

LocalTree: An Efficient Algorithm for Mobile Peer-to-Peer Live Streaming

Bo Zhang S.-H. Gary Chan
Department of Comp. Sci. & Eng.
The Hong Kong Uni. of Sci. & Tech.
Clear Water Bay, Hong Kong
Email: {zhangbo, gchan}@cse.ust.hk

Gene Cheung
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo, Japan 101-8430
Email: cheung@nii.ac.jp

Edward Y. Chang
Director of Research
Google Research
Beijing 100084, China
Email: edchang@google.com

Abstract—Mobile peer-to-peer (P2P) technique is a promising approach to provide live video streaming. In mobile P2P streaming, mobile devices relay their received packets in a multi-hop manner by means of broadcasting in a secondary channel (such as Wi-Fi or bluetooth). We consider the general scenario that both wireless channel and nodes may not be reliable, i.e., packets may be lost in channel transmission and a node may forward its received packets probabilistically (due to, for examples, its power and incentive concerns). Under this scenario, we study how to achieve broadcasting minimizing the energy consumption in the network while meeting a certain stream quality requirement (in terms of received packet loss rate). We first formulate the problem under study, and propose and study a distributed algorithm called LocalTree which addresses the problem. LocalTree takes advantage of stable clusters of users to optimize the construction of its streaming overlay. It combines the strengths of both tree-based and mesh-based algorithms, and is simple and effective. Simulation results show that LocalTree performs similar to tree-based algorithm for a rather stable network, and exhibits robustness similar to unstructured algorithm if the network is dynamic.

I. INTRODUCTION

In traditional live video streaming over mobile networks, each client actively *pulls* the stream from a content server over a wireless wide area network (WWAN). This approach suffers from the weakness that the base station can easily become a bottleneck as the number of users increases. Moreover, it is financially expensive, because video is usually of high bitrate and service providers often charge pullers based on data usage.

With the advancement of mobile device capacities in terms of memory and processing, *cooperative mobile peer-to-peer (P2P) streaming* has been proposed to cost-effectively overcome the above problems. In such approach, peers within a geographical area form an ad-hoc group via their (free) secondary channel (such as Wi-Fi or Bluetooth). A node pulls packets of the video stream from the base station, and shares the received packets

with others by means of broadcasting. In such a way, the streaming cost is amortized among all the peers in the network.

In this paper, we consider video broadcasting in a mobile network. We consider the realistic scenario that connections between peers may not be reliable, and hence relayed/broadcast packets may be lost in the wireless channel (due to, for example, channel fading). Furthermore, a broadcast node may not relay all its received packets to its neighbors, due to its power or incentive concerns. Under these conditions, broadcasting meeting a certain video quality requirement (in terms of loss rate) is an important issue.

We first formulate the problem to address the above. For power conservation and network lifetime extension, our objective is to minimize the energy used in packet transmission in the network. Given the problem formulation, we then propose an efficient and distributed algorithm called *LocalTree*. In LocalTree, *stable* groups of users (e.g., connected users moving in similar direction with each other) are identified in a distributed manner, and then a local tree is constructed for each of such groups. (Indeed stable groups do exist in reality as observed in [1], [2], which show that mobile nodes often exhibit correlated physical movements.) For unstable peers and for inter-group stream distribution, LocalTree adapts unstructured algorithm to construct a streaming mesh.

In LocalTree, peers are organized into two tiers, the base tier and the tree tier. Peers are first connected in a simple unstructured mesh in the base tier. Groups of relatively stable nodes are then identified (by the node and link conditions). They are then connected following a modified tree construction algorithm in the tree tier. Figure 1 shows an example of a network resulted from using LocalTree. Node 1 pulls the stream from the base station. The connections in the base tier (unstructured mesh) are indicated by the dotted lines. Nodes 2 and 3 are the roots of the two disjoint local trees C1 and C2 as indicated by solid lines at the tree tier. While most of the packets are distributed along the trees in C1 and

This work was supported, in part, by the General Research Fund from the Research Grant Council of the Hong Kong Special Administrative Region, China (611209), the Hong Kong Innovation and Technology Fund (ITS/097/09FP), and Google Mobile 2014 and Faculty Research Awards.

