

Distributed Client-Assisted Patching for Multicast Video-on-Demand Service in an Enterprise Network

Md. Mostofa Akbar*, S. M. Farhad, Munima Jahan, Md. Humayun Kabir

Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka-1000, Bangladesh.

* Corresponding author. Tel: +8801715006345; email: mostofa@cse.buet.ac.bd

Manuscript submitted October 30, 2015; accepted July 21, 2016.

doi: 10.17706/jcp.12.6.511-520

Abstract: Video-on-demand services allow users to select their desired video without watching the broadcast. Videos are stored in distributed servers. A single stream from the server can be utilized to satisfy a batch of common client requests. The key problem of this system is to minimize the server load to satisfy maximum number of clients. We propose a distributed client-side caching technique that reduces the server load significantly and increases the scalability of the system. We have implemented our system and compare our system with another state-of-the-art technique. Experimental results suggest that the newly proposed system outperforms the already proposed client assisted patching scheme.

Key words: Distributed network, video-on-demand, multimedia.

1. Introduction

Video on demand (VoD) provides customers with informative and entertaining streams of multimedia and video information. A multicast VoD system (termed as Near VoD service) allows clients to share a server stream by batching their requests [1], and hence, improves channel utilization. The key idea is to avoid transmitting the same packet more than once on each link of the network by having branch routers duplicate and then send the packet over multiple downstream branches. Multicast can alleviate the workload of the VoD server and improve the system throughput by batching requests and reduce the required network bandwidth significantly, thereby decreasing the overall network load. It offers high scalability which, in turn, increases the system capacity to house large number of clients and provides considerable cost/performance benefits. However, it increases initial service latency that may cause some impatient customers to renege. Patching is a multicast technique that enables a server to transmit only the beginning of the entire video data to clients and ensures that clients download the rest data of the video from an ongoing stream [2]. By making multiple clients share an ongoing stream, Patching can reduce server network bandwidth requirements for True VoD (TvoD) services. *Double Patching* [3] ensures that a long patching stream delivers not only essential data for the current client but also extra data for future clients, so it significantly reduces the total amount of video data delivered by all streams. Patching eliminates the service latency imposed by the Batching scheme. The objective of Patching is to substantially improve the number of requests each channel can serve per time unit, thereby sufficiently reducing the per-customer system cost. In Patching scheme channels are often used to patch the missing portion of a service or deliver a patching stream, rather than multicasting the video in its entirety. In Patching, a client might have to download data from both regular multicast and patching channels simultaneously. However, Patching temporarily puts a heavy load on the servers as

patching streams are dedicated to the patched clients.

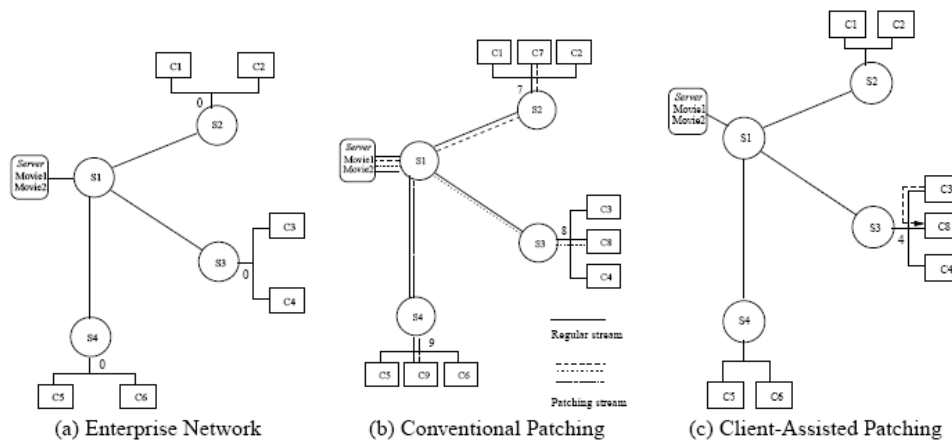


Fig. 1. Client assisted patching and conventional patching in an enterprise network.

