Scalable Video Streaming over ALC (SVSoA): a Solution for the Large Scale Multicast Distribution of Videos

Christoph NEUMANN Vincent ROCA
INRIA Rhône-Alpes, Planète Research Team [1]
655 Avenue de l'Europe, 38334 Saint Ismier CEDEX
France
{firstname.name}@inrialpes.fr

Abstract **- This paper introduces a novel approach, SVSoA, for the streaming of hierarchically encoded videos using multicast IP. Its assets are: (1) it is massively scalable, (2) it is naturally TCP friendly, (3) it is immediately deployable (e.g. it does not rely on any QoS service), (4) it supports clients heterogeneity, (5) it is up to a certain point immune to long bursts of packet losses, and finally (6) it is compatible with any video hierarchical encoding scheme. Many of these features result from the intelligent use of the Asynchronous Layered Coding (ALC) reliable multicast protocol (RFC 3450) as the underlying transport protocol, whereas this latter was not designed for video streaming. SVSoA is well suited to the large scale distribution of videos or television programs over the Internet. Yet it is not suitable for interactive applications like video-conferencing because of the playing delay it induces. This paper introduces our proposal, the key parameters and the associated trade-offs. Two experiments carried out with a full featured implementation of SVSoA and a spatially encoded MPEG-4 video, confirm its benefits.**

Key-Words **- Video streaming, Reliable Multicast, Scalability, ALC**

1. Introduction

1.1 Target and Challenges

This work deals with the streaming of videos, that are either real-time or pre-encoded, to clients who receive and play information on-the-fly. It targets a massively scalable distribution, with potentially several millions of concurrent clients. Multicast-IP is therefore unavoidable. Because of the ubiquity of IP, this multicast routing infrastructure can take advantage of many different technologies on both the core network and the access network (satellites, terrestrial links, cable, DSL, etc.). In this work we merely assume the availability of a multicast routing service without making any assumption on its nature (source specific versus any source), nor on the nature of the underlying physical technology. We make no assumption on the target environment either, which can be either the Internet (if/when multicast routing is available), a site (e.g. a campus or an hotel), or a dedicated broadcasting network (e.g. a cable or DVB network). SVSoA can be used in all these situations.

Because of these assumptions, the client set is generally highly heterogeneous. This heterogeneity must therefore be considered to enable each client to receive the video stream that best fits with its networking and processing capabilities. [1] identifies three key aspects for the acceptance of video broadcasting: scalability, reliability, and quality of the reconstructed video. We fully agree with [1] and this paper emphasizes the reliability of transmissions in front of severe packet loss conditions and the stability of the video quality reconstructed.

1.2 Video Scalability

The advent of recent video codecs like MPEG-2, H.263+, MPEG-4, or H26L has largely improved the streaming possibilities, making it possible to optimize the video quality *over a given bit rate range* instead of *at a given bit rate*. The video scalability feature, also known as hierarchical video coding (both names will be used indifferently), refers to the possibility to see a video at several spatio-temporal resolutions by parsing appropriate portions of the bit-stream. In MPEG-2 and 4, three scalability techniques exist: *Temporal scalability*, *Spatial scalability* and *Qualitative (or SNR, Signal-to-Noise Ratio) scalability*, all three dividing the video in one base and one (rarely more) enhancement video layer. In all cases a partial reception of the enhancement layer will provide very little benefit [2]. On the contrary, with the Fine Granularity Scalability (FGS) [2] a partial reception of the (single) enhancement FGS layer provides an enhancement proportional to the number of bits decoded for each frame. The enhancement layer can therefore accommodate a wide range of bit-rates and offers the possibility to continuously adapt to the available networking bandwidth. But there is a cost and for the same transmission bandwidth, the quality is higher with non-scalable video coding. [3] discusses the Multiple Description (MD) video coding scheme that produces multiple independent layers of the video stream, each of the same importance. This property highly improves robustness, especially if the two layers follow different paths.

This discussion highlights several points: (1) video scalability is a complex feature that often produces a *single* enhancement layer; (2) splitting artificially this enhancement layer, or receiving only a subset of the associated data, does

[1]This work is supported by a research contract with STMicroelectronics since january 2004.

not always produce the expected result; (3) an exception is the MPEG-4 FGS scalability since the enhancement layer can be split into an arbitrary number of sub layers or can be partially received, at the cost of higher complexity. We will see that our proposal is compatible with any scalability approach and does not assume the presence of a fine grained hierarchical encoding.

1.3 Layered and Single Layer Streaming Approaches

Many approaches exist for video streaming [3]:

1.3.1 Layered Streaming Approaches

The streaming of scalable videos fits well with a transmission in cumulative quality layers. A traditional solution consists in mapping these video layers onto several multicast groups. In order to perform congestion control, each receiver dynamically adapts the number of layers received according to the experienced losses [4]. To behave correctly, this solution requires a fine video layer granularity, and a temporal scalability scheme is almost always assumed. Another requirement is that packets sent on the base layer experience no losses, because such losses usually trigger an important distortion in the reconstructed video (inter-layer dependencies). One solution to provide this transmission discrepancy is to protect data sent on the base layer with FEC (Forward Error Correction) techniques [5]. Another solution is to rely on a QoS differentiation mechanism within the network (Int-Serv or Diff-Serv), and to affect packets of the base layer to a prioritized service [6] [7]. This is a major limitation since a QoS service must be deployed between the source and each potential client.

In [7] the authors introduce two Source-Adaptive Multilayered Multicast (SAMM) algorithms that adjust the number of layers and their bit-rate depending on feedback information sent either by network elements and/or by receivers. This approach has major practical limitations since both variants require special features for the routers (priority drop preference, flow isolation, and congestion notification with network-based SAMM).

1.3.2 Single-Layer Streaming Approaches

Another solution consists in having a single video stream, mapped onto a single multicast group [8]. In that case the source adapts the transmission rate (e.g. by changing the video coding) according to RTCP feedback messages that give an indication on the experienced packet loss rate at receivers. Even if RTCP packets are rate-controlled (e.g. not to exceed 5% of the total session bit rate), this solution is not massively scalable. Besides, this solution is single-rate and consequently does not take into account the client heterogeneity.