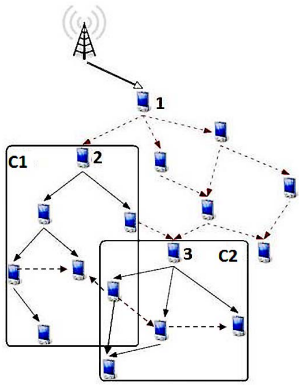


Fig. 1. Illustration of two *localtrees* formed in a mobile network.

C2, in order to meet the QoS requirement, there are some supplementary packet transmissions as indicated also by the dotted lines.

To the best of our knowledge, this work is the first to consider the more realistic scenario of link and node reliability to achieve QoS for mobile P2P streaming. Our approach is that the base tier provides basic robustness, while the tree tier performs further optimization. Our contributions are summarized below:

- We formulate the problem under study, which is live stream distribution to minimize energy consumption meeting a certain packet loss requirement under the conditions that connections and nodes may not be reliable.
- We propose LocalTree, a distributed and efficient algorithm to address the problem.
- We have conducted extensive simulation study on LocalTree. Our results show that LocalTree is indeed efficient, i.e., it meets the QoS requirement by achieving robustness similar to mesh when the network is dynamic, and optimality similar to a global tree when the network is stable.

In general previous work on the overlay topology for mobile P2P live streaming can be divided into two categories: unstructured mesh and global tree. In unstructured mesh, each node makes its own relay decision based on local information [3]. They are not generally optimal in terms of energy consumption. Global tree, on the other hand, seeks to build an optimal spanning tree for stream dissemination [4]. However, it is not efficient for a network with node and link dynamics because of its centralized nature and its complexity. Multicast Overlay Spanning Tree (MOST) connects Multi-Point Relay (MPR) nodes selected by OLSR protocol to form an overlay spanning tree [5]. However this is not energy-efficient, because there are many relays (MPRs), since each node chooses its own MPR set independently. Seldom has any mobile P2P streaming work considered how to achieve QoS under the (realistic) scenario that both nodes and links may not be reliable, as we consider here.

The paper is organized as follows. We first formulate the problem under study in Section II. LocalTree, our proposed two-tier distributed algorithm, is presented in Section III. We present illustrative simulation results and conclusion in Sections IV and V, respectively.

II. PROBLEM FORMULATION

A. Preliminaries

We model the network as a connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of all peers, and \mathcal{E} is the set of the directed edges corresponding to the transmission range between peers, i.e., $(i, j) \in \mathcal{E}$ iff peer i can directly reach peer j (we call peer j a neighbor of peer i). A single source node p pulls the content from the base station and re-broadcasts it to the other peers, where $p \in \mathcal{V}$. (the selection of p is outside the scope of this paper; an approach has been discussed in [3].) One-hop broadcast is assumed in this paper.¹

Note that \mathcal{G} may change over time due to nodes join/leave/failure or mobility. We define the following independent parameters to model the network, for $i, j \in \mathcal{V}$:

- $b_i \in \{0, 1\}$ is the variable indicating whether node i is a broadcaster or not, i.e., $b_i = 1$ if node i is a broadcaster, and 0 otherwise.
- $w_i \in [0, 1]$ is the *relay probability* that node i relays its received packets through broadcasting. Note that even if a node is a selected broadcaster, it may not re-broadcast *all* its received packets due to its concerns on, for examples, battery levels and incentive. w_i hence captures the reliability of the node.
- $c_{i,j} \in [0, 1]$ is the success probability of packet transmission on link (i, j) , where $(i, j) \in \mathcal{E}$. $c_{i,j}$ hence captures the reliability of the connection between i and j . Both w_i and $c_{i,j}$ hence are the stability parameters of the network.
- $\varepsilon_i \in [0, 1]$ is the packet loss rate at node i . To meet a certain loss rate requirement, such loss rate should be less than a certain value $\hat{\varepsilon}$, i.e., $\varepsilon_i \leq \hat{\varepsilon}, \forall i \in \mathcal{V}$.