Client Assisted Patching (CAP) [4] uses client side cache to reduce the server load. In this approach, all patching channels are provided by the cooperative clients rather than the server itself. Thus, the system alleviates server load and the conserved bandwidth can be used to satisfy more multicast groups. It also increases the throughput and scalability of the system. The minimum buffer requirement in the intervals is same as the conventional patching scheme requires. Client Assisted Patching technique is illustrated in Figure 1. The patching stream is released when the missing portion is made up and the client will continue with the regular stream until the end of the session.

Client Assisted Patching reduces the server load by using the client side cache. However client cache is limited and only a small portion of the video can be stored in a single client to patch. In this paper, we have proposed *Distributed Client-Assisted Patching* (DCAP) where different clients will store different portion of the movie unlike the standard client-assisted system. Thus the system can provide a larger amount for patching instead of only the beginning portion.

2. Related Research Work

An adaptive logic designed to support multicast constraints at the client side is illustrated in [5]. A broadcast based network coded technique that allows the implementation of multicast VoD system utilizing a single serve channel is presented in [6]. Multicast VoD has good scalability and excellent cost/performance efficiency (See [7] for an excellent survey of multicast VoD services). However, it is difficult to support VCR-like interactivity with multicast VoD and, at the same time, improve service efficiency. There are several proposals [8]-[10] to solve this problem. One of the earlier periodic broadcast schemes was the Equally-spaced interval Broad casting [1]. Since it broadcasts a given video at equally-spaced intervals, the service latency can only be improved linearly with the increase of the server bandwidth. To significantly reduce the service latency, Pyramid Broadcasting (PB) was introduced in [11]. In PB, each video file is partitioned into the segments of geometrically-increasing sizes, and the server capacity is evenly divided into K logical channels. This ensures a smaller waiting time for every video. Some other works [12], [13] are also discussed in the literature to address different issues of periodic multicast VoD services. Cooperative client approach is recently proposed approach that relies on the cooperation of the video clients in forming an overlay network over which the video is propagated [14], [15]. To eliminate the service latency of batching, several dynamic multicast techniques have been proposed in [15]-[18]. Given that there is an existing multicast video, when to schedule another multicast for the same video is crucial.

The time limit, up to when a newly arrived client will be patched after a multicast session starts, is called

patching window [15], [19]. Some modifications of patching technique are discussed in [17], [20]. Two simple approaches of setting the patching window are discussed in [15]. The first one uses the length of the video as the patching window. That is, no multicast is initiated as long as there is an in-progress multicast session for the video. This approach is called the Greedy Patching because it tries to exploit an in-progress multicast as much as possible. However, over-greed can actually reduce data sharing [15]. The second approach, called the Grace Patching, uses a patching stream for the new client only if it has enough buffer space to absorb the skew. A multicast technique that completely prevents a server from transmitting unnecessary video data like double patching and uses four types of streams: regular stream (R-stream), patching stream (P-stream), short patching stream (S-stream) and linking stream (LK-stream). An R-stream, an S-stream, multicast window W_m and patching window W_p in the proposed XP4S [21]. Range Multicast is a new communication paradigm for VoD applications [18]. This scheme is a shift from a conventional thinking about multicast where every receiver must obtain the same data packet at all times. Client-to-client streaming scheme for VoD application [22] presents a technique to cooperatively stream the video using chaining technique with unicast communication among the clients. An enhanced client-centric approach for efficient video broadcast is described in [23]. P2Cast-an architecture that uses a peer-to-peer approach to cooperatively stream video using patching techniques, while only relying on unicast connections among peers. MegaDrop [24] is a fully decentralized VoD service via P2P techniques. The MegaDrop system not only takes active peers into consideration but also provides mechanisms for discovering inactive peers that contain desired media objects. A Zero Delay Video-on-Demand Scheme is a technique proposed to solve the server delay of high-performance static streaming scheme [25]. It contains the outstanding thoughts from two efficient schemes: GEBB and patching scheme. P-chaining proposes a service scheme based on chaining [26], in which clients as well as the server provide streaming services. In the proposed scheme, services are provided by unicast and managed locally using node lists. Proxy-assisted scalable periodic broadcasting of videos for heterogeneous clients is proposed in [27]. This scheme is proposed to significantly reduce the waiting time of all heterogeneous clients, without the need for any additional backbone bandwidth.

