frames I, B and P. Frame I (Intra frame) is the most important frame which can be decoded independently and used by frames P and B as a reference frame. Frame P (Predictive inter frame) uses one of the previous frames I or P as a reference frame in decoding, while a frame B (Bi-predictive frame) uses both last and next frames I or P as its reference frames [2]. A reference frame can be used by a decoder to recover damaged frames. Figure 1 depicts a classical GOP. In fact, differential frames B and P include fewer bits in comparison with frame I. Hence, higher compression rate is obtained by this standard with same quality in comparison with other video compression standards. Consequently, this standard increases bandwidth utilization more than that of others which is very important in live video streaming, because the source can send more frames in each transmission. This leads to higher throughput of the system and better video quality. Irrespective of many provided benefits by these video compression methods, there are two main problems within them. First, albeit high performance video compression method leads to better video quality on receivers, existing dependencies among frames can increase number of playback skips and finally decrease perceived video quality. For example, according to figure 1, no P-frame is decodable if previous frame I or P has not yet received. Second, it is very important to receive these frames before their playback time.

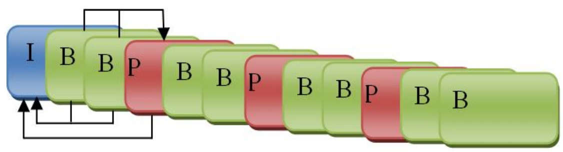


Figure 1. A Simple GoP Structure (G12B2)

In fact, a complete received frame will be ignored if it is received after its playback time. Peer-to-Peer networks degrade the effects of these problems on the perceived video quality by sharing the available resources of peers such as upload bandwidth. In the next section, advantages and disadvantages of different P2P networks are explained in more details.

3. PEER-TO-PEER NETWORKS

Peer-to-Peer networks have been the most important employed infrastructure for data dissemination over the Internet such as file sharing and video streaming. For example, broadcasting of live video stream over some distributed applications such as PPLive [4], CoolStreaming [5] and SopCast [6] has been very popular in the recent years. Moreover, BitTorrent [7] and IPTV [8] employ P2P networks for file sharing and VoD streaming, respectively. P2P networks can be implemented in the application layer as an overlay on top of the existing underlying network. Overlay networks [13] relieve us from changing in the functionality of the network layer. For example, although IP multicast [14] technique provides considerable efficiency in video broadcasting, it imposes high cost to the system. P2P networks are classified into two categories: Tree-based and Mesh-based systems [15]. Tree-based systems are divided into Single-Tree and Multi-Tree structures. Each peer follows following steps to join a tree: (1) each new peer sends a join message to the root node, (2) the root replies it by introducing a proper position in the tree based on the existing policies of the tree construction (3) the new peer joins the network based on the introduced position by the root.

When a peer leaves the network, the tree will be divided into two subtrees. Therefore, the number of playback skips increases in the system. Moreover, the most important drawback of tree-based systems is that no leaf participates in video dissemination, because there is no child under it. Multi-tree systems are introduced to address the last problem. In a Multi-tree system, each peer is as an internal node in only one and as a leaf in all other subtrees. Consequently, all peers participate in video streaming equally [9] which results in higher video quality in the system. However, like tree-based system, peer churning has considerable effects on network throughput and perceived video quality. Figure 2 shows both tree-based and multi-tree-based structures. Here, green node is as an internal node in the middle subtree and as a leaf node in the first and the third subtrees.

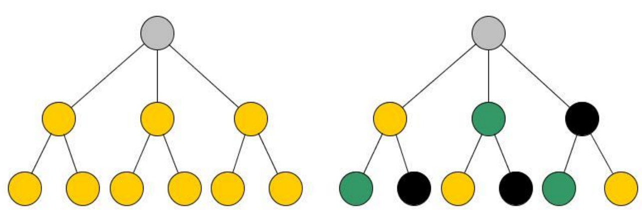


Figure 2. A Tree (left) and a Multi-Tree (right) structure

NICE [13], SplitStream [16] and ZIGZAG [14] are three implemented overlay distribution networks based on multicast-tree. Although tree construction is a simple low cost process, the tree reconstruction due to peer churning imposes high cost to the system. Mesh-based systems are introduced to address the mentioned problems in tree-based networks. When a peer decides to join the network, it follows these steps:

1. It sends a join message to a tracker or bootstrap node
2. The tracker sends back a partial list consisting of the addresses of some existing active peers in the network
3. The new peer sends connection request messages to some or all of the introduced peers
4. Peers send back accept or deny messages to the new peer based on their policies. One of these policies can be the maximum number of neighbors of each peer
5. The new peer establishes its connections to network.

What makes mesh-based systems so interesting is that they are really more robustness in link failure in comparison with tree-based systems, because each peer has enough redundant links to others. Therefore, the overall throughput of the system can be increased by sharing the resources of peers equitably. Peers in a P2P system use two exchange methods for video dissemination including pull and push. Pull-based method is used in mesh-based systems, while tree-based networks employ push approach. Next section describes these two methods in more details. The main focus of this study is to evaluation of the pull-based method precisely. Actually, many previous studies performed in depth surveys definitions and classifications of on P2P systems. Therefore, this study refers interested researcher to [7, 10, 17-19] for more information. Moreover, some recent researches such as [20] present surveys on P2P overlay construction and scheduling.

4. PUSH-BASED AND PULL-BASED METHODS

Although mesh-based systems provide flexible and reliable infrastructures for data dissemination, the necessity of employing an efficient data exchange method is not negligible. Pull and Push are two main methods for delivering video frames in P2P networks [21-23]. Push-based method is more suitable for upload-constrained applications where there are few sources and many receivers. Contrary to push, pull-based method is more useful when there are many sources, but few receivers. It means that pull-based systems are suitable for download-constrained applications. An efficient VoD (Video-on-Demand) streaming method based on pull approach is proposed in [24] to serve a large population of users. A distributed incentive approach based on the pull method is evaluated in [25] n order to propagate layered-live video stream in a mesh-based P2P system. In [26], a fluid approach of BitTorrent file sharing with pull-based method is proposed using mathematical analysis to measure the delivery ratio provided by it. The efficiency of pull approach is measured in [21] based on just two performance metrics including delivery ratio and playback delay. However, the effects of different churn rates did not examined in it. Moreover, like other studies, it mentioned that pull-based approach imposes high network overhead to the network; however, no evaluation on it is performed. All in all, few works evaluated pull-based method in P2P systems and all of them considered few performance metrics with no churn event in the system. In addition, this study introduced a hybrid push-pull approach and compared it with pull-based method. Hybrid method, a new solution based on both pull and push techniques, is introduced to provide better performance in data exchanging. Each peer pushes just received video frames to some of its neighbors randomly. Therefore, although push-based method drastically decreases the end-to-end delay in both mesh- and tree-based systems, it imposes a large number of duplicated video frames in mesh-based systems due to blindly forwarding of frames which results in considerable waste of bandwidth [22, 27, 28]. On the other hand, pull-based method diminishes the number of duplicated video frames dramatically, while it increases the end-to-end delay. Each peer in a pull-based system sends its Buffer Map Status (BMS) in a specific interval time. Each neighbor, which receives this BMS, compares it with its local buffer status and sends a request message which includes the list of missed video frames in its buffer to its neighbor. Neighbors who receive these requests will send requested video frames immediately based on their upload policies. In the recent years, different hybrid push-pull-based methods are introduced to exploit the efficiencies of both pull and push techniques [21, 22, 27-31]. However, their proposed methods usually inherit existing problems in them. To examine these hybrid methods is out of the scope of this study. In near future, this study will analyze the efficiencies of the most important recent proposed hybrid push-pull methods in same conditions. Actually, most of the recently studies, especially those which used network coding [32-34] for efficient video streaming, employs pull-based method for video dissemination over mesh-based P2P systems [35-39]. As a result, this research determined to examine the real efficiency of this method for mesh-based P2P live video streaming in various network conditions. Moreover, peer churning is one of the most important challenges and as an open issue in all P2P systems [40]. Therefore, many different plans are considered for this parameter in the simulation. Next section presents and discusses the obtained results of these assessments.

5. SIMULATION AND DISCUSSION

5.1. Simulation parameters and conditions

For better understanding of the efficiency of pull-based method in high peer churning for P2P live video streaming over mesh-based networks and in order to examine some important open issues in the recent studies such as [10, 41], a precise fully discrete-event-based simulator is designed, implemented and verified using C# programming language. The main goal of using simulator is that it is possible to check, evaluate and observe the behavior of the system carefully and precisely in comparison with implementing it on real testbed such as PlanetLab. However, this study determined to perform this evaluation over PlanetLab in near future. To best of our knowledge, no previous work examined the efficiency of pull-based method in P2P live video streaming in various and different churn rates and network conditions. Moreover, recent studies such as [10, 22, 27, 30] confirmed that there is a trade-off between the BMS exchange interval time and the network overhead. However, none of them performed in depth analysis on this trade-off. Another important aim of this study is to find a suitable answer for this question. Five important performance metrics including playback skip, end-to-end delay, network overhead, initial startup delay and dissemination time are measured by this simulator. The third metric refers to the ratio of the number of useless frames, control and notification messages (e.g. BMS and frame request traffics) to the total number of exchanged messages. One video source disseminates a real video stream sample, Lord of the Ring II (CIF=352x288 and Quantizer Parameter=10), from [42]. The video server propagated 10 minutes of this video stream into the network. Each GoP consists of 12 frames IBBPBBPBBPBB (G12B2). Tables 1, 2 and 4 depict considered parameters and various values for them in the simulation. The fraction of peers is based on the precise measurements by [43]. Moreover, figure 3 and table 3 show 13 different plans for peer churning (joining and leaving the network) which are classified in four different scenarios. Actually, table 3 describes the figure 3 in more details for new interested researchers who want to have more information about the Weibull Distribution model. The thirteenth plan includes no churn. All mentioned join and leave times of peers in table 3 are based on the averaged obtained values for different number of considered peers in the simulation (Table 1). These times are very important for discussing on the obtained results in this section.

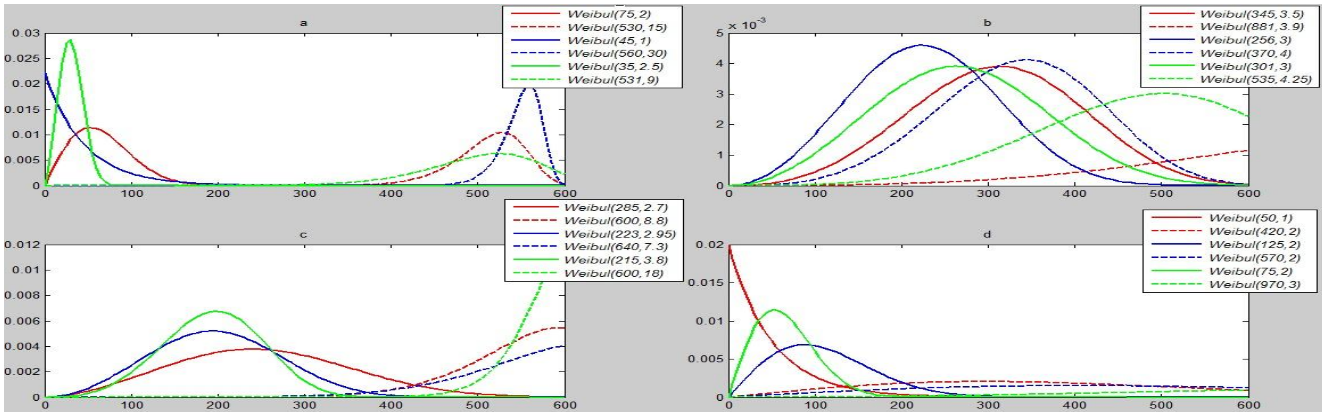


Figure 3. Different Considered Scenarios for Peer Churning including (a) Scenario A (b) Scenario B (c) Scenario C and (d) Scenario D. The axis X shows the live video stream length in second.

