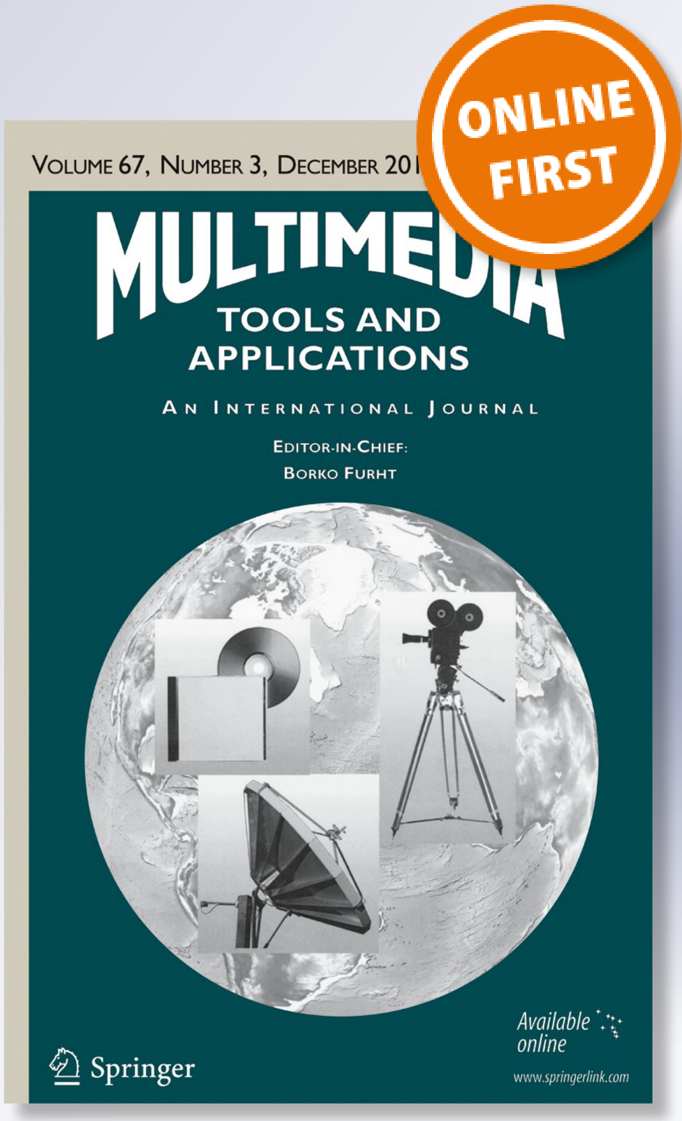
Popularity based hierarchical prefetching technique for P2P video-on-demand

Sivakumar Ganapathi & Venkatachalam Varadharajan

Multimedia Tools and Applications
An International Journal

ISSN 1380-7501

Multimed Tools Appl
DOI 10.1007/s11042-017-5167-y



Multimed Tools Appl
DOI 10.1007/s11042-017-5167-y

Popularity based hierarchical prefetching technique for P2P video-on-demand

Sivakumar Ganapathi[1] · **Venkatachalam Varadharajan**[2]

Received: 31 October 2016 / Revised: 20 July 2017 / Accepted: 29 August 2017

Abstract In Peer-to-Peer Video on Demand System like Video Cassette Recording (VCR) various operations (i.e. forward, backward, resume) are found to be used very frequently. The uncertainty of frequent VCR operations makes it difficult to provide services to play as download and also the workload on VoD server is very high because of it handles the overall network. So it may sometimes hinder the network performance. To overcome these issues we propose to develop popularity based hierarchical prefetching technique in this paper. In this technique, the network is considered in a hierarchical topology and then the popularity level of each video is estimated by the proxy and based on it, the popular video is cached in the proxy and streamed to the required peer nodes in the network. Simulation results of our work show that our proposed prefetching technique outperforms the delay of the previous technique.

Keywords VoD · P2P network · Communication · Prefetching technique

1 Introduction

1.1 Peer-to-peer network

In Peer-to-peer (P2P) network, each peer node links with other peers directly instead of communicating with the server directly. When compared with the centralized network, the P2P network has several advantages such as anonymity can be maintained during communication, ease in developing this network, etc. [14]. In a P2P network, the nodes communicate with the remaining peer nodes, node shares its information, offer services

✉ Sivakumar Ganapathi
sivakumarg0574@gmail.com

1 Department of Computer Science and Engineering, Erode Sengunthar Engineering College, Erode, Tamilnadu, India

2 Department of Computer Science and Engineering, Erode Sengunthar Engineering College, Erode, Tamilnadu, India

to other nodes, carry out various network interactions, etc. A P2P network can operate even in the absence of the centralized controlling node ie., it can operate even as a decentralized network. So, in a P2P network, all the communicating nodes are considered equal or peer. To secure the communicated data, the authentication procedure is used in the P2P network [4].

In the structured P2P network, the peers are linked according to the routing algorithms like the distributed hash table-based (DHT) is utilized in P2P network for indexing. In an unstructured P2P network, the routing algorithms are not utilized for the purpose of organizing and optimizing the connections between the peers. The links are connected randomly and form an unstructured P2P network. The already created links are copied and used to develop new peers and later on the peers create its own links [2].

1.2 P2P video streaming

From the past few years, on the internet, the Video-on-Demand (VoD) streaming service has become extremely popular, for instance: the Internet TV, online video, distance education, etc. In the networks with the dynamic heterogeneous environment, the P2P computing technique is credited as the most effective technique to offer the "play-as-download" VoD services. In order to achieve large-scale VoD services, the gossip-based P2P streaming is considered to robust and dependable after a thorough investigation. In the conventional gossip based P2P network with the VoD servers, every peer arbitrarily links with its peers whenever the playback offsets known as partners are available. In P2P network, the peers keep sharing the available data with its partners and thus when in need of a specific video, the node collects it from its partners.

