Towards a Trustworthy and Controllable Peer-Server-Peer Media Streaming: An Analytical Study and An Industrial Perspective

Zhijia Chen, Hao Yin, Chuang Lin, Xuening Liu, Yang Chen*
Department of Computer Science & Technology, *Department of Electronic Engineering
Tsinghua University, Beijing, China
{zjchen, hyin, clin, xliu}@csnet1.cs.tsinguha.edu.cn, chenyang04@mails.tsinghua.edu.cn

Abstract—**Peer-to-Peer technology gives novel opportunities to define a cost-effective multimedia streaming application, but at the same time, it brings a set of technical challenges due to its dynamic and heterogeneous nature. To guarantee QoS and facilitate management in large scale high-performance media streaming, we extend the current P2P networking towards a novel Peer-Server-Peer (PSP) architecture for media streaming, in which carefully-deployed servers form a Trustworthy and Controllable overlay network to stream P2P cluster peers. An analytical model is presented to calculate the Quality of Experience (QoE) and mapping QoS parameters to prove the effectiveness of Peer-Server-Peer streaming. Joint with the efforts in industry, we explore the feasibility of PSP streaming in the historical context of "Demand Economy" for media streaming and "Best Effort" Internet. The value of this paper lies not in its analytical study of this promising PSP concept with practical implementation but also its insight industrial perspective to attract further application.**

Keywords-P2P ;streaming; QoS; Peer-Server-Peer

I. INTRODUCTION

Due to its advantage in coping with the scalability and bandwidth bottleneck in the traditional Client/Server paradigm, Peer-to-Peer (P2P) technology serves as a promising solution for the popular commercial Internet multimedia streaming and thus there have been tremendous efforts in the design and experimentation of media streaming systems in recent years (i.e, the success of ESM [3] and Coolstream [2]). However, cost-effective large-scale media streaming still remains to be an elusive and challenging goal [1]. Current P2P technology facilitates dynamical utilization of the available resources in network to scale streaming with a cost of losing central manageability and robustness thus seriously limited in providing security and guaranteeing Quality of Service(QoS) which are both crucial for large-scale media streaming application. In lack of efficient server management and support, P2P nodes do not guarantee high availability because of the dynamic and heterogeneous nature of P2P network.

Thus, to meet the needs of large-scale high-performance media streaming, we extend the current Peer-to-Peer networking towards a trustworthy and controllable Peer-Server-Peer (PSP) media streaming, in which carefully-deployed servers facilitates central management and guarantees QoS while P2P clusters promote scalability with the guide of control servers. Based on a model-based analysis and an industrial perspective for application and implementation, the PSP media streaming concept would serve as an ideal solution for large-scale media streaming with QoS and security enhancement, promising much for exploration in both academy and industry.

The rest paper is organized as follows: section 2 introduces our proposed PSP streaming architecture; section 3 designs an analytical model to study its effectiveness; section 4 explores its feasibility from industrial perspective; Section 5 concludes this paper and highlights the contribution.

II. A NOVEL PSP STREAMING ARCHITECTURE

In our Peer-Server-Peer (PSP) structure, media content is first distributed among trusted servers and finally reaches different clusters of peers for their P2P distribution. All those carefully-deployed servers (some may belong to Content Distribution Network (CDN), which is formed by dedicated proxy servers for content distribution) form a Trustworthy and Controllable overlay network, through which servers are used to provide initial content, guide streaming traffic to achieve overall traffic optimization, and conduct AAA and key distribution. Strategically deployed CDN video servers around the Internet enable end users to obtain streaming video from one of the nearby servers to reduce the end-to-end delay and overall network congestion. Meanwhile, the combination of P2P networks around those servers facilitates peers to freely transmit within their P2P clusters to achieve maximum scalability.

In implementation, with no need to build a new infrastructure, we can directly implement our logical PSP concept by arranging servers and peers in the current network with available P2P protocol. Actually we begin with a hybrid of P2P and CDN streaming network, in which the best features of P2P and CDN are combined to deliver the media content from source server in CDN networks to end cache servers who are responsible for building an ad-hoc cluster of P2P peers to transmit the content between each other (Fig. 1). In this way,

---

This work is supported by National Science Foundation of China (No.60673184, No.60473086, No.60473087, and No.90412012); National 863 Program (No.2007AA01Z419); National Basic Research Program of China (973 Program) (No.2007CB310806).

