Video
consumption dramatically outpaces Internet capacity
— exploration, challenges and solutions

Stephen Alstrup, Scott Brown, and Theis Rauhe
Octoshape

Video usage on the Internet is rapidly outpacing the Internet capacity deployed. Imagine if, in five years time, there were 100 times the number of cars on the roads today. That would be comparable to the video usage explosion now happening on the Internet. It is a huge challenge even for the rapidly improving IT sector.

This article discusses the main drivers for this explosion, the challenges and the solutions. First, we will explore the traditional streaming technologies, and their associated challenges of sustaining high-quality video experiences and buffering. Next we will explore how the quality problems can be solved, and how this creates an environment where a suite of technologies can be introduced, including Multicast, to cope with the exploding demand for video traffic.

The President of France invited executives, experts and entrepreneurs worldwide to the eG8 forum held in Paris at the end of May 2011. The goal was to discuss and advise politicians attending the upcoming G8 meeting if political regulation is necessary on the Internet. A major topic was that video is moving to the Internet so rapidly that the Internet may get fully congested if heavy infrastructure investments are not made and / or new innovative solutions are not found. The explosion in video growth is coming from the simple fact that many consumers now have connections to the Internet with capacity sufficient to consume both SD and HD quality video.

174 million US Internet users watched online video in March 2011 for an average of 14.8 hours per viewer. The total US Internet audience viewed more than 5.7 billion sessions during the course of a month [1]. Consumers want content anywhere and anytime. Mobile and landline video is dominating the Internet traffic as successful services become available, such as Netflix in the USA.

The challenges

In order for Internet video streaming to become interesting from a business perspective, it must support mass TV-scale audiences, at a quality level meeting living-room experiences, and at a cost enabling profitable business models.

❍ **Scale**

The Internet capacity today, with traditional technologies and distribution architectures, cannot handle the upcoming load in video traffic. The consumer might well have sufficient last-mile connectivity to the Internet but bottlenecks can arise in the middle mile or at ISP peering points to the Internet.

- **Quality**

  To make Internet video engaging in the living room, the quality needs to at least meet the consumer expectations of the traditional TV experience. For the consumer, quality can go up but never down. Consumers are buying bigger screens, and moving to HD, leaving SD experiences behind. Until recently, streaming typically involved frequent issues such as slow start times and buffering. Recent technologies have focused on the buffering issue by reducing the video quality in an attempt to avoid buffering. This is clearly an improvement, but trading between quality and buffering does not, by definition, meet the sustained quality that consumers expect today.

- **Cost**

  Recently the cost of distribution has reached a price point that makes advertising models profitable for SD-quality video distribution via the Internet. On the other hand, HD and soon 3D streaming require at least five times the bandwidth and therefore five times the cost. In order to facilitate profitable models for HD and 3D video, the cost of distributing the higher quality streams must decrease.

Quality is the solution to the combined problem of scale, cost and quality

This article will illustrate that solving the quality problem enables a set of further innovations that allow for the use of more efficient streaming technologies such as Native Multicasting, and the use of inexpensive cloud resources. The combination of these technologies solves both the scale and cost problem. Hence, quality of distribution is the key to solving all three problems at the same time.

What is the quality of video distribution?

Quality of distribution, at the very minimum, includes the following:

- **Instant On** – Fast video response from click to play;
- **Stability** – No buffering or interruption in the stream;
- **TV grade uptime** – Always on;
- **Sustained high bitrate** – No adjusting the video quality;
- **Fast channel change** – Fast channel switch time.

These characteristics apply anywhere, anytime, on any platform, at any scale. They must hold true while, at the minimum, meeting the consumer expectation of TV quality in the living room. Having a sustained high-quality stream not only enables a TV experience, but also extends the viewer engagement time, reduces support, and enables subscription services as people are willing to pay for quality.

Abbreviations

AMT	Automatic IP Multicast without explicit Tunnels	**FEC**	Forward Error Correction
CDN	Content Delivery Network	**HTTP**	HyperText Transfer Protocol
CEO	Chief Executive Officer	**IP**	Internet Protocol
CPU	Central Processing Unit	**RTT**	Round-Trip Time
CTO	Chief Technology Officer	**TCP**	Transmission Control Protocol
DVR	Digital Video Recorder	**UDP**	User Datagram Protocol

Why is there a quality problem?

Traditional streaming technologies today are built on TCP (e.g. http streaming). TCP was not historically designed for streaming. It was built with acknowledgment-based mechanisms to track and ensure that if the sender sent data, the receiver ultimately will receive it. This reduces the complexity required at the application level (Media player), so it doesn't have to worry about missing any video frames. It unfortunately comes with a cost. To get to the root of the issue a deeper view into the acknowledgement protocol is required.

The sender and receiver have a two-way conversation whereby the receiver acknowledges receipt of the packets sent. For example, the sender sends packet number 1, and the receiver responds with, "Send number 2". The time it takes for a message to be sent and acknowledged over a given path is called the Round Trip Time (RTT). In this way the combination of RTT and the amount of packet loss is limiting how high the bitrate can be. This problem will be explored in further detail, after a review of the common places where RTT / packet loss occur.

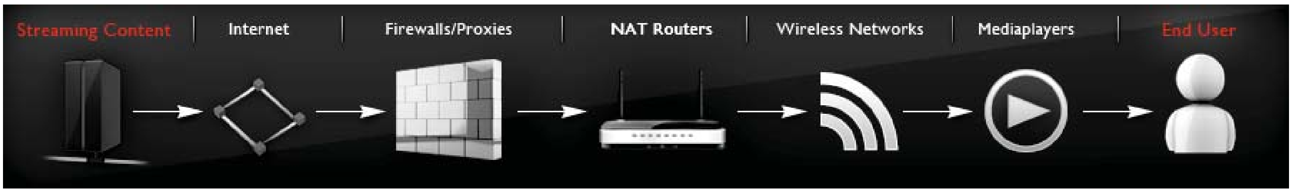

Figure 1
Many things on the way from source to end-user can increase RTT and packet loss

Where the problems surface

While packet-loss issues exist on the Internet, there will always be problems with throughput and stream quality for traditional streaming solutions. Here are some places where the combination is great, causing severe quality problems.