The gossip based P2P VoD servers are capable of attaining increased resilience, keeping the protocol overhead minimum, when considered in comparison with the tree-based P2P technique. So, the gossip-based P2P technique is immensely used [17]. It is hard for the VoD servers to handle VCR tasks like jump, pause, fast forward and rewind because of the robust features of the client interaction. The older versions of the VoD servers managed VCR tasks by maintaining the index overlay level moderate so that the segment relocation can be performed quickly. During the occurrence of the VCR operation, initially, the requested video/data segment is relocated by the peer node, downloaded and then finally played. This technique increases the response latency and thus results in the deterioration of the video quality which can be either freezing of the playback or video blackout. As a result, it is important to attain quick resolution [17].

1.3 Prefetching techniques for P2P video streaming

The prefetching technique is used to minimize the response latency caused due to the interactive requests. This technique is efficient in the P2P network since it is rich in resource. In the prefetching technique, to avoid the problems like burst package loss, and then to enhance the playback quality, the downloading of the video is performed in advance. The conventional prefetching technique is not capable of aiding the VCR operations because the user interaction has dynamic features. It is effective to use prediction based prefetching since it maintains the video segments which have a high probability of being requested in the near future [17].

1.4 Problem definition

Ye TIAN et al. [15] presented popularity oriented proxy caching for peer-assisted Internet video-on-demand streaming services. They developed a practical protocol named PopCap for Internet VoD services. By reducing the video server workload and avoiding inter-domain traffic using their proposed method they reduced the video caching problem. However, delay of their proposed is to be reduced. So, In this paper, we propose Popularity Based Hierarchical Prefetching Technique for P2P Video-on-Demand. Using this method, we form a basic hierarchical structure. Peer nodes in the structure are layered on the basis of time. After the structure formation, the proxy server in the structure caches the video based on popularity.

Rest of this paper organized as follows. Section 2 relates our previous work with our proposed work. Our proposed hybrid of popularity based hierarchical prefetching technique is presented in Section 3. Results of our proposed work are discussed in Section 4. This paper is concluded with Section 5.

2 Related works

Efthymiopoulos Nikolaos et al. [13] have presented a locality aware and balanced overlaying technique for P2P network with live streaming. This technique adjusts to the unstable nature of the member nodes and the overall network. In this paper, two algorithms: placement algorithm and the swapping algorithm are used to develop and maintain the overlay. The two algorithms handle the process of including a new node in the network along with the optimizing process of the overlay for the dynamic and distributed condition.

KalmanGraffi et al. [5] have presented a DHT-based information gathering and analyzing architecture to handle the incoming request in the network. This paper presents an architecture for depicting the distributed video streaming situation. This paper shows the DHT based architecture, which is used to store all the details of the sources providing the required data/content. Initially, a scoring function is designed as a solution which selects the effective stream provider. The decisions related to the balancing of the load are optimized by the scoring function by taking into account the heterogeneity of the peer nodes. According to the second solution provided, the stream provider has selected arbitrarily.

Jun-Li Kuo et al. [11] have proposed a cross-layer design for P2P network over MANET for handling and maintenance of the overlay and then choose the best route for multicasting the video. The delivery proficiency is enhanced by this technique, COME-P2P as it combines P2P DHT-based lookup with the IPv6 routing header. Based on the cross-layer design, mobility is determined by the lower layer and it informs the higher layer, which updates its routing table. Then the higher layer stores all the effective path details and informs the lower layer, which updates its routing table accordingly. This paper efficiently explains the maintenance of the routing paths as stable for live streaming through the IPv6 routing methodology. The propagation delay can be reduced by overlay proximity and the traffic bottleneck can be overcome by hop by hop routing.

Qiang Wang et al. [16] have presented a popularity-aware prefetch technique. According to this technique, the poorly replicated data items which will be requested in the network in the subsequent requests is cached. This makes the network maintenance less expensive.

Changqiao Xu et al. [18] have presented a distributed Storage-assisted Data-driven overlay Network (SDNet) to aid the P2P Video-on-Demand (VoD) services. The SDNet combines the two network structures: a Data-driven Overlay Network (DONet) and a multi-way tree. In the DONet, the buffer overlapping technique and the gossip protocol is used to perform the video distribution regularly. To aid the VoD server functioning, a new algorithm is proposed. This proposed algorithm attains its target by utilizing the multi-way tree structure and also the surplus prefetching buffers. In this technique, the video is fragmented into fixed sized segments, then it is prefetched and cached in the tree structure in a distributed format. The commands for interaction during the multimedia streaming are enhanced by the combined functioning of the DONet-based video delivery and the tree located multimedia components.

Yasuaki YUJI [19] has presented a technique for minimizing the playback suspension in a Video-on-Demand system based on the Peer-to-Peer technology (P2P VoD) in two phases. Initially, a hierarchical P2P network architecture is proposed based on the presence of swarms. Basically, a swarm is a set of peers with alike playback location and these swarms are linked in such a way that the requested video segment is transferred from one swarm to the next in a method similar to that of bucket brigade as the transferred video is controlled by the super peer (SP). In the second phase, the matchmaking technique is proposed which matches the requests with the available uploaders.

Ubaid Abbasi et al. [1] have presented a prefetching technique which works cooperatively and is known as “cooching”. In this technique, prior using the information that is gathered through the gossiping process, the requested fragments of the VCR activities are cached into the session.

Mohamed Hefeeda et al. [6] have proposed a caching algorithm for the P2P network on the basis of object fragmentation, partial admission of the data in predefined proportion and then object eviction.

Chung-Nan Lee et al. [7] have proposed an interleaving two-level cache with network coding (I2CC) scheme. This scheme provided rapid content searching and efficiently supports random seeking with the dynamic nature of peers. They also presented an analytical model for determining appropriate system parameters to reduce the chunk loss probability. Simulation results of this proposed scheme showed that which outperformed existing methods in terms of startup delay, jump delay, and server stress under different peer arrival times and video bitrates.

