ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

TRAFFIC CONTROL MATRICES IN VIDEO STREAMING IN A P2P NETWORK

S.Boopathi*, V. Vijayadeepa
* M.C.A Research Scholar in Computer Science,Muthayammal College of Arts and Science, Rasipuram,Tamilnadu,India.
M.Sc.,M.Phil., Head Academy of Computer Science, Muthayammal College of Arts and Science, Rasipuram,Tamilnadu,India.

ABSTRACT

This system provides the complete information and the complete implementation of the video streaming technology. The main aim of the project is to create a system for the effective video conference over the intranet within the Organization. Currently there are many systems available for videoconference. These systems use the TCP/IP protocols for communicate and also for transfer the frames from one system to another. These systems also follow the stored and forwarded procedure for transfer the frames. Normally these systems can transmit and re-transmit the copy of the frames from the source or server. In order to develop the new system, first we have to understand the history of the existing system. This history will give the complete information about the existing system and also, this history will be the bright introduction for new system.

KEYWORDS: TCP/IP, P2P, XAM, WLAN.

INTRODUCTION

Our system differs from traditional systems as it provides application mobility in a decentralized manner over heterogeneous networks, using different communication technologies. Following the evaluation of our system, we also discuss major challenges and possibilities for the continuing evolution of application mobility systems. In Yu's classification, the application can be migrated within a subnet, between a subnet and the Internet, or between heterogeneous third party controlled networks. The same classification can be done for devices (private or public devices, homogeneous or heterogeneous in nature). Architectural proposals on how to achieve application mobility do exist but actual deployments and evaluations within the field are very scarce. To prove the viability of a concept, it is our belief that it has to be simulated or, even better, prototyped and then examined. In this paper we present applied application mobility, manifested in a peer-to- peer based system called XAM. The system is designed to be decentralized and functional in heterogeneous environments (multiple networks with multiple service providers), thus differing from traditional systems for application mobility and contributing to state-of-the-art research for this mobility type.

This information tells the advantages and the disadvantages of the existing one. So the problems can be identified. This helps for the development for the new system.

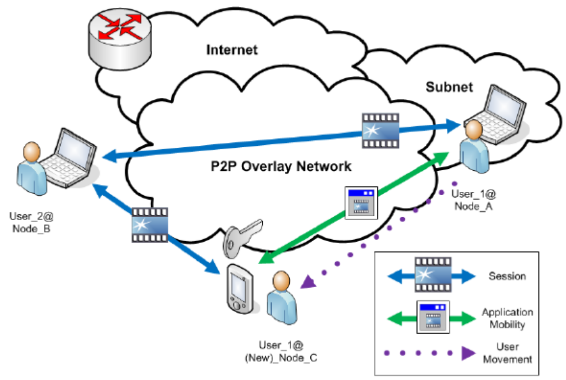

Figure. 1. P2P Overlay

System architecture is based on our initial proposal, extensively described in Figure 1. The architecture aims to fulfill requirements such as ability to handle application distribution and identification, context-awareness, context quality, provision of seamless migration, support for heterogeneous environments and high degree of usability. The architecture (see Figure 1) is based on the peer-to-peer paradigm. This design results in a decentralized system, where new devices should be easily addable while other devices could be removed in an ad hoc manner. The architecture also supports nodes to connect from outside the local sub network, allowing application mobility between heterogeneous networks. Context data is distributed and stored throughout the network, making it available to all peers. This also means that if a device hosting an application is removed from the network, it can automatically locate a peer eligible to become the new host for the application and migrate it accordingly. Heterogeneity support is desirable, as is mobility.

Application mobility should be extended to a multitude of devices and migration should be possible between different networks. The rationale behind the architecture is that the application should run on the device that suits the user best at any given situation. Therefore different kinds of devices in regards of input and output capabilities, level of mobility and other important resources, should be addable to the system. Our general system design differs from traditional systems as it provides application mobility in a decentralized manner over heterogeneous networks.

