2010 10th IEEE International Conference on Computer and Information Technology (CIT 2010)

Server Placement for Peer-to-Peer Live Streaming Systems

Xiaoqun Yuan*, Hao Yin†, Xuening Liu†, Changlai Du†, Geyong Min‡
*Department of Electronics and Information Engineering, Huazhong University of Science Technology, Wuhan, 430074, China
† Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
‡Department of Computing, University of Bradford, Bradford, BD7 1DP U.K.
Email: *yuanxiaoqun@smail.hust.edu.cn
†{ h-yin, liuxn99, duchanglai } @ mails.tsinghua.edu.cn
‡g.min@brad.ac.uk

***Abstract*—With the rapid growth of the scale, complexity and heterogeneity of Peer-to-Peer (P2P) systems, the handle of the peer's network-oblivious traffic and self-organization has become a great challenge. A potential solution is to deploy servers at different locations as appropriate. However, due to the unique features and requirements of P2P systems, the traditional placement models cannot yield the desirable service performance. In this paper, we propose an efficient server placement model for P2P live streaming systems. Compared to the existing solutions, this model takes the Internet service provider (ISP) friendly problem into account and can reduce the cross-network traffic among ISPs. Meanwhile, the peers' contribution is introduced into our model, which makes it more suitable for P2P live streaming systems. Moreover, we deploy servers based on the theoretical solution subject to practical data and apply them to realistic live streaming applications. The experiment results show that our model can reduce the amount of cross-network traffic, improve the efficiency of systems, has a better adaptability to Internet environment, and is more suitable for P2P systems than the traditional placement models.**

***Keywords*-Peer-to-Peer; Live Streaming; Server Placement; Cross-Network Traffic;**

I. Introduction

Live streaming has long been projected as the "killer-application" for the Internet. Only over recent years, this application has been prevalent due to the deployment of increased bandwidth in last-mile. P2P is one of the most popular live streaming delivery technologies owing to its scalability and efficiency, but there are still some challenges to be tackled, such as cross-networks traffic and performance guarantees [1]. To deal with these problems, some proposals tried to address these problems through controlling and managing peers, such as P4P[10], Taming the Torrent[9] and efficient peer matching algorithms[11]. Besides these approaches, a effective solution to these challenges is to deploy servers in proper locations. For example, several proposals have suggested deploying cache in AS and using them to cache fraction content based on some criteria to improve service performance of systems and reduce cross-network traffic[7][8]. While these techniques help reduce cross-network traffic, they are just suitable for these cachable content applications. What's more, there is little work focusing on server placement for P2P live streaming systems.

The problem of server placement has been well studied for web caching and content delivery network (CDN). As a result, many placement algorithms have been proposed and analyzed [2][3][4]. Specifically, there are three types of placement models based on the objects to be optimized: 1) Facility Location problems, which choose the potential locations based on the Minimum placement cost (referred to as Minimum Placement Cost Model (MPCM)); 2) k-Median Problems, which have no cost for opening deployment servers and just minimize the connection cost (referred to as Minimum Connection Cost Model (MCCM))[2][4]; and 3) Integrated Optimization Problem, which take into account both the connection cost and the placement cost(referred to as Liner Trade-off Cost Model(LTCM))[3][5]. Unlike these traditional server placement models, P2P server placement models must take into account the contribution of peers and adapt to the dynamic feature of P2P systems except for the server placement cost and user performance (connection cost). These requirements limit the flexibility of the traditional placement models and make it necessary to develop a new placement model for P2P systems.

In this paper, we focus on the study on server placement for P2P live streaming. We first propose a general P2P server placement model, called Multiply Trade-off Cost Model (MTCM), for P2P systems. Based on the features of P2P live streaming, we further formulate a suitable server placement model and apply it to the real-world system. The major contributions of this paper can be summarized as following: i) we present a general server placement model for P2P systems, which can be flexibly applied in practical systems with different interests; ii) Based on this general model, we formulate a suitable server placement model for our P2P live streaming system. This proposed placement model introduces the contribution of peers into the placement cost function, which it more suitable for the P2P live streaming systems. Moreover, this model takes into account the Internet service provider (ISP) friendly problem, which can reduce the amount of cross-network traffic among ISPs.

