

Streaming And Scheduling For Video Delivery In Device-To-Device Without Quality Compromise

Ilavarasi J¹, Subhashini R², S. Brindha³

¹Research Scholar, St.Peter's University, Chennai.

^{2,3}Asst.Prof. & Dept. of Computer Science & Applications, St.Peter's University, Chennai
ilavarasijp@gmail.com

ABSTRACT- Video streaming is becoming a destroyer application for wireless networks. It has shown recently that a combination of caching on the users' devices and device to-device (D2D) communications yields throughput scalability for very dense networks, which represent critical bottlenecks for wireless local area network (WLAN) and conventional cellular technologies. We do implementations of such caching D2D systems where each device pre-caches a detachment of video files from a library, and users requesting a file that is not in their own library but fetch it delivered through D2D communication. In this paper we describe the network architecture of Zattoo, one of the largest production live streaming providers in Europe at the time of writing, and present a large-scale measurement study of Zattoo using data collected by the provider. PDM to minimize per-packet processing time of a stream, the Zattoo protocol sets up a virtual circuit with multiple fan outs at each device. Device joins a TV channel it establishes a device-division multiplexing(PDM) scheme amongst a set of neighboring peer by building a virtual circuit to each of the neighboring peers. The proposed approaches are shown to represent a peer as a packet buffer, called the MDC, fed by sub-streams incoming from the PDM constructed as described in a local media player if one is running. As packets from each sub-stream arrive at the peer, they are stored in the MDC for reassembly to reconstruct the full stream. portions of the stream that have been reconstructed are then played back to the user. In addition to providing a reassembly area, the MDC also allows a peer to absorb some variability's in available network bandwidth and network delay. We use retransmission to let a peer recover from transient network congestion. A peer sends out a retransmission request when the distance between the repair pointer and the input pointer has reached a threshold of R packet slots, usually spanning multiple segments. A retransmission request consists of an R - bit packet mask, with each bit representing a packet, and the sequence number of the packet corresponding to the first bit. Marked bits in the packet mask indicate that the corresponding packets need to be retransmitted

I. INTRODUCTION

Device to device systems for live streaming have been introduced in recent years. The behavior of these popular systems has been extensively studied in several measurement papers. Due to the proprietary nature of these commercial systems, however, these studies have to rely on a "black-box" approach, where packet traces are collected from a single or a limited number of measurement points, to infer various properties of traffic on the control and data planes. Although

such studies are useful to compare different systems from end-user's perspective, it is difficult to intuitively understand the observed properties without fully reverse-engineering the underlying systems. Numerous commercial systems now offer services over the Internet that is similar to traditional over-the-air, cable, or satellite TV. Live television, time-shifted programming, and content-on demand are all presently available over the Internet. Increased broadband speed, growth of broadband subscription base, and improved video compression technologies have contributed to the emergence of these IPTV services. We draw a distinction between three uses of device-to-device (D2D) networks: delay tolerant file download of archival material, delay sensitive progressive download (or streaming) of archival material, and real-time live streaming. In the first case, the completion of download is elastic, depending on available bandwidth in the D2D network. The application buffer receives data as it trickles in and informs the user upon the completion of download. The user can then start playing back the file for viewing in the case of a video file. Bit torrent and variants are example of delay-tolerant file download systems.

In the second case, video playback starts as soon as the application assesses it has sufficient data buffered that, given the estimated download rate and the playback rate, it will not deplete the buffer before the end of file. If this assessment is wrong, the application would have to either pause playback and re-buffer, or slow down playback. While users would like playback system delivers live streams using a receiver-based, peer division multiplexing scheme. According to recent predictions of the Cisco Visual Networking Index (VNI) [11], the sum of all forms of video will constitute 80% to 90% of global consumer data traffic by 2017, and the traffic from wireless and mobile devices will exceed the traffic from wired devices by 2016. Therefore, efficient video-aware network algorithms for wireless networks are of highest importance. It has been shown recently that the throughput for delivery of wireless video files can be greatly enhanced by device-to-device (D2D) communications, where direct links between pairs of user devices can be set up without requiring to go through a central base station. In particular, these works propose systems where each device caches independently, according to a certain optimal distribution, a subset of popular video files.

