Vidyo®

SVC and Video Communications

Scalable Video Coding Technology for Real-Time Multi-party Video

Alex Eleftheriadis, Ph.D., Chief Scientist

May 2016

ABSTRACT: Scalable Video Coding has revolutionized real-time videoconferencing architectures by delivering high quality, low latency, and error resilient communications at high scale.

Table of Contents

Introduction 3
What is SVC? 3
What's the difference between AVC and SVC? 4
Why use SVC? 6
What is the VidyoRouter? 7
Isn't AVC more efficient than SVC? 9
Scalability and Next-Generation Codecs 10
Scalability, Simulcasting, and SFUs 10
Scalability, Simulcasting, and WebRTC 11
Why will SVC succeed? 12

Author's Bio 13

Resources 14
Vidyo 14

Introduction

Scalable Video Coding, or SVC, has had a significant impact in the videoconferencing industry, and video communications in general.   When Vidyo first introduced SVC back in 2008 very few people realized the transformative power that it contained. Most believed that it is just a better codec, possibly with some improved error robustness. SVC is a key component of Vidyo's VidyoRouter™ architecture and a crucial piece for providing the quality of experience that this system provides.

As with any new technology, it can be difficult in the beginning to fully understand how SVC works, what systems and functionalities it makes possible, and what implications it may have for the entire industry. This article is intended to provide some facts as well as insight into SVC, how Vidyo's VidyoRouter uses SVC in its various functions, as well as the role that scalability has played and will continue to play in the world of video communications.

What is SVC?

First of all, what is SVC? It's an extension of the H.264/MPEG-4 Part 10 Advanced Video Coding standard, often referred to as AVC. AVC has been jointly developed by the International Telecommunications Union's (ITU) Video Coding Experts Group (VCEG) and the International Standardization Organization's (ISO) Motion Pictures Experts Group (MPEG) through a joint group called Joint Video Team (JVT). As a result, AVC has two official names: H.264 from the ITU parent, and MPEG-4 Part 10 from the ISO parent. It appears that people coming from the communications space tend to refer to AVC as H.264, whereas people coming from the broadcast/entertainment space tend to refer to it as AVC or MPEG-4. AVC has been very successful and is the standard used in practically all modern digital video applications: from videoconferencing and YouTube, to Blu-ray DVDs and the iTunes store.

SVC is officially Annex G of the AVC specification. Some people use the term AVC to mean the H.264 specification together with Annex G. That's very confusing. In this article we use the term AVC for the non-scalable portion of H.264, SVC when the scalable part is used, and H.264 when referring to either indiscriminately.

When Vidyo joined the JVT group (in the summer of 2005), it was the only videoconferencing company interested in scalability. In fact, the SVC effort was, at the time, driven by companies interested in broadcast and mobile applications, as well as academics. Vidyo's engineering team worked very actively in the group to ensure that the design was appropriate for the needs of the videoconferencing industry: from providing some 18 technical contributions, creating and making available test video material, to constructing a large part of the conformance bitstreams, and co-editing the conformance specification.

The H.264 specification (in fact, most video coding standards) has a way to indicate which parts of the specification are used in a particular application space: the profiles. A profile is a subset of the coding tools offered by the specification, which are deemed appropriate for a particular application domain. For example, features that increase the end-to-end delay may be acceptable for broadcast video but not for videoconferencing and are thus not included in profiles geared towards videoconferencing. The scalable features of H.264 are contained in its scalable profiles: Scalable Baseline, Scalable High, Scalable Constrained Baseline, Scalable Constrained High, and Scalable High Intra. The profiles designed for videoconferencing applications (as well as, incidentally, mobile devices) are primarily "Scalable Baseline" and "Scalable Constrained Baseline", although the "High" profiles are routinely used in software implementations that operate at HD resolutions.

A related concept to a profile is that of a level. A level defines limits to various operating parameters within a particular profile. For example, it defines the largest picture size that a particular decoder may be able to handle. Profiles and levels are a fairly old concept: your plain old DVD player sports an MPEG-2 Main Profile at Main Level decoder. Your Blu-ray player has an H.264 AVC High Profile decoder at Level 4.1.

What's the difference between AVC and SVC?

