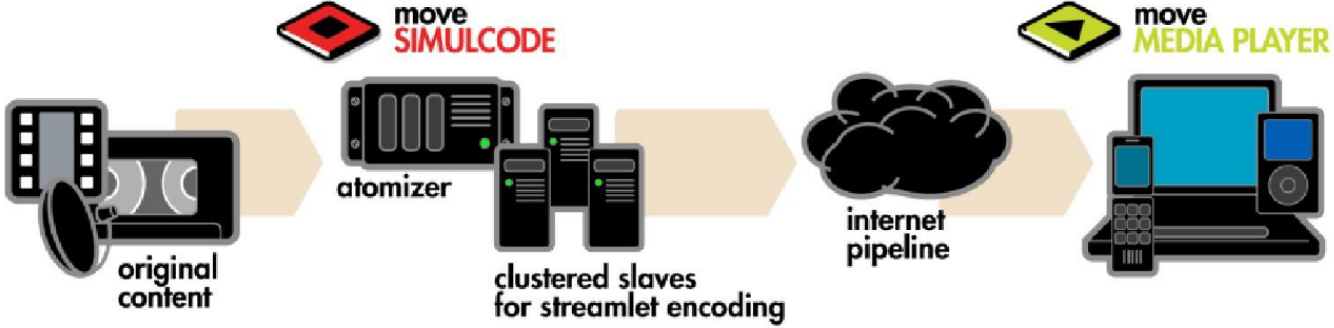


The following claim chart shows exemplary aspects of certain streaming services and products provided by MOVE Networks, Inc. (“MOVE”) that practice the limitations of claim 8 of the ’564 Patent. The chart is exemplary and should not be read to limit MOVE’s practicing of the below-listed claim or any claim with respect to the services or products described below, including MOVE’s Move Media Player (“Media Player”). Nor should this chart be read to limit the claims practiced by MOVE. Additionally, this chart should not be read to limit the acts that infringe the below claim.

The images in this chart are from MOVE’s “Technical Overview of Move Networks Implementation,” labeled as EX2022, MOVE’s “Adaptive Stream,” labeled as EX2077, or “The Next Generation Video Publishing System,” labeled as EX2069.

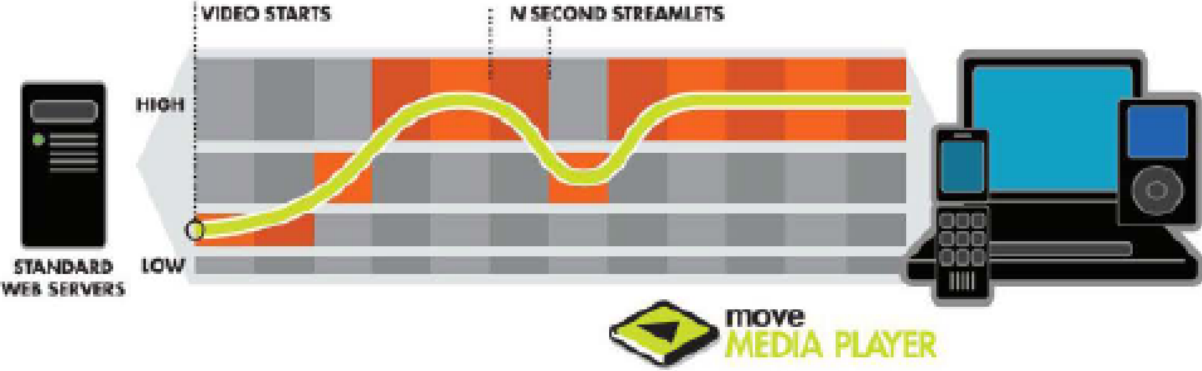
Claim Element	Example Infringement Evidence
[8.pre] A method executable by an end user station to present rate-adaptive streams received via at least one transmission control protocol (TCP) connection with a server over a network, the method comprising:	MOVE’s Media Player operated within a system that operates on an end user’s content player device (“End User Station”) to stream live and on-demand video over a network from a server for playback of the video. Indeed, MOVE’s “Adaptive Stream enable[d] publishers to provide viewers with DVD player-like controls” to “all target platforms (cell phone, laptop, desktop, large screen TV) and bandwidth profiles (dial-up, wireless, mobile, broadband).” EX2069 at 1. For example, “Move Networks provides and operates the premier video distribution platform.” EX2022, at 4. “The architecture of the move platform enables a revolutionary viewing experience giving viewers control over how they watch and interact with the video content. Move Networks manages the systems in the platform to provide an end-to-end broadband solution on behalf of video publishers and broadcasters.” <i>Id.</i> A high-level diagram of the Move platform is shown below:

Claim Element	Example Infringement Evidence
	<i>Id.</i> at 6. As MOVE’s contemporaneous documents explain, MOVE’s delivery protocol includes the use of “multiple TCP sessions to optimize the transmission and the carrying capacity during each unit of time. Using streamlets allows a stream to transfer much more efficiently and effectively through the Internet infrastructure.” EX2069 at 6.

