SOLUTIONS FOR ENABLING COMMUNICATION BETWEEN DIFFERENT PROPRIETARY PEER-TO-PEER SYSTEMS

Renjie Pi, Junjie Tong and Ke Xu
Beijing University of Posts and Telecommunications, Beijing, P.R. China

Keywords: Peer-to-peer systems, Service-oriented architecture, Network storage.

Abstract: The Streaming traffic is one of the fastest growing traffic on the internet. Although the streaming traffic is growing fast, the peer-to-peer systems have solved this problem as its scalability and its fault tolerance against failures of centralized infrastructures. But because of the enormous growth in the number of peer-to-peer applications in recent years, they have occupied the most bandwidth of the internet, and it's essential to integrate them into the global content delivery infrastructure. In this paper, we propose two solutions for communicating between different proprietary peer-to-peer systems. One solution is to add a logical layer for enabling the communications between the existing peer-to-peer systems, while the other one is a service-oriented architecture by using the integrated network storage.

SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

The effectiveness of using the P2P technology for content distribution has been proven by many deployed systems (Emule; PPlive; PPStream; CNTV; Coben, 2003). These P2P systems offer three major types service for the end users. One type of service is P2P file downloading, for example implemented by Emule (Emule) and BitTorrent (Coben, 2003). When a peer is downloading a file, it can get certain part of the file from the other peers which are downloading the same file from the content provider server. And the server load is significantly reduced. The peers may experience different downloading rates, often depending on how much they are able to contribute to the whole overlay network and the corresponding mechanism in the P2P system. The second type is P2P live streaming, for example implemented by PPlive (PPlive) and CBox (CNTV). In a live streaming session, a live video content is disseminated to all users in realtime and the video playbacks on all users are synchronized. There many different overlay structures for implementing this service including Tree-based structure and Mesh-based structure. Tree-based structure includes the single-tree streaming and multi-tree streaming (Liu et al., 2008). The third type is P2P video-on-demand (P2P-VoD). In this service, contents are also delivered by streaming, but peers can watch different parts of a video at the same time, hence diluting their ability to help each other and offload the server. This service needs the peers contribute a small amount storage to offer the data for other peers with a sophisticated distributed scheduling mechanism for directing peers to help each other in real time (Huang et al., 2008).

The P2P technology can be utilized in designing highly scalable and robust applications at low cost, compared with traditional client-server paradigms. But as the number of the P2P systems increases dramatically, some problems are coming up. The first particular problem is the substantial stress that the P2P systems place on the network infrastructure and the users. As the increasing traffic on the internet place too much load on the overlay, the other users on the internet have to suffer the increasing delay time and low performance. And the users have to install the different software to watch the content. The second problem is that more participants beyond traditional P2P streaming vendors are joining the efforts in the development of P2P streaming systems. Some of these additional participants include infrastructure vendors as Akamai, ChinaCache, and ISPs. That is, the P2P streaming ecosystem is becoming an increasing diverse industry with participants from the source, infrastructure, delivery and local P2P distribution to the terminals (Zhang et al.). The third problem is the

Pi R., Tong J. and Xu K..
SOLUTIONS FOR ENABLING COMMUNICATION BETWEEN DIFFERENT PROPRIETARY PEER-TO-PEER SYSTEMS.
DOI: 10.5220/0003590905800585
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (SSE-2011), pages 580-585
ISBN: 978-989-8425-53-9
Copyright © 2011 SCITEPRESS (Science and Technology Publications, Lda.)

proprietary P2P streaming systems using different streaming protocols and it's hard for a peer communicating with the other peers located in the other P2P overlay to save the bandwidth, the storage resource and computing resource.

To solve the above problems, in this paper, we propose two solutions. The first solution is oriented for enabling the peers in different proprietary P2P systems to communicate with each other aiming for accessing the data. By using this solution, it adds a logical layer in the existing P2P systems. And the other solution proposes a service-oriented architecture for developing the P2P systems avoiding the above problems. And this architecture has using the network storage for storing the contents which brings the architecture some new features.

In the following sections, we will first introduce the two solutions and then we will outlook the future solution for the P2P systems which interacts with cloud storage and cloud computing.