CDN guarantees the QoS and P2P enhances the scalability of streaming system.

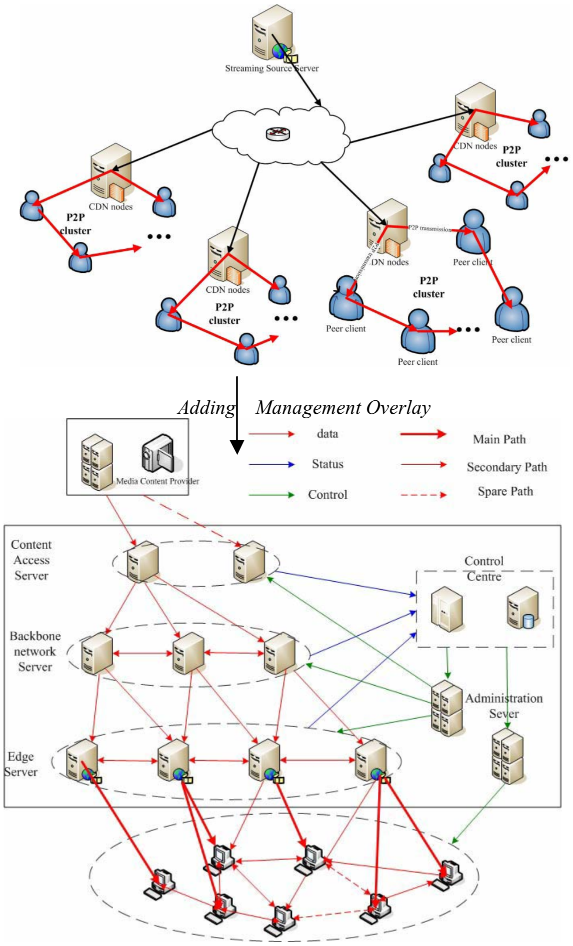

Figure 1. Peer-Server-Peer Streaming Architecture

Then we add a management overlay network above this CDN-P2P Hybrid. This management overlay is responsible for organizing three layers of P2P networks: 1) ISP P2P: to design optimal data path from media source provider to every Internet Service Provider and guarantee their efficient sharing between each other. 2) End Server P2P: to optimally place end servers and to reload traffic from full-loaded server to less-loaded server. 3) User P2P: employ a pure multiple-source to multiple-receiver P2P to achieve maximum scalability in each cluster. Different from traditional C/S architecture, servers in our structure only serve as initial content provider, temporary streaming supporter and overall guider, thus increasing neglectable management overhead without affecting cluster scalability.

Under this PSP architecture, the streaming process at end user can thus be described as follows:

1. A new peer *P* sends a requesting message to the management server *S*

2. *S* redirects *P* to the cluster $L_i(S_i)$,where S_i is the streaming server of P2P cluster $L_i(S_i)$

3. If the available bandwidth from supporting peers in $L_i(S_i)$ outweighs the requesting bandwidth of P , *Stream(P, Supporting Peers)*

4.Else if available S_i bandwidth outweighs the requesting bandwidth of P , *Stream (P,* S_i*)*

5. After *P* completes receiving streams, it becomes a supporting peer in cluster $L_i(S_i)$

6. Otherwise if the cluster cannot serve *P* (no available bandwidth or requested media file), *S* conducts an overall search, returning an optimized and available server for *P*, go to 3

7. If no streaming resource is found, reject *P*

Base on our former success of TrustStream [4], we further combine the advantage of advanced coding techniques (i.e 3D-Wavelete coding) and networking techniques to achieve unprecedented security, scalability, and certain QoS simultaneously. With the benefits of new PSP architecture, QoS can be maintained with the guarantee of streaming servers and enhancement from P2P peers. Scalability is achieved by well adopting the pure gossip-based P2P in content distribution. Heterogeneity is handled by delivering only the layers of content that a receiver can manage with the coding technology. The security issues are addressed by combining our key distribution mechanism and key-embedding scheme [5] in our proposed multicast topology [4] with the separation of the security management from data transmission.

