International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 3, May-Jun-2019, ISSN (Online): 2395-566X

A Review of Flash Crowd Handling In P2p Live Video Streaming Systems

M.Tech. Scholar Mahazbeen Khan **Prof. Rahul Pandey**
Department of Electronics & Communication Engineering
School of Research & Technology, People's University
Bhopal, MP, India

Abstract- **Peer-to-peer systems have greatly enhanced live streaming experience by creating efficient and highly scalable streaming overlays where bandwidth capabilities of all peers can be utilized. However, realization of such systems have been challenged by the phenomenon of flash crowd — the arrival of hundreds of thousands of peers in a very short span of time. Such situations may typically arise at the beginning of live streaming events such as a football match or a live lecture.**

Keywords-**P2P Network, Content Delivery Networks (CDN), Distributed Hash Table (DHT)**

I. INTRODUCTION

1. What is a P2P Network?

A peer-to-peer (P2P) network is in stark contrast to traditional client-server based net- works where functions of client and server nodes are static and differ from each other. In a P2P network, all the nodes (peers) are at equal level and can play the role of both client as well as server. The work load of the network is distributed among the peers with each peer making a portion of its resources, such as processing power, disk storage or network bandwidth, directly available to other peers [1]. The peers are able to organize and collaborate with each other without the need of a central authority.

The P2P architecture was popularized by the development of Napster in 1999. It allowed the users to share mp3 files with other users. A central index list of the shared files was maintained by the Napster server. A user would search for the required file in the list and then directly download it from the providing peer. Since then, P2P systems have evolved both in structure as well as in application. Currently, they are used for a variety of purposes such as file sharing (Bit Torrent [2], Gnutella [3]), multimedia (Skype [4], Cool Streaming [5]), distributed computing (SETI@home [6]), storage services (Free Net [7]) and even digital crypto currency (Bitcoin [8]).

2. Types

The existing P2P architectures can be divided into2 broad categories.

2.1 Unstructured: The peers are not organized in any particular structure and form random connections with each other. Due to lack of structure, query resolution can be done only through flooding or random walk [9]. Even then, there can be no guarantee that the query would be successfully resolved.

2.2 Structured: The resources are distributed in the network according to an algorithm. The algorithm guarantees that a query originating anywhere in the network will end in a definite success or failure in a bounded number of hops. Some examples are Chord [10], Pastry [11] and Tapestry [12].

3. Advantages

P2P networks have numerous advantages over networks based on server-client model. Some of them are:

3.1Efficient- P2P networks can be used to build highly efficient systems as the re- sources of all the peers are available for use.

3.2 Scalable- New peers who join the network brings along additional processing and storage capabilities. Thus, the total capacity of the system grows along with the increase in load on the system. By maintaining fine balance between the two, highly scalable systems can be designed.

3.3 Reliable- In the absence of a central server, P2P networks do not suffer from single point of failure. Even in case of failure of few peers, reliability can be maintained by replicating the data in the network and storing it on separate peers.

3.4 Inexpensive- As P2P networks do not require deployment of additional servers, they are very inexpensive.

4. Disadvantages

Apart from the advantages covered above, P2P networks also suffer from a few disadvantages. Some of them are stated below:

4.1 Difficult administration- In the absence of a centralized authority, the P2P networks becomes difficult to administer. If there is lack of cooperation among the peers, the system performance can be greatly affected.

4.2 Insecure- P2P networks are highly susceptible to attacks by malicious users. They may upload malicious contents, drop routing requests or collude together to disrupt the services of the system.

4.3 Free riding- It is one of the most important challenges to P2P network. A peer may consume resources but may not share their own resulting in bulk of the work being done by a small percentage of peers.

4.4 Network Churn- In P2P networks, apart from serving its own interest, a peer is also responsible for providing services to other peers. Ungraceful exit (without notifying other peers) of a peer from the system can not only disrupt other peers' service but can also cause irreversible data loss if not handled properly.

II.LITERATURE SURVEY

Chen et al. Understands flash crowd as a sudden increase of peer arrival rate. Zhang et al. Expands this understanding and defines flash crowd in terms of shock level. The shock level of a flash crowd is defined as the ratio of the peer arrival rate during and before the flash crowd. Similarly, the capacity of the system is defined as the shock level of the highest flash crowd that the system can survive. A different representation of flash crowd is used. Liu et al. models flash crowd as abrupt arrival of a large number of peers. Since, the latter representation can be considered a special case of the former, the representation used for this thesis.In recent years, a more rigorous analysis has helped broaden the understanding of flash crowd dynamics.

