Technical Report 2003-2-01
Internetworking and Media Communications Research Laboratories
Department of Computer Science, Kent State University
http://medianet.kent.edu/technicalreports.html

Reaction Time Analysis of Self-organizing AVIS Channels on an Active Streaming Services Network (ASN) with Gigabit ANETD Node Cluster

Seung S. Yang & Javed I. Khan
Media Communications and Networking Research Laboratory
Department of Computer Science
Kent State University
233 MSB, Kent, OH 44242
syang|javed@cs.kent.edu
February, 2003

1. Introduction

In this report we preset the control performance of the Active **Video Streaming (AVIS)** system on a Gigabit cluster. AVIS is a distributed video transcoding system, which can use multiple processing capable nodes in a pseudo active network for arbitrary in transit video transcoding. AVIS is a concept demonstration system which allows adaptation at two levels. In the first level AVIS supports video rate adaptation. It can dynamically transform the rate of ISO-/IEC 13818-2 stream [7] using processing capable nodes on any in route processing capable network. In the second level it can self-organize and adapt with respect to the computing power availability in the processing capable network. This reports the deployment and adaptation performance of the system.

1. System Architecture Overview

AVIS is a fully independent application level transport mechanism which can be deployed on various processing capable infrastructures. It has its own module deployment, monitoring and co-ordination mechanisms. To an infrastructure it appears as a collection of communicating modules. Examples of infrastructures on which it can be deployed include Grid, Automatic Computing, or Active Network. It only expects the infrastructure framework to provide remote invocation, and authentication (if any) service. AVIS has a ABONE demonstration system. The system runs of ANETD [8,9] using the Kent VSM/BEE graphical execution environment.

1.1 ANETD & Kent VSM/BEE Architecture

Anetd is an experimental daemon specifically designed to support the deployment, operation and control of active networks. See [8,9] for detail. As shown Figure 1, Anetd acts like node OS. Anetd has some permanent EE, and manage active node. If EE or Active application is required to download for specific channel service, Anetd provides a

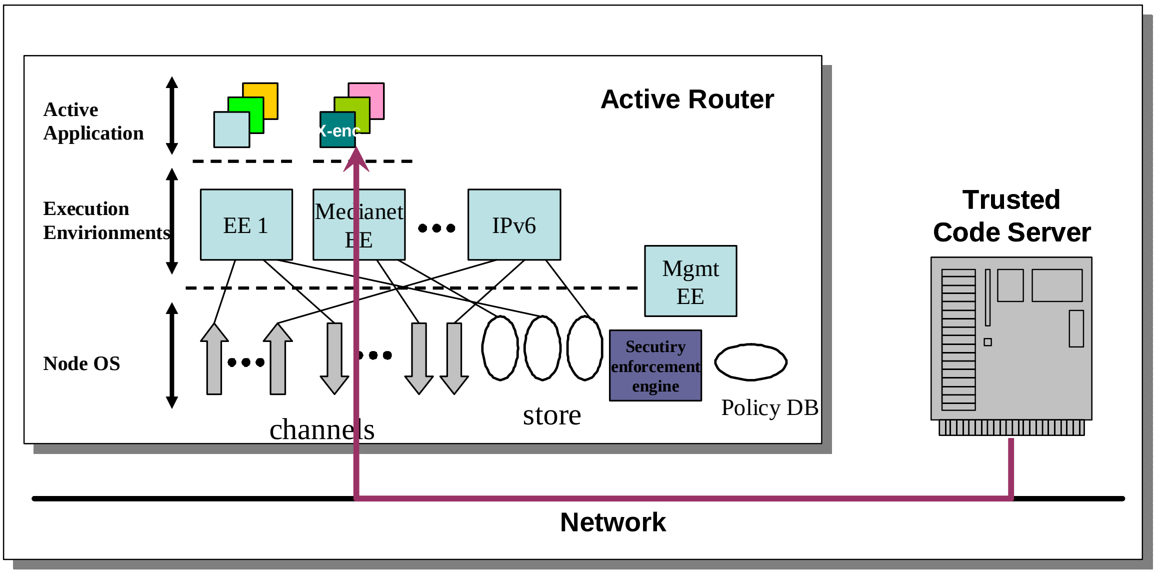

Figure-1 Active Router and Kent VSM/BEE Architecture. Kent VSM/BEE is an Active Network EE and working on Active Router. AVIS System modules could be downloaded from Trusted Code Server