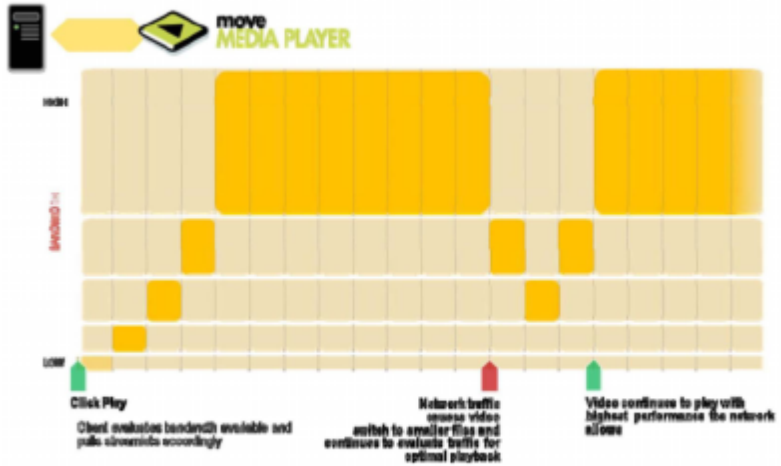
Claim Element	Example Infringement Evidence
	 The diagram illustrates the MOVE system architecture. It begins with 'original content' represented by a film strip icon. An arrow points to an 'atomizer' server icon, which is associated with the 'move SIMULCODE' logo. Another arrow points from the atomizer to a group of server icons labeled 'clustered slaves for streamlet encoding'. A third arrow points from these slaves to a cloud icon labeled 'internet pipeline'. Finally, an arrow points from the pipeline to a 'move MEDIA PLAYER' logo, which is associated with a laptop and smartphone icon. <i>Id.</i> at 7. In this way, MOVE’s solution can be analogized 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.” <i>Id.</i> Thus, MOVE provided a method executable by an end user station to present rate-adaptive streams received via at least one transmission control protocol (TCP) connection with a server over a network.
[8.1] streaming, by a media player operating on the end user station, a video from the server via the at	As explained above, MOVE’s system operated on End User Stations. These End User Stations were equipped to access the Internet via one or more network connections and included at least one processor configured to enable video streaming. <i>See, e.g.</i>, EX2022 at 17-18, 23. The End User Stations also included memory devices having non-transitory machine-readable instructions that caused the End User Stations to establish connection(s) with one or more servers associated with the videos desired for streaming. <i>See</i> EX2021 at 2-7.

Claim Element	Example Infringement Evidence
least one TCP connection over the network, wherein multiple different copies of the video encoded at different bit rates are stored as multiple sets of files on the server,	Further, MOVE’s Media Player automatically placed a streamlet request using a HTTP GET request for each streamlet over one or more network connections to the stream to retrieve a streamlet/file for the selected stream, and the HTTP GET requests identify the selected streamlet, and the requested streamlets are received from the server. <i>See e.g.</i>, EX2022; EX2021. To playback the video normally, the Move Media Player requests sequential files/streamlets from one of the copies from the video server based on the time indexes. <i>See e.g.</i>, EX2022; EX2021. For example, the Move system provides for “[s]imple HTTP protocol transfer of media files from standard Web servers rather than deployment of expensive media servers.” EX2022 at 22. This is further illustrated in the figure below. • Transparent to user • Standard Web Servers or Caches • Multi connection delivery (HTTP/TCP) • Fast Start, Fast Seek, Smooth Adaptive • Client Control: rapid adaptive • Client control: rapid adaptive management • Genuine video <i>Id.</i> In MOVE’s system, the live or on-demand video was encoded using Move Simulcode, which included “[t]he process of atomizing and encoding video for transmission. The first step in this process of preparing video for transmission decompose[d] the captured video content into small streamlets for encoding and transmission The process of Simulcoding refers to the process of producing media that matches multiple viewing profiles (based on device and performance profiles) in a single process.” EX2022 at 10.

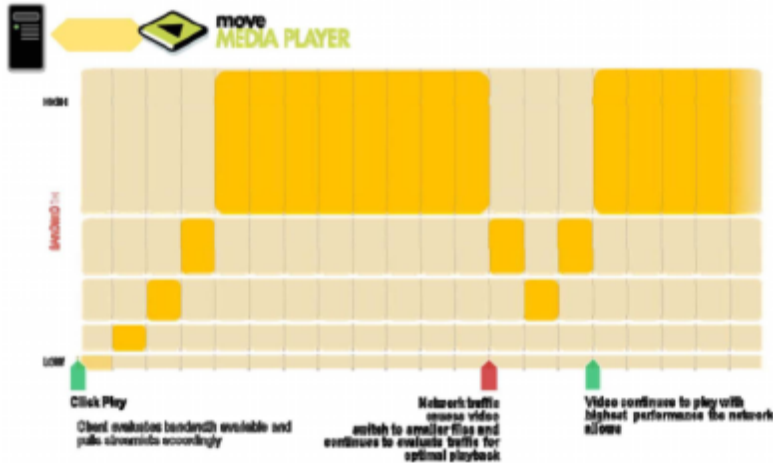
Claim Element	Example Infringement Evidence
	Further, each of the different bitrate/quality streams are stored on and accessed by the server as groups or sets of streamlets/files that correspond to portions of the video, and each group is encoded at the same respective one of the different bitrates. <i>See, e.g.</i>, EX2022; EX2021. For example, “Live content streamlets are encoded ... to provide a self-optimizing, variable bitrate viewing experience.” EX2022 at 10. The figure below provides an overview of this aspect of the Move System. <pre> graph LR SA[Source Acquisition (Standard)] --> QS[Quantum Simulcoding (Enhancement)] QS --> VS[Video Streamlets For Multiple Delivery Profiles] QS --> CU[Content URL • Content ID • Location Data • Play & DRM info] VS --> US[Upload to http Origin Server (Standard)] CU --> US </pre> The diagram illustrates the Move System architecture. It starts with 'Source Acquisition (Standard)' represented by a satellite icon. This feeds into 'Quantum Simulcoding (Enhancement)' represented by a server rack icon. From there, the data is split into two paths: 'Video Streamlets For Multiple Delivery Profiles' (represented by a document icon) and 'Content URL' (represented by a document icon). The 'Content URL' includes 'Content ID', 'Location Data', and 'Play & DRM info'. Both paths lead to 'Upload to http Origin Server (Standard)' represented by a server rack icon. <i>Id.</i> at 11. More specifically as to the different viewing profiles and bitrates, “Move Networks encode[d] to either high definition (16x9) or standard definition (3x4) aspect ratios, using one of the following ‘best practice’ profile sets, based on an aspect ratio (16x9) frames-per-second. Profiles up to 2.4 Mbps are used for video streams intended for distribution over high-bandwidth networks. The low-end profiles include a ‘thumbnail’ profile (Stream Type = T), which is used in multi-player mosaic Web pages, and an ‘emergency profile (Stream Type = E) which provides a minimal experience for poor connectivity or low-bandwidth connections. The remaining profiles are ‘normal’ and are situated at bit rates that offer a smooth end user experience as the Move Media Player up-shifts and down-shifts in response to the network and CPU availability.” <i>Id.</i> at 19. Further, “[w]hen an Internet user request[ed] a video using a standard HTTP ‘GET’ request, streamlets [would be] 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 [time-aligned] order. The delivery protocol uses multiple TCP connections to substantially improve the reliability of the transmission and increase the total capacity during each unit of time.” EX2077 at 1.