A variant, called Simulcast in [2] and Destination Set Grouping (DSG) in [9], consists in generating multiple bitstreams of different bit-rates. Each client chooses the most adequate video bitstream according to its networking and processing capabilities. Switching to another bitstream dynamically is also possible. This solution addresses client heterogeneity but requires to decide, at coding time, for a fixed total bit rate, how many bitstreams should be generated and their bit-rate (DSG uses feedback messages for that).

1.3.3 Digital Fountain's Approach

Since this company proposes a streaming product that at first glance looks close to our proposal, we shortly discuss it here. Since no research report or paper has been published, this presentation is based on their web site [10]. The video stream is partitioned into blocks. The length of the blocks may vary depending on the application or the network features, yet it remains short, around 100 ms. Each block is protected with their proprietary FEC code, and the protection amount chosen depends on the network properties. Using an FEC code allows receivers to reconstruct a block if it received an amount of data equal in length to the original block. This approach is not designed for massive scalability, video scalability, nor receivers heterogeneity (and nothing is said about congestion control).

Our proposal completely departs from the layered or single layer approaches. It vaguely resembles to that of Digital Fountain, but neither the application area nor the techniques are the same as ours. In fact SVSoA largely relies on ALC.

1.4 ALC and Layered Congestion Control Protocols

The Asynchronous Layered Coding (ALC) protocol (RFC 3450) [11] of the IETF RMT working group is a layered reliable multicast protocol. Each receiver chooses how many layers to receive, depending on the bandwidth of its individual access network and on competing traffic. This receiver-driven decision is taken by an associated TCP-friendly layered congestion control protocol (e.g. RLC [12], FLID-SL/DL [13], or WEBRC [14]). Transmissions take place on the session layers either at some fixed predefined bit-rate (RLC, FLID-SL) or using a cyclic, dynamically changing bit-rate (FLID-DL, WEBRC), depending on the associated congestion control protocol. Since neither ALC nor the congestion control protocol use any feedback to the sender, this solution is massively scalable in terms of number of receivers.

ALC is well suited to the transmission of popular content in an "on-demand" mode, where clients join an ALC session, retrieve data, and leave at their own discretion. This is made possible by the large use of FEC (Forward Error Correction) encoding [5], and by the transmission of all the packets (data and FEC) in a random order and continuously on the various ALC layers [15] [16]. This "on-demand" mode is very specific to ALC and other reliable multicast approaches (e.g. NORM) are limited to a "push" synchronous model where all clients are supposed to be ready before the transmission starts. We will see that our approach relies on this "on-demand" model and is intrinsically linked to ALC.

ALC was originally designed for the massively scalable and reliable content delivery, and media streaming was never mentioned to be a possible application. Streaming is not the original intent of ALC. We however show in this paper that when used properly, ALC is suitable for media streaming.

The remainder of this paper is organized as follows: Section 2 introduces the general ideas. Section 3 explains how to initialize the various parameters and what are the associated trade-offs. Section 4 introduces some experimental results obtained on a local testbed with a full implementation of our proposal and of the ALC/RLC protocols. Finally Section 5 concludes the paper.

2. Scalable Video Streaming over ALC

2.1 Principles

The SVSoA approach relies on ALC/UDP/IP as the transport/network layers, and is placed beneath RTP and the server or player application.

2.1.1 Sender behavior

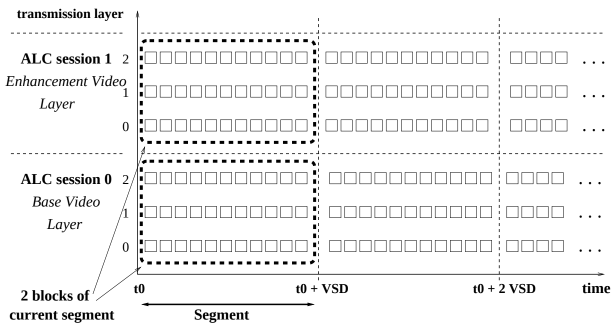


Fig. 1: Sender behavior.

Let's consider a server who needs to stream a hierarchically encoded video. The video consists of a base layer plus one (rarely more) enhancement layer (SVSoA does not require a fine granularity!).

The sender first partitions the video stream into *segments* of approximately the same duration, VSD (Video Segment Duration). By default $VSD = 60$ seconds, but other values are possible (section 3). Each video layer produces a *block*, of duration VSD. Each block is then sent independently on a *distinct ALC session* and thus on a different set of multicast groups as shown in figure 1 (note that video layers and ALC layers are two different concepts). After VSD seconds, the server automatically switches to the next segment, and for each ALC session, the transmission of block n is stopped and replaced by block $n + 1$.

During each period, the packets of a block are not sent sequentially but in a random order and cyclically, in order to offer an "on-demand" delivery mode. FEC packets included by ALC in the data stream enable receivers to efficiently reconstruct missing packets (either lost or not-yet received). This on-demand mode, specific to ALC, is required since receivers do not necessarily join at the beginning of a block transmission, especially on ALC session 1.

2.1.2 Receiver behavior

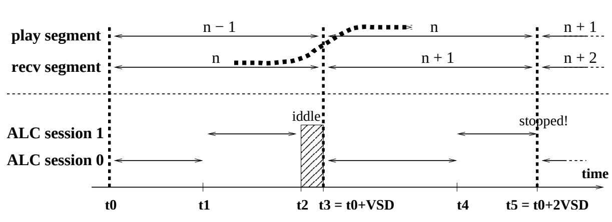


Fig. 2: Receiver behavior.

At the beginning of a period or when a new receiver joins the SVSoA session, a receiver first subscribes to the ALC session 0 where the base video layer of the current segment, n, is sent (from t_0 to t_1 in figure 2). When this block is successfully received, the receiver subscribes to the ALC session of the next enhancement video layer (from t_1 to t_2). This process stops (1) when all blocks have been successfully received, if ever (e.g. at time t_2), or (2) when the transmission of the next segment, $n + 1$, begins (e.g. at time t_5). When transmissions for segment $n + 1$ start, the receiver plays the video of segment n and switches to the first ALC session, and so on. Therefore a receiver always plays the previous video segment while receiving the current one, which of course introduces a playing latency of VSD seconds.

Receivers must of course be synchronized on segment boundaries. This is made possible by dedicated control packets sent regularly by the source [17].

