move networks, inc.

The Next Generation Video Publishing System

Move Networks provides a new standard for online video distribution

move networks, inc.

Address
796 East Utah Valley Drive
American Fork, UT 84003

General Information
Phone: 801.216.8861
Fax: 801.756.5806
info@movenetworks.com

Sales Information
Phone: 801.216.8828
Fax: 801.756.5806
salesoffice@movenetworks.com

The Next Generation Video Publishing System

Executive Summary 3
The Problem with Traditional Streaming Video 4
The Solution: Move Adaptive Stream 6
How Move Adaptive Stream Works 6
The Advantages of Move Adaptive Stream 7
Encoding Advantages: Move Simulcode 8
Transmission Advantages 10
Client Control and Playback Advantages 12
Scalability Advantages 12
Publisher Feedback and Administration Advantages 13
Virtual Timelines 14
Digital Rights Management 15
Summary 15
Appendix A 17
Putting Move Adaptive Stream to the Test 17
Effective Bit Rate 17
Start Delay Time 18
Playback Pauses After Start 18
Watchability 18
Testing Summary & Conclusions 19
Appendix B 20
Feature Comparison Table: Move Adaptive Stream versus Other methods 20
Intelligent Video Services SDK 21
Move Networks Integration for Typical Network Broadcasting Station 21

move networks, inc.

Executive Summary

To date, traditional streaming technology has not been effective in delivering high-quality video content across the Internet. The typical user experience is limited by delayed starts, poor quality, buffering delays, and inadequate navigation. Even under optimum conditions, traditional streaming lacks features such as PVR-like navigation, personalized ad insertion, intelligent indexing and searching, and client usage statistics.

The new technologies offered by Move Networks, Inc., offer a different experience. The Move Networks Publishing System employs streamlet technologies and includes four core services: Move Simulcode, Move Adaptive Stream, Move Publish, and Move Monetize. Together, these technologies and services provide a publishing system with features that have not previously been possible for video streaming:

Current streaming technologies are not up to the challenge of distributing videos, movies, and television programming.

- Video encoding is no longer limited by a computer's processing power because of Move Simulcode, a new process that implements hardware clustering and software pipelining. One process will prepare video for on-demand or live streaming to all target platforms (cell phone, laptop, desktop, large screen TV) and bandwidth profiles (dial-up, wireless, mobile, broadband).

- Streaming is no longer plagued by constant rebuffering, slow starts, and poor quality. Move Adaptive Stream protocol uses a patent-pending approach to eliminate the negative effects of packet loss, latency, and Internet congestion. As a result, Move Adaptive Stream out-performs traditional streaming and progressive streaming.

- Move Adaptive Stream enables publishers to provide viewers with DVD player-like controls and PVR navigation, also known as TiVo™-style navigation and access.

- Live and on-demand streaming benefit from Move Media Player's "time-shifted" playback; users can tune into live TV broadcasts or watch programming which aired previously using a common interface. While viewing favorite programs, users are able to pause live broadcasts. They can also rewind, scrub, and watch in slow motion.

- Video streams are indexed and stored in the Move Publish content management system, making the creation of live or on-demand playlists a simple process. Once ingested into the Move Publish content management system, any number of video segments (or sub-segments) from any number of archived or live video streams can be seamlessly delivered as a single, continuous stream using virtual timeline; this is accomplished without re-encoding or using special editing software.

- For Internet streaming, viewers can tune in late for a live broadcast and either commence watching at the current live time or watch the broadcast from the beginning. It is the same stream in either case, and viewers may find that they can watch portions of the broadcast, skip parts, and eventually catch up with the live presentation.

- Move Publish provides easy mechanisms for using video for viral spreading of content. For example, end users can make playlists from larger content archives, insert video segments in blogs or Web pages, or easily share a favorite video clip with friends or associates.
- The system can scale to millions of users because of Move Adaptive Stream's efficient use of existing Internet caches and related technologies—creating a streaming technology that is scalable.
- Publishers can elect to use the Move Networks Digital Rights Management (DRM) system to guarantee safe viewing and monetization of content.
- Content publishers can view detailed statistics using the system's extensive network, quality, and usage monitoring capabilities.

These and other features are part of the streaming technologies provided by Move Networks. The following sections explore the problems inherent in traditional streaming and how Move Adaptive Stream is able to overcome these challenges and offer a cost-effective solution for real-time as well as on-demand delivery of video content over the Internet.

As television providers improve navigation with PVR-like functionality, as well as capabilities such as On-Demand and Pay-per-View, Internet streaming gets further behind.

The Problem with Traditional Streaming Video

With the rapid rise of the Internet and the extensive adoption of broadband technology by home users, it is tempting to envision the Internet as an ideal distribution medium for videos, movies, and television programming.

Unfortunately, anyone who has tried this knows that current streaming technologies are not up to the challenge. Delivering video over the Internet is fraught with problems—including delayed starts, poor video quality, and long buffering interruptions. Additionally, the size of most streaming video frames is too small to satisfy users who are accustomed to large-screen, high-definition television sets. Even as television providers improve navigation with PVR functionality, as well as capabilities such as On-Demand and Pay-per-View, Internet streaming falls further behind.