mechanism to download the required EE or application from trusted code server, which is set by active node manager. AVIS system is running on this Anetd and uses our Kent VSM/BEE, in here Medianet EE. AVIS system can be downloaded from trusted code server(s) if it required on an active node.

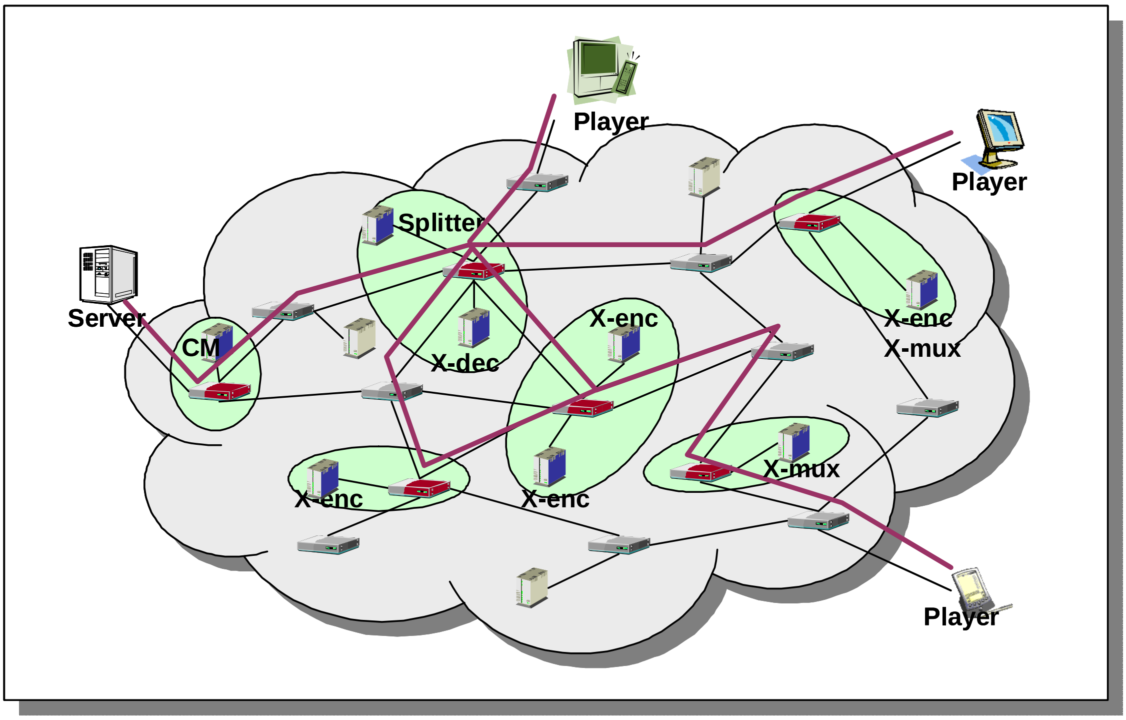

Figure-2 Active Video Streaming System deployment. AVIS modules deployed on a network. It transcodes a stream suitable for players and network environments

1.2 AVIS Architecture

An AVIS channel transcodes a video stream in network for adapting user's requirement and network environments. An AVIS channel transcodes a video in GOP bases and consists of four major components, X-DEC, X-ENC, X-MUX, and Channel Manager, CM. Figure 2, shows an example of AVIS system deployment.