- ***Mobile streaming*** – On mobile networks there are much higher numbers of packets lost, and the latency to receive acknowledgements is significantly higher**.**
- ***Large scale streaming*** – High utilization on the Internet causes congestion and packet loss.
- ***Global streaming*** – Increases both RTT and packet loss, making it close to impossible to stream higher than 600 kbit/s < SD quality using standard streaming technologies.
- ***Wireless routers*** – In apartments, hotels, hot spots with many users and routers, interference will create high packet loss. This can also occur with a single user being far away from the router.
- ***Upstream congestion*** – If one family member is uploading video to a social networking site taking all the available upload capacity, TCP acknowledgements for the downstream viewing experience of other family members can be impacted, making it impossible for the person(s) viewing the stream to achieve high quality.
- ***HD streaming*** – The higher the video resolution, the higher the probability of reaching the maximum bitrate supported on a given TCP session, determined by the combination of packet loss and RTT.

TCP by design cannot sustain high bitrate traffic flows

To further understand quality problems, the TCP Window Size must be defined. This is the amount of data that can be "in flight" or sent from the sender without being acknowledged from the receiver.

Inside of one second there is a discrete amount of time that a sender can send, and a receiver can acknowledge packets. For example, consider the sender is 50ms away from the receiver, hence RTT is 100ms. Given a 64 KB TCP Window, this means that 64 KB can be "in flight" or unacknowledged at a time. Since it takes 100ms for an acknowledgment to happen, then the max throughput possible is 640 KB per second, or about 5 Mbit/s. Doubling the RTT or halving the TCP window results in half the achievable throughput.

The TCP algorithm starts the TCP Window off small, and then grows until a packet is lost to achieve maximum throughput. This loss of a packet is considered a sign of congestion, for which the algorithm responds by drastically reducing the window size to avoid the congestion. This in turn reduces the effective throughput immediately. The TCP window will begin to grow again until another congestion event occurs. Visually the throughput variability looks like an hourglass inside a given set of capacity *(see Fig. 2)*. Additionally on the Internet there is always packet loss, even if there is no congestion. TCP does not know why a packet is lost, and always concludes it must be a congestion problem.

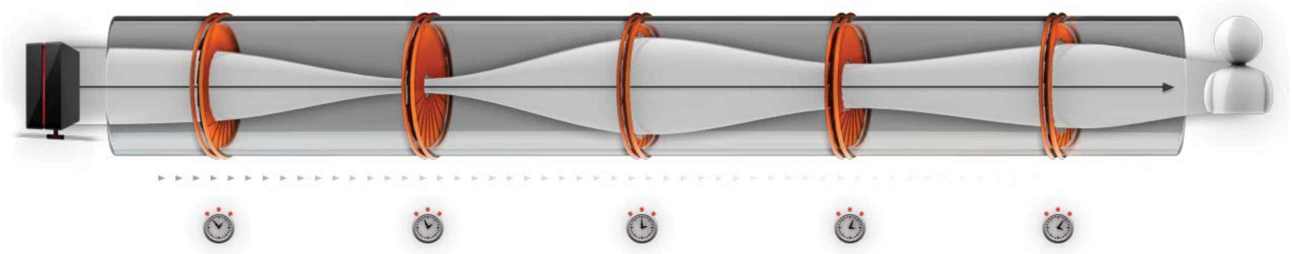

Figure 2
Using traditional streaming technologies the actual throughput of a 1 Mbit/s stream will increase and decrease like an hourglass through a given capacity on the Internet

Hence a 1 Mbit/s TCP stream is never translated to a consistent 1 Mbit/s of throughput on the Internet. It dynamically changes all the time. One moment, for example, it can use 6 Mbit/s of capacity and the next moment only use 100 kbit/s of capacity.

This is easily verified by looking at a network bandwidth monitor when watching a traditional streaming technology. Now consider a scenario where there are two 1 Mbit/s TCP streams passing through a 10 Mbit/s pipe.

As the two 1 Mbit/s streams compete for the capacity with the oscillating nature between high throughput and low throughput, the two streams will from time to time spike at the same time, resulting in congestion and packet loss, which dramatically reduces the throughput available, etc.

If the driver of a car were using the TCP approach, he or she would increase the speed until crashing into another car (congestion problems) or hitting a