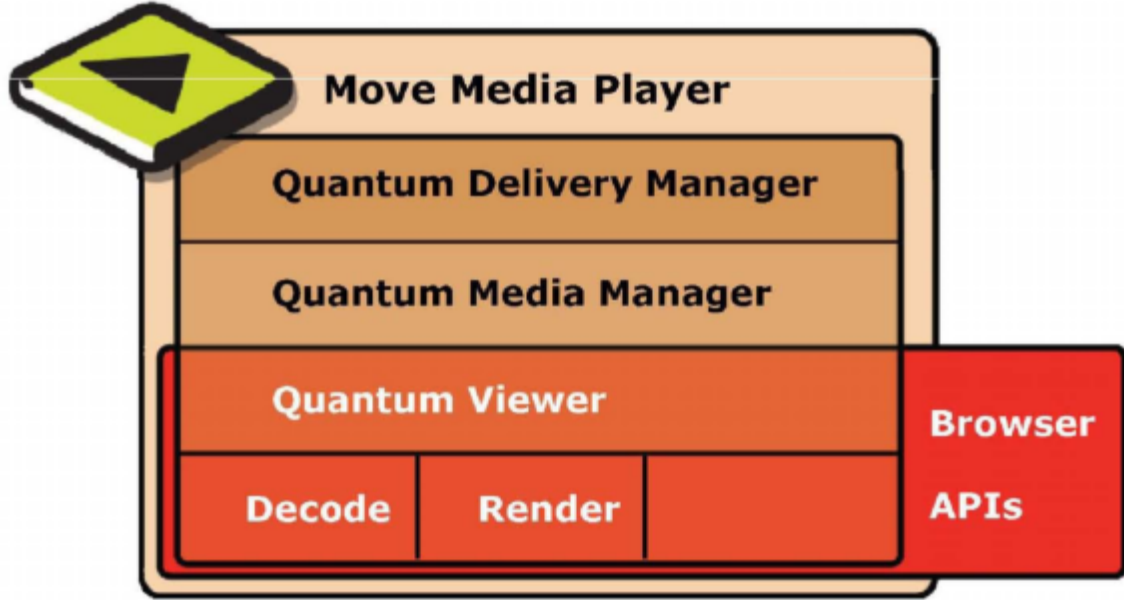
Claim Element	Example Infringement Evidence
	Using streamlets allows a video stream to transfer much more efficiently and effectively through the Internet infrastructure. It also protects against stalls or other glitches.  <i>Id.</i> Thus, Move practiced streaming, by a media player operating on the end user station, a video from the server via the at least one TCP connection over the network, wherein multiple different copies of the video encoded at different bit rates are stored as multiple sets of files on the server.
[8.2] wherein each of the files yields a different portion of the video on playback, wherein the files across the different copies yield the same portions of the	As explained above, the live or on-demand video streamed using MOVE’s system was encoded using Move Simulcode, which included “[t]he process of atomizing and encoding video for transmission. The first step in this process of preparing video for transmission decompose[d] the captured video content into small streamlets for encoding and transmission The process of Simulcoding refers to the process of producing media that matches multiple viewing profiles (based on device and performance profiles) in a single process.” EX2022 at 10. Further, each of the different bitrate/quality streams are stored on and accessed by the server as groups or sets of streamlets/files that correspond to portions of the video, and each group is encoded at the same respective one of the different bitrates. <i>See, e.g.</i>, EX2022; EX2021. For example, “Live content streamlets are encoded ... to provide

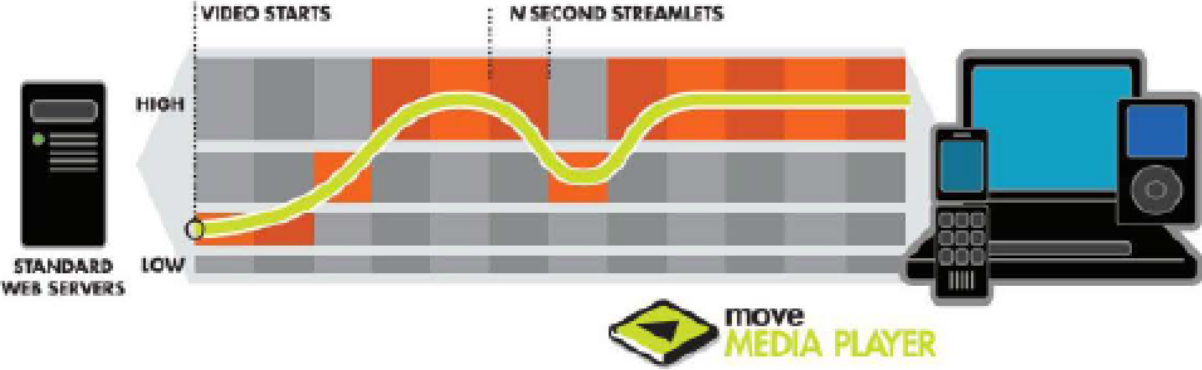
Claim Element	Example Infringement Evidence
video on playback, and wherein each of the files comprises a time index such that the files whose playback is the same portion of the video for each of the different copies have the same time index in relation to the beginning of the video, and wherein the streaming comprises:	a self-optimizing, variable bitrate viewing experience.” EX2022 at 10. The figure below provides an overview of this aspect of the Move System. The diagram illustrates the Move System architecture. It starts with 'Source Acquisition (Standard)' represented by a satellite icon. This feeds into a central processing unit labeled 'Quantum Simulcoding (Enhancement)' which has three vertical bars. From this unit, two paths emerge: one to 'Video Streamlets For Multiple Delivery Profiles' and another to 'Content URL'. The 'Content URL' box lists: '• Content ID', '• Location Data', and '• Play & DRM info'. Both paths lead to a server icon labeled 'Upload to http Origin Server (Standard)'. <i>Id.</i> at 11. More specifically as to the different viewing profiles and bitrates, “Move Networks encode[d] to either high definition (16x9) or standard definition (3x4) aspect ratios, using one of the following ‘best practice’ profile sets, based on an aspect ratio (16x9) frames-per-second. Profiles up to 2.4 Mbps are used for video streams intended for distribution over high-bandwidth networks. The low-end profiles include a ‘thumbnail’ profile (Stream Type = T), which is used in multi-player mosaic Web pages, and an ‘emergency profile (Stream Type = E) which provides a minimal experience for poor connectivity or low-bandwidth connections. The remaining profiles are ‘normal’ and are situated at bit rates that offer a smooth end user experience as the Move Media Player up-shifts and down-shifts in response to the network and CPU availability.” <i>Id.</i> at 19. Similarly, the streamlets across the groups include a time index, are time aligned, and corresponding streamlets across the groups have equal durations. <i>See e.g.</i>, EX2022; EX2021. For example, “Move Networks provides an integrated Content Management System (CMS) for entering the metadata associated with video programming, such as titles, descriptions, topics, program thumbnails, and time offsets (beginning and end points of a video segment). Time offsets can also be automatically detected in some cases based on time indexes and sudden changes in video keyframes.” EX2022 at 25. Further, the figure below provides an exemplary illustration of the duration and alignment of streamlets/files of varying bitrates in the Move system.