A CM decides module deployment map based on network environment and user's requirements. The deployment should reflect user's requirement, current network resources, and channel module requirements. This involves three parties, user, network, and module developers. All parties' information is fed to the CM and CM generates its deployment map to satisfy the requirements under current network resource availability [1]. The CM then requests VM/EE to install the other three modules based on the deployment map. The CM sends configuration information to each module too. An X-DEC decodes original input video stream, schedules transcoding path according to paths property, and dispatches the segmented video stream to the proper X-ENC to maximize its performances and reduce delays. An X-ENC encodes the decoded video segments to suite client requirements and network environments. An X-MUX aggregates all segmented video stream and send it to the client in sequence. These modules are deployed in network and helps in adapting a video stream to suit a client requirement or network constraint. For example, if current X-ENC(s) can not support user required frame rates than the X-DEC schedules to involve more X-ENC(s) to generate more frames per sec. If network bandwidth has changed, same video stream rate may cause network congestion and dropping packets. The user could not tolerable to watch video with lost packets. In this case, the AVIS channel changes the stream rates suitable for current network. This adaptation can avoid network congestion with little sacrifice of video quality.

2. Performance Measurement Plan

In this test we have used a total of seven active nodes connected to ABONE running the ANETD system in the MEDIANET laboratory at Kent State University. Each machines run RedHat Linux 7.1 or 8.0. These are three AMD Athlon 1.4GHz, one AMD Athlon 1.2GHz, one AMD Athlon XP 1800+, one AMD Athlon XP 1700+, and one Intel Pentium III 500MHz dual processor machines. The video server was placed on a Pentium III 500Mhz machine. The VSM/BEE Control Center (CC) program that hosted the Channel Manager ran from an Athlon 1.2GHz machine. The X-DEC was assigned on an Athlon XP 1800+ box, and the X-MUX was assigned on a Athlon XP 1700+ machine. Maximum of up to three X-ENC units were used in this experiment. Those were run on Athlon type of 1.4GHz machines. The logical mapping is shown in figure 3.

The MEDIANET system also has a cable router which enabled these active machines to be configured in multiple autonomous systems routing configurations. All the modules shared at least 100Mbps Ethernet bandwidth. Connections from X-DEC to X-ENCs were 1 Gigabit Ethernet connections. We compared performances under both 10/100 Ethernet connected active cluster vs. Giga bit connected active cluster.

