Technical Report 2001-11-04
Internetworking and Media Communications Research Laboratories
Department of Math & Computer Science, Kent State University
http://medianet.kent.edu/technicalreports.html

SYSTEMS OVERVIEW OF ACTIVE SUBNET DIFFUSION TRANSCODING SYSTEM FOR RATE ADAPTIVE VIDEO STREAMING (NETAVT V2.0)

S. S. Yang and Javed I. Khan

Internetworking and Media Communications Research Laboratories
Department of Computer Science
Kent State University, 233 MSB, Kent, OH 44242

Last Revised October 31, 2001

1 SUBNET DIFFUSION TRANSCODER

As a part of active network high performance team, in our quest for high performance active networking, this experimental system realizes the novel idea of **subnet diffusion computing**. A stream is actively converted while it transits via an active subnet. This system target a network computing scenario, where to meet the compute cycle requirement, if a single active node is not enough then we adaptively on run time can spread the computation on the neighboring cooperating active nodes to garner more computation power. The system assumes a cluster or subnet of sparsely distributed active nodes, rather than one powerful active node assumed in other active networking models.

This distribution contains the second version of KENT MEDIANET LAB implementation of an ISO-/IEC 13818-2 stream subnet transcoding system. The system can perform dynamic video rate adaptation, and perceptual bit-allocation.

Based on our previous work on multiprocessor transcoding, we have built a three component modular transcoder. The transcoding system has the (a) GOP-Encoder (GOP-ENC), (b) decode-demultiplexer (DE-DEMUX) and (c) GOP multiplexer (GOP-MUX). Each component has been designed an an capsule and can be deployed on any active node.

Each node to be active router also needs an operating system extension module called (d) Lightweight Active Node Service (LANS). The transcoding system also has a manager component called (d) Diffusion Transcoding Channel Control Center (T3C), which manages the overall activities of these transcoder components using the regular and extension services. This distribution contains these four components. Also a server and a player have been included for live demonstration. The architectural detail of the system, including the transcoding architecture, rate control mechanics, component interface and deployment protocol, and diffusion jitter control algorithms are described in separate documents.

The work has been funded by DARPA Research Grant F30602-99-1-0515 under it's Active Network initiative.

2 HOW TO INSTALL A SUBNET DIFFUSION TRANSCODER?

You can download latest distribution release from http://www.medianet.kent.edu. After download, use gunzip command to extracts the distribution zip file. This should give three files. (i) a tar file containing the LINUX systems, (ii) A ZIP file containing the Java systems, (ii) a pre-compiled windows 32bit MPEG-2 player.

The windows player xplayer.exe can be directly run on any player machine with Windows98/me/2000 system.

Unzipping the ControlCenter.zip will result in several Java class and image files in a directory called ControlCenter. This can be executed from any machine with Java Runtime environment.

Untaring the tar file will give the remaining components- including the server. These systems have been tested on Redhat Linux 6.2 and 7.1, but basically you can run it any UNIX like operating systems. Each component are in a separate directory below a directory named ‘demo’. Demo will have subdirectories named controller, xrun, xenc, xdec, xmux, xload, server, and test. To generate the executables run make in the top demo directory. It should generate all the required executables.

3 HOW TO TEST RUN A SUBNET DIFFUSION TRANSCODER?

Now, you have executable files for the LINUX machines (the active routers). Typically, the running of the system will require the PXNet.conf file to be preconfigured. We have already added some default configurations. The values can be modified later from the Channel Control Center.

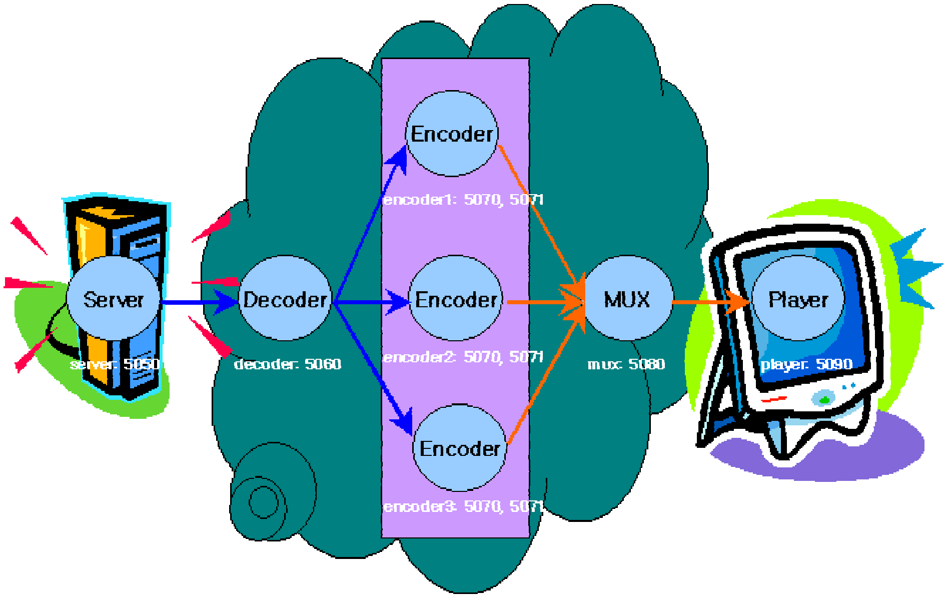
1. First we need to start the LANS to be started on all active routers. This can be done by simply typing “ctrl” in demo directory.