Claim Element	Example Infringement Evidence
	Instant Start, No Buffering: Move Adaptive Streaming  <i>Id.</i> at 24. Thus, MOVE practiced encoding a video at a plurality of different bitrates to create a plurality of streams including at least a low quality stream, a medium quality stream, and a high quality stream, wherein each of the low quality stream, the medium quality stream, and the high quality stream comprises a streamlet that encodes the same portion of the video at a different one of the plurality of different bitrates. MOVE's system also practiced associating streamlets with a time index such that the files whose playback is the same portion of the video for each of the different copies have the same time index in relation to the beginning of the video.
[8.3] requesting by the media player a plurality of sequential files of one of the copies	As explained above, in MOVE's system, the Media Player automatically places a streamlet request using a HTTP GET request for each streamlet over one or more network connections to the stream to retrieve a streamlet/file for the selected stream, and the HTTP GET requests identify the selected streamlet, and the requested streamlets are received from the server. <i>See e.g.</i>, EX2022 at 23; EX2021 at 7-8.

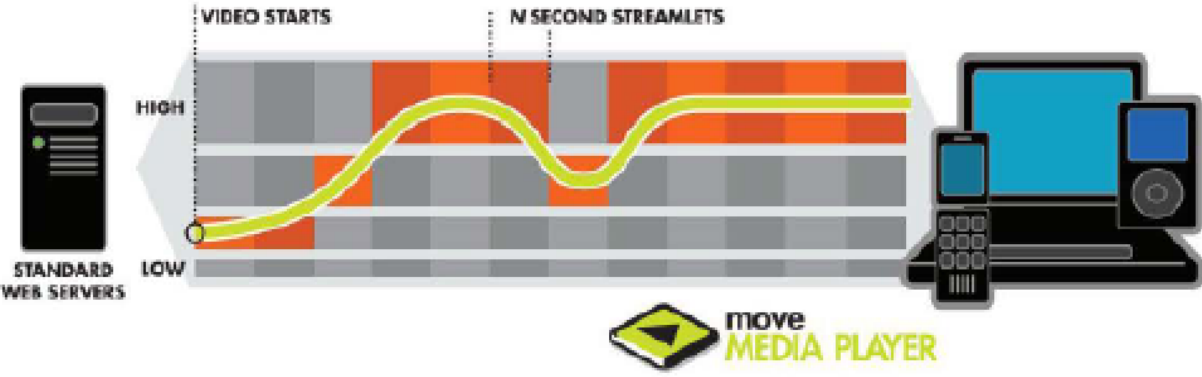
Claim Element	Example Infringement Evidence
from the server based on the time indexes;	The stream selection and automatic streamlet/file requests by the Move Media Player are based upon successive local determinations to shift the playback quality to a higher or lower quality one of the copies and includes repeatedly generating a factor relating to the performance of the network that is indicative of the media player’s current ability to sustain streaming the video using the streamlets/files from different ones of the copies. <i>See e.g.</i>, EX2022 at 23; EX2021 at 8 (“The basic question is not, ‘what is my available bandwidth?’ but, ‘<i>am I receiving the current stream fast enough to present it in real time?</i>’”). The Move Media Player makes such successive determinations to shift to achieve continuous playback of the video using the highest quality determined sustainable such that the player upshifts to a higher quality when the factor is greater than a first threshold and downshifts when the factor is less than a second threshold. <i>See e.g.</i>, EX2022 at 23; EX2021 at 8. In this context, the Move client evaluates bandwidth available and pulls streamlets accordingly, where such “pull” would be made with an HTTP GET request. <i>See EX EX2022 at 23.</i> For example, the Move system “provide[s] a self-optimizing, variable bitrate viewing experience.” <i>Id.</i> at 10. These processes produced several benefits including “continuous playback—no noticeable buffering” and “[q]uality of service optimization based on hardware profiles and speed and quality of the connection to the Internet.” <i>Id.</i> at 11. The Move system was configured to “offer a smooth end user experience as the Move Media Player up-shifts and down-shifts in response to network and client CPU availability.” <i>Id.</i> at 19. More specifically, the “Move Media Player, the Move Networks client software, delivers the video experience to the end consumer. The client decodes and renders the video streamlets to provide a continuous, optimal stream of video. A major function of the client is managing the viewing experience. The Quantum Delivery Manager ensures the player is constantly receiving the streamlets required to deliver a continuous stream of video. The Quantum Media Manager deduces the current environment to tell the Delivery Manager which streamlets to pull into the player. For instance, if the Quantum Media Manager senses that the current environment has poor connectivity, the Quantum Delivery manager will be directed to only request streamlets from low-bandwidth profiles. If the connection speed or quality of the connection suddenly increases, the Quantum Delivery Manager starts downloading higher quality streamlets. The Quantum Media Manager continues to test the environment to provide the best viewing experience currently possible. This ability to ‘upshift’ and ‘downshift’ to multiple viewing profile bit rates ... is called Adaptive Streaming.” <i>Id.</i> at 23. The diagram below illustrates how Move Adaptive Streaming starts instantly, doesn’t stall, and adapts the streaming.