3. Distributed Client-Assisted Patching

It has already been mentioned that Client Assisted Patching [4] reduces the server load by using the client side cache to store the initial portion of the movie. In this approach, all patching channels are provided by the cooperative clients. Thus, the system alleviates server load and the conserved bandwidth can be used to satisfy more multicast groups.

The patch window during the multicasting is defined by the duration when a particular client can join the multicast group. Generally to get the True VoD (TVoD) service a new multicast session is initiated at the end of patch window duration. The number of users in a multicast group depends on the size of window. If the window size is increased we can have less number of simultaneous multicast sessions for a particular video stream and the server load will be decreased. This decreased server load could be utilized for other streams (e.g., movies) to attract more users with increased revenue. But increased window size requires larger buffer in the clients and client buffer is limited. So storing the same portion to all clients can provide only a limited portion as patch to other clients. If the storage required for storing the initial portion of the movies can be distributed among a group of clients then it is possible to support a larger window size with the same amount of buffer in the clients. Thus it requires less buffer and a large multicast group can be supported with the same network bandwidth.

The Distributed Client Assisted Patching technique applies the idea of storing the initial portion of the session among the participating clients. Some observation about the new scheme can be described as

follows if the patching window is doubled (i.e., $2W$ instead of W) keeping the client buffer for patching W :

- The clients are allowed to join an ongoing multicast session within $2W$ time from the beginning of the session. Where W is the patch window in client assisted patching. Thus the effective patching window is doubled.
- Each client in a multicast session will store a portion of the video with length W depending on its arrival time. The distribution policy is described in the later section.
- If a client requests for a movie after starting of multicast session the ADC (Admission Controller, accessible through network) will select one or two patch parents for the client. Different parts of the patching stream may be supplied by different clients.
- In the new scheme the patch streams are distributed among the clients. It means that a client will not store the whole patch stream and the ADC must find multiple clients for patching. The main problem here is the reduced probability of finding a patch client. Thus in some cases waiting time will be increased and the percentage of served will be decreased. But as we are getting larger patch window we can effectively get more clients to be served with single movie stream.

3.1. Distribution Policy

In our proposed system each client in a multicast session will store a portion of the initial part of the movie with length W depending on its arrival time. The distribution policy of the part of the movie to be stored in patched clients is described in the following table.

Table 1. Distribution Policy Used in the Proposed System

Arrival time, t	Part of the Movie Stored, B
$0 \leq t < W/2$	$0 \leq B < W$
$W/2 \leq t < W$	$W/2 \leq B < W + W/2$
$W \leq t \leq W + W/2$	$W \leq B \leq 2W$
$W + W/2 < t \leq 2W$	None

3.2. Analysis for Server Bandwidth Requirement

In our analysis we will refer to the amount of data in terms of their play back duration. Let us assume both batching and patching be combined in this system. It is also assumed that batching and patching window size is W_B and W_P respectively and all clients have enough buffers to store necessary patch stream.

During the patch window of a particular multicast session there will be only one regular multicast stream transmitted by the server. The rest will be patched by the client. During the batching no new clients will be served and they will get the stream in the next multicast session. So the total data delivered by the server during one multicast session for the i th movie M_i is

$$D_s = L_i \tag{1}$$

where, L_i is the length of the i -th movie M_i . The patch streams for the late clients will be provided by the cooperative clients.