SCIENCE AND TECHNOLOGY PUBLICATIONS

2 THE ADDITIONAL LAYER

In this section, we descript the additional layer to the existing P2P systems mentioned in the first solution. We mainly define its location in the P2P systems, its inner construction and its functions. But before that, we first have to define the preconditions of adding this layer.

2.1 Preconditions

Before implementing the additional layer, there have to be some preconditions to be described first. The entities including in a specific P2P system will be defined as below.

The peer: A peer is an end user on which processing the P2P system and not only receives contents and streaming, but also stores and uploads contents and streaming to other participant peers.

The chunk: An integrating content is divided into several chunks by last time or the certain size. A chunk is considered as the basic unit of the partitioned content or streaming. Peers may use a chunk as a unit of storage, advertisement and exchange among peers. Note that a P2P system may use different units for advertisement and data exchange, maybe a chunk or several chunks.

The tracker: A tracker is a server on which processing the software to provide a directory service which maintains a list of peers storing chunks for a specific content and answer the queries from peers for the peer lists.

Above, we describe the three main entities of a specific P2P system. Although there are enough hardware components, we should have corresponding protocols and mechanisms to enable the communications between them. The main protocols of a specific P2P system including the below ones.

Tracker protocol: For enabling the communication between the tracker and the peer, we define this protocol. And this protocol defines the standard format/encoding of information between peers and the tracker, such as peer list, content availability, peers' status and streaming status including online time, link status, peers' capability and some other parameters. And it also defines the standard messages between the peers and the tracker. By using these messages, peers report streaming status and request to the tracker, and the tracker reply to the requests.

The peer protocol: The peers can use this protocol for requesting more potential peers from the other peers. In this protocol, we define the standard format/encoding of information among peers, such as chunk description. And we also define the standard messages among peers defining how peers advertise chunk availability and their neighbor peers' information to each other, as well as the signaling for requesting chunks among peers (Zhang et al.).

Above, we define and describe the main entities and protocols in a specific P2P system. Before enabling the communication between the different proprietary P2P systems, the preconditions include the following ones.

(1)Including the same main entities. There are peers and track in the system. And the contents and the streaming are partitioned by chunks.

(2)Using the same format/encoding of information in the track protocol and the peer protocol. So the messages can be transmitted between different P2P systems and be translated correctly.

By the above conditions, we can show the architecture of a specific P2P system in the following figure.

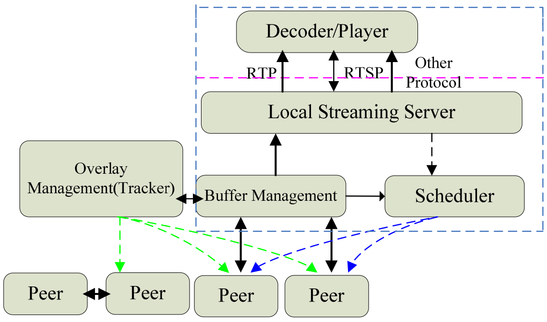

Figure 1: The common architecture of the P2P system.

2.2 Overview of the Additional Layer

In this section, we will describe the location of the additional layer and its corresponding construction and functions, as well as its features.

2.2.1 The Location and Inner Construction

We can see the common architecture of the P2P system, the additional layer will be named P2P-Bridge. And we see its location in figure 2 by modifying the figure 1.

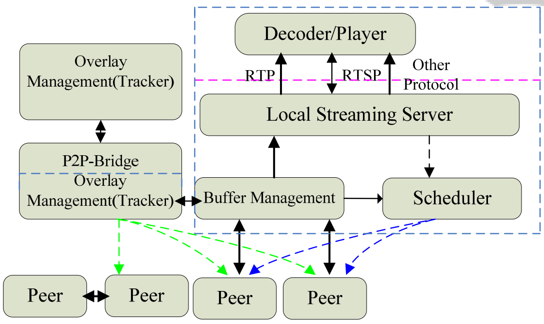

Figure 2: The Location of the P2P-Bridge Layer.

The figure 2 shows the P2P-Bridge's location clearly, and it is located above the overlay management layer implemented by the tracker. Through the P2P-Bridge, the two trackers in different proprietary P2P systems can communicate with each other to exchange the information.

In the P2P-Bridge, it has its own inner construction and we can divide the layer into several logical parts.