These scenarios are carefully designed and each of them consists of three different plans for peer churning. In order to provide comprehensive results, these plans include several flash crowd cases. These scenarios are based on the Weibull distribution (m,n) with mentioned PDF function in table 1. In figure 3, the solid and the dashed lines depict related Weibull distributions for peer joining and leaving, respectively. In the thirteenth plan, no peer churning is occurred. In other words, all peers join the network in the start and remain up to the end of the simulation. The error probability is this simulation was between zero and 10^{-5} which is higher than expected in Ethernet networks. So, this research evaluated the network under lossy channels. Moreover, peers selected 6 neighbors averagely using tracker-based system in [23]. Therefore, the network evaluated under more realistic situation. In our simulation, all peers wait for 500 milliseconds as soon as they receive the first BMS from a neighbor. Although this can slightly increase the overall end-to-end delay in the system, it prevents peers to send many requests to that neighbor which leads to making it as a bottleneck in the system. Based on our experiments, the side effects of existence of some peers as bottleneck points in the system considerably increase the end-to-end delay. Finally, the simulation ran for five times based on the each network conditions set on a cluster-based system consisting of three powerful servers and the averaged results of each conditions set for the considered performance metrics obtained as depicted in the following figures.

Table 1. Considered Parameters in the Simulation

Parameter	Value
BMS Exchange Interval Time	{500,1000,1500} milliseconds
Number of Peers (Network Size)	{50,150,350,600}
Peer Churning (Joining and Leaving the Network)	Five different scenarios including 13 plans
Initial Buffer Time (IBT)	{5,7,10,15} seconds
Number of Neighbors	Random in [4..7]
Network Area Size (Random distribution model is used for positions of peers)	175 ×350 km^2
Employed Weibull PDF Function in Peer Churning	$F(x \mid m,n) = n m^{-n} x^{n-1} e^{-(\frac{x}{m})^n} I_{(0,\infty)}(x)$

Table 2. Fraction of Peers in the Mesh-based System

Fraction of Peers (%)	Download Bandwidth	Upload Bandwidth	Minimum Cross Traffic	Maximum Cross Traffic
16.4	3968	3064	10	17
21.5	3968	2784	8	14
10.8	3968	2424	6	11
15.7	3968	2136	5	7
10.9	3968	1960	4	6
12.8	3968	1656	3	5
11.9	3968	1424	1	3

Table 3. Different Scenarios and Plans for Peer Churning Including Averaged Joining and Leaving Times in Second

Scenario	Peer Joining	Peer Leaving	Plan Num.	Color of Line	The First Peer Arrives at	The Last Peer Arrives at	The First Peer Leaves at	The Last Peer Leaves at	Percentage of Existing Peers in the Network in the 600th Second
A (Figure 3.a)	Abruptly	Suddenly	1	Green	0	70	332	599	6.08%
			2	Blue	0	177	472	594	0.08%
			3	Red	0	165	392	592	0.28%
B (Figure 3.b)	Gradually	Moderately	4	Green	62	480	212	599	20.760%
			5	Blue	47	415	125	532	0%
			6	Red	95	520	390	597	80.4%
C (Figure 3.c)	Gradually	Suddenly	7	Green	55	325	450	599	42.28%
			8	Blue	35	367	367	599	55.52%
			9	Red	45	472	377	599	39.32%
D (Figure 3.d)	Abruptly	Moderately	10	Green	0	165	357	596	79.32%
			11	Blue	2	257	67	597	33.52%
			12	Red	0	192	35	596	13.32%
No Churn	Fixed	Fixed	13	-----	0	0	600	600	100%

Table 4. Characteristics of the Real Disseminated Video Stream in the Simulation

Video Stream	Frame/Second	Layer	GoP Structure	10 Minutes of Video	Quantizer Parameter	Mean Frame Size	Mean Frame PSNR	Mean Frame Bit Rate
Lord of the Ring II	25	Single	G12B2	18.84 MB	10	1389.81 Bytes	36.76 dB	277962.03 bps

In order to make the simulation more similar to real networks, this study considered variable amount of cross traffic in each fraction of peers between a minimum and a maximum rate randomly. These rates changes every 5 seconds. Moreover, each new joined peer selects its neighbors based on the received list of active peers in the network from a tracker (figure 4). Peers are arranged in the network area using the random distribution model. In addition, it is possible to select each peer from each considered fraction in table 2 randomly. Therefore, the network structure is similar to real P2P networks. Flowchart 1 depicts the video decoding process in this study based on the figure 1 and [42, 44].

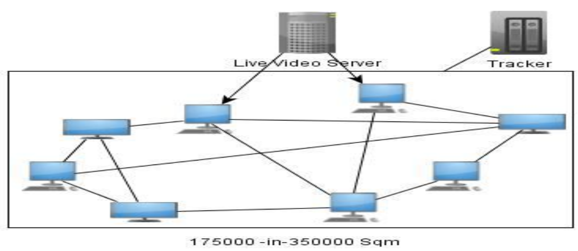
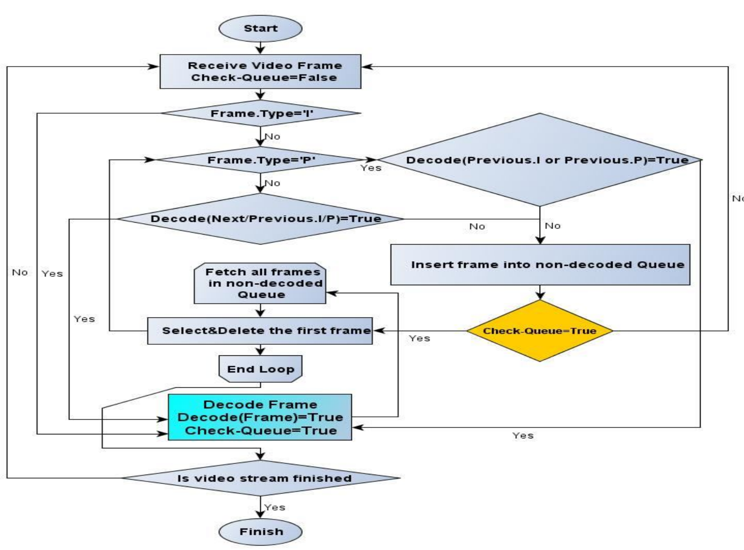


Figure 4. A Simple Used P2P Mesh in the Simulator



Flowchart 1. Decoding Steps of Video Frames in a Peer in the Simulator

In the decoder, a frame P can be decoded if the previous frame P or I is received and decoded completely before. In addition, a frame B can be decoded if both the next and the previous frames P and I are received and decoded successfully. In the flowchart 1, after decoding each frame, all existing frames in the non-decoded queue which their playback time are not passed will be examined to check if there is any frame which can be decoded after decoding the latest decoded frame.

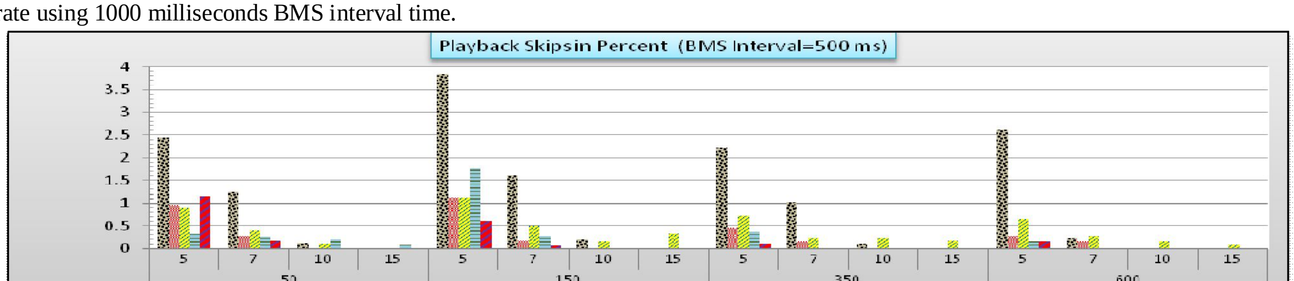
5.2. *Results discussion*

According to tables 1 to 4, different parameters and conditions simulated based on five important performance metrics including Playback skips, Dissemination time, Network overhead, Initial startup delay and End-to-End delay. In the following sections, this research will discuss on the obtained results for each of them in figures 5 to 30 in more details. In figures 5 to 19, the first and the second row in the horizontal axis shows different initial buffer times and network sizes, respectively.

5.2.1. Playback Skip

The number of video frames that a peer cannot play in its playback time refers to the playback skip event. In other words, this metric shows the ratio of the number of skipped frames to the total number of successfully received video frames in a peer. Here, this metric is shown in percent. For example, if a peer receives 4000 video frames out of 10000 disseminated frames by the video server and 20 frames are skipped, then, the playback skip in this peer equals to ((20/4000)*100)= 0.5 percent. Three main reasons exist for playback skip event (video distortion). First, a video frame may arrive after its playback time in a receiver (late arrival). Second, error probability is considered on links. Therefore, a video packet can be affected by error. Finally, existing dependencies among video frames in a GoP can lead to playback skip, even if a frame is received correctly and on time. Figures 5, 6 and 7 depict the obtained results for playback skip events in percent for different number of peers and initial buffer times, when the BMS interval times are 500, 1000 and 1500 milliseconds, respectively.

What can be inferred from these figures is that the BMS interval times equal to 500 and 1000 milliseconds provide small number of playback skips, while it is very high using 1500 milliseconds, especially in scenarios B and C. However, according to figures 11 and 12, the imposed network overhead using 1000 milliseconds is lower than that of 500 milliseconds. In other words, using 500 milliseconds BMS interval time makes peers greedier to request required video frames from that neighbor who announced about the existence of that frame before others, especially when no churn is happened in the system. As depicted in figure 5, plan 13 (no churn) causes the highest playback skip events. This event results in high amount of traffic on this peer and makes it as a bottleneck in the system, especially if it has low upload bandwidth. In addition, the cooperation among peers will be reduced, because, although other peers have this video frame in their buffers, neighbors ask the first peer who announced the existence of it to transfer that frame to them. On the other hand, the BMS interval time equal to 1500 milliseconds forces peers to request missed video frames in their buffers in a longer period of time. Therefore, the probability of missing a video frame in the neighbors' buffers increases. Hence, peers cannot have required frames in proper time, because a peer may receive a video frame after its playback time due to long waiting time for the next BMS interval time event. This leads to many late arrival frames in a peer and finally a large number of playback skips. Consequently, playback skip event occurs many times in that peer. In fact, the system does not work well in both cases. These events happen in lower rate using 1000 milliseconds BMS interval time.