Eunsam Kim and Jonathan C.L. Liu [8] have proposed Integrated Prefetching/Caching (IPC) scheme. This scheme has taken benefit of both prefetching and interval caching using dynamic threshold values. The IPC scheduled incoming streaming requests so that utilization of both cache space and disk bandwidth was maximized. Simulation results of this IPC scheme showed that the number of concurrent streams significantly compared to when either prefetching or caching is employed alone.

3 Popularity based hierarchical prefetching technique

3.1 Overview

In this proposal, we propose popularity based hierarchical prefetching technique for P2P VoD.

In the proposed pre-fetching architecture for VoD in p2p networks, peer nodes form the basic structure, in which, the nodes are layered on the basis of time. The peer nodes which are joined in the beginning form the top most layer. Subsequent nodes are added as the child nodes, which forms the second layer and so on [3]. The design of proxy is proactive thus reducing the workload on VoD server [15]. Proxy keeps a cached copy of videos depending on their popularity [16].

Each peer node at top layer caches the first initial 'k' segments of video and never replaces this part until it departs [3]. Node 2 at second layer requesting video to the proxy is redirected to the node 1 at first layer, if that video is already received in node 1 and node 2 receives the video from the node 1 cache. The initial k segments of video at node 1 are transferred to node 2 and it is updated in the proxy. Now node 1 fetches next k segments from the proxy. If the video is rare and not available on any peer and proxy, then the request is forwarded to the VoD server. Video obtained from the VoD server to the node via proxy keeps the initial part in nodes cache. If the video is requested from another node the cached video is transferred otherwise the video is deleted and updated in the proxy (Fig. 1).

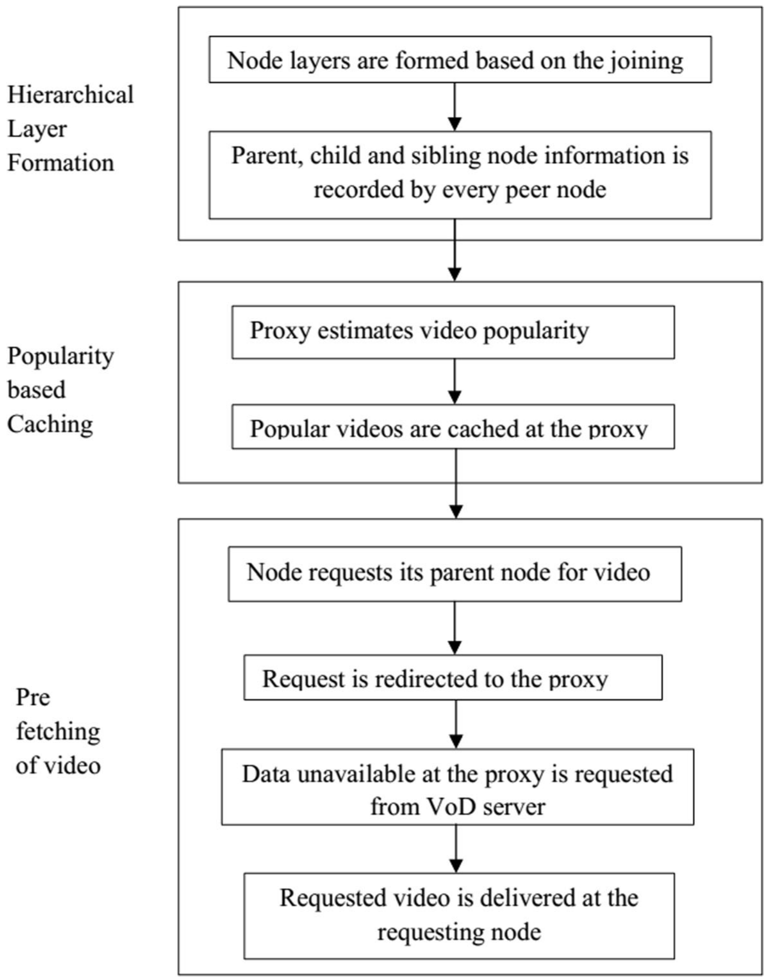


Fig. 1 Block diagram

3.2 Hierarchical prefetching mechanism using proxy

The network is developed in a hierarchical form [3]. The first layers of peer nodes around the VoD server are considered as the parent node to all the nodes in the next layer. The node in the first layer considers the nodes in the second layer as its child nodes. The nodes in the same layer are considered as a sibling of one another. At the prime locations of the network, the proxy is situated.

The hierarchical network is depicted in Fig. 2. It maintains a set of popular videos in its cache and streams it to the requesting node. Every node in need of video will request only its direct parent node and not the server. The request is fulfilled by the requested node with the aid from the proxy. If the requested video is not available in the proxy and other parent nodes, only then the VoD server is contacted. In this way, the workload on the server is reduced extensively. This process is described in algorithm 1.

In Fig. 2, the network hierarchy is depicted with the VoD Server, proxy, and the peer nodes. P1a, P1b, and P1c are the peer nodes belonging to the first layer and are a sibling to each other. P2a, P2b, P2c, P2d, P2e, P2f and P2g are the peer nodes belonging to the second nodes. P1a is the parent node of P2a and P2b and hence P2a and P2b are the child nodes of P1a. Similarly, P2c, P2d, and P2e are the child nodes of the parent node P1b. P2f and P2g are the child nodes of parent node P1c. When child node P2c is in need of a video, it requests its parent node, P1b. The parent node, P1b requests the proxy for the video. If the video is already cached in the proxy, then the proxy streams it and transfers it to the node P2c through P1b. Else, if the requested video is a rare video and hence is not available at the proxy. So, then the proxy requests the VoD server for the video. It streams the video and is delivered to the requesting node, through the proxy and parent node.

