FIGS. 15 and 16 show alternative threading or prediction chain architectures 1500 and 1600, which may be used in video communication or conferencing applications, in accordance with the present invention. Implementations of threading structures or prediction chains 1500 and 1600 do not require any substantial changes to the codec designs described above with reference to FIGS. 2-14.

In architecture 1500, an exemplary combination of layers (S0, L0, and L1) is transmitted over high reliability channel 170. It is noted that, as shown, L1 is part of the L0 prediction chain 430, but not for S1. Architecture 1600 shows further examples of threading configurations, which also can achieve non-dyadic frame rate resolutions.

System 100 and terminal 140 codec designs described above are flexible and can be readily extended to incorporate alternative SVC schemes. For example, coding of the S layer may be accomplished according to the forthcoming ITU-T H.264 SVC FGS specification. When FGS is used, the S layer coding may be able to utilize arbitrary portions of a 'S' packet due to the embedded property of the produced bitstream. It may be possible to use portions of the FGS component to create the reference picture for the higher layers. Loss of the FGS component information in transmission over the communications network may introduce drift in the decoder. However, the threading architecture employed in the present invention advantageously minimizes the effects of such loss. Error propagation may be limited to a small number of frames in a manner that is not noticeable to viewers. The amount of FGS to include for reference picture creation may change dynamically.

A proposed feature of the H.264 SVC FGS specification is a leaky prediction technique in the FGS layer. See Y. Bao et al., _____, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 15th meeting, Busan, Korea, 18-22 April, 2005. The leaky prediction technique consists of using a normalized weighted of the previous FGS enhancement layer current base layer ire. The weighted average is controlled by a weight parameter alpha; if alpha is 1 then only the current base layer picture is used, whereas if it is 0 then only the previous FGS enhancement layer picture is used. The case where alpha is 0 is identical to the use of motion estimation (ME 330, FIG. 5) for the SNR enhancement layer of the present invention, in the limiting case of using only zero motion vectors. The leaky prediction technique can be used in conjunction with regular ME as

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 211 of 830

described in this invention. Further, it is possible to periodically switch the alpha value to 0, in order to break the prediction loop in the FGS layer and eliminate error drift.

FIG. 17 shows the switch structure of an exemplary MCU/SVCS 110 that is used in videoconferencing system 100 (FIG. 1). MCU/SVCS 110 determines which packet from each of the possible sources (e.g., endpoints 120-140) is transmitted to which destination and over which channel (high reliability vs. low reliability) and switches signals accordingly. The designs and the switching functions MCU/SVCS 110 are described in co-filed U.S. Patent Application No. ____ [SVCS]___, incorporated by reference herein. For brevity, only limited details of the switch structure and switching functions of MCU/SVCS 110 are described further herein.

FIG. 18 shows the operation of an exemplary embodiment of MCU/SVCS switch 110. MCU/SVCS switch 110 maintains two data structures in its memory – an SVCS Switch Layer Configuration Matrix 110A and an SVCS Network Configuration Matrix 110, examples of which are shown in FIGS. 19 and 20, respectively. SVCS Switch Layer Configuration Matrix 110A (FIG. 19) provides information on how a particular data packet should be handled for each layer and for each pair of source and destination endpoints 120-140. For example, a matrix 110A element value of zero indicates that the packet should not be transmitted; a negative matrix element value indicates that the entire packet should be transmitted; and a positive matrix element value indicates that only the specified percentage of the packet's data should be transmitted. Transmission of a specified percentage of the packet's data may be relevant only when an FGS-type of technique is used to scalably code signals.

FIG. 18 also shows an algorithm 1800 in MCU/SVCS 110 for directing data packets utilizing Switch Layer Configuration Matrix 110A information. At step [CU/SVCS 110 may examine received p e.g., NAL headers, ming use of H.264). At step 1804, MCU/SVCS 110 evaluates the value of relevant matrix 110A elements for source, destination, and layer combinations to establish processing instructions and designated destinations for the received packets. In applications using FGS coding, positive matrix element values indicate that the packet's payload must be reduced in size. Accordingly, at step 1806, the relevant

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 212 of 830

length entry of the packet is changed and no data is copied. At step 1808, the relevant layers or combination of layers are switched to their designated destinations.

