Peer-Assisted Caching for Scalable Media Streaming in Wireless Backhaul Networks

Hazem Gomaa Geoffrey G. Messier Robert Davies
Department of Electrical and Computer Engineering
University of Calgary, Calgary, AB, Canada
Email: {hagomaa,gmessier,davies}@ucalgary.ca

Carey Williamson
Department of Computer Science
University of Calgary, Calgary, AB, Canada
Email: carey@cpsc.ucalgary.ca

***Abstract*—This paper presents a method for supporting wireless media streaming using a cache that is distributed across the mobile devices in the network. The performance of this scheme is compared to traditional institutional server (IS) caching on a network with a bandwidth constrained wireless backhaul. In addition to traditional caching hit ratio metrics, the paper studies how caching affects the call drop ratio due to limited backhaul bandwidth. These results indicate that the distributed caching method provides better service than IS caching as the number of users is increased. Finally, this paper also presents a scheme for conserving mobile device energy by limiting its participation in the caching scheme. Results show that most of the benefit of the distributed cache can be realized even with relatively few cache assists from each client.**

I. Introduction

Consumer demand for multimedia services using Internet-enabled mobile devices, such as PDAs, smart phones, and wireless laptops, is increasing rapidly. Traditionally, cellular networks have been used to provide service outside of the home, office, or campus. WiFi, however, allows cheaper access to the media content over the Internet.

In most cases, the WiFi Access Points (APs) are connected to the Internet via a wired backhaul network. However, installing a wired infrastructure in places such as parks, sporting arenas, or pedestrian areas can be difficult. As a result, this paper will study a WiFi network with a backhaul made up of dedicated wireless links. The challenge with using a wireless backhaul is providing acceptable multimedia quality over the limited-bandwidth wireless links.

Caching of streaming media objects at points in the network close to the clients is a commonly used approach to overcome these challenges. The studies in [1], [2], [9] and the authors' work in [8] showed that streaming through an institutional network can be improved greatly by caching popular media objects on a dedicated institutional server (IS) cache with large storage capacity. On the other hand, IS cache performance can degrade due to congestion on the backhaul network that connects the IS cache to the clients [3]. This is particularly true for a low throughput wireless backhaul. The IS cache is also a single point of failure for the entire caching scheme.

The approach we consider in this paper utilizes the available storage on neighboring mobile devices to cache media objects. The use of mobile device resources to improve network performance is very much in line with the recent trend in cooperative wireless communications where distributed nodes participate actively in network operation [4], [5]. Caching on mobile devices has the advantage of being distributed, which makes it robust to the failure of individual nodes. The peer-to-peer (P2P) transmission of media objects between clients also reduces the load on the wireless backhaul. However, these advantages have to be balanced with the increased energy consumption of the mobile devices participating in the caching scheme [6], [7].

Our previous work in [8] presented a distributed caching strategy for use with mobile devices. The work in [8] considers the dynamic nature of a wireless network where the clients participating in the caching scheme arrive and depart from the network at random due to client mobility.

This paper extends our previous work in three different ways. First, we consider a finite-capacity wireless backhaul network and determine the number of concurrent clients possible with IS caching and distributed mobile caching. Second, we consider an institutional network architecture that has more than one WLAN and clients on the same WLAN can assist each other via P2P communication. As will be discussed, clients belonging to different WLANs can assist each other over the backhaul network. The third contribution of the paper involves energy conservation. Clearly, battery depletion is a concern when mobile devices are used as caching elements. This paper presents a simple method that limits the number of times a mobile has to provide cached objects. Simulation results show that the benefit of the distributed caching can be achieved when, for each client, the number of cache assists is limited to approximately the number of videos viewed.

The proposed caching system is introduced in Section II. Section III describes the experimental methodology for our simulation work, while Section IV presents the simulation results. Finally, Section V concludes the paper.

II. Proposed Caching System

The proposed system reflects caching activity in a Metropolitan Area Network (MAN) that uses a series of Wireless Local Area Networks (WLANs) or WiFi hotspots, to serve mobile clients, who stream media content from the Internet. Figure 1 shows the proposed system architecture.