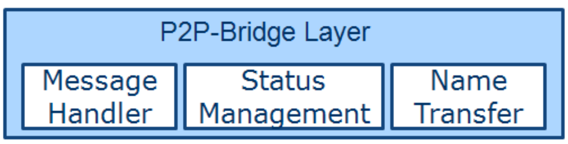

Figure 3: The Construction of the Layer.

Figure 3 shows the components of the layer. It contains three components: the message handler, status management and the name transfer.

The message handler is to handle the message between the trackers. While the message arrives, it analyses it first. And then modifies the message to a unified message with certain format and encoding way.

The status management will keep the status of the other trackers which can communicate with the local tracker. And it also maintains their status including the link status, storage and the resources status of the other P2P overlay.

The name transfer maintains a name table for looking the corresponding name of the content or streaming as the same one may have a different name in another P2P system. And it also maintains some mechanisms for converting the users' requests.

Although we introduce the components briefly, we will describe it fully in the following part.

2.2.2 The Function and the Processing

The main goal of the P2P-Bridge layer is to enable the communication between the different proprietary P2P systems. We can achieve this goal by several steps.

First, we have to handle the messages from different P2P systems. As we have suggested that the format/encoding are the same, so we can understand the messages easily. Then we have to modify the message into a suitable format which used in the P2P system which we want to communicate with. At the same time, we extract the name of the requesting content or streaming from the message. Then we will look into the name tables which are maintained by the name transfer. Then we add the corresponding name into the modified message. All over the processing, we have to maintain some information to choose the most suitable P2P overlay, to ensure the other tracker is not uploaded and refresh the name tables on time.

The message handler is for receiving the message, and modifying the message briefly into the format which is suitable in another P2P system. At the same time, it extracts the name of the content or streaming and passes it to the name transfer.

The name transfer receives the name from the message handler. Then it looks into the name table which it maintains for searching the name in another P2P system. Then the name transfer returns the corresponding name to the message handler, if it successfully finds it.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

The message handler modifies the message again after receiving the name returned from the name transfer.

After doing the above things, the message is modified into the suitable format which the tracker in another P2P system's tracker can understand and handle it.

The status management is for maintains the connection between the trackers and it can also be used to compute the mount of the traffic between the different P2P systems for charging.

2.2.3 Features

By adding this layer in the existing P2P systems, we can easily enable the communication between the different proprietary P2P systems. And there have several features by deploying the layer.

Convenient. We don't have to change the whole architecture or the inner construction of the system.

Manageable. By adding some mechanisms in the status management, we can manage the connections suitable for the whole P2P overlay.

Scalable. We can add the complex component into the P2P-Bridge easily. And it will be compatible for the whole logical layer and the P2P system.

2.2.4 Other Issues

Although the solution deals with the problem well, there are some open issues existing.

The naming space. In the layer, we have to maintain the name tables for searching. Maybe we can unify the naming space first, it will be more flexible. And in (Hu et al., 2009), it proposes an approach to accurately identify different P2P applications from the network traffic for managing a number of network traffic issues.

The chunk information. Maybe the request contains the requested chunks. And as the different strategy of dividing the content, the chunk is defined differently in different proprietary P2P systems. So we have to modify the chunk information or define the unify chunk information first.

The above issues are the two main ones, and maybe other issues existed.

3 THE SERVICE-ORIENTED ARCHITECTURE

For handling the problem of how to enable the communication between different proprietary P2P systems, there some ways have been proposed on the high level.

In (Dabek et al.), it has proposed a common API for structured peer-to-peer overlays and the key abstractions that can be built on them. The authors propose the KBR API for enabling the developers implement it as an RPC program with flexibility and effectiveness. And in (Delmastro et al., 2006), the author has proposed an extension of the common API as a generalization of fixed and mobile structured P2P systems, exploiting the cross-layer approach to export network routing table information and the current state of the overlay even at the application layer.

A Service Oriented Infrastructure consists of four categories of components: Service Planning, Service Systems, Service Management, and Service Stakeholders. The service planning should be driven by business domain requirements. The service systems will be designed and implemented based on above service plan. The service management component includes both system operation management and IT service management in functional level. And the service stakeholders should be identified for roles and responsibilities, and should be aligned with business operational structures (Zao, 2008). We can see its construction clearly from the figure 4.