III. ANALYTICAL STUDY OF PSP MEDIA STREAMING

A. *Service Guaranteed Period*

To evaluate the service provided by the PSP streaming, we define a general metric in view of Quality of Experience (QoE) of end users. The measurement for QoE would make operators, device providers and service providers more convenient to judge and improve the service they provide. Specially we define Service Guaranteed Period (SGP) to indicate when and how long the Quality of Service could be guaranteed for peer media receiver without blur, discontinuity , and image impairment in our proposed PSP streaming.

In the initial stage when there are few requesting peers and the bandwidth of streaming servers could handle those requests, the SGP could be maintained till the number of requesting peers grow out of the capacity of servers but the Peer-to-Peer network has not grown to its mature stage to support the peer requests. We denote the discrete time t_0 as the maximum time the streaming server can directly provide guaranteed service itself when the user number is small; t_g as the starting time when the total capacity of the system with server and the P2P peers together can again support the large sum of users. Note that the unit for t_0 , t_g is time intervals rather than natural time units. Other parameters are defined as table 1.

TABLE I. PARAMETER DEFINITION FOR SGP ANALYSIS

Parameter	Definition
B	Total Bandwidth of one cluster Streaming Server
L	Length of one streaming session (time)
λ	Total number of peer requests per unit time
t	Time interval
$N(t)$	Total number of Peers in one cluster at interval t
$\varepsilon_1(t)$	Ratio of requesting peers at interval t .
$\varepsilon_2(t)$	Ratio of supporting peers at interval t .
α	Ave. bandwidth contribution of all supporting peers
β	Bandwidth required to stream a file

Thus $SGP = (0, t_0) \bigcup (t_g, t\infty)$

$$V_t(SGP) = \begin{cases} 1, t \in (0, t_0) \\ 0 \quad or \quad 1, t \in (t_0, t_g) \cup (t_\infty, \ldots) \\ 1, t \in (t_g, t_\infty) \end{cases}$$

where $t_0 = \max(t | N(t) \le \dfrac{B}{\lambda * \beta * L})$

Then it comes to the problem for calculating t_g . According to the definition of SGP, t_g is the time when the total capacity of the system is equal to the requesting bandwidth. We draw from the wisdom in paper [6] and calculate t_g based on our proposed streaming process described in section 2. We consider a multi-file system with all files having the same time length L and the same required bandwidth β . The L and β is the average value and should be to some extend represent the general situation. For simplicity, we divide the whole system into F independent files, one single file for each subsystem. Let's first consider one subsystem with the Server Bandwidth B_i , requests/unit λ_i and SGP starting time $t_{g,i}$. We assume that the bandwidth is proportionally distributed among subsystem to meet with the needs of requests/unit λ_i so that the maximum capacity could be utilized. The coming rate of requests follows a Poisson Distribution $P_n(t)$. Thus we have

$$P_n(t) = \frac{(\lambda t)^n}{n!} e^{-(\lambda t)} \begin{cases} n = 0,1,2,\cdots; \\ \lambda > 0; t > 0 \end{cases}$$

$$B_i = \frac{\lambda_i}{\lambda} * B, i = 1,2,\ldots, F$$
$$\sum_{i=1}^{i=F} B_i = B, and\ , \sum_{i=1}^{i=F} \lambda_i = \lambda$$
(1)

We define $S(t)$ as the number of supporting peers at interval t . Peers that have bandwidth and media file to support requesting peers are called supporting peers.