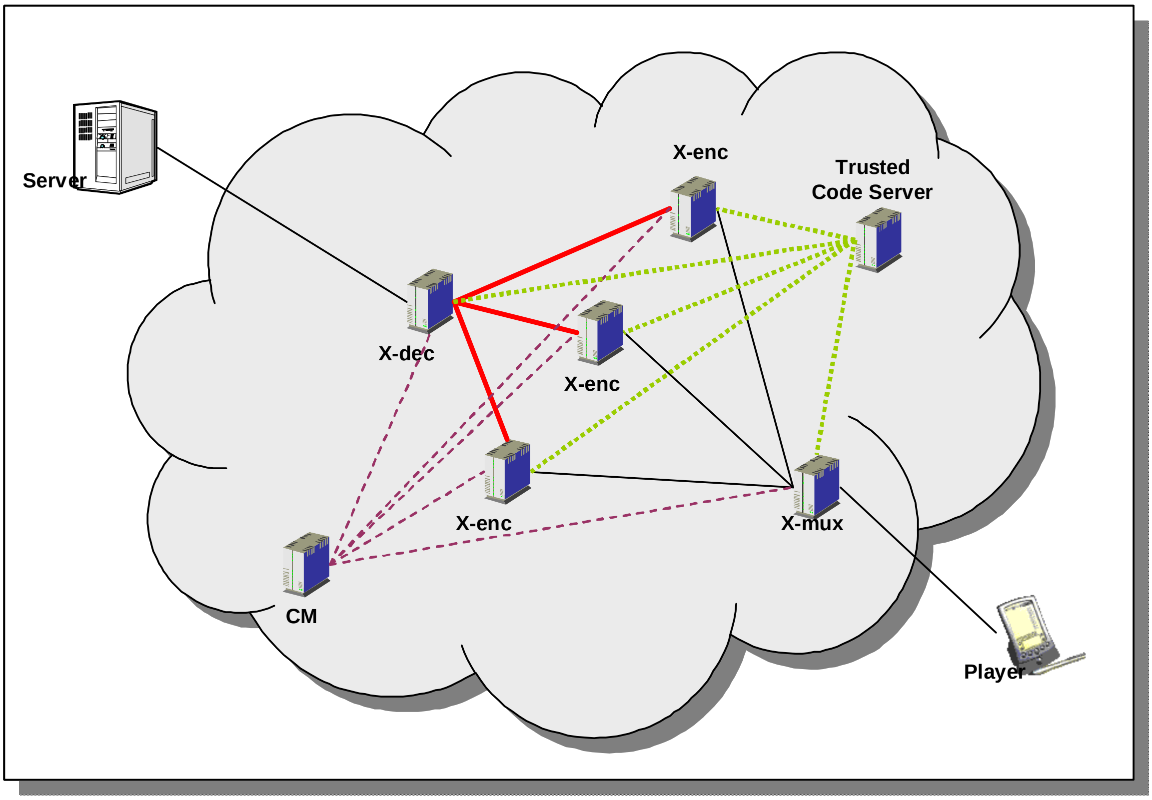

Figure-3 Active Video Streaming Channel test network configurations.

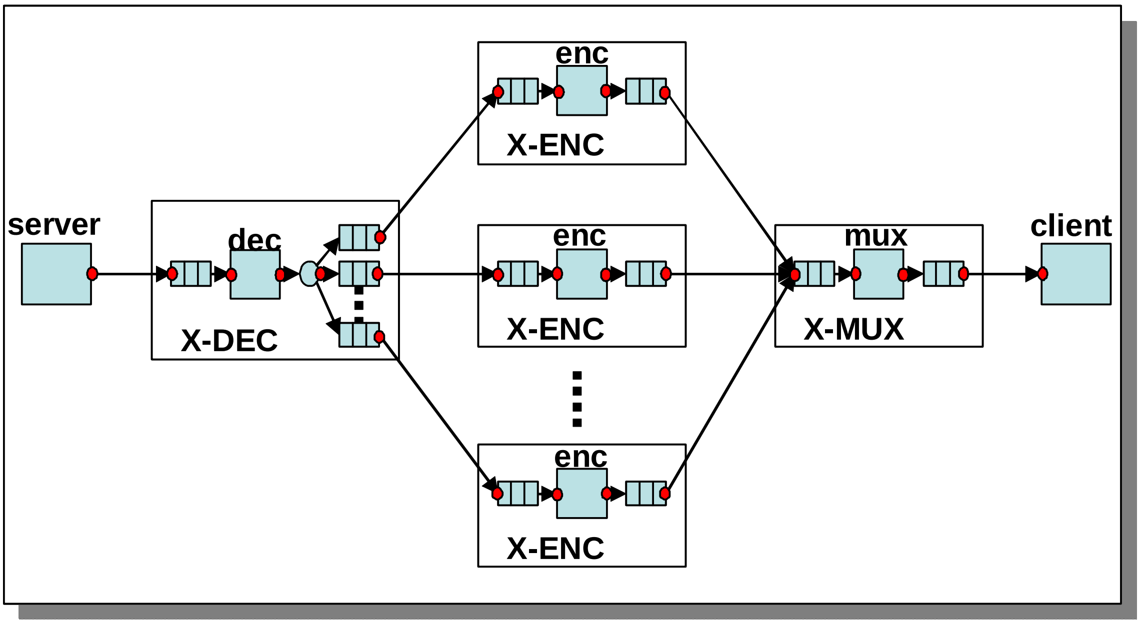

Figure-4 Active Video Streaming Channel Test bed event checking point. All the event is checked in red dot checking point. For example, encoding time is a time from when a video arrived in first red dot of beginning queue to leaving at the last red dot in the queue of X-ENC.