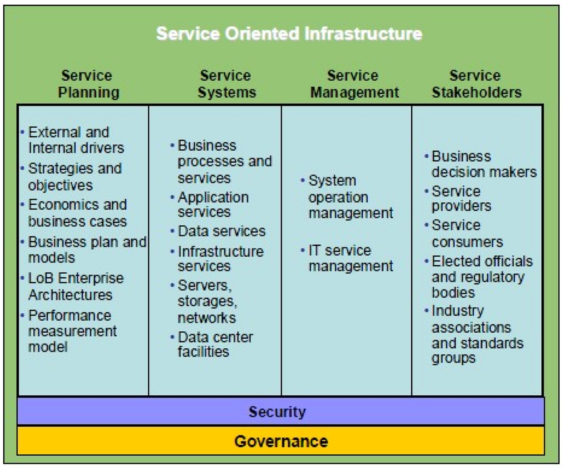

Figure 4: The Construction of the SOI.

As the storage systems are becoming the domain investment in data centers and a crucial assert, making the rate of growth of storage a strategic business problem and a major business opportunity for storage vendors, it's essential for us to concern the storage part in developing the P2P system (Gibson and Meter, 2000).

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

In this paper, we propose the second solution for enabling the communication between different proprietary P2P systems basing on a service-oriented architecture. In the following sections, we will describe the construction of the architecture first, and then its features different from the first solution. And the proposed architecture uses the integrated network storage for providing the common accessing API for the storage or architecture users and developers.

3.1 Overview of the Architecture

In figure 5, it shows the overview for implementing the architecture. We have divided the whole system into four main components.

The content providers are the source server for providing the contents and streaming for the clients. The component below the content providers is the storage network conforming by the servers providing the storage service for the contents and streaming and the accessing service for the proprietary P2P system below. The component below the storage network is the P2P system. There can be different proprietary P2P systems. And the clients who install the proprietary P2P system is the last component in the whole system.

The architecture we proposed is to define unified interface to provide storage service for the content providers, and also define the unified interface to provide accessing service for the different proprietary P2P systems.

3.2 The Construction of the Architecture

In this part, we will describe the logical components constructing the architecture. And introduce the function of the main components.

The architecture provides two main types service. One is the storage service while the other one is the accessing service. First, let's introduce the function of the parts in the two service layers.

In the storage service layer, there are three logical parts. The first one is the naming which assigns a unique name for the content or streaming received from the servers and extract common properties to construct the unique identification for every content and streaming. The second one is the status management part which maintain the information of the source servers especially the load of the servers. The third one is the content publish part which is responsible for choosing proper

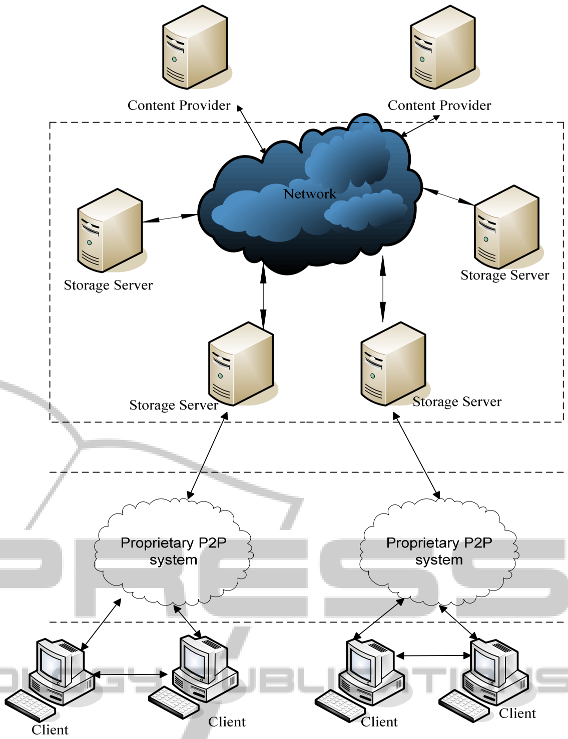

Figure 5: Overview of the Whole Implementation.

