Journal of Universal Computer Science, vol. 15, no. 9 (2009), 2011-2025
submitted: 15/8/08, accepted: 24/4/09, appeared: 1/5/09 © J.UCS

Transmission Latency based Network Friendly Tree for Peer-to-Peer Streaming

Ting Peng
(MOE KLINNS Lab and SKLMS Lab, Department of Computer Science
Xian Jiaotong University, Xian, 710049, P.R.China
MOE Key Laboratory of Specific Region Highway Engineering
Chang'an University, Xian, 710064, P.R.China
t.peng@ieee.org)

Qinghua Zheng
(MOE KLINNS Lab and SKLMS Lab, Department of Computer Science
Xian Jiaotong University, Xian, 710049 China
qhzheng@mail.xjtu.edu.cn)

Yinli Jin
(Department of Automatization
Chang'an University, Xian, 710064, P.R.China
jinyinli@263.net)

Abstract: In Peer-to-Peer(P2P) streaming system, the multicasting tree construction method influences considerably on network load. Under current available strategies, network resources are not used economically, or network resource friendship and the flexibility can't be approached thoroughly. With the increasing application of Internet, network infrastructure itself becomes a precious resource, which should be performed effectively. In our implementation, network transmission delay between the peers is detected, used as practical performance metrics of the P2P streaming system. According to the metrics, network friendly tree is provided as overlay multicasting tree construction strategy for P2P streaming system. In this strategy, additional transmission delays are minimized as new peers enter. The simulation experiment presents that the proposed strategy network friendly tree works better than other contrasts. And the transmission performance improves significantly at minor additional cost.

Key Words: Transmission Latency, Network Friendly Tree, Peer-to-Peer, Streaming, Performance Metrics
Category: C.2.3,C.4,H.4.3,H.5.1

1 Introduction

Nowadays, multimedia and streaming media applications are popularly used in many industries and professions [Vega-Rodriguez et al., 2005]. With such current available streaming media people enjoy multimedia without downloading [Ganjam and Zhang, 2005, Kolbitsch, 2005]. Accordingly, a lot of people watch happenings promptly via Internet. That realization can be approached by the streaming media, delivered through centralized media distribution, content distribution network and Peer-to-Peer(P2P) network [Na et al., 2008, Cheng et al., 2008].

For the average user, centralized media distribution like YouTube provides easy-to-access and easy-to-use services. In comparison, for a broadcasting of high quality streaming media over the Internet to a large-scale audience, the P2P streaming system is a better choice [Peltotalo et al., 2008]. In fact, P2P is one of the most promising solution to this problem. In P2P system, the peers contribute their own bandwidth and processing capability to the whole system. Researches show that P2P system is feasible to support large scale media streaming over Internet [Yiu et al., 2007]. This makes it easy for the streaming media source provider to deliver its content. Perhaps it is the only practical means to deliver streaming media to the masses in time for individuals, thanks to its flexibility, cost-effective and high scalability. When it is employed to deliver streaming media, distribution tree used in P2P system is essential to performance.

A commonly used method is to maintain a centralized directory of all peers in a directory server (such as *ppstream* or *pplive*) [Zhou and Liu, 2005]. The server keeps the global overlay topology among peers. In this system, a new peer's request is directed to the directory server. The directory server then selects the most suitable supplying peers for the new peer, according to its network address and the requested media. This system is easy to maintain. However, another system weakness is that the directory server becomes a single point of failure. If the directory server is not the source of media, its failure impairs the whole P2P system [Padmanabhan et al., 2002]. Hierarchical overlay structure [Tran et al., 2004] and DHT [Sharma et al., 2005] are also popular means for locating of supplying peers. These efforts stress on organizing the peers in P2P system. As network is dynamic, a set of dynamic maintenance scheme is provided in [Liao et al., 2007]. To decrease latency while maintaining the desirable properties of pull-based streaming, combining push and pull mechanisms may offer a good compromise [Narayanan et al., 2007]. But in practice, P2P streaming is an intensive network transmission application. So, efforts should be taken to improve network transmission performance.

Accordingly [YU et al., 2007], P2P overlay performance is not the same as network layer performance. Practically, if network layer performance is not considered, ideal overlay performance can not be approached. Among the currently available P2P streaming technologies, Application Level Multicast Infrastructure(ALMI) is optimized for resource usage [Pendarakis et al., 2001]. In ALMI, all peers are organized as a minimum spanning tree, and all the peers in the P2P streaming system should be known in advance in order to do this. If new peers entering after the tree is constructed, reorganization of the tree is also a resource consuming process. So it is not an ideal strategy in applications where peers may come and leave at anytime.

