Dynamically Optimize Multicast Tree Structure

Ming Zeng, Hongliang Yu, Feng Yang, Weimin Zheng
Department of Computer Science, Tsinghua University, Beijing China

Abstract

Live-streaming applications based on overlay multicast have become very popular recently. There have been many studies on optimizing the overlay construction and improving multicast performance. Most of these studies focused on building period of overlay network or multicast tree. It is not flexible in the heterogeneous and variable Internet environment. In this paper, we present DOMT, an overlay multicast protocol in which the multicast tree can be optimized dynamically. In DOMT, high capability peers get better position in the tree and more peers get near peers as their parents. Most of the previous protocols didn't consider dynamic optimization because its bad impact to stability. In DOMT, we design some mechanisms to avoid the bad impact. The experimental results show that the optimization can decrease the height of multicast tree and shorten the average source-to-end delay. It also proves that the bad impact made by the optimization is acceptable.

Keywords: *Live Streaming; Overlay Multicast; Dynamic Optimization; Gossip*

1. Introduction

With the improvement of Internet, multimedia services have become very popular. Live stream is an important kind of multimedia services. For Live stream applications, IP multicast [1] is the perfect solution. However, it has not been widely deployed for some practical reasons. Application Layer Multicast, which is also called Overlay Multicast [2], is a good substitution of IP multicast. How to build an efficient and scalable overlay multicast construction has becomes a hot research topic.

There are many problems to be solved in a practical overlay multicast system, such as large source-to-end delay, waste of network bandwidth, churn of the overlay. These problems decrease the usability of the live steam applications based on overlay multicast. In order to solve these problems, there have been many studies on how to optimize the overlay multicast. The impact of Heterogeneous Bandwidth [3] is researched for improving the overlay multicast schemes. Locality aware strategies [4] are also used in many optimization schemes, such as AnySee [5]. In the overlay multicast scheme, the tree structure is often build for delivering data. Live stream applications have high play continuity demands, but the peers in the multicast tree can easily crash or leave the system and impact their descendent peers. It's the most important challenge for Tree-based multicast schemes.

There have been some studies on the impact of heterogeneous bandwidth constraints on Multicast Protocols. Many practical multicast protocol and system have considered it when the overlay multicast is established. Actually, outgoing bandwidth capabilities of peers in the overlay can't be known precisely when the peers first join the overlay. Therefore, a peer can't have an appropriate position when it just joins the overlay.

In this paper, we design DOMT (Dynamically Optimize Multicast Tree), a tree-based overlay multicast protocol with high reliability in the dynamic environment. In DOMT protocol, we propose a dynamic optimization scheme to reduce source-to-end delay and the waste of physical network bandwidth.

We design the dynamic optimization algorithm mainly for two objectives. First, reduce the average hops in the overlay network. Second, let more peers have better parent in the tree that is near the child peer in the physical network. It can reduce the average source-to-end delay and the source-to-end delay. It can also save the physical network bandwidth resource because more data packets are transferred between near peers in physical network. There is another good result of optimization. The system will be less influenced when a part of physical network crash or can’t connect with other parts because more peers rely on near peers in the physical network. Dynamic optimization has bad impact to the stability of the multicast service. For solving this problem, we design some mechanisms to alter multicast tree without impact many peers. The experimental results show that the impact made by dynamic optimization is much smaller than the peers leaving and joining.

The rest of this paper is organized as follows. Section 2 discusses the works related to ours. Section 3 describes the design of DOMT protocol. Section 4 presents the experimental results. At last, we conclude our work and outline our future work.

2. Related Works

There have been significant studies on overlay multicast these years. Many overlay multicast protocols and systems employ the tree structure because it’s simple and efficient. There are two kinds of tree structure, single tree and multiple tree. There are many single tree based protocols such as SpreadIt [6], NICE [7] and ZigZag [8]. And Splitstream [9] is a multiple tree protocol. Constructing and maintaining an efficient multicast tree among the overlay peers is the key issue for tree-based protocols. The internal peers in multicast tree often leave or crash in the highly dynamic environment. The peers leaving have impact to their descendant peers. Therefore, many algorithms are devised to repair multicast tree [10]. But the tree-based overlay multicast systems is still suffering from this. Because of this problem, some protocols eliminate trees from overlay multicast, such as Chainsaw [11]. However, recent studies, such as AnySee, prove that tree structure is effective and efficient in practice.

