

COT 6930 – Digital Video Communication – Spring 2004

Term Project Report

Peer-To-Peer network with QoS peer selection

Esfandiar Zolghadr

Ezmail01@yahoo.com

Abstract

Many believe Peer-to-peer (P2P) emerges as a computing paradigm of the future. There is an ever-increasing need for distributed algorithm that would allow P2P applications to scale to a large network of individual servers. Applications such as video streaming or media on-demand have incredibly large bandwidth requirements and bandwidth scalability of P2P can provide the P2P users with unlimited resources. While bandwidth requirements may be fulfilled, the latency caused by poor network quality can cause serious threat to the quality of the user experience particularly in real-time streaming and interactive applications. QoS in P2P networks guarantees better communication quality by selecting best serving peer candidate.

Keywords: *Peer-to-peer, QoS, video, Streaming, P2P*

1. Objective

With increasing interest in P2P computing and video streaming in particular, need for a standard, interoperable, platform independent, and O.S. independent network technology is more than ever. Majority of existing P2P alternatives target solutions for file sharing and messaging applications with video streaming getting very little attention. In media streaming and media on demand user experience affected tremendously by available bandwidth, network quality and delivery methods of streaming video. Deployment of efficient and effective QoS in the network can improve quality of streaming because in decentralized P2P networks, selection of serving peer does not necessarily result in selection of best candidate. We believe using effective QoS

can increase network-level resources as well as presentation-level quality including frame rate and resilience without overloading the network with unnecessary interactions. This study investigates characteristic of P2P networking and proposes a peer selection method based on QoS. We present our implementation and the result obtained from streaming before and after deployment of QoS features.

2. Background

In a P2P network, peers offer services such as file sharing, distributed computing and instant messaging. Among all applications, video streaming has most bandwidth requirements. One of the most distinguished characteristics of P2P networks is scalability and that is because every peer adds considerable amount resources to network capacity upon joining to the network. Although this is not desirable for ISPs, it is perfect for streaming network with virtually no investment on wide pipes and network backbone.

P2P networks

P2P network comes in two flavors of *centralized* and *decentralized*. Centralized network is supervised using one or more servers that enforce policies, manage resources and quality of service. Centralized scheme has limitation particularly in streaming applications where server is flooded with traffic information transmitted to server at each heartbeat. Network performance degrades when latency of server increase. Decentralized P2P is an ad-hoc network of peers and the only supervision is within the peer groups with a peer acting as supervisor. Decentralized P2P is on the edge of the Internet with no DSN or directory authority. Peer discovery is very important part of successful streaming

network. Discovery process is used identify set of the peers that possess the content with sufficient bandwidth, CPU power and authority to stream. This cannot be done without information from sufficient information from potential candidates and QoS reports.

This project is a QoS proposal for Sun's open source P2P solution. JXTA is an acronym for the word "Juxtapose" which refers to side-by-side to client-server computing. JXTA is java implementation of P2P networking with basic capabilities such as peer and group service advertising, discovery and query services. Some basic applications are built in such as short messaging and query services.

3. Description of the Solution

Video streaming demands a serving peer meet minimum bandwidth requirement that is essential to quality of user experience. P2P network facilitates discovery of peers that posses the content however in absence of QoS capability selection of peers will not guarantee quality service or service availability for during the streaming session.

We propose a QoS based ranking of peer that improves network-level and presentation-level quality of steaming in JXTA network. Policy driven QoS discussed here provides resource management and based on that, peer selection techniques to improve and maintain network performance.

JXTA P2P network implementation

JXTA network supports five type of peers differentiated with their role in the network. Minimal edge peers represent smart phones and PDAs with minimal responsibility in the network. Edge peers are regular nodes that serve or served in a network. Rendezvous

peers are peers with more responsibility that edge peers and act as a leader in a group of peers with mutual interest. Rendezvous peers cache a directory that has index to information such as peer IDs, advertised services, modules, module classes and pipes of the same group as well as neighboring groups.