2.1 Event Model:

For event recoding, we have used event check point in the AVIS system as shown in figure 4. Each red dot represents event check point. Network transfer time is counted from the time a packet leaves its queue to the time of the packet is arrived at the beginning of next queue. It includes system time. However, it is normal to consider transfer time in Active Application point of view.

3. Experimental Results

3.1 Automatic Service Deployment

Figure 4 shows the module deployment time in each network. It gathers from two different network topology one for the gigabit Ethernet the other for 100Mbit fast Ethernet. In each case, 5 trials are done. For each trial we observed the module deployment time at the CC. This includes the time (i) CC issues install command (ii) Kent VSM/VEE and Code Server were authenticated by Anetd. (iii) Kent VSM/VEE loaded from the Coder Server to the Designated ABONE node (iv) Kent VSM/VEE load modules and configurations from the Code Server. We recorded the deployment time taken by each of the five components. Figure 5 plots the cumulative time for both networks.

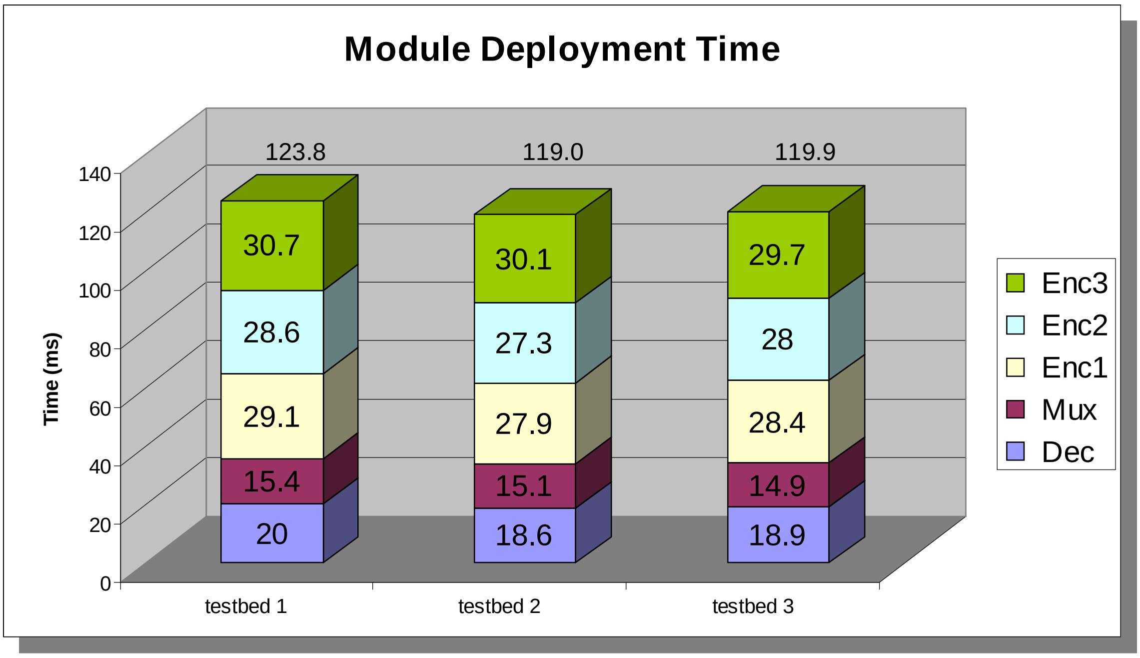

Figure-5 Active Video Streaming Channel Module Deployment Time. The module deployment time is averaged time from 10 trails in each test bed. As shown in this figure, first test bed takes more time while second and third test beds are approximately same.