Gossip algorithms are good for message dissemination in p2p systems [12]. The random choice of gossip targets achieves resilience to random failures. Therefore, Gossip algorithms fit dynamic environment such as Internet. However, gossip algorithms can’t be used as media data delivery protocol in live streaming applications, because of the significant redundancy. The DONet [13] protocol has employed a gossip protocol for membership management. In DOMT protocol, we also employ a gossip protocol for membership management. But we do not eliminate tree-structure. Furthermore, we use the gossip protocol to maintain and optimize the multicast tree.

3. Design and Optimization of DOMT

The running of DOMT can be simply described as follows. First, a reliable gossip-based overlay is constructed. Second, multicast tree is established based on the overlay. Third, when the multicast system is running normally, the optimization alters the multicast tree. At the same time, the Buffer manager guarantees the play continuity during optimization process. The locality awareness policy is used both when the multicast tree is build and optimized.

3.1. Gossip-based Overlay Network

In DOMT, when a peer want to join the overlay multicast system, it joins a Gossip-based overlay network first. The overlay is maintained by a gossip-based membership protocol SCAMP [14]. In SCAMP, every peer establishes partner relation with some other peers randomly. Each peer periodically send heartbeat message to its partners. In DOMT, the partners could be the parent of the peer or help it upgrade or degrade in the multicast tree.

Gossip-based protocols have attractive scalability and reliability properties. We use the Gossip-based protocol to maintain an overlay network, and build multicast tree on it. This scheme is mainly for enhance the stability of the overlay multicast. When a peer crash or leave, its child peers can get new parent from partner list quickly, without long time pause.

3.2. Building Multicast Tree

After the peer joins the Gossip-based overlay network, it joins the multicast tree as soon as possible. When a peer first joins the multicast tree, fast startup deserves prior attention. When the peer gets the first partner, it asks the partner to join the multicast tree as its child. If failed, it will continue trying another partner until success. When several partners can be select as parent, the high level ones in the tree have priority. When a peer leaves the multicast tree and its child peers lose parent temporarily, the child peers will soon choose a new parent from its partners. The high level ones have priority. If no partner can be a parent, it will ask the partners recommend a parent from their partner lists.

In DOMT, a joining peer may not be laid in an appropriate position in the multicast tree, because the bandwidth capability of the peer is not clear. And fast startup is the most important thing for a new comer, not optimization of the multicast tree. During the multicast system is running, the bandwidth capabilities of the peers are clear. And then DOMT can optimize the multicast tree based on the peers' Capability.

3.3. Multicast Tree Optimization

Before describing the optimization algorithm, we give several definitions clearly. First, We define a peer's number of child peer as its contribution to the system. Second, we define a peer's level as follows. The root of the multicast tree is level 0, the highest level. Its child peers are level 1. The child peers of a level n peer are level n+1. Third, We define a peer is full when its bandwidth is not enough to afford a new child. Then we define two optimization operations, Degrade and Upgrade. The optimization algorithm is mainly construct of the two operations.

3.3.1. Degrade Operation. If a peer doesn't have contribution to the system and its level is high (for example: level < 6), it may be degrade. From Figure 1 we can see that the peer H, which have no contribution and at a high level, is degraded. The degraded peer finds a lower level peer as its new parent and leave the original parent.

3.3.2 Upgrade Operation. If a peer has much contribution to the system, it may be upgrade. From Figure 2 we can see that the peer C is upgraded. The upgraded peer finds a high level peer as its parent and leave the original parent.

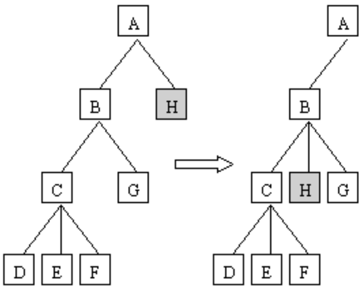

Figure 1. Degrade operation in which H is degraded

3.3.3. Optimization Mechanism. A peer that is not full can be the new parent of the optimized peer. When a peer is degraded or upgraded, it finds a new parent through its partners. The optimizing peer first search suitable parent from its partners, If none of the partners is available as a better parent. It asks its partners to provide their partners' information, and search suitable parent from them.