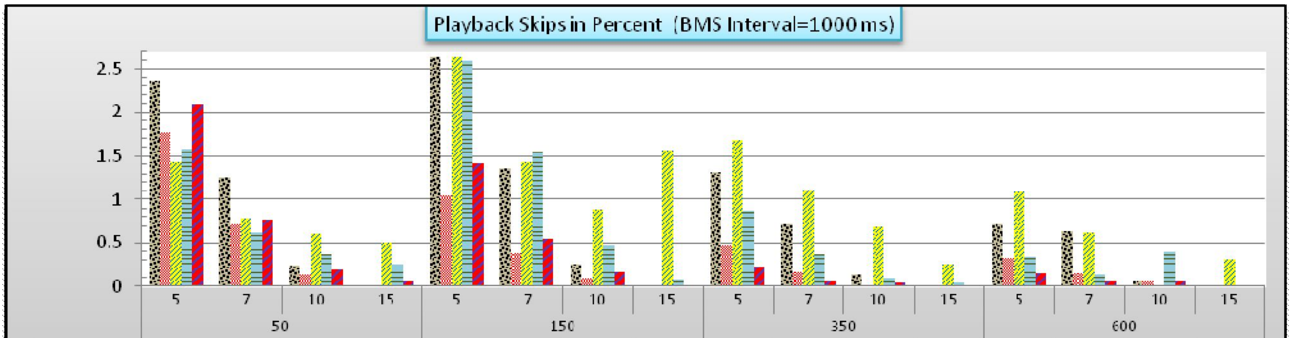


	50				150				350				600			
	5	7	10	15	5	7	10	15	5	7	10	15	5	7	10	15
No Churn	2.44	1.24	0.12	0	3.85	1.62	0.21	0	2.23	1.03	0.11	0	2.63	0.25	0.05	0
Scenario-A	0.95667	0.28333	0.05	0.00667	1.12667	0.17667	0.01	0	0.47	0.17	0.04	0	0.28333	0.16333	0.02667	0
Scenario-B	0.89333	0.40667	0.10333	0	1.13333	0.50667	0.16333	0.34667	0.73667	0.24	0.24	0.19333	0.65333	0.28	0.16333	0.08667
Scenario-C	0.34	0.26333	0.2	0.09	1.77333	0.29333	0.03667	0.02667	0.37667	0.03	0	0	0.16333	0.01333	0.00333	0
Scenario-D	1.15667	0.18	0.04	0.00667	0.62	0.07333	0	0	0.11333	0.01333	0	0	0.16667	0.03333	0	0

Figure 5. Introduced Averaged Playback Skips when BMS Interval equals to 500 ms

In other words, 500 and 1000 milliseconds interval times let all scenarios impose low playback skip by faster exchanging the BMSs among peers and inform their neighbors about the latest status of their buffers. In fact, when the BMS interval time is suitable, except scenario B, the averaged experienced playback skip by peers in all scenarios follows same behaviors so that it slightly fluctuates based on the different numbers of peers and the initial buffer times. Contrary to other scenarios, the observable overlap time between peer joining and leaving is considerable in scenario B. Therefore, the behavior of the system is more dynamic. This clearly shows that the positive effect of considering suitable BMS interval time on the perceived video quality is more than trying to keep the system stable using the plan 13. Hence, although plan 13 causes the highest skip in figure 5, the introduced playback skip by scenario B is considerable in figures 6 and 7.

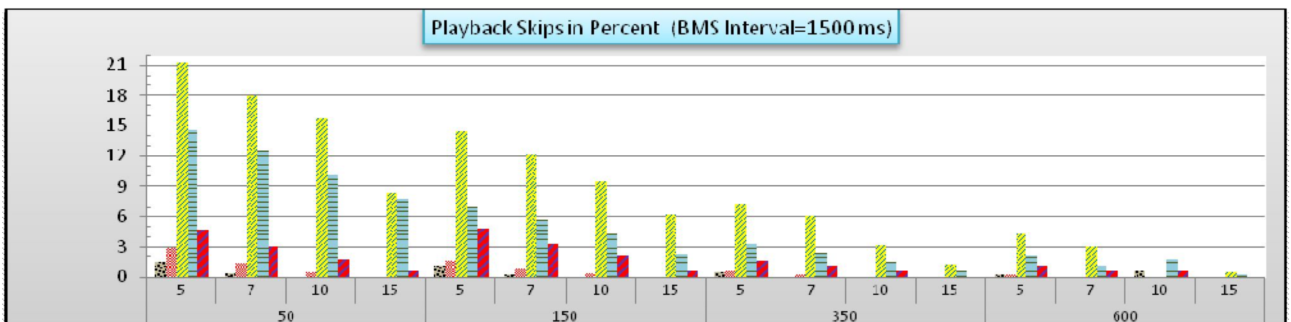
All in all, all scenarios introduce higher playback skips when the BMS interval time is 1500 milliseconds. Another interesting result is that considering the initial buffer time more than 10 seconds has not considerable effects on the experienced playback skip. On the other hand, using 5 and 7 seconds initial buffer times increases the number of playback skips in the system due to this fact that peers have not enough time to buffer required video frames before starting the playback process.



	50				150				350				600			
	5	7	10	15	5	7	10	15	5	7	10	15	5	7	10	15
No Churn	2.36	1.25	0.24	0.01	2.63	1.36	0.26	0.01	1.32	0.72	0.14	0	0.72	0.64	0.06	0
Scenario-A	1.77667	0.72	0.13	0.01	1.05667	0.38667	0.09333	0.01	0.43667	0.15667	0.01333	0	0.31667	0.15	0.05333	0
Scenario-B	1.44	0.79333	0.59667	0.51	2.65	1.43667	0.9	1.56667	1.69333	1.11	0.69333	0.26	1.09	0.63333	0.02	0.30333
Scenario-C	1.58	0.62667	0.37	0.25	2.6	1.55333	0.47	0.08333	0.87333	0.38	0.09333	0.04667	0.34333	0.13	0.40667	0.01333
Scenario-D	2.09	0.77667	0.18333	0.05	1.43	0.54667	0.15667	0.01333	0.23333	0.07	0.04333	0.03333	0.15333	0.06667	0.06	0.01667

Figure 6. Introduced Averaged Playback Skips when BMS Interval equals to 1000 ms

However, peer churning makes the behavior of the system unpredictable somewhat, especially in large network sizes. For example, the averaged introduced skip event to 600 peers in 10 seconds initial buffering is more than 7 seconds for scenario C in figure 6. As a final result of this section and according to the obtained results in this study, pull approach tries to keep the system stable in different scenarios by introducing the least possible amount of playback skip events, especially when peers join or leave the system abruptly. Moreover, the effect of using large initial buffer time for decreasing the number of playback skip events is more than few numbers of peers in the network. Figures 5 to 7 clearly show this assertion where the introduced playback skip to the system decreases sharply and moderately while the initial buffer time and network size increase, respectively.



	50				150				350				600			
	5	7	10	15	5	7	10	15	5	7	10	15	5	7	10	15
No Churn	1.56	0.39	0.03	0.04	1.21	0.36	0.01	0.01	0.55	0.1	0.01	0.05	0.36	0.12	0.67	0.03
Scenario-A	2.95	1.46667	0.54333	0.17667	1.67	0.88333	0.43	0.11333	0.67	0.29333	0.12667	0.01	0.35333	0.20667	0.18667	0.03667
Scenario-B	21.2767	17.98	15.6933	8.44667	14.46	12.2067	9.60333	6.21333	7.26333	6.10333	3.17333	1.32667	4.37	3.11	0.07333	0.59667
Scenario C	14.64	12.5733	10.1167	7.76667	7.02	5.80333	4.3	2.42667	3.45667	2.54667	1.58667	0.66333	2.18667	1.29333	1.79	0.27667
Scenario-D	4.72	3.01333	1.82333	0.7	4.84667	3.44	2.17	0.61333	1.74333	1.28333	0.65333	0.22667	1.21333	0.56333	0.68333	0.14333

Figure 7. Introduced Averaged Playback Skips when BMS Interval equals to 1500 ms

5.2.2. Dissemination Time

The time between sending a video frame in the video source and playing it in a peer refers to the dissemination time. In figures 8 to 10, averaged values of the introduced dissemination time to the system for different numbers of peers in various initial buffer times and network sizes are presented. According to the obtained results, all scenarios show same behaviors such that the introduced averaged dissemination time to the existing peers in the system heightens while the number of peers and the initial buffer time increases. However, contrary to the network size which increases this metric sharply, adding initial buffer time in peers leads to moderately growth in the experienced dissemination time among peers, because peers can exchange their BMSs while they are in initial buffering stages. On the other hand, large network size causes higher end-to-end delay as discussed in section b.4. The higher end-to-end delay is, the higher dissemination time introduces. Moreover, the introduced dissemination time to the systems increases when the BMS interval time changes from 500 to 1000 and 1500 milliseconds. In fact, the effect of increasing in the BMS interval time on the introduced dissemination time is more visible when the network size increases.

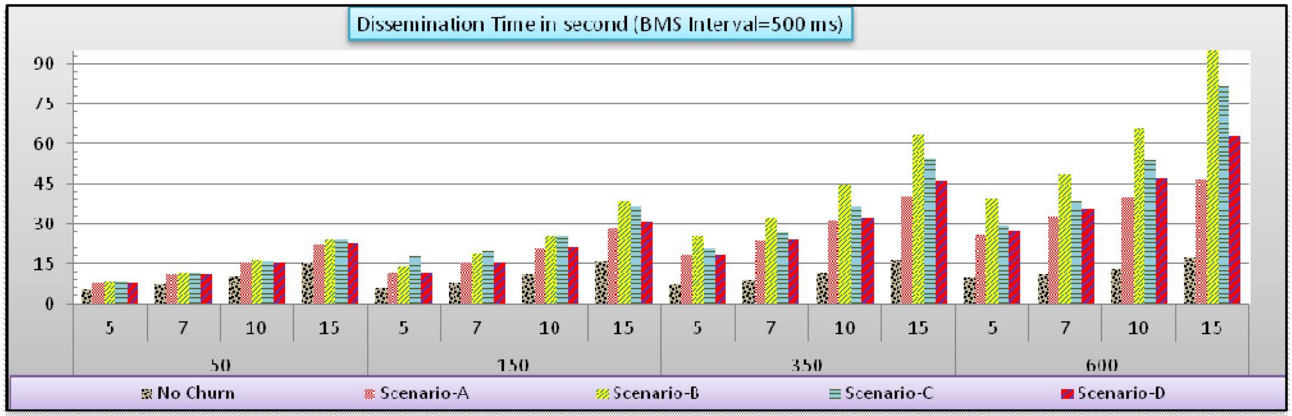


Figure 8. Introduced Averaged Dissemination Time to the System when BMS Interval equals to 500 ms

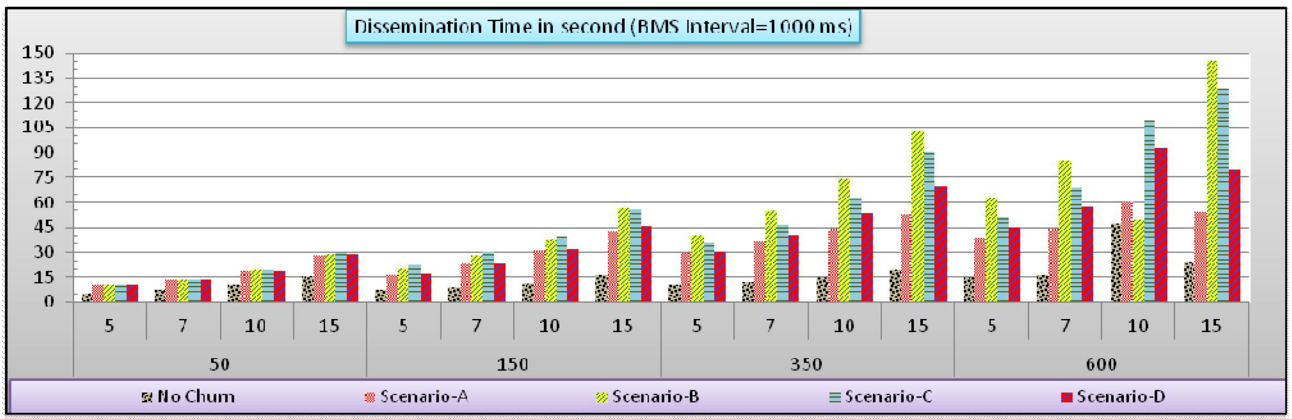


Figure 9. Introduced Averaged Dissemination Time to the System when BMS Interval equals to 1000 ms