2.1.3 Definition of ALC Objects According to the Video Scalability Scheme

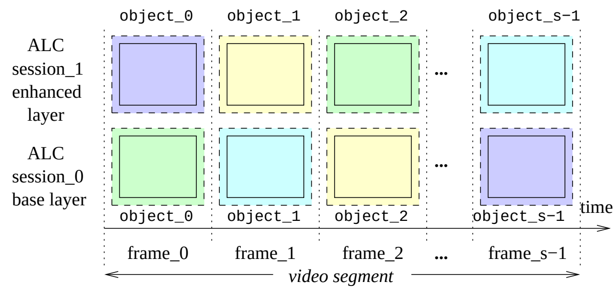


Fig. 3: Examples of object framing with Spatial Scalability.

The global reception efficiency is largely affected by the definition of ALC objects. The Application Level Framing (ALF) paradigm [18] tells us that each object should be autonomous and contain enough information to be processed by a receiver independently of other objects. We applied this principle and identified several framing possibilities. figure 3 shows a possible framing in case of non-FGS spatial scalability. Since a receiver cannot take any benefit from receiving a subset of an enhancement frame, each frame is carried as a distinct ALC object. Other framings are discussed in the extended version [17]. Anyway the idea is always the same: keep objects independent so that they can be processed even if they are not all received within a segment period.

2.2 The Benefits...

The benefits of this approach are numerous, many of them being derived from the use of ALC as the underlying transport protocol.

Massively Scalable

Because no feedback of any kind is used at either ALC, the layered congestion control protocol, or SVSoA, it makes no difference to the video server whether there are very few or several millions of simultaneous clients. Note that RTCP, the control protocol associated to RTP, is not used at all by SVSoA.

Exploits the Intra-ALC Congestion Control

Our approach takes advantage of the TCP-friendly layered congestion control protocol used within each ALC session. SVSoA automatically benefits from the latests developments in TCP-friendly congestion control protocols. This is a major asset since a recent protocol like WEBRC proved to be highly effective: receivers quickly reach the equilibrium point, achieve a good TCP-friendliness and are not affected by the IGMP leave latency issue [14]. The available bandwidth is always used optimally.

Immediately Deployable Anywhere

The SVSoA approach does not rely on any privileged transmission service nor on any specific feature within the backbone and can therefore be immediately deployed anywhere. This is made possible by the fact that a client receives only one video layer at a time, starting by the most important one (base layer). This solution therefore maximizes the probability of receiving the most important data correctly. On the opposite, traditional layered streaming approaches often rely on the presence of a QoS mechanism within the core backbone. This is in practice a major limitation which adds much complexity to the solution and restricts its use.

Addresses the Heterogeneity of Clients

Because ALC addresses the heterogeneity of clients, each SVSoA client receives the amount of video data made possible by its access network, independently of other clients. Besides, all clients are guaranteed to receive a minimum video quality before trying to receive any enhancement information.

Robust in Front of Packet Loss Bursts

Because ALC is a reliable protocol, packet loss bursts are easily recovered, even in case of long lasting bursts (e.g. several tens of seconds), without any brutal impact on the video quality perception. The only requirement is that enough time is left to enable a receiver to receive at least the base video layer during the segment duration. This is in line with [1] that shows that long bursts of packet losses (several seconds) occur quite often in large local area networks.

Independent from the Video Scalability Scheme

Another benefit of SVSoA is that the congestion control efficiency does not depend on the number of the enhancement layers provided by the scalable video codec, and the nature of scalability used by this codec. On the opposite, traditional approaches that rely on a direct mapping between video layers and transmission layers (multicast groups) assume the presence of a fine granularity video encoding. In practice this granularity is usually very low (e.g. the MPEG-4 ISO reference codec we used produces a single enhancement layer).

A Simple End-to-End Solution

Our proposal follows an end-to-end approach. It is a great asset compared to solutions like Content Delivery Networks (CDN) that can also be used for streaming. In that case, a complex distributed infrastructure must be designed, deployed and managed. Besides, video contents are replicated at several locations within the network. No such things are required with an end-to-end solution.

2.3 . . . And the Price to Pay

A High Playout Latency

When a new client joins an ongoing SVSoA session, this client experiences an *initial join latency*:
$BLRT \leq initial\_join\_latency < VSD{+}BLRT < 2{*}VSD$, where $BLRT$ (Base Layer Receive Time) is the minimum time required to get the whole base video layer. The minimum join latency is experienced when the client joins the session $BLRT+\epsilon$ seconds ($\epsilon \ll 1$) before the end of the current segment since he has enough time to get the whole base video layer and can display it immediately after switching to the following VSD period. There is no enhancement layer during this first VSD period but the most important information is displayed. The worst join latency is experienced when the client joins the session $BLRT - \epsilon$ seconds before the end of the current segment, since he needs to wait an additional VSD period.

The second source of latency is the *playing delay*: the video playout is always delayed by the segment duration parameter, VSD (typically 60 seconds, section 3.6). This feature prevents using SVSoA when interactivity (e.g. with tele-teaching) or immediate delivery (e.g. for a sport event coverage) are required.

Additional Traffic

Another drawback is a high cumulative transmission rate at the source, since all layers for all ALC sessions are active. Yet multicast routing limits the traffic carried on the backbone by avoiding transmissions on branches that do not lead to a receiver. In practice, if there is no receiver, the first hop multicast router prunes the traffic totally. Traffic is then restricted to the source's which is rarely an issue.

Let's now consider a client. ALC introduces several inefficiencies: (1) data and FEC packets are of finite number and duplications occur (e.g. the same packet can be received on two different layers at different times, or a packet for an already decoded block can be received later on); (2) "non-systematic" large block FEC codes like LDGM [19] or Tornado [15] also have intrinsic decoding inefficiencies. Some additional traffic will therefore be received, which is unavoidable with a reliable multicast protocol. In section 4.2.1 we show that even in a rate limited environment, SVSoA behaves efficiently and the extra traffic, in fact, enables clients to recover losses.

3. Analysis of the SVSoA Parameters

When deploying our solution some parameters must be adjusted to the target environment (e.g. is it deployed in a closed environment or over the Internet), and to the video features (e.g. the bit-rates of the base and enhancement video layers). In this section we explain how to optimally initialize two key parameters:

- the video segment duration (VSD), and
- the transmission rate on the base layer of each ALC session, b_0.

Several contradictory aspects must be considered when choosing a value for VSD. The first idea is to have a very short VSD in order to reduce the initial join latency and playing delay. But several considerations are against short video segment durations. We now analyze all of these aspects.

