International Journal of Computer Applications (0975 – 8887)
Volume 53– No.6, September 2012

QoS- Continuous Live Media Streaming in Mobile Environment using VBR and Edge Network

V. Saravanan
Asst.Professor in Computer Applications
Hindusthan College of arts and Science, Coimbatore

C. Chandrasekar, Ph.D
Associate Professor
Dept.of Computer Science Periyar University, Salem

ABSTRACT

Live media services in mobile environments are fetching more essential with the explosion of technologies. Live Media streaming, in particular, is a promising technology for providing services such as live news clips, live sports. To avoid service interruption when the users keep moving, proper data management strategies must be engaged. In an earlier paper, the ideas and benefits of two-level cooperative media streaming with headlight pre fetching and dynamic chaining were demonstrated and they using static bit rate delivery. In this paper, we propose the QoS-based dynamic adaptation techniques for the flexible employment and smooth integration of headlight pre-fetching and dynamic chaining to continuously provide quality streaming services to mobile users. And also it provides Variable bit rate delivery. We propose a new mechanism Upload Cache in Edge Networks brings benefit for both end users and service providers. For end users, it shortens the duration while user must stay online for uploading their generated content. Also for service providers, it reduces peak traffic volume between edge networks and data centers by slightly shifting the upload timing without incurring much extra latency overhead added. Our analysis with replaying the captured traffic shows that this mechanism reduces upload tether time of 24% end users by more than half and flattens the traffic peak for the access service provider by 37%. To avoid disconnection and/or service breakdown when the users keep moving, proper data management strategies must be taken by all parties. We propose a two level framework and a set of new techniques for cooperative media streaming in mobile environments.

General Terms
Dynamic headlight pre-fetching, P2P media streaming, QoS-aware adaptation, variable bit rate, caching system, traffic shaping

Keywords
QoS-aware adaptation, variable bit rate, caching system, traffic shaping

1. INTRODUCTION

Media streaming is a promising technology for providing multimedia information services such as news clips, live sports, and hot movies in mobile environments. Effective data management for media streaming is naturally the key to the successfulness of such services. Since many users may request the same media, traditional client server model can easily result in server bottleneck, bandwidth waste, poor cache utilization and longer delay. Furthermore, the mobility of mobile users raises the issue of seamless service hand off. To avoid service interruption, proper data management strategies must be employed. And For media streaming applications, such a harsh environment often leads to undesirable disconnections and playback interruptions. The continuous playing requirement of streaming Medias only adds to the difficulty of the situation. To remedy these problems, it is usually necessary to incorporate certain degree of pre-fetching and caching to transmit the media segments before they are requested. However, user mobility significantly complicates the problem since the exact locations and targets for effective pre-fetching and caching are very difficult to determine reliably. In an earlier paper, the ideas and benefits of two-level cooperative media streaming with headlight pre-fetching and dynamic chaining were successfully demonstrated.

Headlight pre-fetching facilitates the cooperation of streaming access points (SAPs) to cope with the uncertainty of user movement. For each mobile user, a virtual fan-shaped headlight pre-fetching zone is maintained along the direction of movement. The overlapping area of the headlight zone on a particular cell and the accumulated virtual illuminance determine the degree and volume of pre-fetching. Dynamic chaining is for the cooperation among mobile users to maximize cache utilization and streaming benefit. On a media-segment request, a search for a supplying partner, who happens to be viewing the same media, is instantiated. Then, the requesting user can be chained to the supplying partner to receive subsequent segments directly without further intervention. The user can itself be a supplying partner for other users and naturally form a chain of users that are viewing and sharing the same media. As complementary techniques, headlight pre-fetching and dynamic chaining are quite effective for mobile media streaming. However, they operate in session-wide static and distinctive modes. All pre-fetching-related settings are fixed until the next round of pre-fetching. Media segments are supplied either by chaining from a peer or by pre-fetching from the SAP. Moreover, the users do not have control over the cost and quality levels of the streaming services. The performance of both headlight pre-fetching and dynamic chaining can drop significantly under fast changing moving patterns.

