Allan Gu
NGI Research Paper

Streaming Media Extensions to Open Mash and Indiva

1. INTRODUCTION:

One of the most exciting new technologies that have arrived is real-time streaming media. Although current software allows real time video-conferencing between 2 or 3 hosts, when the number of hosts increases scaling becomes a very big issue. This problem can be solved if we used multicast. Multicast allows for the distribution of IP packets to many hosts without having each host sending multiple copies of the packet. Using multicast, the host sends one copy to the multicast address and the routers will replicate the packets and make sure everyone gets a copy of the packet. Unlike TCP/IP, multicast does not provide any guarantees that the packets will get to the user. For the purpose of real-time streaming media, we do not really care if one packet gets lost. As long as most of the packets reach the user it is ok. Because of this, multicast is wonderful technology for real-time streaming media.

Many applications package that will further the development of streaming applications have been developed. One of these packages is the Open Mash toolkit. The Open Mash toolkit contains many layers of abstractions that remove the nitty-gritty details such as encoding and transmitting video packets from users who really don't care how the underlying encoding and transmitting process works. Open Mash also contains the MBone tools so people who have been using the Mbone tools such as VIC can make a quick transition.

Another package is INDIVA, a distributed streaming media and equipment control middleware. Indiva bridges conventional audio/video environment and streaming media. Indiva hides the physical connections and control interfaces giving users software interface to control equipments. Indiva also automatically allocates resources for the user. Indiva contains a set of high-level commands that also hides the details from users who are not interested in those details. Furthermore, Indiva uses a Network File System metaphor. Application mounts a server and issues command to the server. The server will then choose resources on the network that can fulfill the command.

These software packages are good, but it is always a plus to add extensions to them.

2. PROBLEMS:

VIC, a video viewer and transmitter, displays "useless" streams that do not display anything useful. When VIC receives a blue or black stream, it displays it like all other streams. When there are a lot of "useless" streams being sent to the multicast session, the VIC windows can become very crowded and we can quickly run out of screen real estate. Instead of viewing useful streams, the user's screen is filled with streams that are not important to the user.

The useless streams are sent when no one is there, but the machine transmitting the video streams. When a user turns off the camera, the encoder encodes a default blue value and transmits the packets continuously to the multicast session. Another way we have useless streams is when the lights are off. When lights are off and the cameras are on, the encoder picks up a black stream and transmits the stream to the Mbone session.

Indiva is a wonderful middleware but it lacks a caching system for multicast session announcements. We need to create an application for Indiva that will act like NSDR. This application will continuously listen to a well-known multicast address for new multicast announcements and cache it. The application will also remove the caches for any multicast sessions that are no longer available.

3. IMPLEMENTATION:

Remove "Useless" Streams:

The algorithm to remove the blue streams is straightforward. To remove the blue streams we check the YUV frame of the stream we received to see if it is blue. YUV is another representation of colors, like RGB, that takes advantage of the fact that human eyes are much more sensitive to luminance than color. YUV representation allows the computers to represent colors at a lower bit rate.