If k patching streams are initiated between t and $t + \Delta t$, then total data transmitted by k patch stream can be approximated as kt if Δt is negligible. If the probability of initiating k patch stream during Δt is $P(k, \Delta t)$

then the total data delivered by the clients between t and Δt is $\sum_{k=1}^{\infty} ktP(k, \Delta t)$. To calculate the mean total amount of video data delivered by a multicast group we can partition $(0, W_p)$ into $W_p/\Delta t$ small time

segment. Then total data transmitted by the client

$$D_c = \sum_{t=1}^{\lfloor \frac{W_p}{\Delta t} \rfloor} \sum_{k=1}^{\infty} ktP(k, \Delta t) \quad (2)$$

To determine the expression of the probability we assume that the multicast initiation process is Poisson with rate λ . The probability density function is $f_x = \lambda e^{-\lambda x}$, where x indicates the first time of the patch client arrival and f_x indicates the probability that the first client arrives at time x . Now we can derive $P(k, \Delta t) = (\lambda \Delta t)^k \frac{e^{-\lambda \Delta t}}{k!}$. D_c can be derived as follows:

$$\sum_{k=1}^{\infty} ktP(k, \Delta t) = \sum_{k=1}^{\infty} kt(\lambda \Delta t)^k \frac{e^{-\lambda \Delta t}}{k!} = t \sum_{k=1}^{\infty} (\lambda \Delta t)^k \frac{e^{-\lambda \Delta t}}{(k-1)!} = te^{-\lambda \Delta t} \sum_{k=1}^{\infty} \frac{(\lambda \Delta t)^k}{(k-1)!} = te^{-\lambda \Delta t} \lambda \Delta t e^{\lambda \Delta t} = t \lambda \Delta$$

If we set Δt equals to 1 second then we get

$$D_c = \sum_{t=1}^{W_p} t \lambda = \lambda \frac{W_p(W_p + 1)}{2} \quad (3)$$

Since the client request rate is λ , the patch window is W_p and the batch window is W_B the mean interval between two multicast group is $\tau = W_B + W_p$. Now consider N multicast session in a time interval T . Thus $N = T/\tau$. Total data stream supplied by these N streams will be $D_{st} = L_{avg} \times N$

$$\text{Thus the server bandwidth requirements} = D_{st}/T = \frac{L_{avg} \times N \times b}{T} = \frac{L_{avg} \times b}{\tau} = \frac{L_{avg} \times b}{W_B + W_p} \quad (4)$$

Again the patching by the client will be continued within patch window W_p . So the link bandwidth required for patch streams which is provided by the clients to other clients can be measured as

$$\text{Link bandwidth} = \frac{D_c}{W_p} b = \lambda \frac{(W_p + 1)}{2} b \quad (5)$$

where b is the video play back rate and L_{avg} is the average movie length. If we represent the patch window of Client Assisted Patching as W_{cap} we can write

$$W_p = 2W_{cap} \quad (6)$$

From Equation (2.4) we can see that the server bandwidth requirement is inversely proportional to the patch window. So the larger the patch window the less will be the requirement of server bandwidth. Again distributed Client Assisted Patching supports a patch window that is double of the patch window supported by the Client Assisted Patching system. So we can say that our system saves about 35% bandwidth than that of Client Assisted Patching if $W_B = W_{cap}$.