For avoiding 'orphan child', the optimization mechanism is as follows. The upgraded or degraded peer applies the new parent to take it as a child, but do not leave the old parent immediately. If the new parent accepts it as a child successfully, then it leaves the old one.

3.3.4. Optimization Algorithm. The optimization is executed periodically during the multicast system running. It is executed from the top of multicast tree to bottom like this.

(1) The root messages all of its child peers to start optimization.

(2) When a peer receives optimization message, if the peer is full and it has a child with no contribution, the peer message this child to do the degrade operation.

(3) At the same time, the peer selects a child with the most contribution and messages this child to do the upgrade operation. If there is no child whose contribution more than 0, no child can be upgraded.

(4) After Sending degrade and upgrade messages, the optimization process holds for T seconds (for example: 10 seconds). And then, the peer message all its child peers to start optimization.

The parameter T and the optimization frequency can be setup depending on system scale and network condition. Therefore, the dynamic optimization is restrained. There are not many peer altered in the tree during the optimization. Through the optimization, the high contribution peers get better parent in the tree. Therefore, the source-to-end delay of the upgraded peers and their child peers are shortened.

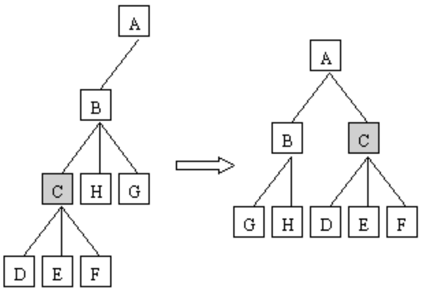

Figure 2. Upgrade operation in which C is upgrade

3.4. Buffer Management

The upgraded peers get new parents that have less source-to-end delay than the old parents. For ensuring play continuity, every peer has a buffer that stores the media data. The video stream is divided into segments of one-second length. When a peer gets media data segments, it keeps the segments in the buffer. After a segment is played, the buffer still reserves the segment for 30 seconds. When a 'slow' child is upgraded to a 'fast' parent, it can also get the segment from the buffer. Therefore, the upgraded peers can play the video smoothly during the upgrade operation.

3.5. Locality Awareness

The locality of the peers in the physical network has impact to the efficiency of the multicast. We use a very simple Locality aware strategy in DOMT protocol. We can easily know which Autonomous System a peer comes from by analyzing its IP address. We define that the peers in the same Autonomous System are near peers. When a peer join the multicast tree or do upgrade and degrade operation, it will first choose the near peers. This policy let more data transfer between near peers, reduce the usage of the physical network bandwidth.

The locality aware strategy can also help the system survive from network crash. Figure 3 plots an example. The peer A, C, D, E belong to Autonomous System 1. The peer B, F, G, H belong to Autonomous System 2. If AS2 crash or its connections to AS1 are break, the connections between A, C, D, E and B, F, G, H are break. In the left multicast tree, which has not been optimized, peer C, D, E are temporarily impacted by AS2's problem because B is C's parent peer. In the right multicast tree, which has been optimized, C has been upgraded to a better parent, peer A, which belongs to the same Autonomous System with C. Therefore, AS2's problem can not impact C, D, E again. The experimental result in Section 4.5 can show the benefit exactly.

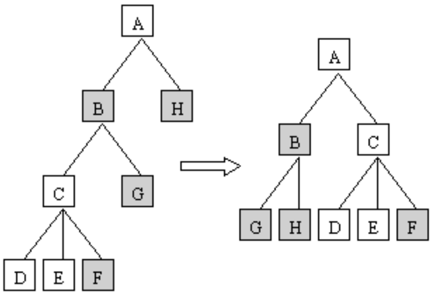

Figure 3. Optimization of the multicast tree in which the peers belong to two different Autonomous Systems

4. Experimental Results

4.1. Simulation Methodology

We generate a network topology by GT-ITM [15] and simulate DOMT protocol running on it. We assume the peers' bandwidth distribution as [16] measured. And we assume that the live stream service is CBR (constant Bit Rate) and the bit rate is 350Kbps. There are about 5000 peers running in the system simultaneously at most. SCAMP protocol is used to maintain the overlay network, and most of the peers have more than 10 partners. The parameter T equals to 10 seconds.

