# An Adaptive Algorithm for Live Streaming using Tree-based Peer-to-Peer Networking in Mobile Environments

Yangjung Kim
Department of Information and Communications Engineering,
Hankuk University of Foreign Studies
zeroplus@hufs.ac.kr

Ilyoung Chong
Department of Information and Communications Engineering,
Hankuk University of Foreign Studies
iychong@hufs.ac.kr

***Abstract*—Media streaming services are becoming extremely popular for end users with mobile devices. Peer-to-peer networking may be the solution to provide efficient streaming services for a large number of concurrent users without infrastructure. The client–server model costs a lot of money to manage and control and client heterogeneity also reduces the overall performance in mobile environments. Peer-to-peer technology can solve this problem and give satisfaction to end users. The technology gives more efficient and scalable solutions than the client-server model. In order to provide live media streaming using peer-to-peer services in mobile environments, a more resilient and cost-effective algorithm to maintain the tree topology is required. In this paper, we provide an adaptive peer-to-peer algorithm for live media streaming in mobile environments. The algorithm uses a minimum parent switching operation using hop count distance to improve the overall performance of the peer tree. We show how the algorithm works for parent selection and peer ordering to enhance the overall performance in the group.**

***Keywords*—Mobile peer-to-peer, Tree-based, Live Media Streaming.**

## I. INTRODUCTION

Media streaming services such as broadcast, multicast, video on demand (VoD) and Internet Protocol Television (IPTV) are getting more popular among mobile users because of their access to advanced broadband access networks. Mobile services for end users are related to live media streaming. Because mobile users are under a highly-moving pattern and have low bandwidth compared to a fixed network, more resilient technology is required to reduce the cost of user control and management. To reduce server overload, a peer-to-peer (P2P) service can be used to provide efficient and cost-effective transport mechanisms without infrastructure. P2P-streaming technology in mobile environments focuses on high scalability and low cost.

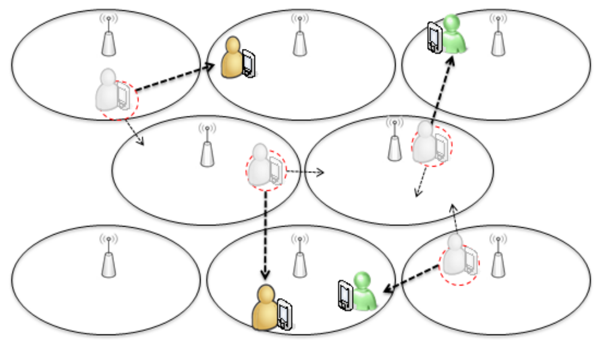


**Figure 1. Example of fast-moving patterns in mobile environmnets.**

In P2P media streaming systems, mobile users would like to watch high-quality video and live streaming such as news, sports and user real-time content. Due to low bandwidth and P2P overlay constraints, P2P algorithms for peer tree topology construction provide low rate video quality to mobile users. Moreover, most algorithms that support seamless streaming cannot improve the quality of media streaming in mobile environments. Seamless media streaming is a more important issue to provide efficient P2P service in the mobile environment. The service in the mobile environment suffers unstable resource and power consumption due to link or node failures, or terrain blockage. The performance of mobile peer-to-peer networks can be reduced by a mobile user's fast moving patterns and the number of different mobile terminals as shown in Fig. 1.

The overall performance of a peer tree is determined by algorithms for peer tree topology construction. So the proposed algorithm reconfigures the P2P data delivery tree on the status of the mobile user. When a mobile peer moves to the place where network resources are limited, all members in the data delivery tree suffer from poor streaming quality. Due to such peers, overall performance of the peer group can be decreased. To overcome the problem, the peer can notify this status to the P2P server and the peer can request a parent peer change. By doing this, all members in the tree maintain good performance and receive efficient live media streaming from the media source in a mobile environment. In this paper, we present a new tree construction algorithm to achieve robust media data transmission in P2P live media streaming under mobile environments. The rest of the paper is organized as follows. In Section II we briefly review related work. In Section III, we study the proposed topology construction algorithm for live media streaming in a mobile environment. We also present operations and procedures for the proposed algorithm. We finally conclude in Section IV.

## II. RELATED WORK

We study how to construct an efficient overlay tree for live media streaming using the proposed adaptive algorithm. There are various algorithms for topology construction in a P2P mechanism. So, this section includes a description of the different types of P2P and existing algorithms for parent selection strategy.