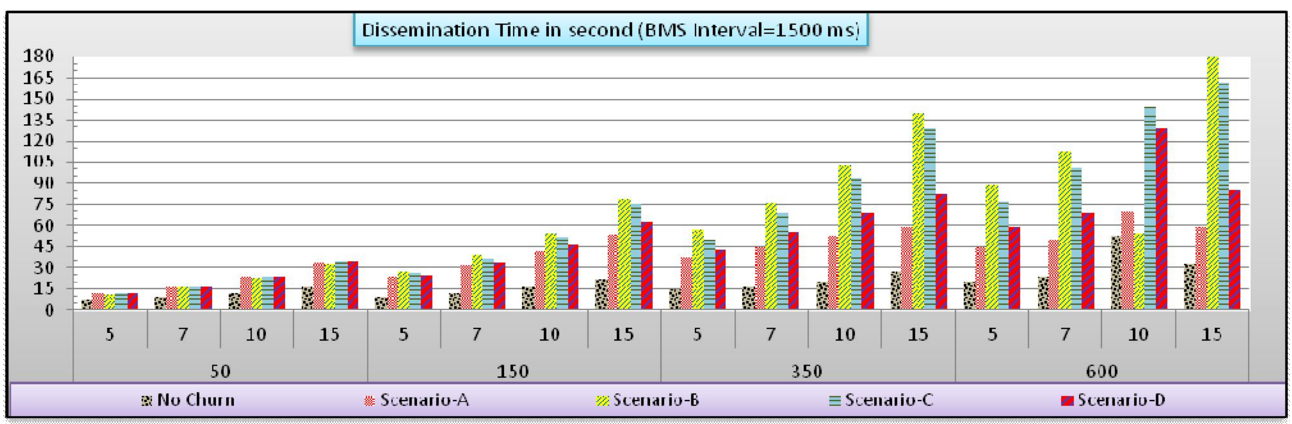


Figure 10. Introduced Averaged Dissemination Time to the System when BMS Interval equals to 1500 ms

In these figures, scenario B causes the highest averaged dissemination time where peers join and leave the network gradually. In fact, in scenario B, the dynamic nature of the system is more than others. His behavior is same in scenario C. Interestingly, the last plan (no churn) introduced the least averaged dissemination time to the system. This clearly shows that the effects of churning are considerable on the system in terms of the required time for delivering video frames to peers, even if it employs a flexible method such as pull approach for data exchanging. Another interesting result is that having high cooperation among peers is more important than using large initial buffer time in them. The plan 13 clearly shows this assertion where peers do not leave the network and have considerable cooperation with each others to disseminate the video frames. As can be seen in figures 8 to 10, contrary to all scenarios including peer churning, the introduced dissemination time did not considerably increase in plan 13 when the amount of initial buffer time increases, especially in large network sizes. Finally, according to the obtained results, large initial buffer time (e.g. 15 seconds) has noticeable positive effect on the introduced dissemination time for those plans which allows peers to have more cooperation with each other (e.g. plan 13, scenarios A and C), when the network size increases (e.g. 600 peers). Figures 9 and 10 clearly explain this reality. This effect is vice versa for scenarios B and C where peers have low cooperation due to this fact that they join the network gradually which results in smaller number of neighbors for each peer.

5.2.3. Network Overhead

This metric refers to the ratio of the useless to the total amount of exchanged traffic in the network in percent. Useless traffic includes buffer map status messages, received duplicated or damaged video frames due to error and control messages such as peer joining and leaving as well as initial peer setup using the tracker. This study determined to measure this metric in order to better understand the efficiencies of pull-based approach in different scenarios in terms of the imposed network overhead to the system. Actually, although a method can show high performance in data dissemination, it may provide it in return for high network overhead. Here, the behaviors of different scenarios are same in figures 11 to 13, where the BMS interval times are 500, 1000 and 1500 milliseconds, respectively. Scenario B introduces the least network overhead to the system, while those scenarios such as A, D and plan 13 (no churn) which let peers join the system intensively imposed high network overhead. The main reason is that a peer should exchange much more traffics with more number of neighbors. Although peers remained in the system for a longer period of time in scenario B in comparison with other churning scenarios, they have not enough opportunities to have enough number of neighbors due to this fact that peers join the network gradually. All in all, there is no big difference among imposed overhead to the system by different scenarios.

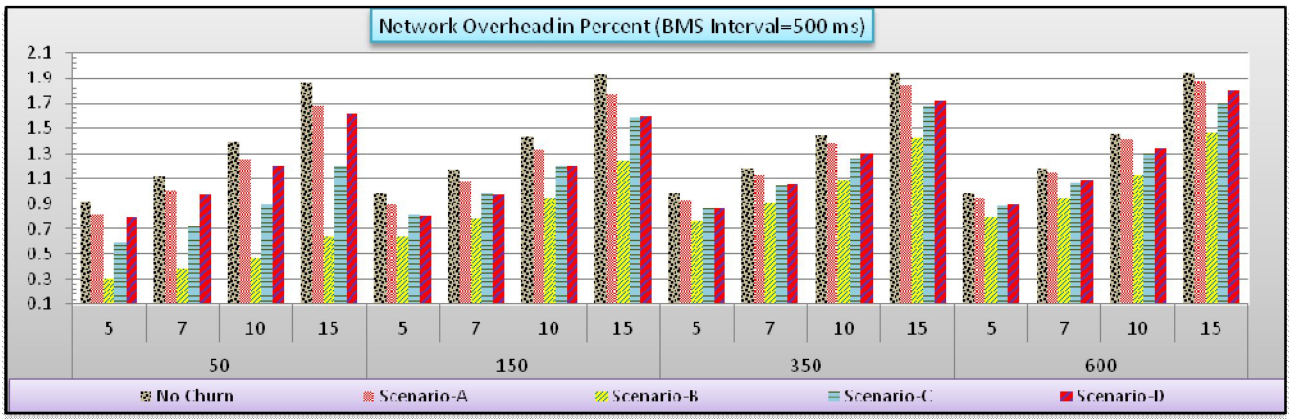


Figure 11. Imposed Network Overhead to the System when BMS Interval equals to 500 ms

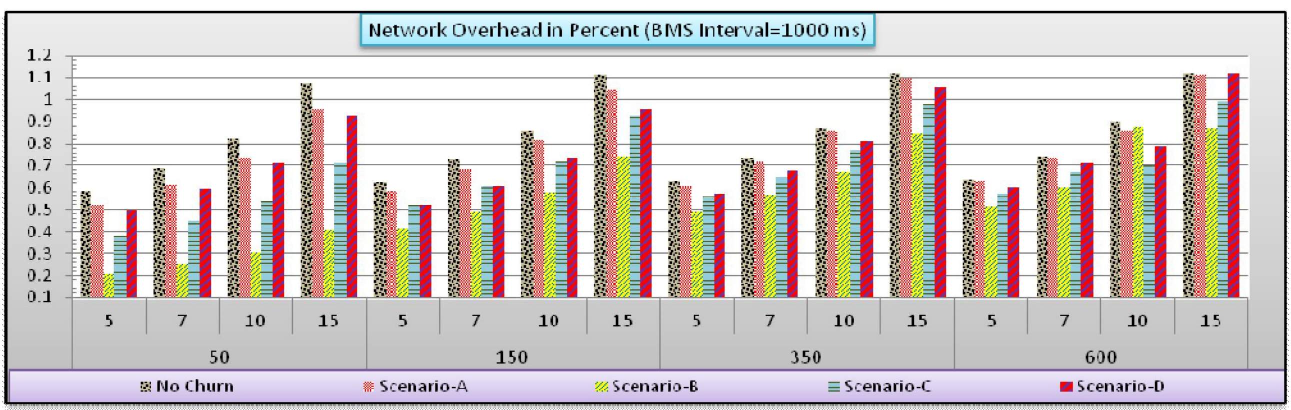


Figure 12. Imposed Network Overhead to the System when BMS Interval equals to 1000 ms

In figure 11, contrary to two other figures, the imposed network overhead to the system is higher, because peers have to exchange their BMSs in shorter period of times. Obviously, except scenario B which causes high dynamic network, the averaged imposed network overhead slightly heightens when network size increases. This clearly shows that the amount of useless traffic does not grow sharply in comparison with the total exchanged traffic in the network while the number of active peers increases. The interesting result is that the averaged amount of this metric heightens when the initial buffer times increases, because peers have more frames in their buffer to exchange which results in more number of frame requests and duplicated video frames. One of the main reasons is increasing in the number of duplicated video frames as depicted in figures 19 to 25 and 28 to 30.

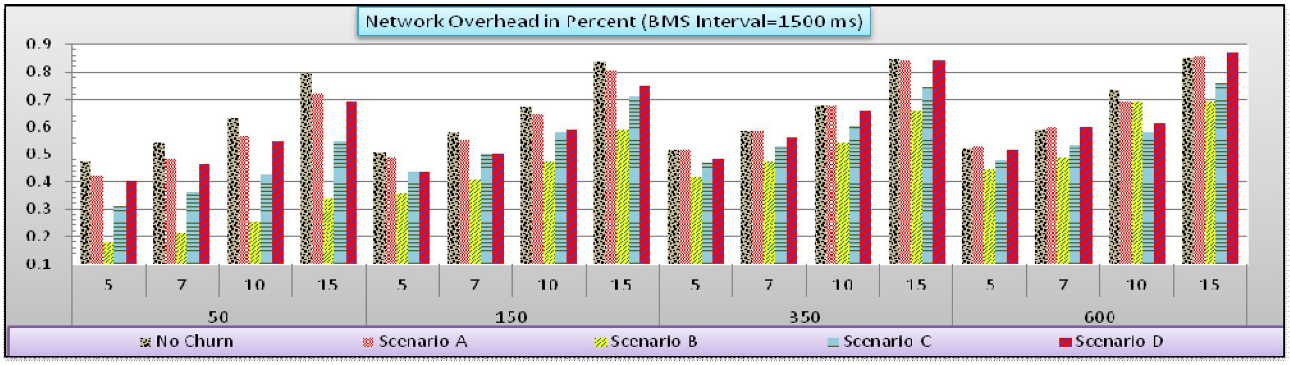


Figure 13. Imposed Network Overhead to the System when BMS Interval equals to 1500 ms

It is necessary to mention that using push-based approach over mesh-based networks introduces much more network overhead to the system, because it blindly transmits just received video frames to others. On the other hand, pull-based approach provides higher performance in terms of the imposed network overhead. This is why it is more suitable for live video streaming over mesh-based networks, especially in dynamic systems. Some recent studies such as [45] used network coding method [35, 46-48] to decrease the side effects of using push method over mesh-based networks.

5.2.4. End-to-End Delay

The time between sending a video frame in the video source and receiving it in a peer refers to the end-to-end delay. Actually, dissemination time includes end-to-end delay. However, this research measured both of them to show the effects of different numbers of peers, various initial buffer and BMS interval times on the elapsed time between receiving and playing a video frame using pull-based method.