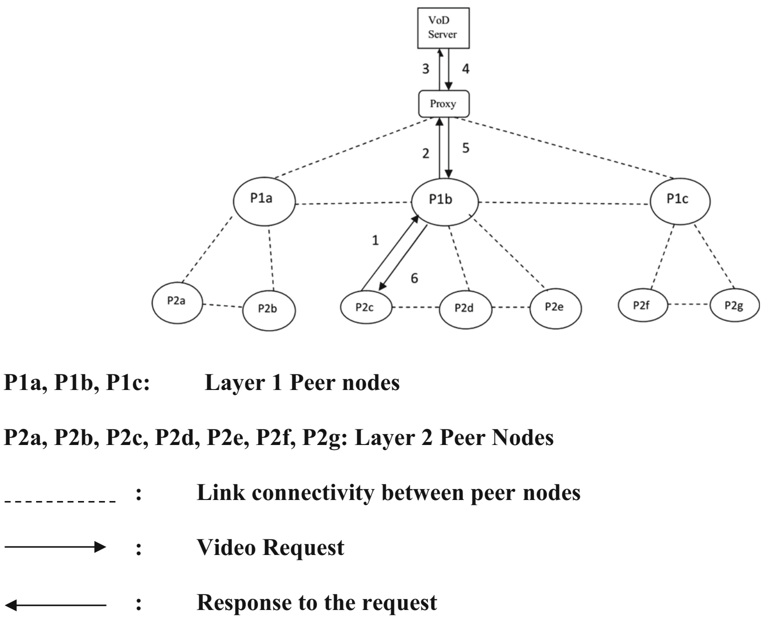


Fig. 2 Hierarchical network

Steps for Hierarchical Prefetching Mechanism using Proxy

1. The VoD server fragments the video into several segments, each fragment of length k.
2. The peer nodes joining the network in the initial stage in terms of time; form the top most layer i.e., the first layer around the server.
3. Peer nodes joining in the next stage, form the second layer.
4. The interval between two consecutive stages will be k.
5. Similarly, the peer nodes joining in the later stages of time will form the consecutive layer.
6. Each node will maintain a record of the information related to its parent node, sibling nodes, and child nodes.
7. In order to reduce the workload on the VoD server, the proxy is used [15] and located at the prime locations of the network.
8. Proxy maintains a cached copy of the videos based on popularity as described in Section 3.3.
9. When a peer node in the subsequent layers needs a video, then it will request its parent node based on the information recorded about to its surrounding nodes; instead of requesting the VoD server.
10. On receiving the request, the parent node checks if the proxy has the requested video.
11. If the proxy node has the requested video, then the parent node will redirect the request to the proxy.
12. The parent peer node caches the first initial 'k' segments of video from the proxy, and transfers it to the requesting child node and doesn't replace this part until it has been completely transferred.
13. Then the parent node caches the next k segments of the video and transfers it to the requesting node.
14. This is performed until the requested video is delivered completely to the requesting child node.
15. If the requested video is not available at the parent node and proxy, then the request is redirected to the sibling nodes and the parent node.
16. If the requested video is available at any of the redirected peers, then it is cached and delivered to the requesting node.
17. If the requested video is not available at any of the sibling node and parent node, then the request is redirected to the VoD server.
18. The VoD server provides the requested video and it is delivered to the requesting node through the proxy.
19. The proxy maintains the initial part of the video in the cache.
20. Later, if this video is requested by any other node, then it is delivered else if the video is not requested by any node for a long time interval, then it is deleted and updated in the proxy.

3.3 Popularity based proxy caching

The proxy caches the video based on popularity. The popularity differentiates the video in high demand from the video in low demand. Popularity is measured on the basis of the data replicated. The video that is replicated many times is considered as highly popular and the video that is poorly replicated is considered as less popular. Proxy measures the popularity of each video through a decentralized approach [16]. The popularity estimation process is described in algorithm 2.

Steps for Popularity based proxy caching

1. For the set of videos $V = \{ v_1, v_2, \ldots, v_n\}$ in the proxy, it broadcasts a "popularity query" for each video, v_i with a predefined Time-to-Live (TTL) counter.
2. This query hops from one node to another, and with each hop, the TTL value reduces by 1.
3. Every occurrence of the video, v_i at each node, is recorded at the proxy.
4. When the TTL counter value reduces to zero, the popularity query expires.
5. Then the proxy measures the occurrence recorded for each video in the surrounding nodes and compares it with the predefined threshold value which defines the average number of occurrences of videos.
6. If the occurrence > threshold, then the video is popular and maintained in the cache.
7. If the occurrence < threshold, then the video is not popular and deleted from the cache.

4 Simulation results

4.1 Simulation settings

In order to test our protocol, NS-2 [12] simulator version 2.29 is used. We have used the BitTorrent packet-level simulator for P2P networks. A network topology is only used for the packet-level simulator. The simulation topology is given in Fig. 3 and the simulation settings are summarized in Table 1. In the topology we have considered 12 peer nodes and the both nodes 0 and 1 are considered as VoD server and proxy respectively.

As we can see from the figure, we have a video server as the root of the hierarchical tree and a proxy is connected to the server. Peer nodes are attached to the proxy on the first level and the bottom most level contains the receivers nodes.

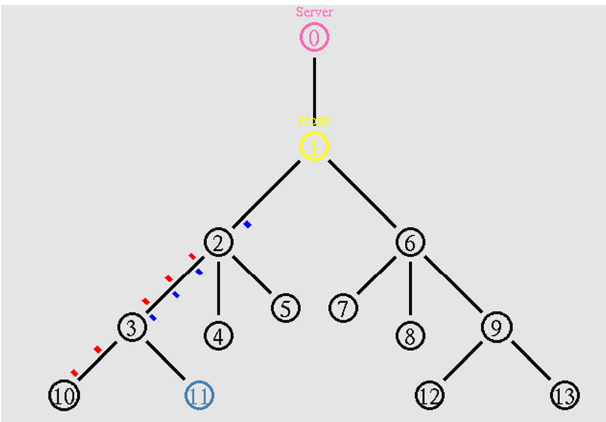


Fig. 3 Simulation topology

Table 1 Simulation parameters

No. of nodes	14
Link bandwidth	2 Mb
Link delay	20 ms
No. of receivers	2
Traffic source	CBR
Traffic rate	100 kb to 500 kb
Packet size	500 bytes
Cache size	50 to 250 bytes
Video trace	JurassikH263-256 k
Simulation time	50 s

4.2 Performance metrics

In our experiment, we measure the following metrics:

Received Bandwidth: It is the number of bits transmitted per second.