2. Then we need to start the Channel Control Center. This can be done by typing “java ControlCenter” in the directory containing the class file.

3. Then we need to start the Player in the player machine. This can be done by typing “xplayer” in the directory containing the xplayer.exe.

4 NETWORK CONFIGURATION

Before we can start the transcoding we need to provide the network configuration to the system via the Channel Control Center. Let us assume that following is the example deployment scenario.

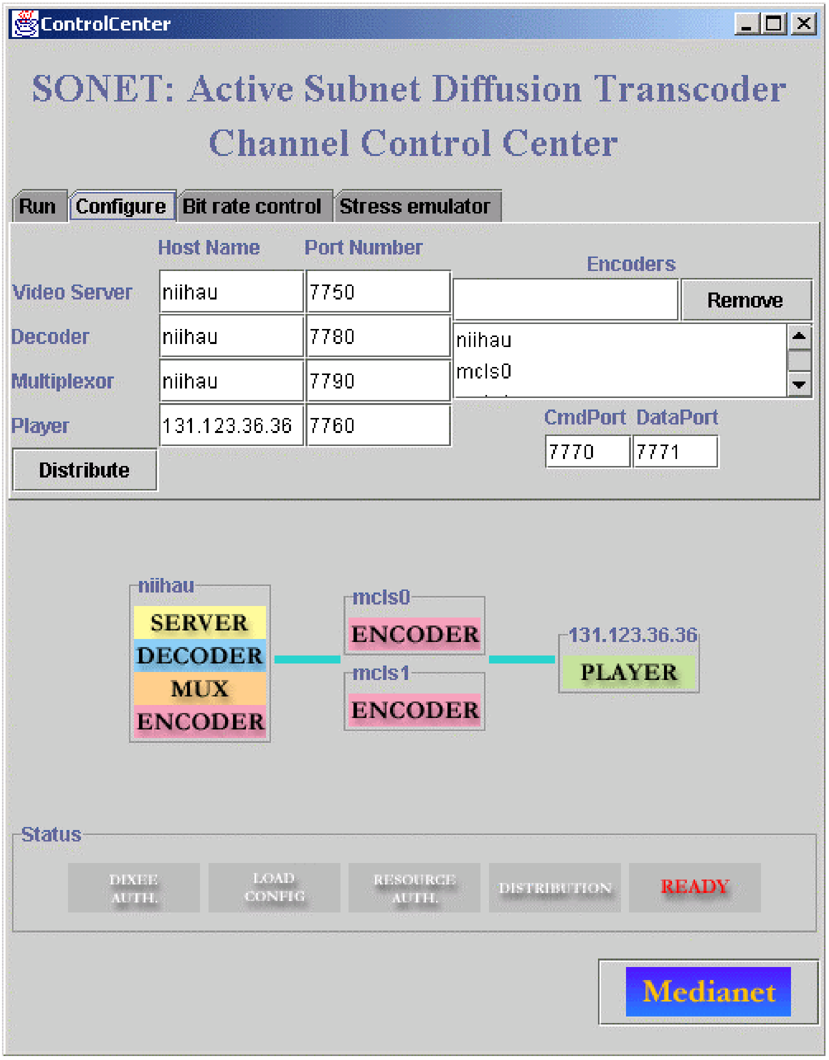


This target network scenario have to be given to the transcoding system. First this network scenario is set by going to the ‘configure’ tab-window of the T3C. When Control Center program shows up, go to configure tab window and make a configurations.

1. In the designated windows type in the DNS names (also it accepts IP address) and port numbers of the components. For running the server, multiplexer, decoder, we designate one node for each. For encoder we can run multiple instances and thus we can designate multiple nodes. Each encoder components needs two ports a control port and a data port. A node can run multiple component instances. A single IP address can appear multiple times. But, port numbers have to be unique (The T3C does not resolve the port number conflict).

2. Once the network designation is complete, press the “Distribute” button to propagate the changed information. It sends appropriate configuration signals to the proper active nodes.