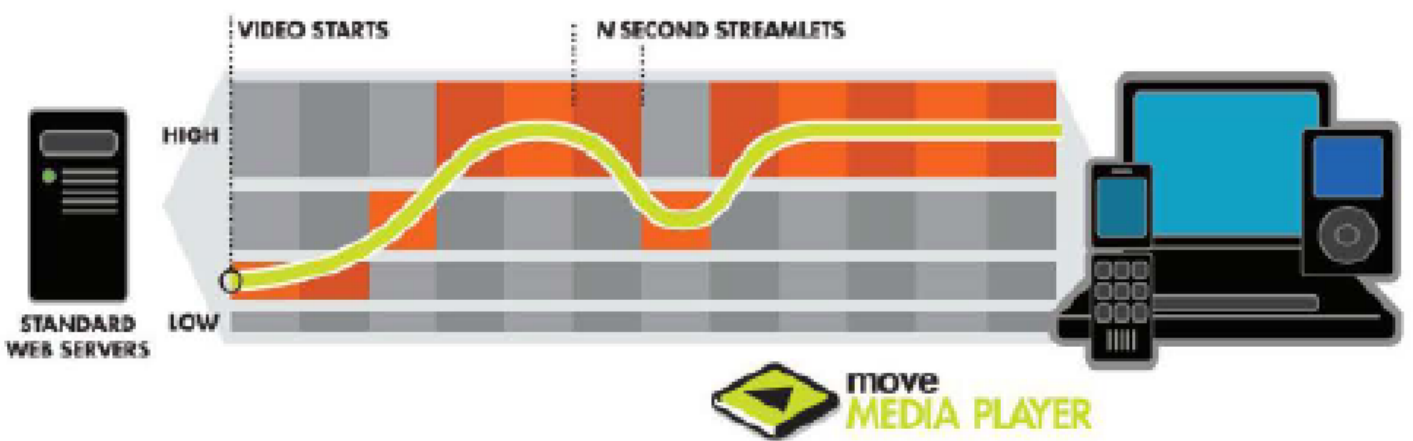
Claim Element	Example Infringement Evidence
	Instant Start, No Buffering: Move Adaptive Streaming  The diagram illustrates the Move Adaptive Streaming process. It features a timeline with four horizontal bars representing different video quality levels: 'video' (top, yellow), 'bandwidth 2x' (yellow), 'bandwidth 1x' (yellow), and 'low' (bottom, yellow). A green arrow labeled 'Click Play' points to the start of the timeline. A yellow bar labeled 'move MEDIA PLAYER' is positioned above the timeline. The timeline shows a sequence of events: 1. 'Click Play' (green arrow). 2. 'Client analyzes bandwidth available and picks streamlets accordingly' (yellow bar). 3. 'Network buffer grows video switches to smaller files and continues to evaluate traffic for optimal playback' (red arrow). 4. 'Video continues to play with highest performance the network allows' (green arrow). The diagram shows the player switching between different quality levels based on available bandwidth and network performance. <i>Id.</i> at 24. Importantly, “the intelligence that controls the quality of service resid[es] in the client.” <i>Id.</i> at 22. The client of the Move system “is always on the watch to use an appropriate stream encoding rate selection so that the available bandwidth is not exceeded by downloading of the component streamlets.” EX2021 at 8. Because “<i>the client</i> [of the Move system] <i>can select and retrieve directly from an array of streamlets of different encode rates</i>, there is no need to inform any media server and ask it to change the data it is sending.” <i>Id.</i> Thus, “[i]f the client observes that over some modest interval of time (say 20 seconds or so) the player is, on average, consuming streamlets more quickly than they arrive, then a stream of lower quality must be selected.” <i>Id.</i> In view of this, the Move Media Player makes successive determinations as to whether or not to shift to achieve continuous playback of the video using the highest quality determined sustainable. See, e.g., EX EX2022 at 23. As part of the streaming, the Move Media Player also requested and received a virtual timeline from the server based on the selected one of the streams. See e.g., EX2022 at 26 (referencing “QVT”). The figure below further illustrates the components of the Move Media Player that render the video.

Claim Element	Example Infringement Evidence
	 The diagram illustrates a media player architecture. A yellow diamond icon with a black triangle points to a stack of components. The stack includes 'Move Media Player' (orange), 'Quantum Delivery Manager' (orange), 'Quantum Media Manager' (orange), 'Quantum Viewer' (orange), and a red box containing 'Decode', 'Render', and 'APIs'. To the right of the red box is a red box labeled 'Browser'. EX2022 at 25. Further, “[w]hen an Internet user request[ed] a video using a standard HTTP ‘GET’ request, streamlets [would be] 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 [time-aligned] order. The delivery protocol uses multiple TCP connections to substantially improve the reliability of the transmission and increase the total capacity during each unit of time.” EX2077 at 1.