Liu et al. examined the fundamental characteristics of flash crowd and proposed a time-scale relationship in mesh-based live streaming systems. They showed that with stringent time constraints as in the case of live streaming systems, the network can scale only up to a limit during flash crowd. This is because only that many peers can be satisfied in the first attempt as is the available surplus bandwidth of the system. But the key insight obtained was that having available surplus bandwidth alone is not sufficient for the system to scale as peers take time to locate the available resources. The terminology used is used to denote the peers who are already connected and able to forward streams as stable peers and the new arrived peers as start-up peers.

Liu et al. also examined the effects of various critical factors such as initial system scale, flash crowd scale, number of partners and per peer upload capacity on the system's capability to handle flash crowd. Based on all these, they designed a simple population control framework.

Zhang et al. used a fluid based model to examine the flash crowd dynamics and estimated the strength of flash crowd in terms of shock level. They showed that using proper population control measures the waiting time of peers can be made to increase logarithmically with the shock level of the flash crowd.

Chung et al. identified peer join process as a system bottleneck during flash crowd. The small pool of stable peers becomes overloaded by surge of join request during flash crowd. Hence, they advocated to alter the serial join process and instead join the peers in batches. This is done by constructing a tree completely out of start-up peers and then connecting it to the existing system.

Wu et al. takes this scheme further and proposes to construct not one but many trees based on the available surplus bandwidth of the system. The underlying principle in these approaches is to isolate the start-up peers and arrange them in a topology before connecting them to the existing network. As seen earlier, single tree-based systems suffer from resource under-utilization and fault tolerance issues. Split Stream tackles this problem by constructing a forest of interior-node-disjoint multicast trees that distributes the forwarding capacity among the participating peers [18]. Every node is an interior node in only one tree and a leaf node in all other trees. Also, depending upon the forwarding capacity of a node, it can select the number of sub-stream trees to join.

Split Stream relies on a structured peer-to-peer overlay to construct and maintain trees. Pastry [11] and Scribe are used to provide this structure. Pastry is a DHT based, scalable, self organizing peer-to-peer network similar to Chord [10] and Tapestry [12]. Scribe is a application level group communication system built upon Pastry.

III. FLASH CROWD IN P2P NETWORK

The phenomenon of arrival of hundreds of thousands of peer in a very short span of time is called flash crowd. Such situations typically arise at the beginning of live streaming events. The newly arrived peers compete for the limited system resources and drastically reduce the performance of the system. The problem of flash crowd is more challenging in live streaming systems due to stringent time constraints associated with the resources. A significant number of newly arrived peer may leave the system if they are unable to meet these stringent time constraints. This adds to the system churn and makes flash crowd handling more difficult.

Flash crowds have been traditionally handled by deployment of additional resources - servers in Cool Streaming+ or content delivery network in SkyNet. However, this method is not cost effective. Moreover, it has been observed that the system performance can be maintained at a high level once a sufficient system scale has been achieved. Hence, the additional resources are only necessary for the initial period and useless afterwards. In recent years, a more rigorous analysis of the problem has helped broaden the understanding of flash crowd dynamics. Having upload bandwidth alone is not sufficient to accommodate the flash crowd as the nodes take time to locate the available resources. Moreover, due to intense competition among the nodes, this available bandwidth is also not fully utilized. Based on these, various population control measures have been suggested for both mesh-based and tree-based systems.

IV. LIVE STREAMING IN P2P NETWORK

In tradition client-server systems, each client sets up a separate and direct connection with the server. It results in bottleneck at the server if the numbers of clients are above a limit. Content Delivery Networks (CDN) are used to alleviate the load on source server and the task of providing streams is done by the content delivery servers. The source node pushes the stream to these content delivery servers from where it is streamed to the requesting clients. Such systems reduce the load on source node but still cannot support very high number of peers.

The use of P2P network has greatly enhanced live streaming experience by creating efficient and highly scalable streaming overlays. In such systems, the requesting clients also act as content providers by forwarding parts of stream that they posses to other requesting peers. As every peer contribute to the aggregate system bandwidth, P2P live streaming systems can support a very large number of peers. Measurement studies on existing systems have shown that the system performance can be maintained at a high level once a sufficient system scale has been achieved [13] [14]. The existing solutions to peer-to-peer live video streaming can be divided into 2 categories

- Tree-Based
- Mesh-Based.