In [Padmanabhan et al., 2003], randomized tree and deterministic tree for P2P streaming are discussed. In randomized trees, which may be relatively deep, transmission delays are cumulated. While in the deterministic ones, depth of the trees decreases. In [YU and WANG, 2007], each node in the system gathers the nodes information at the same subnet, and takes these nodes as its neighbors. It greatly improves the routing performance of the structured P2P network. But, it is only effective when nodes are in the same subnet, which is not versatile.

So, it is necessary to create a versatile and flexible overlay multicast tree with consideration of network resource occupation for P2P streaming system. This paper proposes a Network Friendly Tree(NFT) to address these problems discussed above. In a NFT mounted P2P streaming system, when a new peer wants to join the specified multicasting P2P system, it gets the list of peers in the system via a search protocol provided by the P2P system (e.g. Gossip). The new peer sends measuring packets to the peers, then, round trip delays between the new peer and the peers who are already in the P2P streaming multicasting system are figured out. Then the new peer tries to obtain the media stream from the peers with least transmission delay. In this way, peers can enter the multicasting system at any time, and the efficiency of network resource usage is also reached. Also, contrast experiments have been carried out to ensure practical performance under the proposed strategy.

The rest of the paper is structured as follows. The proposed NFT for P2P streaming is discussed in the next section, experimental results in Section 3, and conclusions in Section 4.

2 Proposed Strategy

To the best of our knowledge, current available strategies don't provide effective transmission tree of P2P steaming system with consideration of saving network resource. Currently, more and more people are using Internet to access information and communicate with each other by means of multimedia. Population suffered from network congestion is constantly increasing. So, we should do something to bring down network congestion.

NFT is a new method of multicasting tree construction for P2P streaming. Its objective is to use network resource more gracefully. It minimizes network resource occupation without impairing the QoS or flexibility. In order to improve reliability and make full use of network resources, topology of the underlying physical network is concerned in NFT. It is used to overcome overwhelming consumption of network resource in current P2P streaming tree generating strategies.

For network resource is occupied by the system only when streaming data are transmitted through the network, network resource can be saved when traf-

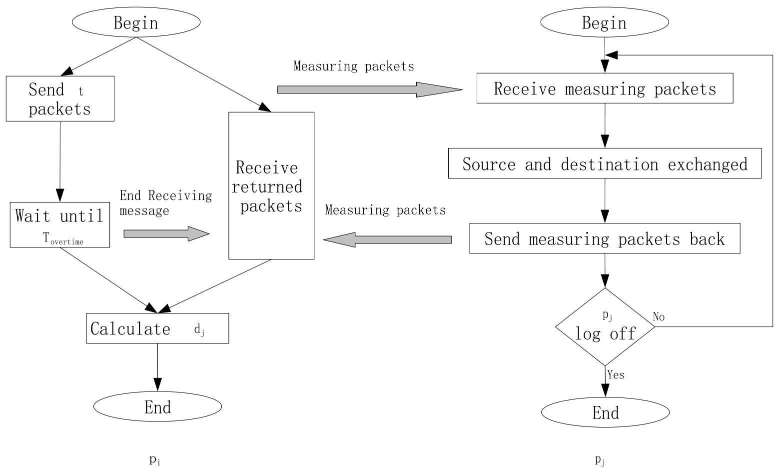

Figure 1: Transmission Delay Measuring Between p_i and p_j

fic time of data is decreased. Since transmission delays between the peers are measured before the tree is constructed, NFT focuses on minimizing network transmission delay between the peers. In doing this, the oversized occupation of network resource drops off.

2.1 Performance Metrics

In P2P streaming system, media data are transmitted across network infrastructure. Data transmission performance is affected by capability of network facilities, background data flow and other factors. These factors should be considered in P2P streaming system. Nevertheless, data sent by other applications are hard to manage. So, for a specified application, the practical way to minimize network resource occupation is to cut down its own resource occupation. We exert ourselves to accomplish this task.

In our implementation, we assume primarily that all peers have enough processing power and enough bandwidth to transmit the streaming data. So, our research converges how to minimize network resource occupation when data is transmitted along the network paths. In practice, the availability of network resource lies on transmission path, topological distribution of peers, data stream bandwidth of the application, etc. Such amount of uncertain factors shall bring in inaccurate measurement of network usage. One currently used approach is, with help of transmission delay, to evaluate network usage [Pendarakis et al., 2001].