3.3. Complexity Analysis

The complexity of Admission-Control in Client Assisted Patching is $O(sE \log V + m^2 + n_c E \log V + n^2_c E)$ which is the sum of the complexity of two threads Batched-Requests-Processing and Online-Requests-Processing and the procedure *Init-Admission-Controller*. Here, s , m , V , E , n_r , and n_c indicates number of servers, movies, nodes, edges, clients seeking admission and clients already admitted respectively. Here admission controller invokes the procedure *selectPatchingParent* once to get a single

patch parent. Thus the highest order term in the time requirement to execute the procedure is n_c^2E . But in the Distributed Client Assisted Patching it requires two patch parents. The first parent will provide the first part of the required patch and the second parent will provide the rest. So the admission controller has to invoke the procedure *selectPatchingParent* twice to get two different patch parents. Thus the highest order term in the time requirement to execute the procedure will be $2n_c^2E$. So the total time complexity of the proposed system can be described as: $O(sE \log V + m^2 + n_cE \log V + 2n_c^2 E)$.

4. Performance Study

We study the behaviour of our approach and evaluate its performance based on the simulation results. We simulate Distributed Client Assisted Patching using Parsec, a C-based parallel discrete event simulation language. We also simulate Client Assisted Patching to make a comparison between our approach and Client Assisted Patching on various metrics. The same simulation settings has been used and the same assumptions were made as mentioned in [4].

4.1. Comparison with Client Assisted Patching

In Fig. 2 we observe that distributed scheme outperforms the Client Assisted Patching scheme as the distributed Client Assisted Patching scheme provides the scope of more patching streams. From the figure we observe that for the server with higher bandwidth and low request rate the percentage served is almost 100% for both the schemes. But when the server bandwidth is very low with high request rate both the system will perform worse as there will be contention for resources.

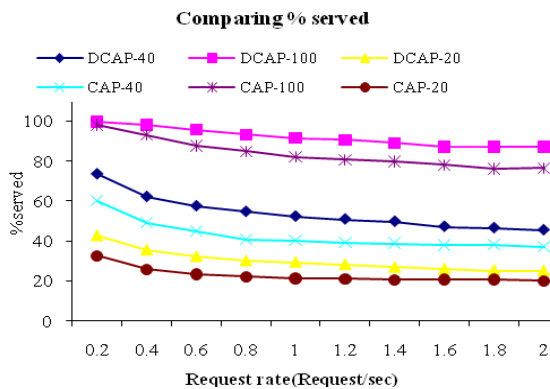


Fig. 2. Percentage of served requests for different request-rates and different server bandwidth.

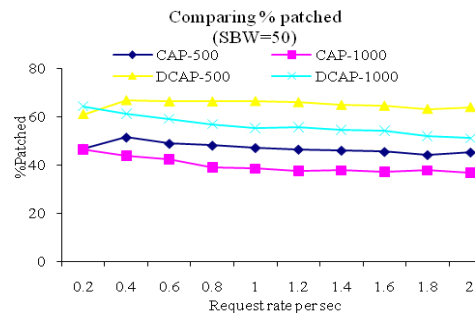


Fig. 3. Showing percentage of patched requests for different request rates with medium server bandwidth and different number of clients. (The number in the legend represents the number of users.)

It is observed from Fig. 3 that percentage of patching degrades with the increase in request rate. For a system with low server bandwidth there will be less number of multicast movie streams and the large number of clients will be competing for patching. This will create contention in link bandwidth and the clients will be rejected for patching when there are not enough resources in links. We do not observe this degrading behaviour of the percentage of patched clients in Fig. 4 as there are sufficient resources. Here we observe that when both the systems have enough server bandwidth Distributed Client Assisted Patching performs 30% better (on the average) than the client assisted patching as this new system provides a larger patching window. When the available server bandwidth is lower as shown in the Fig. 3 the performance of the Client Assisted Patching further degrades where as our system shows almost the same performance as before.

Fig. 5 shows that the average waiting time increases with the increase in request rate. An environment with higher request rate makes shortage of resources. So some clients may need to wait for the necessary

resources. Actually the lack of server bandwidth may not allow frequent batching for a multicast movie stream and clients need to wait for a long time to get available resources. This waiting time for batching plays a significant role in average waiting time. The Distributed Client Assisted Patching overcomes this problem by allowing the broader scope of patching and hence reduces the average waiting time which is observed in the figures.