4.2. Stable Simulation

Firstly, we evaluate the optimization algorithm in an ideal environment that there are 5000 peers running in the system, and no peer leave or join the system during the simulation period. Figure 4 plots the decrease of average source-to-end hop on the overlay network caused by the optimization. Figure 5 plots the decrease of average source-to-end delay. The source-to-end delay is not always equal to the delay of playing. For example, an upgraded peer's source-to-end delay is shortened, but its delay of playing is not changed. From Figure 4 and Figure 5, we can see the effectiveness of the optimization scheme clearly. The decrease of source-to-end hop and delay means that the system spends less network bandwidth and the peers get data from source faster.

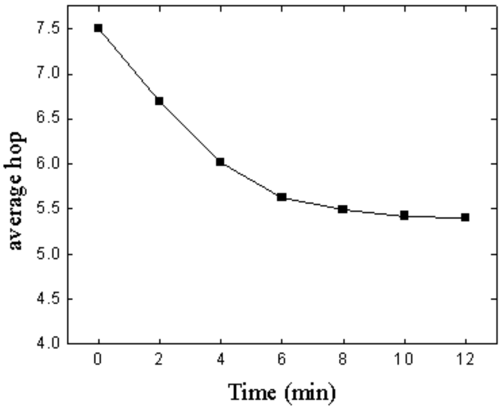

Figure 4. Average source-to-end hop V.S. Run time

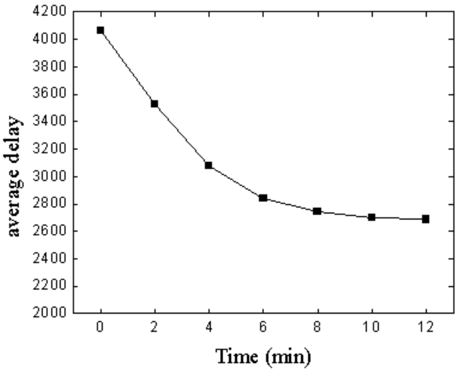

Figure 5. Average source-to-end delay V.S. Run time

4.3. Dynamic Simulation

Different from the previous simulation, all of the rest simulation is based on a dynamic environment. We design the simulation process as follows. The overlay network and multicast tree are build as we described before, but the optimize do not start until the system is stable. We assume that the number of peers remains at about 5000 when we start to optimize the system. The average lifetime of the peers is 40 minutes, with exponential distribution.

Figure 6 and Figure 7 plots the optimization result. The results show the difference between the optimized system and not optimized system clearly. In contrast to the result of static simulation, the optimization algorithm got similar effectiveness, however, the result is a little worse than before. The experimental results prove that the optimization algorithm is effective in a dynamic environment.

The simulation results of not optimized system show that the average hop and delay are fluctuating during the simulation period. The main reason is the different property of the leaving peers. In the first 2 minutes, 12 high level peers (level < 4) leave the system. In the second 2 minutes, the number is 22. We can see that the optimized system is also influenced by this factor. The optimization efficiency in the first 2 minutes is much better than in the second 2 minutes.

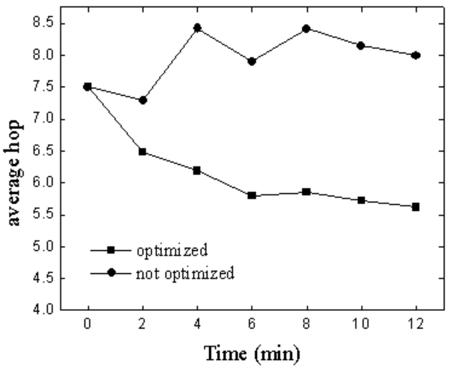

Figure 6. Average hop difference between the optimized system and not optimized system