3.1 Storage Requirements

The first limitation we may thought is the required storage capacity at a server or at a receiver. Because of room limitation, we do not detail it here. Yet in [17] we show that a high quality video (2 Mbps encoding rate) requires only 29.3 MB of storage capacity with $VSD = 60$

seconds which is fairly reasonable. Similarly storage requirements at a receiver are low. This is all the more true with lightweight hosts (PDAs or smartphones) because of the limited bandwidth of the access network and/or the limited display capabilities (e.g. 3.8 MB are needed with a 256 kbps video). *We therefore consider that storage at a sender or receiver is not a problem.*

3.2 Impacts of the IGMP Leave Latency

The IGMP leave latency, i.e. the delay between when the last receiver of a LAN leaves a multicast group and its effect, is of importance. This latency is usually 3 seconds but can be higher depending on the IGMP implementation. In addition to this delay, the multicast routing protocol itself can add its own pruning delay. Let $igmp\_leave\_lat$ be the sum of these latencies. The IGMP leave latency is known to affect the behavior of a layered congestion control protocol like RLC or FLID-SL. The only exceptions are the FLID-DL and WEBRC protocols who counteract this latency thanks to a dynamic layering approach [13] [14].

Even with FLID-DL or WEBRC, our approach is still affected by the IGMP leave latency whenever a receiver changes of ALC session, for instance to receive the enhancement layer, or when switching to a new video segment. During $igmp\_leave\_lat$ seconds, packets of the previous ALC session still flow up to the receiver's LAN, thereby preventing a normal behavior of the new ALC session. The impacts of this latency are given by equation 1:

$$igmp\_ineff\_ratio = \begin{cases} (vlay\_nb * igmp\_leave\_lat)/VSD & \text{if } vlay\_nb > 1 \\ 0 & \text{if } vlay\_nb = 1 \end{cases} \quad (1)$$

where $vlay\_nb$ is the total number of video layers. The higher this inefficiency ratio, the higher the percentage of time wasted because of the IGMP and multicast routing protocol latencies. In our case ($igmp\_leave\_lat = 3$ seconds and $vlay\_nb = 2$), $igmp\_ineff\_ratio = 10\%$ with $VSD = 60$ seconds.

The IGMP leave latency largely impacts the solution efficiency and using a video segment duration of 60 *seconds is only possible with two video layers* (common case). Having a higher number of video layers requires to increase the VSD parameter.

3.3 Impacts of the Congestion Control Protocol during the Startup Phase

Another aspect to consider is the *congestion control protocol behavior during the startup phase for a given ALC session.* Because of this protocol, the reception rate at a client progressively increases until it reaches a "fair share" of the available bandwidth between the source and the client (the exact fairness definition depends on the protocol used). The time required to reach the steady rate is not so small compared to the VSD parameter and therefore it must be considered. This problem affects a client each time he joins a new ALC session, to receive the enhancement video block or when switching to the following video segment.

With RLC, in [17] we show that the amount of data received through a single ALC session during the startup phase, at time $t = i * t_0$, multiple of the RLC's time slot period t_0, is:

$$Rx(t = i * t_0) = \frac{i(i+2)b_0t_0}{3} \quad (2)$$

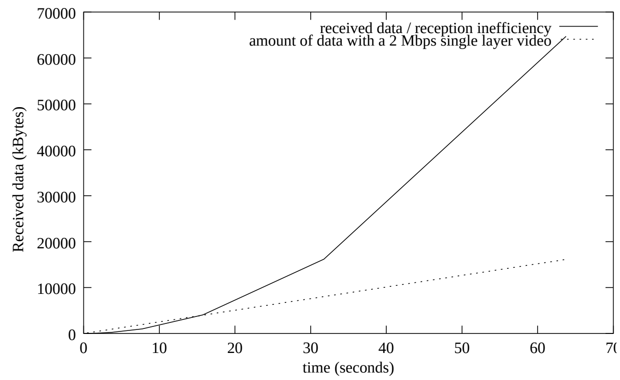


Fig. 4: Amount of data received in the startup phase of RLC.

b_0 being the transmission rate of the base RLC layer. In practice ALC introduces some inefficiency (section 2.3). To take it into account, we introduce a *global reception inefficiency ratio*, assumed constant:

$$rx\_ineff = \frac{nb\ received\ pkts}{nb\ usefull\ pkts} \geq 1$$

The minimum time required to entirely receive one video block, of enc_rate encoding rate, with the associated ALC session, is the minimum solution i of equation:

$$\frac{Rx(i)}{rx\_ineff} \geq enc\_rate * i * t_0$$

Figure 4 shows the $Rx(i)/rx\_ineff$ curve and the amount of video data curve as a function of time, when: $b_0 =$ 160 kbps, $t_0 = 0.25$ sec, $rx\_ineff = 1.66$ [16], and $enc\_rate =$ 2 Mbps. The minimum duration of a segment VSD_{min} is the intersection of the two curves. We find: $VSD_{min} \approx 63 * t_0 = 15.75$ sec. With two video layers, each encoded at 2 Mbps, we have to double VSD_{min} (31.5 sec).

A detailed analysis and the same calculations for the FLID-SL and WEBRC congestion control protocols can be found in [17].

3.4 Packet Loss Recovery Capabilities

The VSD parameter has a direct impact on the packet loss recovery capabilities. Losses in the Internet usually occur in bursts, because of router congestion problems or routing instability. Even in large local area networks, long bursts of packet losses (of several seconds) are more common than one would expect [1]. Our proposal easily copes with this kind of losses. Thanks to ALC's reliability mechanisms (i.e. the large use of FEC), these losses can usually be recovered, at least for the base layer which contains the most valuable video information. Intuitively, the longer the video segment duration (VSD), the greater the immunity to losses, and the longer the loss burst that can be recovered. In this section we analyze the SVSoA robustness assuming that a single burst, of duration $loss\_dur$, occurs during a video segment.

The goal is to *have an idea on how to initialize the VSD parameter to obtain a certain target robustness*, and what are the other parameters that affect this robustness. The simplification made (single loss burst) does not catch the SVSoA behavior in front of other loss models (e.g. with random isolated losses, or in case of several small loss bursts rather than a single long burst). Yet our scheme also brings some robustness in front of other loss models even if it is not considered in the present analysis. By default, we only consider the base video layer in this analysis, and no guaranty is given for the enhancement layer(s).