A study conducted by StreamGuard.net examined more than 2,400 different Internet video streams. They concluded that:

- 51% of stream viewers were frustrated with the experience.
- 17% of streams had an immediate fatal error.
- 25% of streams took between 5 and 10 seconds to start playing, and they stopped to rebuffer before the video was finished.

With current technologies, dial-up users do not typically attempt video streaming, and broadband users often give up after prolonged waits or unresponsiveness due to regular rebuffering.

Clearly, streaming video content over the Internet is problematic, and the weaknesses inherent in current streaming technologies pose many challenges for Internet-based streaming content providers. Consider the following problems with traditional streaming technologies:

- Traditional streaming requires synchronization between a media server and the client-based player. In order to have a high-quality playback experience, a significant amount of data must be transferred per unit of time. The protocols (e.g., RTP/RTSP) currently deployed to manage this high transfer rate have not yet demonstrated the flexibility and responsiveness needed to maintain synchronized communication in the face of Internet congestion, latency, and packet loss. The current workarounds encourage the content publisher to reduce the image quality or shrink the frame size, or they force the user to experience long start and seek times so that the client-based player has time to buffer content. While these workarounds improve the situation by reducing the volume of data sent or allowing the player to get ahead, the viewing experience is severely sacrificed.

- The process of encoding video files for streaming over the Internet is extremely complicated and demands extensive CPU power. This can be very expensive in terms of equipment, resources, and personnel. Often, content publishers are forced to limit quality in order to support live streams or to encode on-demand streams targeted at a bandwidth-diverse customer set.

- Streaming live video feeds is problematic. Internet streams created from live video feeds are usually fraught with glitches and interruptions. Once again, image quality, size, and features are sacrificed because of limited encoding CPU resources and the absolute need to keep up with the live feed.

- Current streaming technologies do not provide the degree of playback control that users are accustomed to with DVD or television viewing. Users want the kind of flexibility that a PVR or DVD player provides, including the ability to fast forward, rewind, scrub, timeshift, or play in slow motion.

Users want the kind of flexibility that a PVR or DVD player provides.

- Current streaming solutions are server-centric. They require a specialized media server (such as the Microsoft Media Server) using specialized protocols (such as RTP and RTSP). IT managers are not able to treat video streaming like other data types (Web objects, graphics, Web pages, etc.) and thereby leverage the significant investment already made in Web servers and related technologies. In addition, streaming clients must establish direct network connections with the media server, which limits scalability. A single media server has a fixed number of streaming sessions it can support; to add users more servers must be added in a linear fashion. Maintaining the client-server connection requires paying for high-cost bandwidth concentrated at the server site.

- Unlike standard Web pages and graphics, video streaming isn't cacheable by Web cache servers on the network or by the browser on a local hard drive. Once the stream has completed, the data are gone. The client must directly contact the media server and download the streaming file every time the user wants to view a given video. This process is very inefficient and negatively affects the scalability of the entire system.

- Content owners do not feel comfortable streaming highly valuable content due to the potential for content theft. The available digital rights management (DRM) solutions are very complicated and require significant resources to implement.

Video downloads and progressive streaming provide one approach for resolving some of the issues addressed above. However, as this paper shows, these approaches do not solve all of the problems, and they create new problems for the content publishers and viewers. For example, progressive streaming creates several new problems: 1) the user must wait to view content until the player determines that enough of the content (all of the content, in the case of pure download) has been downloaded to allow for reliable viewing; 2) the user cannot seek to content that is ahead of the current download point; and 3) if the user abandons viewing, the content publisher may be paying to download bits that are never viewed. In fact, if the user decides to abandon 2 minutes' worth of downloaded content after only 30 seconds of viewing, 75% of the bits are wasted. Users with broadband connections make that waste worse by downloading at a higher speed-to-playback time ratio.

The reason traditional streaming doesn't work very well is that it attempts to do the impossible.

The points listed above show that the reason traditional streaming doesn't work very well is that it attempts to do the impossible—traditional streaming tries to use a server-centric synchronization protocol to tame the variability of the Internet. The communication between client and server is simply not timely enough to deal with the inconsistencies of Internet connectivity; this fact will remain even as the Internet grows in size and scope or as bandwidth is increased. The balance of this paper discusses the next generation in Internet streaming—Move Adaptive Stream.

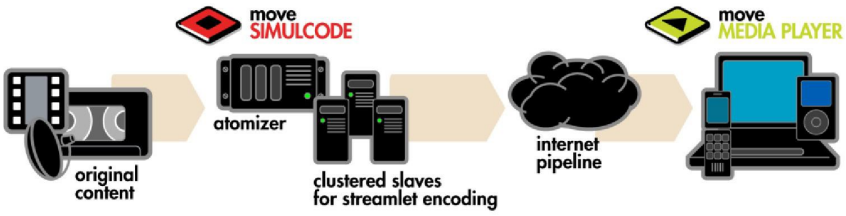
The Solution: Move Adaptive Stream

Move Networks, Inc. has developed streaming technologies that utilize a different protocol than existing technologies for delivering and experiencing video content over the Internet. An adaptive and effective solution, Move Adaptive Stream is designed to work despite the Internet's frequent problems of high latency, packet loss, and congestion.

Move Adaptive Stream is the most adaptive and effective solution available anywhere and the only solution designed to work despite the Internet's frequent problems of high latency, packet loss, and congestion.

