Flourishing in the Internet: the huge success of live peer-to-peer media streaming

Zhiguang Qin, Xiaodong Wei, Mengjuan Liu, and Lei Yang
Department of Computer Science and Technology
University of Electronic Science and Technology of China, Chengdu, China
{zgqin, xdwei, mjliu, ly}@uestc.edu.cn

***Abstract*—Peer-to-Peer (P2P) media streaming networks have attracted significant research interest recently. However, it is challenging to design live P2P media streaming systems because of the stringent time constraints on the delivered media streams, which require more efficient and resilient overlay. In this paper, we focus on live P2P media streaming networks, a promising application flourishing in the Internet and which requires the distribution of live multimedia content to subscribers. We discuss several live P2P streaming systems, present a basic taxonomy of P2P overlays and closely examine these three approaches, tree-based, mesh-based, tree-mesh-hybrid-based, and summarize the major issues about them. we also categorizes peer selection schemes which are essential for the construction of a P2P streaming network.**

Keywords-media streaming; peer-to-peer; live

I. INTRODUCTION

Before the deployment of peer-to-peer (P2P) networks, the client-server architecture and the content distribution network (CDN) along with IP multicast were the most desirable solutions to support media streaming. These solutions are gradually losing ground because of the widespread deployment of P2P networks with their unique characteristics. A major advantage of a P2P network is that each peer contributes its own resources to the network. As a result, there is an increase in the amount of overall resources of the network, such as bandwidth, storage space, and computing power. Consequently, P2P solutions overcome the bottleneck problem at the server in a client-server model. CDN alleviates the same bottleneck problem by introducing more dedicated servers at geographically different locations which also results in expensive deployment and maintenance. Lastly, IP multicast has good scalability only in theory because its actual deployment across the Internet is limited.

Media streaming has a tight time constraint in that the playback starts soon after the streaming begins and the stream should be played back continuously; whereas traditional P2P file downloading has no such requirement on the downloading order of different blocks of a file. In addition, the file is accessed by a user only after the whole file has been downloaded. These differences required more improvements to the architectural design of P2P file downloading protocols to readily address the timing constraints and to provide good media quality for P2P media streaming protocols.

In this paper, we focus on live P2P media streaming networks, a promising application flourishing in the Internet. We discuss two important components in the design of live P2P streaming networks: content delivery and peer selection. The former considers how to efficiently distribute a media stream through the established architecture, and the latter deals with how to select a right set of peers to construct an efficient and resilient architecture of a P2P streaming network. The rest of the paper is organized as follows. We survey several typical architecture topologies in Section 2, and content delivery in Section 3. Section 4 categorizes peer selection schemes. Finally Section 5 concludes the paper.

II. SURVEY: P2P LIVE STREAMING SYSTEMS

The distinguishing and stringent requirements of media streaming necessitate fundamentally different design decisions and approaches. We review the state-of-the-art of peer-to-peer technologies for live media streaming, and present a taxonomy of various solutions that have emerged. In particular, three broad approaches have emerged: tree-based approaches, mesh-based approaches and tree-mesh-hybrid approaches. We survey typical systems and their differences.

A. systems based on a tree

Borrowing ideas from IP multicast, tree-based systems are simple, efficient, and scalable. There are two types of tree-based protocols, including single tree protocols (see Fig 1.), such as ESM/NARADA, NICE, ZIGZAG, SCRIBE, and multiple tree protocols [14] [15] (see Fig. 2). Zigzag [13] constructs a multicast tree with multi-layer hierarchical clustering concepts, similar to the approach adopted by Nice. The size of Zigzag clusters is kept between k to 3k-1, same as that in Nice. However, unlike Nice, Zigzag separates administrative organizations with data forwarding organizations. Nice always uses the head of a cluster to forward both the data content and the control message to the other members in the cluster, whereas Zigzag uses head node to maintain control message exchanges only and additional node, called foreign head, to forward data content. With such a different control and data forwarding strategies, Zigzag guarantees that the height of the tree built is O($\log_k N$) where N is the total number of peers and k is a constant.