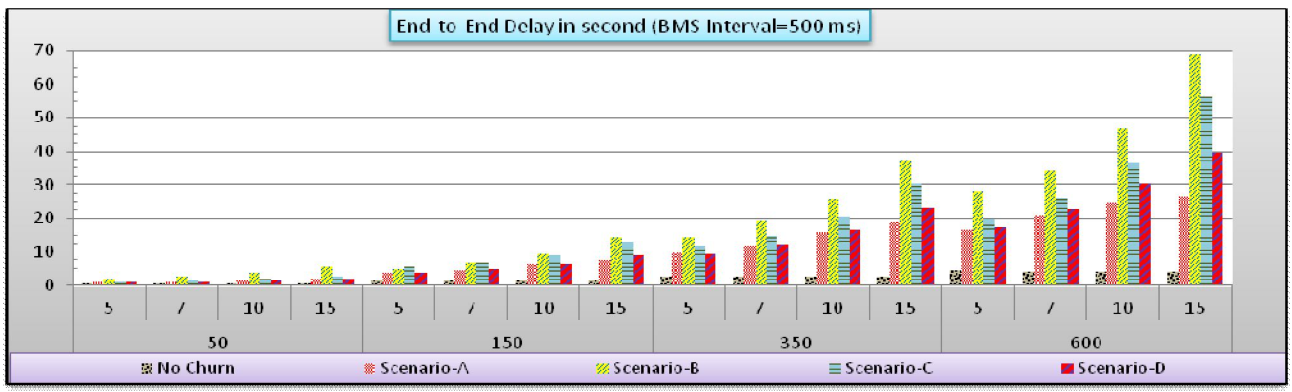


Figure 14. Introduced End-to-End Delay to the System when BMS Interval equals to 500 ms

What can be inferred from figures 14 to 16 is that the introduced end-to-end delay sharply increases when the number of peers changes from 100 to 600, especially in those scenarios which peers join the network gradually. In fact, in P2P systems, small number of peers causes less cooperation among them. The last plan (no churn), introduced the least end-to-end delay, because each peer has enough number of neighbors to get required video frames from them. In contrast, peers have fewer opportunities to have enough number of neighbors in scenarios B and C, where peers join the system gradually. On the other hand, scenarios A and D, which let peer join the system abruptly, complete the mesh network in a shorter time by introducing more number of peers with free links to the existing peers in the system as their neighbors.

Like the network size, increasing in the initial buffer time moderately increases the introduced end-to-end delay to the system, because peers are eager to receive required video frames in this stage. In fact, the initial startup delays of peers increases which lead to higher end-to-end delay in the system. In other words, each peer waits for receiving the requested frames without sending any additional request as happen in safe point region in playback time. Safe point region is a critical area such that all frames within it are close to their playback time and peers will send additional requests to receive frames in this region as soon as possible. This decreases end-to-end delay. On the other hand, no peer consider safe point region while it is in the initial buffer stage. Therefore, they will not greedy in receiving video frames which results in higher end-to-end delay. To consider safe point region in initial buffer stage, based on our experiments, makes the system inefficient, because many peers start this stage simultaneously. Next section describes Initial Startup Delay (ISD) in more details. As a final examination on the end-to-end delay metric, plan 13, scenario A, scenario D, scenario C and scenario B averagely introduced the lowest to the highest amount of delays to the system, respectively.

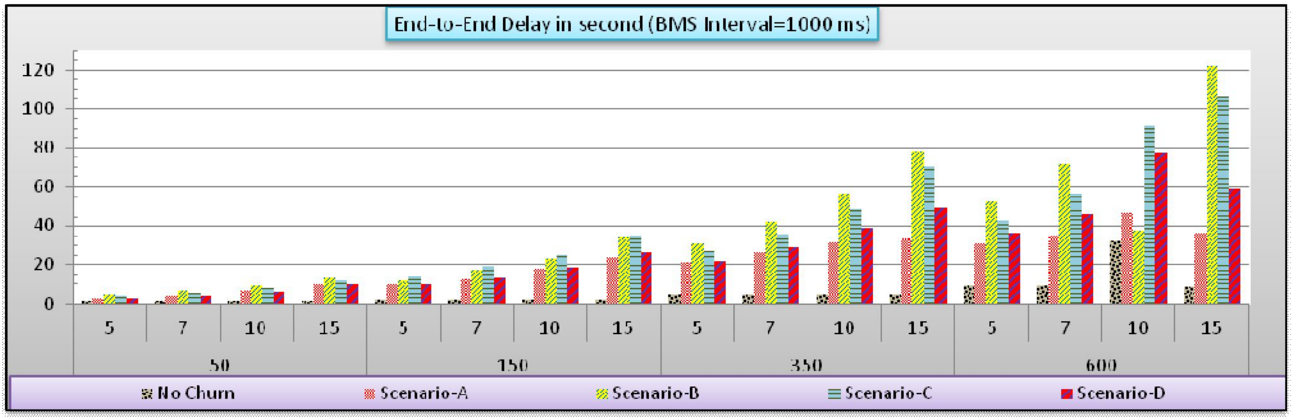


Figure 15. Introduced End-to-End Delay to the System when BMS Interval equals to 1000 ms

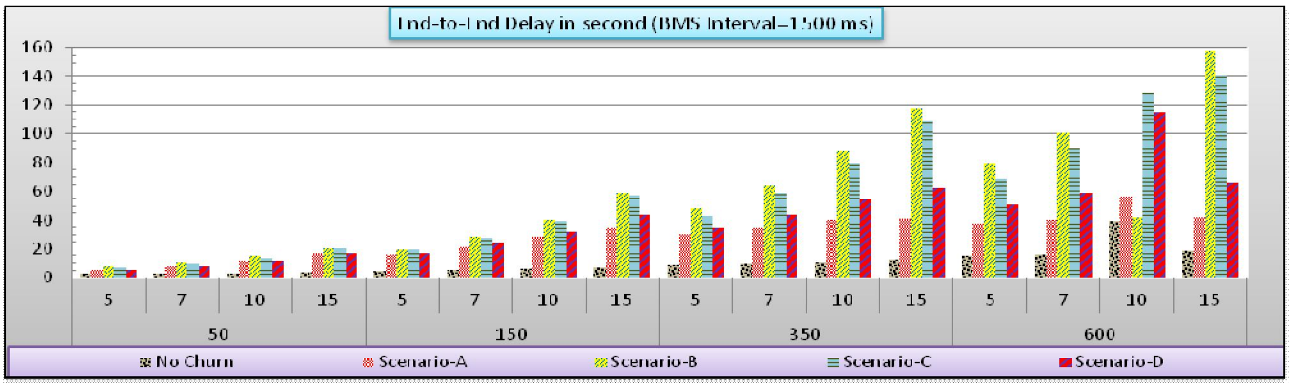


Figure 16. Introduced End-to-End Delay to the System when BMS Interval equals to 1500 ms

5.2.5. *Initial Startup Delay (ISD)*

Initial Startup Delay (ISD) is the elapsed time in a peer between receiving the first video frame and finishing initial buffering duration. In other words, this metric shows the averaged required time in peers for receiving requires video frames for the considered initial buffer time in table 1 before starting the playback process. As depicted in figures 17 to 19, the plan 13 (no churn) introduces the highest ISD to the system for different amount of BMS interval times, because the number of peers who exist in the network and are looking for required video frames is the maximum. This shows that high density of peers in a network has noticeable effects on ISD. On the other hand, other scenarios (churning) imposed lesser ISD to the system. The main reason is that peer crowd in the system is not high where they join and leave the network during the video dissemination time. In fact, the introduced ISD to the system is directly related to the number of active peers in the system not the behavior of the system for peer joining and leaving. Moreover, obtained results depict that increasing in the initial buffer time sharply increases the experienced ISD in the system and this is not directly related to the used scenario for peer churning, because peers have to wait much more time to finish their initial buffer stage. As mentioned before, increasing in the experienced amounts of ISDs affects the introduced end-to-end delays. In the next section, some additional results will be presented to complete our evaluation on pull-based method in P2P live video streaming over mesh-based networks.

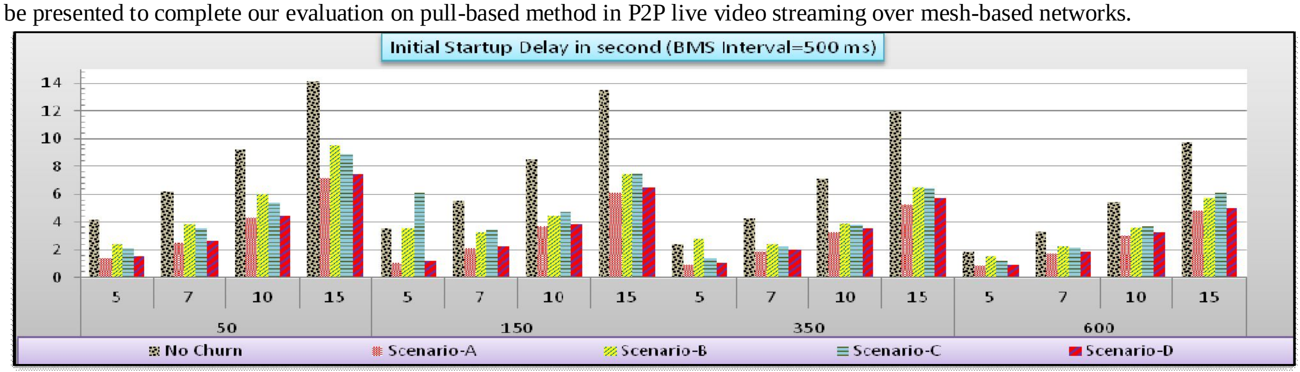


Figure 17. Introduced Initial Startup Delay to the System when BMS Interval equals to 500 ms

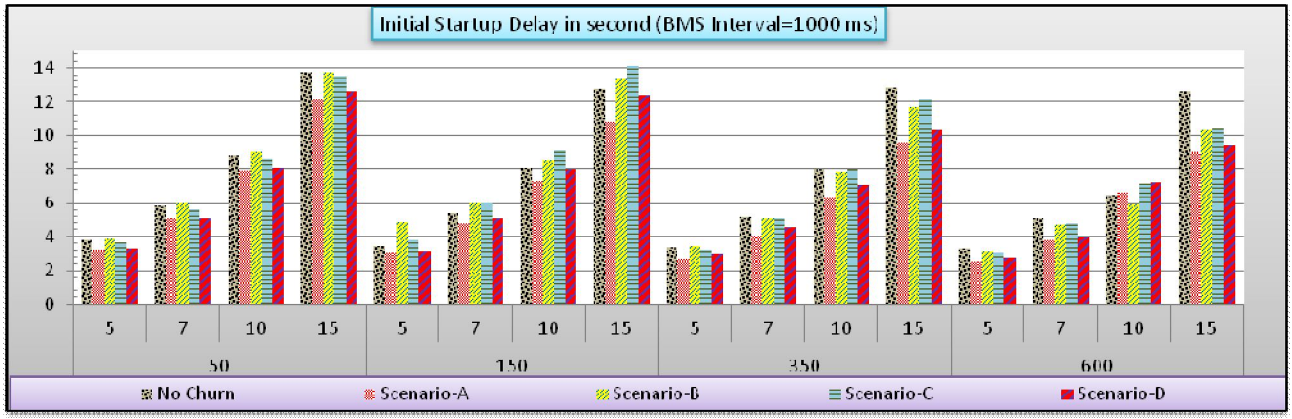


Figure 18. Introduced Initial Startup Delay to the System when BMS Interval equals to 1000 ms