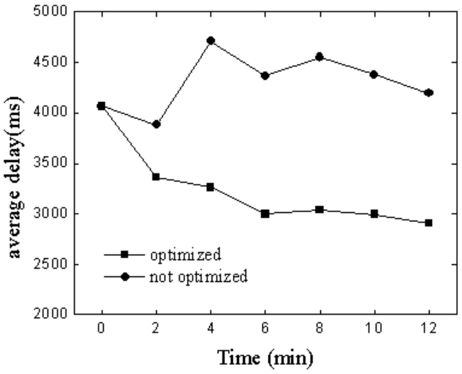

Figure 7. Average delay difference between the optimized system and not optimized system

4.4. Bad Impact of Optimization

Dynamic optimization is a kind of churn of the multicast tree, and has bad impact to some of the peers. The experimental result showed in Figure 8 is to prove the cost of dynamic optimization is acceptable. We define that if a peer leaves the multicast system, all of its descendent peers that get media data from it, no matter directly or indirectly, are temporarily impacted. When a peer is degraded during the optimization, its source-to-end delay increases. Therefore, the playing may be paused, waiting for media data. We define that the degraded nodes are temporarily impacted. When a peer is upgraded during the optimization, its source-to-end delay decreases. Here we have a mechanism to ensure the play continuity. As we described before, every peer has a buffer to keep played media data for 30 seconds. In the experiment, we find that the max decrease of source-to-end delay in the optimization period is less than 15 seconds. Therefore, the upgraded peer can ask the new parent to send the needed data that is stored in the buffer. So the upgraded peers are not temporarily impacted.

In Figure 8，One line shows the number of peers temporarily impacted by the optimization. The other line shows the number of peers temporarily impacted by peers leaving and joining. From this experiment, we can see that the bad impact of optimization is much smaller than peer leaving and joining. So, the cost of the optimization is acceptable.

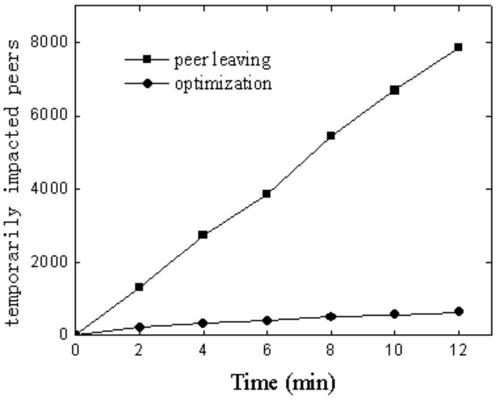

Figure 8. The total number of temporarily impacted peers cause by peers leaving and optimization

4.5. Impact of Autonomous System Crash

Sometimes, some Autonomous System may crash or have some problem. In the experiment，We simulate this phenomenon and plot the impact. We made one of the Autonomous Systems crash once at different crash time. The Autonomous System has about 750 peers running in the system. They leave the system at the crash time. Although the multicast tree can quickly recover from the crash, there are a lot of temporarily impacted peers. Figure 9 plots the number of temporarily impacted peers at different crash time. We can see that the number of temporarily impacted peers is decreased by optimization. This result proved our viewpoint described in Section 3.5.

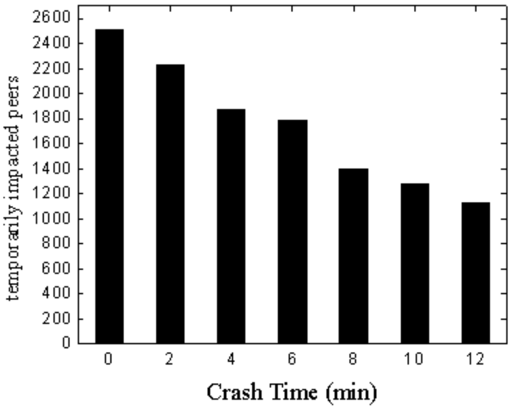

Figure 9. The number of temporarily impacted peers when an Autonomous System crashes

4.6. Optimization Frequency

Optimization frequency has impact to the system. If the optimization is executed too frequently, there will be too many peers impacted by the optimization and the multicast service will be unstable. If the optimization frequency is low, the efficiency will be worse. It’s because the peers leaving churn the multicast tree and interrupt the optimization strategy. Figure 10 shows the optimization results at different frequency. If the optimization is executed every 4 or 8 minutes, the results are worse.