1.Tree-Based

The peers are organized into a tree-shaped overlay with static parent-child relationship among them. In single-tree approaches as shown in Figure 1, a single tree rooted at the source node is constructed and the stream is delivered through a push mechanism where each node forwards the data to all of its child nodes. This approach is not resilient to system churn as removal of one node from the network leads to disruption of services of all of its descendant nodes. Moreover, the outgoing bandwidth of leaf nodes is not used resulting in poor resource utilization. Examples include Nice [15] and Zigzag [16].

In multi-tree approaches, the stream is divided into multiple sub-streams using appropriate data encoding technique such as Multiple Descriptive Coding (MDC) [17] and each sub-stream is pushed over a separate tree. Peers choose the number of trees to connect to depending on their download bandwidth. To ensure equitable workload distribution and to minimize the effect of churn, a peer is placed as an internal node in only one tree and as a leaf node in all other trees. If such a node leaves the network, at most one sub-stream is affected. Figure 2 gives an example of multi-tree construction with two sub-streams. Examples include Split stream [18], Bullet [19] and Coop Net [20].

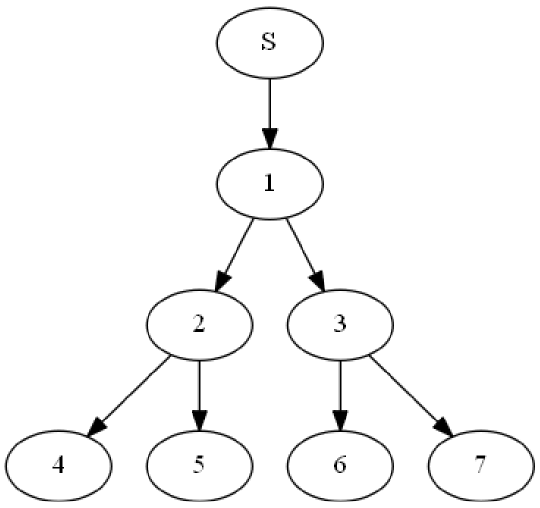

Figure 1 Peer organization in single tree-based live streaming system.

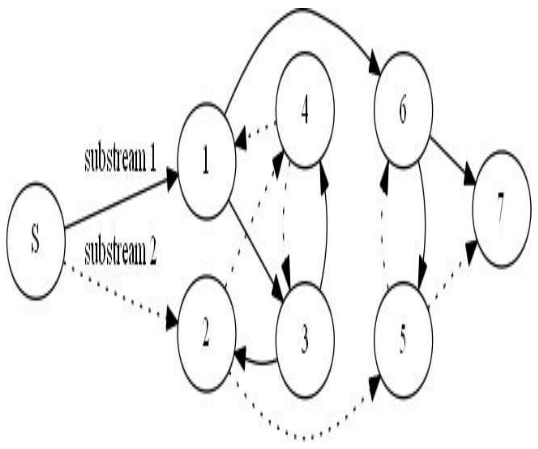

Figure 2: Peer organization in multiple tree-based live streaming system

2.Mesh-Based

Such systems form a random connected overlay of peers and utilizes swarming content delivery to exchange packets. Every node on joining the system gets a list of random nodes from the bootstrapping server. It then periodically reports the newly available packet to its child peers and at the same time requests new packets from its parent peers. An example of mesh-based system is given in Figure 3. The bi-directional arrows represents that the parent-child relationship among the peers is dynamic and is determined by data availability. Mesh-based systems are more resilient to system churn. Examples include Coolstreaming [5] and Prime.

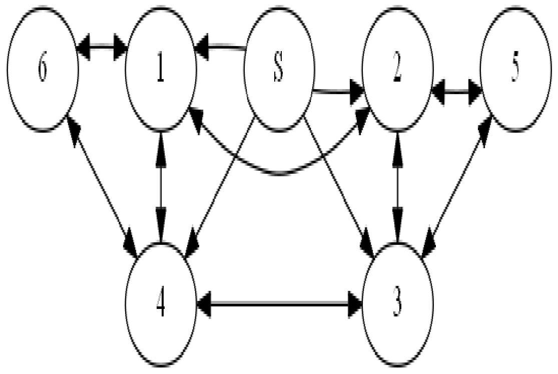

Figure 3 Peer organization in mesh-based live streaming system.