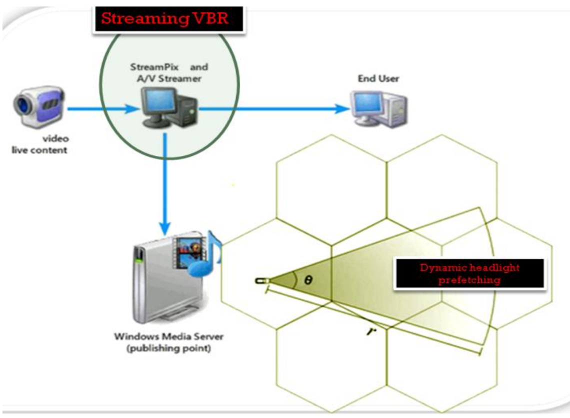


Fig. 1. Proposed Process

QoS is about provisioning network services to the level of guarantee required by the application layer services.In this paper, we propose QoS-based dynamic adaptation techniques for the flexible employment and smooth integration of headlight pre-fetching and dynamic chaining to continuously provide quality-streaming services to mobile users. To cope with the deficiencies of previous techniques, QoS-aware dynamic headlight pre-fetching is proposed to dynamically adjust the pre-fetching scheme in response to the fast changing or unstable moving patterns. Specifically, key QoS parameters and cost factors are identified and formulated to characterize the current status of the pre-fetching and streaming services. By continuously monitoring the QoS and cost variations, we can identify the exact timing and targets for adaptation.

Adaptive P2P media streaming, on the other hand, is for the cooperation between mobile users such that multiple streaming sources from different partners can be used to increase the likelihood of successful chaining. More specifically, instead of solely relying on one supplying partner for media segments as in previous approach, we maintain a supplying partner set which consists of relatively stable peers who can provide the needed segments. The streaming service is then split among the supplying partners to speed up download time and improve stability at the same time.

In addition to the adaptive-streaming techniques, a QoS based scheme is developed to dynamically trigger and proportionally adjust the pre-fetching degree when the stability and quality of P2P streaming service vary. The technique facilitates smooth integration of headlight pre-fetching and dynamic chaining to continuously provide quality-streaming services to mobile users. Extensive simulation and performance evaluation results demonstrate that the proposed QoS-based dynamic-adaptation techniques significantly improve the service quality and streaming performance of cooperative media streaming in mobile environments.

The Internet has been originally designed for the model of host-to-host data communication. However, people have observed apparent trends of content domination in the current Internet usage. Research efforts have been put into content retrieval in the Internet, which dates back to the web proxy caches that have been comprehensively studied since decades ago. Other popular topics in this field include network accelerator content distribution networks (CDN) and so on. Recently, researchers are exploring the redesign of the Internet communication model to follow the content domination trend, and their achievements are generally called content oriented network architectures. In the light of this observation, we propose upload cache in edge networks, a new mechanism assisting upload of UGC of end users within edge networks. In this mechanism, an end user will not directly upload its pieces of content to the remote servers, but put them to a gateway server located within the same edge network with the user. The gateway caches the received content and schedules its delayed upload to the destination servers without incurring much extra delay in the total elapsed time for the upload, without the involvement of the end user in the latter step. Deploying upload cache in edge networks brings benefit for both end users and service providers. First, for end users, it shortens the duration while user must stay online for uploading their generated content, thanks to the short round-trip time (RTT) and ample bandwidth between users and gateway. In addition, from the service provider point of view, traffic peak of the edge networks or the destination servers could be flattened because the inter-network upload from the gateways to the destination servers may be scheduled at later time. Finally, the cached piece of uploaded content may be reused for populating download cache so that even the traffic of the first content retrieval of the edge networks may be saved. We recognize that a similar feature of proposed architecture could be achieved by Delay-Tolerant Networking (DTN) Architecture.