Figure 5 illustrates the robustness problem when $VSD = 60$ seconds and with a video encoding rate $enc\_rate = 2$ Mbps. Let t_{min} be the time required to receive the amount of video data sent during VSD seconds:

$$\frac{Rx(t_{min})}{rx\_ineff} = VSD * enc\_rate$$

In that case the maximum loss duration is the extra time available at the end of the video segment: $VSD - t_{min}$, and we find (graphically) a value of ≈ 32 seconds.

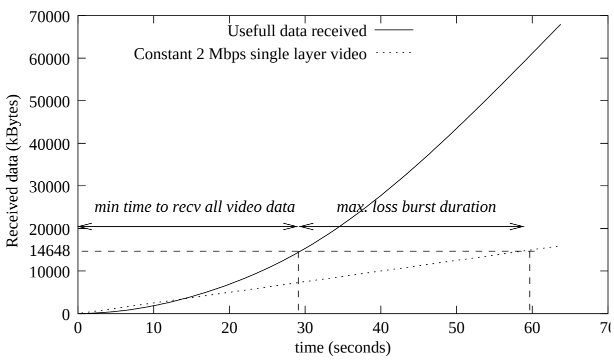


Fig. 5: Maximum recoverable loss burst length; here the loss burst occurs at the end of the video segment period (60 s).

This is in fact an upper bound and the robustness is largely impacted by the position of the loss burst in the video segment. The maximum recoverable loss period is indeed reduced when the loss period starts in the middle of the video segment, because of the congestion control algorithm which slows down the reception rate after a loss. Depending on the length of the burst, the congestion control algorithm restarts reception at a subscription level j smaller than the subscription level i before the start of the burst: $0 \leq j \leq i - 1$. The worst case, when all layers are dropped ($j = 0$), is illustrated in figure 6. The maximum recoverable burst length is then only 17 seconds which is now a lower, pessimistic, bound.

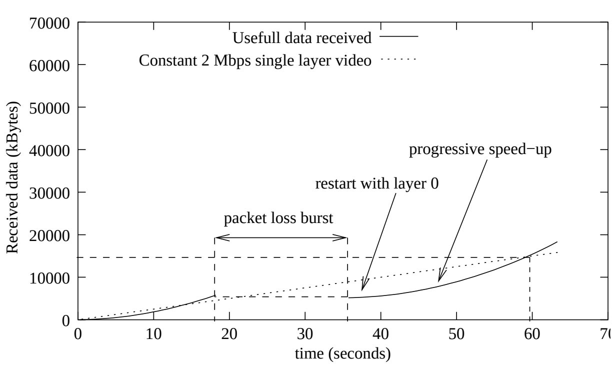


Fig. 6: Impacts of a loss burst in the middle of the video segment period (60 s).

Let t_1 and t_2 be respectively the time before and after the loss burst, of duration t_{loss}. In [17] we show that, the minimum recovery capability is reached at $t_1 = \sqrt{\frac{enc\_rate*VSD*b_0}{6*t_0*rx\_ineff}}$:

$$t_{loss\_min}(VSD) = VSD - \sqrt{\frac{2 * enc\_rate * VSD * b_0}{3 * t_0 * rx\_ineff}} \quad (3)$$

Figure 7 illustrates the maximum recoverable burst length as a function of VSD and t_1. It uses the same parameters as in section 3.3. We find: $t_{loss\_min}(60s) = 16.70$ sec. These curves confirm the high importance of the position of the loss burst in the video segment, and that of the VSD parameter.

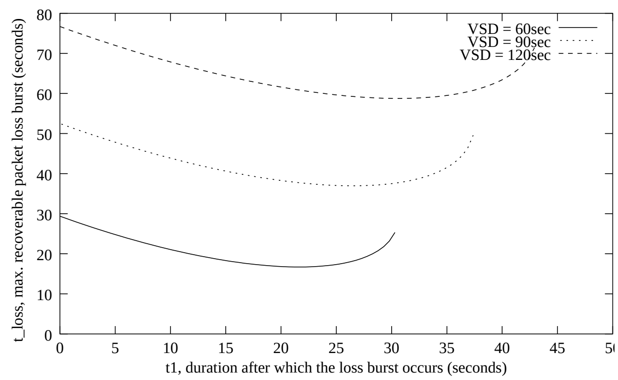


Fig. 7: Maximum recoverable packet loss burst length when using RLC.

A detailed analysis and the same calculations for the FLID-SL and WEBRC congestion control protocols can be found in [17].

3.5 Transmission Rate of the ALC Session

Another important parameter is b_0, the transmission rate of the base ALC layer. A higher b_0 leads to a faster reception in the startup phase. But a high b_0 also limits the possibilities to serve low end receivers. Therefore the following aspects must be considered:

- Required reception time in front of the video encoding rate: The higher the video encoding rate, the larger b_0 should be.

- The target environment (closed network, Internet,...): This target environment defines the possible heterogeneity of the users and their access networks. When serving Internet clients, b_0 should be compliant with the slowest possible client.

3.6 Initializing SVSoA in Practice: a Summary

In practice SVSoA can be correctly initialized according to the following steps:

step–1: Retrieve the average encoding rate enc_rate and number of layers of the video, $vlay\_nb$ (often 2).

step–2: If $vlay\_nb \geq 2$, then use $VSD = 30 * vlay\_nb$ to keep the $igmp\_ineff\_ratio$ constant and equal to 10%. If $vlay\_nb = 1$, then IGMP has no effect. Use $VSD = 60$ seconds for instance.

step–3: Retrieve the t_0 parameter of RLC. Estimate the global reception inefficiency of ALC and of the FEC code (e.g. we use 1.66).

step–4: Define the transmission rate on the base ALC layer, b_0. This choice depends on the target environment since this is the minimum reception rate.

step–5: Calculate the minimum loss burst immunity for the base video layer, $t_{loss\_min}(VSD)$. With RLC use equation 3. If this value is judged too low, it means that the video bit-rate is too high compared to the transmission rate. So increase the b_0 value and reiterate at step 4. Alternatively, set a higher VSD and reiterate at step 5.