V.CONCLUSION

Flash crowd in P2P live streaming systems is a challenging phenomenon. As discussed earlier, the problem of flash crowd has seen considerable interest in recent years and various population control measures have been suggested. In this thesis, the main focus is on live streaming in tree-based P2P systems. Some proposals have been suggested to handle flash crowd in tree-based live streaming systems, but they adopt a centralized algorithm. Such solutions suffer from scalability and fault tolerance issues. As part of this thesis, a distributed algorithm that can organize the newly arrived peers in multiple sub-stream trees with minimal central control is proposed.

REFERENCE

[1] R. Schollmeier. "A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications". In: Peer-to-Peer Computing, 2001. Proceedings. First International Conference on. 2001, pp. 101–102. DOI: 10.1109/ P2P.2001.990434.
[2] Bit Torrent. URL: http://www.bittorrent.com/.
[3] Gnutella.URL:http://en.wikipedia.org/wiki/Gnutella.
[4] Skype Free Calls to friends and family. URL:http://skype.com.
[5] Xinyan Zhang et al. "Cool Streaming/DONet: a data-driven overlay network for peer-to-peer live media streaming". In: INFOCOM 2005. 24th Annual Joint Con- ference of the IEEE Computer and Communications Societies. Proceedings IEEE. Vol. 3. IEEE. 2005, pp. 2102–2111.
[6] David P Anderson et al. "SETI@ home: an experiment in public-resource computing". In: Communications of the ACM 45.11 (2002), pp. 56–61.
[7] Ian Clarke et al. "Freenet: A distributed anonymous information storage and re- trieval system". In: Designing Privacy Enhancing Technologies. Springer. 2001, pp. 46–66.
[8] Satoshi Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: Consulted 1.2012 (2008), p. 28.
[9] Christos Gkantsidis, Milena Mihail, and Amin Saberi. "Random walks in peer-to- peer networks". In: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. Vol. 1. IEEE. 2004.
[10] Ion Stoica et al. "Chord: A scalable peer-to-peer lookup service for internet ap- plications". In: ACM SIGCOMM Computer Communication Review 31.4 (2001), pp. 149–160.
[11] Antony Rowstron and Peter Druschel. "Pastry: Scalable, decentralized object lo- cation, and routing for large-scale peer-to-peer systems". In: Middleware 2001. Springer. 2001, pp. 329–350.
[12] Ben Y Zhao et al. "Tapestry: A resilient global-scale overlay for service deploy- ment". In: Selected Areas in Communications, IEEE Journal on 22.1 (2004), pp. 41– 53.
[13] Xiaojun Hei et al. "A measurement study of a large-scale P2P IPTV system". In: Multimedia, IEEE Transactions on 9.8 (2007), pp. 1672–1687.
[14] Bo Li et al. "Inside the new coolstreaming: Principles, measurements and per- formance implications". In: INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. IEEE. 2008.
[15] Suman Banerjee, Bobby Bhattacharjee, and Christopher Kommareddy. Scalable application layer multicast. Vol. 32. 4. ACM, 2002.
[16] Duc A Tran, Kien A Hua, and Tai Do. "Zigzag: An efficient peer-to-peer scheme for media streaming". In: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies. Vol. 2. IEEE. 2003, pp. 1283–1292.
[17] Vivek K Goyal. "Multiple description coding: Compression meets the network". In:Signal Processing Magazine, IEEE 18.5 (2001), pp. 74–93.
[18] Miguel Castro et al. "Split Stream: high-bandwidth multicast in cooperative environments". In: ACM SIGOPS Operating Systems Review. Vol. 37. 5. ACM. 2003, pp. 298–313.
[19] Dejan Kosti´c et al. "Bullet: High bandwidth data dissemination using an over- lay ". In: ACM SIGOPS Operating Systems Review. Vol. 37. 5. ACM. 2003, pp. 282–297.
[20] Venkata N Padmanabhan, Helen J Wang, and Philip A Chou. "Resilient peer-to- peer streaming". In: Network Protocols, 2003. Proceedings. 11th IEEE International Conference on. IEEE. 2003, pp. 16–27.