Large-Scale Personalized Video Streaming with Program Insertion Proxies

Jack Brassil, Taehyun Kim[1]
Mobile and Media Laboratory
HP Laboratories Princeton
HPL-2004-161
September 27, 2004*

multimedia, internet

Increasingly intelligent internet overlay networks promise to deliver streaming media and value-added media services in ways that can not be easily achieved with conventional broadcast networks. Such an overlay would allow an individual viewer or groups of viewers to receive unique programming content (e.g., commercial advertisements, entertainment) that matches their previously specified preferences. Toward this end, we introduce a scalable overlay network architecture and signaling mechanism that permits dynamic program insertions in live, high quality video streams transmitted over IP networks. We describe the implementation of an application proxy that dynamically inserts pre-recorded video programs into NTSC D1 quality Motion-JPEG streams with no visible artifacts. We argue that on-demand program switching is merely a first step towards a new generation of media services; increases in computing power will ultimately permit network-based proxies to manipulate and augment multimedia content as it flows through the network.

* Internal Accession Date Only
[1] College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280
Approved for External Publication
© Copyright IEEE 2004. To be published in IEEE Communications Magazine, 3rd Quarter, 2004

Large-Scale Personalized Video Streaming with Program Insertion Proxies

Jack Brassil and Taehyun Kim

***Abstract*—**

Increasingly intelligent internet overlay networks promise to deliver streaming media and value-added media services in ways that can not be easily achieved with conventional broadcast networks. Such an overlay would allow an individual viewer or groups of viewers to receive unique programming content (e.g., commercial advertisements, entertainment) that matches their previously specified preferences. Toward this end, we introduce a scalable overlay network architecture and signaling mechanism that permits dynamic program insertions in live, high quality video streams transmitted over IP networks. We describe the implementation of an application proxy that dynamically inserts pre-recorded video programs into NTSC D1 quality Motion-JPEG streams with no visible artifacts. We argue that on-demand program switching is merely a first step towards a new generation of media services; increases in computing power will ultimately permit network-based proxies to manipulate and augment multimedia content as it flows through the network.

I. Introduction

During the late 1990s Content Distribution Networks (CDNs) emerged as a preferred architecture for large scale broadcasting of audio and video on IP networks. By deploying networks of edge servers based on inexpensive commodity computers, a new generation of broadcasters sought to address several distribution problems simultaneously. Consumers could receive relatively higher quality media content sent directly from nearby servers, overcoming some of the limitations of a best-effort delivery service. Media creators could reach an untapped audience of internet-connected devices, yet be relieved of the burden of handling distribution themselves. And network operators could rely on inexpensive content storage at edge servers to avoid the addition of costly bandwidth otherwise needed to satisfy the broadcasting demand.

Today streaming CDNs are poised to move to their next stage of development. In this stage we foresee networks providing intelligent services that distinguish their offerings from conventional broadcast mediums. Widespread adoption of internet broadcasting requires not only matching services provided by conventional broadcast networks, but offering services unrivaled by those networks. Yet despite decades of research and commercial development of IP-based video systems, we find that numerous conventional broadcasting services have yet to be demonstrated in IP networks. One such example is a cable television headend's ability to insert local commercial advertisements into a network feed. Performing such an insertion in a video stream on IP networks is not only an intriguing technical challenge, but is also crucial to the economic model that supports the entire distribution system.

We claim that a dynamic program insertion service can not only be readily achieved in an IP setting, but can be vastly superior to that found in conventional broadcast networks. In particular, we envision increasingly *personalized* program insertions. Such targeting could benefit both viewers and advertisers, driving growth in IP-based distribution as well as the development of new and compelling services.

Realizing this vision requires us to overcome several technical challenges. In this article we present a dynamic program insertion service based on an overlay network we call a Content Service Network (CSN) [11]. A major component of this service network is an application proxy capable of switching between high quality video streams without introducing any unappealing glitches or visual artifacts. Conventional broadcast network operators are also keenly aware of the potential benefits of enhanced programming services, including both program switching and insertion, so we provide an overview of existing and proposed protocols providing program announcements and signal insertions in Digital Television settings.

A preliminary version of this article appeared at NOSSDAV'03.

II. System architecture

A *dynamic program insertion service* switches video streams dynamically on behalf of either content providers or users. A conventional insertion at a cable headend attempts to seamlessly interrupt the retransmission of a downstream network feed, substitute a locally-available, pre-recorded commercial spot, and then return to the network feed. Stream switching is triggered by a signal which can be either embedded in a network (video) feed, sent out-of-band on a control channel, or be delivered by a local controller.