How Move Adaptive Stream Works

The technologies leveraged by Move Adaptive Stream differ greatly from traditional streaming technologies by breaking up large video files into many small files called streamlets. Streamlets are multi-second video segments that are delivered using a highly efficient transmission protocol. While traditional streaming solutions are stored on specialized media servers, such as the Microsoft Media Server or Real Network's Helix streaming server, streamlets are stored on standard HTTP Web servers. When an Internet user requests a video (which is done using a standard HTTP "GET" request), streamlets are transferred over the Internet from the origin Web server to the client's Web browser or client application where they are reassembled in the correct order. The delivery protocol uses multiple TCP sessions to optimize the transmission and the carrying capacity during each unit of time. Using streamlets allows a video stream to transfer much more efficiently and effectively through the Internet infrastructure. It also protects against stalls or other glitches.



By way of analogy, traditional streaming could be compared to a one-lane highway (a single TCP session) during rush hour traffic (a large amount of data traveling between server and client). However, Move Adaptive Stream could be compared to a multi-lane freeway where traffic stalls in one lane can be minimized by changing lanes, significantly expanding the aggregate traffic capacity. To carry this analogy one step further, because Move Adaptive Stream puts the intelligence in the client, the client functions as the "driver" and makes instantaneous decisions to speed up, slow down, or change lanes to minimize delays and maximize viewing experience. On the other hand, traditional streaming would force drivers to report their status and wait for a traffic controller (the media server) to give instructions to speed up or throttle back.

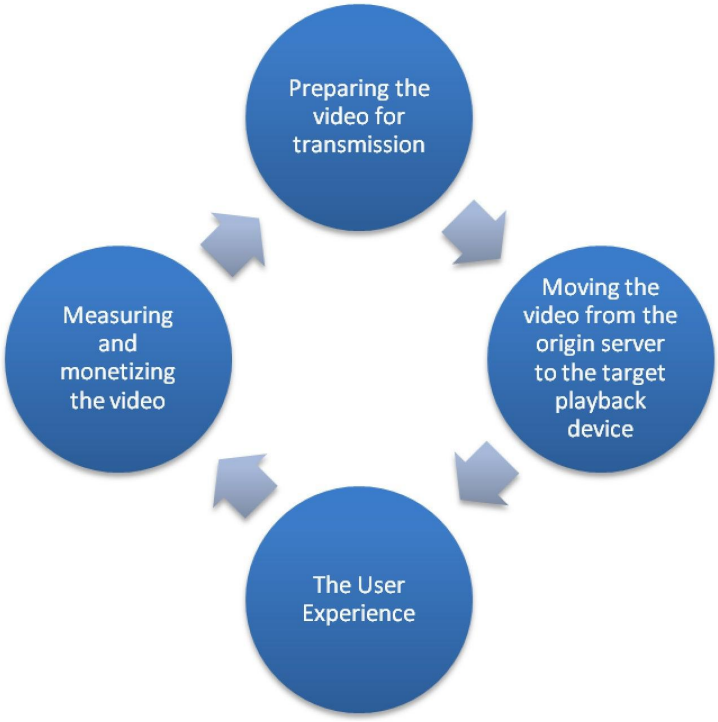
Traditional streaming would force drivers to report their status and wait for a traffic controller (the media server) to give them driving instructions.

The Move Networks Publishing System is discussed in greater detail in the following sections.

The Advantages of Move Adaptive Stream

Move Adaptive Stream offers dramatic advantages over traditional streaming technologies. It provides more than a new protocol for streaming; the Move Networks Publishing System provides a better way to publish and experience video on the Internet. The advantages can be grouped into five categories that correlate with the streaming process:

- **Encoding**—preparing the video for transmission
- **Transmission**—delivering the video from the "origin server" to the target playback device
- **Client Controls and Playback**—the user experience
- **Scalability**—accommodating large numbers of users
- **Publisher Feedback and Administration**—measuring and monetizing the video



Move Simulcode is not held hostage by limited CPU resources.

The advantages of Move Adaptive Stream during the encoding process are discussed first.

Encoding Advantages: Move Simulcode

Streaming begins with preparing the video for delivery over the Internet. The publisher makes many decisions at the outset that will significantly impact the viewing experience. If the encoding process is not fully functioned and flexible, the publisher is forced to limit the viewing experience.

The traditional process of encoding or rendering a video file for viewing is an intensive process that requires a large amount of CPU power. As such, it is very expensive and relatively inflexible. The Move Networks Publishing Platform implements Simulcoding, a process of hardware clustering and software pipelining, to achieve complete flexibility and maximum quality. Unlike standard encoding for live streaming, Move Simulcode is not held hostage by limited CPU resources.

Standard live encoding is limited by available CPU power. For live encoding, there is a fixed-duration encode sampling window, wherein the encoding software must analyze and encode the video according to the parameters set by the publisher. The stream quality will always be constrained by what the server can accomplish during this fixed sampling window; if server resources are exhausted, the encoding algorithm must downgrade quality to enable the process to keep pace.

