

OFFERING DIFFERENTIATED SERVICES IN PEER-TO-PEER MULTIMEDIA MULTICAST

Kan-Leung Cheng Xing Jin S.-H. Gary Chan

Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

ABSTRACT

In traditional overlay multicast network, all peers are treated as equal regardless of their importance and contribution to the network. In this paper, we consider that each user may have different cost function depending on his privilege, delay to source, available bandwidth, etc. We propose a fully distributed and scalable protocol to construct an overlay tree to minimize the overall cost of the users. It uses aggregation to account for the total cost of one's descendants, and reduces tree cost by a continuous improvement process. Through simulation, we show that our protocol converges reasonably fast. By comparing with other schemes, our resultant overlay tree offers differentiated services to users by appropriately taking into account individual user cost functions.

1. INTRODUCTION

There has been much recent interest in peer-to-peer (P2P) multimedia streaming to a group of users, where users play the role of both a server and a client to help each other in streaming data.¹ Applications of interest include distance learning, Internet TV (IPTV), video conferencing, etc. Previous protocols have not sufficiently considered service differentiation based on user privilege and requirement, i.e., all users are treated as equal regardless of their importance and contribution to the network. We consider in this paper how to offer differentiated services to users for peer-to-peer streaming.

Indeed, there are many applications where differentiated services are best to offer. For example, in a distance learning application, the students taking the course should enjoy better streaming quality (in terms of delay, loss rate, etc) than those auditing it. In an IPTV network, the premium users who pay more should also enjoy better service than others. Moreover, if the user streaming quality depends on his contribution (e.g., upstream bandwidth donated), it would become an incentive

This work is supported, in part, by the Area of Excellence in Information Technology of the University Grant Council (AoE/E-01/99), Competitive Earmarked Research Grant of the Research Grant Council (HKUST6156/03E) in Hong Kong and the Innovation and Technology Commission of the Hong Kong Special Administrative Region, China (GHP/045/05).

¹In this paper, we use "users", "peers", and "nodes" interchangeably.

mechanism for the users to donate more, making the P2P system perform better. Consequently, the resource can be more effectively allocated and utilized.

We hence propose and investigate a peer-to-peer streaming protocol which offers differentiated services. We consider that every user has his own cost function which may depend on his importance in the network, distance to the source, his contribution to the network, etc. The objective is to minimize the overall cost of all users, given their individual functions. We propose a fully distributed protocol to achieve it. Our protocol uses aggregation to account for the total cost of one's descendants. By means of an improvement algorithm using the aggregate information, the tree cost can be continuously reduced close to its optimum.

Our protocol has the following properties:

- *Service differentiation*: The system appropriately differentiates users by providing better service to high-priority users (i.e. users with a steeper cost function);
- *Efficiency*: The overlay constructed is efficient in terms of stress (i.e., network bandwidth utilization) and stretch (streaming delay);
- *Scalability*: The protocol is scalable in the sense that it is fully distributed, i.e., a peer does not need to have a complete and global knowledge on the current network and overlay structure. Furthermore, its performance scales well with the group size; and
- *Adaptability*: The protocol is able to adapt to changes in network conditions (such as latency and bandwidth) and member dynamic (such as joining and leaving) to reduce the tree cost of the system.

We evaluate our protocol by simulation with p2pns [1]. Our results show that the protocol achieves good network performance in term of stress and stretch. In addition, the overlay tree converges quite fast close to the optimum. Our protocol effectively offers differentiated services to users by appropriately taking into account their individual cost functions.

We briefly review previous work as follows. Most overlay multicast protocols proposed, such as HMTP, OMNI, NICE,

etc., assume that all users are equal [2, 3]. Our work differs by allowing users to have different cost functions related to their performance requirements on the streaming quality, and addressing how to provide differentiated quality of service in such context. OverQoS has been proposed to smooth packet losses, prioritize packets and provide bandwidth guarantees for different users [4]. However, it has not addressed how to construct an *overlay tree* for differentiated services, which we investigate here. Ahsan Habib et al. propose a streaming system which provides differentiated services by controlling the set of “suppliers” as potential parents of a node [5]. The streaming quality is controlled indirectly through the choice of suppliers. Our protocol, on the other hand, directly changes the structure of the overlay tree in order to achieve better service differentiation.

This paper is organized as follows. We present in Section 2 the problem formulation and protocol description, followed by its evaluation in Section 3. We conclude in Section 4.