Here we assume that a content provider distributes a live video stream via an IP network, possibly operated by an internet broadcaster, network affiliate, or CDN operator. As is the case in conventional broadcast networks, a content provider does not typically own distribution infrastructure, and naturally relies on network operators as distribution partners. Note that the service we consider inserts either live or pre-recorded content into a live stream. That is, the primary content (i.e., network feed) is being received in real-time and is not distributed in advance to the network operator. This distinguishes this service from that of multimedia proxies one might find in a CDN that cache multimedia content for future delivery. The service we provide is also located *inside* a distribution network (i.e., not at a receiver), again distinguishing it from the sequential playout of content retrieved and rendered by a Synchronized Multimedia Integration Language (SMIL) [5] capable media player. Therefore, a network-based multimedia program insertion proxy requires computational resources to process signaling requests and perform program switching, and possibly storage resources for caching pre-recorded content.

A. Service Model

Figure 1 depicts a service model for providing dynamic program insertion in a CSN. This scalable architecture not only enables the dynamic placement of a commercial advertisement within an on-air stream, but permits the insertion of *customized* or personalized advertisements, targeting different advertisements to a large number of distinct groups of receivers based on their specific interests.

The principal CSN system components are:

- *Application Proxy (AP) servers*: Network-based AP servers help deliver value-added services by providing computational and storage resources. For customized advertisement insertion, an AP server performs switching between an on-air stream and a locally stored advertisement. Each AP server maintains different advertisements based on user interest categories. The on-air stream may be multicast from the content source to the APs to achieve efficient distribution. Scalable video multicast techniques can also be used to accommodate variability of available bandwidth and AP server heterogeneity. Each AP also receives and processes signaling messages which trigger insertions into the network feed.
- *Redirectors*: Redirectors direct service requests to AP servers based on user interest. A service request (e.g., description of preferred programming) is initiated by a receiver. A redirector that receives the service request replies with the IP address of an appropriate AP server based on the receiver's interest category. A redirector might select a specific AP from a set with the desired content by also considering the following factors: locality, server load, and type of service.
- *Services Distribution and Management (SDM) server*: An SDM server supervises and manages information on the service execution environment, such as the topological placement of AP servers and the demography of user groups. This information is propagated to redirectors so that they can make a correct decision about which AP server should be assigned to a receiver request.

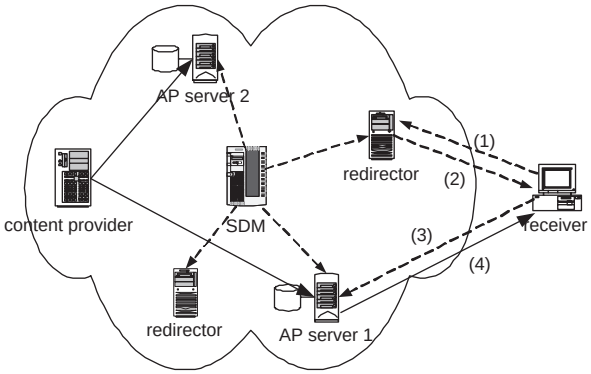

Fig. 1. A model for the dynamic program insertion service over CSN

An example of a system interaction for the user group based redirection is shown in Figure 1. Dotted lines specify the control flows, such as a service request or a connection setup. Solid lines specify the media flows consisting of an on-air stream and advertisement. The interactions in this scenario are de-

scribed as follows:

(1) A receiver sends a service request to a redirector along with a *cookie*. The cookie contains user interest information that is stored by the receiver in advance.
(2) The redirector parses the cookie and selects an appropriate AP server based on the service environment information delivered by an SDM server. The redirector replies for the receiver request to redirect it to AP server 1 which manages and maintains the content for the user group associated with this request.
(3) The receiver sets up a connection to AP server 1 to receive a customized video stream.
(4) Upon the receipt of appropriate signaling messages, the dynamic program insertion function in AP server 1 places customized advertisements into the on-air stream, and forwards the customized stream to the receiver.

Note the intended similarity between the service model in Figure 1 and the current digital cable TV network. A typical digital cable TV network consists of four components: Program Provider (PP), System Operator (SO or MSO), Network Operator (NO), and subscribers. An AP provides a service normally associated with an SO, because the SO places value-added local commercial advertisements in entertainment programs. This service is normally provided today by a digital stream management system (e.g., Terayon's Network Cherrypicker [15]) located at an SO headend. We will return to a detailed discussion of Digital Television systems in Section IV.