A simple Internet model is adopted where there is one Origin Media Server (OMS) as a root and a MAN router on the

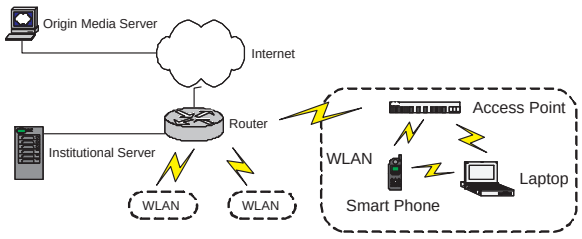

Fig. 1. Network topology.

Internet edge. This router is also the location for the IS. The IS functionality includes managing the MAN databases (client locations, session information, location of cached objects etc.) and may provide the MAN with caching capability.

The MAN Backhaul Network (MANBH) consists of one router and a variable number of WLAN APs. The router connects a group of APs via 802.11g operating on wireless channel A. Using a separate network interface, each AP connects to a group of mobile clients via channel B in 802.11g. These mobile clients join or leave the WLANs at random times. The reason we include APs between the router and the clients is to enhance the wireless communication with the low-power mobile devices [6], [7].

Using 802.11g on channel C, P2P communication is possible between clients that are geographically close to each other. Since 802.11g enables a relatively long P2P link, it is assumed that all clients within the range of the same AP can also establish a P2P link with each other.

Note that each AP/Client has two wireless network interfaces, and each one operates on a different channel to allow concurrency between incoming and outgoing media streams. Moreover, 802.11g can support 4 different wireless channels concurrently in use [10] and our network topology respects this constraint. Also, we assume 802.11g WLAN with effective capacity of 16 Mbps [3], [11], which can deliver media streams (400 kbps per client) for up to 40 clients.

In this study, we consider both distributed and IS caching scenarios. For IS caching, clients can receive data from the OMS or from the IS cache. For distributed caching, clients can receive data from the OMS or from other clients. For security and privacy purposes, we limit the data cached on a mobile client to the media objects requested by that client. In all caches, only full caching of media objects is allowed and it is assumed that the cached media objects never expire. Hence, no refresh is required to keep them up-to-date.

The distributed cache protocol allows cached data to travel from one client to the other via a P2P connection if the two clients belong to the same WLAN, or via a connection through the MANBH, if clients belong to different WLANs.

The protocol also conserves client energy by ensuring each client only provides cached objects a limited number of times. A constant A_C denotes the maximum number of times a client is required to serve a cached media object to another client.

Finally, the protocol optionally allows redundant caching of media objects in the distributed cache. When enabled, more than one client is allowed to cache the same media object at a time. When that object is requested from the cache, the device that has provided the fewest cache assists is selected to provide the object. A client streaming session can be initiated by requesting a streaming media object, which for simplicity is assumed to have Constant Bit Rate (CBR). We assume passive clients, who stream the entire requested media without any early termination, pause, rewind, or random jumps. Each session consists of a Sink Node (SK), which is requesting the media object, and a Server Node (SV), which is responsible for providing the media object. In an example session, the SK client (selected randomly from a group of online clients) sends an initiate message to the OMS. When the router receives this message, it decides whether to reassign the SV designation from the OMS. In the case of conventional IS routing, the SV can be changed to the IS cache if it contains the requested media object. In the case of distributed caching scheme, the SV can be changed to a client if that client has the media object and has assisted the caching scheme fewer than A_C times. It is then determined whether the MANBH path between the SV and SK has sufficient capacity to accommodate the media object transmission. The session is dropped if any link along the path has insufficient capacity. If the session is established, the capacity occupied by the session on each link is equal to the media object bit rate. This capacity is occupied for the duration of the object. Note that, in the case when the SV is a client, we assume that the client serving a media object doesn't disconnect until the complete object has been sent.

During transmission of the media object, the object can be cached either at the IS or in the SK, depending on the scenario. The new object is immediately stored if cache capacity is available. If not, a replacement algorithm is used to evict one or more media objects and make room for the new object.

III. Evaluation Methodology

The proposed caching system is evaluated using a discrete-event simulator. Using the GISMO Toolset [12], the time-stamps of streaming sessions are generated according to a Poisson arrival process. A client can request any one of M distinct media objects with popularity distributed according to a Zipf-like distribution. The size of these objects is lognormally distributed with mean object size 11.9 MB which corresponds to a mean duration of 4.16 min at bit rate of 400 kbps [13], [14].