The basic difference between SVC and AVC is that the former encodes the video signal as a set of layers. The various layers depend on each other, forming a hierarchy. A particular layer, together with the layers it depends upon, provides the information necessary to decode the video signal at a particular fidelity. Fidelity here concerns one or more of spatial resolution, temporal resolution, or signal-to-noise ratio (SNR)[1]. The lowest layer, i.e., the layer that does not depend on any other layer, is called the base layer and offers the lowest quality level. Each additional layer improves the quality of the signal in any one of the three dimensions (spatial, temporal, or SNR).

Figure 1 shows video coding in a non-scalable fashion. This is how most, if not all, AVC-compatible videoconferencing encoders operate. Each square is a picture, whereas the numbers in the bottom row indicate time instances. The first picture, which is of type "I" is coded without referring to any other picture (the I comes from the word "intra"). All other pictures are of type "P", indicating that they are coded using prediction from a previously coded picture. The arrow indicates the source picture of the prediction as well as the target picture. We observe that there is just a single layer, forming a never-ending chain of pictures.

[1] SNR is a measure of distortion of the compressed video signal vs. its uncompressed version. In SNR scalability, the extra information provided by the enhancement layer changes neither the spatial resolution nor the temporal one. It rather reduces the distortion of the compressed video signal – equivalently, it increases its SNR.

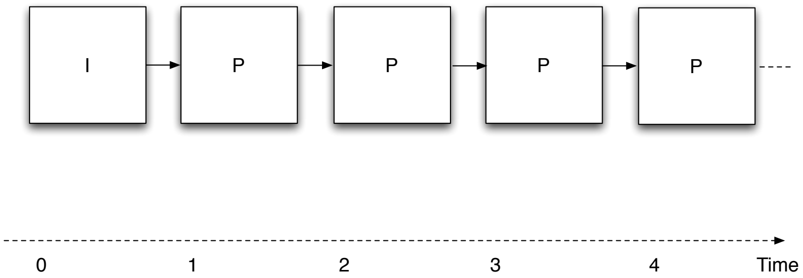

Figure 1: Non-scalable video encoding

Figure 2 shows scalable video coding where temporal scalability is used. Observe that the prediction arrows are organized so that, in this example, three different layers are defined (L0 through L2). The pictures in the diagram are offset vertically for visually separating the three layers. Each layer requires the lower layers in order to be decoded, but not any of the higher layers. This allows to remove pictures, starting from the top layers, without affecting the decodability of the remaining pictures. For example, let's assume that the pictures shown in the figure are displayed at 30 fps. If we drop all L2 pictures, the remaining ones can be decoded without any problem, and produce a video at 15 fps. If we further remove all L1 pictures, then again the remaining ones (L0) can be decoded and produce a video at 7.5 fps.

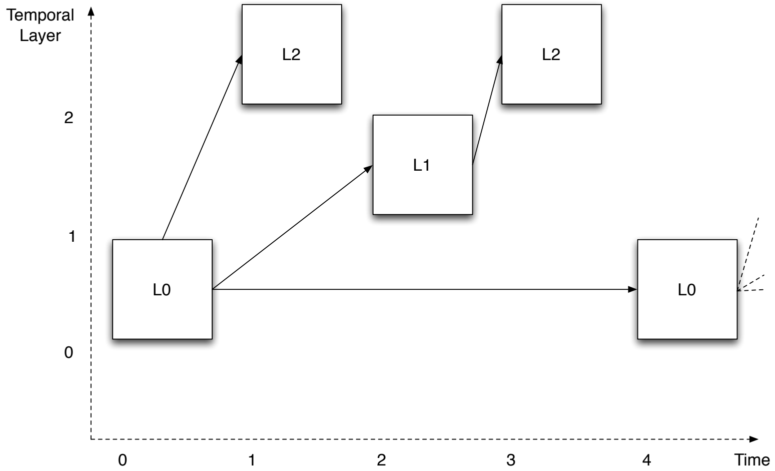

Figure 2: Temporal Scalability

We can expand this structure to include spatial scalability, as shown in Figure 3. Each picture now has two parts: the B part for the base layer resolution picture, and the S part for the spatial enhancement layer that allows us to produce a full resolution picture. The spatial enhancement can have either 2 or 1.5 times the resolution of the base in each of the horizontal and vertical directions. This allows spatial scalability between, e.g., VGA and QVGA (ratio of 2) as well as 1080p and 720p (ratio of 1.5). Spatial scalability can be combined with temporal scalability (and SNR) in a completely independent way. Assuming that the full rate and resolution in this example is HD (720p) at 30fps, then we can have any combination of HD and quarter HD resolutions as well as 30, 15, or 7.5 fps.