### A. Topology Construction Approaches

The topology construction approach in P2P can be broadly classified into *tree-based* and *mesh-based*. Tree-based approaches [1] and [2] are rooted at the media source. The source manages tree construction information and maintains the tree topology according to a peer joining or leaving. The media streaming data is transmitted along the peer data delivery trees from the source as the root. So, packets are transmitted through one of the transmission trees. Mesh-based approaches [3], [4], [5], [6], and [7] have no global structure to maintain. When a peer joins a membership tree, the peer obtains a list of chosen nodes. The peer randomly chooses neighbor nodes from which to pull media content. The media content is broken down into small pieces and disseminated in a swarm. So neighbor peers use a gossip protocol to exchange buffer map information.

TABLE I. TYPES OF PEER-TO-PEER NETWORKS.

| | P2P Organization | | |
|---|---|---|---|
| | ***Mesh pull*** | ***Tree push*** | ***Hybrid*** |
| Algorithm | Centralized, Decentralized, Superpeer | Loosely structured, DHT, Tree | Hybrid |
| Examples | Napster, Gnutella, KaZaA, GUESS, BitTorrent, CoolStreaming | Freenet, CAN,Chord, Pastry, oStream, SplitStream | Edutellar, Structured Superpeer s, Anysee, AHLSS |

Algorithms such as Centralized, Decentralized and Superpeer are a kind of mesh-based topology construction. DHT and Tree use loosely-structured tree-based topology construction as shown in Table I. Hybrid approaches combine both tree-based and mesh-based approaches. In this mechanism, peers are separated into Super Peers and Normal Peers based on capabilities such as CPU speed, upload bandwidth, and status. The tree-based approach is applied to Super Peers, and the mesh-based approach is applied to Normal Peers.

Because of inefficient peering strategies and video chunk scheduling schemes, mesh-based live streaming architecture has the weak points of long startup delays and playback lag that ranges from several seconds to a couple of minutes. So we focus on the tree-based approach mainly because it is more stable than the mesh-based approach live streaming in maintaining and constructing an efficient topology in mobile environments.

### B. Algorithms for Parent Selection

In order to construct a topology using a tree-based approach, the tree management strategy is most important to maintain the overall performance in the tree. One of the simplest algorithms of tree-based topology construction is the *random algorithm* [11]. A host chooses a random parent with spare bandwidth capacity. This algorithm requires no global information of tree topology so the algorithm efficiently lets new peers join the membership of the tree. *Longest-first algorithm* [13] predicts the future and decides which nodes are stable by remembering how long a peer has stayed in the system. In a *minimum depth algorithm*, a peer selects the parent with the minimum depth using the algorithm. The algorithm gets a tradeoff between simplicity and high overhead. The first high-bandwidth algorithm allows late-arriving peers to preempt the positions of existing peers with a much smaller bandwidth [13]. So the algorithm achieves a minimum bandwidth tree. A *heap algorithm* [13] proposes a sift-up approach during the normal streaming process. The algorithm uses the same algorithms for peer joining and leaving as the minimum-depth algorithm. In the basic algorithm, peers with a high volume of media data can move to a higher level in the tree as shown in Figure 2. So it can be used to give good quality peers.

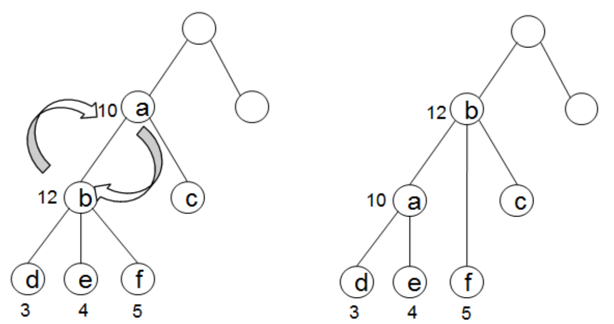



**Figure 2. Illustration of the sift-up operation.**

## III. THE PROPOSED ALGORITHM FOR LIVE STREAMING IN MOBILE ENVIRONMENTS

Due to inefficient topology construction strategies and video chunk scheduling schemes, a mesh-based approach has difficulty in supporting live media streaming services. The cost to maintain the P2P overlay in a mobile environment increases exponentially and frequent joining and leaving operations decrease the overall performance of the peering tree. Moreover, peer heterogeneity can worsen the situation of other peers with good capacity. The approach with maintenance algorithms keeps the global server cost low. So we focus on the tree-based approach and the tree-based approach supports reliable and stable live streaming services to peers in mobile environments.