2. Headlight Pre-fetching and Dynamic Chaining

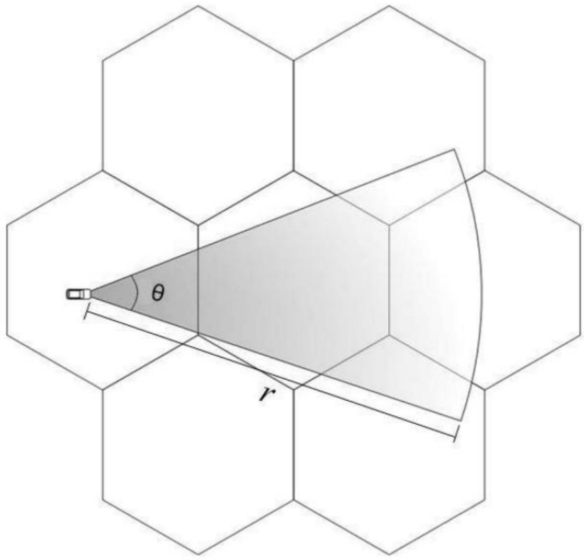


Fig. 1(a). The headlight pre-fetching zone

This paper is self-contained and it has provided a diminutive analysis of the key techniques introduced in the service architecture consists of streaming media servers (SMSs), SAPs, and mobile users. The SAP-level cooperation is to assist the headlight pre-fetching for the proper data management and seamless handoff. Dynamic-chaining methods form user level P2P cooperation to share media segments with each others. Headlight pre-fetching is to maintain a headlight pre-fetching zone as shown in Fig. 1 along the direction of user movement to serve as a prediction of possible future interaction with neighboring cells, and also it helps in providing predictive pre-fetching services. The radius r establishes the extent of look ahead. The angle θ reins the period of coverage. All SAPs of the cells overlapped with the zone are selected as the pre-fetching SAPs. The SAPs farther away are assigned fewer segments to save the cost. The SAPs closer by allocated with more segments to prevent undesirable interruption. The accumulated virtual illuminance on a cell is used to determine the segment assignment. To deal with changing mobility patterns, headlight shifting and headlight sharing are used to improve the level of cooperation among SAPs. Headlight pre-fetching is shown to significantly improve the performance of streaming services in comparison with on-demand services based on simple look ahead pre-fetching. However, in the face of fast changing or highly unstable moving patterns, headlight pre-fetching may suffer from sudden increase in playback interruptions and segment download time. Dynamic chaining is to explore client storage and user level cooperation to facilitate seamless streaming services even when disconnected from the SAP. On a segment request, a search for a supplying partner among the neighboring peers viewing the same media is initiated. Once located, the requesting client is chained after the found peer to receive the streaming segments in a P2P fashion. The viewed segments can be shared with other peers in similar way to form chains of peers. Since the cost of the user level P2P streaming is much lower than the direct service from the SAP, dynamic chaining provides even more saving when applied with headlight pre-fetching. However, a client is chained to only one supplier which may be unstable. Even with belt sharing to quickly identify an alternative supplier, frequent breakage of the chain can still incur excessive overhead. We propose techniques to overcome the deficiencies discussed above. In particular, we introduce QoS and adaptive processing concepts for improving the reliability of pre-fetching as well as increasing the chance of successful chaining.

3. VARIABLE BIT RATE

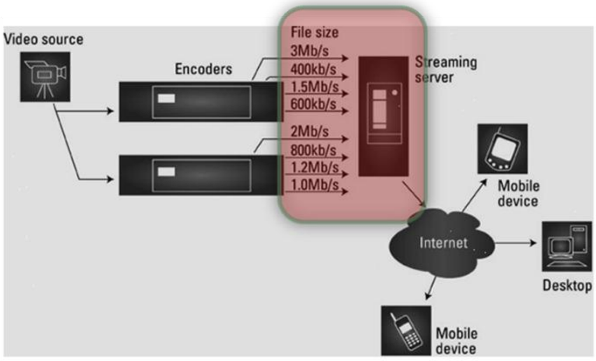


Fig. 1(b). Process VBR Result

Variable bitrate (VBR) is a term used in telecommunications and computing that relates to the bitrate used in sound or video encoding. As opposed to constant bitrate (CBR), VBR files vary the amount of output data per time segment. VBR allows a higher bitrate (and therefore more storage space) to be allocated to the more complex segments of media files while less space is allocated to less complex segments. The average of these rates can be calculated to produce an average bitrate for the file.

MP3, WMA, Vorbis, and AAC audio files can optionally be encoded in VBR. Variable bit rate encoding is also commonly used on MPEG-2 video, MPEG-4 Part 2 video (Xvid, DivX, etc.), MPEG-4 Part 10/H.264 video, Theora, Dirac and other video compression formats. Additionally, variable rate encoding is inherent in lossless compression schemes such as FLAC and Apple Lossless.