2. PROBLEM FORMULATION AND PROTOCOL DESCRIPTION

2.1. Problem formulation

In this section, we formulate our problem and its objective. The overlay network is modeled as a complete directed graph denoted by $G = (V, E)$, where V is the set of vertices representing peers and $E = V \times V$ is the set of edges representing the overlay links. Among the peers, there is one source s with the set of receivers given by $V \setminus \{s\}$. We denote the latency $d(i, j)$ as the unicast path latency from peer i to peer j . There is an out-degree bound on each peer imposed by either its processing capacity or its upstream bandwidth.²

The overlay tree is a directed spanning tree of G rooted at the source, with the edges directed away from the root. The parent of peer i (other than the source) is denoted by $parent(i)$. The set of children of peer i on the tree is denoted by $children(i)$. The source-to-peer overlay delay of peer i is D_i , which is the summation of all the unicast latencies along the overlay path from s to peer i on the tree.

We consider the cost function of a user as a function $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$, denoted as $f(\vec{a}, \vec{x})$, where $\vec{a} \in \mathbb{R}^n$ is a n -tuple vector of some cost parameters which may be different for different users. \vec{a} is the weighting factor of some function on $\vec{x} \in \mathbb{R}^m$, which is an m -tuple variable of quality metrics indicating the quality of service the user is currently receiving. For example, \vec{x} may include source-to-peer delay, PSNR of the video, jitter, loss rate, etc. f is a monotonically increasing function on the components of \vec{x} . We further consider that f satisfies the following properties:

²For a bound imposed by the upstream bandwidth, let B (bits/s) be the bitrate of the data stream. Further let the upstream bandwidth of peer i be u_i (bits/s). Then peer i can forward data to at most $\lfloor u_i/B \rfloor$ children, which is its out-degree bound.

Shift property: Given $\vec{a} \in \mathbb{R}^n$ and $\vec{x} \in \mathbb{R}^m$, f satisfies $f(\vec{a}, \vec{x} + \vec{d}) = f(\vec{a}_{\vec{d}}, \vec{x})$, for any $\vec{d} \in \mathbb{R}^m$ and some $\vec{a}_{\vec{d}} \in \mathbb{R}^n$. We call $\vec{a}_{\vec{d}}$ the shifted cost vector of \vec{a} , denoted as $shift(\vec{a}, \vec{d}) = \vec{a}_{\vec{d}}$.

Summation property: For any $\vec{a}_1, \vec{a}_2 \in \mathbb{R}^n$ and $\vec{x} \in \mathbb{R}^m$, f satisfies $f(\vec{a}_1, \vec{x}) + f(\vec{a}_2, \vec{x}) = f(\vec{a}_3, \vec{x})$ for some $\vec{a}_3 \in \mathbb{R}^n$. We denote this as $\vec{a}_3 = \vec{a}_1 \oplus \vec{a}_2$.

These properties of f , though appeared contrived, are in fact quite general. For example, if $m = 1$, then the Fourier series $f(\vec{a}, x) = a_0 + \sum_{k=1}^n (a_{k,s} \sin k\omega_0 x + a_{k,c} \cos k\omega_0 x)$ where $\vec{a} = (a_0, a_{1,s}, a_{1,c}, a_{2,s}, a_{2,c}, \dots)$ or a polynomial function $f(\vec{a}, x) = \sum_{k=0}^n a_k x^k$ clearly satisfies these properties.³ Therefore, for any arbitrary cost function, we may expand or approximate it by a Fourier series, polynomial function, or any other series of similar type (usually in the form of summing a sequence of eigen functions).

Given f , our objective is to minimize the total cost of the overlay tree T as given by $C(T) = \sum_{v \in V} f(\vec{a}_v, \vec{x}_v)$, subject to the out-degree bounds of the users. This problem is NP-hard because it is a generic version of “degree-constrained minimum average-latency problem”. We hence describe a distributed heuristic to solve the problem.

For concreteness and ease of exposition, in the following, we consider using the source-to-peer delay D as the quality metric (i.e. $m = 1$), though it is straightforward for our protocol to be extended to multiple metrics.

2.2. Protocol description

In this section, we present a distributed heuristic for the above optimization problem. A new incoming node first contacts a rendezvous point to get a random list of current members. It starts off by selecting a unsaturated members as its parent (by probing the listed members). Each peer i in the network keeps its overlay path from the source to itself, obtained simply by appending its parent path with its own ID. It also keeps its total delay D_i from the source on the tree, given by $D_i = D_{parent(i)} + d(i, parent(i))$. Furthermore, it also keeps the aggregate cost vector \vec{A}_i of the subtree rooted at itself such that $f(\vec{A}_i, D_i)$ is the total cost of all its descendents and itself. Using the properties of f , it can be shown that (by mathematical induction)

$$\vec{A}_i = \vec{a}_i \oplus \bigoplus_{j \in children(i)} shift(\vec{A}_j, d(i, j)),$$

where

$$\bigoplus_{j=1}^n \vec{a}_j = \vec{a}_1 \oplus \vec{a}_2 \oplus \dots \oplus \vec{a}_n.$$

Each peer (except the source) periodically runs an improvement algorithm in a distributed manner to find a new parent so as to reduce the total cost of the tree. In the algorithm, a peer maintains a priority queue, called unchecked

³This can be generalized to $m \geq 1$ by using $f'(\vec{a}, \vec{x}) = \sum_{i=0}^n f(\vec{a}, x_i)$.