Average end-to-end delay: The end-to-end delay is averaged over all surviving data packets from the sources to the destinations.

Throughput: It is the number of packets received successfully at each receiver.

The proposed Hierarchical caching Technique (HCT) is compared with "PopCap" technique [15], "Cooching" technique [1] and EWMA [9] based prefetching.

4.3 Results

4.3.1 Scenario-1

Initially, at 1.0 s, a request for the specific video is sent from 3 to the Proxy. At the 2.5 s, node 10 also request for the same video which can be obtained from the cache of node 3.

Based on cache size In order to test the performance of caching, we increase the cache size from 50 to 250 bytes and measured the above metrics.

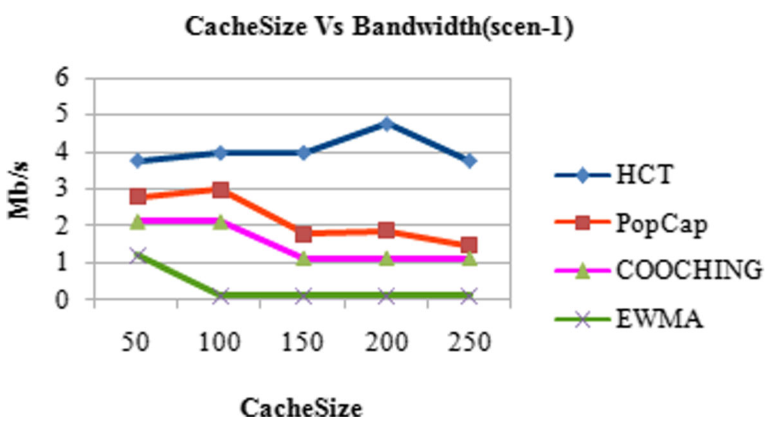


Fig. 4 Cache size vs Bandwidth

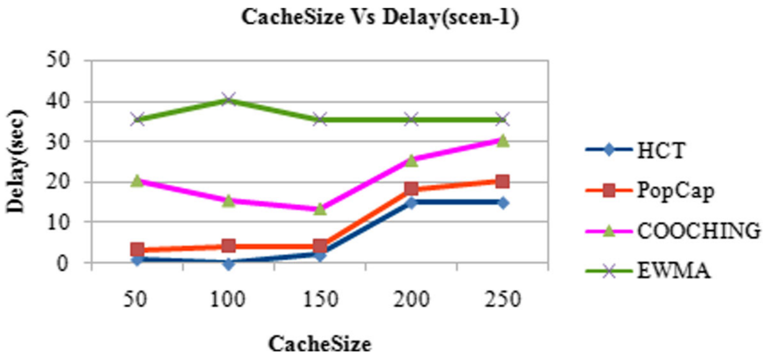


Fig. 5 Cache size vs Delay

The Figs. 4, 5 and 6 give the performance of both the protocols for scenario-1 when the cache size is increased. We can see that the proposed HCT outperforms the PopCap, Cooching and EWMA in all the figures.

Figures 4, 5 and 6 show the results of bandwidth, delay and throughput by varying the cache size from 50 to 250 for the CBR traffic in PopCap, Cooching, EWMA and HCT protocols. Figure 4 shows the comparison of bandwidth of our proposed approach with the existing works. Because of our proposed popularity based proxy caching, bandwidth of our proposed approach is improved 95%, 98% and 98.7% than that of PopCap, Cooching and EWMA respectively. Comparison of delay of our proposed approach with the existing works is shown in Fig. 5. Due to given the priority for popularity based videos, the prefetching time or delay is reduced to 49%, 56% and 94% than that of PopCap, Cooching and EWMA. Figure 6 shows the comparison of throughput of our proposed approach with the existing works. Compared to PopCap, Cooching and EWMA, throughput of our proposed approach improved 95%, 97% and 98.2% respectively. Because of our proposed popularity based proxy caching, bandwidth utilization and throughput of the network are improved and prefetching time is also reduced.

Based on exponential traffic rate In order to test the performance of buffering and streaming, we increase the traffic sending rate from 100Kb to 500Kb.

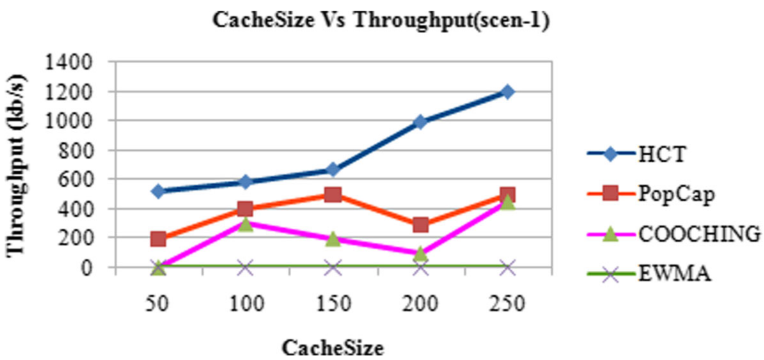


Fig. 6 Cache size vs Throughput

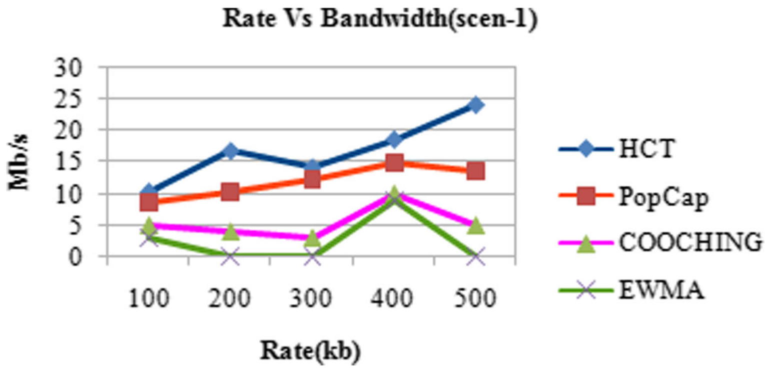


Fig. 7 Rate vs Bandwidth

The Figs. 7, 8 and 9 give the performance of both the protocols for scenario-1 when the traffic rate is increased. We can see that the proposed HCT outperforms the PopCap, Cooching and EWMA in all the figures.