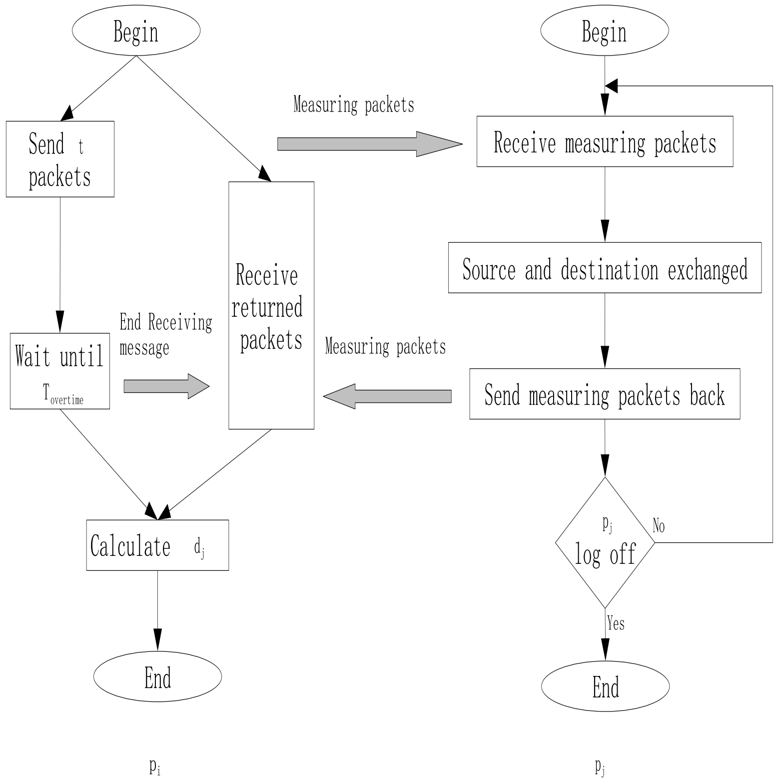

Figure 2: Construction Process of Network Friendly Tree

In P2P streaming system, all peers but the root one obtain the streaming from their parent. Network resource usage can be evaluated by the transmission delays from the parent peers to the child ones, for network infrastructure is serving the streaming data during transmission. So, transmission delays between the peers are used to set up performance metrics. Network resource in P2P streaming system is occupied only when data are transmitted between the peers.

For a multicasting P2P system comprising n peers, the set of all peers is denoted as S_{total}, and the peers are denoted as $p_1, p_2 ... p_n$. In order to make the performance metrics fit to evaluate different multicast tree to be used in P2P streaming system, we use the average transmission delay d_a from the parent peer as the performance metrics. d_a is calculated according to (1). Here, d_i stands for transmission delay from parent peer of p_i to p_i itself. Lower value of d_a means less network resource occupation to serve the same set of peers.

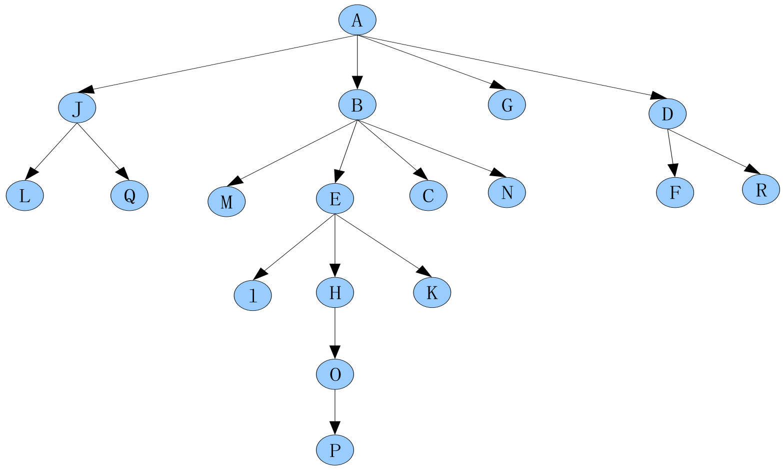

Figure 3: Sample of Randomized Tree

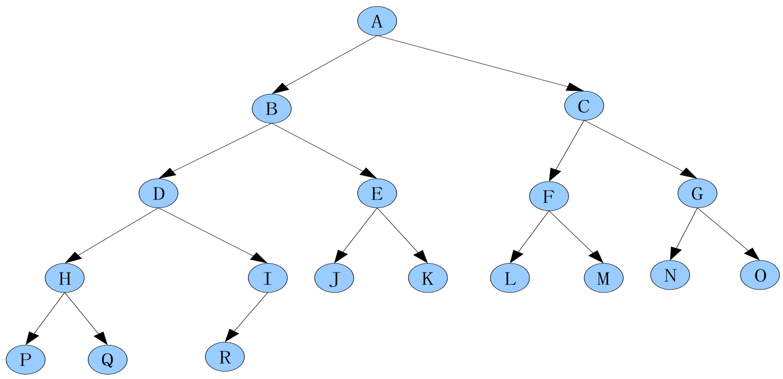

Figure 4: Sample of Deterministic Tree

$$d_a = \frac{\sum_{i=1}^{n} d_i}{n} \tag{1}$$

In practice, different P2P streaming systems have different set of peers. Thus, the scale and population of the whole system vary significantly. They may vary from a campus size P2P system to a global system. Accordingly, transmission delays in different systems vary greatly.