Note that our formulation is sufficiently general to capture node mobility effect through parameters w_i and $c_{i,j}$. For a highly dynamic network, both w_i and $c_{i,j}$ may be low. On the other hand, if the network is rather stable and nodes are willing to share, both w_i and $c_{i,j}$ can be close to 1. (Traditional works on global tree assume the special case both that w_i and $c_{i,j}$ are 1.)

B. Problem Formulation

By definition, we require $b_p = w_p = 1$ and $\varepsilon_p = 0$ for source node p .

A packet is received at node i , where $i \neq p$. Let peer j be a neighbor of i . Clearly, the event that the packet is successfully received at j is given by the occurrence of the following independent events: 1) i receives the

¹One-hop broadcast means delivering a packet to all directed neighbors as defined in \mathcal{E} .

packet without error. This is with probability $(1 - \varepsilon_i)$, which is, given meeting its QoS requirement, at least $(1 - \hat{\varepsilon}_i)$; 2) i is the broadcaster and relays the packet. This is of probability $b_i w_i$; and 3) the broadcast packet reaches j , which is with probability $c_{i,j}$. Therefore, the loss rate at j from i is at most

$$1 - (1 - \hat{\varepsilon})b_i w_i c_{i,j}. \quad (1)$$

As the neighbors are independent, to meet the QoS requirement we need, $\forall j \in V$,

$$\prod_{i:(i,j) \in \mathcal{E}} (1 - (1 - \hat{\varepsilon})b_i w_i c_{i,j}) \leq \hat{\varepsilon}. \quad (2)$$

In order to minimize the energy consumption (and hence maximizing the network lifetime), our optimization objective is to select broadcasters in constructing a cooperative network to minimize the packet transmissions, i.e.,

$$\min \sum_{i \in V} b_i w_i, \quad (3)$$

subject to Equation (2).

Note that, due to QoS requirement, in our optimal solution a peer j may have multiple supplying parents in the constructed overlay. This means that the resultant overlay may be a directed acyclic graph (DAG).

III. LOCALTREE

We now present in detail our distributed algorithm LocalTree to address the problem. The proposed algorithm consists of two parts, one for the *base tier* and the other for the *tree tier*.

A. Algorithm for the Base Tier

We first construct an unstructured mesh as the base tier. The reason for including an unstructured overlay is to accommodate time-varying changes and node/connection unreliability of the network topology. In the base tier, peers utilize only local neighbor information to make independent distributed decisions on whether to rebroadcast a packet or not.

A peer periodically broadcasts its buffermap, which is simply a bitmap indicating whether it has received the corresponding packet ID or not. As a result, a node knows the received states of its neighbors. For node i , we define the *occupancy factor* of packet k as the fraction of its neighbors with the packet, i.e.,

$$B_i^k = \frac{\# \text{ of neighbors have received packet } k}{\text{Total \# of neighbors of } i}. \quad (4)$$

Clearly node i , once receiving a packet k , has to decide whether to broadcast it or not. In order to reduce the number of broadcasters (and hence energy consumption), it may broadcast only if B_i^k is below a certain threshold B . To avoid collision, it should wait for a certain time d_t to see if any "better" node will have broadcast the same packet. Obviously, a "better" node

Algorithm A Packet forwarding for the base tier

Given: Node i receives packet k which falls within its buffermap window
if $B_i^k \leq B$ **then**
 Wait for a delay d_t ;
 Update B_i^k ;
 if $B_i^k \leq B$ **then**
 Broadcast packet k ;
 end if
end if

is the one with lower occupancy factor of the packet, because its broadcast can benefit higher fraction of neighbors. Therefore, d_t should be monotonically increasing with B_i^k . An example of d_t is $d_t = \eta B_i^k$, where η is the maximum delay for such operation. We summarize in Algorithm A the detailed steps for the base tier according to the description above.

B. Algorithm for the Tree Tier

The algorithm for the base tier constructs a mesh. It provides a network for further optimization of the energy consumption. Given that some clusters of users may exhibit stability (in terms of the nodes and their connections), the base-tier mesh is further optimized by the tree-tier algorithm.