$$S_i(t) = N_i(t) \times \varepsilon_1(t) \tag{2}$$

During time interval t , with the best use of bandwidth from server and supporting peers, the number of new supporting peers produced between two consecutive time intervals t and $t + 1$ can be expressed as:

$$S_i(t+1) - S_i(t) = \frac{B_i}{\beta} + S_i(t) * \frac{\alpha}{\beta} \tag{3}$$

$$\Rightarrow S_i(t) + \frac{B_i}{\alpha} = (S_i(0) + \frac{B_i}{\alpha}) * (\frac{\alpha + \beta}{\beta})^t \tag{4}$$

$$\Rightarrow S_i(t) = \frac{B_i}{\alpha} \left[\left(\frac{\alpha + \beta}{\beta} \right)^t - 1 \right] \tag{5}$$

According to the definition of SGP, the total system bandwidth at $t_{g,f}$ is equal to the bandwidth from all requesting peers. Thus we have:

$$B_i + \alpha \times S_i(t_{g,i}) = \lambda_i L \beta \tag{6}$$

$$\Rightarrow B_i + \alpha \times \frac{B_i}{\alpha} \left[\left(\frac{\alpha + \beta}{\beta} \right)^{t_{g,i}} - 1 \right] = \lambda_i L \beta \tag{7}$$

$$\Rightarrow t_{g,i} = \log_{\frac{\alpha+\beta}{\beta}} \left(\frac{\lambda_i L \beta}{B_i} \right) \tag{8}$$

According to formula (1) and make the system SGP time average of that of the subsystems, we have

$$\begin{aligned} t_g = \frac{\sum_{i=1}^{F} t_i}{F} &= \frac{1}{F} * \frac{\sum_{i=1}^{F} \lg(\frac{\lambda_i L \beta}{\frac{\lambda_I}{\lambda} B})}{\lg(\frac{\alpha + \beta}{\beta})} \\ &= \frac{1}{F} * \frac{F * \lg(\frac{\lambda L \beta}{B})}{\lg(\frac{\alpha + \beta}{\beta})} \\ &= \frac{\lg \lambda L \beta - \lg B}{\lg(\frac{\alpha + \beta}{\beta})} \end{aligned} \tag{9}$$

The total number of peers with guaranteed service at time t_g can be calculated from formula (2) and (6)

$$\begin{aligned} N(t_g) &= \frac{1}{\varepsilon_1} \sum_{i=1}^{F} S_i \\ &= \frac{1}{\varepsilon_1} \sum_{i=1}^{F} \frac{\lambda_i L \beta - B_i}{\alpha} \\ &= \frac{1}{\varepsilon} \frac{\sum_{i=1}^{F} \lambda_i L \beta - \sum_{i=1}^{F} \frac{\lambda_i}{\lambda} B}{\alpha} \\ &= \frac{1}{\varepsilon_1} * \frac{\lambda L \beta - B}{\alpha} \end{aligned} \tag{10}$$

With the formula in (9) and (10), we can thus simulate the SGP time of the PSP streaming and calculate the total peers in the cluster whose service are guaranteed. Fig.2 shows the bandwidth increase with the contribution of both server and P2P cluster at different time stage. In our PSP streaming, QoS is guaranteed with an increasing bandwidth, first only by server, then server-peer co-existence and finally most by peers.

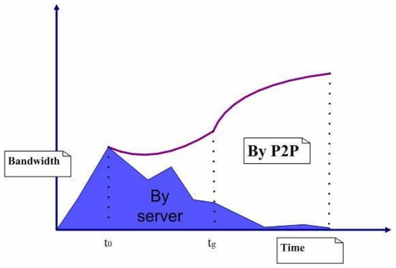

Figure 2. SGP time interval and bandwidth increase

B. *Bandwidth Bottleneck & Service Reject Rate*

Inside the performance of Service Guaranteed Period (SGP) in PSP streaming, it is the QoS parameter bandwidth that affects the mapping QoE. To benefit from the centralized manageability while offloading the traffic from the bottleneck bandwidth, our PSP streaming is controlled with management servers, which keep track of the peer topology and manage how the multimedia data should be sent.

Then one important problem arises: does the PSP structure have some bandwidth bottleneck and would that affect the service perceived by end users?