Literature survey: Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into account for developing the proposed system.

Product features: Zattoo device to device live streaming system was a free to use network serving over 3 million registered users in eight European countries at the time of study, with a maximum of over 60,000 concurrent users on a single channel. The system delivers live streams using a *receiver-based, peer division multiplexing* scheme. We presented a receiver-based, peer-division multiplexing engine to deliver live streaming content on a peer-to-peer network. The same engine can be used to transparently build a hybrid P2P/CDN delivery network by adding Repeater nodes to the network. By analyzing large amount of usage data collected on the network during one of the largest viewing event in Europe, we have shown that the resulting network can scale to a large number of users and can take good advantage of available uplink bandwidth at peers.

II. METHODOLOGY

a. Zattoo video to video system: Zattoo device to device live streaming system was a free to use network serving over 3 million registered users in eight European countries at the time of study, with a maximum of over 60,000 concurrent users on a single channel. The system delivers live streams using a *receiver-based, peer division multiplexing* scheme as described in to ensure real-time performance when peer uplink capacity is below requirement, Zattoo subsidizes the network's bandwidth requirement The Zattoo system rebroadcasts live TV, captured from satellites, onto the Internet. The system carries each TV channel on a separate device to device delivery network and is not limited in the number of TV channels it can carry. Although a peer can freely switch from one TV channel to another, and thereby departing and joining different device to device networks, it can only join one device to device network at any one time. We henceforth limit our description of the Zattoo delivery network as it pertains to carrying one TV channel on the Zattoo network.

b. Live data streaming: Zattoo live streaming protocol as a receiver-based, peer division multiplexing protocol the details of peer-division multiplexing is described in a while the details of how a peer manages sub-stream forwarding and stream reconstruction is described in receiver-based peer division multiplexing has also been used by the latest version of Cool Streaming device to device protocol though it differs from Zattoo in its stream management. As packets from each sub-stream arrive at the peer, they are stored in the IOB for reassembly to reconstruct the full stream. Portions of the stream that have been reconstructed are then played back to the user. In addition to providing a reassembly area, the IOB also allows a peer to absorb some variability's in available network.

c. peer-division multiplexing (PDM): PDM per-packet processing time of a stream, the Zattoo protocol sets up a virtual circuit with multiple fan outs at each peer. When a peer joins a TV channel, it establishes a peer-division multiplexing (PDM) scheme amongst a set of neighboring peers, by building a virtual circuit to each of the neighboring peers. Baring departure or performance degradation of a neighbor peer, the virtual circuits are maintained until the joining peer switches to another TV channel. With the virtual circuits set up, each packet is forwarded without further per-packet handshaking between peers. We describe the PDM boot strapping mechanism in this section and the adaptive PDM mechanism to handle peer departure and performance technique.

III. RESULTS

We have shown that the resulting network can scale to a large number of users and can take good advantage of available uplink bandwidth at peers. We have also shown that error-correcting code and packet retransmission can help improve network stability by isolating packet losses and preventing transient congestion from resulting in PDM reconfigurations.

Traces	Minimum(DB)	Maximum(DB)
Video trace 1	28.4835 db	36.8063 db
Video trace 2	25.3273 db	36.0470 db
Video trace 3	28.8283 db	37.1691 db
Video trace 4	24.3273 db	35.2584 db
Video trace 5	29.4835 db	37.8063 db

We have further shown that the PDM and adaptive PDM schemes presented have small enough overhead to make our system competitive to digital satellite TV in terms of channel switch time, stream synchronization.

The following tables are used to represent the video traces to prove the quality of pixels

Theme:

There is an emerging market for IPTV. Numerous commercial systems now offer services over the Internet that is similar to traditional over-the-air, cable, or satellite TV. Live television, time-shifted programming, and content on demand are all presently available over the Internet. Increased broadband speed, growth of broadband subscription base, and improved video compression technologies have contributed to the emergence of these IPTV services.