Services are functions and tasks that a peer or the group of peers, offer to rest of the world. There is a thin line between a service and an application, conceptual structure of the JXTA software architecture is shown in Figure-1.

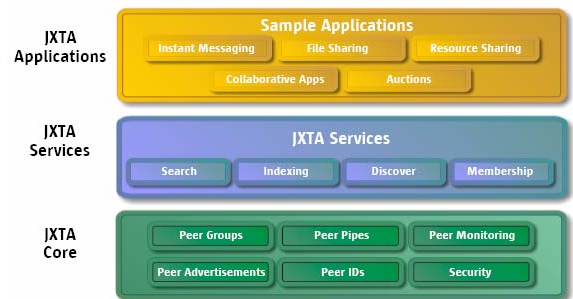


Figure 1. JXTA software architecture

Implementation of QoS on JXTA paradigm follows a hierarchical approach such that QoS enabled peers in a group report their state and status to rendezvous peers. Peers update their status when changes in their available resources occur or they shut down properly. To avoid surprises, when peers are being considered for streaming, they are contacted and their status is refreshed. Rendezvous can maintain QoS database for their peers without flooding the network with messages. QoS records are advertisements generated by QoS client service running at each peer. Using advertisement makes management, search and updates of QoS possible through existing system protocols. One of the most important benefits of using available protocols is that QoS information can be

globally accessible through out the network. Also, storage efficiency is maintained by using existing infrastructure instead of creating a new one.

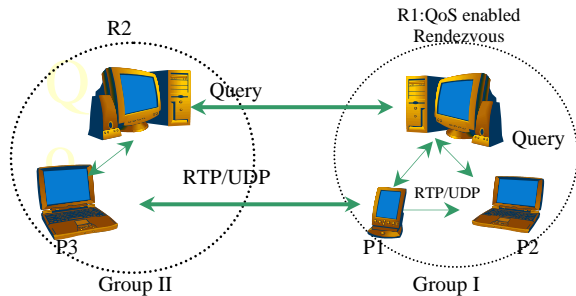


Figure 2. QoS Mechanics

QoS is a JXTA service that runs on all participating peers and their rendezvous peers. Rendezvous must collect QoS services from peers and cache the information until used during the discovery and peer selection. QoS collects and stores information such as bandwidth, traffic and performance factor. When a discovery request for the content on the peer is received by its rendezvous, the ranked QoS data along with peer id (PID) and is sent to the requesting node. Requesting node will collect replies returned from all peers in range of maximum neighborhood count and process the qualifications of each candidate peer select the best peer in the list.

Figure-2 shows a discovery request initiated from P1 and sent to its group rendezvous, R1. Resources within the group-I are already cached in R1 so R1 contacts all the peers to refresh their status. If still available, P2 is entered into ranking list and meanwhile, R2 the rendezvous of Group-II is contacted. When R2 finds the resource on P3, contacts with it and if still up and available, reports P3 to R1. Once all the responses are collected, R1 complete final ranking list and reports it to P1 to initiate its direct point-to-point communication with the best serving peer.

Peers communication

Once the best peer is selected the serving peers' PID is given to requesting peer to establish a direct, point-to-point connection and begin the streaming session. The communication line between the two is termed Pipe and that in simple form is an abstract entity with two end points. The end-points at server side is an output port and the serving peer must have input end-point. This determines type of pipes from serving peer or server peer viewpoint. Figure-3 illustrates direct pipe between peer-A and peer-B.

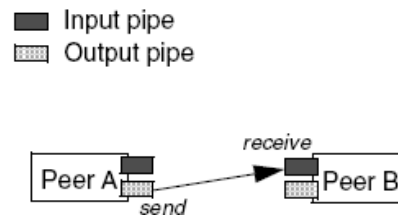


Figure 3. Point to point pipe

There are other cases where selected peer has enough resources to serve more than one peer or only one peer contains the requested information. In this case a configuration using a pipe called propagate pipe is used.