Algorithm 1 Improvement algorithm: improve(i : node)

```
 $n \leftarrow$  a random node picked on root path of  $i$ 
if  $n$  can accept child AND  $D_i > D_n + d(i, n)$  then
  node  $i$  joins  $n$  as its parent
  return
end if
Except the ancestor of  $i$ , enqueue all other children of  $n$  into UPQ
while UPQ is not empty do
   $j \leftarrow$  extractMin(UPQ)
  if  $D_i > D_j + d(i, j)$  then
    if  $j$  can accept child then
      node  $i$  joins  $j$  as its parent
      return
    else  $\{j$  is saturated $\}$ 
      Enqueue all children of  $j$  into UPQ
    end if
  end if
if  $r(i, parent(j)) + r(j, parent(i)) > \theta$  then
  swap the parents of  $i$  and  $j$ 
  return
end if
end while
```

peer queue (UPQ), which stores peers to be examined in ascending order of their source-to-peer delay. The queue is initialized by randomly picking a node n in the peer's path toward the root. We show in Algorithm 1 the algorithm to reduce the total cost. A peer i moves up the tree with a new parent j if this reduces the cost and the parent can accept child. That means it moves up the tree if $f(D_i) > f(D_j + d(i, j))$. As f is monotonic increasing, this is equivalent to $D_i > D_j + d(i, j)$. If the potential parent can no longer accept any child, then a lower-priority child would be swapped with i . To do this, we define a cost reduction function $r(u, v)$ for a node u if it moves to a new parent v . Clearly, $r(u, v) = \text{old cost} - \text{new cost} = f(\vec{A}_u, D_u) - f(\vec{A}_v, D_v + d(u, v))$. If the swap of i and j leads to some cost reduction (i.e. $r(i, parent(j)) + r(j, parent(i)) > \theta$, for some threshold θ used for stabilization), then the swap would be done. We see from above that our improvement algorithm can continuously reduce total cost without oscillation.

3. ILLUSTRATIVE SIMULATION RESULTS

In this section, we study the performance of our protocol by means of simulations.

3.1. Simulation Environment and Metrics

We have implemented the above scheme using a simulator called p2pns [1]. The network topology has 10,000 nodes generated by the BRITE Internet topology generator. The topology is based on the two level top-down hierarchical model where both the AS and the router level use the power-law following Barabasi-Albert generation model.

We uniformly pick one of the following five upstream bandwidths for each peer in accordance of the upstream bandwidth distribution of Gnutella peer observed by Saroiu et al. [6]: 50 kbps, 400 kbps, 800 kbps, 2000 kbps and 7000 kbps.

We consider a stream with bitrate $B = 200$ kbps.⁴ The maximum number of children a peer can support is thus given by the ratio of its upstream bandwidth capability and B .

We use the following performance metrics to evaluate the system performance:

- Physical Link Stress, defined as the number of identical data packets carried in a physical link.
- Stretch, also known as relative delay penalty (RDP), defined as the ratio of source-to-receiver overlay latency to source-to-receiver unicast latency. This measures how well an overlay tree matches the underlying multi-cast topology based on shortest path routing.
- Total cost, defined as the sum of the cost values of all users.

In our simulations, we use the second order polynomial function as the cost function, i.e., $f(\vec{a}_v, D_v) = a_0 + a_1 D_v + a_2 D_v^2$, where $\vec{a}_v = \langle a_0, a_1, a_2 \rangle$ is the cost vector of user v and D_v is the source-to-peer delay of user v (in milliseconds). Using this function, the \oplus operator and *shift* function can be obtained accordingly. For example, $shift(\langle 0, 0, 1 \rangle, 3) = \langle 9, 6, 1 \rangle$ because $f(\langle 0, 0, 1 \rangle, D_x + 3) = (D_x + 3)^2 = 9 + 6D_x + D_x^2 = f(\langle 9, 6, 1 \rangle, D_x)$.

3.2. Illustrative Results

We compare our protocol with HMTP [2] and OMNI [3]. Figure 1 shows the steady state network performance of the three protocols. All users have the same cost vector $\langle 0, 1, 0 \rangle$. Figure 1(a) shows that the stress values of our protocol is lower than OMNI and higher than HMTP. In Figure 1(b), we show the average stretch versus group size. The average stretch of our protocol is the lowest, especially for medium and large group sizes. Both the stress and stretch of our protocol increase slowly with the group size. It shows that our protocol is scalable to a large number of users. Figure 1(c) shows the total cost of the three protocols versus group size. As expected, our protocol achieves the lowest total cost since our protocol targets on minimizing the total cost. The cost reduction is substantial (by about 50% for a group size of 512).