Multiple tree have been introduced to: 1) improve resiliency to group and network dynamics with redundancy by using multiple paths from the source to each destination; 2) increase the supply bandwidth by using bandwidth from all nodes that can supply it, which cannot be done in single trees (i.e., by definition leaf nodes cannot forward any data);

978-1-4244-7237-6/$26.00

3) address receiver heterogeneity by varying the number of trees a host joins or the number of parents a node chooses; and 4) address forwarder heterogeneity by sending multiple streams and reducing the bandwidth of each stream, allowing low bandwidth hosts to forward to a few children.

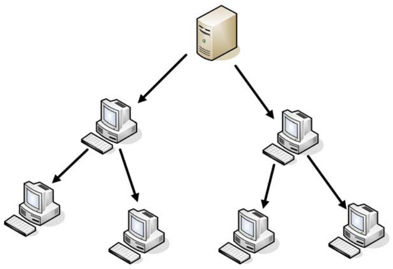

Figure 1. a single-tree-based system

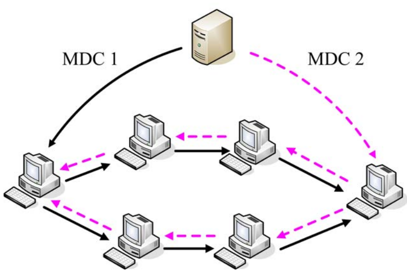

Figure 2. a multiple-tree-based system

CoopNet and SplitStream both organize participating peers into multiple, diverse trees and push each sub-stream of the content through a specific tree. This enables all participating peers to contribute their outgoing bandwidth and also limits the impact of a peer departure to a single tree.

B. systems based on a mesh

Since the tree-based approaches are vulnerable with dynamic group variations, mesh-based unstructured framework has recently become popular owing to its inherent robustness, such as DONet [11]. Now, many P2P live streaming systems adopt mesh-based approach for the construction of the overlay like PPLive, QQLive, MySee, AnySee etc.

In a mesh-based system, a peer can decide a chunk pull schedule that specifies from which peers to download which chunks. Then it will send requests to its neighbors to pull missing chunks. Redundant chunk transmissions can be avoided since a peer only downloads a missing chunk from only one neighbor. Frequent buffer map exchanges and pull requests do incur more signaling overhead and might introduce additional delays in chunk retrieval. To address this problem, some proposals are suggested, like distributing high-capacity nodes uniformly across the system in [2] and network coding technologies in [1] etc.

C. systems based on a tree-mesh-hybrid

One major drawback of tree-based streaming systems is their vulnerability to peer churn. A peer departure will temporarily disrupt video delivery to all peers in the sub-tree rooted at the departed peer. Mesh-based approaches form an unstructured overlay for nodes to exchange data, which greatly enhances the resilience. It however suffers from an efficiency-latency tradeoff, given that the data have to be pulled from mesh neighbors with periodical notifications. Some researches [7] [8] focus on tree-mesh-hybrid systems which utilize a novel and efficient push-pull streaming mechanism to fetch data from neighbors with low latency and high robustness.

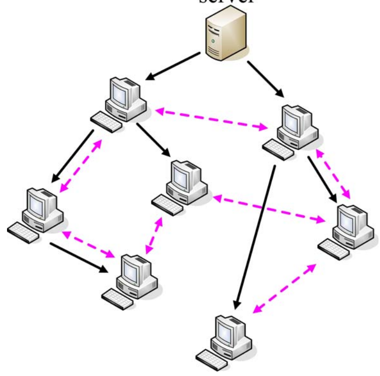

Figure 3. a tree-mesh-hybrid-based system

GridMedia [6], a widely deployed system, achieves near-optimal rates using a push-pull scheme that constructs a set of trees out of a mesh-based unstructured overlay. Their main advantage is that multiple consecutive packets are pushed down the tree along the same paths, resulting in predictable traffic flows and low control traffic.