For distribution of the peers has significant influence on network resource usage, our performance metrics d_a is also impacted by this factor. So, d_a is not suitable to evaluate multicast tree constructed from different set of peers. It is only suitable to evaluate multicast tree constructed according to different methods from the same set of peers. In other word, it is suitable to evaluate different streaming multicast trees constructed from the same set of peers, which means a limitation in specified applications.

2.2 Transmission Delay Between Peers

For transmission delays between the peers have significant influence on network infrastructure occupation, we should consider this factor carefully in order to make a network friendly implementation of P2P system. The essential feature of NFT is stressed on this factor. In order to construct NFT for P2P streaming, transmission delays between peers are detected. Transmission delay measuring process is shown in Fig.1. When p_i wants to measure transmission delay between p_i and p_j, it sends t UDP packets containing the time stamp to the destination peer p_j . At the same time, p_i starts to receive the returned packets from p_j. When p_j receives the packets, it exchanges their source and destination address of the packets, then, sends them back immediately. After t packets are sent, p_i waits until overtime threshold $t_{overtime}$ expires. Then, p_i terminates packets receiving process. Transmission delay is computed according to the time stamps of the returned packets. In our experiments, t is set to 10, $t_{overtime}$ is set to 4 seconds. Transmission delay between any two peers is calculated according to the round trip time of the returned packets.

2.3 Tree Construction

In our implementation, transmission delays between the peers are used as network resource occupation metrics. Our objective is to minimize transmission delays between the peers. In this way, network resource occupation of the whole P2P streaming system will be cut down. The problem becomes a Steiner tree problem, if transmission delays between the peers are considered as distances [Bozorgzadeh et al., 2001]. In theory, it is an NP-Complete problem, if additional peers may be introduced into the system to intransit data at specified position. In practice, only peers in the P2P system can cooperate in the system and transmit data to other peers. Thus, additional peers are not available in practical P2P systems. This constraint makes the problem to be a minimum spanning tree [Pendarakis et al., 2001].

In fact, in order to construct a minimum spanning tree, all the peers in the system must be known before the tree is constructed. Otherwise, reorganizing the tree may be a resource consuming process.

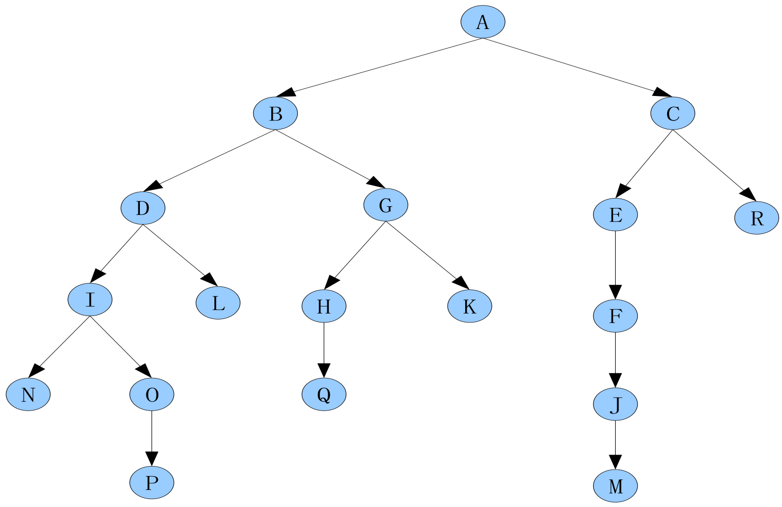

Figure 5: Sample of Network Friendly Tree

So, what we need is to minimize network transmission delay without impairing the flexibility. This is what NFT provides to us. As shown in Fig.2, in NFT when a new peer p_{new} wants to enter the P2P streaming system, it sends a request to the P2P system, and gets the list of the peers. Then, p_{new} selects c peers of S_{total} at random as a subset of the peers, the subset of the peers is denoted as S_{sel}.