We call our implementation XAM, featuring peer-to-peer based application mobility with heterogeneous devices and networks. The system consists of four tiers: network infrastructure and devices, an overlay network, a migration manager and the migrate applications. This paper is mainly focus on the process of video streaming technology. The effective communication will be done by using the video streaming methodology. There are many systems using this video streaming technology. But they cannot provide effective communication with the systems. This may have the main demerit based on the protocol. The transmission is done through the TCP/IP protocols for communication and also for transferring the frames from one system to another. These systems also follow the stored and forwarded procedure for transfer the frames. Normally these systems can transmit and re-transmit the copy of the frames from the source or server. There are several modules used in this project to accompany the process of video streaming efficiently. At least one device in every subnet is given the role of rendezvous peer, making it possible for devices outside the network to connect to the peer group.

RELATED WORK

This paper describes the Stanford P2P Multicast (SPPM) streaming system that employs an overlay architecture specifically designed for low delay video applications. In order to provide interactivity to the user, this system has to keep the end-to-end delay as small as possible while guaranteeing a high video quality. A set of complimentary multicast trees is maintained to efficiently relay video traffic and a Congestion-Distortion Optimized (CoDiO) scheduler prioritizes more important video packets. Local retransmission is employed to mitigate packet loss. Real-time experiments performed on the Planet-Lab show the effectiveness of the system and the benefits of a content-aware scheduler in case of congestion or node failures.

In recent years, video streaming over the Internet has become more and more popular due to the increasing availability of bandwidth and recent advances in video coding technologies. In several application scenarios, such as IP-TV, the same video content has to be transmitted to a large population of users. Content Delivery Networks are often used to support large numbers of users but they require the deployment of special infrastructure. As an alternative, Peer-to-Peer overlay networks have been considered for delivery of multimedia. The idea is that users who are interested in a video stream can act as relay points and forward the data to other users, thus allowing the system to scale with the number of nodes involved in the communication. P2P systems are widely used for video file sharing. For live multicasting, however, the problem is much more challenging, as the overlay network must guarantee high reliability of the connections and a constant flow of data, as well as low startup latencies. Several different architectures have been proposed in the literature that can be grouped in two different categories. Mesh-based aim to construct an overlay mesh whose connections are maintained through "gossiping". Even though these solutions provide good error resilience and network utilization performance, they are usually characterized by high startup delays. This is mostly due to the Push-Pull approach followed in the dissemination of the video data that requires the receivers to coordinate the download of different portions of information. On the other hand, Tree-based approaches simply push video packets along routes that are determined by constructing one or many multi cast trees rooted at the media source. These solutions may lead to low latencies but they often assume the source signal to be encoded using Multiple Description Coding (MDC). With this approach independent descriptions can be sent over different network paths. The quality of the reconstruction at the decoder increases with the number of descriptions received. Alas, MDC introduces substantial redundancy relative to single-description coding. This redundancy unnecessarily degrades the rate-distortion performance, if no packet losses or node failures occur.

A middleware orchestrating migrations is installed on each potential host device. This middleware is a further development of the A2M migration manager, an open source middleware for application mobility. The middleware is responsible for listening to incoming migration requests and provides a GUI for the user to manage applications, save states, review system status and request to retrieve or store an application. The migration sequence is carried out in the following steps:

1) Migration initialization. Migration can be initiated either through push (the user wants to send the application to another device) or pull (a user wants to fetch an application). The latter method can be invoked through the GUI or by using an RFID key ring to inform the system that the user has switched device. The following steps will assume migration initialization through pull;
2) Application search. The peer uses the Discovery protocol to find an application Host. Application advertisements containing information about the application and its requirements are collected;
3) Evaluation of host eligibility. The application requirements from the received advertisement are compared with the capabilities of the device. If the rules are passed, the device is considered a eligible host;
4) Migration setup. A socket is created to allow the migration of the application from the original host to the new device. The Pipe Binding Protocol is used to create a unicast communication channel between the two devices. When the host device receives the socket request it also suspends the application and saves its states;
5) Application migration. The application is moved from one device to another along with its states. The states are stored in an external XML-file. This phase lasts until the last ACK is sent and the files are completely written to the new disc;
6) Socket closure. The XTA pipe used for migration is closed;
7) Post-migration Context advertising. Context advertisements are propagated to inform the peers of the new host roles;
8) Application resumption. States are initialized and the GUI is updated. For our experiments, we developed a simple gaming application allowing two users to play a game of Battleships online. The default size of the application is 1340 kB while the states (current ship positions, player name, user preferences etc) are stored in a 2 KB XML document.

Developing a deep insight into how traffic flows through the network is non-trivial to network operators in network design and management, including traffic engineering, failure recovery, bandwidth provision, etc. The network traffic is usually illustrated by a traffic matrix (TM), which presents traffic volumes between each pair of ingress and egress nodes (e.g., routers) in the network. As basic input information, TM in the context of the Internet is crucial for a wide range of traffic engineering (TE) tasks, such as network planning and load balancing. Estimation approaches based on partial network information are well accepted to derive traffic matrices because of the excessively high cost of direct online measurement. The estimation problem can be briefly described as follows. Let y be the column vector of measured link loads and x the traffic matrix reorganized as a column vector. The routing matrix is denoted by A, where Aij is 1 if link i serves in the route(s) of node pair xðjÞ, or 0 otherwise. Then the relationship of the three parameters can be expressed as y ¼ Ax. We can obtain the link load vector y and routing matrix A through SNMP measurements and IGP link weights together with network topology information, respectively. However, the computation of traffic matrix x from the equation above is not straightforward. Since the number of node pairs is much larger than that of links, the matrix A is therefore less than full rank, making the fundamental problem an ill-posed system.

METHODS

Researchers have proposed a variety of methods and models in recent years to make a more convenient and precise estimation. In both the methods and the models are well summarized. These works mainly focus on the estimation of matrices for general traffic regardless of the type of traffic carried over the network. The large volume of P2P traffic significantly increases the load on the Internet, making networks more vulnerable to congestion and failure, and hence brings new challenges to the efficiency and fairness of networks. There has long been a desire for Internet Service Providers (ISPs) to obtain P2P traffic matrices so as to improve overlay routing schemes in a more friendly way for both users and network operators.

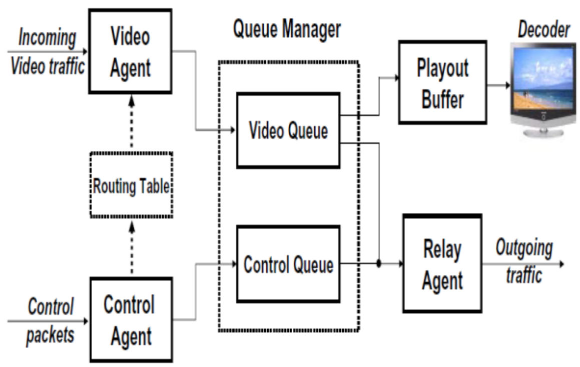

Figure 2. Video streaming Process

Existing models designed for general traffic (e.g., the gravity model) fail to capture the features of P2P traffic, leading to undesirable estimation errors for P2P traffic. Therefore, we argue that a model designed especially for estimating P2P traffic is needed and greatly useful. In this paper, we propose a model to estimate P2P traffic matrices based on a close analysis of the traffic characteristics in P2P systems. To capture the critical properties of the P2P traffic, we take the following physically meaningful factors into consideration. Firstly, the number of peers is considered because, intuitively, networks with more peers might have larger volumes of P2P traffic. Another factor is the traffic localization ratio, which covers the internally exchanged portion of P2P traffic. Last but not least, the distance between different networks is also considered which can precisely reflect the peer selection strategy of the concerned system.