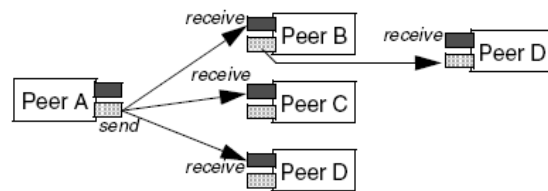


Figure 4. Propagate pipe

As Figure-4 shows, serving peer-A send the streaming data to multiple endpoints or destinations using a propagate pipe.

Pipe is an abstract entity but in JXTA every pipe has ID and advertisement of its own. It can be searched and discovered and queried.

Peer QoS reports

All the QoS enabled peers must report their relevant information to their rendezvous when they join the peer group and keep their information up to date all the time. To minimize the network interactions required, messaging kept at minimum only for important changes.

Messages communicated with rendezvous are XML documents with following fields mandatory to QoS operations:

- PID
- Bandwidth (Made Available to P2P)
- Memory
- Disk space (Available to P2P)
- Processor performance index

JXTA allows all the messages and advertisements be altered and customized but QoS requires a minimum set of information be included.

Ranking criteria

Transient nature of P2P participation requires careful ranking and selection of serving node as well as keeping the list updates during a streaming session. Since peer discovery in JXTA is asynchronous, the ranking list is kept until end of session in case the selected server, fail or disconnect from network. The selection is made based on minimum streaming requirements and other available resources. Most important requirement is availability of bandwidth that is essential to presentation-level quality that is seamless streaming with acceptable frame-rate and bit-rate.

If streaming content is available on a set of n peers, $\{p_1, p_2, \dots, p_n\}$ and consequently the available bandwidth on each of these servers is $\{b_1, b_2, \dots, b_n\}$, then serving peer will be ,

a peer such as P_s , that satisfies the condition, $P_s \in \{p_n\}$ and $\min\{b_n\} > b_{\min}$, where b_{\min} is minimum bandwidth required and $\{b_n\}$ bandwidth corresponding to serving candidate peers. Our first serving peer criteria is $P_s = \max\text{Bandwidth}\{p_n\}$, which satisfies our main requirement. Now, if $\min\{b_n\} > b_{\min}$ is not true and even worst $\max\{b_n\} < b_{\min}$ then the presentation level QoS compensate for lack of bandwidth by setting up streaming session with frame-rate or bit-rate such that it does not exceed the maximum bandwidth.

Bandwidth required

Required bandwidth is calculated using following equation:

$$\text{Required Bandwidth} = \frac{\text{FrameSize} * \text{PixelDepth} * \text{FramePerSecond}}{\text{Compression Rate} * \text{BitsPerByte}}$$

This is ideal bandwidth and if cannot achieve, then QoS reduces the criteria by quality trade-off.

Recovery from failure

If a current serving peer is detected as suffering from sudden failure or bandwidth degradation, QoS service will migrate all of its' streaming load to another serving peer from the ranking list. Therefore, the set of candidate serving peers in one streaming session will change dynamically during the session.

Buffer control

To accommodate the dynamic set of serving peer and the end-to-end network congestion, QoS requires a suite of buffer control mechanisms, which involves more concurrency and scheduling complexity than the traditional buffer control mechanisms in client-server streaming.

Delay analysis

Receiving peer cache a moving window of the most recent content that they have received. Assume a client can buffer m minutes worth of video (we assume that playback rate is constant) and now playing video frames at the position t_c minutes from the beginning of the video at time t . It caches the most recent m minutes of the video, $[(t_c - m)^+, t_c]$ and continuously caches the most recent content as time goes along. When the serving node packets do not arrive at the scheduled time, a request is sent to all the candidates on the top of the list requesting streaming from point t_c . The peer with the highest bandwidth that replies first, takes over the streaming.

4. Relevant Results

The proposed QoS service was implemented and deployed in a group with different computer configurations. Processors used in this experiment were at least Pentium-III-640 MHz with 256 Mega byte of memory.

JXTA source was used to implement our QoS enabled networking. Two types of QoS services created one for rendezvous and one for peers.

A computer with Pentium-IV 3.02 GHz was selected as rendezvous and server software started and a group was created for our experiments. Peers running JXTA QoS enabled client software started and advertisements initiated and cached on the server. To verify the correctness of the peer selection algorithm peers participated with different bandwidth.