VBR are that it produces a better quality-to-space ratio compared to a CBR file of the same data. The bits available are used more flexibly to encode the sound or video data more accurately, with fewer bits used in less demanding passages and more bits used in difficult-to-encode passages.

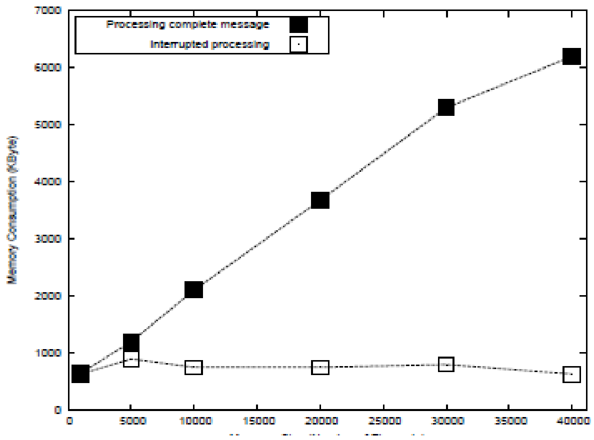


Fig.1(b.1).ProcessVBR

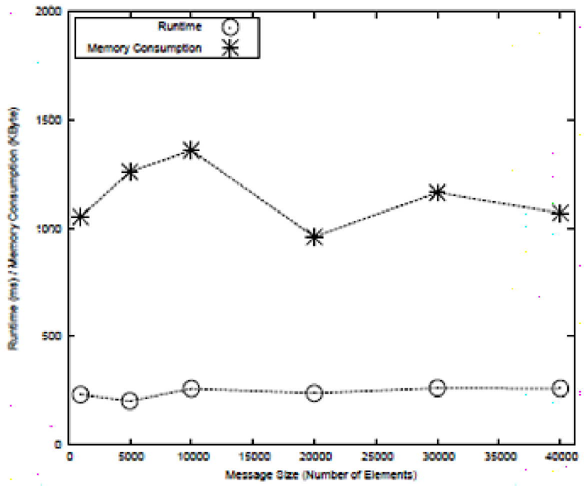


Fig. 1(b.2). Process VBR Result

4. Variable Bit Rate Compared with CBR

Constant Bit Rate encoding (CBR) and Variable Bit Rate encoding (VBR) are two techniques for controlling the bit stream of the compressed video file. Simply stated, encoding via CBR produces a file that has a constant bit rate throughout. In contrast, encoding via VBR varies the bit stream according to the complexity of the video file, while achieving the same average data rate as CBR.

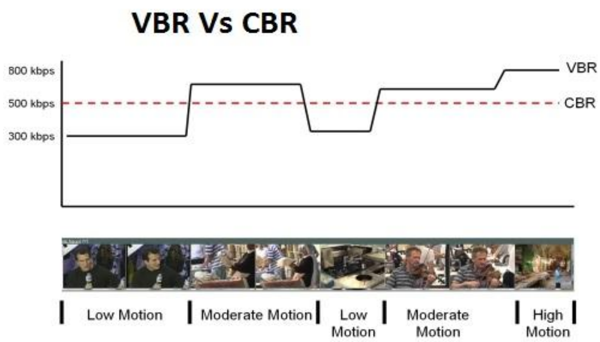
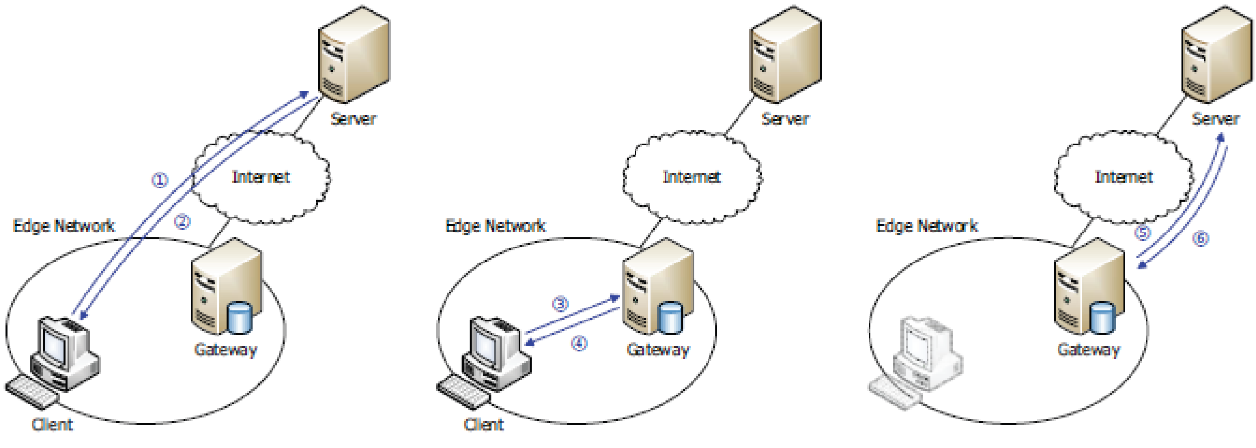