978-0-7695-4108-2/10 $26.00 © 2010 IEEE
DOI 10.1109/CIT.2010.88

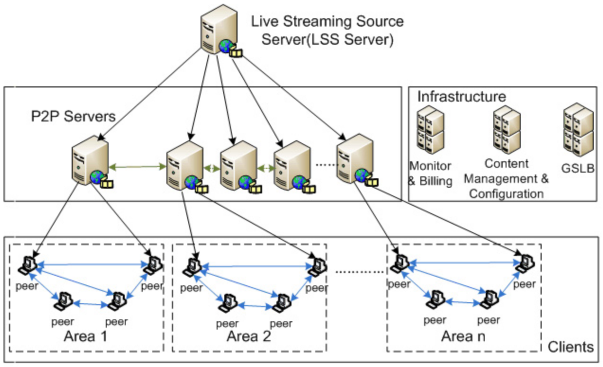


Figure 1. Overview of P2P Live Streaming System

The rest of this paper is organized as follow. In Section II, we firstly present the overview of our P2P live streaming system, then propose a general placement model for P2P systems and a suitable placement model for our P2P live streaming system. Section III evaluates the performance of our model using trace-based simulation. Section VI concludes the paper.

II. Problem Formulation

In this session, we focus on how to formulate server placement model for P2P live streaming systems.

A. the P2P Live Streaming System

Firstly, we present the basic P2P live streaming system model in this session, including the underlying assumptions and notations, as listed in Table I.

Our system is composed of three major components: the live streaming source server (LSS server), P2P servers and peers, as illustrated in Fig.1. In the system, the LSS server generates streaming resource and deliveries it to P2P servers deployed in different locations; P2P servers receive streaming and forward it to their clients (peers). All peers served by P2P servers in the same location are organized as a mesh based P2P subsystem, peers in this subsystem which upload and download the streaming in P2P fashion.

Table I
Notation

Notation	Definition
U	The total bandwidth capacities of servers.
M	The number of potential locations which servers may be selected to deploy.
r_s^k	The capacities of servers in region k.
ρ_k	The average fraction contribution capacity of peers of server k
R	The live streaming rate of our P2P live streaming system.

To build our server placement strategy model, which can reveal the essential aspects of practical systems, yet be still simple enough to yield relevant insights, we make the following assumptions: i) All peers are divided into some regions, each of which has a potential location for server placement and the latency for peers to access servers in the same region is fixed. ii) We know the density of possible peers in each region and the connection map of the Internet, which means we not only get the delay of server to peers within the same region, but also get the delay of server to peers in different regions. iii) All the peers in the same region are organized as a mesh based P2P subsystem, and peers in different region do not exchange information with each other.

B. A General Server Placement Model for P2P Systems

1) Problems Formulation: Essentially, given a constrained server capacity U, our server placement model can be described to determine which potential locations are selected to deploy the servers, and how much bandwidth capacity are presented for each location with diverse deployment cost and connection cost. Based on the definition mentioned above, this problem can be formulated as a classical 0-1 multiple knapsack problem with respect to different objective functions:

Minimize:

$$\sum_{i=1}^{M}\sum_{j=1}^{M} L_{ij}{}^{p} B_i C_i x_i y_{ij} \tag{1}$$

Subject to:

$$\sum\nolimits_{i=1}^{M} B_i \leq U \tag{2}$$

$$\sum_{\mathrm{i}=1}^{\mathrm{M}} \mathrm{y_{ij}} = 1, 1 \leq i, j \leq M \tag{3}$$

where L_{ij} denotes the connect cost, B_iC_i presents the placement cost, $p \geq 0$ is the control parameter which determines the participating degree of the connection cost on server deployment, $x_i \in \{0, 1\}$ denotes whether a potential location $1 \leq i \leq M$ is selected to be deployed, C_i denotes the cost for deploying unit server in region i, $y_{ij} \in \{0, 1\}$ denotes whether peers in region j are served by servers in region i, and L_{ij} denotes the connection cost (such as the latency) for peers in region j to access servers in region i. Constraint (2) denotes that the total demand capacity from all deployed servers is no more than the expected total capacity and Constraint (3) expresses that peers in one region are served by servers deployed in a certain region.