Observations: The bulk of module deployment occurs at the initial casting of the AVIS channel. The module deployment time is part of the initial reaction time – or cold start delay. As evident, the entire AVIS channel module deployment took approximately 0.15 seconds. In video this is about 5 frame delay. At run time only X-END modules are expected to be dynamically deployed to compensate for active networks processing capacity flux. This takes in the order of 30 ms- which is about 1 frame delay.

Networks, where code server or control center is located in further away, smart enhancements in the module installation schemes like caching, module downloading from nearest replication server, common module pre-loading, pre-staging etc., can be contemplated. Such schemes can easily bring back the cost to the level achieved here.

3.2 Processing Capacity Adaptation

A processing capacity adaptation is happens when a user or a channel manager needs more frame per second for quality transcoding output. In this experiment, a user requests the boosting of frame per seconds, i.e. a user initiate a request event for FPS adaptation. When the processing capacity adaptation events are received on the X-DEC, it uses more X-ENCs by sending transcoding commands and data to more X-ENCs for achieving FPS adaptation.

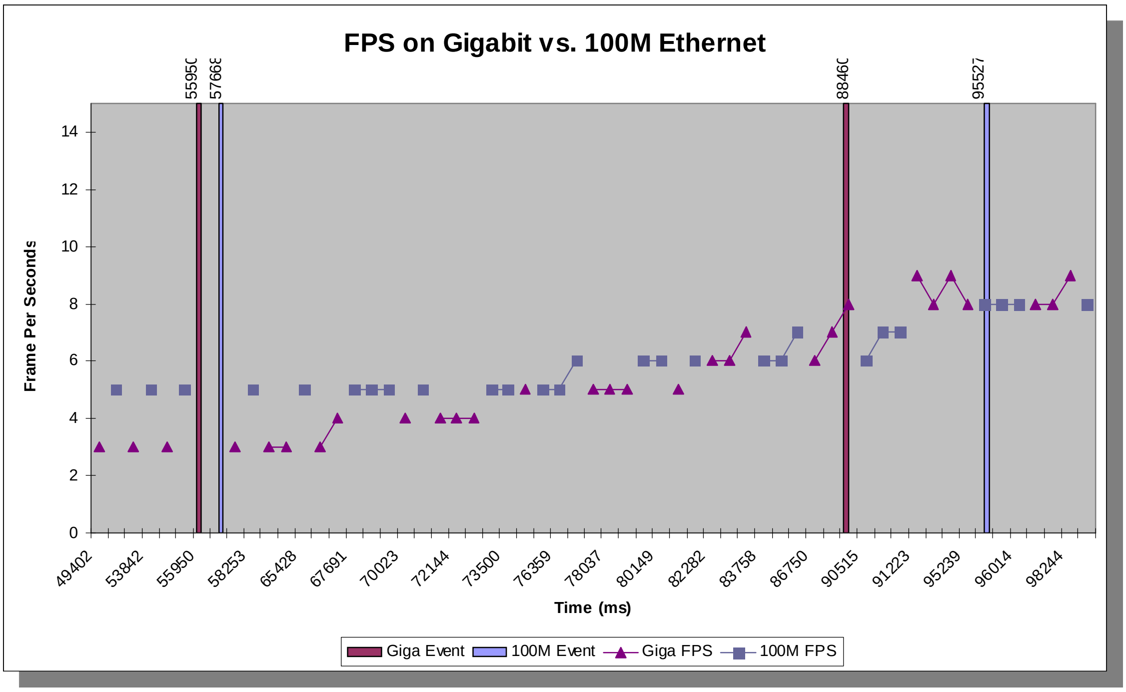

Figure-6 FPS adaptation reaction time. As shown in this figure, the reaction time on Gigabit Ethernet is little bit faster than 100M Ethernet.