Then, p_{new} calls the network transmission delay measuring function to obtain roundtrip delays between p_{new} and the peers in S_{sel}. The roundtrip delays between p_{new} and $p_i(p_i \in S_{sel})$ are denoted as $d_i(i \in 1...c)$. According to the results, the peers in S_{sel} are sorted by increasing order of roundtrip delays as $p_{sort,1}, p_{sort,2}...p_{sort,c}$. Then, p_{new} tries to contact the peers of S_{sel} in turn, obtains the streaming data from the peer with minimum transmission delay. When $p_{sort,i}$ receives the connecting request from p_{new}, if its bandwidth and processing power are both enough to serve another child peer, it sends *acknowledgement* message to p_{new}. Otherwise, it sends *reject* message to p_{new}. If p_{new}, receives the *acknowledgement* message sent by $p_{sort,i}$, it establishes connection with $p_{sort,i}$ and obtain streaming data from it. In this way, NFT is established. We can infer that NFT tries to obtain data with minimum additional transmission delay. Here, the complexity of the method lies on c.

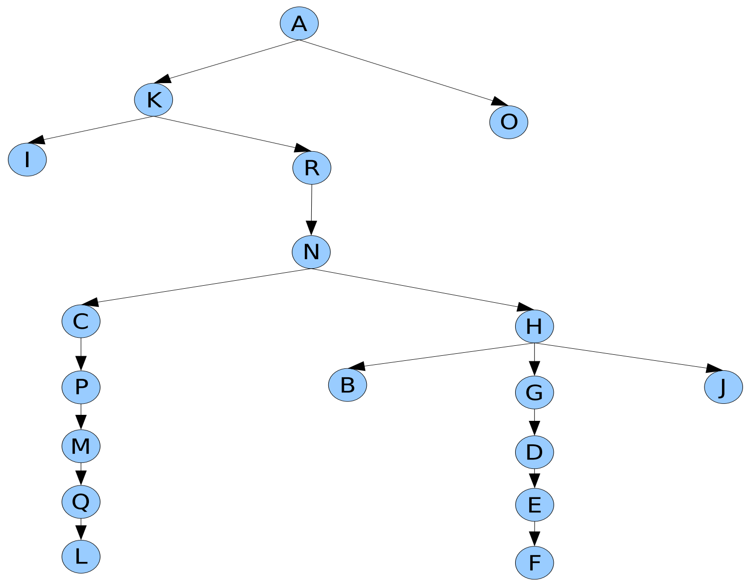

Figure 6: Sample of ALMI Tree

3 Experimental Results and Discussion

In order to find out practical performance of NFT, comparative experiments are carried out. In order to make our experiment suitable to evaluate large scaled performance of different strategies, *PlanetLab* is used as platform to obtain transmission delays between the peer. For *PlanetLab* is a world wide system, transmission delays obtained from it is ideal for our simulation experiment.

For *PlanetLab* has more than 900 nodes located all over the world, it is an ideal testbed for global applications. In order to estimate practical performance of the proposed NFT and other strategies, we selected 18 nodes of *PlanetLab* to obtain transmission delays. The list of the nodes is shown in Tab.1. There are nodes in North America, Asia, Europe. Locational diversity and globally distribution of the nodes are ensured. Then, transmission delay measuring program are deployed for these nodes and transmission delays between the nodes are obtained.

At first, transmission delays between these peers are measured. Based on measured transmission delays, the P2P multicast tree are constructed as randomized tree, deterministic tree, NFT and ALMI Tree [Pendarakis et al., 2001].

Table 1: List of *PlanetLab* Nodes Used in Transmission Delay Measuring

Node Number	Domain Name of the Nodes
A	planetlab11.millennium.berkeley.edu
B	planetlab10.millennium.berkeley.edu
C	pli1-pa-3.hpl.hp.com
D	planetlab2.lkn.ei.tum.de
E	planetlab3.csail.mit.edu
F	thu2.6planetlab.edu.cn
G	righthand.eecs.harvard.edu
H	planetlab2.csail.mit.edu
I	planet1.zib.de
J	planetlab1.csg.uzh.ch
K	planet2.scs.cs.nyu.edu
L	planetlab1.xeno.cl.cam.ac.uk
M	thu1.6planetlab.edu.cn
N	planet1.manchester.ac.uk
O	planet2.manchester.ac.uk
P	planet2.zib.de
Q	planet1.scs.cs.nyu.edu
R	planetlab2.iii.u-tokyo.ac.jp