Move Simulcode breaks this CPU barrier by implementing hardware clustering and software pipelining. While standard streaming technologies have to live in a "stream-of-consciousness

mode," Move Simulcode creates a parallel processing architecture, where a pipeline is defined as all of the processing needed for each video streamlet. The following workflow will help to clarify the pipeline associated with each streamlet:

1. A live feed, or a watch folder containing video files, is connected or associated with the Move Media System.
2. The Move Simulcode master controller pre-samples video in real-time into smaller video segments called streamlets (1 to n seconds of the live stream).
3. The start of a pipeline begins when each streamlet is assigned to one of the clustered servers.
4. The clustered server processes the streamlet and applies the publisher-determined parameters (bit rate, frame size, frame rate, codec type, constant or variable bit rate, 1-pass or 2-pass encoding, etc.) bit rate by bit rate.
5. All target bit rates are output as separate small files and stored on a standard Web server (no media server is needed).
6. At the completion of the work for this pipeline, the Move Simulcode master controller continues to pre-sample and assign successive streamlets to available clustered CPUs. A cluster is composed of 1 to n CPUs that take assignments from the master CPU for encoding.

Publishers can easily target cell phones, laptops, desktops, or even large screen TVs within a single process and with the optimum stream.

As illustrated above, every 1 to n seconds a new raw streamlet is assigned by the master controller to a new pipeline that runs in parallel with preceding pipelines. A publisher may decide that its target audience will be viewing a stream using a wide variety of devices with varying bandwidth. This may lead to a publisher-defined profile specifying a low-end bit rate of 50 kbps, an upper-end bit rate of 1.5 Mbps, with numerous bit rates in between. For each bit rate, the publisher can specify frame size, frame rate, codec type, constant or variable bit rate, 1-pass or 2-pass encoding, etc. There is no need for the publisher to worry about server-based ceilings. If the cluster's maximum output is not fast enough, more servers are added to the cluster.

Bit rates can change quickly and seamlessly as network conditions fluctuate.

No other encoding process offers publishers the high level of flexibility that Simulcoding allows. With Move Simulcode, publishers can rely on a single process to both encode and optimize live, simulated-live, or on-demand streaming to a wide variety of end-user devices, including cell phones, laptops, desktops, or even large screen TVs. The result: cell phone users, desktop users, and laptop users can be watching live television with an appropriately optimized video stream.

The output of this ingestion process is a set of streamlets for each bit rate specified in the profile. Once created, the streamlets are then stored on an industry-standard Web server, requiring no modifications or additional software. As the client protocol needs to upshift or downshift the bit rate, the correct time-indexed streamlet from the appropriate bit rate set is retrieved from the server. There is a streamlet of equal play length for every bit rate specified in the profile. Therefore, the player can easily interchange bit rates by retrieving the appropriate time-indexed streamlet from the desired bit rate pool. Thus, bit rates can change

quickly and seamlessly as network conditions fluctuate, and because each streamlet is relatively small, seeking and starting are quick and efficient.

Move Simulcode ingests and outputs a video file as a collection of streamlet files of equal duration for each bit rate, allowing the ingestion process to perform certain advanced encoding operations that are not possible with traditional systems.

Consider just one of the many unique features. Dual-pass variable bit rate (VBR) encoding is possible for live Internet TV broadcasts. This is not possible for live streams in any other system because dual-pass techniques require the encoding process to examine the entire video stream once and then apply encoding on the second complete pass-through of the content. For live broadcasts, there is simply no possible way to sample the entire broadcast once for analysis and a second time for encoding. However, Move Simulcode only deals with a few seconds of video during each encoding pass and permits these video files to benefit from clustered hardware and software pipelining. The process can keep pace with a live broadcast as long as sufficient servers are applied to the encoding process. Therefore, it is now possible to apply the needed resources to maximize quality with a dual-pass VBR encode on live broadcast streams.

Variable bit rate encoding greatly improves the quality of the streaming by applying a higher number of available bits to scenes with high motion and fewer bits to scenes with little motion or low complexity.

VBR encoding greatly improves the quality of the streaming by applying a higher number of available bits to scenes with high motion and fewer bits to scenes with little motion or low complexity. The result is a sharp, near-HD quality video picture.

With this in mind, the delivery advantages associated with Move Adaptive Stream are discussed next.

Transmission Advantages

The Internet experiences intermittent network congestion problems, which are manifest by increased data packet delivery latency (time delay) and outright packet loss. Problems can be created at every point in the delivery path, from an in-house WiFi (radio interference), to shared cable (as neighbors start downloads), through the various Internet backbone routers, and up to the origin server. These congestion issues are systemic to all Internet communication, and Internet protocols have been designed around these characteristics; for example, Internet protocols are designed to implement best effort delivery and to drop and delay packets when congestion occurs.

TCP, the protocol used by most Internet traffic, is designed to slow down and back off when it detects lost packets. This is how the Internet deals with congestion problems. Because of this strategy, users will notice that identical Web pages load at times very quickly and at other times very slowly. This type of unpredicatble latency and erratic availability of bandwidth is extremely destructive to video streams and prevents them from displaying content smoothly. Because of these TCP design characteristics, UDP—another Internet transport protocol—would normally be preferred over TCP for streaming video. However, due to firewall and NAT issues, TCP is used by almost all streaming deployments, traditional or progressive. Therefore, TCP and its inherent design limitations have a major impact on video delivery.