To validate our model, we evaluate its performance using traffic traces collected from both the real P2P video-on-demand (VoD) & file-sharing applications. Using real P2P traffic datasets derived from a P2P Video on Demand (VoD) system and a P2P file-sharing application, we explore how parameters in the P2P model affect the estimation accuracy. To the best of our knowledge, this is the first work that deals with the estimation of P2P traffic matrices. Therefore, we also evaluate the estimation accuracy of our model through a comparison with two typical models proposed for general traffic matrices, namely the gravity model and the independent connection (IC) model. Evaluation results show that the newly proposed P2P model outperforms the other two models in several metrics, including spatial and temporal estimation errors, stability in the cases of oscillating and dynamic flows and estimation bias.

 ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

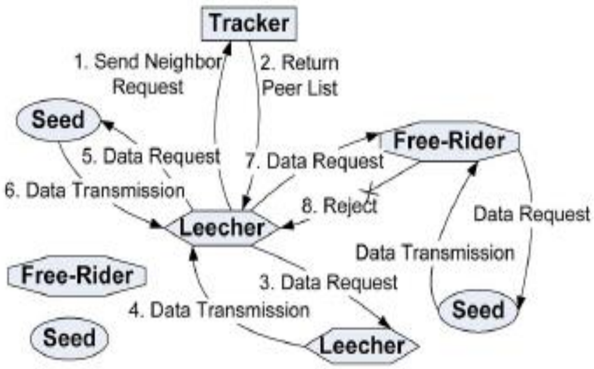

Figure 3. Working Process of a Bit Torrent

IMPLEMENTATION

Authentication with the server:
Every system before starting to transmit or receive Live Media data need to be authenticated by the authenticating server located at any one of the buildings.

RTP transmit system:
In this system, we use RTP protocol since it is best effective for live media transmission. The output should be transmitted in JPEG/RTP or JPEG format so that the transmission is made easy and faster. The JMF API is used to read the source and convert it to JPEG/RTP or RAW_RTP packet data.

Also, we need to specify the port through which we are going to transmit the Data to the requests. Create a processor for the specified media locator. Create an RTP session to transmit the output of the processor to the specified IP address and port no.

RTP receive module:
The output that's being transmitted by the sender system has to be captured by the program and be viewed on the live panel. The panel that is received may be in JPEG/RTP or RAW_RTP. It depends on how the transfer takes place. We need to specify the IP from which Media data are received, the port number through which the server is transmitting the data. It allows receive media streams containing any number of tracks from a number of different hosts/transmitter on the network simultaneously.

Session Tracking:
In the previous module the captured live media is transmitted to a specific receiver (Unicasting). Now, the live media is needed to be transmitted to a list of receivers (Multicasting). And we need to keep track of the session for each media track of the processor.

Create player and RTP managers:
On Receipt of RTP data from the different senders, the received data has to be processed separately. We need to create different players for viewing various media files collected at different ports from various senders involves a few basic steps. The first step is creating the URL. The next steps are to create the player itself. To do this, you utilize the manager class. The Manager class is actually the hub for getting both time base and the players. You need to give the manager the URL where the media file is located.

RTP Monitor:
The RTCP traces RTCP packets such as:
a. Time interval
b. Sender name
c. Host name
d. Packet sequence no

Format:
time interval from sender_name@host_name ssrc=sequence_no

Example:
10:2:12 SR from ruban@home2003 ssrc= 510638401

Storing previous configuration:
When we configure the transmission media that is when we specify the IP address and port number of Transmitting system and receiving system for a particular transmission session, we need to store the configuration for the later working. So that, we need not repeatedly specify the IP address and port number. Each system has specific Home directory. The data's are stored in "xmit.dat" file to this directory. The Home directory path can be finding by System.getProperty ("user.home") method.