Unlike a system with static clients, the identities of the online clients in this system vary with time. Node turnover due to client mobility in the WLAN is modelled as a batch arrival and departure process. At the end of timestep Δt, K_T clients are randomly chosen to depart the network and are immediately replaced. The number of residual clients that were not replaced is denoted K_R such that the total number of active clients on the network, $K_A = K_T + K_R$, remains constant. Note that a client does not rejoin the network after going offline. As a result, the total number of clients that participate at some point during the entire simulation time, T, is equal to $K = K_T T/\Delta t + K_R$. If Q is the average number

of media requests from a client per hour, then the total number of sessions during the entire simulation is $K_A QT$.

Table I lists the parameters of the two network traffic traces used in our simulations. The first trace models a group of users accessing Internet media objects with typical statistics from the literature. We reused values introduced in [9], [13], [15] to decide the number of objects, object popularity, simulation duration, and number of sessions for Trace 1. The second trace represents a workload model in time of high load where a dense population is interested in streaming the same media objects. This could occur during a sporting event or concert where clients are accessing the same highlight footage at slightly different times. Thus, we assumed a small number of objects (GISMO default value for object popularity [12]), which are requested by many clients in a short timespan.

Our system performance metrics are number of clients in service (number of simultaneous client media streams that can be supported), drop ratio (the total number of sessions dropped due to lack of network link capacity during the simulation), and hit ratio (total number of sessions served by the caches divided by the total number of sessions initiated by all clients during the simulation assuming that no session is dropped). Note that any request that could be served by the cache but is dropped due to the lack of bandwidth is still counted as a cache hit.

In addition to finite-sized caching, the special cases of infinite IS caching (Inf-IS) and infinite distributed client caching (Inf-CC) are considered. For Inf-CC, each mobile has an infinite cache. When finite caching is used, the replacement algorithms used to evict objects and create new space in the cache are Least-Frequently-Used (LFU), Least-Recently-Used (LRU) and SIZE, which removes the largest object [1], [2].

TABLE I
SIMULATION PARAMETERS FOR SYNTHETIC WORKLOADS

Parameter	Trace 1	Trace 2
Number of objects (M)	38,865	20
Object popularity (α)	0.47, 0.8	0.73
Time duration (T)	12 hours	2 hours
Concurrent clients (K_A)	220	7000
Client batch size (K_T)	170	5000
Residual clients (K_R)	50	2000
Average request arrival rate per client (Q)	7 per hour	2 per hour
Number of sessions (X)	18,480	28,000
Maximum number of assists per clients (A_C)	11	0-5
Number of APs	8	8 - 40
Per-client cache size	0-50 MB	300 MB
IS cache size	0-25 GB	300 MB

IV. SIMULATION RESULTS

A. Trace 1

For Trace 1, Fig. 2 shows the number of clients in service versus time, Fig. 3 shows dropped session ratio, and Fig. 4 shows cache hit ratio. The drop ratio and the hit ratio are first measured using object popularity skew parameter $\alpha = 0.47$ and then repeated using $\alpha = 0.8$ in order to study the effect of the object popularity distribution on the system performance. Note that in distributed caching, Trace 1 doesn't allow redundant caching and clients have the option to use P2P or the MANBH infrastructure to assist each other.

Fig. 2 shows that the number of simultaneous clients that can be supported using Inf-IS cache is constant at 40 clients, which reflects the wireless router capacity as discussed in Section II. However, in case of using Inf-CC the number of simultaneous clients that can be supported increases slightly especially in case of $\alpha = 0.8$. The simulation results show that neither the IS cache size nor the replacement algorithm affect the drop ratio for fixed α. The drop ratio in case of $\alpha = 0.47$ is 64.06%, while the drop ratio is 63.61% in case of $\alpha = 0.8$. The reason is that the router is the bottleneck in the path between the clients and the IS cache. Thus, whether we assume that the media objects have been cached in the IS cache or not, the router throughput will limit the flow from the IS cache to the clients.

In Fig. 3, however, the increase of the client cache sizes reduces the drop ratio, since using client caches reduces the pressure on the router in two ways: Clients are able to assit one another with P2P transmissions, and clients are able to find objects in their own caches.