Another hybrid tree/mesh design mTreeBone [9] that leverages both overlays. The key idea is to identify a set of stable nodes to construct a tree-based backbone, called treebone, with most of the data being pushed over this backbone. These stable nodes, together with others, are further organized through an auxiliary mesh overlay, which facilitates the treebone to accommodate node dynamics and fully exploit the available bandwidth between overlay nodes.

In the hybrid design shown in Fig 3, most media content will be delivered along the tree using push mode. when peers can't obtain data by push approach, e.g. a neighbor node in GridMedia leaves or an unstable node in mTreeBone fails, the node still can easily pull the missing data from its mesh neighbors.

III. DATA DELIVERY CHALLENGES

It is challenging to deliver media content through peer-to-peer network because of the stringent time constraints, which require more efficient and resilient data delivery algorithm. With a large number of P2P streaming protocols proposed, data delivery algorithm generally fall into three strategic categories. tree-based Push streaming strategies organize participating peers into one or more multicast trees, and disseminate streaming content along these trees. In contrast, in mesh-based pull streaming strategies, the streaming content is presented as a series of segments, each representing a short duration of playback. And in tree-mesh-hybrid systems, push-pull streaming strategies are receiving more and more interesting.

A. data delivery in tree-based systems

Similar to an IP multicast tree formed by routers at the network level, users participating in a video streaming session can form a tree at the application layer that is rooted at the video source server. Each node receives the video from its parent peer at the level above and forward the received video to its children peers at the level below. For a large single tree system, the central server might become the performance bottleneck and the single point of failure. To address this, various distributed algorithms, e.g. in [13], have been developed to construct and maintain streaming tree in a distributed way. However, it has been shown that single tree streaming still cannot recovery fast enough to handle frequent peer churn. Another major drawback of the single-tree approach is that all the leaf nodes don't contribute their uploading bandwidth. Since leaf nodes account for a large portion of peers in the system, this greatly degrades the peer bandwidth utilization efficiency.

CoopNet protocol was designed by Microsoft Research and Carnegie Mellon University [15]. Different from others, CoopNet uses MDC (multiple description coding) and P2P approach to deliver live streaming to clients. CoopNet constructs multiple distribution trees for peers, each for one description coding stream. The approach, applying multiple description coding streams over multiple distribution trees, strengthens fault-resilient capability. Whenever a peer fails, the descendents of the fault peer will be blocked only from receiving the coding streams forwarded by the fault peer. They still can receive all other coding streams normally. Consequently, descendents of a fault peer will experience a graceful degradation of stream quality, but not be totally blocked off. Management of the distribution trees is based on a centralized scheme imposed on a set of sever nodes. One of criticism of the CoopNet approach is the centralized control strategy and the control overhead required for management of multiple distribution trees.

B. data delivery in mesh-based systems

In mesh-based pull streaming strategies (e.g., [11]), the media content is presented as a series of segments, each representing a short duration of playback. Every peer maintains a list of neighboring peers, and periodically exchanges segment availability information of streaming buffer maps with its neighboring peers. Based on such information, segments are pulled from appropriate neighbors, in order to meet their playback deadlines. Compared to tree-based push strategies, mesh-based pull strategies take advantage of the philosophy that gossiping segment availability is more resilient to peer dynamics and simpler to implement, which is commonly adopted in BitTorrent-like file swarming systems. However, such an advantage is achieved at the cost of increased delay of distributing streaming content to all participating peers, due to delays caused by periodic buffer map exchanges [3] [4]. Nevertheless, most real-world systems are implemented using pull-based mesh strategies, mainly due to their simplicity.