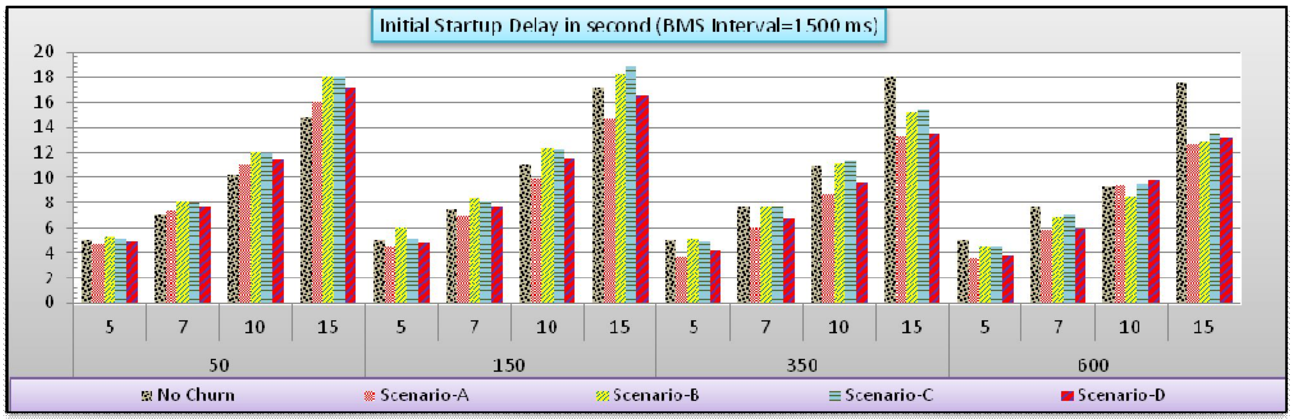


Figure 19. Introduced Initial Startup Delay to the System when BMS Interval equals to 1500 ms

5.2.6. Further Discussions

One of the most important metrics for evaluating the performance of an exchange method is the number of introduced duplicated packets to the system by it. This metrics shows how the employed method affects the experienced network throughput in the system. Therefore, this study measured and plotted this metrics for each scenario separately as depicted in figures 20 to 25. Contrary to figures 20 to 24, figure 25 compares considered scenarios based on the different amounts of BMS interval times.

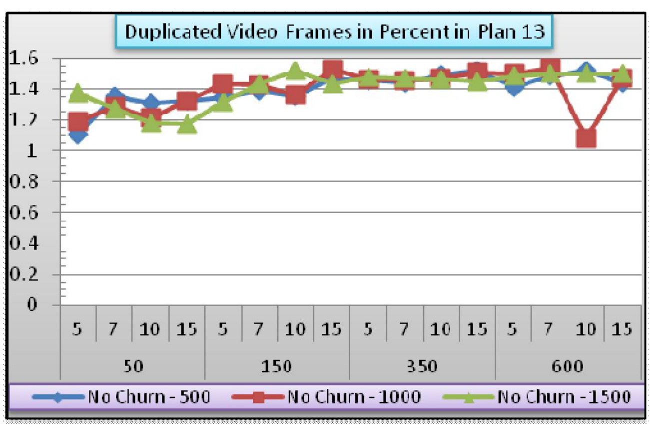


Figure 20. Plan 13 (No Churn)

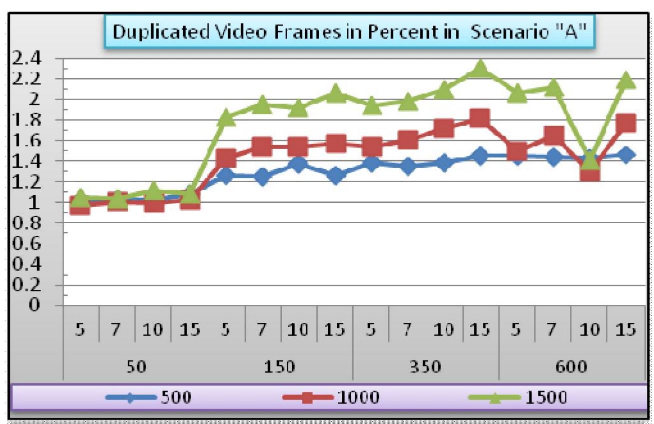


Figure 21. Scenario A

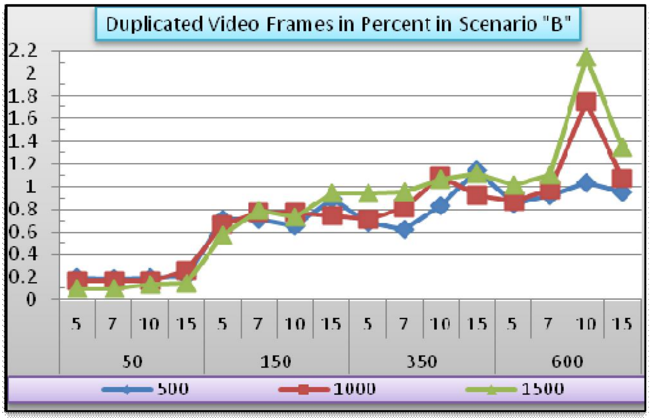


Figure 22. Scenario B

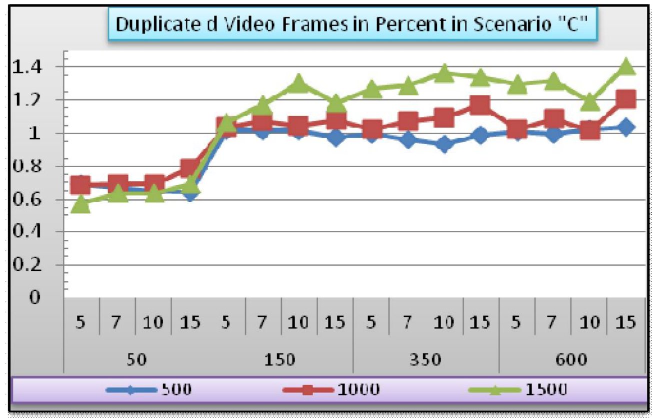


Figure 23. Scenario C

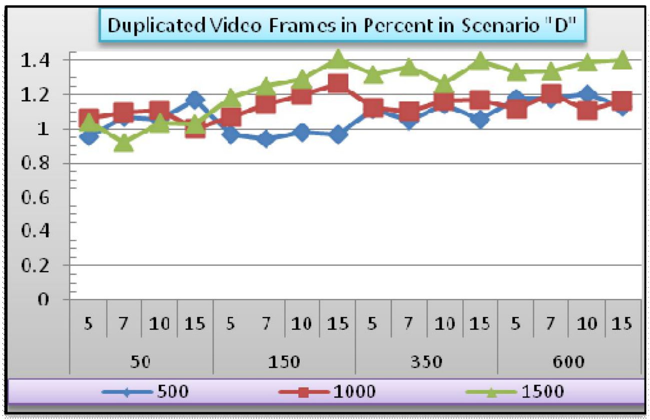


Figure 24. Scenario D

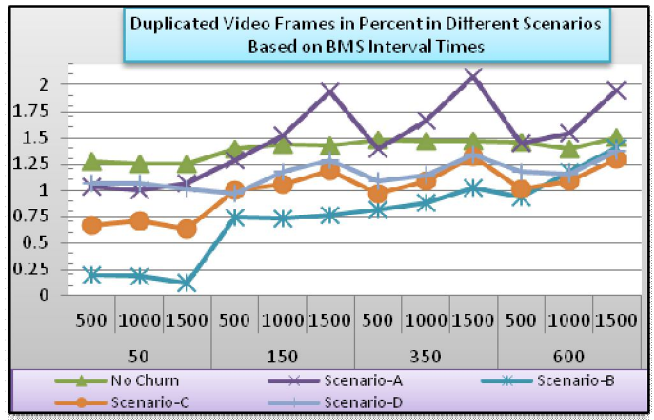


Figure 25. Comparison of Different Scenarios

As can be seen in figure 20, different values of BMS interval times have not considerable effects on the introduced duplicated video frames to the system in plan 13. This is approximately same for scenario B. However, different amounts of BMS interval times caused greater differences among the introduced number of duplicated video frames in scenarios A and D where peers join the network abruptly is a short period of time, especially for large network sizes. As mentioned before, safe point area is a critical region such that all frames within it are close to their playback time. Obviously, in all scenarios, the number of duplicated video frames increases when the network size growths, because more number of peers exist in the network. However, the effects of different amounts of initial buffer times did not follow exactly same behaviors in different scenarios due to unpredictable conditions of the system when peers join and leave the network continually.

As mentioned before and according to the figure 25, using 500 milliseconds BMS exchange interval time makes peer greedier to get required video frames. To clarify this, suppose that there are three peers in the network. Peer 1 has frame B1 from GoP 88 and peer 2 sends a request to peer 1 for getting it. When peer 1 receives this request, it puts it in a FIFO queue for further processing in future. In the meantime, peer 2 comes near the playback time of frame B1 and enters the safe point region as depicted in figure 26. In our study, the length of this region is one second which means this region includes 25 frames. Peer 2 knows that peer 3 has frame B1 and it has not yet received it from peer 1. Therefore, it requests this frame from peer 3. In this time, peer 1 sends frame B1 to peer 2. In this sense, peer 2 will receive two copies of this frame which leads to duplicated video frame. Contrary to 500 ms, in 1000 and 1500 milliseconds cases, peers send their BMS in longer interval time. As a result and according to figure 25, using 1000 milliseconds BMS interval time introduces same number of duplicated video frames as 500 milliseconds in different scenarios. However, using 1500 milliseconds interval time causes larger number of duplicated frames in large network sizes, because the probability of having frames in the safe point region will be increased and this makes peers greedier like 500 milliseconds scenario. Then, the peer will send another request for this frame immediately which may lead to more number of duplicated video frames. Figure 26 shows the considered buffer structure in this study for as peer 2 after receiving frame B1in GoP 88. This structure is used for all peers. Here, the media player in this peer is playing frame B1 (the second bit in the safe point region) in GoP 88. As depicted, frames B2, B4, B5, B6 and B8 from this GoP are not exist in the buffer. Therefore, peer should send another request for getting them from its neighbors, because all of them are in the safe point region close to their playback times. It is necessary to mention that, 0 and 1 shows the absence and the existence of a frame in buffer, respectively.

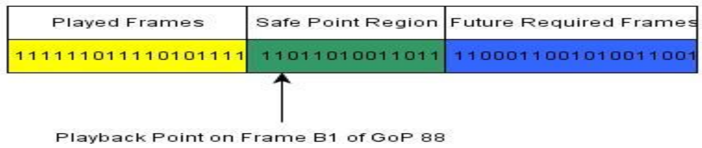
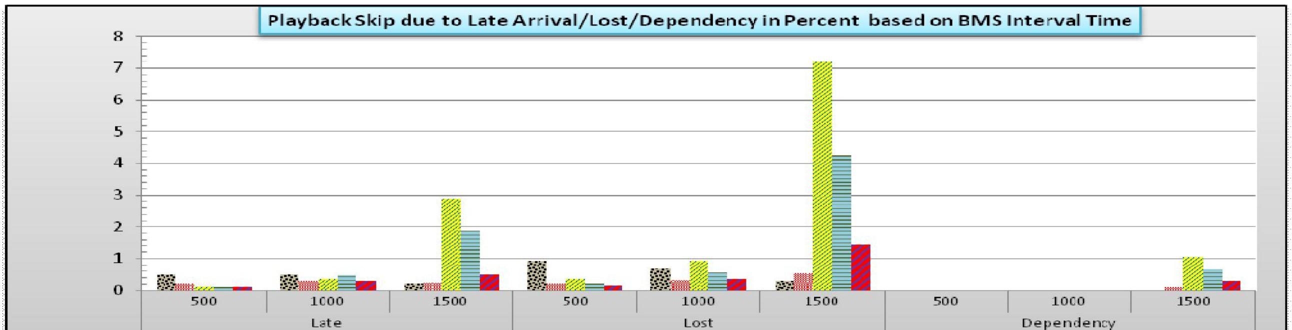


Figure 26. Buffer Map Structure in Peer 2

Moreover, in our study, each peer indicates all existing video frames in its buffer in the BMS messages while it is in the initial buffering stage. On the other hand, they announce the latest statuses of their buffers in the BMS messages from the playback point up to the T_{BMS} seconds after the playback point such that T_{BMS} is considered equal to initial buffer time in the simulation. The simulator ran for several times and the obtained results are examined for finding the optimal value for T_{BMS}. As a final evaluation, this study analyzed the introduced playback skip to the system due to mentioned reasons in section b.1. Figure 27 shows that channel error (lost frames) is the most important reason of many skip events in P2P systems using pull-based method.