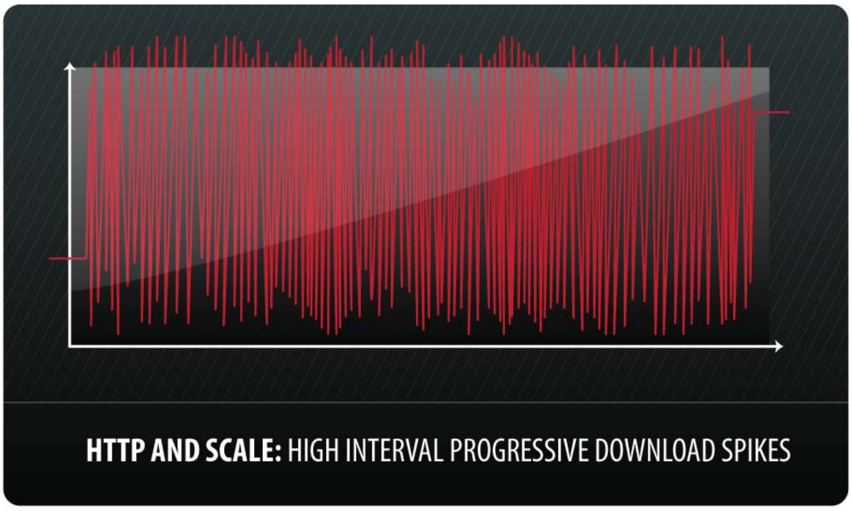

Figure 3
HTTP chunk-based streaming pulling a 1 Mbit/s stream

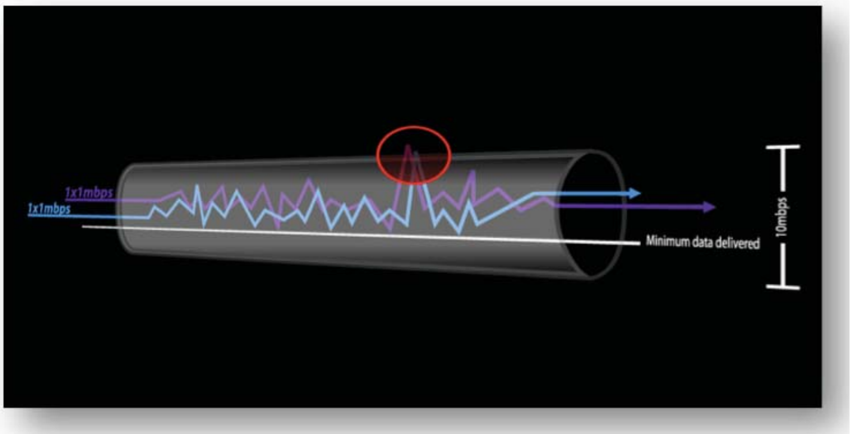

Figure 4
Two 1 Mbit/s traditional streams can be too much for a 10 Mbit/s pipe

speed bump (a small packet loss). In both cases the driver would take his/her foot off the accelerator ... then slowly speed up again, well above the speed limit, until the next incident. Clearly if everyone was driving using this logic, the pace of the traffic would slow to a crawl. To maximize throughput both on the highway and on the Internet, traffic must flow with a constant speed until a real problem occurs, then the speed must be reduced until the problem has been resolved.

Summarizing the challenges for TCP ... packet loss and RTT control and constrain the throughput and thus the quality, with streaming technologies relying on the congestion mechanisms in TCP. TCP constantly increases and decreases the bandwidth consumption for the same bitrate, resulting in congestion, even in scenarios when sufficient capacity actually exists.

Traditional "solution" to the TCP problem.

Traditional streaming service providers (CDNs) combat this RTT / packet loss problem by moving streaming servers close to the edge, or building networks to make the servers look "close". Being close to the edge reduces the RTT and the probability of packet loss. This is the reason why the marketing tactics of traditional content delivery networks focus on the number of edge hosts, or the number of geographies in which their infrastructure is distributed. The market is responding to a fundamental truth of TCP-based technologies where the distance the server is from the consumer is directly related to the quality of the video experience they can achieve. Said another way, "Distance equals Quality". The paradox is that even having servers close does not "fix" the issue. There are still many routers and networks between the edge server and the consumer. Understanding this requires an exploration into the networks past the "edge".

- ***Mobile streaming*** – The majority of the RTT and packet loss in these networks arises after the edge server between the cell tower and the device.
- ***Large scale streaming*** – As more people tune into a stream, the more congestion occurs on peering links. The more congestion, the more TCP-acknowledgement retransmissions occur. As this increases, the more the TCP bandwidth consumption fluctuates up and down, creating more congestion. Furthermore, typically a CDN can have many machines, but never enough locally at large events. Viewers are directed to servers further away, thus reducing quality and increasing congestion.
- ***Global streaming*** – There is no CDN that has machines deployed in regions to the level of achieving a consistent quality globally. They may have a large footprint and higher quality in the USA, but that does not equate to quality in Asia and vice versa. The same concept applies looking at more specific regions. For example, a CDN may have many machines in France but few in Germany. In these cases the user may be very far from the edge.
- ***Wireless routers*** – Wireless technologies traditionally have fluctuating RTT and packet loss due to local interference. These problems are not addressed by having edge servers close by.
- ***Cloud CDN Services*** – The largest investment in Internet infrastructure today is focused on cloud infrastructures. Many CDN providers are trying to utilize these clouds as streaming networks. These infrastructures don't match the "Distance equals Quality" rule and are purpose-built to be more centralized, gathering massive CPU and Internet capacity together.
- ***Telco / Operator-based CDN*** – Telcos and operators are enabling their networks to become CDNs, which are facing the same issues. Standard adaptive quality is likely to be achievable inside their own network, but the audiences of the media companies are unlikely to all reside inside the zone of a single Telco or operator, leaving these users far away from the edge.

There are many techniques trying to work around the throughput and quality constraints of TCP-based streaming. However a good solution does not exist for providing sustained high-quality streaming using TCP. In light of this, the industry has swiftly begun moving to adaptive bitrate technologies to trade off buffering for a reduced video quality. This is a work-around, not a fix. These technologies are far from perfect. Recent scientific experimental analysis has concluded [2]:

> ***"Overall, it is clear that the existing adaptive HTTP streaming players are still at their infancy. The technology is new and it is still not clear how to design an effective rate-adaptation logic for a complex and demanding application (video streaming) that has to function on top of a complex transport protocol (TCP)."***

More details can be found in [3].

Making best-effort networks perform like provisioned networks

Octoshape has developed an innovative streaming transport protocol, which is resilient to the traditional issues of RTT / packet loss. The technology provides an efficient use of Internet capacity today and can deliver consistent sustained high video quality for the consumers, and smooth deterministic flows for the network operators.

This technology breaks the relationship between distance and quality, making best-effort networks perform like provisioned networks. The technology suite is called ***Infinite HD***.