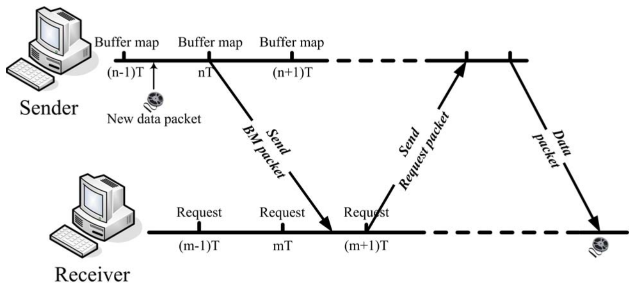

Figure 4. Example of data delivery in mesh-based systems

C. Collaborative Push-Pull Data Delivery in tree-mesh-hybrid systems

Non-hybrid systems have inherent drawbacks as we presented in previous discussion, some P2P streaming reviews [7][8] proposed tree-mesh-hybrid architectures which integrate both tree-based and mesh-based systems' advantages can fully utilize the end node resources, and provide better live media service. Now, we will introduce the data delivery in GridMedia which is in large-scale deployment and the idea of the mTreeBone.

In GridMedia, the streaming packets are classified into pulling packets and pushing packets. Just as the name implies, a pulling packet of a node is delivered by a neighbor only when the packet is requested, while a pushing packet is relayed by a neighbor as soon as it is received. Each node works under pure pull mode in the first time interval when just joining. After that, based on the traffic from each neighbor, the node will subscribe the pushing packets from its neighbors accordingly at the end of each time interval. Author proposed a simple roulette wheel selection scheme to allocate pushing packets in the next time interval to each neighbor (see Fig. 5). The selection probability of a neighbor is equal to the percentage of traffic from that neighbor in the previous time interval. Meanwhile, the lost packets induced by the unreliability of the network link or the neighbors failure will be pulled as well from the neighbors, where the roulette wheel selection scheme is also used to select the suppliers of each packet from neighbors.

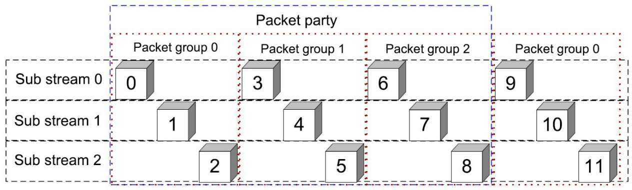

Figure 5. mechanism of a roulette wheel selection in [6]

Shown in the real trace studies by author, the performance of an overlay closely depends on a small set of backbone nodes (also stable nodes). With a stable node identification algorithm to predict the stability of a node, mTreeBone divided the nodes into a small set of stable nodes and non-stable nodes, the stable nodes construct the tree-based backbone of mTreeBone, Other non-stable nodes are attached to the backbone as outskirts. Most of the streaming data are pushed through the tree-based backbone and eventually reach the outskirts, some missing data will be pulled from its mesh neighbors. For example, When an unstable node fails or leaves, it will not affect the data pushed along the treebone. On the other hand, the treebone nodes are stable and seldom leave. even a leave happens, the impact can be remarkably mitigated with the help from the mesh overlay. If node A is affected, it can easily pull the missing data from its mesh neighbors before it re-attaches to the treebone.

IV. PEER SELECTION CRITIRION

Peer selection is a key step to construct an overlay of a P2P live streaming system. In a typical P2P overlay network, peers always come from different autonomous systems, they freely join or depart the P2P network at any time which leads to dynamics and uncontrollability of the overlay. It is challenging to select a right set of peers to construct P2P overlay which can provide a perfect performance. Peers are usually selected to achieve the following three goals: 1) to System Robustness and Network Load Balancing, 2) to Streaming QoS Support, 3) to Overlay Network Locality-Aware.

A. to System Robustness and Network Load Balancing

Some early peer selection strategies considered this goal. And random policy and round-robin policy are the delegation of these approaches. They have a simple protocol, can enhance the robustness of the system and balance the network load.