Security enforcement:
A sender/transmitting system can be able to prevent themselves from transmitting Live Media to some of the receiving systems. This can be achieved by two ways. Once, the user sitting in the sender system can able to prevent manually by without adding those Receiver IP Addresses in the target. Another way is that the Authorized persons only receive the transmitted media. The unauthorized persons can't able to Login.

GUI:
GUI module is designed to make the user friendlier to the System. It can specify the port through which the Transmission is to occur, the system to which he is going to transmit the Media, port through and to which the data is transmitted etc. The System is designed with the help of Swing with Look and Feel.

Suspension:
The time it takes to store application states. This is performed on the original host in parallel with Migration setup;

Migration:
The time it takes to open a socket between the host device and the host to-be plus the time it takes to migrates the application and its states plus the time it takes to close the socket;

Resumption:
The time it takes to load the application states and update the GUI on the new host;

Context Management/Context-awareness:
The time it takes to find and evaluates the eligibility of the potential host device and update the system with the context changes. Context propagation involves both pre- and post-migration context advertisements and calculations.

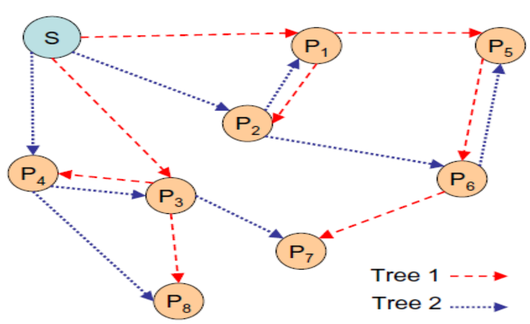

Figure 4. An overlay has two multi-cast trees

Through our decentralized architecture, we achieve a lower degree of network dependency compared to centralized systems and also simplify addition and removal of new nodes. The vulnerability to the loss (or removal) of key nodes decreases as context data is distributed throughout the network. For application mobility to be useful outside the laboratory setting, it must support heterogeneous devices and networks. As these networks can be controlled by different network providers and have different configurations and limitations, our architectural proposal was created with a real world setting in mind, not restricting its deployment to a laboratory, LAN or other user controlled network infrastructure.

Architectures and deployments are all application mobility approaches, not designed for and/or not evaluated using heterogeneous networks with different service providers. We use three different sources of input to inform application migration: device context (e.g. input/output capabilities and system resources), user context (e.g location and preferred host device) and application context (requirements etc.). In a larger setting, context could include more values but also more parameters.

The peer-to-peer network could still be used for propagation, but to minimize migration latency, calculations and context compilation should be handled by a Context Manager, e.g. a middleware communicating with the Migration Manager.

CONCLUSION AND FUTURE ENHANCEMENTS

In this paper, we presented a decentralized system for application mobility, supporting heterogeneous devices and networks, thus differing from traditional centralized application mobility systems deployed in single subnets. After enter in to the system the user can find out, How to transmit and receive the frames in the local intranet. It also provides the detail information about the system for the further development. A prototype was designed and evaluated. System performance was compared with recognized requirements for application mobility, showing the viability of both the concept and our chosen architecture. Additional outcomes of our experiments were that the traditional migration phases of suspension-migration- resumption proved to be insufficient to describe the migration process. We also discussed major challenges and possibilities for the continuing evolution of systems supporting application mobility. Future work should target these areas.