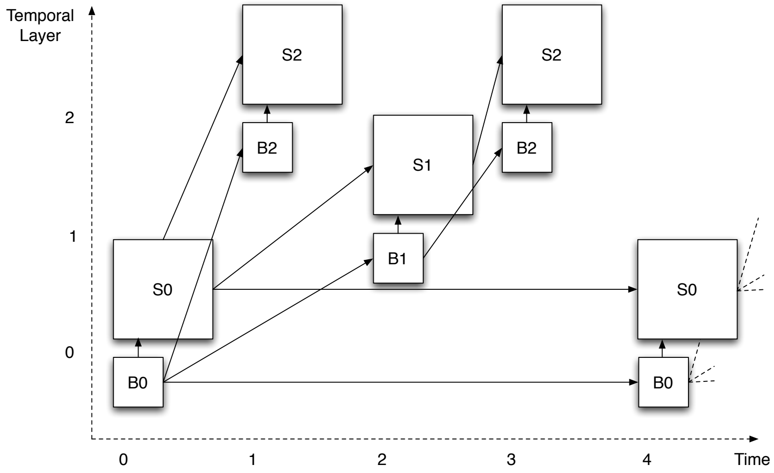

Figure 3: Spatial and temporal scalability

Note that these are not the only possible scalability structures – the standard provides considerable flexibility in structuring both spatial and temporal prediction structures.

A very important aspect of SVC is that the base layer conforms to AVC by design. In other words, the base layer of an SVC signal is decodable by an AVC decoder. This ensures that an SVC signal is backwards compatible with AVC, albeit at a lower level of fidelity than the full signal if more than one layer is involved. Note that an SVC encoder does not need to always produce layered bitstreams – if it operates in an environment where scalability is not required or desirable, then it can produce traditional AVC streams.

Why use SVC?

SVC provides for a representation of the video signal that enables easy adaptation. In other words, adaptation without having to decode, process, and re-encode the signal. If we want to change the picture resolution or temporal frame rate, then the only thing we need to do is eliminate the appropriate blocks from the diagrams in Fig. 1. Thinking of these blocks as data packets that are transported over a network, this translates to just eliminating the corresponding network packets from the transported bitstream. The implication of this is very significant: it suggests a re-engineering of the classical videoconferencing system architecture that is based on the Multipoint Control Unit, or MCU.

The MCU is a complex device that receives multiple encoded video signals, decodes them, composites them together on a new picture, re-encodes, and finally transmits the coded signal to the intended recipient. Clearly, this is a very complex operation from a computational point of view. In addition, the operation introduces considerable delay in the system, with 150-200 msec being typical. For comparison, the ITU mandates an upper limit of end-to-end delay for long-distance telephone of 180 msec. Beyond 180 msec the delay becomes disconcerting to the parties who are communicating. Clearly then, interactive multipoint communication with an MCU is very difficult. Finally, there is quality loss due to the cascaded encoding passes. Note that these problems are inherent in the architecture and do not go away regardless of how many resources one 'throws' at the problem. For example, improving the speed of the DSP processors inside an MCU will only marginally reduce the delays.

Use of SVC allows us to eliminate the MCU, and replace it with a much simpler device, what we call the VidyoRouter.

What is the VidyoRouter?

The VidyoRouter is an application-level router whose job is to receive scalable video (and audio) from each participant and selectively forward scalable layer packets to each of the other participants. This is shown in Fig. 4. Instead of having to perform complicated signal processing operations like the MCU, the VidyoRouter only has to examine packet headers and decide whether to forward a packet or not.