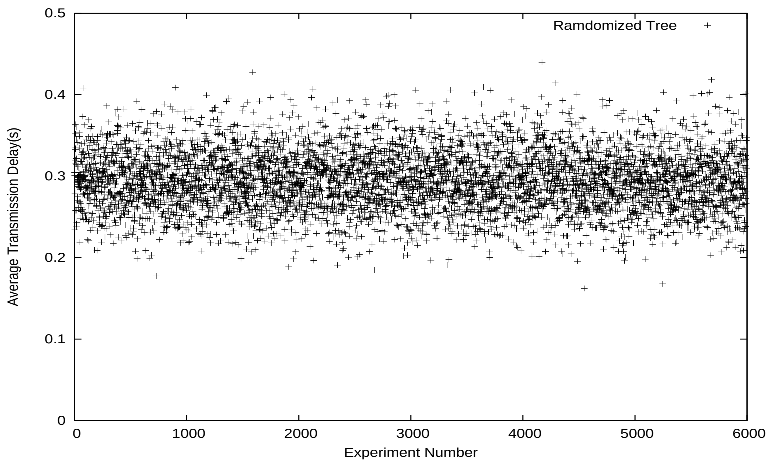

Figure 7: Experimental Results of Randomized Tree

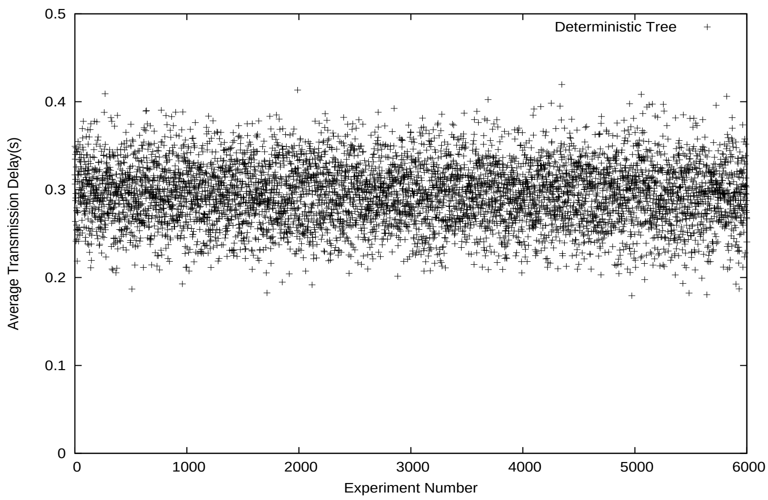

Figure 8: Experimental Results of Deterministic Tree

In order to find out the relationship between construction cost and its perfromance, different c values are adopted in NFT construction. Then, practical performance of the trees are evaluated according to (1).

3.1 Results

In order to acquire practical network resource usage of different multicasting tree for P2P streaming, we implemented simulation program according to randomized tree and deterministic described in [Padmanabhan et al., 2003]. We also implemented NFT according to Fig.2. ALMI tree is constructed according to [Pendarakis et al., 2001]. There are 18 nodes included in our simulation. The number of their sequences entering the P2P streaming system is 18!. It is a really large number, not practical to include all the entering sequences in the simulation. So, 6000 of the sequences generated at random are used in the simulation. During the simulation, transmission delays between the nodes are set according to previously described test carried at $PlanetLab$. For each random entering sequence of the nodes, randomized tree, deterministic tree, NFT and ALMI tree are constructed separately. Then, d_a of each constructed tree is calculated as performance metrics.

For nodes entering sequence $A, B, C, ...R$, constructed trees are shown in Fig.3, Fig.4, Fig.5 and Fig.6. In randomized trees, like a sample shown in Fig.3, new peers chose parent peer at random. In deterministic trees, like a sample shown in Fig.3, new peers chose parent with consideration to make a fertile tree.

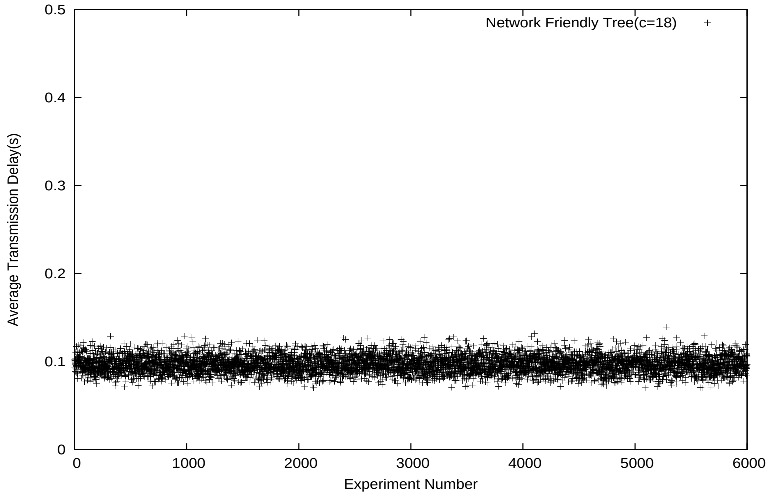

Figure 9: Experimental Results of Network Friendly Tree

Table 2: Statistics of Experimental Results

Case	Mean Value	Standard Deviation	Standard Error
Randomized Tree	0.295340	0.000472	6.09737×10^{-6}
Deterministic Tree	0.295978	0.000457	5.89583×10^{-6}
NFT(c=3)	0.171100	0.000343	4.43063×10^{-6}
NFT(c=6)	0.122800	0.000201	2.60006×10^{-6}
NFT(c=10)	0.103390	0.000144	1.85397×10^{-6}
NFT(c=14)	0.097750	0.000128	1.65877×10^{-6}
NFT(c=18)	0.096764	0.000127	1.63649×10^{-6}
ALMI Tree	0.059618	0	0