Recall that there is a puller p in the network. A node continuously updates its hop distance to p , which can be done by taking the minimum of the hopcounts to p of the most recent packets it received. If it re-broadcasts a received packet, it increments the hop count by 1 and piggybacks this information to the packet.

A stable hop distance of node i signal that its neighborhood may be stable and hence a local tree may be formed. In this case, node i claims itself to be the local tree root by sending a *LocalRoot* message containing its ID, a unique localtree id, and a TTL controlling the broadcast scope and hence tree depth.

In order to avoid collisions and to favor the nodes closer to p to claim the tree root earlier, the sending of the *LocalRoot* message at node i is delayed by a certain time proportional to i 's hop distance. During this delay, the following conditions may occur:

- There is no other *LocalRoot* message received. In this case, node i becomes the local root.
- There is another *LocalRoot* message received. In this case, node i examines the hop distance of the node ID of this message.
 - If the hop distance of i is lower, it ignores the *LocalRoot* message;
 - Otherwise, it needs to decide whether its upstream forwarder j of the message is stable or not. This is done by examining the loss rate due to j , i.e., $1 - (1 - \varepsilon_j)w_j c_{j,i}$. If it is lower than $a\hat{\varepsilon}$, for

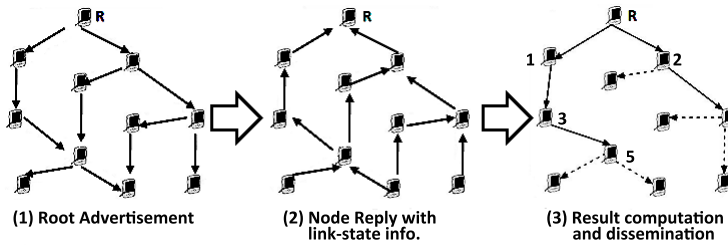


Fig. 2: Illustration of local tree formation.

some $0 \leq \alpha \leq 1$, j is considered stable and the node i joins the root-claimer. Otherwise, node i ignore the *LocalRoot* message.

Once node i decides to join a root, it first drops its own pending *LocalRoot* message (if one exists). It then periodically sends its stability parameters (w_i and $c_{i,j}$) and link-state information to the root.

A root hence collects link-state information and stability parameters from all the joined nodes. It computes the local tree using this information. The computed result is then disseminated to all its joined members.

For the local tree computation, we adapt a modified greedy algorithm proposed in [6]. The proposed algorithm tries to greedily grow a minimum connected dominating set (MCDS) from a given vertex of a graph. The modification is that, instead of simply marking each node as covered or uncovered, each node remains uncovered and keeps evaluating Equation (2) until the QoS is met. In general, the algorithm results in a DAG. For the special case where w_i and $c_{i,j}$ are both equal to 1, the result reduces to a maximum leaf tree.

We illustrate in Figure 2 how a local tree is formed. First, node R claims itself to be the local tree root, and advertises his claim. Second, nearby nodes who wish to join reply with their stability parameters and link-state information, i.e, direct neighbors of the sender. Third, node R computes an optimized tree with all the collected information. Finally, the computation result is distributed, and a local tree (possibly supplemented with a mesh) is formed.

IV. ILLUSTRATIVE SIMULATION RESULTS

In our simulation, a certain number of nodes are randomly placed in a square area (200x200 units), each with a certain transmission range (50 units). One of them is randomly assigned the *puller*. Unless otherwise stated, we use the following baseline parameters: number of nodes = 25, $B = 0.33$, $w_i \sim U[0.7, 1.0]$, $c_{i,j} \sim U[0.8, 1.0]$, $\hat{\epsilon} = 0.05$, $\alpha = 0.7$, and buffermap size = 40. Because the performance depends on the product of $w_i c_{i,j}$, we let $\Gamma_{i,j} = 1 - w_i c_{i,j}$, which can be interpreted as the packet loss probability from i to j . Obviously the network is not reliable for high value of $E[\Gamma_{i,j}]$.