	Late 500	Late 1000	Late 1500	Lost 500	Lost 1000	Lost 1500	Dependency 500	Dependency 1000	Dependency 1500
No Churn	0.524	0.51	0.222	0.922	0.687	0.301	0.065	0.045	0.043
Scenario A	0.212	0.299	0.243	0.225	0.319	0.53	0.01	0.017	0.103
Scenario B	0.135	0.365	2.889	0.372	0.937	7.197	0.012	0.044	1.047
Scenario C	0.116	0.469	1.887	0.222	0.584	4.255	0.004	0.03	0.648
Scenario D	0.132	0.295	0.498	0.143	0.35	1.449	0.007	0.021	0.297

Figure 27. The Effects of Three Existing Reasons of the Playback Skip on it

Figure 27 is plotted based on the experienced number of playback skips in different BMS interval time. In other words, this figure indicates that how each of late arrival, lost and dependency factors causes skip event in different BMS interval times when the initial buffer time and the network size change. For example, 0.524% skip is happened averagely due to late arrival frames when the BMS interval time was 500 milliseconds, while network size and initial buffer time changed according to mentioned values in table 1. As can be seen in this figure, scenarios B and C causes the largest averaged number of playback skips in the system. The main reason is the high imposed playback skipped in figure 7 when the BMS interval time was 1500 milliseconds. In the performed simulation in this study, playback skip due to late arrival happens when a frame arrives late in a peer after its playback time. On the other hand, if a frame arrives on time before its playback time and the decoder cannot play it because of existing dependency in the GoP, a skip due to dependency is happened. For example, suppose that frame B2 in GoP 90 arrives on time while frame I from this GoP does not exist in the buffer. In this sense, a skip event due to dependency will occur. Finally, because of existence of error on channels, a frame can be lost on the channel. Moreover, it can be arrived while a part of its data is damaged due to error. In this case, a skip event due to lost is happened. Figure 27 shows both of these cases as lost. As a final result of this figure, comparing introduced skip event in different BMS interval times clearly shows that many skip events are happened using 1500 milliseconds in comparison with 500 and 1000 milliseconds BMS interval time, especially for dependency factor. As a result, according of all obtained results in this study, this research deducted that 1000 milliseconds BMS interval time is the best value for exchanging BMSs among peers.

Finally, according to this fact that playback skip is the most important performance metric for measuring the efficiency of an exchange method in video streaming applications and for better understanding the most important reasons of playback skip event, figures 28 to 30 shows the averaged values of this metric for different scenarios in percent in another point of view based on the different caused factors including *l*ate arrival frames (*L*), l*o*st frames (*O*) and *d*ependency among frames(*D*) for different numbers of peers and initial buffer times in 500, 1000 and 1500 milliseconds BMS interval time, respectively. Late arrival (L) and lost frames (O) cause large number of playback skips. Playback skip due to dependencies among video frames is just noticeable using 1500 milliseconds BMS interval time, because peers wait longer period of time for exchanging their BMSs among themselves.

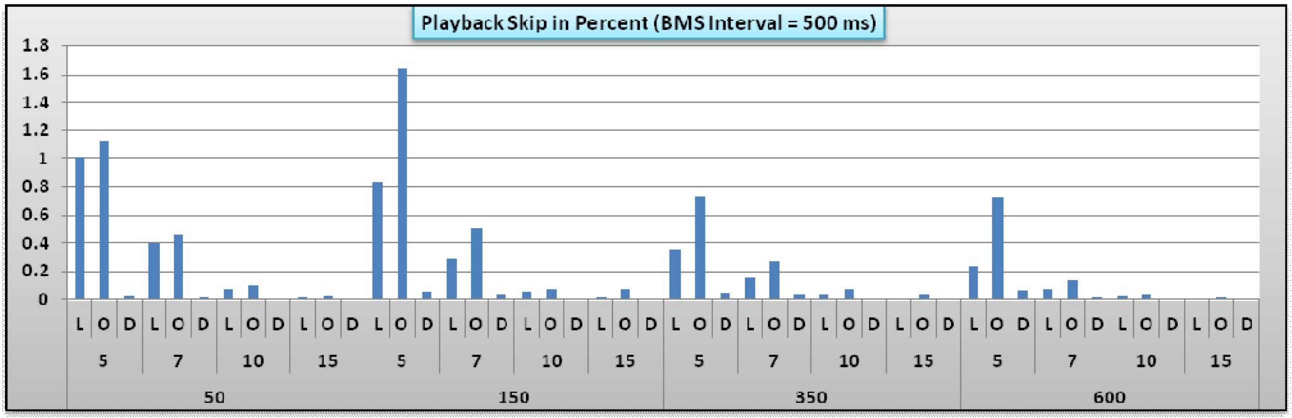


Figure 28. Averaged Experienced Playback Skip Due to Different Factors for 500 ms BMS Interval Time

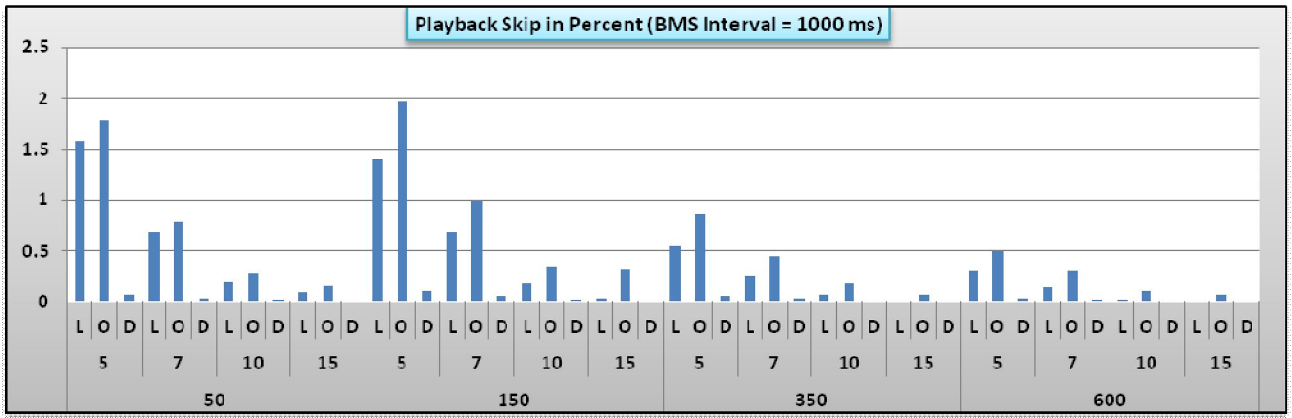


Figure 29. Averaged Experienced Playback Skip Due to Different Factors for 1000 ms BMS Interval Time

These figures clearly show the effects of different network sizes on playback skip event. In fact, the cooperation among peers is low due to less number of active neighbors when there are few numbers of peers in the system (e.g. 50 and 150). On the other hand, large network sizes (more than 250 based our additional experiments) increase the required cooperation among peers.

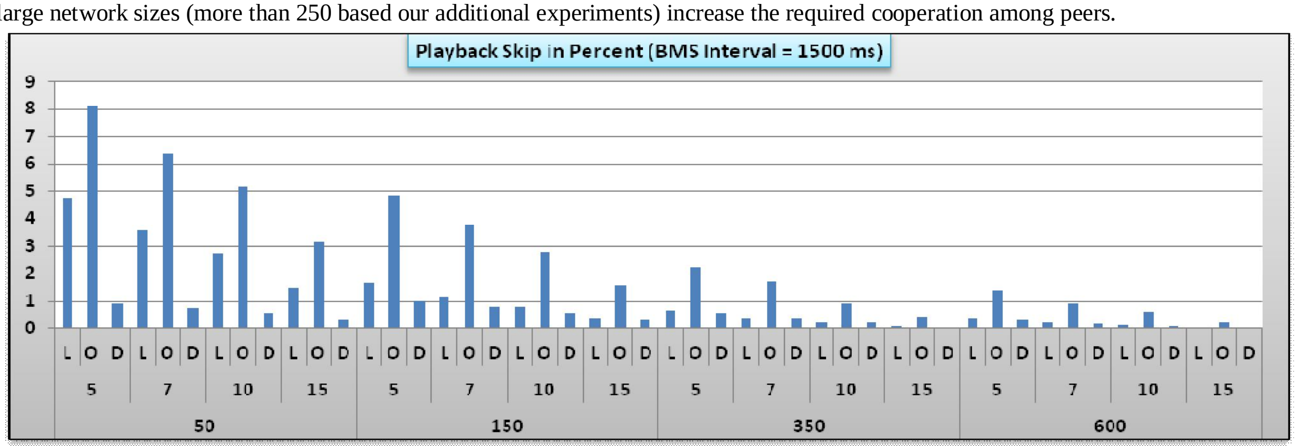


Figure 30. Averaged Experienced Playback Skip Due to Different Factors for 1500 ms BMS Interval Time

6. CONCLUSION

Live video streaming has been the most attractive traffics over the computer networks. P2P systems provide low cost flexible and reliable infrastructures for live video dissemination. Peers use two main exchange methods in these systems including pull- and push-based approaches. In the recent years, pull-based method is employed by mesh-based P2P systems. However, none of them examined its efficiencies in different network conditions and peer churning scenarios completely and precisely. Therefore, this study analyzed pull-based method under different and various network conditions such as high peer churning for better understanding its efficiency in P2P live video streaming. The obtained results based on five performance metrics including playback skip, end-to-end delay, dissemination time, initial startup delay and network overhead show that this method is really robustness in dynamic networks. Moreover, this research evaluated the introduced duplicate video packets using different parameters and scenarios. All in all, it is necessary to improve the efficiency of pull-based method more and more, especially in large network size and very dynamic networks. In other words, peers cannot enjoy the benefits of being a member of mesh network if the dynamic behavior of the system increases. Scenario B clearly confirms this assertion where there is a large overlap between peer joining and leaving which leads to high dynamic behavior in the system. Contrary to peer joining, peer leaving has more considerable effects on the performance of the network. Another obtained result is that 1000 milliseconds BMS exchange interval time provides the best efficiency according to the measured metrics in different network conditions and scenarios. Previous studies used this value; however, none of them clearly and precisely explained the main reason of using it. Here, this study made its reason clear by providing many results based on the considered performance metrics. Moreover, mesh networks' performances decrease while the network size is small or very large.

In near future, contrary to this study which single layer video is disseminated to the P2P system, we will evaluate Scalable Video Coding (SVC) as a multi-layer video stream in pull-based P2P live video streaming to understand the provided efficiency by pull method in this type of video encoding. SVC is more suitable for heterogeneous P2P systems. Moreover, we will evaluate the efficiency of pull-based method when there are at least two video servers which disseminate the live video stream in the network simultaneously. It is necessary to mention that using many video servers causes some new challenges to the system such as what is the best method for synchronization among them and how we can employ a fair load balancing approach on these servers to serve peers. Finally, we determined to evaluate all of these conditions over PlanetLab which is a real testbed for P2P streaming. All in all, the obtained results in this study help new interested researchers in data dissemination over P2P systems to completely understand the efficiency of pull-based method under different network conditions and churning scenarios, especially in live video streaming.