Obviously, this model take into account both of the important aspects indicated by MPCS and MCCS, with the control parameter $p \geq 0$ providing flexible design choices. For instance, when p is customized to 0, it becomes the classical MPCS. When p is customized to 1, it makes a balanced consideration between MPCS and MCCS.

C. Cost Function Determination for P2P Live Streaming

1) Placement Cost Function: The core goal of our placement model is to minimize the placement cost, so we need to define placement cost function in our P2P live streaming system firstly. Substituting the client number and unit placement cost of each region into model (1), then we can describe the placement cost function f_{pc} as:

$$f_{pc} = \sum_{i=1}^{M} B_i C_i x_i = \sum_{i=1}^{M}\sum_{j=1}^{M} n_j C_i x_i y_{ij} R \tag{4}$$

where n_j denotes the peer number of region j.

Equation (4) defines an absolute total placement cost and needs to be normalized as follows: i) use the peer density instead of the peer number for each region; 2) classify potential locations based on the unit placement cost and customize all value of the same class to the highest value of this class, then customize the value of the highest unit placement cost to 1 and normalize the placement cost of the other regions.

After normalizing these parameters, we get:

$$\sum_{i=1}^{M} B_i C_i x_i = \sum_{i=1}^{M}\sum_{j=1}^{M} n_j d_j c_i x_i y_{ij} \tag{5}$$

where c_i denotes the unit placement cost of region j after normalization and $d_j = \frac{n_j}{\sum_{j=1}^{M} n_j}$, expressing the client density of region j.

Unlike the traditional content service systems, P2P live streaming systems are more scalable and cost-effective because of the peers' contribution. Studies[6][12] have shown that the scalability of P2P live streaming systems is the function of the average fraction contribution capacity of peers ρ_j. The scalability model shows that only when $\rho_j < 1$, servers need to provide extra bandwidth capacity to maintain the view smooth of peers. In fact, the average contribution bandwidth is less than download bandwidth, which means $\rho_j < 1$. So ρ_j has an significant influence on the server placement cost and the user experience and becomes the important factor we need to take into account.

We define the fraction contribution capacity of servers as $\frac{r_s^j}{r_s^j + r^j}$. For region j, if $\rho_j < 1$, from our P2P Scalability Model, we have

$$\frac{\mathrm{r_s^j}}{\mathrm{r_s^j} + \mathrm{r^j}} = 1 - \rho_j \tag{6}$$

Substituting Eq. (6) and $r_s^j + r^j = n_j * R$ into Eq. (5), averaging and normalizing the related parameters, the placement cost function can be finally described as follow:

$$f_{pc} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{M}} \mathrm{B_i C_i x_i}}{\sum_{\mathrm{i=1}}^{\mathrm{M}} \mathrm{B_i}} = \sum_{i=1}^{M}\sum_{j=1}^{M} d_j c_i (1-\rho_j) x_i y_{ij} \tag{7}$$

2) Connection Cost Function: Another important goal of our model is to keep user experience from decreasing too much, so our model needs to take into account user experience. However, user experience of live streaming is affected by available bandwidth from the streaming source to end users, and it is harder to estimate available bandwidth than latency in the dynamic Internet environment. Fortunately, the existing studies [13][14][15] have demonstrated that it is possible to substitute delay for available bandwidth. For instance, Downey [13] has shown that the available bandwidth is the function of the round-trip time (RTT). Leighton [15] has found that the throughput of two nodes had a negative correlation with the response delay. Chen [14] has also shown that the user experience of live application has a negative correlation with the response delay. So we consider that the available bandwidth has a negative correlation with the RTT, which means that the smaller RTT between the streaming source and end user, the better of the user experience.