Here are some examples of what the technology can achieve:

- ***Source Signal Contribution*** – Stream 5 Mbit/s signals sustained (not adaptive) over the open Internet from an event venue in Paris to a TV production centre in Shanghai, then distributed over the terrestrial television network.
- ***Global Reach*** – Live streaming from anywhere to anywhere (Brazil, Singapore, Korea, Germany, the USA, Australia) with sustained high quality (not adaptive) and a globally consistent delta from an encode time of 4 seconds.
- ***Large-Scale Events*** – Facilitated the largest events on the Internet today above 1 Million simultaneous viewers, for the largest media companies proving that every single bit can be delivered, and the quality can be sustained to the consumer.
- ***Efficient / High Quality Mobility*** – Providing constant high bitrate to mobile devices while driving around a city.

Clearly, the technology also performs exceptionally, delivering sustained high-quality video in normal Internet-streaming scenarios. The Octoshape ability to sustain high quality over best-effort networks is founded in the core algorithms employed in the Octoshape transport protocol, using a unique resilient-coding scheme inside a UDP transport. This scheme enables among other things:

- Streaming integrity surviving the packet loss without the need for retransmission or the data overhead of forward error-correction (FEC) schemes.
- Pulling data from multiple sources simultaneously. Sources can be of various types such as cloud and Multicast, while only actually consuming the data rate necessary to recreate the stream.
- Maximum achievable bitrate detected and sustained as an inherent part of the flow-control design. As an example, a 1 Mbit/s stream looks like a consistent 1 Mbit/s flow over the Internet.

Note: In situations where UDP is not available, Octoshape transport protocol can use TCP. However, Octoshape uses its own resiliency and throughput models, treating TCP as UDP and not relying on the standard TCP mechanisms and constraints that come with it.

Exploration into how Octoshape delivers a 1 Mbit/s live stream

Octoshape deploys software on or near a standard encoder called the Octoshape Broadcaster. There is also an Octoshape App on the receiving device. In between the Octoshape Broadcaster and the App is the OctoshapeCloud. The OctoshapeCloud consists of a mix of resources including a suite of Multicast technologies and multiple cloud infrastructures (i.e. Amazon EC2). The Octoshape server software exists in the OctoshapeCloud.

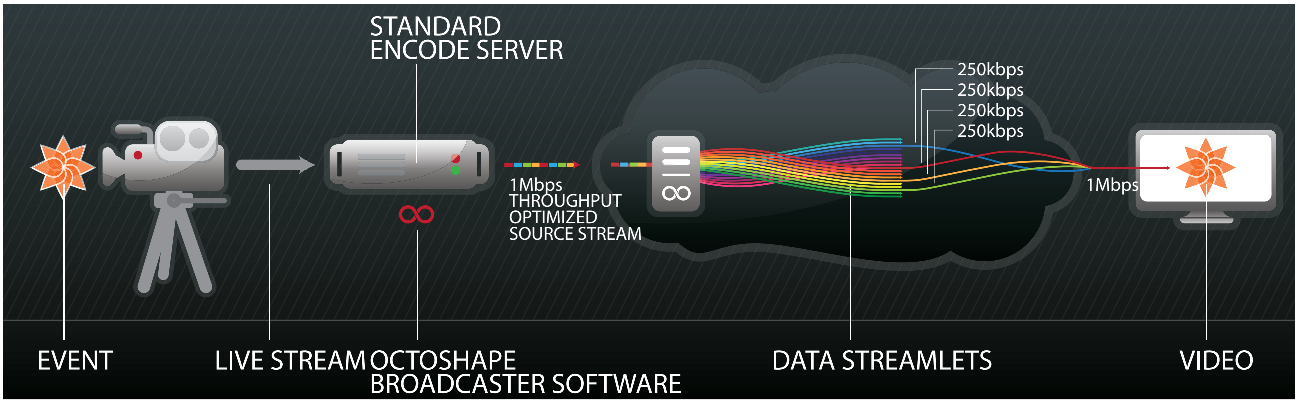

Figure 5
Octoshape from encoder to end-user

In the OctoshapeCloud, the stream is broken up into many unique streamlets and replicated. This example assumes the streamlets are each 250 kbit/s. The streamlets are coded in a way that 20 unique streamlets may exist across the OctoshapeCloud, but only four are needed to recreate the original 1 Mbit/s stream for the consumer.

When the Octoshape App at the client requests the 1 Mbit/s stream, several things happen in parallel:

a) The maximum bitrate to the Octoshape App is detected and the App is fed with a higher bitrate flow than 1 Mbit/s to quickly start streaming and create a small buffer.

b) The Octoshape App becomes aware of the existence of multiple video streamlet sources. The App can tune into many streamlet sources in parallel. The App constantly prioritizes the sources based on latency, packet loss, locality, and other parameters to create a resilient mesh of sources ranked by quality and localization. If a streamlet source is behind a congested route or a source becomes unavailable, the Octoshape App can pull in streamlet data from any other source to take its place immediately without stream interruption. As a result, any source can drop out, any packet can be lost, a data centre can go offline, or a fibre could be cut, and the consumer will not see a frame drop in the video.

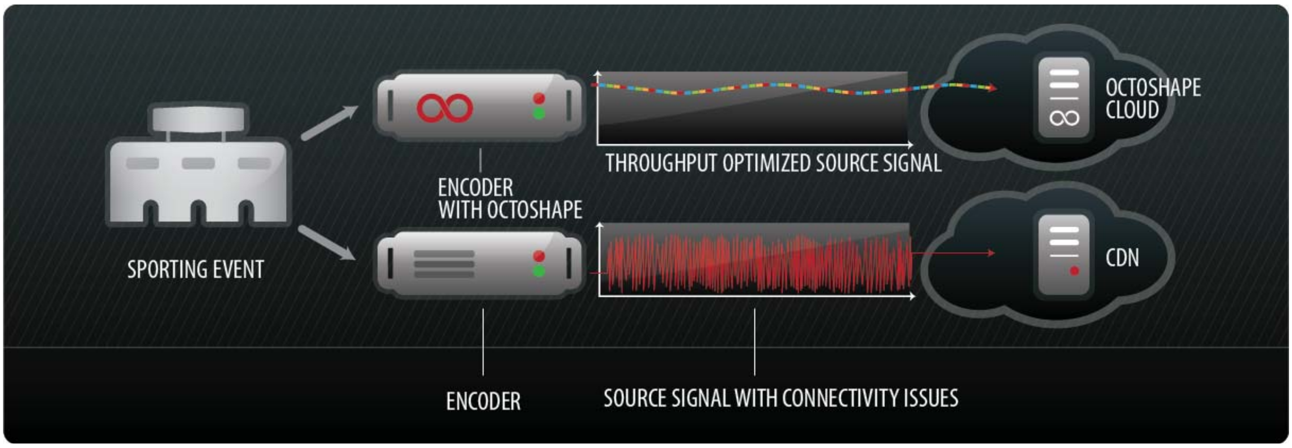
OCTOSHAPE
CLOUD
THROUGHPUT OPTIMIZED SOURCE SIGNAL
ENCODER
WITH OCTOSHAPE
SPORTING EVENT
CDN
ENCODER
SOURCE SIGNAL WITH CONNECTIVITY ISSUES