The proposed algorithm uses hop count distance to estimate the distance between the media source and the client peer destination. If a client peer is moving toward the media source, the distance will become shorter. Finally, the final hop count will be less than the initial value. In mesh-based approaches, one of the key factors to change peers for good performance is delay. The delay metric is a very changeable factor in deciding to change the peer's tree level. Moreover, the metric can be more frequently changed as mobile peers move in mobile environments. We also consider the hop count, so that it can be used to change the tree level in mobile environments.

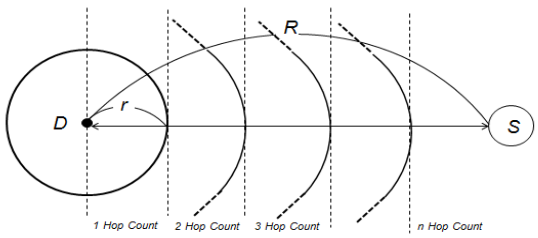



**Figure 3. Hop count distance between media source and client peer.**

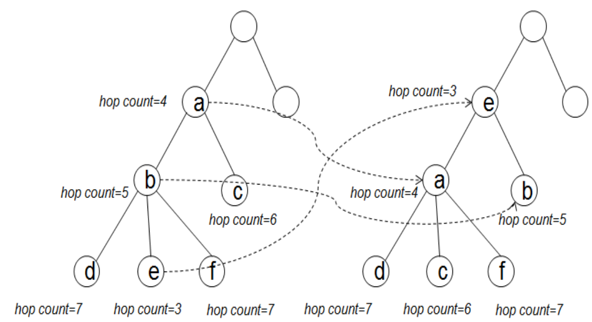
In the proposed algorithm, a peer requests a move to a higher level in the tree topology based on hop count distance first. When peers keep close to the source, they obtain more efficient performance. The algorithm allows a peer with much smaller hop count to preempt the positions of existing peers with a small hop count. So the algorithm achieves a minimum bandwidth tree. As shown in Fig. 3, there are many hop counts {1, 2, 3, … n} between the data source and a client peer. $r$ represents the transmission range of each peer, and is equal to 1 hop count. $R$ is the total distance between data source and a client peer. The minimum distance between a parent and a client peer can be written as

$$minimum\ distance = \min_{i \in N} \sum_{i=1}^{n} r_i \qquad (1)$$

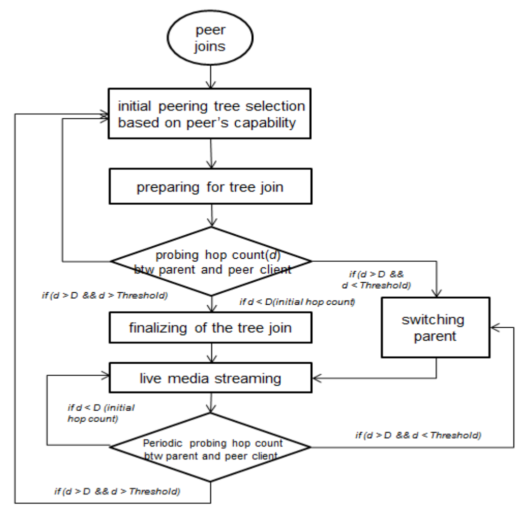
To improve the overall performance of tree members, it is necessary to maintain short tree-depth, that is, a minimum distance between parent and a peer client. If a peer has good capability with sufficient bandwidth, the peer can join the peering tree with high performance. The burden of media sources can be increased by a number of heterogeneous clients. When a tree topology is constructed, the proposed algorithm considers this heterogeneity problem. Depending on the peer's performance, a peer can be classified and assigned to the appropriate tree topology. So the overall performance stays more stable, and prevents performance deterioration due to a peer with poor capacity by classifying the grade of service based on a peer's capabilities.