To address this question, we analyze the PSP streaming at the cluster side with one server to stream P2P peers. In the implementation, a layered coding is adopted for data transmission. We denote L as the number of layers and b_i as the bitrate for each stream where $i \in \{0,...,L-1\}$. Only the most important base layer b_0 is directly sent to every peer in the cluster from server in the initial stage. N is defined as the total number of peers in one P2P cluster. Thus the bottleneck bandwidth required for PSP streaming framework B_O can be calculated as:

$$B_O = \sum_{i=1}^{L} b_i + (N-1)b_1 \tag{11}$$

which is much less than the bottleneck bandwidth requirement for client/server streaming framework, whose bottleneck is $N\sum_{i=1}^{L} b_i$. In addition, the higher number the P2P cluster size, the more traffic may be offloaded from the bottleneck link, thus providing better video quality.

With the guarantee of bandwidth, we can expect a rather low service reject rate in our PSP streaming (in our survey, we find over 90% of end users are extremely concerned with the service reject rate of streaming service).

We define $R(t)$ as the service reject rate at time t.

Thus $R(t) = \dfrac{rej(t-\Delta t, t+\Delta t)}{req(t-\Delta t, t+\Delta t)}$

Where $rej(t)$ denotes the number of rejected requests while $req(t)$ denotes the number of total requests during time interval t. From the conclusion in section 3.1, we could calculate $R(t)$ as

$$R(t) = \begin{cases} 0\%, t \in (0, t_0) \\ 0-100\%, t \in (t_0, t_g) \cup (t_\infty, ...) \\ 0\%, t \in (t_g, t_\infty) \end{cases} \tag{12}$$

By adopting our PSP streaming scheme, the reject rate should follow the trends shown in the illustrative figure 3. During time interval when system bandwidth cannot meet all requests, streaming sessions start at the very beginning of that period and the subsequent requests will be rejected after the bandwidth is utilized. As the P2P network develops, the system capacity grows so that time with low reject rate would be longer. When it reaches, the service should be guaranteed and the reject rate would be then rather low (merely 0%).

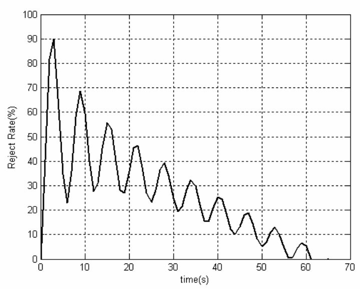

Figure 3. Service Reject Rate

IV. THE INDUSTRIAL PERSPECTIVE FOR PSP STREAMING

In industrial world, two factors weigh heavily over the popularity of live streaming. One is the general atmosphere of “Demand Economy” which has tremendous needs for media streaming. A recent survey conducted by StreamMedia.com and the Aberdeen Group indicates that over 74.1% of business and personal users access streaming media at least 2~3 times per week. But on the other hand, the nature of “Best Effort” Internet seems to put sand in the wheel of large-scale application of live media streaming. Actually, live media streaming has always been regarded as one of the potential killers over the Internet for years; it remains as a suspectable issue whether the best-effort Internet can ultimately deliver a high quality of live streaming service that satisfies the ever-increasing number of end users [1].

The positive side is we have undergone no shortage of technical innovations from many different perspectives to meet the drive of “Demand Economy” for live streaming service. In industrial world, there came the particular content distribution network (CDN) providers like Akamai, who strategically place a large number of video servers around the Internet to enable end users to obtain streaming video nearby, thus reducing the end-to-end delay and overall network congestion [9].

But the deployment of CDN servers is expensive and the streaming services are not guaranteed in face of a large flash crowd due to limited number of CDN servers. We then observed the popularity of P2P streaming company PPlive, who support over 100,000 daily online viewers synchronously. However without efficient management and supporting servers, pure P2P streaming companies have problems for copyright protection for videos and authenticating schemes to charge customers (thus making this company profitable). Meanwhile, in case of certain unpopular content with limited number of users, the quality of service would again become a problem.

To address above issues of pure P2P streaming companies, there comes the success of streaming companys like Rawflow with adoption of both P2P and CDN, which demonstrates the effectiveness of a Peer-Server-Peer media streaming in their real industrial application. Rawflow claims for higher bitrate streams, such as a 500Kbit/s video stream, and a bandwidth savings of up to 60% [7]. Recently the CDN service provider, ChinaCache[8], deploys the RawFlow technology in its media distribution network and serve customers with P2P live broadcasting in China. They together well handled audience peaks and huge demand in the heavy traffic time of 2006 Chinese New Year's Eve [7].