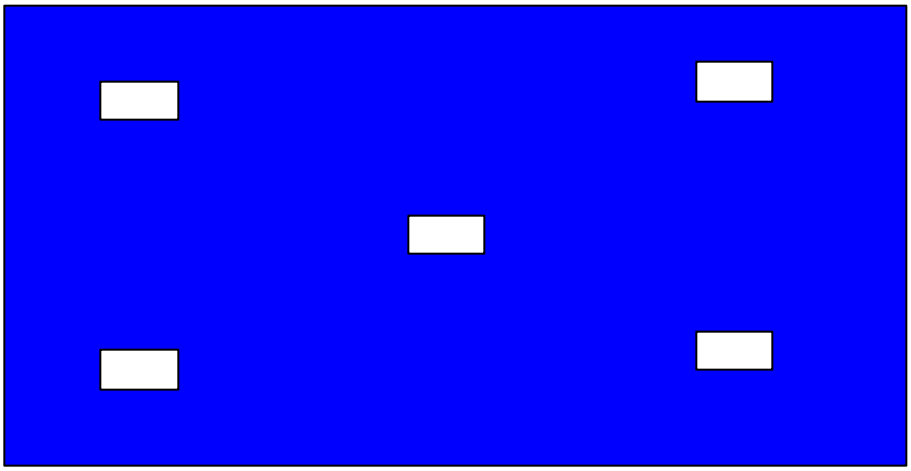


Fig1. Positions Checked

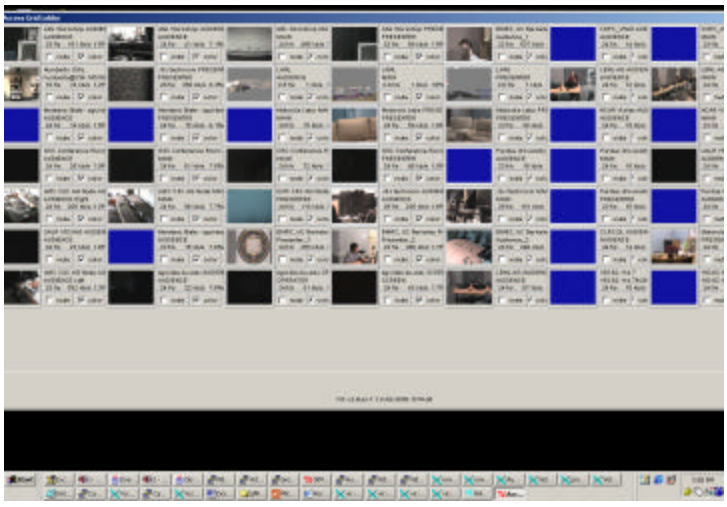
We check the pixel values at 5 positions in the image. Each position is composed of a block of 4 pixels. We select the 4 corners of the image and the center of the image to sample to determine if we have a blue image. We compared each of the pixels at the 5 positions of 4 pixels with the standard value for the color blue. If each pixel has the value of 35, 212, 114, then we say that the image is a blue image. If any of the frames we received is blue, we automatically remove the stream from VIC. We are fairly confident that if all 20 pixels of the image are blue, then the entire image is blue. It is unlikely that a normal image will have 20 blue pixels at different positions.

The algorithm to detect black stream is more complicated than the algorithm to detect blue stream. Unlike the blue streams, we cannot compare the pixels with a standard YUV value because there are many different YUV representations for black. Furthermore, sometime in the black image we may have one or two pixels that are not black, but the overall image is still black.

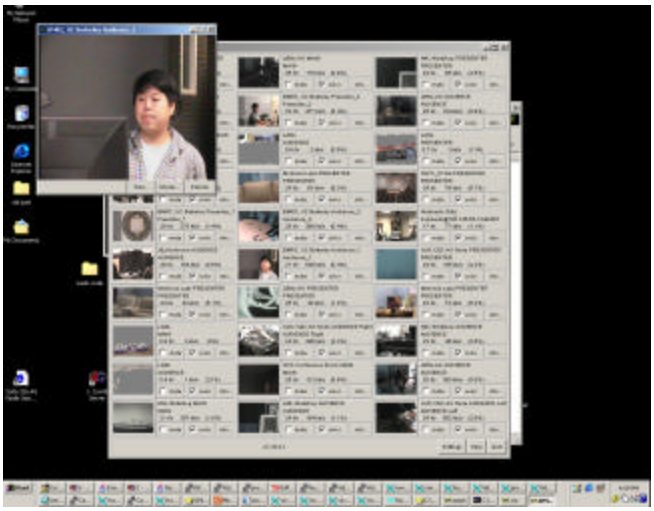
To detect black streams, we examined a total of 80 pixels. If we have more samples, then we can be more confident with our prediction. Because we can't compare values, we set up a simple statistical prediction. We count the number of pixels within a certain range of values. We created four bins (H0, H1, H2, H3) to hold the number of different pixels we see. Three of the bins hold the number of different shades black we have seen and the last bins holds the number of other colors we have seen so far. The range of black are (Y < 24, 120 < U&V < 136), (Y < 40, 120 < U&V < 136), and (Y<60, 104<U&V<152). In addition, we also calculate the average values of each of the pixels we sample. If we are going for the aggressive suppression of black streams, then if half of the pixels are in the range of Y < 24, 120 < U&V < 13 and the average pixels

values are Y < 28, 120 < U&V < 13, we say that the stream is black. If we are using normal suppression, then we need 70 out of the 80 pixels we sample to have Y < 24, 120 < U&V < 13. In addition we cannot have sampled more than 8 pixels colors other than black. When we use the normal suppression criteria, we are trying not to return any false positives and error on not being black.

Once we have a prediction, we tell VIC to drop all packets from the source that is sending the blue or black stream and then removed the thumbnail from the VIC window. We create a callback that will check if the sender is still sending blue or black stream. If the source is no longer sending blue or black stream, we put the thumbnail back in the VIC window. However, if it is still sending blue or black stream, then we create another callback to check the stream later. A command argument was also implemented in VIC so that a user can decide on what they want to suppress. The –S 1 will suppress blue streams, -S 2 will suppress black streams, and –S 3 will aggressively suppress black streams.