B. Design Objectives

The design and implementation of an AP-based program inserter will be our focus for the remainder of this article. Care must be taken in developing the program inserter to provide good performance for the overall system, which should include the following requirements:

- *High quality video*: A dynamic program inserter must be capable of switching high data rate on-air streams delivered by a content provider. Because of the heterogeneity of content providers, inserters must ultimately be capable of switching high quality video distributed in a variety of formats (e.g., MPEG-2, Motion-JPEG). With the increasing speed of the Internet backbone transmission links (e.g., OC-48 and above) and increasing demand for high quality reception, we anticipate rapid increases in the bandwidth of streams arriving to APs. Therefore, we have established an initial requirement that our program inserter be capable of processing streams of CCIR 601 quality video [7].
- *Seamless program switching*: Even a casual viewer of conventional broadcast television has observed that program switching is often poorly executed. This is particularly evident when transitions result in 'deadtime', or when one commercial advertisement begins only to be abruptly replaced by a second. In some cases, lack of synchronization is effectively concealed by special effects (e.g., fade-to-black, split screens). Our goal is to achieve seamless stream switching, such that receivers do not perceive any glitches or visual artifacts at program transitions.
- *Reliability*: Signals for video switching synchronize a content provider and an AP server, and coordinate the delivery of downstream video. These signals must be delivered with high reliability. For this reason we employ a *cueing protocol* [6] capable of either being sent over a reliable channel (where available) or providing arbitrary reliability by transmitting redundant packets. Redundant transmissions can be used when operating over a unidirectional link, or when signaling over a best effort network.

C. Cueing Protocol

A program inserter receives and processes signals that trigger video switching. The signals notify the program inserter of events, such as the beginning and end of an interstice or program gap suitable for an advertisement insertion. Hence, an application level signaling protocol is required to support dynamic program insertion.

Such a signaling mechanism was proposed in [6]. The authors introduced a media-independent protocol for delivering time-sensitive program information, such as program timing, structure, or identity information. A specific program event is indicated by a *cue* packet. It is cue packets that signal when to start or stop a program insertion. Figure 2 shows the payload format of a cue packet.

A program event is specified by a combination of cue type and duration. The meaning of the cue type

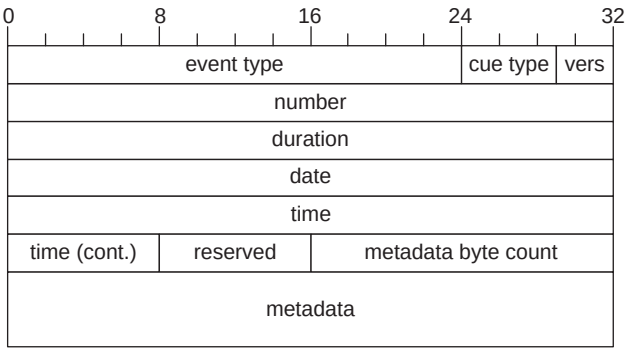

Fig. 2. Payload format of a cue packet

field and the duration field are described as follows. Four (logical) cue types are defined:

- *EN (Event Notification)* specifies the beginning of an event.
- *ET (Event Termination)* specifies the termination of an event.
- The *EP (Event Pending)* cue notifies an AP server of an upcoming event.
- The *EC (Event Continuing)* cue notifies an AP server of an on-going event.

The duration field indicates the remaining time to complete an event. The specific meaning of duration is different according to cue types: 1) A duration of an EN cue specifies the expected time until the termination of an event. 2) A duration of an EP cue indicates the time until the beginning of an event. 3) A duration of an EC cue specifies the expected time until the termination of an on-going event.

Each cue payload is encapsulated within a separate Real-Time Transport Protocol (RTP) packet with dynamically assigned Payload Type (PT) so that a cue packet transmitted in-band with media is easily distinguished from media packets. Though not required, we have assumed the presence of cues embedded within the downstream network video feed to facilitate processing at the AP (i.e., in-band cues). When an AP receives the video stream, the embedded cue packets can be filtered out by the AP server and not delivered to receivers. The separation of cues from media packets helps a system operator to add or remove cues from the stream.

III. Implementation

We implemented a prototype of a dynamic program inserter on top of the video component of *RTPtv* [8]. RTPtv is a system for transmitting production quality video over an IP network. For video capture and playout, RTPtv requires end systems to be equipped with an LML33 Motion-JPEG codec card developed by Linux Media Labs [10]. The codec can encode an analog input into a sequence of JPEG images in real-time. The maximum throughput of the codec is rated as high as 29.5Mbps, which is suitable for generating a D1 quality NTSC video stream with full frame rate.

For efficient processing of an interlaced analog input, the codec generates and groups two JPEG images together. The first image corresponds to the even field of the analog input and the second image does to the odd field. Since there is no field-to-frame conversion, the processing overhead is reduced.

Each image is decomposed into a few restart intervals which begins with a restart marker indicating an encoding or decoding process is reset. Restart markers are embedded in media packets to cope with the packet loss, since the video in a restart interval can be decoded independently. When unreliable transport layer protocols are employed, the effect of packet loss is limited within a restart interval and it does not propagate into the subsequent frames.

For media data delivery, RTPtv employs RTP over UDP [12]. RTP provides a real-time end-to-end delivery including payload type identification, sequence numbering, and time stamping. An RTP flow has an associated Real-Time Control Protocol (RTCP) which monitors the quality of service and delivers information about a session. Large JPEG images are fragmented and encapsulated into multiple RTP-JPEG packets.