Figure 1(b-3). Constant and Variable bit rate

In Figure 1(b-3), which illustrates a file with low motion, easy to compress scenes, and high motion, hard to compress scenes? Both techniques achieve the same average data rate over the duration of the file, but the red CBR line stays constant throughout, while the black VBR line varies with the amount of motion in the scene. In general, VBR should produce a higher quality file than CBR because it allocates file data rate as necessary to maximize quality. The downside is stream variability, since the per-second bit rate can vary significantly from section to section. If you were producing video for a tight, relatively low bit rate connection in a cell phone, this variability could interrupt playback. Also, when distributing via a streaming server, which meters out the video to the remote viewers as needed, the consistent data rate of a CBR encoded file is easier to administrate. For this reason, the generally accepted best practice is to use CBR when producing for streaming delivery, and VBR when producing for progressive download.

shown in Fig. 2(b). Besides the actual data of the content, the client will also transfer both the destination of the upload and the token received from the destination server to the gateway. However, no privacy information, such as end user's account or password on the destination server, is provided to the access service provider. Finally, the gateway verifies the token with the destination server. Once the indirect upload is acknowledged by the server, the gateway uploads the temporary stored data of the content to the server. This step, as shown in Fig. 2(c), doesn't involve the client at all, so the client can leave the network as soon as it finishes uploading the content to the gateway.



(a) client requests for token from server (b) client upload token and content to gateway (c) gateway upload content to server

Fig 2: edge networks

5. Edge Network and upload cache

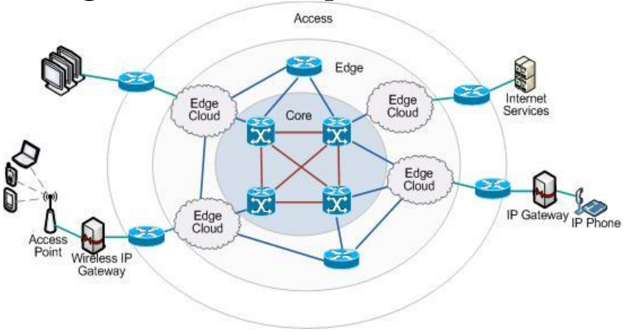


Fig. 2(a). Edge Network Process

Fig. 2 demonstrates the Edge Network. The essential in sequence on the intention server and the gateway, the client can initiate a typical upload process with upload cache in edge networks, as illustrated in Fig. 2. First, as shown in Fig. 2(a), the client requests for an oblique upload indication for a piece of content from the destination server. The server authenticates the privilege of user uploading such content and provides the token to the client that can be used for indirect upload only once and for the specific content.

When the client has received the token, it communicates with the gateway to upload the piece of content to the gateway, as

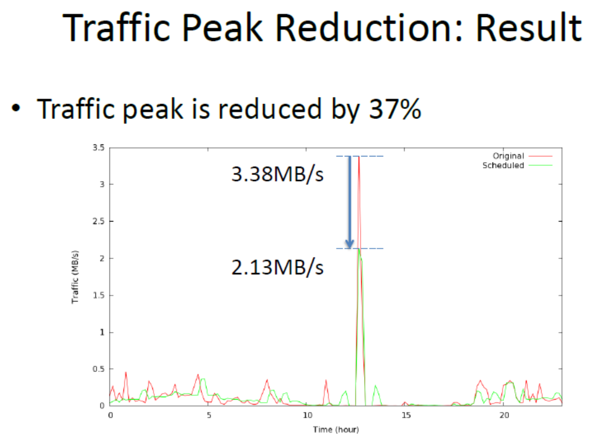


Fig. 2(b). Cache Result

6. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the work.