Figure 6
Octoshape throughput-optimized source signal contribution compared to traditional streaming from venue

The same resiliency created from the OctoshapeCloud to the App is built in to the connection between the broadcaster and the OctoshapeCloud. This pathway in the "First Mile" is perhaps the most important. The stream quality to the consumer can only be as good as the contribution feed into the system. In addition, the system enables a redundant encoder failover facility, even if the encoders exist in different locations.

Octoshape quality over best-effort networks (summary)

Traditional streaming solutions are built upon TCP with the built-in constraints of “Distance equals Quality”. Octoshape has broken the relationship between Distance and Quality by creating a unique throughput-optimized streaming transport that is resilient to packet loss, and enables efficient use of multiple video sources simultaneously, opening up all the paths into the last mile. The unique coding scheme of the streamlets gives the Octoshape App the freedom to combine and replace streamlet data sources. This enables optimal quality, traffic localization in the network, and facilitates a fully redundant failover system.

The UDP-resilient stream also facilitates a smooth and deterministic flow of traffic whereby a 1 Mbit/s feed looks to the network like a smooth 1 Mbit/s flow of data, avoiding the congestion problems that the TCP spike creates, and makes it possible to calculate how many streams can go through a pipe.

Having broken the “Distance equals Quality” relationship, high-quality video can attain global reach and data can be served from locations where bandwidth, power, cooling and space are inexpensive, thus reducing the cost of streaming distribution.

Note: The Octoshape technology is also capable of switching between multiple bitrates but this rarely happens as the bitrate is selected accurately first, and the technology can sustain the throughput required.

These core innovations have made way for dramatic architectural improvements, enabling Internet distribution methods that have before been challenged. Two of these innovations are Octoshape’s Multicast Suite of technologies and the use of cloud infrastructures as described in the next section. For more information on the core technology of Octoshape please refer to [4][3].

Cloudmass

Today, there is a “race in the market” to deploy cloud infrastructure. Many large providers are deploying an immense infrastructure of centralized, shared network and computers platforms. Looking at the capacity deployed in just one of the top ten providers reveals more computers and network provisions than the largest CDNs on the planet.

With Octoshape breaking the relationship between Distance and Quality, the cloud centralized infrastructure (opposite to edge servers) can be used for delivering high-quality video. However, the Cloud also represents shared resources. One moment 100 Gbit/s is available, the next moment only 50 Gbit/s. The Octoshape App overcomes this problem by using multiple sources simultaneously, and can switch sources transparently as needed. This concept extends to using a pool of resources that span many cloud providers all around the world, regardless of region. These sources can be dynamically provisioned and combined, selecting the cheapest and highest quality resources that avoid congestion. This is Octoshape Cloudmass. Cloudmass was first launch in October 2010 for a popular Korean music award show called the *Golden Disk Awards*. The streaming was presented on MSN in seven different countries in the Asia region, in HD with DVR functionality.

“*We at Jcube and Microsoft are thrilled to deliver the Golden Disk Awards with the HD quality to the users of JoinsMSN.com in Korea and MSN Asia*” said Sang-soon Park, CEO of Jcube. “*The quality streaming is evidence of how the new portal engages and entertains its users and audience. Thanks to Octoshape, we could meet our users’ satisfaction.*”

The scale impact of using clouds

To grasp the impact of Cloudmass, we must first explore the effect the Octoshape Multicast suite of technologies has on a pool of bandwidth capacity. For example, in a given scenario, Octoshape can make a CDN with 50 Gbit/s of deployed capacity perform and deliver the capacity at the level of a CDN with 500 Gbit/s deployed. This was the case for the coverage of Barack Obama’s Inauguration as President of the USA in 2009, where Octoshape was deployed at a smaller CDN provider that served as much video traffic as one of the largest CDNs in the space.

Additionally, Octoshape was able to sustain the bitrate through the peak of the event, where traditional streaming technologies, crippled by congestion, would only have been able to sustain 70 percent of the video bitrate, and then would have buckled under the load.

Taking into account the Octoshape Multicast magnifying glass effect and dynamically deploying it across fifteen percent of one of the large cloud providers, it can instantly serve at the capacity level of one of the largest CDNs. If fifteen percent of the top ten cloud infrastructures were dynamically aggregated together at once, using Octoshape, they could recreate the capacity of the entire world's CDNs streaming capacity combined today.

More information about cloud streaming can be found in this analysis by the The Diffusion Group [5].

Multicast suite of technologies

Octoshape's suite of three Multicast technologies – Native Source-Specific Multicast, AMT (Automatic IP Multicast without explicit Tunnels), and Octoshape Simulated Multicast – provide the magnification effect on top of the Cloudmass infrastructure.