3. Note a XPlayer should be started separately on a node. The T3C does not invoke the player (but needs the player location information for security).



ControlCenter

SONET: Active Subnet Diffusion Transcoder Channel Control Center

	Host Name	Port Number
Video Server	niihau	7750
Decoder	niihau	7780
Multiplexor	niihau	7790
Player	131.123.36.36	7760

Encoders: niihau, mcls0

CmdPort	DataPort
7770	7771

5 Component Deployment

Click on the “run” tab window on T3C. Press “start”. This (deploys and) activates all the modules in the designated places. The transcoding starts based on client request. The “stop” button can be used to halt transcoding operation. The “reset” button can be used to deactivate and then start them again.

6 Playing the Video

Now the player can be started by selecting “play” from players “run” menu. There is already a sample video called SVR.m2v” in the server directory. The system will begin playing this video. Note: It may take 5-10 seconds for video to show up on the player.

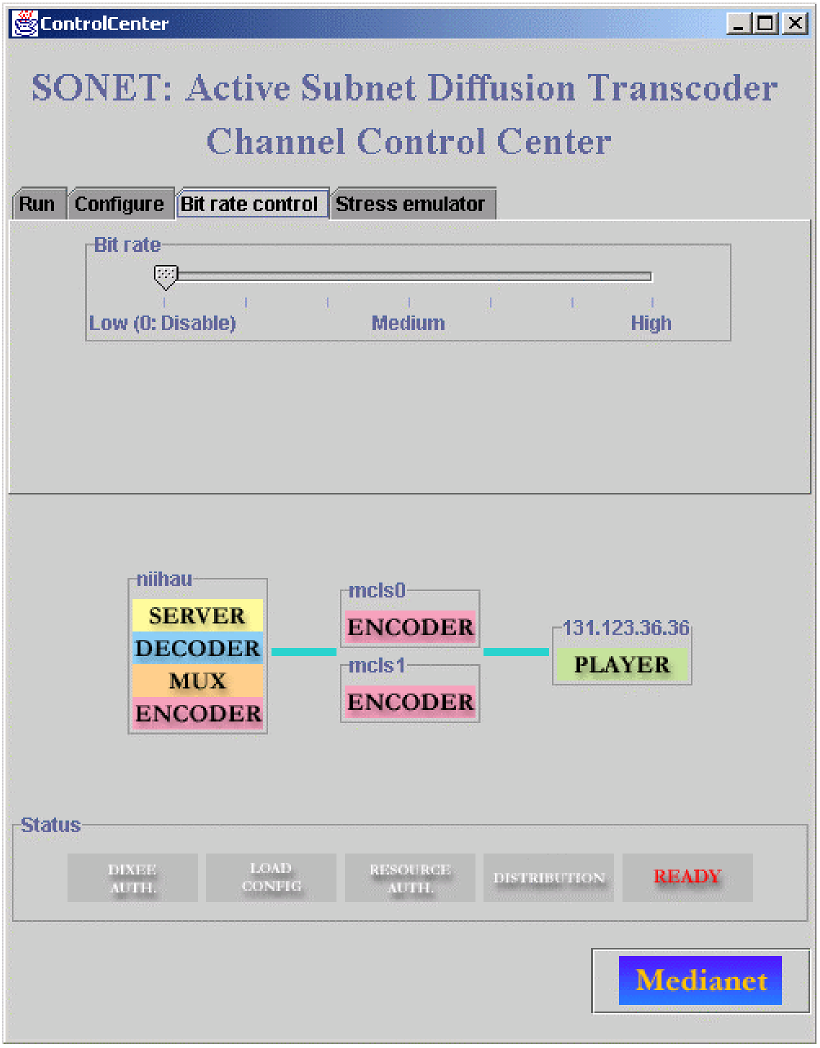
7 How to Serve your own Video

In svr directory, you can find a sample MPEG2 video stream file "svr.m2v." It can easily be replaced by any other MPEG-2 video stream. The default encoding sequence GOP size N=12 and M=3. If a video stream with different sequence is used the test.par file should be modified in the test subdirectory.