In our system, there are two types of RTT: the intra-RTT between two nodes within the same region and the inter-RTT between two servers in different regions. In order to simplify, we assume that the intra-RTT in the same region has the same value, and these value of Intra-RTT can be ignored in comparison with the inter-RTT. Then, the connection cost can be defined as follows:

$$f_{cc} = \sum_{\mathrm{i=1}}^{\mathrm{M}}\sum_{\mathrm{j=1}}^{\mathrm{M}} \mathrm{L_{ij} x_i y_{ij}} = \sum_{i=1}^{M}\sum_{j=1}^{M} n_j L_{ij} x_i y_{ij} \tag{8}$$

where L_{ij} denotes the RRT between the peers in region j and the servers in region i. Obviously, the value of inter-RRT is finite and has a constraint as follows:

$$L_{ij} = \begin{cases} \varphi(l_{ij}, l_{jj}) & \text{if } \varphi(l_{ij}, l_{jj}) \le L_{thr} \\ \infty & \text{the others} \end{cases} \tag{9}$$

where L_{thr} is the threshold of delay, denoting the maximal delay peers can be bared, and L_{ij} is the function of intra-RRT of two nodes in region j and the inter-RRT of two nodes in region j and region i, for example, $L_{ij} = k_1 * l_{ij} + k_2 * l_{jj}$, $k_1, k_2 \in R^+$. Since $l_{jj} << l_{ij}$, we have $L_{ij} = k * l_{ij}$, $k \in R^+$. Constraint (12) shows that if the value of inter-RRT is larger than the threshold, the link from the servers to the peers will be dropped off because of the bad user experience.

Averaging and normalizing Eq. (8) and (9), the connection cost function can be described:

$$f_{cc} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{M}}\sum_{\mathrm{j=1}}^{\mathrm{M}} \mathrm{L_{ij} x_i y_{ij}}}{\sum_{\mathrm{i=1}}^{\mathrm{M}} \mathrm{n_j}} = \sum_{i=1}^{M}\sum_{j=1}^{M} d_j l_{ij} x_i y_{ij} \tag{10}$$

Subject to:

$$l_{ij} = \begin{cases} k * l_{ij} & \text{if } l_{ij} \le L_{thr}/k \\ \infty & \text{the others} \end{cases} \tag{11}$$

D. Server Placement Model for our P2P live streaming system

Substituting Eqs. (7), (10) and (11) into model (1) and customizing the control parameter p to 0.5, our model can be described as fellow:

Minimize:

$$\sum_{i=1}^{M}\sum_{j=1}^{M} d_j c_i (1-\rho_j)(d_j L_{ij})^{0.5} x_i y_{ij} \tag{12}$$

Subject to:

$$\sum_{\mathrm{i=1}}^{\mathrm{M}} \mathrm{y_{ij}} = 1, 1 \le i, j \le M \tag{13}$$

$$\sum_{\mathrm{j=1}}^{\mathrm{M}} \mathrm{n_j y_{ij} R} \le B_i, 1 \le i, j \le M \tag{14}$$

$$\sum\nolimits_{i=1}^{M} B_i \le U, 1 \le i \le M \tag{15}$$

$$l_{ij} = \begin{cases} k * l_{ij} & \text{if } l_{ij} \le L_{thr}/k \\ \infty & \text{the others} \end{cases} \tag{16}$$

Constraints (14) and (15) are the bandwidth constraints. That is, the total demanded bandwidth capacity from all the peers served by the servers in region j is no more than the allocated capacity and the total demanded bandwidth capacity form all peers is no more than our expected bandwidth capacity U. Constraint (13) denotes that all the peers in region j are not serviced by the servers in different regions. $p = 0.5$ denotes that the goal of our placement model is to minimize the placement cost while keeping the user experience from decreasing significantly.

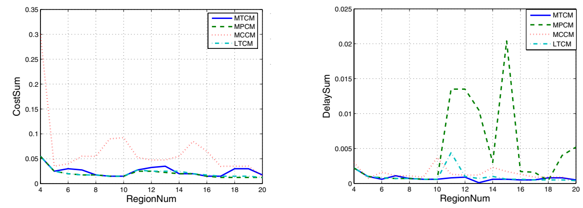
III. Performance Analysis

In this section, we present a theoretical solution to our P2P system with some measured and statistical data, then we compare the performance of our model to the other three traditional models.