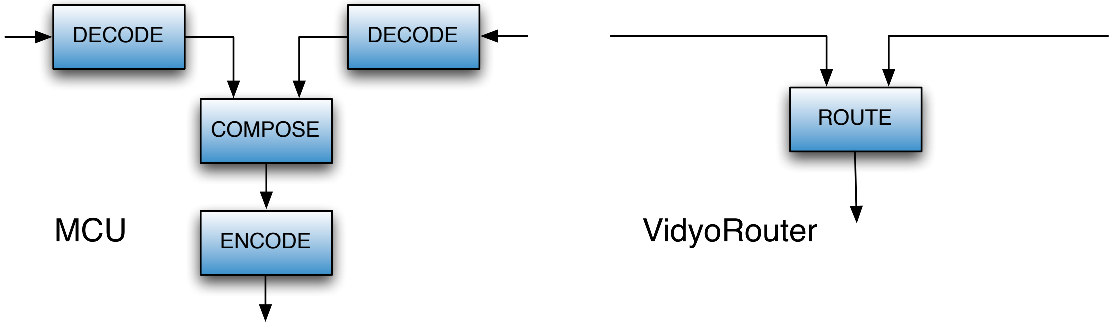

FIGURE 4: VidyoRouter vs. MCU

With the VidyoRouter architecture, high-end features such as rate matching and personalized layout become routing decisions. Fig. 5 depicts a VidyoRouter that receives a high-resolution signal from a transmitting endpoint and serves three different receiving endpoints, each with a different screen resolution and network bitrate availability. The VidyoRouter will only forward the packets that make sense for each particular receiving endpoint. In fact, these decisions can be made dynamically. Imagine, for example, that you resize a video window in your desktop to make it smaller – the VidyoRouter can then drop any high resolution enhancement layers that it may be sending if they do not make sense for the new window size. More importantly, these adaptation mechanisms can be used to handle dynamic variations in the available network bitrate.

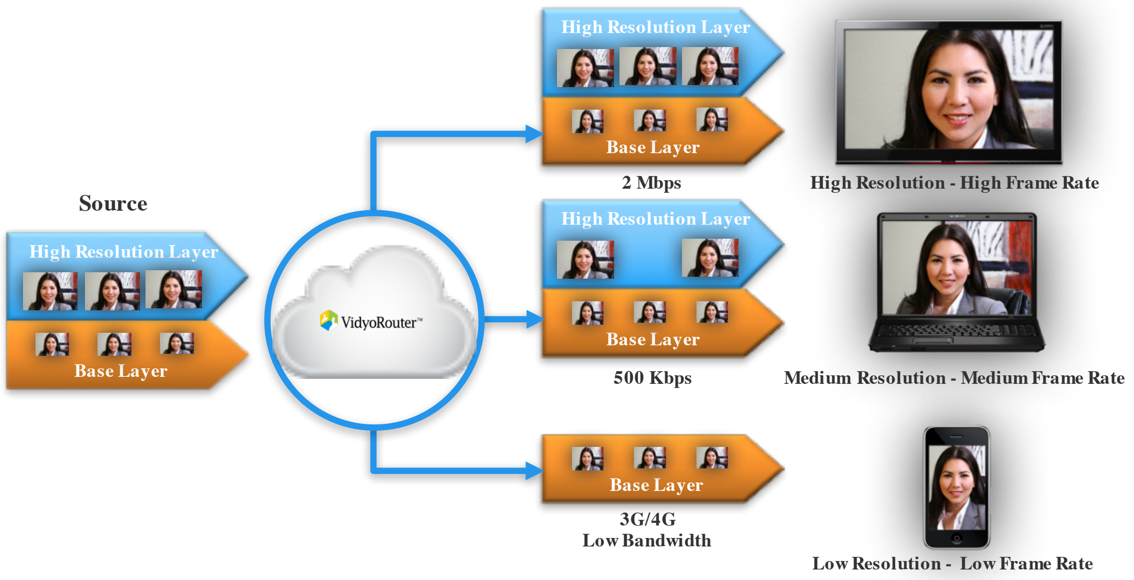

FIGURE 5: The VidyoRouter and Adaptive Video Layering

The fact that adaptation is so readily available in the scalable video signal is extremely important in its own right. There are, however, two key features that the VidyoRouter brings to video communication systems that make it both unique and uniquely exciting.