TCP-based media servers typically rely on RTP/RTSP to control video flow. The streams served get "stuck" because the protocol is not responsive enough to keep pace with common Internet communication issues discussed above—i.e., packet loss, latency, and congestion. During video streaming the client is constantly communicating the status and health of the communication link. The client will forward statistics on such things as packet loss and latency. With this information, the media server makes decisions concerning whether to throttle the bit rate up, down, or drop into emergency "slideshow" mode. In practice, these mechanisms do not work well because the client-server communication overhead creates additional latency problems and TCP slows down when network congestion is detected. As the decision maker, the server must respond with a change in bit rate, but in order to avoid over-responding, the server waits for several problems to occur in close proximity. This slowness in server response, coupled with the fact that the server and protocol-precision are not sufficient to deal with the fluid nature of Internet traffic, leads to playback glitches or even stalled communication sessions.

Another major issue that plagues both traditional streaming (RTP/RTSP) and progressive streaming is the use of a single TCP connection. Using only a single TCP connection leaves the process susceptible to single failures, as the previously-mentioned highway analogy suggests. A single TCP connection that has experienced slight packet delay or loss causes the entire stream to temporarily freeze or severely reduces connection efficiency.

Move Adaptive Stream immediately adapts to network conditions by retrieving the most appropriate bit rate from among the streamlets stored on the Web server during the ingestion process.

Move Adaptive Stream mitigates the reliability, efficiency, and latency issues of traditional streaming. The secret lies in the use of multiple TCP connections between the client and the origin Web server and the use of client-side decision-making to retrieve the streamlets with the optimum bit rate. Rather than trust the media server to throttle bit rates up or down, Move Adaptive Stream immediately retrieves the bit rate that best matches current network conditions.

Move Adaptive Stream establishes multiple network connections with the origin server (a process called *virtual pipelining*) and transfers multiple streamlets at a time (a process called *parallel retrieval*). By doing this, reliability is increased, efficiency is raised to near maximum, and latency is virtually eliminated. Video continues to flow since it is highly unlikely that all TCP connections will be backing off, slowing down, or stalling at the same time. Because multiple responses are always being transmitted, the latency of each new request (after the initial first request) does not affect performance.

The client software is always monitoring streamlet retrieval success to ensure that the appropriate data stream rate is used. The client's goal is to retrieve streamlets fast enough to present the streamed content in real time with no pauses.

Because the client can directly select and retrieve data from an array of chronologically congruent streamlets that support different encoding rates, there is no need to inform or coordinate with any media server to change the data it is sending. This innovative flexibility in the streamlet architecture allows the client to quickly adapt quality to changing network performance.

Virtual pipelining opens up many new possibilities for how the user controls the video stream.

Client Control and Playback Advantages

The ability of the client to directly select among any available streamlets has other substantial implications. First, changing the time position for playback is trivial. The client simply requests streamlets that correspond to the content's new time index. This same advantage also translates to fast forwarding, rewinding, and scrub playback.

Second, the first streamlets retrieved when changing position can be of a lower quality (and therefore faster to retrieve) than the streamlets currently being played. This results in a shorter lapse of time before the user begins to see the content at the new time point. And yet, it is virtually unnoticeable to the user. After content begins playing again, the bit rate increases back to the highest quality possible for the viewer's bandwidth connection.

The streamlets downloaded by the client are locally cached on the user's computer (unless publishers, for security reasons, do not want content stored locally or want to minimize local storage). Not only does this reduce latency, but it also provides playback advantages. The cached streamlets can be reused if the user requests a change in playback position (such as a rewind or scrub bar movement).

Locally caching streamlets facilitates other innovative features. When a publisher creates a virtual timeline playlist, streamlets can be delivered far in advance of the current playback position and saved in the local cache for quicker, more efficient display.

If the user wants to fast forward, the player finds content preloaded at defined intervals from which it can provide an instantaneous change in playback position.

Extending this ability further, a Web page, script, or database can be coded to provide the player with specific index points within the content. The player can also be provided with other content relevant to the current position of the video being played.

The moment the user makes a selection, the new content begins playback immediately.

Intracontent index points can include such things as scene changes, sports plays, or news stories. Related content might include other episodes of the same program. It could also include searches by such metadata as date, subject, or cast member. It may also include chronological time lines. The player can preload the first streamlet from each index point and cache it locally. When the user selects a new index point, the new content begins playback immediately.

Perhaps one of the most significant capabilities of downloading and dealing in streamlets is the ability to pause a live stream. With traditional streaming, the need for tight synchronicity, coupled with the issues described above relating to media servers, prevent pausing, rewinding, or scrubbing live streams. All of these features are now possible using Move Adaptive Stream. What is the result in terms of overall user experience? High-quality video enhanced with PVR-like controls, ease of navigation, and the flexibility of the Internet.

The use of streamlets also provides another distinct advantage: enhanced scalability.

Scalability Advantages

Because Move Adaptive Stream does not use a specialized media server, or any kind of control connection with any entity outside the client, it is much more scalable than traditional streaming technologies. This is because streamlet files are simply that—files.

Streamlets are static by nature. Retrieving streamlets involves nothing more than asking the Web server to return them using standard HTTP requests. No custom application programming at the Web server is required. No dynamic content is created.