From Fig. 6 we can see that our scheme needs about 25% less bandwidth than that of Client Assisted Patching on the average. This is less than the mathematical analysis presented earlier. The main reason behind this is the patch clients which were served directly from the server. We also see that Conventional Patching scheme requires huge bandwidth compared to other two schemes because in conventional patching all patch streams are provided from the server.

As per complexity analysis presented in Section 3.3 Distributed Client Assisted Patching needs more computation to find a patch parent than Client Assisted Patching. This is observed in our simulation results as shown in Fig. 7. The quadratic pattern of time requirement curve justifies the complexity analysis expression.

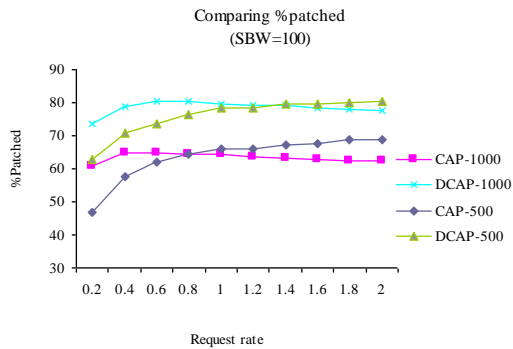


Fig. 4. Showing percentage of patched requests for different request rates (request/sec) with high server bandwidth and different number of clients. (The number in the legend represents the number of users.)

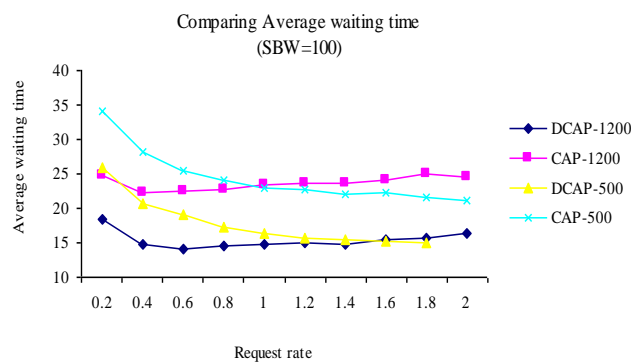


Fig. 5. The average waiting time for a client with different request rate (request/sec) for 100 unit of server bandwidth. (the number in the legend represents the number of users.)

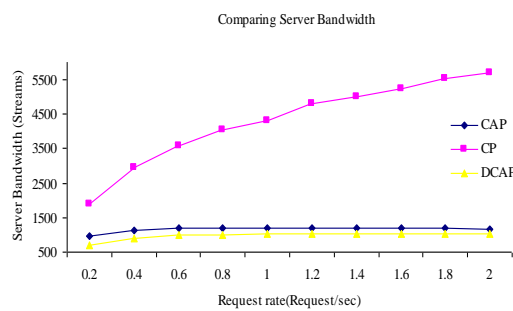


Fig. 6. Server bandwidth requirements for different request rates of client-assisted, distributed client assisted and conventional patching schemes (cp means conventional patching).

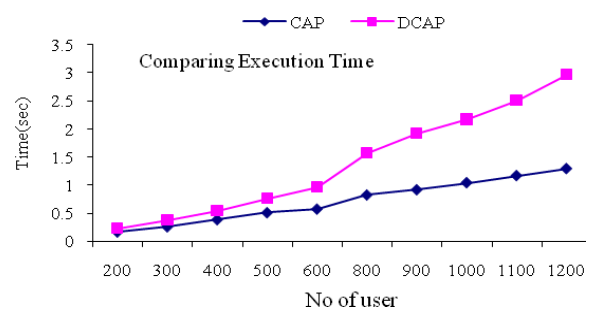


Fig. 7. Computation time of conventional and client assisted patching schemes with varying number of clients in the system.