A. Dynamic Program Inserter Implementation

Figure 3 illustrates our implementation of an AP-based dynamic program insertion module. The inserter handles two video streams: 1) an RTP stream transmitted by a content provider and 2) a Motion-JPEG stream maintained in local storage. Stream switching is triggered by the arrival of a cue signal.

To realize a smooth transition, an inserter has one video memory and two different frame memories: the video memory which contains JPEG images to be displayed is placed on the codec. Frame memory 1 located in main memory is required to process the received on-air stream in the RTP-JPEG format. When frame memory 1 is active and connected to the video memory, the input RTP stream is forwarded to receivers, while at the same time the dynamic program inserter parses the RTP/RTP-JPEG header, pro-

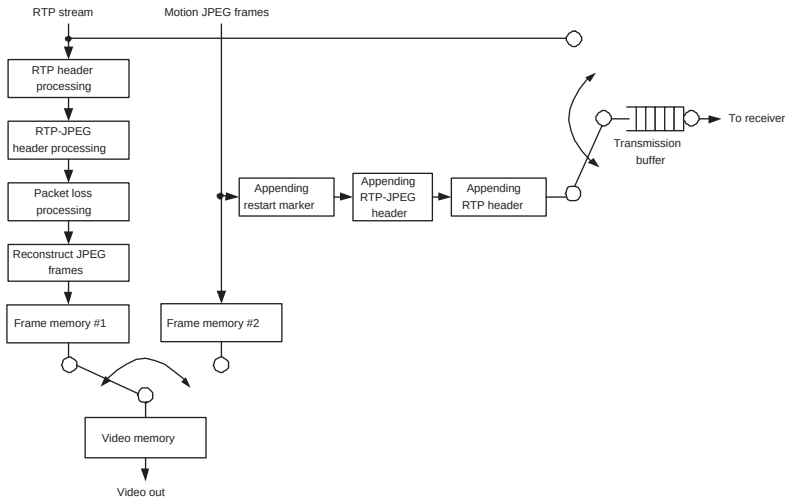

Fig. 3. An architecture of a dynamic program inserter

cesses packet loss using restart markers, and reconstructs the original sequence of JPEG frames. The resulting JPEG sequence is passed to the video memory for display. The inserter can extract and reconstruct JPEG frames from a RTP-JPEG stream since the boundary of a JPEG frame is demarcated by the marker field (M) of the RTP header.

Frame memory 2 processes advertisement streams stored in the local system. The streams are encoded in the Motion-JPEG format consisting of entropy coded image data along with the JPEG header. When frame memory 2 is active, the inserter simply reads Motion-JPEG frames from a local disk and stores them in the frame memory. However, the inserter has to generate RTP/RTP-JPEG headers and embed restart markers to deliver an RTP stream to receivers. The conversion of a sequence of JPEG frames to an RTP stream can be easily implemented by manipulating the video server module in RTPtv: instead of processing video frames encoded by the LML33 codec, the inserter can be redirected to packetize the advertisement stream of the Motion-JPEG format. Since the Motion-JPEG format of the advertisement stream is exactly the same as that of LML33, the packetization module in RTPtv can be used without any modification.

The RTP-JPEG format is not used for stored video clips since some field information in an RTP header (e.g., sequence number, synchronization source (SSRC), contributing source (CSRC), timestamp) should be adjusted dynamically based on different source/timestamp information. This architecture does not incur much overhead, since there is no size limitation encountered in storing advertisement clips compared with relatively small payload size in network transmission. It also eliminates the processing overhead in parsing and reconstructing the original Motion-JPEG stream.

Transitions between frame memory 1 and 2 are triggered by cue packets. When frame memory 2 is filled up with a complete JPEG frame and the beginning of the advertisement insertion event is indicated, the program inserter activates the second memory and the advertisement stream is transmitted. In the same way, the inserter moves back to frame memory 1 when the event is terminated and a JPEG frame is reconstructed in frame memory 1. It should be noted that visual artifacts are prevented on video switching, since the switching does not happen if the target frame memory does not have a complete JPEG frame.

B. Protocol Implementation

We implement the cueing protocol by embedding cue packets in an RTP stream. When an on-air stream is a live program, cue packets are inserted where appropriate between RTP-JPEG media packets. For pre-recorded video, a content provider would need to generate an RTP-JPEG stream (e.g., using `rtptools` [13]) and embed cue packets in that stream.

There are several approaches to achieve reliable switching with cue packets transmitted over a best-effort delivery service such as UDP, including: 1) RTP error correction techniques, 2) out-of-band delivery over a reliable transport protocol, or 3) redundant cue delivery. We employed the last approach; a content provider issues multiple cues over time to signal the same event.