REFERENCES

[1] Requirements, evaluation criteria, and submission templates for the development of IMT-Advanced, ITU-R, Accessed May 2014, [Online] Available: http://www.itu.int/md/R07-SG05-C-0068.
[2] K. Xu,M. Shen, andM.Ye, ‘‘AModelApproach to Estimate Peerto-Peer Traffic Matrices,’’ in Proc. IEEE INFOCOM, Apr. 2011,pp. 676-684.
[3] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian,
[4] A. Nucci,M. Crovella, and C. Diot, ‘‘Traffic Matrices: Balancing Measurements, Inference andModeling,’’ in Proc.ACMSIGMETRICS,2005, pp. 362-373.
[5] H. Xie, Y. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz,‘‘P4P: Provider Portal for Applications,’’ in Proc. ACM SIGCOMM, Aug. 2008, pp. 351-362.
[6] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, ‘‘Modeling TCP Throughput: A Simple Model and Its Empirical Validation,’’ in Proc. ACM SIGCOMM, Sept. 1998, pp. 303-314.
[7] Yin Zhang's Abilene TM. [Online]. Available:http://www.cs.utexas.edu/~yzhang/\\research/AbileneTM/
[8] Internet Studies. [Online]. Available: http://www.ipoque.com/sites/default/files/\\mediafiles/documents/internet-study-2008-2009.pdf
[9] BitTorrent. [Online]. Available: http://www.bittorrent.com/PPLive. [Online]. Available: http://www.pplive.com.
[10] CTorrent. [Online]. Available: http://ctorrent.sourceforge.net/.
[11] Planetlab Testbed. [Online]. Available: http://www.planet-lab.org.
[12] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, ‘‘Fast Accurate Computation of Large-Scale IP Traffic Matrices From Link Loads,’’ in Proc. ACM SIGMETRICS, 2003, pp. 206-217.
[13] X. Zhang, J. Liu, B. Li, and T.P. Yum, ‘‘CoolStreaming/DONet: A
[14] Data-DrivenOverlay Network for Peer-to-Peer Live Media Streaming,’’ in Proc. IEEE INFOCOM,Mar. 2005, pp. 2102-2111.
[15] Z. Ma, K. Xu, J. Liu, and H.Wang, ‘‘Measurement, Modeling and Enhancement of BitTorrent-Based VoD System,’’ Comput. Netw., vol. 56, no. 3, pp. 1103-1117, Feb. 2012.
[16] Y. Huang, T. Fu, D.M. Chiu, J.C.S. Lui, and C. Huang, ‘‘Challenges, Design and Analysis of a Large-Scale P2P-VoD System,’’ in Proc. ACM SIGCOMM, Aug. 2008, pp. 375-388.
[17] D. Qiu and R. Srikant, ‘‘Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks,’’ in Proc. ACM SIGCOMM, Aug. 2004, pp. 367-377.

 ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

[18] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, ‘‘Should Internet Service Providers Fear Peer-Assisted Content Distribution?’’ in Proc. IMC, 2005, p. 6.
[19] B. Liu, Y. Cui, Y. Lu, and Y. Xue, ‘‘Locality-Awareness in BitTorrent-Like P2P Applications,’’ IEEE Trans. Multimedia, vol. 11, no. 3, pp. 361-371, Apr. 2009.
[20] M. Ye, J. Wu, and K. Xu, ‘‘Caching the P2P Traffic in ISP Network,’’ in Proc. IEEE ICC, May 2008, pp. 5876-5880.
[21] A. Medina,N. Taft, K. Salamatian, S. Bhattacharyya, and C.Diot, ‘‘TrafficMatrix Estimation: Existing Techniques andNewDirections,’’ in Proc. ACM SIGCOMM, Aug. 2002, pp. 161-174.
[22] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, ‘‘An Information-Theoretic Approach to Traffic Matrix Estimation,’’ in Proc. ACM SIGCOMM, Aug. 2003, pp. 301-312.
[23] V. Erramill, M. Crovella, and N. Taft, ‘‘An Independent-Connection Model for TrafficMatrices,’’ in Proc. IMC, 2006, pp. 251-256.
[24] A.R. Bharambe, C. Herley, and V.N. Padmanabhan, ‘‘Analyzing and Improving a BitTorrent Network’s Performance Mechanisms,’’ in Proc. IEEE INFOCOM, Apr. 2006, pp. 1-12.
[25] S. Saroiu, P. Gummadi, and S. Gribble, ‘‘A Measurement Study of Peer-to-Peer File Sharing Systems Proc.Multimedia Comput. Netw., Jan. 2002, pp. 156-170.