5. Conclusion

Since streaming of any multimedia object like high quality video consumes a significantly large amount of network resources, network bandwidth limitation is the major constraint in most of the multimedia systems. Multicast Video on-Demand (MVoD) systems with Patching are scalable and cheap-to-operate. To

optimize system performance, over the past few years extensive research has been done on MVoD. Even though the significant progress has been made, it is still regarded as challenging research domain in VoD service. In our proposed Distributed Client Assisted Patching a client gets a larger amount of patch stream from two different patch parents. Thus our system provides larger patch window.

The mathematical analysis and simulation results reveal that the new scheme performs better in terms of number of served requests and bandwidth requirements for servers with expense of additional computation time to find one additional patch parent. It can be shown mathematically that increasing the patch window decreases the server bandwidth requirement. But increasing patch window (even if the clients have the higher capacity) makes it difficult to find the patch parents and there is a chance of overloading the network. Finding the optimal patch window size is a potential future in the area of Video on Demand.

Acknowledgement

The authors acknowledge Bangladesh University of Engineering and Technology to support this research by providing logistic support for computing facility of Department of CSE and funding necessary fees to attend conference.

References

- [1] Dan, A., Sitaram D., & Shahabuddin, P. (1994). Scheduling policies for an on-demand video server with batching. *Proceedings of the Second ACM International Conference on Multimedia* (pp. 15–23). New York, NY, USA.
- [2] Hua, K. A., Cai, Y., & Sheu, S. (1998). Patching: A multicast technique for true video-on-demand services. *Proceedings of the Sixth ACM International Conference on Multimedia* (pp. 191-200). New York, NY, USA.
- [3] Cai, Y., Tavanapong, W., Hua, K. A., *et al* (2007). A double patching technique for efficient bandwidth sharing in video-on-demand systems. *Multimedia Tools Appl*, 32(1), 115–136.
- [4] Farhad, S. M., Akbar, M. M., Kabir, M. H., *et al* (2009). Multicast video-on-demand service in an enterprise network with client-assisted patching. *Multimedia Tools Appl*, 43(1), 63–90.
- [5] Dubin, R., Dvir, A., Hadar, O., Harel, N., & Barkan, R. (2015). Multicast adaptive logic for dynamic adaptive streaming over http network.. *Proceedings of Workshops in Computer Communications* (pp. 269-274). Hong Kong.
- [6] Fu, A., & Sadeghi, P. (2014). A network coding algorithm for video on demand multicast using a single server channel. *Proceedings of Workshop in Communications Theory* (pp. 5-10). Sydney, NSW, Australia.
- [7] Ma, H., Shin, K. G., *et al* (2002). Multicast video-on-demand services. *SIGCOMM Computer Communication Review*, 32(1), 31-43.
- [8] Abram-Profeta, E. L., & Shin, K. G. (1998). Providing unrestricted VCR functions in multicast video-on-demand servers. *Proceedings of the IEEE International Conference on Multimedia Computing and Systems* (pp. 66–75). Washington, DC, USA.
- [9] Almeroth, K. C., Ammar, M. H., *et al* (1996). On the use of multicast delivery to provide a scalable and interactive video-on-demand service. *IEEE Journal on Selected Areas in Communications*, 14, 1110–1122.
- [10] Liao W., Li, V. O. K., *et al*. (1997). The split and merge protocol for interactive video-on-demand. *IEEE MultiMedia*, 4(4), 51-62.
- [11] Aggarwal, C., Wolf, J., & Yu, P. (1996). A permutation-based pyramid broadcasting scheme for video-on-demand systems. *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems* (pp. 118-126). Hiroshima, Japan.
- [12] Hua, K. A., & Sheu, S. (1997). Skyscraper broadcasting: A new broadcasting scheme for metropolitan