Because streamlet files are static objects, they are cacheable by standard Web cache servers according to HTTP standards. If a streamlet file is infrequently requested, then the Web cache server soon eliminates it from its cache. However, if it is frequently requested, it may stay cached indefinitely.

It is important to underscore that because streamlet files are static objects, they are not only cacheable, but assumed cacheable according to HTTP standards.

Any cache server in the network path between the origin server and the client will cache streamlets so that any near-term reselection of the same content is served from the nearest cache server and not the origin server. This benefits the origin Web server by eliminating redundant requests.

While the streamlet caching architecture provides many benefits for single users, there are also considerable benefits for multiple users of the same content. A unique characteristic of Web caching is that the more users that access the same content, the more scalable the system becomes. This is because more of the content is cached by multiple cache servers, thereby reducing the load on the origin server. Whenever there are network paths common to multiple users consuming recently used content, there is a huge increase in scalability.

In addition to increasing transmission and caching scalability, Move Adaptive Stream also provides content publishers with the ability to monitor and analyze the system's overall performance. Move Adaptive Stream's administration advantages are discussed next.

Publisher Feedback and Administration Advantages

Traditional streaming technology fails to give the publisher feedback about the experience the user actually had while watching a particular video. These statistics are inherently session based. Actual user behavior is estimated from server logs Consequently, the publisher may be paying to stream content that is rarely or never watched. Alternatively, the publisher may have a video that continually fails to start or content that rebuffers continually. But the publisher receives none of this information when viewer-based tracking is not available.

The more users that access the same content, the more scalable the architecture.

With Move Adaptive Stream's client-side control, the publisher can extract useful feedback such as how many users watched which videos and how long they watched. The client records quality issues related to start times, pauses, stalling interruptions, and delivery speed. Advertisers also benefit from knowing the demographic of the viewership and knowing how many times their ads were viewed, rather than knowing only a count of how many times the server delivered an ad to the client. Furthermore, feedback reports can be received in near-real time to allow for the quick remedy of any of the quality issues mentioned above (start times, stalling, bit rates, etc.).

We must emphasize how revolutionary this feature is to the video publishing industry. It's akin to the Nielsen ratings for television. Without this viewership information, a publisher would never know how many people are watching a show (or if anyone is watching at all). Armed with this critical information, the publisher now has true monetization power—the power to make educated decisions about content and delivery quality. These reported numbers can be used to improve the product, satisfy the customer, and increase revenues, as guaranteed targeting leads to higher CPMs and niche audiences. With Move Adaptive Stream, a publisher

can be in the mode of expanding revenue and monetization rather than in the cost-centric mode of restraining usage.

Virtual Timelines

Move Adaptive Stream provides for playlisting and the integration of multiple on-demand and live streams into a single, continuous stream. Video streams are indexed and stored in Move Adaptive Stream's content management system, making the creation of live or on-demand playlists quick and easy. Once ingested into Move Adaptive Stream's content management system, any number of video segments (or sub-segments) from any number of archived or live video streams can be seamlessly delivered as a single, continuous stream using a virtual timeline; this is accomplished without re-encoding the video or using any special editing software. Consider the following possibilities:

- Content viewers maintain a playlist of their favorite video clips and share these with friends and associates. As these clips are shared, only a link to the content is transferred, and the software is able to retrieve and play the specified video segment(s) using Move Adaptive Stream's index points and URLs.

- A small television station has 20 pieces of content they want played as a group, back to back. The station creates a virtual timeline playlist that specifies each element in the order desired; no additional preparatory work is required. The station may choose to anchor the timeline's relative time to real time, such that playback seems more like live (e.g., if an episode is scheduled to air at 7:00, and a user starts viewing the episode at 7:10, the user will miss the first 10 minutes of content, creating the feel that the user is joining a live program already in progress.). Or the station may choose to start play at the beginning of each video segment.

- A larger television station has a lineup that contains pre-recorded programming, live coverage of tonight's football game, followed by more pre-recorded programming. There is no way to predict the precise time the game will end. However, if the publisher has created a virtual timeline playlist containing both the pre-recorded and live stream URLs, the Move Media System will seamlessly manage the streaming operation. The publisher does not need to stand by and manually make the switch back to pre-recorded content after the football game ends because Move Adaptive Stream simply follows the URLs in the virtual timeline playlist.

- Program metadata can be stored in the virtual timeline file or in a separate file that is hosted on the Web server. Custom metadata are available during streaming and also can be downloaded by clients just-in-time. This allows last-minute updating and live altering of custom metadata.

The virtual timeline's new indexing, scripting, and playlisting capabilities make using video easy, powerful, and much more affordable than the infrastructure it would require to encode and edit the video content using a traditional streaming solution.

Digital Rights Management

When content is ingested it is broken into streamlets, as described above. These are stored on a secure server for playback. Using Move Adaptive Stream's DRM system, an additional layer of encryption can be added to further protect the content against theft or unauthorized use. The DRM system uses a combination of asymmetric and symmetric encryption schemes to protect the content. A complete description of the DRM technology is beyond the scope of this paper. However, the following represents a brief list of what publishers can do:

- Authorize users to view a stream once or an unlimited number of times.
- Permit unlimited viewings but restrict viewings to a particular machine, a region of the world, or a limited period of time.
- Permit viewing but not copying, or allow only one copy with a timestamp that prevents viewing after a certain time.
- Charge per view, per unit of time, or per episode.