Figures 7, 8 and 9 show the results of bandwidth, delay and throughput by varying the rate from 100 to 500Kb for the CBR traffic in PopCap, Cooching, EWMA and HCT protocols. Figure 4 shows the comparison of bandwidth of our proposed approach with the existing works for varying rates. Bandwidth of our proposed approach is improved 49%, 54% and 68% than that of PopCap, Cooching and EWMA respectively. Comparison of delay of our proposed approach with the existing works for varying rates is shown in Fig. 8. Delay of our proposed approach is reduced to 95%, 98% and 98.6% than that of PopCap, Cooching and EWMA. Figure 9 shows the comparison of throughput of our proposed approach with the existing works for varying rates. Compared to PopCap, Cooching and EWMA, throughput of our proposed approach improved 95%, 97% and 97.8% respectively.

4.3.2 Scenario-2

In this scenario, nodes 7 and 8 send a request for specific video at 1.5 s. At 2.5 s node 12 also request for the same video which can be obtained from the cache of node 6.

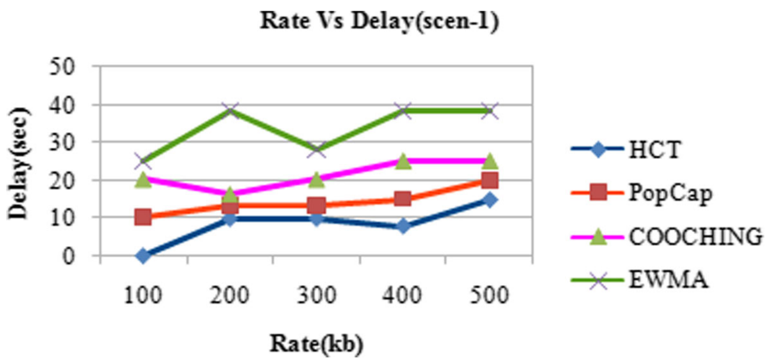


Fig. 8 Rate vs Delay

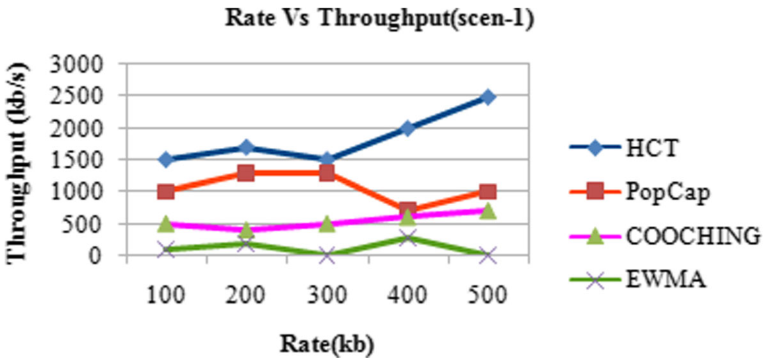


Fig. 9 Rate vs Throughput

Based on cache size In order to test the performance of caching, we increase the cache size from 50 to 250 bytes and the performance is evaluated. When the proxy cache size is small, PopCop outperforms HCT. This is because when the benefit of proxy caching is not significant, by using the accurate information on the number of video replicas cached by the P2P network, PopCop can make a better use of the peers' cache space than HCT. However, with the increase of the proxy cache size, HCT outperforms PopCop. This is because HCT makes better use of the proxy cache than PopCop, and more importantly, peers under HCT can cooperate better with the proxy by avoiding caching videos that are also likely to be cached by the proxy. Also our proposed work is also compared with Cooching [1] and EWMA [10] based prefetching. Cooching maximized the throughput of the network and EWMA based prefetching technique not only improved the performance of the prefetching strategy but also reduced the server load. So we compared these three existing work with our proposed work to prove that this work is superior to the previous works.

The Figs. 10, 11 and 12 give the performance of both the protocols for scenario-2 when the cache size is increased. We can see that the proposed HCT outperforms the PopCap, Cooching and EWMA in all the figures.

Figures 10, 11 and 12 show the results of bandwidth, delay and throughput by varying the cache size from 50 to 250 for the CBR traffic in PopCap, Cooching, EWMA and HCT