Multicast – reborn

Multicast has historically been restricted to provisioned networks. It is used in the enterprise, or in managed IPTV systems, where the congestion and packet loss can be controlled and managed. There are many reasons Multicast has not been widely adopted across the public Internet, one being that it is not a resilient transport mechanism in native form, making it hard to get high quality in best-effort networks. Another is that Native Multicast requires Multicast configurations being enabled in every network routing device between the sender and the receiver. The Octoshape streaming protocol handles the first problem related to quality. This enables Multicast to become reliable over best-effort networks. The second represents an all-or-nothing problem. Either there is Multicast enabled everywhere from source to receiver or Multicast cannot be used. As a result Multicast has had limited adoption on the Internet. AMT has been introduced to create a bridge for this last problem, by making it possible to use Multicast on only a part of the Internet. In this way Multicast can be incrementally deployed without enabling Multicast protocols on every device.

Octoshape provides a transparent path to enable and gain all the efficiency benefits of Multicast today, using Octoshape Cloudmass and Simulated Multicasting. Then if the telco or operator deploys AMT and Native Multicast, the Octoshape App will inherently take advantage of these resources as well. This puts the rollout timeline in the telco's or operator's hands. Furthermore, using multiple sources in parallel as described above, the resources are all used transparently and simultaneously without interfering in the consumer's sustained HD experience, or having concerns about misconfiguration or software errors in the Multicast rollout.

In November 2010, AT&T and Octoshape combined forces using the AMT approach with Octoshape high-quality technology solutions for an event delivering streams to an audience located in more than 100 countries. John Donovan, CTO AT&T, posted a blog entry about the event saying:

> ***"At the event, AT&T and Octoshape, a next-gen streaming technology provider, delivered an "instant-on" high quality viewing experience to a wide range of connected devices and operating systems."***
>
> ***"Bringing this capability to the open Internet could fundamentally change the rules that govern online distribution today." [6]***

The following sections will cover the entire Multicast suite in more depth. Full detailed technical information about the whole suite of multicast from Octoshape can be found in [7].

Octoshape Simulated Multicast

Octoshape Simulated Multicast is described in detail in [4], while below there is a brief overview. Historically, technologies that try to replicate the efficiency of Multicast have focused on central bandwidth reduction at the cost of reducing quality. The Octoshape approach has focused on quality first, resulting in a transport protocol that is RTT / packet-loss resilient, multi-source, multi-path, and efficiently enables a Simulated Multicast.

While using Simulated Multicast, the Octoshape App is not only receiving from the OctoshapeCloud but also other Octoshape Apps that are tuned into the same stream at the same time. This is executed reliably using the Octoshape resilient transport layer with the possibility of mixing sources as described above. The criteria of optimizing quality and keeping traffic local are consistent through Octoshape App monitoring and selecting the best available resources. The only difference is that a number of conservative limits are applied on the use of other Octoshape Apps as a resource. This is applied to ensure only idle resources are being used, balancing and capping the use of such resources in a consumer- and network-friendly way. The criteria has been designed such that the method adapts to simulate Native Multicast even when Multicast not is available, thus the name Simulated Multicasting.

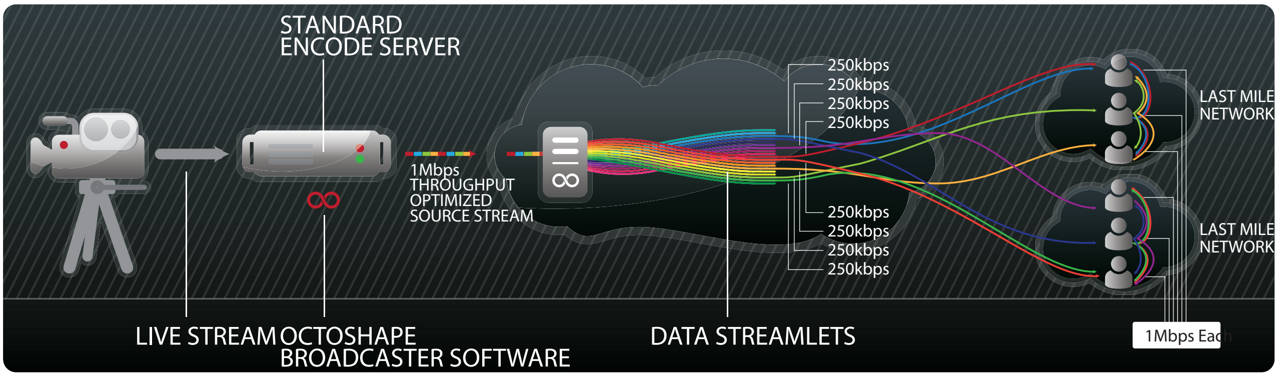

Figure 7
Simulated Multicast using Octoshape's resilient code scheme

The benefits are mutual:

- keeps the traffic local in the network and thereby avoids congestion problems, making large-scale high-quality streaming possible.
- localizing the traffic also reduces the ISP, telco and operator costs for network peering.
- can reduce bandwidth costs by 80-97% (the variance depending on environmental parameters).
- Further increases quality by avoiding and minimizing congestion on the Internet.

Native Source-Specific Multicast

In this model, Octoshape provides software to the last-mile provider that resiliently pulls a stream, or set of streams, into the last mile and injects it into the Native Multicast environment of the provider. The provider would give Octoshape a pool of SSM (s,g) addresses to manage for these streams. By doing this, the SSM address becomes the preferred resources in the OctoshapeCloud. Seamless switching between streaming data from the cloud and for Native Multicast resources happens while the consumer is watching. This is transparent to the viewing experience for the end user. In case there is packet loss on the Native Multicast feed, the Octoshape App fills the missing packets from other resources in the OctoshapeCloud, using the multi-source approach. In addition if there is some Multicast failure or a device preventing Native Multicast from working, the Octoshape App will transparently switch to using Simulated Multicast.