Fig. 4 shows that the hit ratio increases as the cache size increases, which implies traffic reduction on the backbone network. Fig. 4 shows also that the hit ratio increases as α increases either for Infinite cache or finite cache. Thus, the more the object popularity is skewed, the more valuable the cache. Moreover, Fig. 4 shows that, in both caching schemes, LFU achieves the best hit ratio for both values of α. In case of IS cache, SIZE achieves the worst hit ratio when $\alpha = 0.8$, while LRU is the worst when $\alpha = 0.47$. In case of distributed cache, LRU and SIZE achieve almost the same performance for both values of α.

The results of this experiment show that the IS cache reduces the load on the Internet as well as OMS more effectively than the distributed client caches. The results also show that distributed cache, rather than IS cache, has the advantage of saving the MANBH bandwidth. However, client caches have a drawback of depending on limited batteries, while IS cache has an unlimited power source. The energy consumption by the client increases as the number of cache assists provided increases. Another factor, which affects the energy consumption of clients, is the distance between clients and the APs. With fewer APs, the average client distance increases, and more energy is consumed

B. Trace2

As discussed in Section III, Trace 2 represents a workload model in time of high load where a dense population is accessing a few media objects. Therefore, in the distributed cache scheme, the protocol in Trace 2 adopts two assumptions in order to minimize the MANBH traffic and avoid overwhelming the MANBH infrastructure. First, redundant caching is allowed. Second, only P2P assist between clients in the same WLAN (no client assists using MANBH infrastructure is allowed)

Two experiments are conducted using Trace 2. First, the number of clients in service versus time is shown in Fig. 5 for two values of A_C. Fig. 6 and 7 show the session drop ratio and the cache hit ratio, respectively, as a function of A_C, and for different numbers of APs.

Fig. 5 shows that for $A_C = 3$, the number of clients in service increases to a steady state value. This increase is slower when there are more APs because we limit the client assists to be within the same WLAN. So, it takes more sessions to cache all the media objects within each WLAN (Note that the total number of clients is always constant, and as the number of APs increases, the number of clients that belong to a single AP decreases). Yet the further increase in A_C doesn't lead to increase in the number of clients in service.

Fig. 6 and 7 show that, in case of Inf-IS, the drop ratio and the hit ratio are not affected by the number of APs. The reason again is that the router is the bottleneck in the path between the clients and the IS cache regardless of the number of APs serving the clients.

In case of Inf-CC, Fig. 6 and 7 show that the drop ratio increases as the number of APs increases, while hit ratio decreases. As the number of APs increases, the number of clients who can assist each other decreases. Thus, clients try to retrieve more media objects from OMS which results in more dropped sessions, and fewer cache hits.

Fig. 6 and 7 also show that the drop ratio decreases as A_C increases, while hit ratio increases. As more P2P assists are allowed, more objects are retrieved from client caches. Thus, the hit ratio increases and the drop ratio decreases.

The results of this experiment show that the Inf-CC outperforms the Inf-IS cache greatly at high load. Although the hit ratio is better in case of Inf-IS, the high drop ratio means that the clients are unable to retrieve the cached objects from IS due to insufficient bandwidth, which explains the small number of clients that can be supported simultaneously in case of Inf-IS.

The results also show that, in case of Inf-CC, to achieve better hit ratio, more energy per client is required, either to increase the A_C or to reduce the number of APs. On the other hand, using more APs implies not only more cost, but also fewer concurrent clients that can be supported over a short timespan.

Finally, in distributed caching scheme, preventing clients from using the MANBH infrastructure results in a reduction in drop ratio from 90%, in case of IS cache, to less than 10% when $A_C = 3$. On the other hand, in Trace 1, the reduction was negligible since clients are allowed to use MANBH infrastructure, otherwise the results of Trace 1 would show lower drop ratio and also lower hit ratio.