Many parameters are only approximations. This is not a major issue yet since several equations of previous sections (e.g. $t_{loss\_min}(VSD)$) are for the worst case and rely on pessimistic assumptions (e.g. the client drops all the layers after the loss burst).

4. Experimental Results

4.1 Experimental Conditions

4.1.1 The Streaming Approaches Compared

We implemented our proposal using our MCL library that implements the ALC/RLC protocols [20], and the MPEG4IP [21] MPEG-4/RTP application.

We compare the SVSoA performances with that of a "classical" FEC based streaming scheme inspired from [1]. It uses a *single* ALC session, each video layer is mapped onto a distinct ALC layer, and it uses RLC for congestion control, like SVSoA. The sender sends each frame sequentially. Reed-Solomon FEC protection is used for the base video layer, using the same parameters as in [1] (30 FEC packets are added to each block of 225 source packets). The enhancement video layer is sent without FEC protection, since it has less importance. Adding proactive FEC undoubtedly consumes additional bandwidth, but it enables a fair comparison with SVSoA which heavily relies on FEC. Interested readers can refer to [17] for additional comparisons with a classical streaming scheme which does not include any FEC protection. Results are globally similar to that obtained with the "classical" FEC based scheme.

4.1.2 Video Encoding and SVSoA Parameters

Our tests use a 120 second video, encoded off-line with a constant 25 fps frame-rate, and using a spatial scalability encoding which can only provide two video layers. The bit-rate of both layers is *on average* 667 kbps each (without FEC), but the instantaneous bit-rate fluctuates around this average.

SVSoA is initialized with $b_0 = 131$ kbps and $VSD = 60$ seconds. The theoretical minimum loss recovery capability is equal to $t_{loss\_min} = 21.0$ seconds.

4.1.3 Testbed and Scenarios

Our testbed (figure 8) is composed of four PC P-III/Linux attached to two different Ethernet subnets. The right subnet has a limited bandwidth of 10 Mbps. Host C in the middle is the multicast router and uses the `mrouted` multicast routing daemon. Host A is the streaming server and host D the client. Two scenarios are considered:

1. *Limited bandwidth* to evaluate both streaming approaches in presence of a concurrent network flow, in a limited bandwidth environment. During this test, host B sends a constant bit-rate 8.2 Mbps UDP flow to host D. Host D receives both the UDP flow from B and the video from A.

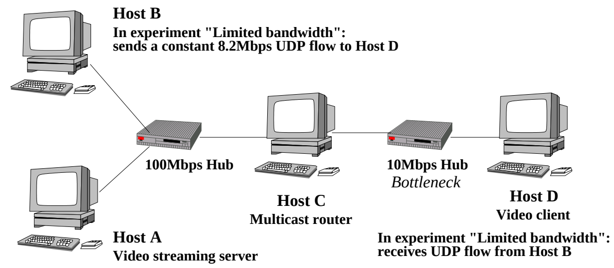


Fig. 8: Our testbed.

2. *Long burst of packet losses* to evaluate the error recovery capabilities of both approaches. 10 seconds after the start of the video transmission, we unplugged the network cable during 15 seconds, and reconnect it up to the end of the test. No background traffic is used in this test.

4.2 Experimental Results

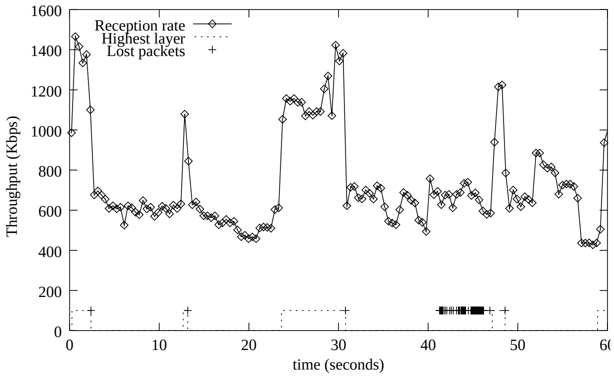
4.2.1 Behavior in a Limited Bandwidth Environment

With the classical approach, the transmission rate for each video layer is highly dependent on the instantaneous video bit-rate. A 255/225 Reed-Solomon FEC encoding operates on 1-2 seconds chunks of frames within the video. Therefore the transmission rate is constant for these short periods. When the cumulative rate on both layers exceeds the available bandwidth (e.g. after 3 seconds in figure 9 (a)), packets get lost. According to RLC, the client then drops a layer, which reduces the video quality, in order to adapt to the available network bandwidth. Some time after 12 seconds, the enhancement layer is once again added, which triggers losses, and the client drops the higher layer one more time. We see that the rough granularity provided by the spatial scalability encoding does not enable an efficient behavior. We also see that losses may affect both video layers (e.g after 30 seconds or after 43 seconds (figure 9 (b))). FEC protection on the base video is obviously not enough at these moments. Each loss on the base layer prevents the client to decode the associated frame on the higher layer if received which amplifies the phenomenon.

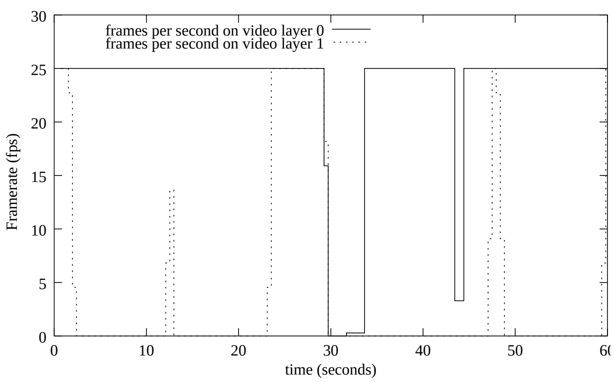
This is not the case with SVSoA where the reception rate is only determined by the congestion control protocol, independently of the video bit-rate (compare figures 9 (a) and 10 (a)). Transmissions are smoothed over the time which is an advantage from a networking point of view. Figure 10 (b) shows that the frame rate of the base layer is constant during the whole test, and a minimum video quality is provided to the client. However the frame rate of the enhancement layer is relatively low since the spare bandwidth is not sufficient to receive more frames.