Scope:

The application buffer receives data as it trickles in and informs the user upon the completion of download. The user can then start playing back the file for viewing in the case of a video file. Bit torrent and variants are example of delay-tolerant file download systems. Video playback starts as soon as the application assesses it has sufficient data buffered that, given the estimated download rate and the playback rate, it will not deplete the buffer before the end of file. If this assessment is wrong, the application would have to either pause playback or rebuffered, or slow down playback. While users would like playback to start as soon as possible, the application has some degree of freedom in trading off playback start time against estimated network capacity. Most video on demand systems are examples of delay-sensitive progressive download application.

Features: Zattoo peer-to-peer live streaming system was a free to use network serving over 3 million registered users in eight European countries at the time of study, with a maximum of over 60,000 concurrent users on a single channel. The system delivers live streams using a *receiver-based, peer division multiplexing* scheme. We presented a receiver-based, peer-division multiplexing engine to deliver live streaming content on a peer-to peer network. The same engine can be used to transparently build a hybrid P2P/CDN delivery network by adding Repeater nodes to the network. By analyzing large amount of usage data collected on the network during one of the largest viewing event in Europe, we have shown that the resulting network can scale to a large number of users and can take good advantage of available uplink bandwidth at peers.

Modeling rules:

There are several common modeling rules when creating DFDs:

1. All processes must have at least one data flow in and one data flow out.
2. All processes should modify the incoming data, producing new forms of outgoing data.
3. Each data store must be involved with at least one data flow.
4. Each external entity must be involved with at least one data flow.

IV. TECHNICS AND ALGORITHM:

PEER-DIVISION MULTIPLEXING:

To minimize per-packet processing time of a stream, the Zattoo protocol sets up a virtual circuit with multiple fan outs at each peer. When a peer joins a TV channel, it establishes a peer-division multiplexing (PDM) scheme amongst a set of neighboring peers, by building a virtual circuit to each of the neighboring peers. Baring departure or performance degradation of a neighbor peer, the virtual circuits are maintained until the joining peer switches to another TV channel with the virtual circuits set up, each packet is forwarded without further per-packet handshaking between peers. We describe the PDM boot strapping mechanism in this section and the adaptive PDM mechanism to handle peer departure and performance Degradation. The PDM establishment process consists of two phases: the *search phase* and the *join phase*. In the search phase, the new, joining peer determines its set of potential neighbors. In the join phase, the joining peer requests peering relationships with a subset of its potential neighbors. Upon acceptance of a peering relationship request, the peers become neighbors and a virtual circuit is formed between them.

SEARCH PHASE: To obtain a list of potential neighbors, a joining peer sends out a SEARCH message to a random subset of the existing peers returned by the Rendezvous Server. The SEARCH message contains the sub-stream indices for which this joining peer is looking for peering relationships. The sub stream indices are usually represented as a bitmask of n bits, where n is the number of sub-streams defined for the TV channel. In the beginning, the joining peer will be looking for peering relationships for all sub-streams and have all the bits in the bitmask turned on. In response to a SEARCH message, an existing peer replies with the number of sub-streams it can forward. From the returning SEARCH replies, the joining peer constructs a set of potential neighbors that covers the full set of sub-streams comprising the live stream of the TV channel. The joining peer continues to wait for SEARCH replies until the set of potential neighbors contains at least a minimum number of peers, or until all SEARCH replies have been received. With each SEARCH reply, the existing peer also returns a random subset of its known peers. If a joining peer cannot form a set of potential neighbors that covers the entire sub streams of the TV channel, it initiates another SEARCH round, sending SEARCH messages to peers newly learned from the previous round. The joining peer gives up if it cannot obtain the full stream after two SEARCH rounds. To help the joining peer synchronize the sub-streams it receives from multiple peers, each existing peer also indicates for each sub-stream the latest sequence number it has received for that sub-stream, and the existence of any quality problem. The joining peer can then choose sub-streams with good quality that are closely synchronized.