The redundant cue delivery scheme worked as follows. The first EP packet is delivered 8 seconds be-

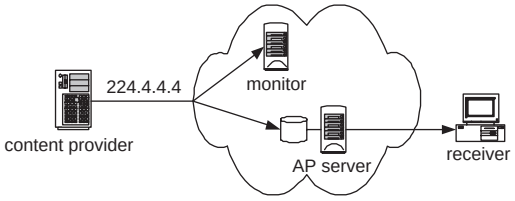

Fig. 4. Testbed implementation

fore the program switching so that a dynamic program inserter is ready. This lead time is consistent with the early notification time used to prepare analog tape equipment for insertions in analog cable systems. On receiving the EP packet, the inserter opens an advertisement file and loads initial few frames. A redundant EP packet is sent 0.5 second before the program switching time. To instruct an AP to perform the insertion, an EN cue is delivered at the desired instant of program switching. Note that the loss of an EN cue is not critical if any EP packet is received, since the duration field in EP cues indicates the time until the beginning of advertisement insertion. During the advertisement insertion, EC cues are transmitted once every second notifying an on-going advertisement insertion. At the end of advertisement insertion, an ET cue is issued and the video memory is switched to frame memory 1. Note that the duration field in EC cues indicating the expected time of event termination improves the reliability, hence the loss of EC or ET cues is not so critical as long as the inserter receives any of them.

C. Testbed and Results

Figure 4 shows our testbed implementation which comprised four Linux-based desktop computers running the 2.2.16 kernel. One system operated as a content source, transmitting a 5–20 Mbps Motion-JPEG stream with embedded cues (i.e., a network feed). For our testing we transmitted content from both a live camera source as well as pre-recorded video to establish repeatability of our results.

A second machine performed the program insertion upon receipt of the incoming video. The third machine operated as a typical receiver, rendering and displaying the content. Each machine was equipped with the LML Motion-JPEG codec to support encoding/decoding of the high rate video stream. There was an optional machine that performed monitoring using `rtpdump` [13]. Since the content provider multicast a video stream to a multicast address of 224.4.4.4, the monitor simply joined the multicast group and subscribed to the RTP stream.

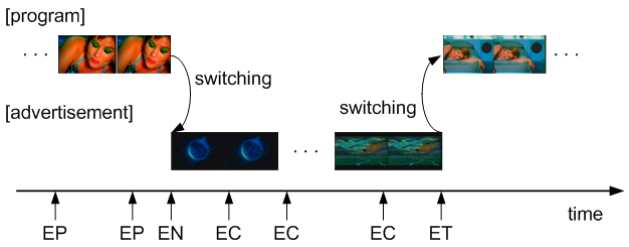

Fig. 5. Program insertion experiment timeline

Fig. 6. Screen snapshot

A timeline for an insertion experiment is illustrated in Figure 5. The video component of a popular music video was forwarded to a proxy, which inserted an advertisement at 15 seconds and then returned to the music video at 35 seconds. Figure 6 shows a screen snapshot at the insertion node, which rendered and displayed the video content being forwarded as well as displayed the arrival of cue packets, switching operations, and errors in a secondary text-based window.

As we had hoped, the program insertions occurred flawlessly. Figure 7 depicts part of the `rtpdump` output captured by the monitoring machine. Numbers in the first column specify the timing information in seconds when an RTP/RTCP packet is delivered. The other values are corresponding to the field information in the RTP/RTCP packet.