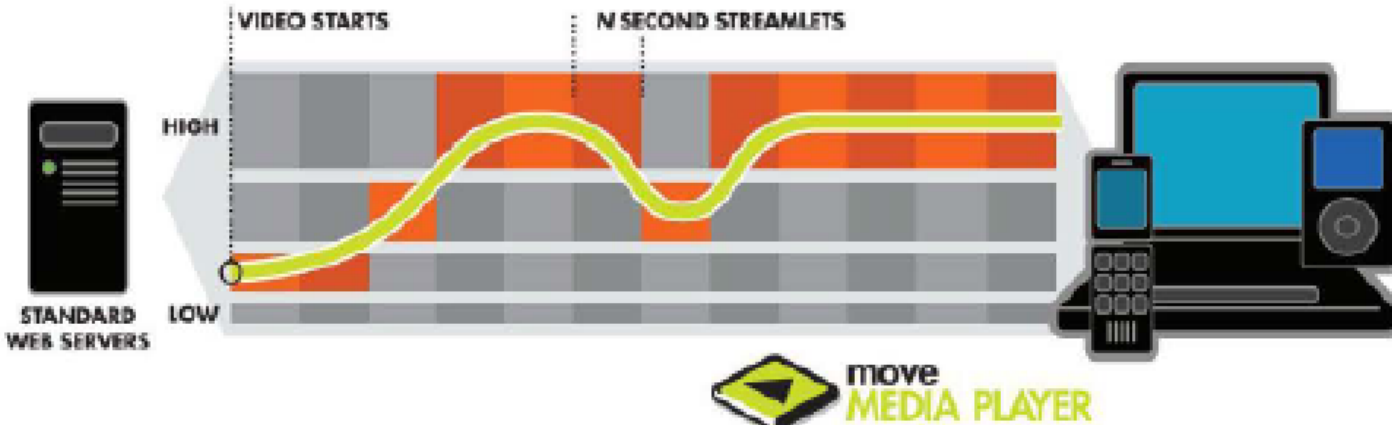
Claim Element	Example Infringement Evidence
	Using streamlets allows a video stream to transfer much more efficiently and effectively through the Internet infrastructure. It also protects against stalls or other glitches.  <i>Id.</i> Thus, MOVE practiced requesting by the media player a plurality of sequential files of one of the copies from the server based on the time indexes.
[8.4] automatically requesting by the media player from the server subsequent portions of the video by requesting for each such portion one of the files from	As shown above, the MOVE system would request and receive the requested streamlets from server(s) via the one or more network connections. For example, MOVE’s system provided for “[s]imple HTTP protocol transfer of media files from standard Web servers rather than deployment of expensive media servers.” EX2022 at 22. This is further illustrated in the figure below.

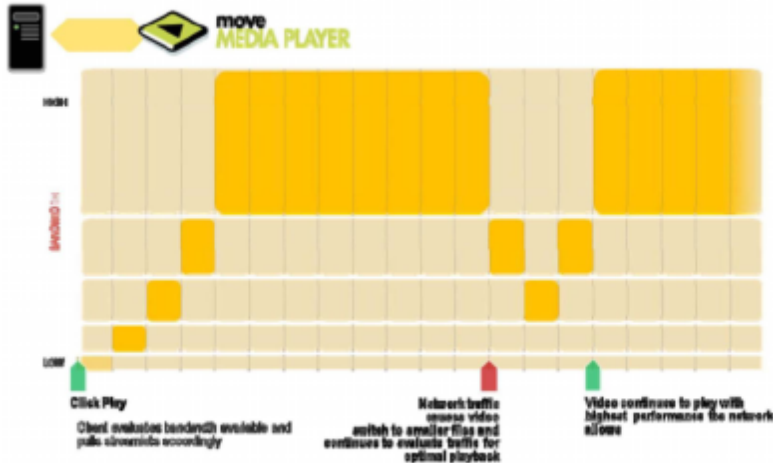
Claim Element	Example Infringement Evidence
one of the copies dependent upon successive determinations by the media player to shift the playback quality to a higher or lower quality one of the different copies,	<i>Id.</i> Further, “[w]hen an Internet user request[ed] a video using a standard HTTP ‘GET’ request, streamlets [would be] 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 connections to substantially improve the reliability of the transmission and increase the total capacity during each unit of time.” EX2077 at 1. The Move Media Player makes such successive determinations to shift to achieve continuous playback of the video using the highest quality determined sustainable such that the player upshifts to a higher quality when the factor is greater than a first threshold and downshifts when the factor is less than a second threshold. <i>See e.g.</i>, EX2022 at 23; EX2021 at 8. In this context, the Move client evaluates bandwidth available and pulls streamlets accordingly, where such “pull” would be made with an HTTP GET request. <i>See</i> EX2022 at 23. “Because Move Adaptive Stream puts the intelligence in the client, the ‘driver’ can make instantaneous decisions to speed up, slow down, or change lanes to minimize delay or maximize viewing experience.” EX2077 at 2. For example, the Move system “provide[s] a self-optimizing, variable bitrate viewing experience.” <i>Id.</i> at 10. These processes produced several benefits including “continuous playback—no noticeable buffering” and “[q]uality of service optimization based on hardware profiles and speed and quality of the connection to the Internet.” <i>Id.</i> at 11. The Move system was configured to “offer a smooth end user experience as the Move Media Player up-shifts and down-shifts in response to network and client CPU availability.” <i>Id.</i> at 19. More specifically, the “Move Media Player, the Move Networks client software, delivers the video experience to the end consumer. The client decodes and renders the video streamlets to provide a continuous, optimal stream of video. A major function of the client is managing the viewing experience.