A. Theoretical Solution

Model (12) is a classical multiple (0,1) knapsack problem with multiple constraints, which can be solved by many classical algorithms [17]. Then we apply some practical data to our model and get the theoretical solution. In this paper, the placement cost for each location is from empirical data of deploying servers in each region in China, the placement delay is from ping data of each two potential locations in corresponding regions and the peers density is from the table II[16] (in China, all Internet users are divided into 34 administrative regions, with each operated by a provincial ISP. Since reducing the cross-network traffic is our goal of placement model, we divide the Internet user distribution into 34 regions. Then we select 20 from these 34 regions as potential locations to meet the demand of our practical system and assume the peer density is the same as the Internet user density for each region).

For our solution, $x_i = 1$ means that the potential location in region i is chosen to accommodate servers, $y_{ij} = 1$ denotes that the peers in region j is served by servers in region i, $\sum_{j=1}^{M} Rn_j x_i y_{ij}$, $x_i = 1$ and $1 \le i, j \le M$ expresses the demand bandwidth capacity B_i for servers in region i and $\sum_{i=1}^{M} B_i x_i$, $1 \le i \le M$ denotes the total demand bandwidth capacity for our system.



(a) Vs.Total Placement Cost (b) Vs.Total Placement Delay

Figure 2. Relationship between Region Number and Total Placement Cost and Total Delay for Different Placement Models($\rho = 0$)

B. Analysis of Theoretical Solutions

Fig.2 presents the results of two goals for four models: the two classical placement models of MPCM, MCCM and two trade-off placement models of MTCM and LTCM, with the region number increasing from 4 to 20 with step 1. We customize p to 0.5 for MTCM , and minimize $0.7*f_{pc}+0.3*f_{cc}$ for LTCM for two reasons: i) our goal is to minimize placement cost while keeping the user experience form decreasing significantly; ii) the two trade-off models use a similar participating degree of the connection cost to select deployment locations. We customize ρ to 0 so that we can check whether our model is suitable the traditional live streaming systems.

Table II
Population and Density of Some Regions in China

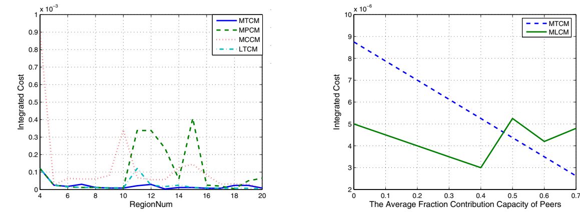
Notation	AH	BJ	CQ	FJ	GD	HEB	HEN
Population(M)	3.37	4.68	2.2	5.16	18.31	6.31	5.17
Percent	2.5%	3.4%	1.6%	3.7%	13.4%	4.6%	3.8%
Notation	HLJ	HUB	HUN	JS	JX	LN	GZ
Population(M)	3.66	5.32	4.08	10.27	2.85	4.83	1.42
Percent	2.7%	3.9%	3.0%	7.5%	2.1%	3.5%	1.0%
Notation	SC	SD	SX	SXS	YN	ZJ	
Population(M)	6.9	11.26	3.8	3.95	2.75	9.77	
Percent	5.0%	8.2%	2.8%	2.9%	2.0%	7.1%	

Fig.2(a) plots the relationship between the total placement cost and the number of regions and Fig.2(b) plots the placement delay. From Fig.2(a), we can find that MCCM has the worst performance for placement cost while MPCM has the best performance. The performances of MTCM and LTCM are slightly worse than MPCM but much better than MCCM. This is because that the goal of MPCM is just to minimize the placement cost and MCCM to delay. But MTCM and LTCM take into account both the placement cost and connection cost and make a tradeoff of them. Fig.2(b) shows the similar results for the connection cost.

Comparing Fig.2(a) with Fig.2(b), we find that both MTCM and LTCM can reduce the placement cost greatly while keeping the user experience from decreasing too much, and they have similar performance for placement cost and connection cost. So we can not tell which model has a better performance. In order to further distinguish the performance difference of the two models for the traditional live streaming systems, we define a integrated cost $f_{pc} * f_{cc}$, which denotes the integrated performance of placement models. Obviously, the lower the integrated cost is, the better performance the placement strategy may achieve. Fig.3 presents the relationship between integrated cost and region number and ρ for different placement strategies. Fig.3(a) plots the relationship between the integrated cost and region number for these four models, with the same parameters shown in Fig.2. From Fig.3(a), we find that MTCM and LTCM have much smaller integrated cost than MPCM and LTCM, and similarly, there is no explicit performance of integrated cost between LTCM and MTCM.