The possibilities are endless because of the encryption technologies in Move Networks' Digital Rights Management system. The Move Adaptive Stream DRM system maximizes content, enables monetization of content, and implements both encryption schemes and business rules for when and how content can be seen.

The Digital Rights Management (DRM) system enables publishers to maximize and monetize their content.

Summary

In conclusion, Move Adaptive Stream represents a significant shift in streaming technology:

- Streaming is no longer plagued by constant rebuffering, slow starts, and poor video image quality.
- Move Adaptive Stream enables publishers to provide viewers with DVD player-like controls and PVR navigation, similar to TiVo™ navigation and access. Live and on-demand streaming benefits from Move Adaptive Stream's "time-shifted" playback. In addition, it is now possible to pause, rewind, or scrub live streams.
- A virtual timeline allows for powerful automation and playlisting.
- The system can scale to millions of users because of Move Adaptive Stream's cacheable streamlets and streaming protocol.
- Publishers can elect to use Move Adaptive Stream's Digital Rights Management (DRM) to keep content safe and maximize monetization.
- Content publishers can view detailed statistics using the system's extensive network, quality, and usage monitoring capabilities, thus always working to improve broadcasts.

With Move Adaptive Stream, the Internet now becomes a viable streaming option for distributing movies, television programming, and other video-based content at quality and reliability levels consistent with traditional television broadcasting.

Appendix A

Putting Move Adaptive Stream to the Test

The following test measures how the various video delivery protocols respond to different levels of simulated network congestion problems. This is done by adding network latency and random packet loss. During these tests, conditions are kept constant in order to isolate and precisely measure their effect on the protocols. In the real world of the Internet, these conditions come and go; however, the way they will affect video delivery is the same.

For progressive delivery, 3 numbers are given for 3 types of content or Web sites:

- Vividas (750 kbps encode)
- Apple Quicktime trailers (800 kbps encodes)
- CBSNews (480 kbps encodes)

For RTP or RTSP delivery:

- Video.MSN.COM (220 kbps encodes)

For streamlet delivery:

- BYU.TV (variable bit rates, 150 to 677 kbpss)

Effective Bit Rate (percent of possible, averaged over 5 test runs)

800 kbps home DSL with client-side added latency (in milliseconds), packet loss

Protocol	No added	50ms 0%	50ms 3%	50ms 5%	100ms 0%	100ms 3%	100ms 5%
RTP/RTSP (UDP)							
RTP/RTSP (TCP)*	100%	78%	66%	39%	63%	38%	29%
Progressive (TCP) Vividas	98%	80%	41%	35%	82%	30%	19%
Apple	92%	63%	31%	26%	57%	22%	16%
CBS	93%	68%	31%	26%	39%	29%	19%
Move Adaptive Stream (TCP)**	100%	91%	83%	68%	82%	60%	49%

*On several of the test runs, the protocol went to 80 kbps slideshow mode.

**On several of the test runs the protocol stayed around 320 kbps.

move networks, inc.

Start Delay Time (in seconds, averaged over 5 test runs)
800 kbps home DSL with client-side added latency (in milliseconds), packet loss

Protocol	No added	50ms 0%	50ms 3%	50ms 5%	100ms 0%	100ms 3%	100ms 5%
RTP/RTSP (UDP)							
RTP/RTSP (TCP)	8 sec	11 sec	21 sec	48 sec	22 sec	29 sec	57 sec
Progressive (TCP) Vividas	4 sec	5 sec	7 sec	7 sec	5 sec	7 sec	10 sec
Apple	8 sec	48 sec	242 se	290 se	119 se	358 se	720 se
CBS	9 sec	12 sec	25 sec	28 sec	22 sec	31 sec	48 sec
Move Adaptive Stream (TCP)	3 sec	3 sec	3 sec	4 sec	3 sec	4 sec	5 sec

Playback Pauses After Start (percent of time waiting during 1 minute, averaged over 5 test runs)
800 kbps home DSL with client-side added latency (in milliseconds), packet loss

Protocol	No added	50ms 0%	50ms 3%	50ms 5%	100ms 0%	100ms 3%	100ms 5%
RTP/RTSP (UDP)							
RTP/RTSP (TCP)	0%	5%	23%	22%	18%	18%	6%
Progressive (TCP) Vividas	1%	38%	62%	74%	34%	78%	83%
Apple	13%	28%	17%	12%	13%	32%	14%
CBS	15%	25%	32%	30%	25%	30%	45%
Move Adaptive Stream (TCP)**	0%	0%	0%	0%	0%	0%	0%

*There is a relationship between Start Delay Time and Playback Pauses. In general, the larger the Start Delay, the less Playback Pauses.
**The protocol always decreases the video bit rate before a pause is required.