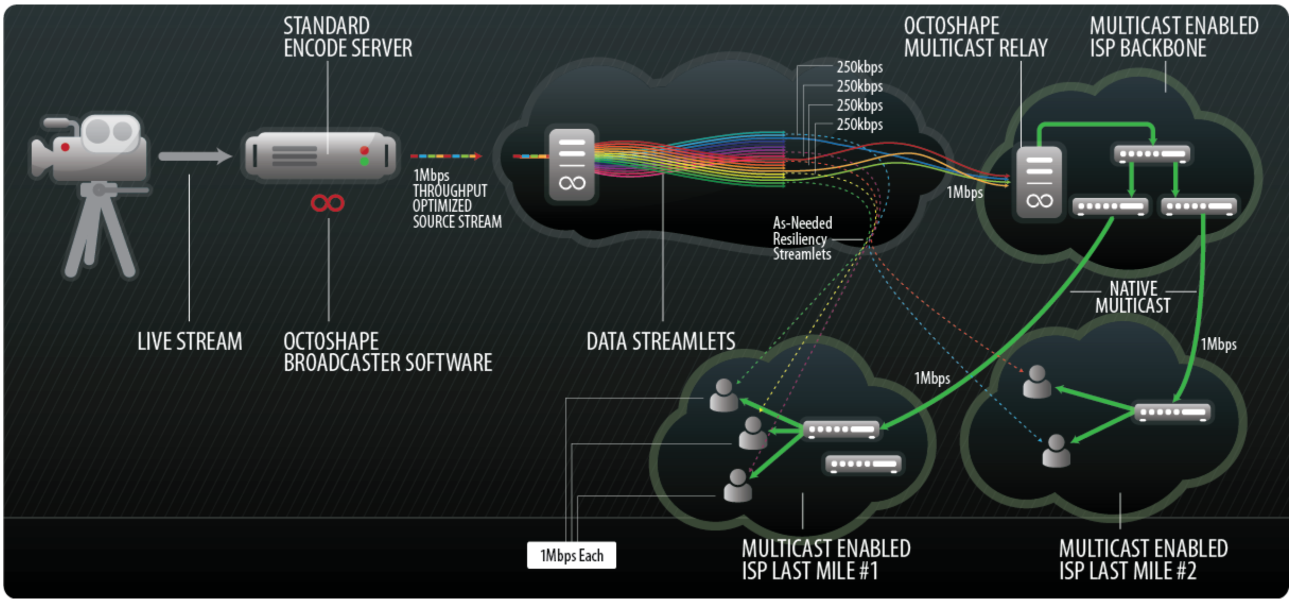
STANDARD ENCODE SERVER
OCTOSHAPE MULTICAST RELAY
MULTICAST ENABLED ISP BACKBONE
250kbps
250kbps
250kbps
250kbps
1Mbps THROUGHPUT OPTIMIZED SOURCE STREAM
1Mbps
As-Needed Resiliency Streamlets
NATIVE MULTICAST
1Mbps
1Mbps
LIVE STREAM
OCTOSHAPE BROADCASTER SOFTWARE
DATA STREAMLETS
1Mbps Each
MULTICAST ENABLED ISP LAST MILE #1
MULTICAST ENABLED ISP LAST MILE #2

Figure 8
Native Multicast using Octoshape resilient code scheme

AMT

AMT is another option for efficiently moving video data to the edge of the network in instances where Native Multicast is not enabled. AMT is a Multicast tunnelling process built into router code that can bridge a Multicast and non-Multicast domain. It can extract one copy of the video into the last mile, and serve multiple copies as a relay from there. For a detailed explanation of AMT, we refer to [8].

For this case, the last-mile provider has some portions of the last mile enabled with Native Multicast, and some that are not. As in the previous scenario with Native Multicast, Octoshape can inject the streams into the Native Multicast domain of the last mile.

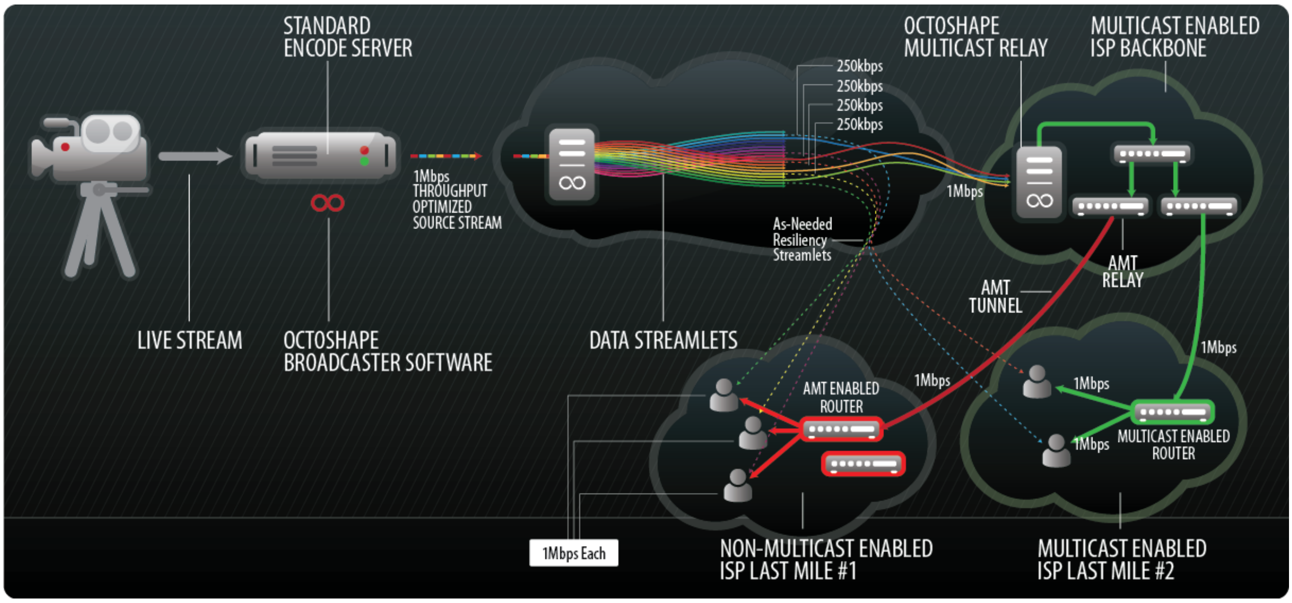
STANDARD ENCODE SERVER
OCTOSHAPE MULTICAST RELAY
MULTICAST ENABLED ISP BACKBONE
250kbps
250kbps
250kbps
250kbps
1Mbps THROUGHPUT OPTIMIZED SOURCE STREAM
1Mbps
As-Needed Resiliency Streamlets
AMT RELAY
AMT TUNNEL
1Mbps
LIVE STREAM
OCTOSHAPE BROADCASTER SOFTWARE
DATA STREAMLETS
AMT ENABLED ROUTER
1Mbps
1Mbps
1Mbps
MULTICAST ENABLED ROUTER
1Mbps Each
NON-MULTICAST ENABLED ISP LAST MILE #1
MULTICAST ENABLED ISP LAST MILE #2