We next discuss the convergence time of our protocol. We form an initial tree at time tick one by randomly joining 256 peers into network. Then, in each time tick, we visit all peers in random order. For each peer, we employ the improvement algorithm to reduce the total cost. Figures 2 show our protocol's convergence property. Clearly, the total cost of the system quickly decreases and converges to a certain value (in around 5 time ticks). The convergence value is only about 24% higher than the total cost of a star topology where each peer is directly connected to the source (i.e. the optimum, ignoring the out-degree bound).

⁴Our results do not depend on any specific values of bandwidth and streaming rate; they depend only on out-degree bound of the nodes.

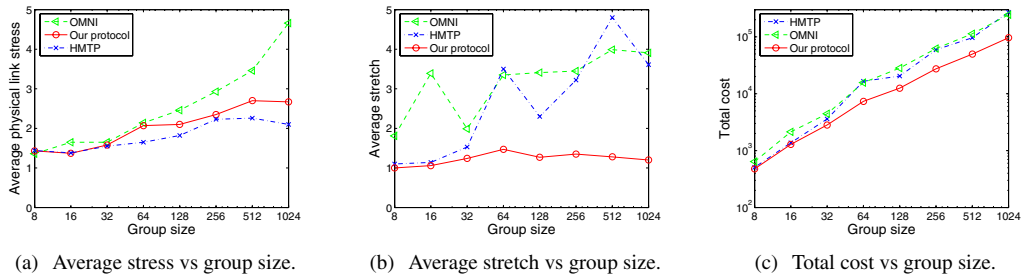


Fig. 1. Network performance.

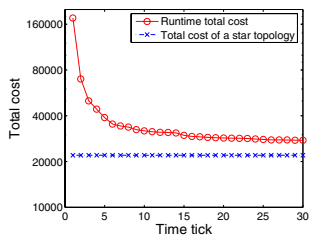


Fig. 2. Evolution of the total cost.

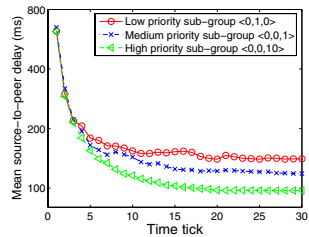


Fig. 3. Costs of three priority groups.

Figure 3 shows the differentiation of service among heterogeneous users with different cost vectors. In this simulation, all users are randomly assigned a cost vector from three different sub-groups – high, medium and low priorities so that the size of each sub-groups is approximately the same. The cost vectors of these sub-groups are $\langle 0, 0, 10 \rangle$ (highest priority), $\langle 0, 0, 1 \rangle$ and $\langle 0, 1, 0 \rangle$ (lowest priority), respectively. The average unicast delay between two peers is 127.77ms. From the figure, the source-to-peer delays of the three different sub-groups are clearly differentiated (after time tick 10). The low-priority sub-group has the highest delay while the high-priority sub-group has the lowest delay. It shows that our protocol can provide differentiated quality of services according to the user privilege.

4. CONCLUSION

In this paper, we propose a protocol to support service differentiation in overlay tree. In the protocol, the privilege of a

user is indicated by a user-specific cost function. The function is parameterized by a cost vector together with dependent variables in term of quality metrics (e.g. source-to-peer delay). The objective of the system is to minimize the overall cost of all participating users.

Our protocol is fully distributed and scalable. Simulation results show that the network performance (i.e. stress and stretch) of our protocol is comparable to other well-known overlay multicast protocols, while achieving much lower cost. Our protocol quickly converges to some value close to optimum and is able to provide differentiated services based on the user privileges.

5. REFERENCES

- [1] Vladimir Blagojevic, “A Simulator fo Peer-to-Peer Overlay Algorithms,” *MSc thesis, York University, Toronto*, June 2004.
- [2] Beichuan Zhang, Sugih Jamin, and Lixia Zhang, “Host Multicast: A Framework for Delivering Multicast to End Users,” in *Proceedings of Infocom*, June 2002.
- [3] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, “Construction of an efficient overlay multicast infrastructure for real-time applications,” in *Proceedings of Infocom*, 2003.
- [4] Lakshminarayanan Subramanian, Ion Stoica, Hari Balakrishnan, and Randy Katz, “OverQoS: An Overlay Based Architecture for Enhancing Internet QoS,” in *USENIX NSDI 04*, Mar. 2004.
- [5] Ahsan Habib and John Chuang, “Service Differentiated Peer Selection: An Incentive Mechanism for Peer-to-Peer Media Streaming,” *IEEE Trans. on Multimedia*, 2006.
- [6] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, “A measurement study of peer-to-peer file sharing systems,” in *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)*, Jan. 2002.