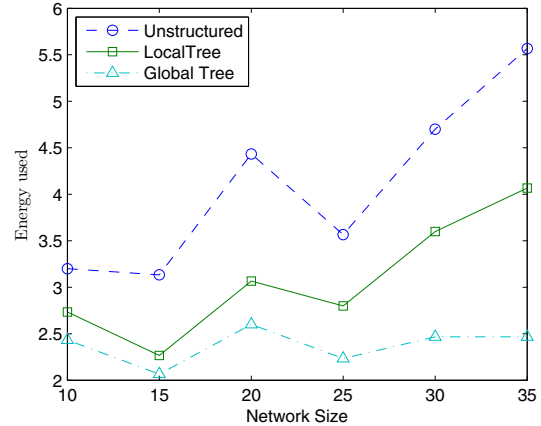


Fig. 3: Energy cost vs. network size.

We evaluate using the following metrics:

- *Energy cost* for stream distribution, which is measured by the average of $\sum_{i \in V} b_i w_i$ over all simulation runs.
- *Number of broadcasters*, which is the mean number of nodes relaying packet over all simulation runs.
- *Packet loss rate*, which is the ratio between the number of lost packets and the total number of transmitted packets, averaged over all nodes. Note that some packets may not be received in time before the buffermap window shifts to a new position. In this case these packets miss their playback deadline and are considered lost.

We compare LocalTree with two other algorithms: a modified global tree algorithm where link-state information is globally for each node to compute a global optimal DAG, and a generic unstructured mesh algorithm using base-tier algorithm.

Figure 3 shows the energy cost for different network size. For this simulation both w_i and $c_{i,j}$ are set to be 1, i.e., the energy cost is equal to the number of broadcasters used. As we can see, for the global tree algorithm, the consumed energy does not sensitively increase with the number of nodes. This is because, given a fixed size of area, more nodes do not necessarily mean more broadcasters. However, for the unstructured algorithm, this cost increases with larger network size. This is because more nodes are likely to trigger unnecessary and inefficient relays. LocalTree performs much better than the unstructured algorithm, due to its optimization by the proposed *tree tier*.

We plot in Figure 4 the numbers of broadcasters versus network unreliability given by $E[\Gamma_{i,j}]$. Global tree algorithm chooses the least number of broadcasters in stable networks. However in instable networks (i.e, when $E[\Gamma_{i,j}]$ increases), this number rises quickly. This is because each node is more likely to select be a broadcaster. Increase in transmission unreliability only marginally

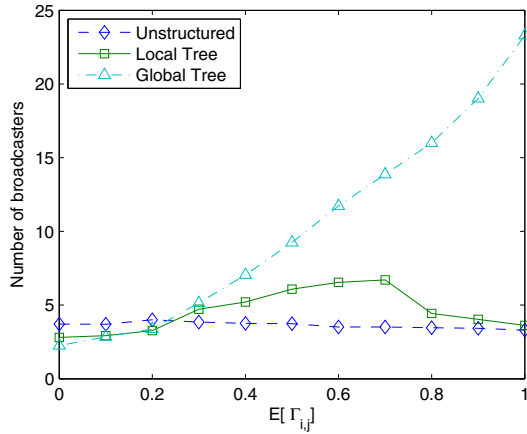


Fig. 4: Total number of broadcasters occurred vs. $E[\Gamma_{i,j}]$.

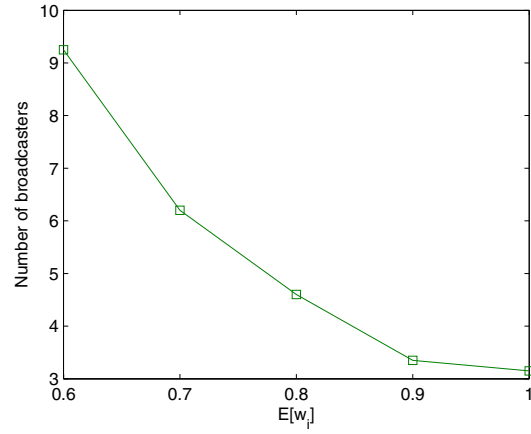


Fig. 6: Total number of broadcasters vs. $E[w_i]$.