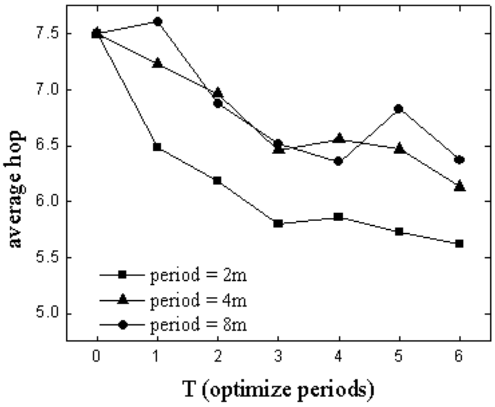

Figure 10. Optimization Efficiency V.S. Frequency

5. Conclusions and Future Work

In this paper, we propose an overlay multicast protocol, DOMT. We used a gossip-based protocol to maintain the overlay network, improving the reliability and scalability. And we design a dynamic optimization algorithm to improve the performance of DOMT. The Heterogeneous Bandwidth of the peers and Locality aware strategy are considered in DOMT, and help to optimize the multicast tree. The experimental results show dynamic optimization have much benefit to the multicast system, and will not do harm to the stability of system if we have good control mechanism.

This work proves that dynamic optimization is helpful to optimize overlay multicast. However, there are a lot of progresses we can do to this simple algorithm. Although we have some mechanisms to avoid bad impact to the system made by dynamic alter the multicast tree, there might be some unknown problem during dynamic optimization in practice. We will have more experiments to discover the problem, and solve them.

Acknowledgment

This work is supported by China National Natural Science Foundation (NSFC) No. 60433040.

References

[1] R. Perlman, "Models for IP Multicast", in Proceedings of *IEEE International Conference on Networks*, 2004.

[2] 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 *IEEE INFOCOM*, 2003.

[3] A. Bharambe, S. Rao, V. Padmanabhan, S. Seshan, and H. Zhang, "The impact of heterogeneous bandwidth constraints on dht-based multicast protocols", in *International Workshop on P2P Systems*, 2005.

[4] Y. Liu, X. Liu, L. Xiao, L. Ni, and X. Zhang, "Location-Aware Topology Matching in P2P Systems", in Proceedings of *IEEE INFOCOM*, 2004

[5] Xiaofei Liao, Hai Jin, Yunhao Liu, Lionel M. Ni, and Dafu Deng "AnySee: Peer-to-Peer Live Streaming", in Proceedings of *IEEE INFOCOM*, 2006

[6] H. Deshpande, M. Bawa, and H. Garcia-Molina, "Streaming live media over peer-to-peer network", *Technical Report*, Stanford University, 2001.

[7] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast", in Proceedings of *ACM SIGCOMM*, 2002.

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

[9] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-bandwidth Content Distribution in Cooperative Environments", in Proceedings of *ACM SOSP*, 2003.

[10] D. A. Tran, K. A. Hua, and T. T. Do, "A peer-to-peer architecture for media streaming", in *IEEE J. Select. Areas in Comm.*, vol. 22, Jan.2004.

[11] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, A. E. Mohr, "Chainsaw: Eliminating Trees from Overlay Multicast", In *International Workshop on P2P Systems*, 2005.

[12] P. Eugster, R. Guerraoui, A. M. Kermarrec, and L. Massoulie, "From epidemics to distributed computing", in *IEEE Computer Magazine*, 2004.

[13] Xinyan Zhang, Jiangchuan Liu, Bo Li, and Tak-Shing Peter Yum, "Data-Driven Overlay Streaming: Design, Implementation, and Experience", in Proceedings of *IEEE INFOCOM*, 2005.

[14] A. J. Ganesh, A. M. Kermarrec, and L. Massoulie, "Peer-to-peer membership management for gossip-based protocols", *IEEE Transactions on Computers*, 52(2), Feb. 2003.

[15] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to Model an Internetwork", in Proceedings of *IEEE INFOCOM*, 1996.

[16] S. Saroiu, P. K. gummadi, and S. D.Gribble , "A Measurement Study of Peer-to-Peer File Sharing Systems", In Proceedings of *MMCN*, 2002.