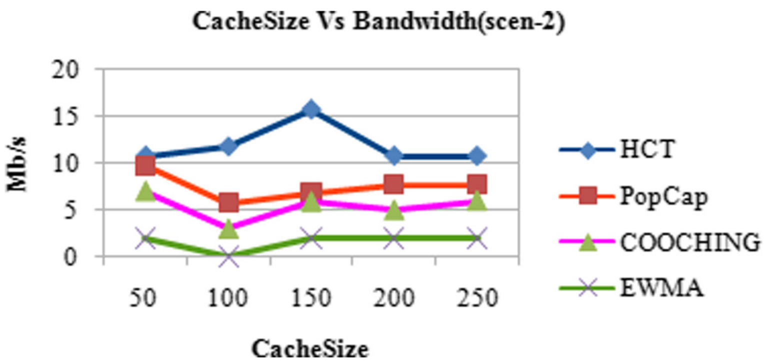


Fig. 10 Cache size vs Bandwidth

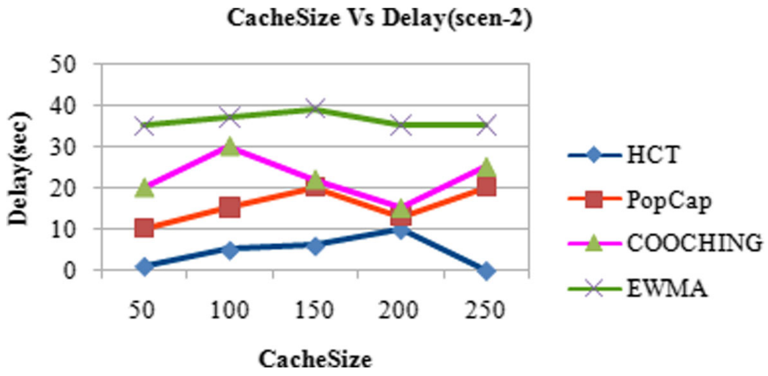


Fig. 11 Cache size vs Delay

protocols. Figure 10 shows the comparison of bandwidth of our proposed approach with the existing works for varying cache size. Bandwidth of our proposed approach is improved 99%, 99.3% and 99.6% than that of PopCap, Cooching and EWMA respectively. Comparison of delay of our proposed approach with the existing works for varying cache size is shown in Fig. 11. Delay of our proposed approach is reduced to 64%, 73% and 84% than that of PopCap, Cooching and EWMA. Figure 12 shows the comparison of throughput of our proposed approach with the existing works for varying cache size. Compared to PopCap, Cooching and EWMA, throughput of our proposed approach improved 64%, 82% and 93% respectively.

Based on rate In order to test the performance of buffering and streaming, we increase the traffic sending rate from 100Kb to 500Kb.

The Figs. 13, 14 and 15 give the performance of both the protocols for scenario-2 when the traffic rate is increased. We can see that the proposed HCT outperforms the PopCap, Cooching and EWMA in all the figures.

Figures 13, 14 and 15 show the results of bandwidth, delay and throughput by varying the rate from 100 to 500Kb for the CBR traffic in PopCap, Cooching, EWMA and HCT protocols. Figure 13 shows the comparison of bandwidth of our proposed approach with the existing

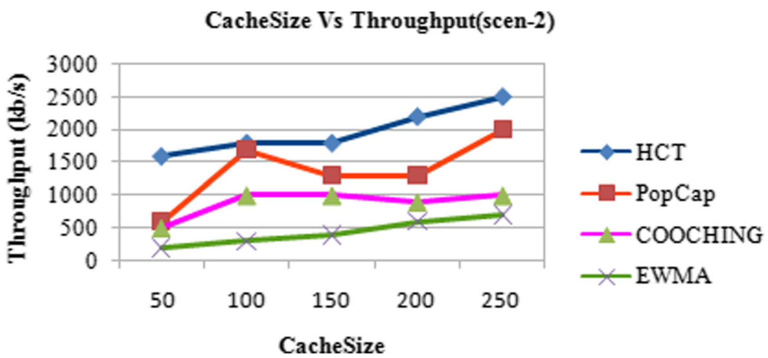


Fig. 12 Cache size vs Throughput

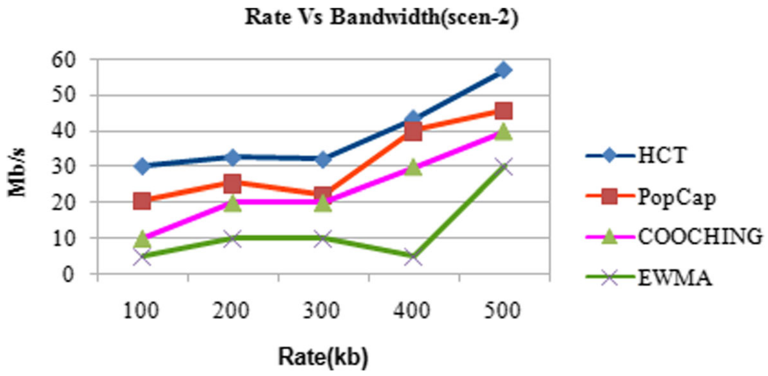


Fig. 13 Rate vs Bandwidth

works for varying rates. Bandwidth of our proposed approach is improved 99%, 99.6% and 99.8% than that of PopCap, Cooching and EWMA respectively. Comparison of delay of our proposed approach with the existing works for varying rates is shown in Fig. 14. Delay of our proposed approach is reduced to 64%, 83% and 95% than that of PopCap, Cooching and EWMA respectively. Figure 15 shows the comparison of throughput of our proposed approach with the existing works for varying rates. Compared to PopCap, Cooching and EWMA, throughput of our proposed approach improved 64%, 72% and 86% respectively.

5 Conclusion

In this paper, we have proposed popularity based hierarchical prefetching protocol. In this protocol, initially, the network is designed in the form of layers, with the nodes joining the network in the initial time intervals being considered as the first layer nodes. Nodes joining the network in the consecutive intervals of time are considered as considered as consecutive layers. Then the proxy located at the prime network locations estimate the video popularity and the highly popular videos are cached at the proxy. Finally, when a node needs a video, it requests its parent