7. CONCLUSION

In this paper, we have presented QoS-based dynamic adaptation techniques to extensively improve the responsiveness of n-level cooperative streaming service with headlight pre-fetching, VBR and Edge Network, upload cache and P2P chaining. In the face of fast changing moving patterns, the techniques achieve their effectiveness by continuous monitoring of QoS and processing cost, and then adjust the streaming service strategies accordingly. In this smooth integration of the n-levels of cooperation provides even higher quality of services to mobile users at lower cost. A new mechanism – upload cache in edge networks, which benefits both end users and service providers. For end users, it significantly shortens the duration while user must stay online for uploading their generated content. For service providers, it flattens the traffic peak for the edge networks or the destination servers. Such caching functionality is achieved by indirectly uploading to the gateways in edge networks with the token retrieved from destination servers and leaving gateways to complete the actual upload to the destination servers. The gateway schedules its delayed upload to the destination servers without incurring much extra delay in the total elapsed time for the upload, and without the involvement of the end users. The gateway also delegates the destination server for serving the uploaded content in the edge network. Our evaluation shows Good result comparing past methods.

8. REFERENCES

[1] G. Anastasi, M. Conti, E. Gregori, A. Passarella, and L. Pelusi, “An Energy-Efficient Protocol for Multimedia Streaming in a Mobile Environment,” J. Pervasive Computing and Comm., vol. 1, no. 4, pp. 301-312, Dec. 2005.

[2] D.A. Tran, K. Hua, and T. Do, “A Peer-to-Peer Architecture for Media Streaming,” IEEE J. Selected Areas in Comm., Special Issue on Advances in Service Overlay Networks, vol. 22, no. 1, pp. 121-133, Jan. 2004.

[3] Data Streams: Models and Algorithms, C.C. Aggarwal, ed. Springer Verlag, 2007.

[4] T. Kwon, Y. Choi, C. Bisdikian, and M. Naghshineh, “QoS Provisioning in Wireless/Mobile Multimedia Networks Using an Adaptive Framework,” Wireless Networks, vol. 9, no. 1, pp. 51-59, Jan. 2003.

[5] B. Li and K.H. Wang, “NonStop: Continuous Multimedia Streaming in Wireless Ad Hoc Networks with Node obility,” IEEE J. Selected Areas in Comm., vol. 21, no. 10, pp. 1627-1641, Dec. 2003.

[6] Kunavut, Kunagorn and Teerapat Sanguankotchakorn. “Performance Evaluation of Ad Hoc Routing Protocols to Deliver MPEG-4 Traffic.” *Proceedings of the 12th IEEE International Conference on Communication and Technology (IEEE ICCT 2010)*. Nanjing, China. November 11-14, 2010, pp. 207-210.

[7] M L Sharma,Noor Fatima Rizvi,Nipun Sharma,Anu Malhan,Swati Sharma, “Performance Evaluation of MANET Routing Protocols under CBR and FTP traffic classes”. *International Journal of Computer Technology and Applications*. Vol 2 (3), 392-400

[8] W. Navidi and T. Camp, “Stationary Distribution for the Random Waypoint Mobility Model,” *Technical Report MCS*-03-04, *IEEE Transactions on Mobile Computing*, Vol. 3, No. 1, 2004. pp. 99-108.

[9] D. Bhavyesh, A. Ajith, G. Crina and S. Sugata, “Impact of Node Mobility on MANET Routing Protocols Mod-els,” *Journal of Digital Information Management*, Vol. 5, No 1, 2007, pp. 19-24.

[10] Chu, Y-H., Chuang, J., and Zhang, H. (2004), ‘A Case for Taxation in Peer-to-Peer Streaming Broadcast’, ACM SIGCOMM Workshop on Practice and Theory of Incentives and Game Theory in Networked Systems (PINS), Portland, OR, August, 2004.

[11] Banerjee, S., Bhattacherjee, B. and Komareddy, C. (2002), ‘Scalable Application Layer Multicast’, In Proceedings of ACM SIGCOMM, 2002, Pittsburgh, USA, 2002, pp.205–217.