Arriving cue packets are shown in bold in Figure 7. The first EP packet was delivered at 6.998 seconds, 8 seconds before the pending insertion. Note that `rtpdump` does not recognize cue packets, since the payload type for cue packets is not defined and was arbitrarily assigned as `pt=35` in the experiment. The second EP cue was delivered at 14.473 seconds, a half second before the program insertion. An EN cue specifying the beginning of the program insertion was received at 14.974 seconds, and EC cue

```
6.998000 RTP len=1449 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=11648 ts=24334800 ssrc=0x3d4011eb
6.998000 RTP len=146 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=26 (JPEG,0,90000) seq=11649 ts=24334800 ssrc=0x3d4011eb
6.998000 RTP len=44 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=35 (????,0,0) ext_type=0x4d ext_len=1 ext_data=00000d18

14.473000 RTP len=1416 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=17450 ts=25008909 ssrc=0x3d4011eb
14.473000 RTP len=628 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=26 (JPEG,0,90000) seq=17451 ts=25008909 ssrc=0x3d4011eb
14.473000 RTP len=44 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=35 (????,0,0) ext_type=0x4d ext_len=1 ext_data=00000d18

14.974000 RTP len=1424 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=17839 ts=25053950 ssrc=0x3d4011eb
14.974000 RTP len=671 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=26 (JPEG,0,90000) seq=17840 ts=25053950 ssrc=0x3d4011eb
14.974000 RTP len=44 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=35 (????,0,0) ext_type=0x4d ext_len=1 ext_data=00000d08

15.974000 RTP len=1415 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=18603 ts=25144031 ssrc=0x3d4011eb
15.974000 RTP len=1006 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=26 (JPEG,0,90000) seq=18604 ts=25144031 ssrc=0x3d4011eb
15.974000 RTP len=44 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=35 (????,0,0) ext_type=0x4d ext_len=1 ext_data=00000d20

16.942000 RTP len=1431 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19346 ts=25231109 ssrc=0x3d4011eb
16.942000 RTP len=1364 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=26 (JPEG,0,90000) seq=19347 ts=25231109 ssrc=0x3d4011eb
16.973000 RTCP len=48 from=224.4.4.4:2347
(SR ssrc=0x3d4011eb p=0 count=0 len=6 ntp=3236008067.4056610870 ts=25232614 psent=215955 osent=289639897)
(SDES p=0 count=1 len=4 (src=0x3d4011eb CNAME="vertigo"))
16.973000 RTP len=1395 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19348 ts=25232610 ssrc=0x3d4011eb
16.973000 RTP len=1458 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19349 ts=25232610 ssrc=0x3d4011eb
16.973000 RTP len=1466 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19350 ts=25232610 ssrc=0x3d4011eb
16.974000 RTP len=1465 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19351 ts=25232610 ssrc=0x3d4011eb
16.974000 RTP len=1417 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19352 ts=25232610 ssrc=0x3d4011eb
16.974000 RTP len=1364 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19353 ts=25232610 ssrc=0x3d4011eb
16.974000 RTP len=1357 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19354 ts=25232610 ssrc=0x3d4011eb
16.974000 RTP len=1405 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19355 ts=25232610 ssrc=0x3d4011eb
16.974000 RTP len=1452 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19356 ts=25232610 ssrc=0x3d4011eb
16.974000 RTP len=1394 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19357 ts=25232610 ssrc=0x3d4011eb
16.974000 RTP len=1383 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=19358 ts=25232610 ssrc=0x3d4011eb
16.974000 RTP len=979 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=26 (JPEG,0,90000) seq=19359 ts=25232610 ssrc=0x3d4011eb
16.974000 RTP len=44 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=35 (????,0,0) ext_type=0x4d ext_len=1 ext_data=00000d20

17.975000 RTP len=1465 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=20126 ts=25322692 ssrc=0x3d4011eb
17.975000 RTP len=534 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=26 (JPEG,0,90000) seq=20127 ts=25322692 ssrc=0x3d4011eb
17.975000 RTP len=44 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=35 (????,0,0) ext_type=0x4d ext_len=1 ext_data=00000d20

34.959000 RTP len=1394 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=0 pt=26 (JPEG,0,90000) seq=33234 ts=26852573 ssrc=0x3d4011eb
34.959000 RTP len=1331 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=26 (JPEG,0,90000) seq=33235 ts=26852573 ssrc=0x3d4011eb
34.959000 RTP len=44 from=224.4.4.4:2346 v=2 p=0 x=1 cc=0 m=1 pt=35 (????,0,0) ext_type=0x4d ext_len=1 ext_data=00000d10
```

Fig. 7. An `rtpdump` output of a program insertion experiment

packets were issued in every second (e.g., at 15.974, 16.974, 17.975 seconds).

Observe that an RTCP message was received at 16.973 seconds, which was delivered over a different port number from that of RTP. The message carries the sender report (SR) and the source description (SDES) information. The SR contains transmission and reception statistics from a sender, such as the NTP timestamp, the RTP timestamp, sender's packet count representing the number of RTP packets, and sender's octet count representing the number of payload octets. The SDES contains source description items, such as CNAME, username, e-mail address, phone number, user location, or application name. In this example, only the CNAME item (`"vertigo"`) was received, since CNAME is mandatory in [12].

Another observation is that a JPEG image was delivered over 12 RTP packets with the sequence numbers from 19348 to 19359. Using the length information, we can find that these packets carry 16.5 Kbytes of image data. Since a video frame generated by the LML33 codec consists of two JPEG images specifying interlaced video fields, each image separated by the marker bit corresponds to either an even or odd field of the input video. Therefore, it can be estimated that the data rate is approximately 16.5 Kbytes/field, or 8 Mbps.

Finally an ET cue was received at 34.959 seconds and the program insertion was finished. Note that in a deployed system all cue packets could be removed rather than forwarded to receivers. On the other hand, the same kind of event notification information that facilitates program switching could be very valuable to receivers capable of processing them (e.g., digital video recorders).

Interested readers are encouraged to view the captured video of this experiment from our web site [9].

IV. Program Announcements and Insertions in Digital Television

The development of protocols and proxies to provide content insertion in RTP-based media transport systems has been motivated by the desire to emulate the operation of program insertions in cable headends. However, with the advent of Digital Television (DTV), that model has evolved considerably in recent years. In this section we describe those recent changes, and compare and contrast certain comparable protocols with those of an RTP-based system using program cues.