Claim Element	Example Infringement Evidence
	As explained elsewhere, the Quantum Delivery Manager of the Move system ensures the player is constantly receiving the streamlets required to deliver a continuous stream of video. The Quantum Media Manager deduces the current environment to tell the Delivery Manager which streamlets to pull into the player. For instance, if the Quantum Media Manager senses that the current environment has poor connectivity, the Quantum Delivery manager will be directed to only request streamlets from low-bandwidth profiles. If the connection speed or quality of the connection suddenly increases, the Quantum Delivery Manager starts downloading higher quality streamlets. The Quantum Media Manager continues to test the environment to provide the best viewing experience currently possible. This ability to ‘upshift’ and ‘downshift’ to multiple viewing profile bit rates ... is called Adaptive Streaming.” <i>Id.</i> at 23. The diagram below illustrates how Move Adaptive Streaming starts instantly, doesn’t stall, and adapts the streaming. 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.  <i>Id.</i> Thus, Move practiced automatically requesting by the media player from the server subsequent portions of the video by requesting for each such portion one of the files from one of the copies dependent upon successive

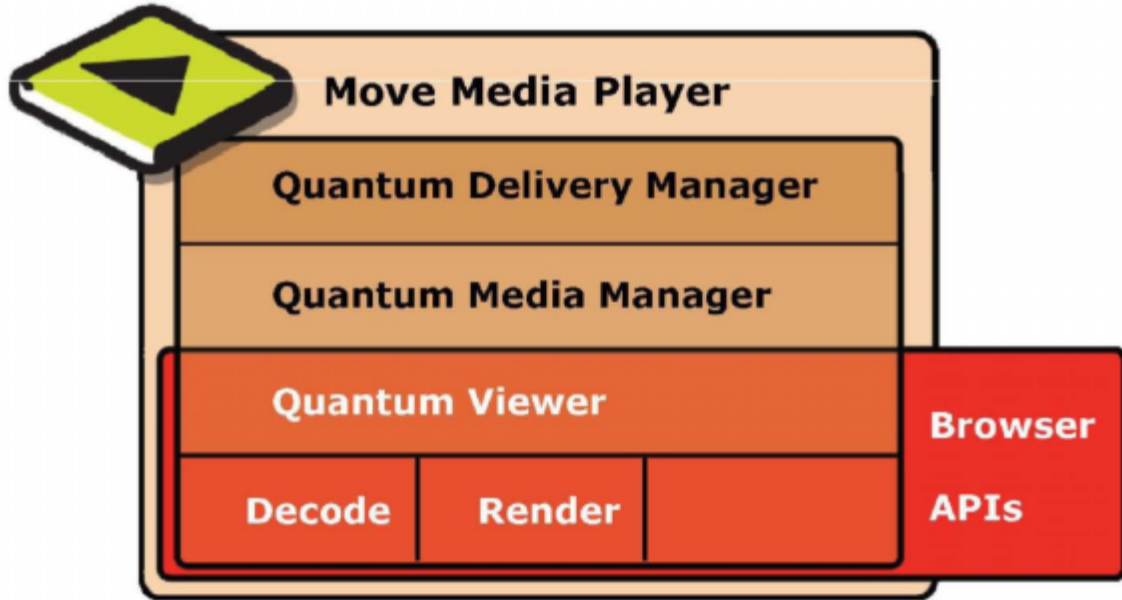
Claim Element	Example Infringement Evidence
	determinations by the media player to shift the playback quality to a higher or lower quality one of the different copies.
[8.5] the automatically requesting including repeatedly generating a factor indicative of the current ability to sustain the streaming of the video using the files from different ones of the copies, wherein the factor relates to the performance of the network; and	The stream selection and automatic streamlet/file requests by the Move Media Player are based upon successive local determinations to shift the playback quality to a higher or lower quality one of the copies and includes repeatedly generating a factor relating to the performance of the network that is indicative of the media player's current ability to sustain streaming the video using the streamlets/files from different ones of the copies. <i>See e.g.</i>, EX2022 at 23; EX2021 at 8 (“The basic question is not, ‘what is my available bandwidth?’ but, ‘<i>am I receiving the current stream fast enough to present it in real time?</i>’”).  EX2077 at 2. The stream selection and automatic streamlet/file requests by the Move Media Player are based upon successive local determinations to shift the playback quality to a higher or lower quality one of the copies and includes repeatedly generating a factor relating to the performance of the network that is indicative of the media player's current ability to sustain streaming the video using the streamlets/files from different ones of the copies. <i>See e.g.</i>, EX2022 at 23; EX2021 at 8 (“The basic question is not, ‘what is my available bandwidth?’ but, ‘<i>am I receiving the current stream fast enough to present it in real time?</i>’”).

Claim Element	Example Infringement Evidence
	The Move Media Player makes such successive determinations to shift to achieve continuous playback of the video using the highest quality determined sustainable such that the player upshifts to a higher quality when the factor is greater than a first threshold and downshifts when the factor is less than a second threshold. <i>See e.g.</i>, EX2022 at 23; EX2021 at 8. In this context, the Move client evaluates bandwidth available and pulls streamlets accordingly, where such “pull” would be made with an HTTP GET request. <i>See</i> EX2022 at 23. MOVE’s contemporaneous documents also analogize MOVE’s system to a “multi-lane freeway where traffic stalls in one lane can be minimized by changing lanes, significantly expanding the aggregate traffic capacity.” <i>Id.</i>
[8.6] making the successive determinations to shift the playback quality based on the factor to achieve continuous playback of the video using the files of the highest quality one of the copies determined sustainable at that time, wherein the making the successive determinations to shift comprises upshifting to a higher quality one	As explained above, in MOVE’s system, the Media Player automatically places a streamlet request using a HTTP GET request for each streamlet over one or more network connections to the stream to retrieve a streamlet/file for the selected stream, and the HTTP GET requests identify the selected streamlet, and the requested streamlets are received from the server. <i>See e.g.</i>, EX2022 at 23; EX2021 at 7-8. The stream selection and automatic streamlet/file requests by the Move Media Player are based upon successive local determinations to shift the playback quality to a higher or lower quality one of the copies and includes repeatedly generating a factor relating to the performance of the network that is indicative of the media player’s current ability to sustain streaming the video using the streamlets/files from different ones of the copies. <i>See e.g.</i>, EX2022 at 23; EX2021 at 8 (“The basic question is not, ‘what is my available bandwidth?’ but, ‘<i>am I receiving the current stream fast enough to present it in real time?</i>’”). The Move Media Player makes such successive determinations to shift to achieve continuous playback of the video using the highest quality determined sustainable such that the player upshifts to a higher quality when the factor is greater than a first threshold and downshifts when the factor is less than a second threshold. <i>See e.g.</i>, EX2022 at 23; EX2021 at 8. In this context, the Move client evaluates bandwidth available and pulls streamlets accordingly, where such “pull” would be made with an HTTP GET request. <i>See</i> EX2022 at 23. For example, the Move system “provide[s] a self-optimizing, variable bitrate viewing experience.” <i>Id.</i> at 10. These processes produced several benefits including “continuous playback—no noticeable buffering” and “[q]uality of service optimization based on hardware profiles and speed and quality of the connection to the