So, deterministic tree has least height. In proposed NFTs, like a sample shown in Fig.5, new peers chose parent with consideration of introducing minimum additional transmission delay. In ALMI tree, the peers are organized as a minimum spanning tree. Transmission delay of the whole tree is minimized. The d_a of the trees shown in Fig.3 ,Fig.4, Fig.5 and Fig.6 are $0.41s$, $0.32s$, $0.11s$ and $0.06s$ separately.

Our simulation experiment was carried out with 6000 peer entering sequences. For each peer entering sequence, randomized tree, deterministic tree and NFT are constructed. Then, d_a of the trees for each sequence were computed. The

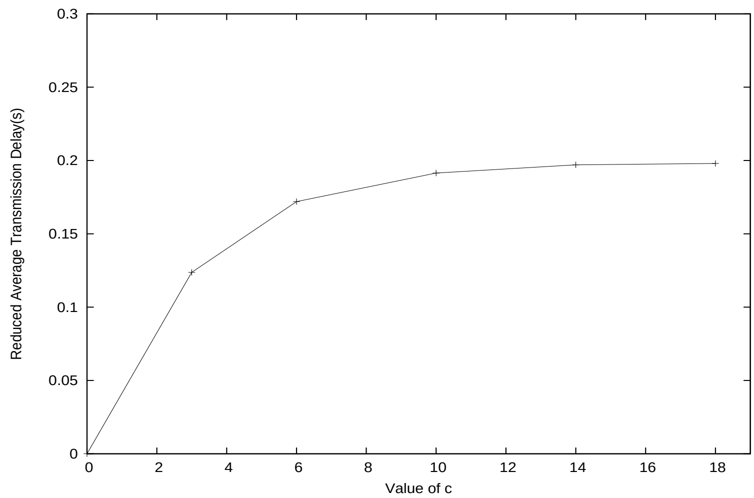

Figure 10: Relationship of c and Reduced Average Transmission Delay

result of randomized tree, deterministic tree and NFT are shown in Fig.7, Fig.8 and Fig.9 separately. Statistics of the results are listed in Tab.2.

The results indicate, randomized tree doesn't differ from deterministic tree significantly. We can infer that randomized tree and deterministic tree almost have the same performance according to our metrics. d_a of them are concentrated to 0.3 s. But the proposed NFT has an outstanding performance, its d_a approaches to 0.1 s.

The performance of a NFT mounted P2P system with 18 peers is shown in Fig.10. When c is set to 0, NFT regresses to randomized tree; when c is set to 3, performance of NFT increases significantly; when c is set to 6, its performance is even better; when c larger than 10, the construction cost of the tree increases with no responding return. Among all the trees, ALMI tree has minimum average transmission delay.

According to the results, ALMI tree has the most outstanding performance. Nevertheless, all the peers and transmission between them should be known before the tree is constructed. This request is not easy to satisfy, for peers are entering and leaving the P2P system at any time. In general, the proposed NFT is an ideal multicasting tree constructing strategy. If NFT is employed in P2P streaming system, practical performance of the system will be improved significantly without impairing the flexibility. In order to obtain ideal cost-effective, the value of c should be chosen properly.

4 Conclusions and Future Work

With the increasing application of Internet, network infrastructure itself becomes a precious resource. In this paper, performance metrics with consideration of network occupation is proposed. According to the metrics, NFT is provided as P2P streaming strategy.

In NFT, new peers select parent peers with consideration of introducing minimum additional transmission delay. Thus, transmission delay between the peers of the whole system is decreased. The simulation experiments suggest that NFT occupies less network resources than randomized tree and deterministic tree to serve the same set of peers. With minor cost of additional transmission delay measurement, dramatic promotion of its performance comes off. It is more friendly to network resources than others. Unlike ALMI [Pendarakis et al., 2001], in NFT, peers can join the P2P streaming system in any order. New peers do not disturb the joined ones. It makes trees effective without reduction of flexibility. In general, NFT is a network friendly, flexible multicast tree construction strategy for P2P streaming systems. It consumes less and serves more.

Practically, network condition varies from time to time [Borzemski, 2007]. Therefore, the P2P multicasting tree should accommodate this factor. But in this paper, these factors are not concerned. They shall be analyzed in the future.