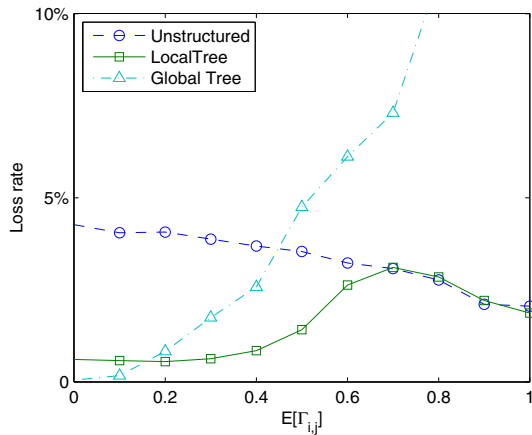


Fig. 5: Packet loss rate vs. $E[\Gamma_{i,j}]$.

affects unstructured algorithm, because a broadcaster is more likely to cover more nodes. LocalTree combines the strengths of both schemes, i.e., it exhibits similar optimality as global tree in stable network, while achieving similar performance as a mesh in instable network.

Figure 5 illustrates packet loss rates versus $E[\Gamma_{i,j}]$. Loss rate of global tree algorithm is the lowest in stable network, but rises drastically as the nodes/connections becomes unreliable. LocalTree scheme performs similarly to global tree in stable network, and quickly converges to unstructured approach in instable network.

Figure 6 shows the number of broadcasters versus $E[w_i]$, where w_i are distributed uniformly with upper bound 1. $c_{i,j}$ for all links are fixed to 1. The number of broadcasters decreases with w_i , because nodes are more likely to share all its received packets. Such drop is quite significant initially, showing the importance of node sharing in reducing the number of broadcasters in the network.

V. CONCLUSION

In this paper, we have considered the problem of video broadcasting in mobile P2P network, under the (realistic)

scenario that both nodes and links may not be reliable. The objective is achieve broadcasting meeting a certain loss requirement. We have formulated the problem, and presented LocalTree, a scalable and energy-efficient algorithm, to address the problem. The algorithm operates with two tiers. In the *base tier*, mobile nodes follow a fully distributed and independent relay algorithm, while in the *tree tier*, relatively stable groups are identified and an optimized local DAG is computed. With the two-tier operation, LocalTree is able to adapt different network dynamics.

We have conducted extensive simulation study on LocalTree performance. Simulation results show that LocalTree achieves optimality similar to tree-based algorithm for a rather stable network, and exhibits robustness similar to unstructured algorithm if the network is dynamic.

REFERENCES

- [1] K. Blakely and B. Lowekamp, "A structured group mobility model for the simulation of mobile ad hoc networks," in *MobiWac '04: Proceedings of the second international workshop on Mobility management & wireless access protocols*. New York, NY, USA: ACM, October 2004, pp. 111–118.
- [2] D. Ciullo, V. Martina, M. Garetto, and E. Leonardi, "Impact of correlated mobility on delay-throughput performance in mobile ad-hoc networks," in *INFOCOM'10: Proceedings of the 29th conference on Information communications*. Piscataway, NJ, USA: IEEE Press, 2010, pp. 1738–1746.
- [3] M.-F. Leung and S.-H. G. Chan, "Broadcast-based peer-to-peer collaborative video streaming among mobiles," *Broadcasting, IEEE Transactions on*, vol. 53, no. 1, pp. 350–361, march 2007.
- [4] O. Badarneh, M. Kadoch, and A. ElHakeem, "Multilayered video multiple trees multicast algorithms for heterogeneous wireless ad hoc networks," in *Network Computing and Applications, 2008. NCA '08. Seventh IEEE International Symposium on*, jul. 2008, pp. 220–223.
- [5] G. Rodolakis, A. M. Naimi, and A. Laouiti, "Multicast overlay spanning tree protocol for ad hoc networks," in *WWIC '07: Proceedings of the 5th international conference on Wired/Wireless Internet Communications*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 290–301.
- [6] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, pp. 374–387, 1998, 10.1007/PL00009201. [Online]. Available: <http://dx.doi.org/10.1007/PL00009201>