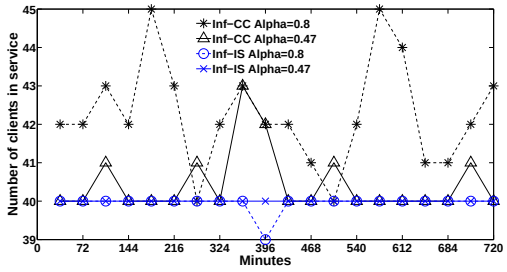

Fig. 2. Number of clients in service vs. Time

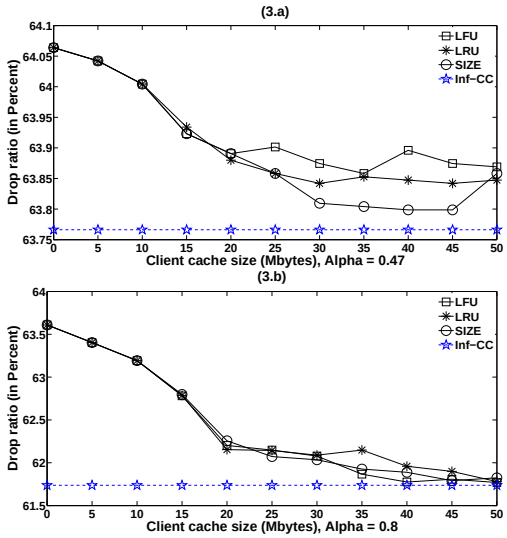

Fig. 3. Drop ratio vs. Client cache size

V. Conclusions

In this paper, we presented a distributed caching strategy for use with mobile devices in a WiFi network with a bandwidth-constrained wireless backhaul.

The proposed protocol allows for cached data to be shared among clients while, at the same time, conserving client energy by limiting participation in the scheme. Simulation results illustrate that IS caching outperforms distributed caching for a limited number of concurrent media streams. However, it's also shown that distributed caching increases the scalability of the network. This scalability is achieved at only a modest additional energy expenditure from the mobile devices.

Acknowledgements

Financial support for this research was provided by TRLabs (Telecommunications Research Laboratories, Alberta).

References

[1] S. Jin, A. Bestavros, and A. Iyengar, "Accelerating Internet Streaming Media Delivery Using Network-Aware Partial Caching", *Proceedings of IEEE International Conference on Distributed Computing Systems* (ICDCS), Vienna, Austria, pp. 153-160, July 2002.