Figure 9
Combining Native, AMT, Simulated Multicast and OctoshapeCloud streaming

The client that is on the non-Native Multicast portion of the network seeks to tune to the stream. The media player requests the stream from Octoshape, and the OctoshapeCloud immediately sends instant stream start data to the last mile, enabling the video to begin playing. The OctoshapeCloud begins to send alternative sources from which the App can pull resilient data. Among these sources is the Native SSM address.

The Octoshape App in the background attempts to tune into this SSM address and immediately finds it unavailable since it is not connected to the Native Multicast domain. The Octoshape client then sends an Anycast request out to find the closest AMT relay, if available. The closest relay responds to the request, and tunnels to the Native Multicast domain to pull one copy of the stream into the AMT relay.

The Octoshape App then begins to receive the feed from the AMT relay. If bits are dropped along the way, the Octoshape App fills the holes by drawing from the OctoshapeCloud as in the other cases described above.

Note: AMT Multicast models apply in enterprise and last-mile networks.

In a current Technical Review article on IPv6 [9], the exhaustion of the IPv4 address space is described. The only expansion plan for more IP address space is IPv6 with native Multicast support. IPv6 should gain significant traction over the next decade: the Octoshape solution will inherently support Multicast over IPv6, enhancing video distribution by solving the quality issues over best-effort networks.

Conclusions

The volume of video traffic being carried on the Internet now is exploding and new innovative solutions are needed to handle the load. In addition, as streaming is moving to “Over The Top” delivery models, the quality must be a TV experience.

Stephen Alstrup is a leading expert in algorithms, who has published a number of papers in the areas of data structures and, in particular, tree structures. Prior to founding Octoshape, he was associate professor of The IT University of Copenhagen, where he established and led the Algorithm Group. Dr Alstrup received his Ph.D. in Theoretical Computer Science from the University of Copenhagen in 1999. During his professional carrier, he has worked as a consultant and in research groups on large scale problems for enterprises such as AT&T, Google and Microsoft.

Theis Rauhe is a world-renowned algorithm scientist, making significant contributions in the areas of data structures, lower bounds and software engineering. Prior to founding Octoshape, he was associate professor of The IT University of Copenhagen and a member of various research committees. He has lead several large software projects founded by organizations such as the European Union. Dr Rauhe received his Ph.D. in Theoretical Computer Science from BRICS at the University of Aarhus in 1999.

Scott Brown joined Octoshape in January of 2009. Prior to this, he ran Content Delivery Services and Media Technology Strategy for Turner Broadcasting Inc. He spent 11 years at Turner Broadcasting and AOL, focused on media infrastructure development and high-scale media delivery on the Internet. He holds several streaming media technology patents and has architected streaming solutions for the largest events on the Internet to date, including coverage of the 2009 Presidential Inauguration with 1.34 million simultaneous users.

Octoshape has addressed and solved the quality problems for streaming on the open Internet, rather than working around the problem by adapting to the ever diminishing capacity available. These adaptive bitrate technologies simply substitute buffering for low video quality, failing to address the expectations of the audience for a living-room experience. By solving for quality, Octoshape has enabled streaming in high quality over long distances, using a wide set of cost-efficient resources such as Multicast and Cloud infrastructures. This increases scalability and reduces the cost of distribution to meet the exploding demand for Internet video, today and tomorrow. Furthermore, the solution enables global federated content distribution, Multicast-enabled enterprise streaming, and reduces traffic in all peering points. The system is codec-, format- and DRM-agnostic and supports all popular OS platforms and browsers.

References

[1] comScore: **comScore Releases March 2011 U.S. Online Video Rankings**
March 2011.

[2] S. Akhshabi, A. Begen and C. Dovrolis: **An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP**
MMSys '11 Proceedings of the second annual ACM conference on Multimedia systems.

[3] **Octoshape solutions, breaking the constrains of scale, quality, cost and global reach in content delivery**, 2011

[4] S. Alstrup and T. Rauhe: **Introducing Octoshape – a new technology for large-scale streaming over the Internet**
EBU Technical Review No 303, July 2005

[5] The Diffusion Group: **The Cloud Effect: Can Cloud Resources be used to deliver quality, scalable, and affordable OTT Video**, 2010

[6] J. Donavan blog on **InnovationSpace**

[7] Octoshape's Multicast Technology Suite: **The next-Gen CDN alternative for Large-Scale, Cost-optimized Global HD streaming**.

[8] T. Kernen and S. Simlo: **AMT– Automatic IP Multicast without explicit Tunnels**
EBU Technical Review, 2010 Q4

[9] T. Kernen: IPv6 and content providers
EBU Technical Review, 2011 Q2

This version: 27 June 2011

Published by the European Broadcasting Union, Geneva, Switzerland

ISSN: 1609-1469

Editeur Responsable: Lieven Vermaele

Editor: Mike Meyer

E-mail: tech@ebu.ch

The responsibility for views expressed in this article rests solely with the authors