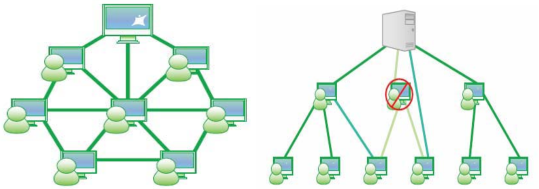

Figure 4. The Rawflow ICD Network topology[7]

Inside the success of Rawflow is its technique of combining streaming server and P2P clients. Rawflow’s Intelligent Content Delivery (ICD) technology enables streaming content providers to instantly scale their webcasts to serve large audiences at dramatically reduced bandwidth and infrastructure costs. Figure 4 shows a network formed by ICD to distribute a stream [7]. Only a fraction of users are directly connected to the stream’s source server. The majority of users are receiving portions of the stream from other users with latent upload resources. The more users in the network, the more resources become available, so there is no incremental cost as the audience scales.

V. CONCLUSIONS

We explore our paper with the focus on what is inside in the organization of the PSP streaming concept, why this architecture would work efficiently and whether it is practical to implement in industry. Thus instead of conducting meaningless simulations for this large-scale practical application, we prove it from theoretical level and explore the application from real industrial perspective, which should be of more value to the future real implementation.

Though similar ideas have been presented in the CDN-P2P hybrid media streaming [6], we are the first to present a more general PSP concept and practical architecture for implementation, with analysis from both academic model and industrial perspective. It is expected our exploration would help demonstrate the effectiveness of the new structure of PSP networking for streaming and attracting more application.

The main innovations of this paper include: 1) a novel Trustworthy and Controllable Peer-Server-Peer (PSP) structure that possesses the unprecedented capability to provide a practical solution for large-scale Internet media streaming. 2) An analytical model to evaluate the QoE and QoS for live media streaming and demonstrate the effectiveness of PSP architecture. 3) An industrial perspective to explore the feasibility of PSP in the historical context of the tussle between “Demand Economy” model and “Best Effort Internet”.

REFERENCES

[1] J.-C. Liu, S. Rao, B. Li and H. Zhang, “Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast,” Proceedings of the IEEE, Special Issue on Recent Advances in Distributed Multimedia Communications, 2007.

[2] X.Zhang, J.Liu, B.Li, and T.-SP Yum, “CoolStreaming/ DoNet: A data-Driven Overlay network for peer-to-peer live Media Streaming,” in Proc. IEEE INFOCOM’05, Miami, FL, March,2005.

[3] Yang-hua Chu, Sanjay Rao and Hui Zhang, "A Case For End System Multicast", Proceedings of ACM Sigmetrics, Santa Clara,CA, June 2000, pp. 1–12.

[4] H. Yin, C. Lin, F. Qiu, X. Liu, and D. Wu, “TrustStream: A Novel Secure and Scalable Media Streaming Architecture”, ACM Multimedia, Singapore, Nov. 6--12, 2005.

[5] H. Yin, C. Lin, F. Qiu and B. Li, “CASM: A Content-Aware Protocol for Secure Video Multicast,” IEEE Transactions on Multimedia, 8(2): 270-277, April 2006.

[6] Y. Tu, J. Sun, M. Hefeeda, and S. Prabhakar, “An Analytical Study of Peer-to-Peer Media Streaming Systems”, ACM Transactions on Multimedia Computing, Communications, and Applications, Vol: 1(4): 354-376. November 2005.

[7] [Online]. Available:www.rawflow.com

[8] [Online]. Available: http://en.chinacache.com/

[9] A.-J. Su, D. Choffnes, A. Kuzmanovic, and F. Bustamante, “Drafting Behind Akamai (Travelocity-Based. Detouring),” in Proceedings of ACM SIGCOMM’06, Pisa, Italy, September 2006.