- video-on-demand systems. *Proceedings of ACM SigComm*. (pp. 89-100). New York, USA.
- [13] Jung, J., & Lee, D., *et al.* (1997) Harmonic broadcasting for video-on-demand service. *IEEE Transactions on Broadcasting*, 43(3), 268-271.
- [14] Guo, M., & Ammar, M. H. (2004). Scalable live video streaming to cooperative clients using time shifting and video patching. *Proceedings of Infocom* (pp. 1501-1511). Hong Kong.
- [15] Guo, M., & Ammar, M. H. (2004). Scalable live video streaming to cooperative clients using time shifting and video patching. *Proceedings of Infocom* (pp. 1501-1511). Hong Kong.
- [16] Carter, S. W., & Long, D. D. E. (1997). Improving video on demand server efficiency through stream tapping. *Proceedings of 5th International Conference on Computer Communications and Networks* (pp. 200–207). Las Vegas, NV, USA.
- [17] Ma, H., Shin, K. G., Wu, W., *et al.* (2005). Best-effort patching for multicast true VoD. *Service, Multimedia Tools and Applications*, 16(1), 101-122.
- [18] Hua, K. A., Tran, D. A., *et al.* (2005). Range multicast for video on demand. *Multimedia Tools Appl.*, 27(3), 367-391.
- [19] Cai, Y., Hua, K., & Vu, K. (1999). Optimizing patching performance. *Proceedings of the IS&T/SPIE Conference on Multimedia Computing and Networking* (pp. 204–215). San Jose, CA, USA.
- [20] Cai, Y., & Hua, K. A. (1999). An efficient bandwidth-sharing technique for true video on demand systems. *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)* (pp. 211–214). New York, USA.
- [21] Natarajan, A., Cai, Y., Wong, J., *et al.* (2009). An enhanced client-centric approach for client video broadcast. *Multimedia Tools Appl.*, 43(2), 179-193.
- [22] Dakshayini, M., & Nair, T. R. G. K. (2010). Client-to-client streaming scheme for VOD applications. *CoRR*, abs/1005.5436.
- [23] Natarajan, A., Cai, Y., Wong, J., *et al.* (2009). An enhanced client-centric approach for client video broadcast. *Multimedia Tools Appl.*, 43(2), 179-193.
- [24] Chen, J., Leu, J., Chen, Y., Wei, H., & Shih, W. (2011). Megadrop: A cooperative video-on-demand system in a peer-to-peer environment. *Journal of Information Science and Engineering*, 27(4), 1345–1361.
- [25] Wang, Y., Zhang, Y., Huang, X., *et al.* (1997). A new zero-delay video-on-demand scheme. *Journal of Information & Computational Science*, 7, 2122-2129.
- [26] Kim, H., Yeom, H. Y., *et al.* (2008). P-chaining: A practical vod service scheme astronomically handling interactive operations. *Multimedia Tools Appl.*, 39(1), 117-142.
- [27] Kusmirek, E., Du, D. H., *et al.* (2008). Proxy-assisted periodic broadcast for video streaming with multiple servers. *Multimedia Tools Appl.*, 36(3), 243-266.



Md. Mostofa Akbar received his B.Sc. and M.Sc. engineering degree from BUET in 1996 and 1998 respectively. He received his PhD in computer science from University of Victoria, Canada in 2002. He is working as Professor of Dept of CSE, BUET.



S. M. Farhad is working as Associate Professor of the Dept. of CSE, BUET. He received his B.Sc. and M.Sc. engineering degrees from BUET in 2003 and 2007 respectively. He received his PhD in computer science from University of Sydney, Australia in 2012.



Munima Jahan received her B.Sc. engineering degree in CSE from Khulna University in 2004. She received her M.Sc. engineering degree from BUET in 2011. Now she is pursuing her PhD in University of Calgary.



Md. Humayun Kabir received his B.Sc. and M.Sc. engineering degrees from BUET in 1993 and 1998 respectively. He received his PhD in computer science from University of Victoria, Canada in 2005. He is working as Professor of the Dept. of CSE, BUET.