JOIN PHASE: Once the set of potential neighbors is established, the joining peer sends JOIN requests to each potential neighbor. The JOIN request lists the sub-streams for which the joining peer would like to construct virtual circuit with the potential neighbor. If a joining peer has l potential neighbors, each willing to forward it the full stream of a TV channel, it would typically choose to have each forward only $1/l$ -th of the stream, to spread out the load amongst the peers and to speed up error recovery, as described in selecting which of the potential neighbors to peer with, the joining peer gives highest preference to topologically close-by peers, even if these peers have

less capacity or carry lower quality sub-streams. The “topological” location of a peer is defined to be its subnet number, autonomous system (AS) number, and country code, in that order of precedence. A joining peer obtains its own topological location from the Zattoo Authentication Server as part of its authentication process. The list of peers returned by both the Rendezvous Server and potential neighbors all come attached with topological locations. A topology-aware overlay not only allows us to be “ISP-friendly,” by minimizing inter-domain traffic and thus save on transit bandwidth cost, but also helps reduce the number of physical links and metro hops traversed in the overlay network, potentially resulting in enhanced user perceived stream quality.

V. CONCLUSION

We have presented a receiver-based, peer-division multiplexing engine to deliver live streaming content on a device to device network. The same engine can be used to transparently build a hybrid P2P/CDN delivery network by adding Repeater nodes to the network. By analyzing large amount of usage data collected on the network during one of the largest viewing event in Europe, we have shown that the resulting network can scale to a large number of users and can take good advantage of available uplink bandwidth at peers. We have also shown that error-correcting code and packet retransmission can help improve network stability by isolating packet losses and preventing transient congestion from resulting in PDM reconfigurations. We have further shown that the PDM and adaptive PDM schemes presented have small enough overhead to make our system competitive to digital satellite TV in terms of channel switch time, stream synchronization, and signal lag.

Zattoo performs a NAT detection procedure upon player startup to identify its NAT configuration and advertises it to the rest of the Zattoo network. Zattoo’s NAT detection procedure implements a UDP-based standard STUN protocol which involves communicating with external STUN servers to discover the presence/type of a NAT gateway. The communication occurs in UDP with no reliable transport guarantee. Lost or delayed STUN UDP packets may lead to inaccurate NAT detection, preventing NATed clients from contacting each other, and therefore adversely affect their sharing performance.

REFERENCE

- [1]. R. auf der Maur, “Die Weiterverbreitung von TV- und Radioprogrammen über IP-basierte Netze,” in *Entertainment Law (f. d. Schweiz)*, 1st ed. Stämpfli Verlag, 2006.
- [2]. UEFA, “Euro2008,” <http://www1.uefa.com/>.
- [3]. S. Lin and D. J. Costello, Jr., *Error Control Coding*, Pearson Prentice-Hall, 2004.
- [4]. S. Xie, B. Li, G. Y. Keung, and X. Zhang, “CoolStreaming: Design, Theory, and Practice,” *IEEE Trans. on Multimedia*, vol. 9, no. 8, December 2007.
- [5]. K. Shami *et al.*, “Impacts of Peer Characteristics on P2PTV Networks Scalability,” in *Proc. IEEE INFOCOM Mini Conference*, April 2009.
- [6]. Bandwidth-test.net, “Bandwidth test statistics across different countries,” <http://www.bandwidth-test.net/stats/country/>.

EXPLORATIONS ON ENGINEERING LETTERS (EEL)
VOLUME 1, ISSUE 1 (2016):PP.85-91
SANA ACADEMIC PRESS

- [7]. X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "Insights into PPLive: A Measurement Study of a Large-Scale P2P IPTV System," in *Proc. IPTV Workshop, International World Wide Web Conference*, 2006.
- [8]. B. Li *et al.*, "An Empirical Study of Flash Crowd Dynamics in a P2P-Based Live Video Streaming System," in *Proc. IEEE Global Telecommunications Conference*, 2008.
- [9]. J. R. et al, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," 1993, RFC 3489.
- [10]. A. Ali, A. Mathur, and H. Zhang, "Measurement of Commercial Peerto- Peer Live Video Streaming," in *Proc. device to deviceStreaming*, August 2006.