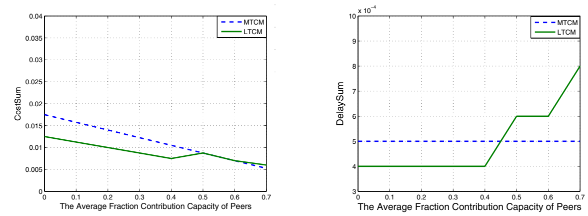
However, ρ is an important factor that server placement needs to take into account for our P2P live streaming system. Fig.3(b) and Fig.4 plot the results of three goals for MPCM and LTCM after introducing different values of ρ into P2P systems. In Fig.3(b) and Fig.4, the three goals are integrated cost, placement cost and delay, the number of region is customized to 20, the range of ρ is from 0 to 0.7 and the step is 0.05. As shown in Fig.3(b) and Fig.4, the slopes of these three criterions for LTCM change three times: the first at $\rho = 0.4$, the second at $\rho = 0.5$ and the third at $\rho = 0.6$, while these slopes of MTCM is invariable. So we can find that as ρ increases, LTCM changes its selecting location and bandwidth capacity while MTCM has the same location, just changes the demanded bandwidth capacity. This is due to the different optimization strategies of the two models. MTCM integrates all factors into a single optimization objective model (13), so ρ can be considered as multiplication coefficient, just affecting the capacity and having nothing to do with the location selection. On the contrary, LTCM formulates the placement problem as a two-objective optimization model $(0.7 * f_{pc} + 0.3 * f_{cc})$, and ρ just affects f_{pc}. So, with $\rho \nearrow$, $(0.7 * f_{pc}) \searrow$, and f_{cc} affects the decision-making greatly, which may lead to the change of solution. Clearly, our model has a better adaptability of Internet environment and is more suitable for P2P systems.

Obviously, comparing with other placement models, our placement model is more general and can be applied to different service patterns by customizing ρ to corresponding value. When $\rho = 0$, our model degenerated into the traditional server placement model, has a good performance for the integrated cost and is suitable for these patterns and



(a) Vs.RegionNum (b) Vs.ρ

Figure 3. Relationship between Integrated Cost and ρ and Region Number for Different Placement Models



(a) Vs.Total Placement Cost (b) Vs.Total Delay

Figure 4. Relationship between ρ and Total Placement Cost and Total Delay for Different Placement Models(Region Number=20)

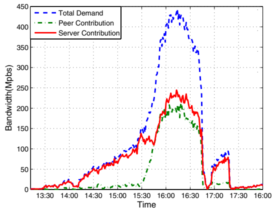
when $\rho > 0$, our model has a better adaptability to the Internet environment and is more suitable for P2P systems because of the contribution of peers.

IV. P2P Live Streaming Systems Analysis

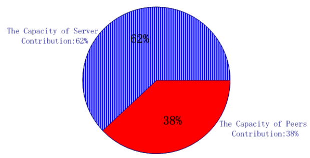
Based on the theoretical solution, we deploy servers in selected locations and allocate corresponding capacity to these servers in China. Then we apply our P2P system to several Internet live broadcast applications to further deep into the characteristics of our placement model. For the proposes of this study, we focus on one of our live broadcast, Premier Wen Talks Online with Citizens on Feb 28, 2009. The live broadcast starts at the time 14:30pm, lasts two and a half hours, and ends at time 17:00pm. During this period, the event begins at time 15:00pm, and ends at time 17:00pm (Fig.5(a)). The peak number of pee for this broadcast was more than 120,000 with an total demand bandwidth of more than 35Gbps.