A video clip with 320x200 resolution and duration of ten minutes was replicated on all three peers and appropriate advertisement

was created and cached to server for the purpose of caching.

Now using another peer QoS discovery request is sent to the group rendezvous. QoS service queries rendezvous cache searching for QoS enabled peers that advertised for the video clip. Table-1 is the list of serving candidate peers returned to requesting peer.

Peer ID	Bandwidth	CPU	Memory
<PID-1>	540,000	600	256
<PID-2>	360,000	1,200	256
<PID-3>	200,000	600	512

Table 1. QoS ranking list

Our experiment shows that requesting peer selects the peer on the top of the list identified as <PID-1> with 540,000 kbps shared bandwidth. Since every peer carried only one session at a time, the stress on the serving computer is not significant so is the memory. The connection established and session started and streaming started playback after few minutes the connection from server side was interrupted and approximately two second later streaming started from <PID-2> with slight frame freeze. The experiments repeated and this time the stream session setup to have larger buffer as big as three second. Using bigger buffer prevented freeze frame flaw when first server disconnected.

5. Conclusions

QoS server selection guarantees the server with sufficient resources will be selected. Also, list of next best candidates are reserved for duration of streaming session in case current selected server fail or disconnect for any reasons. The result expected to be more promising in real world P2P network where participating peers share their small uplink bandwidth where for guaranteed service using more than one server at any given time is required. Overall

this work shows successful implementation of the P2P networking and their application in streaming. ■

video-on-demand systems,” in *Proc. IEEE International Conference on Multimedia Computing and Systems*, June 1997.

References

- [1] Y. Guo, K. Suh, J. Kurose, and D. Towsley, “A directory-based peer-to-peer on-demand streaming service,” Tech. Rep. UM-CS-Tech-Report, Department of Computer Science, University of Massachusetts Amherst, 2002.
- [2] Q. Ma and P. Steeniste, “On path selection for traffic with bandwidth guarantees,” in *Proc. International Conference on Network Protocols*, October 1997.
- [3] D. Eager, M. Vernon, and J. Zahorjan, “Minimizing bandwidth requirements for on-demand data delivery,” in *IEEE Transactions on Knowledge and Data Engineering*, Sep/Oct 2001.
- [4] E. Zegura, K. Calvert, and S. Bhattacharjee, “How to model an internet network,” in *Proc. IEEE INFOCOM*, April 1996.
- [5] H. Deshpande, M. Bawa, and H. Garcia-Molina, “Streaming live media over peers,” Tech. Rep. 2002-21, Stanford University, March 2002.
- [6] V. Padmanabhan, H. Wang, P. Chou, and K. Sripanidkulchai, “Distributing streaming media content using cooperative networking,” in *Proc. IEEE Workshop on NOSSDAV*, May 2002.
- [7] Y. Guo, K. Suh, J. Kurose, and D. Towsley, “P. cast: P2p patching scheme for vod service,” Tech. Rep. UM-CS-2002-034, Department of Computer Science, University of Massachusetts Amherst, 2002.
- [8] D. A. Tran, K. A. Hua, and T. T. Do, “Layered range multicast for video on demand,” in *Proc. International Conference on Computer Communications and Networks (IC3N’02)*, October 2002.
- [9] S. Jin and A. Bestavros, “Cache-and-relay streaming media delivery for asynchronous clients,” in *International Workshop on Networked Group Communication*, October 2002.
- [10] M. Kamath, K. Ramamritham, and D. Towsley, “Continuous media sharing in multimedia database systems,” in *Proc. of 4th International Conference on Database Systems for Advanced Applications (DASFAA’95)*, April 1995.
- [11] A. Dan and D. Sitaram, “A generalized interval caching policy for mixed interactive and long video environments,” in *SPIE Multimedia Computing and Networking Conference*, January 1996.
- [12] S. Sheu, K. A. Hua, and W. Tavanapong, “Chaining: A generalized batching technique for