Figure 6 shows the frame rate adaptation time. The x-axis is a time sequence and y-axis is showing frame per second. Square dot is for 100M fast Ethernet, while triangle dot is for gigabit Ethernet event of a GOP-ed packet arrival. Left side bars are event times of FPS rate adaptation request. The right side bars are event times of FPS rate adaptation achieved. The distance of same colored bar is FPS adaptation time.

Observations: The frame rate adaptation took some where from 1.2 to 21 seconds. However, a full effect of the adaptation took about 32 to 38 seconds. This large delay can probably be explained by pipe backlog phenomena. The system averages all frame rates in its current window-- not just one instance of arriving of a packet. Therefore, before the new frame rate adaptation takes effect, the previously scheduled video segments already in the pipe should be processed. Also, the new effective video segments must be transcoded and transferred through the X-MUX before the client would notice the change.

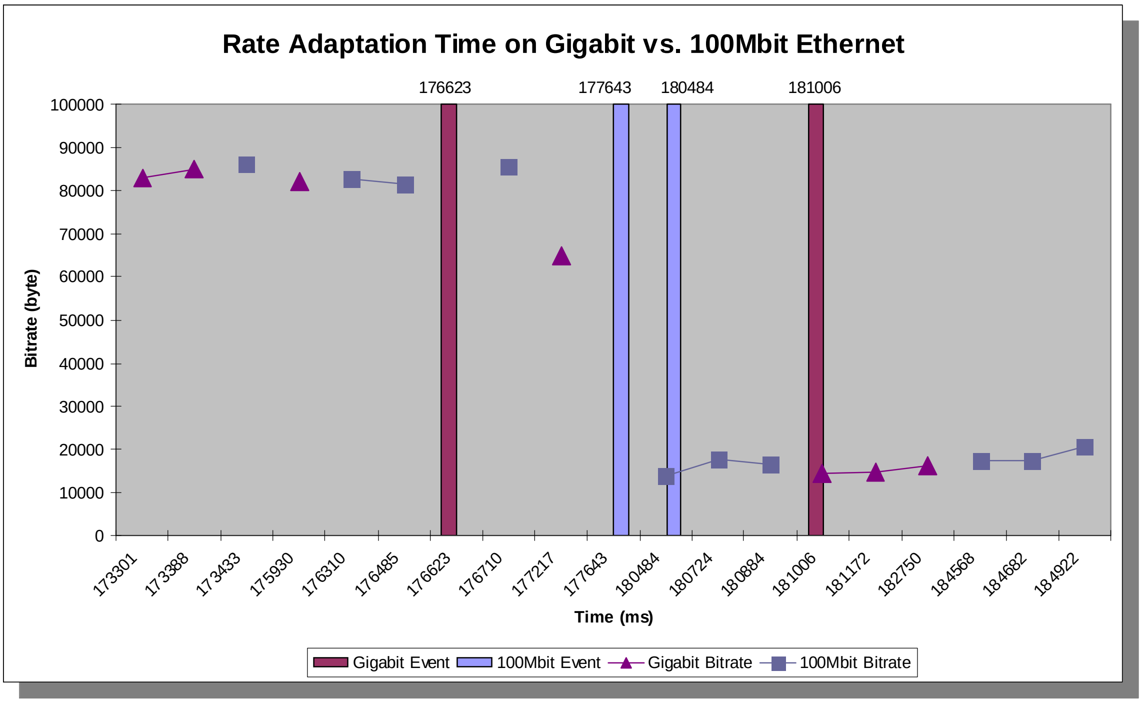

Figure-7 Rate adaptation reaction time. As shown in this figure, the reaction starts 1 to 3 seconds while the completion of reaction takes 3 to 5 seconds.

3.3 Stream Rate Adaptation

A stream rate adaptation happens when network resources, like bandwidth, is dropped and couldn't achieve desired resources from network for transcoding. A stream rate adaptation request is generated by channel manager to adapt the network environment changes. However, in here, channel manager generates a stream rate adaptation request event for experimental purpose. The stream rate adaptation request directly sends to the each X-ENC. When each X-ENC receives the stream rate adaptation request, it is immediately transcoding stream with newly given stream rates.