8 Transcoder Bit-Rate Control

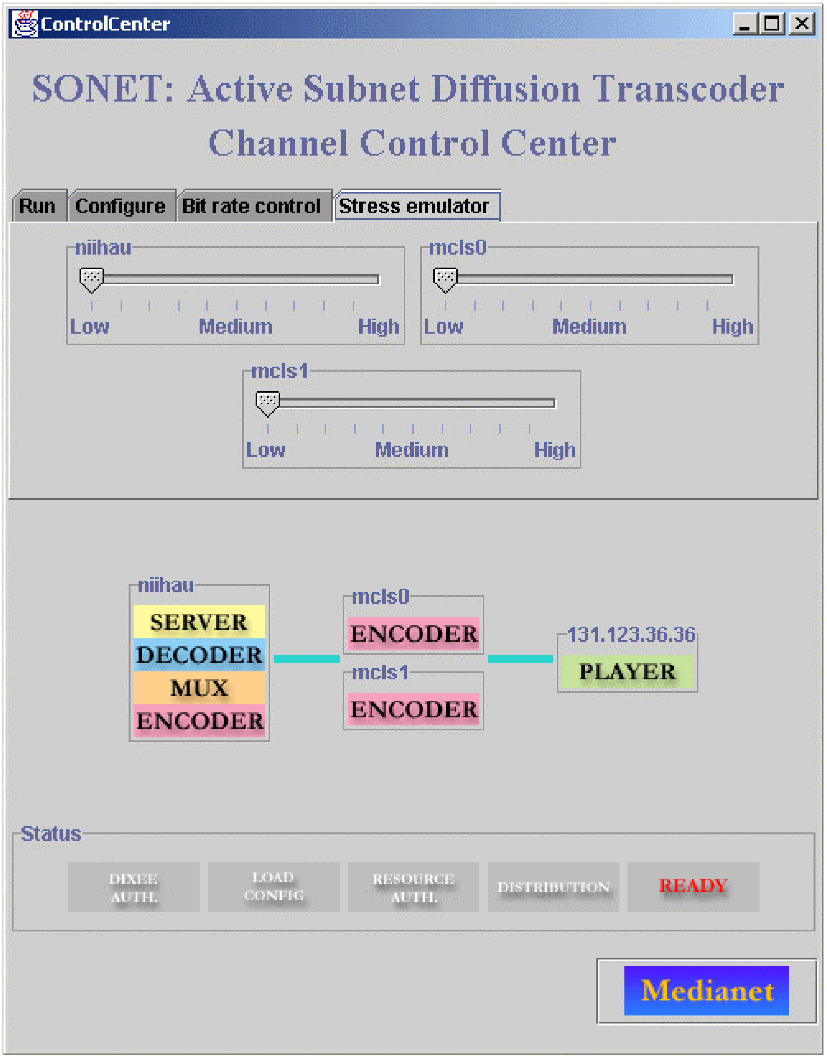
The overall outgoing bit-rate of the transcoder can be adjusted. Go to "Bit rate control" tab window in T3C and change the value. The other slide bar will show the current outgoing frame rate sensed at the multiplexer. This frame rate is averaged over a specified number of GOPs.

Note: There is a file called "gopsize." in the mux directory. This file contains a number which signifies the number of GOP's on which this average is calculated.



9 Active Node Stress Emulation

To observe the adaptive behavior of the system, we have also developed a **stress emulator** that can emulate various levels of active computing cycles allocated to the encoder components in the nodes. For emulating stress on transcoding nodes, “stress emulator” tab window gives handle for stress control. A High means high background load for the node and thus lower cycles available to the encoder component and vice versa.



10 Syntax of PxNet.cnf

ServerName	ServerPort
PlayerName	PlayerPort

EncLuncher	CmdPort DataPort
DecoderName	DecoderPort
MuxName	MuxPort
EncLuncherNo	TotalEncoderLuncherNumber
EachEncoderName	
EachEncoderName	

The ServerName and ServerPort are the host name which will runs a server and it's port number.
The PlayerName and PlayerPort are the host name which will runs a player and it's port number respectively.
In EncLuncher filed, CmdPort and Data Port are command port and data port for the encoder luncher.
The DecoderName and DecoderPort are the host name of a decoder and a port number for it.
The MuxName and MuxPort are the hostname and a port number for a multiplexer.
TotalEncoderLuncher should be changed to total number of encoder luncher.
EachEncoderName should be changed to a host name for each encoder.

11 REFERENCES

1. Javed I. Khan and S. S. Yang, **ARCHITECTURE OVERVIEW OF MEDIANET MULTIPROCESSOR TRANSCODER,** Technical Report 2000-08-01, Internetworking and Media Communications Research Laboratories, Department of Math & Computer Science, Kent State University [http://medianet.kent.edu/~javed/medianet/techreports/active-xcoder/TR2000-08-01-mpAVTarch.pdf]

2. Javed I. Khan, Seung Su Yang, Qiong Gu, Darsan Patel, Patrick Mail, Oleg Komogortsev, Wansik Oh, and Zhong Guo Resource Adaptive Netcentric Systems: A case Study with SONET- a Self-Organizing Network Embedded Transcoder, Proceedings of the ACM Multimedia 2001, October 2001, Ottawa, Canada, pp617-620.