A. Cross-Network Traffic

Since servers in the same location organize their peers into a P2P subsystem, we can manage and monitor our P2P system effectively using some technologies. For example, through DNS redirection, P2P subsystem can be aware of the distribution of its peers, and then assign an appropriate list of neighbors to each peer. So we can roughly evaluate the cross-network traffic of our system by analyzing the veracity of DNS-based redirections [6]. Results show more than 70% of peers can be assigned to the region into which they fall. Thus, we can reduce the cross-network traffic significantly by deploying servers in appropriate locations.

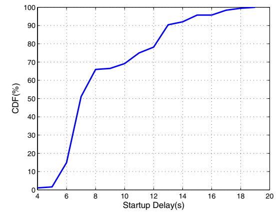


(a) One Server

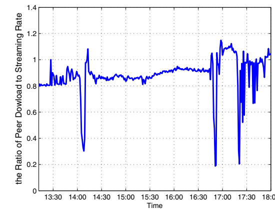


(b) P2P System

Figure 5. The Average Contribution of Server and Peer in Live Broadcast of our System



(a) Average Startup Delay



(b) the Ratio of Average Peer Download to Streaming Rate

Figure 6. The Startup Delay and Average Peer Download in Live Broadcast of our System

B. Placement Cost

Fig.5 presents the contribution of peer and server for this live broadcast application of our system. Fig.5(a) plots the total demand, server contribution and peer contribution bandwidth of peers served by one server. Fig.5(b) plots the fraction bandwidth of server contribution and peer contribution in this application. Fig.5(a) reveals that the peers' contribution is increasing with the creasing of peers' total demand(the fraction of peers upload contribution is less than 10% at time 15:30 pm while it is more than 40 % at time 15:50 pm). Fig.5(b) shows that the fraction download contribution of peers is 38%. The result demonstrates that we can satisfy peers' requirements by providing much less bandwidth, for example, 62% of total demand bandwidth. Obviously, we can improve the utilization of P2P systems and save placement cost by using our placement model, especially at the high demand bandwidth time.

C. User Experience

1) Startup Delay: The work of J. Liu suggests that the startup delay is a crucial factor in user experience – users are likely to get frustrated and leave if they perceive high startup delays[18]. Fig.6(a) shows average startup delays of this broadcast application. Results are as fellows. First, our results indicate that about 55% of peers have startup delays between 5s and 8s, from 15% to 70%. Second, 96% of the peers wait less than 15s for playback to commence from the time the user clicks on the stream hyperlink. Obviously, comparing the other live streaming systems like CoolStreaming in which most peers have startup delays greater than 30s[19], our P2P live streaming system can provide significantly faster startup performance.

2) Peer Download: We use the ratio of average peer download to streaming rate to evaluate the quality of the streaming viewing of peer. Clearly, the larger the ratio is, the higher the quality is. Fig.6(b) plots the relationship between this ratio and time. From Fig.6(b), we find that: 1) at most time during this application, this ratio is more than 80%, and at some time, the ratio is up to 115%, which means that during this live broadcast, users enjoy high streaming viewing quality, and 2) there are serval disturbance during this period: one is at about 14:10 pm, the other is at about 17:50 pm. the first disturbance happens because of the switching of service, the second disturbance is due to the leaving of most peers at the end of live broadcast.

The results of Fig.6 demonstrate that by deploying servers in selected locations, our P2P live streaming system not only provides faster startup performance but also maintains the high quality of the streaming viewing of peers.

V. Conclusion

In this paper, we have presented an efficient server placement model for the P2P systems which can be flexibly applied in practical systems with regard to different interests. Then we have introduced peers' contribution into our placement model and formulated a suitable placement model for our P2P live streaming system. Moreover, we have evaluated the performance of different models using trace-based simulation, achieved our theoretical deployment solution based on practical data, deployed servers in proper locations and applied them to practical live broadcast applications to test our design goals.

The results of theoretical analysis and practical experiment have shown that our placement model has the important advantages including: 1) our model unifies the important aspects indicated by full new MPCM and full new MCCM models, has a better adaptability of Internet environment and is more suitable for P2P live streaming systems than the traditional placement models; 2) our model takes into consideration of the peers' contribution, which can improve the efficiency of P2P systems at the time of high demand bandwidth; 3) our model involves the ISP friendly problem into account, and thus reduce the cross-network traffic among ISPs.