### *A. The minimum parent swtiching algorithm*

As shown in Fig. 4 (b), when a peer joins the peer group, an appropriate tree class can be classified based on a peer's capability, and then the peer joins the tree topology. If hop count distance is larger than the peer's parent, the proposed algorithm exchanges two nodes to keep a short distance between data source and client peer. This operation gives efficient and quality performance for all members in the tree topology. If a peer finds it difficult to join a good parent, the peer can be allowed to rejoin the tree in the initial position. When a peer moves to the place where bandwidth is limited, the minimum parent switching operation is activated through the periodic distance probing as illustrated in Fig. 4 (a). If the distance ($d$) is greater than the initial distance ($D$) and also greater than the threshold value, the peer can be forced to rejoin.




**(a) Illustration of the minimum parent switching operation.**




**(b) Overall procedure of the proposed algorithm.**

**Figure 4. The switching operation and overall procedure of the proposed algorithm,**

Client heterogeneity leads to serious reduction of the overall performance. When a peer joins a peer tree, first, the peer is classified by its capability as shown in Fig. 4 (b). In this procedure, the client peer is categorized using Algorithm 1. Finally, the peer is assigned to a group with a relevant service level.

Algorithm 1: Pseudocode for peer join using the proposed operation.

**Input:**

$X_D$ : the initial distance of client peer X
*MT_ profile* : the capacity of user's mobile terminal
$L_{pc}$ : the list of peer clients in a peer tree

```
join_flag = false;
do{
 obtain the MT_profile from the incoming peer client(C);
 if( C has enough capability for live streaming){
    C join a quality group or can be assigned to relevant group;
 }
 probe the initial distance between its parent and C;
 push C into the group ;
join_flag = true;
}while(join_flag== false and L_pc != 0)
```

In Fig. 4 (b), the procedures of the proposed algorithm have two operations for probing distance between parent and the peer. A peer with a short distance moves up the peer tree through the ordering procedure in the operation. It should be noted that the minimum parent switching operation has the threshold to prune a peer client with a long distance value. So the total cost to maintain a peer tree can be reduced by the operation. If a peer client has a smaller distance value than its parent and the value is less than the threshold to prune, the tree will be reordered in the switching manner. If a peer client has a shorter distance value than its parent and the value is no less than the threshold, the rejoin process will be applied to the peer client.

Algorithm 2: Pseudocode for the minimum parent switching operation.

**Input:**

$X_D$ : the initial distance of client peer X
$X_d$ : the probing distance of client peer X
$D_{thx}$: the threshold value to limit the tree depth
$L_{pc}$ : the list of peer clients in a peer tree

```
for i = 1 to N do{
  find the client peer with maximum distance in the list L_pc(i);
  if (parent has large distance value than a peer client){
    if(the distance is less than the threshold (D_thx)
      recursive_swap(L_pc(i).parent, L_pc(i).client);
    else
      rejoin a tree process;
  }else{
    join the tree and ready to receive live media streaming
}
}
```

As described in [13], *the random algorithm* has serious overhead due to processing complexity in the protocol. And also *the longest-first algorithm* has easy extension to generate a peer tree in order of peer arriving order. The time-ordered peer tree can generate long tree-depth. Long tree-depth may result in high protocol overhead and frequent node failures. Therefore, long tree-depth is not suitable in mobile environments. So the proposed operations can improve the overall performance of a peer tree in mobile environments and help to construct more efficient short tree-depth topology for mobile peers. Finally, the algorithm allows all peers in the tree to achieve good quality and receive live streaming data in mobile environments.

### B. Implementation considerations

Peers in a P2P live streaming session in a mobile environment can be less stable and dynamic. Therefore, implementation considerations of P2P live streaming deeply focus on tree construction and maintenance. In [15], mesh-based approaches such as PPLive [9], PPStream [10] and CoolStreaming [16] achieve better performance than tree-based approaches in a fixed network. Because of the peer's unexpected join/leave nature in P2P, the streaming tree has constraints to provide a good performance. So we suggest using the tree-based live streaming tree construction in mobile environments. Moreover, the tree needs to be stable as soon as possible to minimize the disruption of live streaming. The proposed algorithm to construct a streaming tree for live streaming in mobile environments has several considerations:

- Classification of joining peer's capability
- Maintaining short tree-depth
- Parent exchange before disruption
- Distance-ordering among participating peers
- Periodic probing between parent and a peer client
- If a peer has no capability to join a tree, the peer should *rejoin* a tree.