Following the random policies, peer Y chooses one of its children at random as the target t, and redirects peer X to t. Such a policy requires minimal state at Y. On an average, the tree is expected to be balanced. And round-robin (RR) policies demand Y maintains a list of its children. Y forwards X to the child, t, at the head of the list. The child, t, is then moved from the head to the end of the list. Such a policy requires some state maintenance, also is expected to keep the tree balanced. SpreadIt and Peercast [16] proposed by Standford P2P group adopt these two polices.

B. to Streaming QoS Support

Live P2P media streaming such applications are delay and loss rate sensitive, in case packets from neighbor peers incur large delays or high loss rate, the quality of the application is severely degraded. To the best of our knowledge, parameters determine the quality of service are bandwidth, delay, loss rate or other self-defined standards.

The goal of Overcast's tree algorithm [17] is to maximize bandwidth to the root for all nodes. At a high level the algorithm proceeds by placing a new node as far away from the root as possible without sacrificing bandwidth to the root. The tree protocol begins when a newly initialized node contacts the root of an Overcast group. The root thereby becomes the current node. Next, the new node begins a series of rounds in which it will attempt to locate itself further away from the root without sacrificing bandwidth back to the root. In each round the new node considers its bandwidth to current as well as the bandwidth to current through each of current's children. If the bandwidth through any of the children is about as high as the direct bandwidth to current, then one of these children becomes current and a new round commences. In the case of multiple suitable children, the child closest to the searching node is chosen. If no child is suitable, the search for a parent ends with current.

Another research group from HKUST proposed the concept of power [3] in network given by the ratio of throughput and delay. By maximizing the network power, their solution for live streaming achieves very low delay. Then a simple distributed algorithm was proposed where peers select their parents based on the power concept. The algorithm makes continuous improvement on delay until some minimum delay is reached. New peer i checks its delay with respect to each of potential parents j. Next, peer i requests the residual bandwidth of j and evaluates its power by Equation $\min(R(j),s)/D_i(j)$.Actually, power is the amount of data sent from the peer in unit time. By choosing the parents with large power, it prevents the newcomer from connecting to biased parents, such as parents with large bandwidth but high delay or parents with low delay but insufficient bandwidth.

C. to Overlay Network Locality-Aware

Most P2P systems rely on application layer routing based on an overlay topology, which is largely independent of the Internet routing and topology. Peer selection schemes determined the construction of P2P application layer routing, affected the cross-ISP traffic and utilization of the underlying Internet resource. E.g. one user from Beijing and another user in Shanghai watch the same programme in a P2P streaming system, if peers connect neighbors randomly or no other effective peer selection runs, the users from same ISP wouldn't discover each other. Instead, they may connect the users from other countries which leads to the degradation of the quality of service and also a significant waste of network bandwidth.

Anysee [10] adopts an inter-overlay optimization scheme, in which peers can join multiple overlays based on their locality and delay according to its landmark, so streaming service quality is guaranteed by using the nearest peers, even when such peers might belong to different overlays. The landmark value of each end host is generated by the corresponding geometrical or network information, where the geometrical and network information can be configured by users themselves or obtained from some free IP-to-Geometrical-Address databases. According to the rule, the peers adjacent in landmark space are adjacent in physical network with a high probability. This will improve global resource utilization and distribute traffic to all physical links evenly.

mOverlay [12] presented a new protocol to generate an unstructured overlay to achieve locality characteristic. They build a two level overlay with network proximity. Specifically, a low overhead locating algorithm is proposed to form the overlay with locality characteristic. One of the key characteristics is that the overlay is constructed using dynamic landmarks (versus other existing approaches that use fixed landmarks). without relying on the static landmark, mOverlay solution has strong scalability and high robustness. Meanwhile, the locating complexity is also rather low.

V. CONCLUSIONS

In this article, we reviewed the state-of-art of live P2P media streaming. On one hand, peer-to-peer solutions have shown great advantages in supporting media streaming, as witnessed by their increasingly widespread deployments. On the other hand, there are a number of key technical challenges that need to be overcome before the peer-to-peer solutions can approach the service quality of conventional broadcast and cable TV. In the long term, most of the challenges have to do with the limited amount of access capacity in the Internet. As broadband networks become more ubiquitous and higher-speed, the issues of peer dynamics and incentive will become more important.