Acknowledgment

This work was funded by the Project 60873254 supported by NSFC, the Project 20090460317 supported by China Postdoctoral Science Foundation, and the Tsinghua-ChinaCache CDN Research Institute Project.

References

[1] P.Rodriguez, S-M.Tan and C.Gkantsidis, *On the Feasibility of Commercial, Legal P2P Content Distribution*, in Proceedings of SIGCOMM Comput Commun. Rev, Vol. 36, No. 1, pp.75-78 2006.

[2] M.Charikar, S.Guha, $\hat{E}$.Tardos and D.B.Shmoys, *A constant-factor approximation algorithm for the k-median problem*, In Proceedings of the 31st Annual ACM Symposium on Theory of Computing, pp.1-10, May 1999.

[3] K.Jain and V.V.Vazirani, *Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation*, In Journal of the ACM, pp. 274-296, 2001.

[4] C. W.Cameron,S.H.Low and D. X.Wei, *High-Density Model for Server Allocation And Placement*, In Proceedings of ACM SIGMETRICS 02, Vol. 30, No. 1, pp:152-159, 2002.

[5] M. P$'$al,$\hat{E}$.Tardos and T.Wexler, *Facility Location with Nonuniform Hard Capacities*, In Proceedings of the 42nd IEEE Symposium on the Foundations of Computer Science, pp:329-338, 2001.

[6] H.Yin, X.Liu, V.Sekar, F.Qiu, C.Lin, H.Zhang and B.Li, *Design and Deployment of a Hybird CDN-P2P System for Live Video Streaming: Experiences with LiveSky*, In Proceedings of the seventeen ACM international conference on Multimedia 2009, pp. 25-34, 2009.

[7] O. Saleh and M. Hefeeda, *Modeling and Caching of Peer-to-Peer Traffic*, In Proceedings of Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols '06, pp.249-258, 2006.

[8] M. Hefeeda and O. Saleh, *Traffic modeling and proportional partial caching for peer-to-peer systems*, ACM Transactions on Networking, Vol 16, No.6, pp.1447-1460, 2008.

[9] D R. Choffnes and F E. Bustamante, *Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems*, in Proceedings of SIGCOMM '08, Vol. 38, No. 4, pp:363-374, 2008.

[10] H. Xie, et cl,*P4P: Provider Portal for Applications*, in Proceedings of SIGCOMM '08, Vol. 38, No. 4, pp: 351-362, 2008.

[11] C-H. Hsu and M. Hefeeda, *ISP-Friendly Peer Matching without ISP Collaboration*, in Proceedings of the 2008 ACM CoNEXT Conference, 2008.

[12] X. Yuan, H.Yin, X.Liu and C.Du, *A Dynamic Resource Provision Algorithm for Multi-channel P2P Live Streaming Systems*, technology report. http://www.cdnlab.org/index.php/staff/38-xiaoqun-yuan.

[13] A B. Downey, *Using pathchar to estimate Internet link characteristics*,in Proceedings of SIGCOMM'99, 1999.

[14] K. Chen, et cl, *Quantifying Skype User Satisfaction*,in Proceedings of SIGCOMM'06, 2006.

[15] T. Leighton, *Improving Performance on the Internet*,ACM QUEUE , Vol. 6, No. 6, pp.20-29, Oct,2008.

[16] *Statistical Survey Report on The Internet Development in China*, http://www.cnnic.net.cn/en/index/0O/02/index.htm.

[17] A. Martello and P. Toth, *Knapsack Problems:Algorithms and Computer Implenentations*, Wiley-Interscience Series In Discrete Mathematics And Optimization , pp.157-188, 1990.

[18] J. Liu, S. G.Rao, B. Li and H. Zhang, *Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast*, In Proceedings of the IEEE, Special Issue on Recent Advances in Distributed Multimedia Communications, Vol.96, pp.11-24, 2008.

[19] B. Li, S. Xie, G. Y. Keung, J. Liu, I. Stoica, X. Zhang and H. Zhang, *An Empirical Study of the CoolStreaming System*, IEEE Journal on Selected Areas in Communications, Vol.25, pp.1627-1639, 2007.