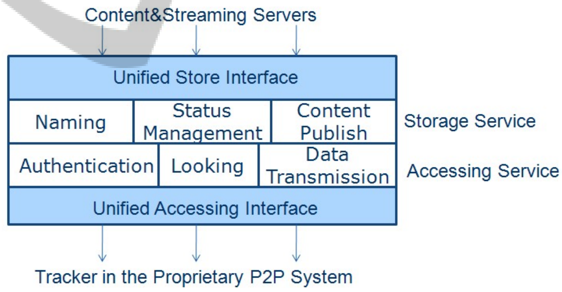

Figure 6: The Construction of the Architecture.

storage server to publish the content or streaming correctly.

After using the storage service, the contents and streaming will be stored in the storage network.

Then the content will be published to the different P2P systems by sending messages to the tracker, maybe its name will be changed at the end users software.

Then the clients can get the data by using the unified accessing interface. And between the clients, they can also help each other by the proprietary protocol.

In the accessing service, there are also three parts. One is the authentication which is for registering and recognizing the tracker server in the proprietary P2P system. The second part is the looking or searching which is for looking the clients' requesting source in the storage network. And the data transmission part is responsible for the data transmission between the storage servers and the clients.

There also important problem exists, and that is how to construct the storage network. It has relationship of publishing the resources, looking for the resources, managing the resource on the storage network. And maybe we can implement the P2P technology in the storage network for the performance and scalability of the overlay.

Although there is no data exchange between the different proprietary P2P systems, but they get the content and resources from the same overlay and this architecture integrates the resources and the P2P systems. Considering the P2P system as a part, it can be seen the data transmission between different proprietary P2P systems and we also add the message function in the architecture for the communications between the trackers.

4 COMPARISON

We have proposed two solutions for enabling the communication between different proprietary P2P systems emphasizing different aims. The first one is to revolute the existing systems easily. And the other one is for the integrating the P2P resources and interviewing.

The second difference is the cost. The first one is low while the other one is high. As the second one needs more participants.

The third difference is the design model. The first one is modifying the existing ones, while the other one oriented at the level of the architecture with some new features.

5 CONCLUSIONS

Although we have proposed two solutions for reduce the wasting of the content and streaming storage resources in the P2P systems and enabling the developing process of the P2P system easier. We lack the thinking about the mobile environment to implement the P2P software. And in the future, there will be more flexible and scalable solutions for the P2P content and streaming distribution even the data distribution in uniformed format.

ACKNOWLEDGEMENTS

This work is supported by the National Key project of Scientific and Technical Supporting Programs of China (Grant Nos.2008BAH24B04, 2008BAH21B03; the National Natural Science Foundation of China (Grant No.61072060)); the program of the Co-Construction with Beijing Municipal Commission of Education of China.

REFERENCES

Emule, http://www.emule.com/

PPlive, http://www.pplive.com/

PPStream, http://www.ppstream.com/

CNTV, http://www.cntv.cn/index.shtml

B. Coben. "Incentives Build Robustness in Bittorrent". http://bitconjure.org/BitTorrent/ bitttorrentecon.pdf, May 2003.

Yang Liu, Yang Guo, Chao Liang. "A Survey on peer-to-peer video streaming systems". December 6, 2007. Springer Science&Bussiness Media, LLC 2008.

Yan Huang, Tom Z. J. Fu, Dah-Ming Chiu. "Challenges, Design and Analysis of a Large-scale P2P-VoD System". SIGCOMM'08, August 17-22, 2008.

Zhang Y., Zong N., Camarillo G., Seng J., and Y. Yang. "Problem Statement of P2P Steaming Protocol". http://datatracker.ietf.org/doc/draft-ietf-ppsp-problem-statement/?include_text=1.

Chen-Chi Wu, Kuan-Ta Chen, Yu-Chun Chang, Chin-Laung Lei. "Peer-to-Peer Application Recoginition Based on Signaling Activity". 2009. ICC '09. IEEE International Conference on Communications, June 2009.

(Dabek et al.) Frank Dabek, Ben Zhao, Peter Druschel, John Kubiatowicz. "Towards a Common API for structured Peer-to-Peer Overlays". http://project-iris.net/irisbib/papers/iptps:apis/.

Delmastro F., Conti M., Gregori E.. "P2P common API for structured overlay networks: a cross-layer extetnsion". July, 2006.

Yan Zhao. "Service Oriented Infrastructure Framework". 2008.IEEE Congress on Services, July 2008.

Garth A. Gibson, Rodney Van Meter. "Network Attached Storage Architecure". Communications of the ACM, November 2000. Vol 43. No 11.