Claim Element	Example Infringement Evidence
of the different copies when the at least one factor is greater than a first threshold and downshifting to a lower quality one of the different copies when the at least one factor is less than a second threshold; and	Internet.” <i>Id.</i> at 11. The Move system was configured to “offer a smooth end user experience as the Move Media Player up-shifts and down-shifts in response to network and client CPU availability.” <i>Id.</i> at 19.  EX2077 at 2. More specifically, the “Move Media Player, the Move Networks client software, delivers the video experience to the end consumer. The client decodes and renders the video streamlets to provide a continuous, optimal stream of video. A major function of the client is managing the viewing experience. The Quantum Delivery Manager ensures the player is constantly receiving the streamlets required to deliver a continuous stream of video. The Quantum Media Manager deduces the current environment to tell the Delivery Manager which streamlets to pull into the player. For instance, if the Quantum Media Manager senses that the current environment has poor connectivity, the Quantum Delivery manager will be directed to only request streamlets from low-bandwidth profiles. If the connection speed or quality of the connection suddenly increases, the Quantum Delivery Manager starts downloading higher quality streamlets. The Quantum Media Manager continues to test the environment to provide the best viewing experience currently possible. This ability to ‘upshift’ and ‘downshift’ to multiple viewing profile bit rates ... is called Adaptive Streaming.” EX2022 at 23. The diagram below illustrates how Move Adaptive Streaming starts instantly, doesn’t stall, and adapts the streaming.

Claim Element	Example Infringement Evidence
	Instant Start, No Buffering: Move Adaptive Streaming  The diagram illustrates the Move Adaptive Streaming process. It features a vertical axis on the left with labels 'high', 'bandwidth 1/2', and 'low'. The horizontal axis represents time. A yellow arrow labeled 'move MEDIA PLAYER' points to the right. The diagram shows a sequence of yellow blocks representing video segments. The first block is at the 'high' bandwidth level. A green arrow labeled 'Click Play' points to the start of the first block. Below the first block, text reads: 'Client evaluates bandwidth available and picks streamlets accordingly'. The second block is at the 'bandwidth 1/2' level. Below it, text reads: 'Network buffer causes video switch to smaller files and continues to evaluate buffer for optimal playback'. The third block is at the 'low' bandwidth level. Below it, text reads: 'Video continues to play with highest performance the network allows'. The diagram shows the player adapting to changing bandwidth conditions by switching to lower quality streamlets when bandwidth is low and switching back to higher quality when bandwidth improves. <i>Id.</i> at 24. Importantly, “the intelligence that controls the quality of service resid[es] in the client.” <i>Id.</i> at 22. The client of the Move system “is always on the watch to use an appropriate stream encoding rate selection so that the available bandwidth is not exceeded by downloading of the component streamlets.” EX2021 at 8. Because “<i>the client</i> [of the Move system] <i>can select and retrieve directly from an array of streamlets of different encode rates</i>, there is no need to inform any media server and ask it to change the data it is sending.” <i>Id.</i> Thus, “[i]f the client observes that over some modest interval of time (say 20 seconds or so) the player is, on average, consuming streamlets more quickly than they arrive, then a stream of lower quality must be selected.” <i>Id.</i> In view of this, the Move Media Player makes successive determinations as to whether or not to shift to achieve continuous playback of the video using the highest quality determined sustainable. See, e.g., EX EX2022 at 23. As part of the streaming, the Move Media Player also requested and received a virtual timeline from the server based on the selected one of the streams. See e.g., EX2022 at 26 (referencing “QVT”). The figure below further illustrates the components of the Move Media Player that render the video.

Claim Element	Example Infringement Evidence
	The diagram illustrates the architecture of the Move Media Player. It consists of a main container labeled 'Move Media Player' which contains three stacked components: 'Quantum Delivery Manager', 'Quantum Media Manager', and 'Quantum Viewer'. The 'Quantum Viewer' component is further divided into three sub-sections: 'Decode', 'Render', and an unlabeled section. To the right of the 'Quantum Viewer' component is a red box labeled 'Browser APIs'. A yellow diamond icon with a black arrow is positioned above the 'Quantum Delivery Manager' component. EX2022 at 25. Thus, Move practiced making the successive determinations to shift the playback quality based on the factor to achieve continuous playback of the video using the files of the highest quality one of the copies determined sustainable at that time, wherein the making the successive determinations to shift comprises upshifting to a higher quality one of the different copies when the at least one factor is greater than a first threshold and downshifting to a lower quality one of the different copies when the at least one factor is less than a second threshold.
[8.7] presenting the video by playing back the requested media files with the media player on the end user	As has been described above, the Move Media Player would render and present the requested and received streamlets on the end user station/apparatus in ascending order of playback time to playback the video normally. <i>See e.g.</i>, EX2022 at 23; EX2021 at 7. For example, “Move Media Player, the Move Networks client software, delivers the video experience to the end consumer. The client decodes and renders the video streamlets to provide a continuous, optimal stream of video.” EX2022 at 23. The figure below further illustrates the components of the Move Media Player that render the video.

Claim Element	Example Infringement Evidence
station in order of ascending playback time.	 Move Media Player Quantum Delivery Manager Quantum Media Manager Quantum Viewer Decode Render Browser APIs EX2022 at 25. Thus, MOVE’s system provided received first streamlets for playback of the video on the end user station in order of ascending playback time.