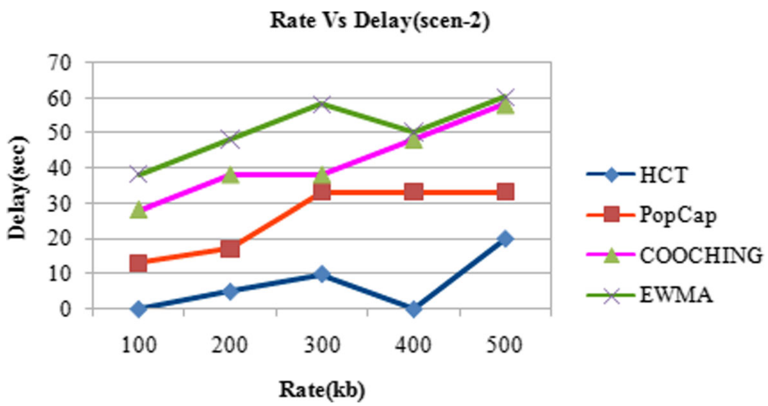


Fig. 14 Rate vs Delay

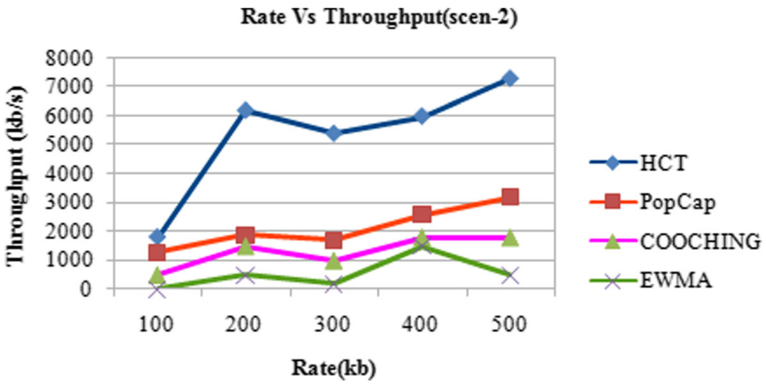


Fig. 15 Rate vs Throughput

node. The parent node satisfies its request with the help of the proxy. Thus, the network needs are fulfilled without burdening the VoD server. As a future work, data in the peers will be scheduled using an optimization algorithm in the hierarchical form of P2P network.

References

1. Abbasi U, Ahmed T (2010) Architecture for cooperative prefetching in P2P video-on- demand system. Int J Comput Netw Commun (IJCNC) 2(3)
2. Arulkumar CV, Jeyakumar K, Malarmathi M, Shanmugapriya T (2012) Secure communication in unstructured P2P networks based on reputation management and self certification. Int J Comput Appl (0975–8887) 44(15)
3. Chen Y-W, Huang Y-H (2011) An interactive streaming service over peer-to-peer networks. International conference on software and computer applications, Singapore
4. Cheng W, Tan Z (2012) Correlation trust authentication model for peer-to-peer networks. Proceedings of the 2012 2nd international conference on computer and information application
5. Graffi K, Kaune S, Pussep K, Kovacevic A, Steinmetz R (2008) Load balancing for multimedia streaming in heterogeneous peer-to-peer systems. NOSSDA, Braunschweig
6. Hefeeda M, Saleh O (2008) Traffic modeling and proportional partial caching for peer-to-peer systems. IEEE/ACM Trans Networking 16(6)
7. Kao H-H et al (2016) I2CC: interleaving two-level cache with network coding in peer-to-peer vod system. J Netw Comput Appl 60:180–191 Web
8. Kim E, Liu JCL (2017) An Integrated Prefetching/Caching Scheme In Multimedia Servers. Journal of Network and Computer Applications 88:112–123 Web
9. Liu P et al (2013) Server load based prefetching strategy for P2P VoD streaming. Computer Science and Network Technology (ICCSNT), 2013 3rd international conference on. IEE
10. Liu P et al (2013) Server load based prefetching strategy for P2P VoD streaming. Computer Science and Network Technology (ICCSNT), 2013 3rd international conference on. IEEE
11. Mushtaq M, Ahmed T (2007) Hybrid overlay networks management for real-time multimedia streaming over P2P networks. IFIP International Federation for Information Processing, pp 1-13
12. Network Simulator: http:///www.isi.edu/nsnam/ns
13. Nikolaos E, Athanasios C, Spyros D, Odysseas K (2008) L-CAN: locality aware structured overlay for P2P live streaming. Springer, Berlin Heidelberg
14. Takeda A, Chakraborty D, Kitagata G, Hashimoto K, Shiratori N (2008) A new scalable distributed authentication for P2P network and its performance evaluation, 12th WSEAS international conference on COMPUTERS, Heraklion, Greece, July 23-25
15. Tian Y, Liu B, He Z (2010) PopCap: popularity oriented proxy caching for peer-assisted internet video-on-demand streaming services. Front Comput Sci China 4(4):500–515
16. Wang Q, Daudjee K, Tamer Özsu M (2008) Popularity-aware prefetch in P2P range caching, Peer-to-Peer network applications. Springer

17. Xu T, Wang W, Ye B, Li W, Lu S, Gao Y (2009) Prediction-based prefetching to support VCR-like operations in gossip-based P2P VoD systems, 15th international conference on parallel and distributed systems
18. Xu C, Muntean G-M, Fallon E, Hanley A (2009) Distributed storage-assisted data-driven overlay network for P2P VoD services. IEEE Trans Broadcast 55(1)
19. Yuji Y, Fujita S (2013) Swarm architecture toward P2P VoD without playback suspension. International conference on parallel and distributed processing techniques and applications (PDPTA), Las Vegas, USA

Sivakumar Ganapathi obtained his Bachelor's degree in Computer Science and Engineering from Bharathiar University, Coimbatore, India. Then he obtained his Master's degree in Computer Science and Engineering from Anna University Coimbatore, India. Currently, he is a Associate Professor in the Department of Computer Science and Engineering, Erode Sengunthar Engineering College, Erode District, Tamilnadu, India. He is a Part time research scholar of Anna University, Chennai, Tamilnadu, India. His specializations include Video Streaming, Open Source based Web Technology and Big Data Analytics. His current research interests are P2P Video Streaming, Hadoop Architecture and Big Data Analytics methods.

Dr. Venkatachalam Varadharajan obtained his Bachelor's degree in Electronic and Communication Engineering from Bharathiar University, Coimbatore, the M.S. in Computer Science from BITS, Pilani, India, the Master's degree in Computer Science and Engineering from National Institute of Technology (formerly Regional Engineering College), Trichy, and the Ph.D. degree from Anna University Chennai. Currently, he is a Professor of Computer Science and Engineering Department and Principal of Erode Sengunthar Engineering College, Erode Dt., Tamilnadu, India. His area research includes Networking, Network Security (Intrusion Detection System), Video Streaming, Image Processing, Data Mining. Currently, he is guiding 12 Ph.D. Research Scholars.