Two distinct protocols are used in DTV to provide advertisement insertion cues to MSOs and program information to receivers. Cues accompany the network feed to signal program insertion opportunities.

Because of the coexistence of both analog and digital transmission systems, both analog cues (i.e., out-of-band cue tones) and digital cues continue to be used and supported by headend automation equipment. The Society of Cable Telecommunication Engineers (SCTE) DVS 253 standard specifies the digital cue packets, and indicates how the packets trigger digital program insertions at standard-compliant cable headends [14]. Each digital cue message specifies the number and duration of available insertion opportunities, and specific splice points in the transport stream.

On the terrestrial leg from an MSO to a receiver (e.g., set top box) the Advanced Television Systems Committee (ATSC) has standardized the *Program and System Information Protocol* (PSIP) in ATSC Standard A/65 [2]. PSIP is essentially a collection of data organized in tables to convey both *system information* and *program data*. System information permits navigation between virtual channels multiplexed within the DTV transport stream, while program information details program content to facilitate browsing and program selection via an Electronic Program Guide (EPG).

The base set of PSIP tables and their functions are as follows:

- *System Time Table (STT)* provides time-of-day service.
- *Rating Region Table (RRT)* provides program rating service known as the *Television Parental Guidelines*.
- *Virtual Channel Table (VCT)* provides a list of active virtual channels, channel identification information, and associated channels to facilitate channel navigation by receivers.
- *Master Guide Table (MGT)* provides a master index service for other tables, and conveys system information.

ATSC also recommends a rate at which each table should be issued for proper system performance [1].

The combination of information found in the VCT, EIT and STT can be used by a capable receiver to create an electronic program guide, and also to associate or switch between programs on different virtual channels. The VCT is used by receivers to indicate the appropriate channel to tune; VCTs are issued at 400 ms intervals. A typical initial use of the VCT by a receiver will be to associate a program simulcast in both digital and analog, and to enable automatic 'switch over' to the preferred format.

TABLE I
RECOMMENDED PSIP TABLE UPDATE INTERVAL.

PSIP Table	Interval
MGT	150 ms
VCT	400 ms
EIT-0	500 ms
STT	1 s
EIT-1	3 s
EIT-2, 3, ... , 128	60 s
DCC (in progress)	150 ms
DCC ($<$ 5 s following DCC)	150 ms
DCC ($>$ 5 s following DCC)	5 s
DCC (10 s prior to DCC)	400 ms
DCCSCT	60 s
DCC (in progress)	150 ms
RRT	60 s

An Event Information Table (EIT) carries program schedule information for each virtual channel. Each EIT carries program information for three hours of programming, and up to 128 EITs may be issued to communicate 16 days of program information. Transmitting a minimum of four EITs is required, while maintaining 24 is a recommended practice. Each EIT, when sent at the recommended update rate, consumes only about 128 bps of bandwidth. The *current* EIT, referred to as EIT-0, must contain certain information about current or immediately pending programs including closed caption and ratings information. It is recommended that EIT-0 be sent every 500 ms, EIT-1 be sent once every 3 seconds, and remaining EITs be issued once every minute.

Each EIT contains the following information:

- Program start time
- Program duration
- Program title
- Program description (optional text)
- Program content advisory data (optional)
- Caption service
- Audio service

The program title is recommended to fit within 30 characters to support the widest variety of receivers including appliances with simple character LCDs. Additional text can be sent as a program description or in a separate Extended Text Table (ETT) as needed.

TABLE II
PSIP TABLE SIZE AND BIT RATE.

Parameter	STT	MGT	VCT	RRT	EIT	ETT
Max. Section Size (bytes)	1024	4096	1024	1024	4096	4096
Max. Sections per Table	1	1	256	1	256	1
Max. Table Size (kbytes)	1	4	256	1	1024	4
Typical Table Size (bytes)	20	138	443	901	356 per channel	520 per event
Max. Bit Rate (kbps)	250				250	250

A relatively recent addition to PSIP has been the addition of two optional tables to support a *Directed Channel Change.* This service enables broadcasters to assert a virtual channel change in either an interactive or automatic fashion. Such a channel switch could be unconditional, or based on subject matter or program ID, geographic or demographic information, or a content advisory.

The RTP-based cueing protocol model differs from the model used in DTV in several regards. First, a single end-to-end protocol is envisioned to incorporate time-sensitive program information transfer and trigger remote 'channel switching' operations. The rationale for this approach is that we envision an IP media architecture where program operations may be performed at multiple points in the path between the content origin and the receiver, including at the receiver itself. The architecture of existing DTV systems – satellite broadcast followed by terrestrial distribution – will be only one possible architecture for IP-based media distribution. As program insertion operations may occur at multiple points in the path between source and destination, there is no natural or preferred protocol termination point (other than at the receiver). Since program cues arriving with the network feed need not be forwarded on towards a receiver, private signaling can be achieved between any cooperating parties on the downstream path.