First, the VidyoRouter design introduces very little delay. As it does not perform any signal processing, the delay from its input to its output is very small, typically well below 20 msec. Comparing that with the 150-200 msec of a typical MCU, it becomes obvious that truly interactive, multi-point video communication now becomes feasible. There are few things as important for the quality of experience in both visual and audio communication as the end-to-end delay.

The second feature is error resilience. Use of SVC coupled with the VidyoRouter enable operation of a video communication system with packet losses as high as 20%. Traditional videoconferencing systems using AVC start breaking down at 2 to 3% packet loss rates. This difference is very significant because it allows SVC to operate with extremely high quality on the public Internet. Traditional systems based on AVC suffer from severe artifacts that are very annoying to end-users. The mechanisms for making this robustness possible are sophisticated, capitalize on the scalable nature of the video signal representation, and involve pro-active (forward error

Benefits of the VidyoRouter Architecture:

- Low delay
- Error Resilience
- High Capacity

correction – FEC) and re-active operations (retransmissions) by the endpoints and the VidyoRouter. Due to temporal scalability, these operations do not increase the end-to-end delay and hence the delay properties of the system are not affected by the presence of errors. This is a significant breakthrough in packet video systems research, in that a very difficult problem that persisted for more than 20 years has finally been solved.

But this is not all – the extremely low delay of the VidyoRouter makes it possible to use them in cascaded configurations. This was practically impossible with the traditional transcoding MCU architecture due to its very high delay. It is now possible to create sophisticated configurations that follow performance or administrative requirements.

This type of scalability (as opposed to coding scalability) is extremely important from a practical point of view. A single VidyoRouter today on a single socket 8-core server can support 100 HD connections. Growing this to thousands of connections, either in the same location or in a distributed fashion is straightforward. The benefits are even more significant for global, cloud-based deployments.

For the first time, the complexity of videoconferencing is brought down to the level of any other network application. For comparison, look at the architecture of web servers. Imagine the web if the page to be displayed on your browser was in fact rendered at the server, rather than in your browser. The server would have an extremely high computational load, and it would be impossible to support a large number of users. The web could never grow to what it is today. Web servers actually only retrieve files and send them to the user’s browser for composition and rendering on his/her computer. This design pushes the intelligence (and the computational load) to the edge – to the end user’s device, away from the device used by the service itself. This allows one to create servers that can support thousands of users in a cost-effective fashion, and scale globally as the service grows.

Isn’t AVC more efficient than SVC?

The benefits of SVC come at some cost. If we only look at the bits produced by and SVC and an AVC encoder for similar level of quality, then SVC will require about 10-15% more bits than AVC (this depends on the sophistication of the encoder). But it is very important to realize that this “overhead” gets us. We get low end-to-end delay and tremendous error robustness – none of this would be possible with AVC alone. In fact, if we take into account the quality loss caused by lost packets, we quickly see that AVC is *much less efficient* than SVC because its quality deteriorates very quickly. As a bonus, we get a server that easily scales to 100’s of users, and all high-end videoconferencing system features such as personalized layout and rate matching.

Focusing only on compression efficiency is a very narrow viewpoint, as it ignores what really happens in a complete system. By jointly considering both compression efficiency *and* network transport, one can design solutions in which features on both fields work together to address system-level problems.

It is also important to remember that SVC is an extension to AVC. This means that, when conditions allow, in other words when the network has extremely high reliability, all endpoints have similar access speeds to the network, and all endpoints have similar encoding and decoding capabilities, then an SVC system can revert to use plain AVC. In practice, however, when looking at users across an enterprise, we rarely have access to such a superbly reliable network, and we practically never have homogeneous access networks or endpoints.

We can make a parallel with cars and 4-wheel drive. While it is true that the 4-wheel drive will burn more fuel, who would want to venture out on an icy road with it? Plus, you can turn it off when you don't need it. The reality of course is that, in the global Internet, the road is always icy and treacherous for our packets.

Scalability and Next-Generation Codecs

SVC as a video coding standard has been finalized and published in Nov. 2007. As mentioned earlier, Vidyo has been very active in the standardization process, and led the efforts from the videoconferencing applications space. We also co-developed the RTP payload format for SVC, RFC 6190.