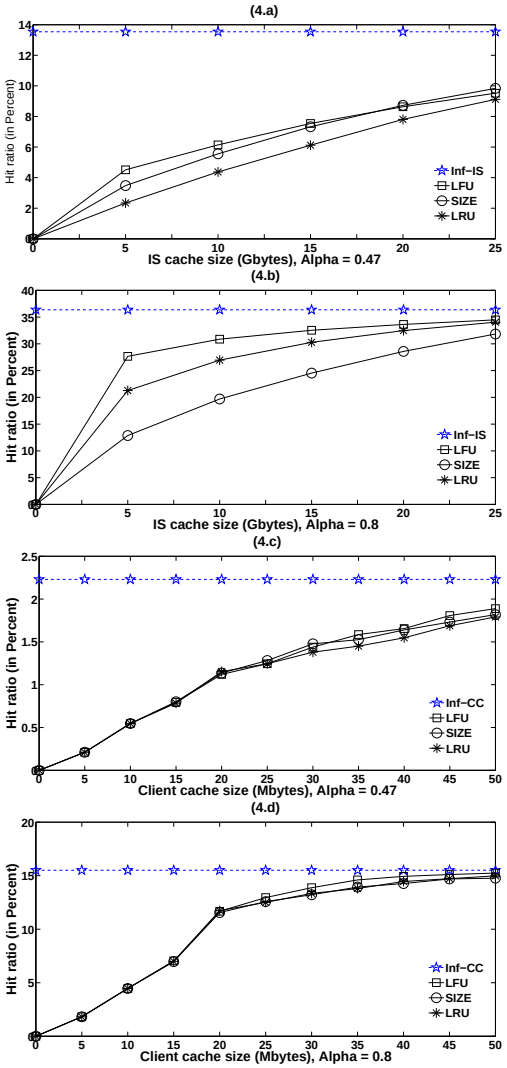

Fig. 4. Hit ratio vs. Cache size

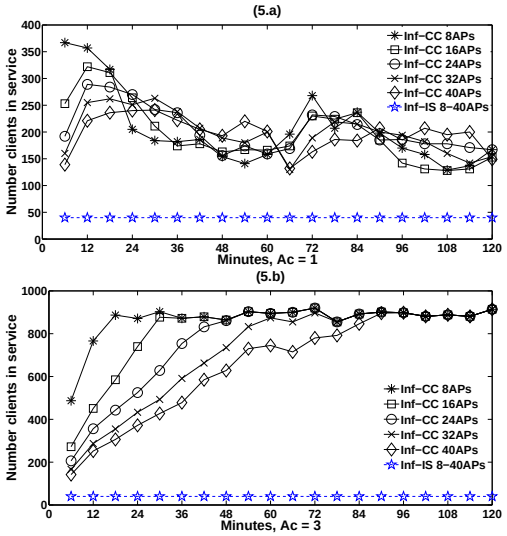

Fig. 5. Number of clients in service vs. Time

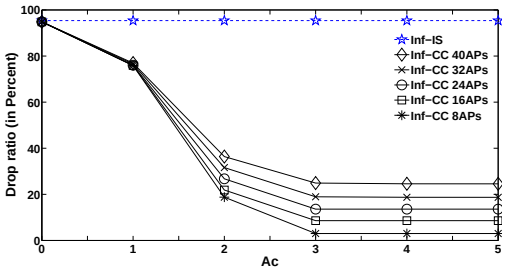

Fig. 6. Drop ratio vs. A_C

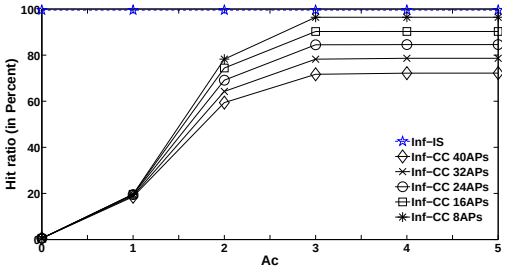

Fig. 7. Hit ratio vs. A_C

[2] P. Cao and S. Irani, "Cost Aware WWW Proxy Caching Algorithms", *Proceedings of USITS*, December 1997.

[3] X. Cao and C. Williamson, "Towards Stadium-Scale Wireless Media Streaming", *Proceedings of IEEE/ACM MASCOTS*, pages 3342, Monterey, CA, September 2006.

[4] A. Bestavros and S. Jin, "OSMOSIS: Scalable Delivery of Real-Time Streaming Media in Ad-Hoc Overlay Networks", *Proceedings of IEEE ICDCS Workshop on Data Distribution in Real-Time Systems*, Providence, RI, pp. 184-195, May 2004.

[5] S. Jin and A. Bestavros, "Cache-and-Relay Streaming Media Delivery for Asynchronous Clients", *Proceedings of the 4th International Workshop on Networked Group Communication*, Boston, MA, October 2002.

[6] Motion Computing, "Designing WLAN (802.11) to Support Tablet PC Mobility". *Motion Mobility Services*, 2007

[7] M. Stemm, P. Gauthier, D. Harada, and R. H. Katz. "Reducing Power Consumption of Network Interfaces in Hand-Held Devices". *Proc. 3rd International Workshop on Mobile Multimedia Communications*, Princeton, NJ, Sept. 1996.

[8] H. Gomaa, G. Messier, R. Davies, C. Williamson, "Media Caching Support for Mobile Transit Clients", *Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, , Marrackesh, Morocco, pp. 79-84, 2009.

[9] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Watch global, cache local: YouTube network traffic at a campus network: measurements and implications", *Technical Report 07-39, Department of Computer Science, University of Massachusetts Amherst*, 2007.

[10] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh, "Exploiting Partially Overlapping Channels in Wireless Networks: Turning a Peril into an advantage", *Proceedings of ACM IMC, Berkeley*, CA, pp. 311-316, October 2005.

[11] J. Jun, P. Peddabachagari, and M. Sichitiu, "Theoretical Maximum Throughput of IEEE 802.11 and its Applications", *Proceedings of the 2nd IEEE International Symposium on Network Computing and Applications*, Cambridge, MA, pp. 249-256, April 2003.

[12] S. Jin and A. Bestavros, "GISMO: Generator of Streaming Media Objects and Workloads", *ACM Performance Evaluation Review*, Vol. 29, No. 3, 2001.

[13] A. Abhari and M. Soraya. "Workload generation for YouTube". *Multimedia Tools and Applications*. June 2009.

[14] P. Gill, M. Arlitt, Z. Li, and A. Mahanti "'YouTube Traffic Characterization: A View From the Edge", *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 15-28, San Diego, USA, 2007.

[15] M. Chesire, A. Wolman, G. Voelker, and H. Levy, "Measurement and analysis of a streaming workload". *Proceedings of USITS*, March 2001.