The video stream rate adaptation time is shown in figure 7. X-axis is time sequence while y-axis shows a GOP-ed packet size in bytes. Left side bars are event happening time of rate adaptation request. Right side bars are adaptation complete time. Again the distance of same colored bar is bit rate adaptation time.

Observations: Above figure shows that the bit rate adaptation time is considerably faster than frame rate adaptation. It does effects the video segments, which are on the encoders at the time of adaptation request received. This short adaptation time is critical in case of network congestion happened. It can avoid network congestion, while it still sending video stream to a user.

4. Discussion

In this experiment we observed that there were some effect of Gigabit Ethernet but it was limited. In this pilot study we focused on AVIS control plane. The connection time is dominated mostly by the network transmission delay. This has not been noticeably improved in Gigabit Ethernet. Also, OS scheduler affects the behavior of the AVIS system because AVIS system uses multi-thread for data transmission. This is beyond application bases time measurements. The adaptation times were dominated by the servers' computational speed constraints.

We plan to conduct further experiment with additional X-ENC pipes. Also, at system level it will be important to identify OS level time consumptions and scheduling penalty on AVIS system. It will identify further impact of the Gigabit Ethernet vs. 100M Ethernet. In this experiment we strictly avoided any significant Gigabit specific reengineering of the application. However, if one needs to take advantage of Gigabit links additional reengineering have to be performed at MPEG-2 transport layer where larger application data units have to be constructed.

5. References

[1] Seung S. Yang, Javed I. Khan, Delay and Jitter Minimization in Active Diffusion Computing, International Symposium on Applications and the Internet, SAINT 2003, pp. 292-300, Florida, January 2003.

[2] Javed I. Khan, Seung S. Yang, Qiong Gu, at el. Resource Adaptive Netcentric System: A case Study with SONET – a Self-Organizing Network Embedded Transcoder. In Proceedings of the ACM Multimedia 2001, pp617-620, Ottawa, Canada, October 2001.

[3] Seung S. Yang, and Javed I. Khan. System Overview of Active Subnet Diffusion Transcoding System for Rate Adaptive Video Streaming. Technical Report: 2001-11-04, Kent State University, [available at URL http://medianet.kent.edu/technicalreports.html, also mirrored at http://bristi.facnet.mcs.kent.edu/medianet]

[4] Javed I. Khan and Seung S. Yang. Resource Adaptive Nomadic Transcoding on Active Network, International Conference of Applied Informatics, AI2001, Insbruck, Austria, Feburary 19-22, 2001.

[5] Javed I. Khan and Seung. S. Yang, A Framework for Building Complex Netcentric Systems on Active Network, Proceedings of the DARPA Active Networks Conference and Exposition, DANCE 2002, San Jose, CA May 21-24, 2002, IEEE Computer Society Press.

[6] Javed I. Khan, Seung S. Yang, Darsan Patel, et. al, Resource Adaptive Netcentric Systems on Active Network: A Self-Organizing Video Stream that Auto Morphs Itself while in Transit via a Quasi-Active Network, Proceedings of the DARPA Active Networks Conference and Exposition, DANCE 2002, San Jose, CA May 21-24, 2002, IEEE Computer Society Press

[7] Information Technology- Generic Coding of Moving Pictures and Associated Audio Information: Video, ISO/IEC International Standard 13818-2, June 1996.

[8] Steve Berson A Gentle Introduction to the ABone, (ISI), OPENSIG 2000 Workshop, Napa, CA, 11-12 October 2000. http://www.isi.edu/abone/intro.html

[9] Steve Berson, Bob Braden, Steve Dawson. Evolution of an Active Networks Testbed, Proceedings of the DARPA Active Networks Conference and Exposition 2002, pp. 446-465, San Francisco, CA, 29-30 May 2002.