Watchability (subjective, function of start delays, playback pauses, and general viewing experience)
800 kbps home DSL with client-side added latency (in milliseconds), packet loss

Protocol	No added	50ms 0%	50ms 3%	50ms 5%	100ms 0%	100ms 3%	100ms 5%
RTP/RTSP (UDP)							
RTP/RTSP (TCP)	Yes	Yes	No	No	No	No	No
Progressive (TCP) Vividas	Yes	Yes	No	No	Yes	No	No
Apple	Yes	Yes	No	No	Yes	No	No
CBS	Yes	Yes	No	No	No	No	No
Move Adaptive Stream (TCP)	Yes	Yes	Yes	Yes	Yes	Yes	Yes

DISH TECHNOLOGIES L.L.C., Exhibit 2069, p. 19
fuboTV MEDIA INC., and Yanka Industries, Inc., v. Dish Technologies L.L.C., IPR2024-00919

Testing Summary & Conclusions

Move Adaptive Stream and RTP/RTSP work best under variable network conditions because they both have the capability of changing the video bit rate during playback, whereas Progressive does not. All protocols work better when the video bit rate is significantly less than available network bandwidth. Move Adaptive Stream and Progressive take advantage of this condition to get way ahead of playback (Move Adaptive Stream stays 20 seconds ahead, and Progressive doesn't stop until the end of content); Microsoft's RTP/RTSP uses additional bandwidth to decrease startup buffering time. Both Move Adaptive Stream and RTP/RTSP use extra network bandwidth to upshift to higher bit rates (if available).

As these tests indicate, Move Adaptive Stream is superior to RTP/RTSP and Progressive downloads in all aspects of video delivery: start and seek times, susceptibility to playback hiccups, and timely delivery of the maximum amount of data (yielding higher quality video). The superiority of Move Adaptive Stream is especially clear and important during network congestion problems (all protocols work flawlessly on a fast, perfect network). At the fundamental level, Move Adaptive Stream works better because it can almost always seamlessly adapt to changing Internet conditions and the others often cannot.

Progressive usually requires the viewer (Apple, Vividas) to select the video bit rate (high, medium, or low) and then deal with the consequences of that choice (long start time and pauses during playback if too high; lower quality video if too low). In a few Progressive implementations (CBS News), the player software selects the video bit rate from past history, which is often too high or too low for current conditions. Progressive uses TCP, which can significantly slow down under network congestion.

RTP/RTSP will decrease the video bit rate according to current network conditions, but typically only after a number of playback problems. These decisions are made by the server, but lag because they are from reports sent from the client. RTP/RTSP will wait a long time before increasing the bit rate because, since data is just sent at "rate," it has limited visibility into improved network conditions. Since RTP/RTSP is usually sent over TCP, it suffers from the TCP slowdown problems.

The Move Adaptive Stream client monitors current network conditions and can upshift or downshift bit rates every 2 seconds, if needed. It also largely overcomes the TCP slowdowns by using multiple TCP connections (not all connections will slow down at once, which keeps data flowing).

Streaming video over the Internet today has significant problems because of inherent and fundamental limitations in the existing video delivery protocols. These problems won't go away by deploying more or faster broadband; in fact, network congestion will get worse the more video is streamed. Only Move Adaptive Stream seamlessly adapts to whatever the reality of the Internet is at any time. Move Adaptive Stream consistently delivers the highest possible quality viewer playback experience: fast starts and seeks, virtually no pauses or video glitches, along with the highest quality video from the highest bit rate encodes. Move Adaptive Stream provides Internet video streaming quality that is on a par with regular TV, plus the functionality and features of the Internet.

DISH_337-TA-1265_00033667

DISH TECHNOLOGIES L.L.C., Exhibit 2059, p. 19
Aylo Freesites Ltd. v. Dish Technologies L.L.C., IPR2024-00045

move networks, inc.

Appendix B

Feature Comparison Table: Move Adaptive Stream versus Other methods

	Move Adaptive Stream	RTP/RTSP	Progressive
Auto bit rate determination at start	Y	Y	N
User not required to select bit rate	Y	Y	N
Quick start under all conditions	Y	N	N
Adjusts bit rate	Y	Y	N
Adjusts bit rate without playback pauses	Y	N	N
Playback always at highest available bit rate	Y	N	N
Avoids getting too far ahead, wasting bandwidth	Y	Y	N
Rewind	Y	N	Y
Fast Forward	Y	N	Y (only after down-loaded)
"Scrub" navigation	Y	N	Y (only after down-loaded)
Scalable to High Def (encodes > 2 megabits)	Y	N	N
Live highest quality 2-pass variable bit rate encoding	Y	N	N
Live time shifting	Y	N	N
Cost savings: Delivery from generic Web servers	Y	N	Y
Cost savings and scalability: uses of proxy caches	Y	N	Y
Can stream live video streams	Y	Y	N
Can pause, rewind, and scrub live streams	Y	N	N
One process for ingesting on-demand and live video	Y	N	N
Ability to insert advertisements and mark metadata without additional products or add-ons	Y	N	N

DISH TECHNOLOGIES L.L.C., Exhibit 2059, p. 20
Aylo Freesites Ltd. v. Dish Technologies L.L.C., IPR2024-00045

Move Video Services SDK

Move Networks provides an SDK that includes multiple APIs for the design and implementation of the video experience. This SDK includes all of the scripts and functionality required to provide a rich and immersive experience to the end user. Move Networks provides additional implementation guidance and training around the use of the SDK.

Move Networks Integration for Typical Network Broadcasting Station

The final diagram provided below is for a typical local broadcasting channel to understand the required system components and functions to deliver their video content online. The diagram represents systems and processes rather than physical infrastructure needs (such as hardware requirements, etc.).