4.2.2 Behavior With a Long Burst of Packet Losses

The reception for both approaches is totally interrupted while the cable is unplugged (between $t \approx 10s$ and $t \approx 25s$ in figures 11 (a) and 12 (a)). With the classical approach, the reception restarts with the base video layer only, the enhancement layer being only added after some time. With SVSoA the reception of the base layer continues after the

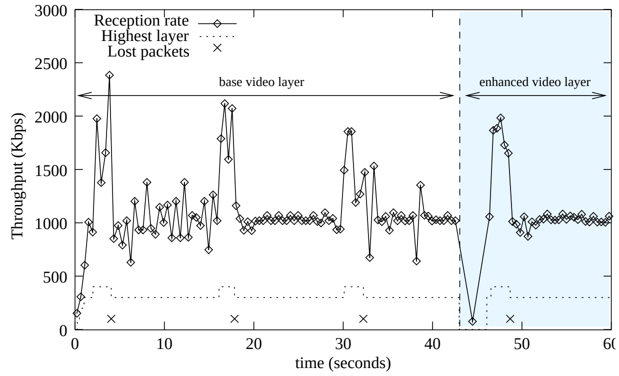


(a) Reception rate

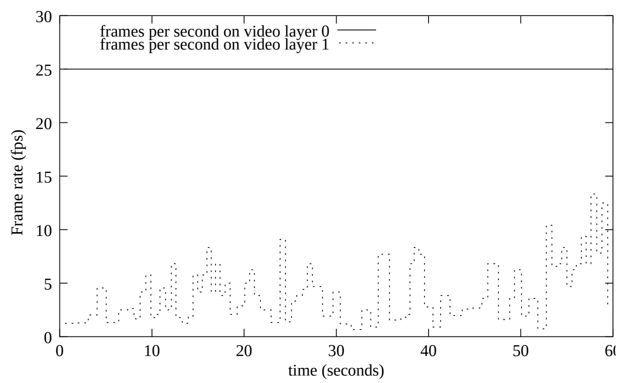


(b) Frame rate

Fig. 9: Tests with limited bandwidth and the classical FEC based approach.



(a) Reception rate



(b) Frame rate

Fig. 10: Tests with limited bandwidth and SVSoA.

burst, until completely received.

The frame rate of the resulting video at the client shows a big hole on both video layers with the classical approach (figure 11 (b)). No frame can be displayed while the cable stays unplugged. For the enhancement layer, this time is longer since the ALC layer is only added after an additional delay (RLC behavior). FEC protection cannot do much in front of such a long burst of losses.

With SVSoA the client receives the whole base video layer, thereby assuring a minimum video quality. The frame rate of the enhancement video layer is not optimal though because the client needs more time to reconstruct the base layer, which leaves less time for the enhancement layer. But the lost frames of the enhancement layer are spread equally over the whole segment, which provides a *globally constant video quality*.

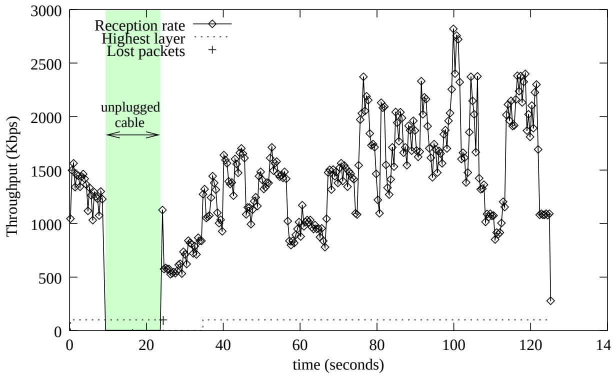
4.3 Discussions

In the two selected scenarios, SVSoA largely outperforms the classical FEC based streaming approach. This classical approach, which serves as a reference, is obviously not optimal, and could benefit from several improvements, like: (1) the use of a QoS service to protect the base layer, (2) the use of a finer video scalability granularity, and (3) a really constant bit-rate video encoding. Yet SVSoA assumes none of these features, which are three major, very restrictive assumptions.

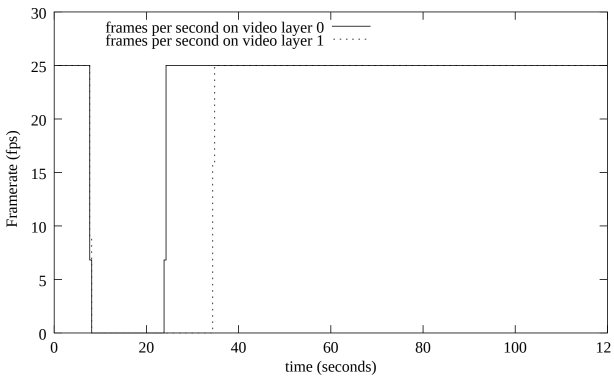
The proactive FEC protection used is not sufficient to reconstruct all losses of the base video layer. Adding more FEC is not the solution since it increases the consumed bandwidth, leading to more losses. All solutions requiring some feedback information from the receivers to adjust the amount of proactive FEC create scalability problems. Besides the ideal amount is not the same for all clients, all the time, and defining several parallel streams is then unavoidable. This is why we chose to use the FEC scheme proposed by [1].

The experiments show that SVSoA behaves well in a congested environment, no matter how many layers the video codec produces. Congestion control enables to adapt to the available bandwidth. Because of the natural buffering capability of SVSoA, packet losses, even in case of long bursts, do not automatically lead to frame losses and sudden video quality changes. They only increase the time spent receiving a given layer, and thus reduces the probability of receiving the enhancement information entirely. And because of the high randomness of transmissions introduced in an ALC session, the video quality is almost constant during each VSD period.

Even if the experiments carried out involve a single client, the SVSoA scalability is guaranteed by the total

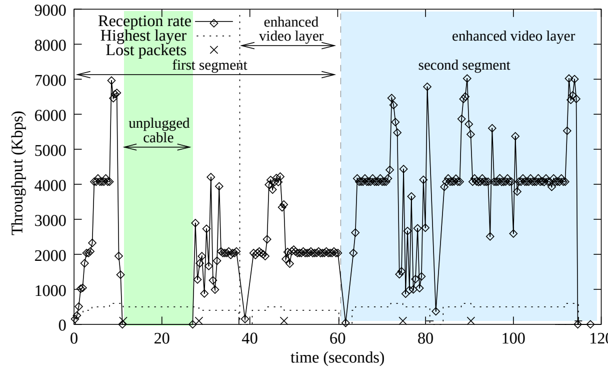


(a) Reception rate

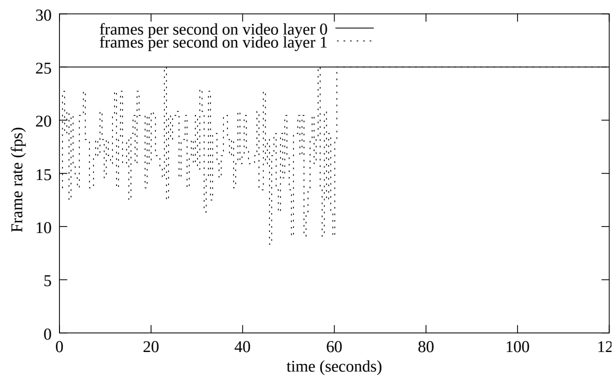


(b) Frame rate

Fig. 11: Tests with a 15 second loss burst and the classical FEC based approach.