Acknowledgment

This work is supported by NSFC (60825202, 60633020) and National High Tech. Development Plan (2006BAH02A24-2, 2006BAF02A00, 2006BAK11B02, 2007AA01Z475).

References

[Borzemski, 2007] Borzemski, L. (2007). Internet path behavior prediction via data mining: Conceptual framework and case study. *Journal of Universal Computer Science*, 13(2):287–316.

[Bozorgzadeh et al., 2001] Bozorgzadeh, E., Kastner, R., and Sarrafzadeh, M. (2001). Creating and exploiting flexibility in steiner trees. *Design Automation Conference, 2001. Proceedings*, pages 195–198.

[Cheng et al., 2008] Cheng, D., Jin, Y., and Qian, J. (2008). A layered multicast tree for reliable real-time video transmission of omn. In *Intelligent Computation Technology and Automation (ICICTA), 2008 International Conference on*, volume 2, pages 69–72.

[Ganjam and Zhang, 2005] Ganjam, A. and Zhang, H. (2005). Internet multicast video delivery. *Proceedings of the IEEE*, 93(1):159–170.

[Kolbitsch, 2005] Kolbitsch, J. (2005). Fine-grained transclusions of multimedia documents in html. *Journal of Universal Computer Science*, 11(6):926–943.

[Liao et al., 2007] Liao, R., Yu, S., and Dong, L. (2007). A novel peer management model for live peer-to-peer streaming. *Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007. International Symposium on*, pages 538–541.

[Na et al., 2008] Na, Y., Abdullaev, S., and Ko, F. (2008). An optimization of cdn using efficient load distribution and rads caching algorithm. *Journal of Universal Computer Science*, 14(14):2329–2342.

[Narayanan et al., 2007] Narayanan, S., Braun, D., Buford, J., Fish, R., Gelman, A., Kaplan, A., Khandelwal, R., Shim, E., and Yu, H. (2007). Peer-to-peer streaming for networked consumer electronics [peer-to-peer multimedia streaming]. *Communications Magazine, IEEE*, 45(6):124–131.

[Padmanabhan et al., 2003] Padmanabhan, V., Wang, H., and Chou, P. (2003). Resilient peer-to-peer streaming. In *Network Protocols, 2003. Proceedings. 11th IEEE International Conference on*, volume 1, pages 16–27.

[Padmanabhan et al., 2002] Padmanabhan, V., Wang, H., Chou, P., and Sripanidkulchai, K. (2002). Distributing streaming media content using cooperative networking. In *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video.*

[Peltotalo et al., 2008] Peltotalo, J., Harju, J., Jantunen, A., Saukko, M., Vaatamoinen, L., Curcio, I., Bouazizi, I., and Hannuksela, M. (2008). Peer-to-peer streaming technology survey. *Networking, 2008. ICN 2008. Seventh International Conference on*, pages 342–350.

[Pendarakis et al., 2001] Pendarakis, D., Shi, S., Verma, D., and Waldvogel, M. (2001). Almi: An application level multicast infrastructure. In *3rd USNIX Symposium on Internet Technologies and Systems (USITS '01)*, pages 49–60, San Francisco, CA, USA.

[Sharma et al., 2005] Sharma, A., Bestavros, A., and Matta, I. (2005). dpam: a distributed prefetching protocol for scalable asynchronous multicast in p2p systems. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 2, pages 1139–1150 vol. 2.

[Tran et al., 2004] Tran, D., Hua, K., and Do, T. (2004). A peer-to-peer architecture for media streaming. *Selected Areas in Communications, IEEE Journal on*, 22(1):121–133.

[Vega-Rodriguez et al., 2005] Vega-Rodriguez, M. A., Gomez-Pulido, J. A., and Sanchez-Perez, J. M. (2005). Case studies in tele-education: Research and projects. *Journal of Universal Computer Science*, 11(9):1555–1567.

[Yiu et al., 2007] Yiu, W.-P., Jin, X., and Chan, S.-H. (2007). Challenges and approaches in large-scale p2p media streaming. *IEEE Multimedia*, 14(2):50–59.

[YU et al., 2007] YU, J., HU, X., YI, P., and Binqiang, W. (2007). Quantitative evaluation on routing performance on network-level of structured p2p network. *Computer Engineering*, 33:7–9.

[YU and WANG, 2007] YU, J. and WANG, B.-q. (2007). Redirect-based routing optimizing method on structured p2p network.computer engineering and applications. *Computer Engineering and Applications*, 43:132–134.

[Zhou and Liu, 2005] Zhou, M. and Liu, J. (2005). A hybrid overlay network for video-on-demand. In *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, volume 2, pages 1309–1313 Vol. 2.