The demonstration of SVC's superiority for real-time applications resulted in all codecs after H.264 having full support for scalability. H.265 or High Efficiency Video Coding (HEVC) incorporated temporal scalability in its version 1 (2013), and spatial scalability in version 2 (2014), again with Vidyo's contributions. VP8 and VP9 have had temporal scalability from the very beginning, and VP9 has been extended with spatial scalability through joint work between Vidyo and Google. Vidyo has contributed to the RTP payload formats of all these codecs.

The great interest in SVC has also resulted in industry organizations wanting to further refine the core specifications in order to ensure interoperability. The International Multimedia Telecommunications Consortium (IMTC) has published two specifications, one for H.264 and one for H.265, detailing scalable encoder configurations suitable for Unified Communications applications. It is also developing certification programs that will ensure interoperability between different vendors' products at a high level of functionality. The work is performed by the Scalable and Simulcast Video Activity Group (SSV AG), co-led by Vidyo, Polycom, and Avaya.

Scalability, Simulcasting, and SFUs

A simplistic way to provide multiple representations of a video signal is to produce multiple encodings. Contrary to scalable coding, this does not require any new coding tools: you just run an encoder multiple times, feeding it its time with the video at different resolutions. This produces multiple independent bitstreams, and take more bits than scalable coding (50% more than the single high resolution stream is typical). Due to the lack of dependency between the two streams, this also tends to be less robust. This scheme is called "simulcasting" (originating from "simultaneous broadcasting") and can be considered a corner case of scalable coding.

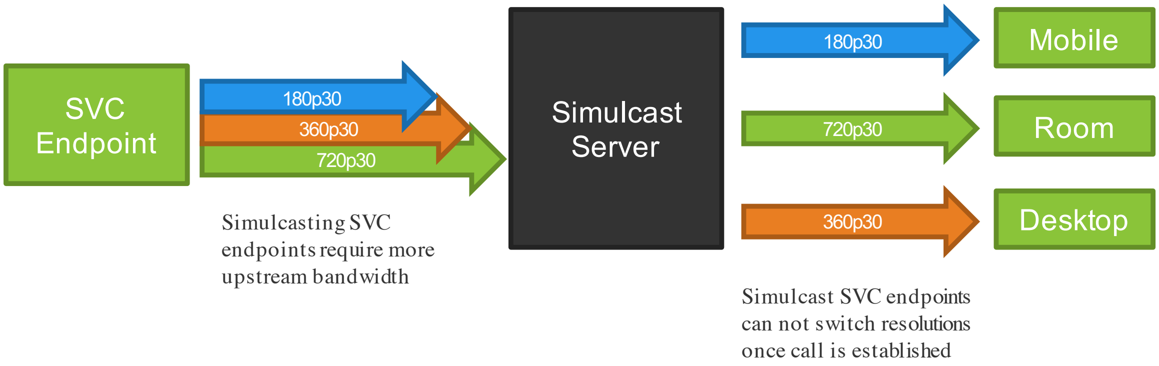
SVC Endpoint
180p30
360p30
720p30
Simulcasting SVC endpoints require more upstream bandwidth
Simulcast Server
180p30
720p30
360p30
Mobile
Room
Desktop
Simulcast SVC endpoints can not switch resolutions once call is established

Figure 6: Simulcast architecture

The motivation behind simulcasting is that you can use a server that operates similar to a VidyoRouter without using scalable coding: an endpoint transmits both high and low resolution streams to the server, and the server then selects with stream to forward to each receiving participant. Simulcasting places its overhead on the worse possible link: the uplink from the transmitting endpoint to the server. This tends to be the most problematic, e.g., with ADSL lines. One advantage of the simulcast technique is that the highest resolution bitstream can be decoded by a legacy (non-scalable) decoder. It is therefore attractive to companies that have a large number of deployed legacy hardware decoders that cannot be upgraded.