ACKNOWLEDGMENT

This work is sponsored by the .National Natural Science Foundation of China (Grant No.60873075) , the China 863 National High Technology R&D Program (Grant No. 2009AA01Z422) and the Engagement Fund of Ministry of Education of China (Grant No:708088).

REFERENCES

[1] Q. Wang, L. Vu, K. Nahrstedt, and H. Khurana, “Identifying Malicious Nodes in Network-Coding-Based Peer-to-Peer Streaming Networks,” in Proc. IEEE INFOCOM, 2010.

[2] F. Picconi and L.Massouli´e, “Is there a future for mesh-based live video streaming?” Proc. of P2P, 2008.

[3] D. Ren, Y. T. Li, and S. H. Chan, “On reducing mesh delay for peer-to-peer live streaming,” INFOCOM 2008, 2008.

[4] X. Hei, Y. Liu and K. Ross, “IPTV over P2P Streaming Networks: The Mesh-Pull Approach,” IEEE Comm. Magazine, pp. 86-92, 2008.

[5] D. R. Choffnes and F. E. Bustamante, “Taming the Torrent: A Practical Approach to Reducing Cross-ISP Traffic in Peer-to-peer Systems,” Proc. of SIGCOMM, 2008.

[6] M. ZHANG, Y. XIONG, Q. ZHANG, and S. Q. YANG, “Optimizing the Throughput of Data-Driven Peer-to-Peer Streaming,” IEEE Transactions on Parallel and Distributed System, 2008.

[7] J. Liu, S.G. Rao, and B Li, “Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast,” Proceedings of the IEEE, vol. 96, pp. 1l-24, 2008.

[8] J. Ghoshal, L. Xu, and B. Ramamurthy, “Network Architectures for Live Peer-to-Peer Media Streaming,” UNL Department of Computer Science and Engineering,Technical Report TR-UNL-CSE-2007-020, 2007.

[9] [9] E. Wang, Y. Xiong, and J. Liu, “mTreebone : A hybrid tree/Meshoverlay for applicationlayer 1ive video multicast,” The Proceedings of the 27th IEEE Intemational Conference on Distributed Computing Systems, 2007.

[10] X. Liao, H. Jin, Y. Liu, M. Ni, and D. Deng, “Anysee: Scalable live streaming service based on inter-overlay optimization,” INFOCOM, 2006.

[11] X. Zhang, J. Liu, B. Li, and T. S. P. Yum, “Coolstreaming/donet: A data-driven overlay network for efficent media streaming,” IEEE INFOCOM 2005, Miami, US, Mar. 2005.

[12] X. Zhang, Q. Zhang, and Z. Zhang, “A Construction of Locality-Aware Overlay Network: mOverlay and its performance,” IEEE JSAC Special Issue on Recent Advances on Service Overlay Networks, 2004

[13] D. Tran, K. Hua, and S. Sheu, "Zigzag: An Efficient Peer-To-Peer Scheme for Media Streaming", Proceedings of IEEE INFOCOM, 2003.

[14] M. Castro, P. Druschel, A. M. Kermarrec, A. R. A. Nandi, and A. Singh, SplitStream: High-bandwidth content distribution in a cooperative environment,” SOSP, 2003.

[15] V. N. Padmanabhan, Wang H J, and Chou P A, “Distributing Streaming Media Content Using Cooperative Networking,” Proceedings of ACM NOSSDAV, Miami Beach, FL, USA, 2002.

[16] H. Deshpande, M. Bawa, and H. Garcia-Molina, “Streaming live media over a peer-to-peer network,” Technical Report 2002-21, Stanford University, 2002

[17] J. Jannotti, D. K. Gifford, and Kirk L Johnson, “Overcast: Reliable multicasting with an overlay network,” Proceedings of the Fourth Symposium on Operating Systems Design and Implementation, 2000.