(a) Reception rate



(b) Frame rate

Fig. 12: Tests with a 15 second loss burst and SVSoA.

absence of feedback packets. Scalability is only limited (!) by multicast routing protocols.

The SVSoA performances experienced are limited by some sub-optimal solutions in the ALC implementation used [17]. More recent congestion control protocols (WEBRC), more efficient large block FEC codes, and the meta-object LDPC encoding we introduced in [19] would largely improve the performances. This is left to future works.

5. Conclusions and Future Works

This paper introduces a novel multicast streaming solution, SVSoA, for the streaming of hierarchically encoded videos. Our solution is in fact mid-way between reliable multicast file transfer and streaming. It directly benefits from the ALC reliable multicast protocol assets in terms of unlimited scalability, congestion control, client heterogeneity support, and error recovery, although ALC was originally not intended for streaming. SVSoA limits the video quality instability by smoothing the effects of packet losses over periods of one minute, and it avoids video cuts, even in case of long bursts of packet losses. This feature is of utmost importance for the general acceptance of video streaming solutions by the end users. Finally, the solution is simple and can be deployed immediately since it does not assume any specific service within the network. Being totally end-to-end, making an intelligent use of well understood and standardized protocols (ALC and WEBRC are RFCs), and being capable of working with any video coding scheme, are additional assets.

The large latency (around one minute) introduced by SVSoA is the price to pay for these benefits. If it prevents its use for interactive applications like video-conferencing, or for the timely delivery of information (e.g. a sport event coverage), we are convinced that most other video content distribution schemes can benefit from it. And in such cases having a robust, stable, and scalable video streaming scheme remains the prime requirement.

Future works will consist in reducing the reception inefficiency of the underlying ALC implementation, in particular by adding meta-object large block FEC encoding. We plan to further experiment in operational environments and on wireless links where SVSoA's robustness may beneficially counteract the instability observed in such networks. Finally we will study complementary mechanisms to reduce the startup latency and the use of SVSoA for the streaming of non scalable videos.

References

[1] S.D. Servetto, R. Puri, J-P. Wagner, P. Scholtes, and Martin Vetterli, "Video multicast in (large) local

area networks," in *IEEE INFOCOM'02*, June 2002.

[2] W. Li, "Overview of fine granularity scalability in mpeg-4 video standard," vol. 11, no. 3, Mar. 2001, IEEE Transaction on Circuits and Systems for Video Technology.

[3] B. Li and J. Liu, "Multirate video multicast over the internet: an overview," vol. 17, no. 1, Jan. 2003, IEEE Network, "Multicasting: an enabling technology".

[4] J. O'Donnell M. Nilsson, D. Dalby, "Layered audiovisual coding for multicast distribution on ip networks," in *6th IEEE, European Workshop on Distributed Imaging*, Nov. 1999.

[5] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft, *The use of Forward Error Correction (FEC) in reliable multicast*, Dec. 2002, Request for Comments 3453.

[6] S. Bajaj, L. Breslau, and S. Shenker, "Uniform versus priority dropping for layered video," in *ACM SIGCOMM'98*, Sept. 1998.

[7] B. Vickers, C. Albuquerque, and T. Suda, "Source adaptive multi-layered multicast algorithms for real-time video distribution," vol. 8, no. 6, Dec. 2000, IEEE Transaction on Networking.

[8] J. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the internet," in *ACM SIGCOMM'94*, Oct. 1994.

[9] S. Cheung, M. Ammar, and X. Li, "On the use of destination set grouping to improve fariness in multicast video distribution," in *IEEE INFOCOM'96*, Mar. 1996.

[10] *Digital Fountain home page*, http://www.digitalfountain.com.

[11] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, and J. Crowcroft, *Asynchronous Layered Coding (ALC) protocol instantiation*, Dec. 2002, Request for Comment 3450.

[12] L. Vicisano, L. Rizzo, and J. Crowcroft, "Tcp-like congestion control for layered multicast data transfer," in *IEEE INFOCOM'98*, Feb. 1998.

[13] J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, and W. Shaver, "Flid-dl: Congestion control for layered multicast," in *2nd Workshop on Networked Group Communication (NGC2000)*, Nov. 2000.

[14] M. Luby, V. Goyal, S. Skaria, and G. Horn, "Wave and equation based rate control using multicast round trip time," in *ACM SIGCOMM'02*, Aug. 2002.

[15] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *ACM SIGCOMM'98*, Aug. 1998.

[16] V. Roca and B. Mordelet, "Improving the efficiency of a multicast file transfer tool based on alc," in *IEEE International Symposium on Computer Communications (ISCC'02)*, July 2002.

[17] C. Neumann and V. Roca, "Scalable video streaming over alc (svsoa): a solution for the large scale multicast distribution of videos," Research Report 4769, INRIA, Mar. 2003.

[18] D. Clark and D. Tennenhouse, "Architectural considerations for a new generation of protocols," in *IEEE SIGCOMM'90*, Sept. 1990.

[19] V. Roca, Z. Khallouf, and J. Laboure, "Design and evaluation of a low density generator matrix (ldgm) large block fec codec," in *5th International Workshop on Networked Group Communications (NGC'03)*, Sept. 2003.

[20] V. Roca and al., *MCLv3: an Open Source Implementation of the ALC and NORM Reliable Multicast Protocols*, URL: http://www.inrialpes.fr/planete/people/roca/mcl/.

[21] *MPEG4IP project home page*, http://sourceforge.net/projects/mpeg4ip/.