When the VidyoRouter was introduced in 2008, its principle of operation was very novel. In fact, the “RTP Topologies” RFC (RFC 5117) which was published at the same time, and which provides a survey of the various real-time communication architectures that can be implemented with RTP, did not foresee it. In October 2013 I coined the term “Selective Forwarding Unit” (SFU) to describe the operation of the selective forwarding server, regardless if it is using scalable video or simulcasting. In the revision of RFC 5117 that was published in November 2015 (as RFC 7667) the term has been adopted and its operation is now described in detail.

Two of today’s largest public deployed systems for real-time video communication, Google+ Hangouts and Microsoft’s Skype and Skype for Business, use scalable coding with simulcasting.

Scalability, Simulcasting, and WebRTC

WebRTC is gaining considerable momentum in the industry as the architecture of choice for browser-based and mobile endpoints. As users and vendors gain experience with WebRTC, it becomes clear that SFU-type server architectures are a necessity for high-quality, multi-point video. In fact, as WebRTC is multi-stream by design, i.e., an endpoint receives multiple video and audio streams rather than a single one, the SFU design is a perfect match. As a result, simulcast has been incorporated into the scope of the WebRTC 1.0 specification, whereas scalability will be part of WebRTC-NV (“Next Version”). Scalability is already supported in ORTC, the API originally incorporated into Microsoft’s

Edge browser, which has now been merged into the main WebRTC 1.0 specification. The WebRTC API itself has little support today in terms of codec configurations, but we expect that this will change to allow the usage of SFUs.

Why will SVC succeed?

Any new technology has to fight its way in the marketplace and win the hearts and minds of its users. There are, I think, two reasons why SVC and the VidyoRouter/SFU are succeeding already. First, for the first time they bring the complexity of videoconferencing systems down to the level of any other network application. This is very important for scaling videoconferencing to be an application that is used by everybody. Second, it brings the quality of the experience for the user to a level that is excellent even when operated on the public Internet – the technology becomes transparent. Today's users will tolerate nothing less. These two features will finally allow videoconferencing to be truly used anywhere, anytime, on any device. Indeed, if yesterday's technology, the MCU, was good enough, we would not see videoconferencing remain just a niche application. Today's design, quality, and ease of use, together with richer deployment options such as via APIs and the cloud, enable many more applications to incorporate video and allow orders to magnitude of more users to enjoy this wonderful communication tool.

Author’s Bio

Dr. Alex Eleftheriadis is the Chief Scientist and co-founder of Vidyo. He drives the technical vision and direction for Vidyo and also represents the company on standardization committees and technical advisory boards. He is an award-winning researcher, bringing over 24 years of research experience in video compression and communications to his role at Vidyo. Prior to Vidyo he was an Associate Professor of Electrical Engineering at Columbia University. Alex has more than 100 publications, holds more than 100 patents (several of which are used in Blu-ray Disc, H.264/AVC, and ATSC digital television systems), and has served as the Editor of the MPEG-4 Systems specification, Co-Editor of the H.264 SVC Conformance specification, and Co-Editor of IETF’s RTP Payload Format for SVC. He is Vice President and a Member of the Board of Directors of IMTC and co-chairs the IMTC Scalable and Simulcast Video Activity Group. His awards include a Marie Curie Chair from the European Commission, the ACM Multimedia Open Source Software Award, and the NSF CAREER Award. He received a Ph.D., M.Phil., and M.S. degrees in Electrical Engineering from Columbia University in 1995, 1994, and 1992, and a Diploma in Electrical Engineering and Computer Science from the National Technical University of Athens, Greece in 1990.

Resources

Find more information about the VidyoWorks™ platform and the Vidyo products described in this paper by using the links listed below.

Vidyo

- Vidyo web site:
  http://www.vidyo.com
- Vidyo Resources (White Papers, Case Studies, Data Sheets, etc.):
  http://www.vidyo.com/resources/

Vidyo, Inc. (Corporate Headquarters)
433 Hackensack Ave., Hackensack, NJ 07601, USA
Tel: 201.289.8597 Toll-free: 866.998.4396
Email: vidyoinfo@vidyo.com

Vidyo®

EMEA
emea@vidyo.com
+33 (0) 488 718 823

APAC
apac@vidyo.com
+852 3478 3870

INDIA
india@vidyo.com
+91 124 4111671

www.vidyo.com

Rev: 2016-06-02