A second difference between an RTP-based cueing protocol and DTV protocols is that the latter is closely tied to MPEG-2 transport streams, while the former is intended to be largely independent of the encoding of media. As new applications are created which demand new media encodings, these encodings could be rapidly embraced without necessarily requiring changes to the mechanisms supporting program announcements and insertions.

Another distinction between PSIP and program cues is the timeliness of information. Program cues are intended to carry highly time sensitive signaling. Other mechanisms, such as HTTP, could easily be used for longer term program information, such as might be required to complete a daily or weekly electronic program guide. By contrast, information in EIT-0 is time-sensitive, while the EITs describing future program scheduling are primarily informational and time-insensitive in nature.

Yet another difference is found in the location of the boundary drawn between program content and program metadata. EIT-0 provides closed captioning information, which can arguably be considered program content rather than metadata. While it is possible to carry such information in a program cue, in an IP-based system one might instead send this information as a separate RTP-based text stream.

While program *insertion* via directed channel change is facilitated by PSIP, the sort of individualized advertisement insertions we have considered with program cues have yet to be put in practice in a DTV setting. In principle, cues can be extended to signal time-sensitive information well beyond the narrow scope of program content, for example, in applications yet to be considered. Of course, PSIP is extensible as well; extensions have been added in ATSC Standard A/90 to provide for the announcement of emerging data services [3]. In particular, this standard defines a new *Data Event Table* (DET) to announce the data component of a service; this table serves a function similar to the EIT in announcing audio/video program content. In addition, a new *Long Term Service Table* (LTST) has been defined to selectively announce data programs to begin far in the future.

Recognizing a potential need, ATSC has also recently published ATSC Standard A/93, which specifies a mechanism for generic application triggers [4]. The principal need envisioned for triggers are those applications that preload bulk data, and subsequently

signal its activation to provide a synchronization of data and video.

V. Conclusion

We have proposed a system architecture capable of providing customized video program insertions to groups of receivers based on their specified interests. Our implementation of a program inserter demonstrates that seamless transitions are achievable in full-frame rate NTSC quality video streams transmitted over IP networks. We have also examined the similarities and differences between the cueing protocol suggested for use in IP-based distribution systems and comparable protocols that have been adopted in digital television settings.

We believe that the widespread adoption of broadcast video distribution over IP networks depends on the creation of value-added services that can not be easily matched with conventional broadcast networks. These services are likely to be implemented inside the network by intelligent nodes that manipulate video content as it flows through them. Though we have focused on a relatively straightforward program switching application, it appears that the real-time semantic manipulation of program content is now within our reach, representing a fertile area for new research by the media networking community.

VI. Acknowledgment

We are grateful to Professor Lawrence Rowe and Matthew Delco for the development of the RTPtv platform and their support with this project.

References

[1] Advanced Television Systems Committee, "A broadcasters' guide to PSIP," Oct. 2002.

[2] Advanced Television Systems Committee, "Program and system information protocol for terrestrial broadcast and cable (revision A) and amendment no. 1," *ATSC Standard A/65A*, 2000.

[3] Advanced Television Systems Committee, "Data broadcast standard," *ATSC Standard A/90*, 2000.

[4] Advanced Television Systems Committee, "Synchronized/asynchronous trigger standard," *ATSC Standard A/93*, 2002.

[5] J. Ayars, et al., "Synchronized multimedia integration language (SMIL 2.0)," *W3C recommendation*, `http://www.w3c.org/TR/smil20`, 2001.

[6] J. Brassil and H. Schulzrinne, "Structuring internet media streams with cueing protocols," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, Aug. 2002.

[7] CCIR, "Digital methods of transmitting television information," *Recommendation 601*, 1986.

[8] M. R. Delco, "Production quality internet television," *Berkeley Multimedia Research Center TR 2001-161*, Aug. 2001.

[9] T. Kim, *[online]*, `http://www.cc.gatech.edu/computing/Telecomm/people/Phd/tkim/cueing.html`, 2003.

[10] Linux Media Labs, *Product description*, `http://www.linuxmedialabs.com/lml33doc.html`.

[11] W.-Y. Ma, B. Shen, and J. Brassil, "Content services network: the architecture and protocols," *Proceedings of WCW 2001*, Boston, MA, Aug. 2001.

[12] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," *Internet Engineering Task Force, RFC 1889*, Jan. 1996.

[13] H. Schulzrinne, *RTP Toolset*, `http://www.cs.columbia.edu/IRT/software/rtptools`.

[14] Society of Cable Telecommunications Engineers, "ANSI/SCTE 33 2001 (DVS 253) digital program insertion messages for cable," `http://www.scte.org/documents/pdf/ ANSISCTE352001DVS253.pdf`, 2001.

[15] `http://www.terayon.com`.