## IV. CONCLUSION

In this paper, we have presented an adaptive peer tree topology construction algorithm to achieve robust media data transmission for P2P live media streaming in mobile environments. These users receive P2P VoD data in mesh-based approaches for good performance in a wired network. Many algorithms have been studied in order to reduce the start-up delay of mesh-based approaches. Currently, more studies move to a tree-based approach to achieve better performance in mobile environments. In other words, mobile users want to receive similar speed rates of wired networks. The time-ordered peer tree such as *the longest-first algorithm* and *the random algorithm* can generate long tree-depth. Long tree-depth may result in high overhead and node failures. Therefore, long tree-depth is not suitable in mobile environments. So the proposed operations can improve the overall performance of a peer tree in mobile environments and help to construct more efficient short tree-depth topology for mobile peers. Control overhead to maintain a peer tree topology can be reduced and the proposed algorithm improves the overall performance in a peer tree.

We are currently implementing the proposed algorithm. So we will compare efficiency with existing algorithms such as *random algorithm*, *longest-first algorithm*, *minimum-depth algorithm* and *heap algorithm*. Moreover, we believe that the algorithm can obtain reasonable results by comparisons with related algorithms.

## ACKNOWLEDGMENT

This research was supported by the Korean Ministry of Knowledge Economy under the Information Technology

Research Center support program supervised by the National IT Industry Promotion Agency (NIPA-2011-(C1090-1111-0013)).

## REFERENCES

[1] Y. Cui, B. Li, and K. Nahrstedt, “oStream: Asynchronous Streaming Multicast in Application-Layer Overlay Networks,” IEEE J. Selected Areas in Comm., vol. 22, no. 1, pp. 91-106, Jan. 2004.

[2] M. Castro, P. Druschel, and A.-M. Kermarrec, “SplitStream: High-Bandwidth Content Distribution in a Cooperative Environment,”Proc. Second Int’l Workshop Peer-to-Peer Systems (IPTPS ’03),Feb. 2003.

[3] B.Cohen, “Incentives Build Robustness in BitTorrent,” Proc.Wksp. Econ. Peet-to-Peer Sys., erkeley, CA, June 2003

[4] V. Venkataraman and P. Francis. Chunkyspread: Multi-tree Unstructured Peer-to-Peer Multicast. In The Fifth International Workshop on Peer-to-Peer Systems, February 2006.

[5] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum. DONet/CoolStreaming: A Data-driven Overlay Network for Live Media Streaming. In Proceedings of IEEE Infocom, Miami 2005.

[6] Tran DA, Hua K, Do T (2003) ZIGZAG: An efficient peerto-peer scheme for media streaming. In: Proceedings of IEEE INFOCOM

[7] Magharei N, Rejaie R (2007) Prime: peer-to-peer receiverdriven mesh-based streaming. In: Proceedings of IEEE INFOCOM

[8] Magharei N, Rejaie R, Guo Y (2007) Mesh or multiple-tree:a comparative study of live p2p streaming approaches. In:Proceedings of IEEE INFOCOM

[9] PPLive. PPLive Homepage. http://www.pplive.com

[10] PPStream. http://www.ppstream.com

[11] K. Sripanidkulchai, A.Ganjam, B.Maggs and H.Zhang. The Feasibility of Supporting Large-Scale Live Streaming Applications with Dynamic Application End-Points. Proc. Of ACM SIGCOMM, 2004, Portland, Oregon, USA.

[12] R. Lobb, A. P. Couto da Silva, E. Leonardi, M. Mellia, M. Meo,“Adaptive Overlay Topology for Mesh-Based P2P-TV Systems” ACM NOSSDAV 2009, Williamsburg, Virginia, June 2009

[13] Stephen A. Jarvis, Guang tan, Daniel P.spooner, “Construction reliable and efficient overlays for P2P live media streaming”, I.J. of SIMULATION Vol. 7 No.2

[14] Runzhi Li, Qishi Wu, Yunyue Lin, Xukang Lu, Zongmin Wang, “On topology construction in layered P2P live streaming networks”, NOMS2010

[15] Magharei N, Rejaie R, Guo Y, "Mesh or multiple-tree:a comparative study of live p2p streaming approaches", IEEE INFOCOM

[16] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, “DONet/CoolStreaming: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming,”in IEEE INFOCOM, vol. 3, Mar. 2005, pp. 2102 – 2111