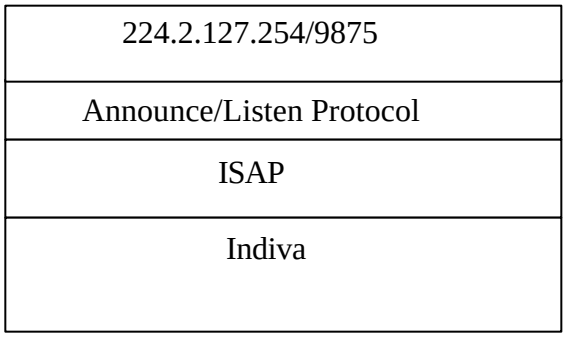
No Suppression



Aggressive Suppression

ISAP:

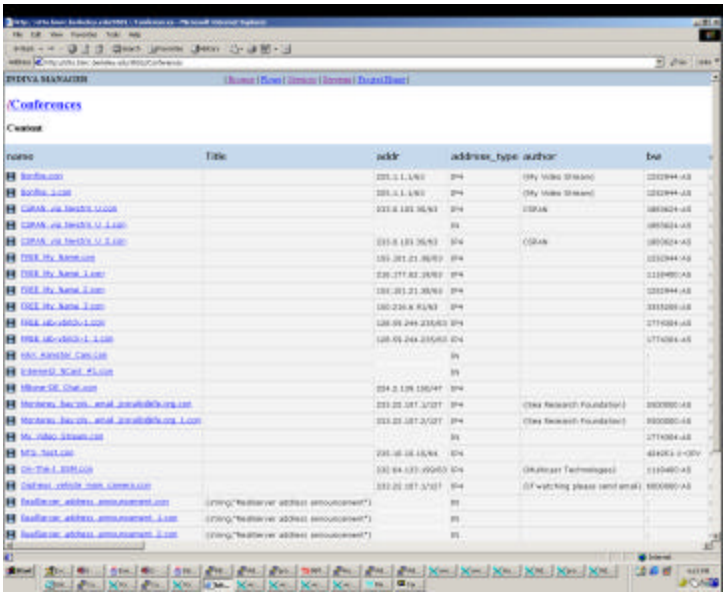
ISAP is a service that caches active Mbone sessions for Indiva. ISAP is like a middle layer, which connect Indiva to the SAP (Session Announcement Protocol) announcement address. To implement ISAP we used the announce/listen protocol that has already been implemented in the OpenMash toolkit. ISAP listens to the well-known multicast address for SAP announcements.



When ISAP receives an announcement it checks to see whether the announcement is new. If it is new, then ISAP will execute the mkcon rpc command in Indiva Manager. ISAP will pass the attributes of the announcement received to Indiva manager so it can create the conference. ISAP also add an entry into an array, which holds all of the distinct SAP announcements ISAP has received. SAP requires that each different Mbone session announcement have a unique identification. We hash the identification number and use it as the index to the array that is holding the SAP announcement received.
By using this identification number, we can decide whether an announcement is new or not. If the announcement is one that we have seen before, we will just discard the

announcement. Once the indiva manager creates the conference, we will update the listing on the web interface. If within 30 minutes we do not received an announcement for one of sessions we have seen, we will remove execute Indiva manager rpc command remove to remove the conference and subsequently from the web page. Anyone who wishes to see what current Mbone sessions are available can just check the web page created by Indiva.

We periodically write the sessions we have seen onto disk. This is just a fail-safe mechanism. This mechanism ensures that we do not lose any information about the sessions that we have seen. Whenever ISAP starts, it checks the file for sessions it has seen before. If a file doesn't exist, ISAP will create that file and store all information in that file.



Web Page with Listing of Conference

3. FUTURE

There are things that need to be added to the extensions that have been made. We should determine whether a stream is blue or black on multiple frames instead of just one frame. So instead of checking only one frame to see if it is black or blue, we will check multiple consecutive frames. If twenty consecutive frames are blue or black, then we can discard the stream. As of right now, if one frame is blue, we say the stream is blue. It is very clear that basing our prediction on one frame is not very accurate consider that most streams are transmitted at twenty-five frames per second.

Furthermore, we want to be able to dynamically changes the positions in the image we check to determine whether it is blue or black. If we are able to change positions we check dynamically, we can be surer that our prediction is correct. As of now, the algorithm checks predetermined positions to see whether it is blue or black.

There are some improvements that we can make to ISAP. A lot of times, we may still receive SAP announcements for Mbone sessions that are no longer available. We need to check the SAP announcements for whether the session is still active. If it is not active, then we do not want to create a conference and an entry on the web page. In addition, we want to allow the users to click on the link on the web page to join and participate in a Mbone session.

Besides adding improvements to the extensions that have been made, I want to continue adding extensions to the Open Mash toolkit and indiva middleware system. I want to add more services to indiva such as a web cast control system.