REFERENCES

[1] Zahariadis, T., G. Pau, and G. Camarilo, 2011, *Future media Internet.* Communications Magazine, IEEE. **49**(3): p. 110-111.

[2] Merani, M.L. and D. Saladino, 2010, *Live Video and IP-TV*, in *Handbook of Peer-to-Peer Networking*, X. Shen, et al., Editors. Springer US. p. 985-1024.

[3] Annapureddy, S., et al., *Is High-Quality VoD Feasible using P2P Swarming?*, in *in Proc. of the 16th international conference on world wide web (WWW)*. 2007. p. 903-912.

[4] YouTube. Available from: http://www.youtube.com.

[5] PPLive. Available from: http://www.pplive.com.

[6] Apostolopoulos, G., W.-t. Tan, and J. Wee, 2002 of Conference. *Video Streaming: Concepts, Algorithms, and Systems*: HP Laboratories Palo Alto.

[7] Chen, C., 2008, *Peer-to-Peer Live Video Streaming Services: Past, Current, and the Future.* Winter.

[8] Zhengye, L., et al., 2009, *LayerP2P: Using Layered Video Chunks in P2P Live Streaming.* Multimedia, IEEE Transactions on. **11**(7): p. 1340-1352.

[9] Montazeri, S., M. Fathy, and R. Berangi, 2008, *An adaptive fair-distributed scheduling algorithm to guarantee QoS for both VBR and CBR video traffics on IEEE 802.11e WLANs.* EURASIP J. Adv. Signal Process. **2008**: p. 1-12.

[10] Ramzan, N., H. Park, and E. Izquierdo, 2012, *Video streaming over P2P networks: Challenges and opportunities.* Signal Processing: Image Communication. **27**(5): p. 401-411.

[11] Richardson, I.E., ed.2010, *The H.264 advanced video compression standard.*

[12] Wiegand, T., et al., 2003, *Overview of the H.264/AVC video coding standard.* Circuits and Systems for Video Technology, IEEE Transactions on. **13**(7): p. 560-576.

[13] Tarkoma, S., 2010 of Conference. *Overlay Networks*: Taylor and Francis group.

[14] Brogle, M., et al. *A performance comparison of native IP Multicast and IP Multicast tunneled through a Peer-to-Peer overlay network.* in *Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09. International Conference on.*

[15] Buford, J.F., H. Yu, and E.K. Lua, 2009 of Conference. *P2P Networking and Applications*, ed. M.I.T. David Clark. United States of America: Morgan Kaufmann Publishers is an imprint of Elsevier. 438.

[16] Castro, M., et al., 2003, *SplitStream: high-bandwidth multicast in cooperative environments.* SIGOPS Oper. Syst. Rev. **37**(5): p. 298-313.

[17] Zhang, X. and H. Hassanein, 2012, *A survey of peer-to-peer live video streaming schemes – An algorithmic perspective.* Computer Networks, ELSEVIER. **56**(15): p. 3548-3579.

[18] Buford, J.F., H. Yu, and E.K. Lua, 2009, *Chapter 8 - Peer-to-Peer Content Delivery*, in *P2P Networking and Applications*. Morgan Kaufmann: Boston. p. 183-202.

[19] Liu, Y., Y. Guo, and C. Liang, 2008, *A survey on peer-to-peer video streaming systems.* Peer-to-Peer Networking and Applications. **1**(1): p. 18-28.

[20] Li, J., 2008, *On peer-to-peer (P2P) content delivery.* Peer-to-Peer Networking and Applications. **1**(1): p. 45-63.

[21] Meng, Z., et al., 2007, *Understanding the Power of Pull-Based Streaming Protocol: Can We Do Better?* Selected Areas in Communications, IEEE Journal on. **25**(9): p. 1678-1694.

[22] Felber, P., et al., 2012, *PULP: An adaptive gossip-based dissemination protocol for multi-source message streams.* Peer-to-Peer Networking and Applications. **5**(1): p. 74-91.

[23] Xiaojun, H., L. Yong, and W.R. Keith, December 2007, *Inferring Network-Wide Quality in P2P Live Streaming Systems.* IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. **25**(9): p. 1640-1654.

[24] Hamra, A., E. Biersack, and G. Urvoy-Keller, 2004, *A Pull-Based Approach for a VoD Service in P2P Networks*

High Speed Networks and Multimedia Communications, Z. Mammeri and P. Lorenz, Editors. Springer Berlin / Heidelberg. p. 995-1006.

[25] Liu, Z., et al., *Using layered video to provide incentives in P2P live streaming*, in *Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV.* 2007, ACM: Kyoto, Japan. p. 311-316.

[26] Qiu, D. and R. Srikant, 2004, *Modeling and performance analysis of BitTorrent-like peer-to-peer networks.* SIGCOMM Comput. Commun. Rev. **34**(4): p. 367-378.

[27] Huong Bui Thi, L. and N. Hoai Son. *A Low-delay Push-Pull Based Application Layer Multicast for P2P Live Video Streaming*. in *Knowledge and Systems Engineering (KSE), 2011 Third International Conference on.*

[28] Narkdej, K., S. Choochaisri, and C. Intanagonwiwat. *A Reinforcement-Based Push-Pull Approach for Peer-to-Peer Live Streaming*. in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on.*

[29] Giacomazzi, P. and A. Poli. *Push-pull techniques in peer-to-peer video streaming systems with tree/forest topology*. in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on.*

[30] Duhwan, J., et al., 2011, *Adaptive Push-Pull Protocols for P2P-Based Video Streaming.* IEICE Transactions on Communications. **94-B**(10): p. 2759-2762.

[31] Zhang, M., et al., *A peer-to-peer network for live media streaming using a push-pull approach*, in *Proceedings of the 13th annual ACM international conference on Multimedia.* 2005, ACM: Hilton, Singapore. p. 287-290.

[32] Barekatain, B. and M.A. bin Maarof. *Network coding efficiency in live video streaming over Peer-to-Peer Mesh networks*. in *Information Technology in Asia (CITA 11), 2011 7th International Conference on.*

[33] Nguyen, K., T. Nguyen, and S.-C. Cheung, 2010, *Video Streaming with Network Coding.* Journal of Signal Processing Systems. **59**(3): p. 319-333.

[34] Fragouli, C. and E. Soljanin, 2007 of Conference. *Network Coding Applications*: now published Inc.

[35] Mirkhosraie, S. and M. Hefeeda, *Live Peer-to-Peer Streaming with Scalable Video Coding and Network Coding*, in *In Proc. of ACM Multimedia Systems (MMSys'10).* 2010: Scottsdale. p. 123-132.

[36] Wang, M. and L. Baochun. *Lava: A Reality Check of Network Coding in Peer-to-Peer Live Streaming*. in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE.*

[37] Anh, N., L. Baochun, and F. Eliassen. *Chameleon: Adaptive Peer-to-Peer Streaming with Network Coding*. in *INFOCOM, 2010 Proceedings IEEE.*

[38] Tabatabaii, H.S.A., M. Khansari, and H.R. Rabiee. *LiveCod: A mesh-pull P2P live streaming system with XOR-based Network Coding*. in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE.*

[39] Magharei, N. and R. Rejaie, 2009, *PRIME: peer-to-peer receiver-driven mesh-based streaming.* IEEE/ACM Trans. Netw. **17**(4): p. 1052-1065.

[40] Lou, X. and K. Hwang, 2012, *Quality of data delivery in peer-to-peer video streaming.* ACM Trans. Multimedia Comput. Commun. Appl. **8s**(1): p. 1-23.

[41] Bo, L. and Y. Hao, 2007, *Peer-to-peer live video streaming on the internet: issues, existing approaches, and challenges.* Communications Magazine, IEEE. **45**(6): p. 94 - 99.

[42] Arizona, A.S.U. *Video Traces Research Group on MPEG-4*. 2009; Available from: http://trace.eas.asu.edu/mpeg4/index.html/.

[43] Liu, Z., et al., *Substream trading: Towards an open P2P live streaming system.*, in *IEEE Conference on Network Protocols (ICNP'08)*. 2008: Orlando, FL. p. 94–103.

[44] Auwera, G.V.d., T.D. Prasanth, and M. Reisslein, September 2008, *Traffic and Quality Characterization of Single-Layer Video Streams Encoded with H.264/MPEG-4 Advanced Video Coding Standard and Scalable Video Coding Extension.* IEEE Transactions on Broadcasting. **54**(3): p. 698-718.

[45] Wang, M. and B. Li, 2007, *R2: Random Push with Random Network Coding in Live Peer-to-Peer Streaming.* Selected Areas in Communications, IEEE Journal on. **25**(9): p. 1655-1666.

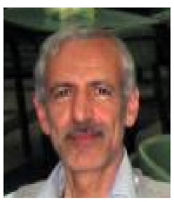
[46] Heide, J., et al., 2012, *Chapter 4 - Network Coding in the Real World*, in *Network Coding*, M. Muriel and S. Alex, Editors. Academic Press: Boston. p. 87-114.

[47] Li, Y.-S., H.-Y. Teng, and R.-H. Hwang, *P2P SVC-encoded video streaming based on network coding*, in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*. 2010, ACM: Caen, France. p. 1277-1281.

[48] Lu, J., S. Xiao, and C.K. Wu, 2010, *Efficient SVC transmission system based on network coding.* Xi'an Dianzi Keji Daxue Xuebao/Journal of Xidian University. **37**(3): p. 405-411.

BIOGRAPHY

Behzad Soleimani Neysiani earned his BSc in computer software engineering form IAUN in 2009. He is now a master student at the computer engineering faculty in IAUN. His research interests encompass peer-to-peer networking, computer networks and software engineering. His master is in the area of P2P live video streaming.



Assoc. Professor Naser Neematbakhsh is currently affiliated with faculty of computer engineering, Isfahan university, Iran. He received his BSc degree from Isfahan university in 1973, MS degree from Worcwstechnic Inst., USA in 1978 and PhD from Bradford University, UK, in 1989. His main research interests include Modeling, Performance evaluation, Reliability and software engineering.

Behrang Barekatain earned his BSc and MSc in computer software engineering in 1996 and 2001, respectively. He has more than 15 years experience in computer networking and security. Now, he is affiliated with the computer engineering faculty in Islamic Azad University, Iran, since 2002. He is now a PhD candidate at the Department of Computer Networks of the Computer Science and Information System Faculty (FSKSM) in Universiti Teknologi Malaysia (UTM). His research interests encompass wired and wireless networking, Peer-to-Peer networking, network coding, video streaming, network security and design. His PhD is in the area of P2P live video streaming over wired and wireless mesh networks using network coding.

Professor Mohd Aizaini Maarof is currently affiliated with faculty of computer science and information system in Universiti Teknologi Malaysia (UTM). He obtained his PhD degree at the Faculty of Computer Science, Aston University of United Kingdom in 1999, and specializes in IT security. He earned his MSc and BSc in computer science from Central Michigan University (1988) and Western Michigan University (1986), USA, respectively. His research interests encompass computer networks, IT security, Intrusion Detection System and Image Forgery Detection System.

It is necessary to say thanks to Assoc. Prof. Dr. Kamran Zamanifar for his support in this research. He is now an Assoc. professor in computer engineering faculty at University of Isfahan, Iran. He received his PhD in parallel and distributed systems from Leeds University, UK, In 1995. He is now an Assoc. professor in computer engineering faculty at University of Isfahan. His current research interests are in P2P systems, parallel and distributed systems, pervasive computing, and Cloud computing as well as soft computing.