With reference to FIGS. 18 and 20, SVCS Network Configuration Matrix 110B tracks the port numbers for each participating endpoint. MCU/SVCS 110 may use Matrix 110B information to transmit and receive data for each of the layers.

The operation of MCU/SVCS 110 based on processing Matrices 110A and 110B allows signal switching to occur with zero or minimal internal algorithmic delay, in contrast to traditional MCU operations. Traditional MCUs have to compose incoming video to a new frame for transmission to the various participants. This composition requires full decoding of the incoming streams and recoding of the output stream. The decoding/recoding processing delay in such MCUs is significant, as is the computational power required. By using scalable bitstream architecture, and providing multiple instances of decoders 230A in each endpoint terminal 140 receiver, MCU/SVCS 110 is required only to filter incoming packets to select the appropriate layer(s) for each recipient destination. The fact that no or minimal DSP processing is required can advantageously allow MCU/SVCS 110 to be implemented with very little cost, offer excellent scalability (in terms of numbers of sessions that can be hosted simultaneously on a given device), and with end-to-end delays which may be only slightly larger than the delays in a direct endpoint-to-endpoint connection.

Terminal 140 and MCU/SVCS 100 may be deployed in different network scenarios using different bitrates and stream combinations. TABLE II shows the possible bitrates and stream combinations in various exemplary network scenarios. It is noted that base bandwidth/total bandwidth >=50% is the limit of DiffServ layering effectiveness, and further a temporal resolution of less than 15 fps is not useful.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 213 of 830

TABLE II

Bitstream Components for Various Network Scenarios

	HRC	LRC	Total line speed	HRC vs. LRC bandwidth
Client transmits	L0 +L1 = 100	S0+S1+L2 +S 2 = 150+100+150=400	500	1:4
SVCS reflects for CIF recipient	Same	Same	500	1:4
SVCS for lower speed client 1	L0+L1 =100	S0 + ½ × (S1+S2) +L2 = 150+100=250	350	1:2.5
SVCS for lower speed client 2 QCIF view at 30 fps	L0+L1 =100	L2 = 100	200	1:1
SVCS for lower speed client 3 CIF view at 15 fps	L0 = 50	L1 + S0+S1 = 50+150	200	1:1
SVCS for lower speed client 4 QCIF at 15 fps	L0=50	L1=50	100	1:1
SVCS for very low speed client CIF 7.5 fps	L0=50	S0=50	100	1:1

Terminal 140 and like configurations of the present invention allow scalable coding techniques to be exploited in the context of point-to-point and multi-point videoconferencing systems deployed over channels that can provide different QoS guarantees. The selection of the scalable codecs described herein, the selection of a threading model, the choice of which layers to transmit over the high reliability or low reliability channel, and the selection of appropriate bitrates (or quantizer step r the various layers are relevant design ch may vary with cular implementations of the present invention. Typically, such design choices may be made once and the parameters remain constant during the deployment of a videoconferencing system, or at least during a particular videoconferencing session. However, it will be understood that SVC configurations of the present invention offer the flexibility to dynamically adjust these parameters within a single videoconferencing session. Dynamic adjustment of the parameters may be desirable,

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 214 of 830

taking into account a participant's/endpoint's requirements (e.g., which other participants should be received, at what resolutions, etc.) and network conditions (e.g., loss rates, jitter, bandwidth availability for each participant, bandwidth partitioning between high and low reliability channels, etc.). Under suitable dynamic adjustment schemes, individual participants/endpoints may interactively be able to switch between different threading patterns (e.g., between the threading patterns shown in FIGS. 4, 8, and 9), elect to change how layers are assigned to the high and low reliability channels, elect to eliminate one or more layers, or change the bitrate of individual layers. Similarly, MCU/SVCS 110 may be configured to change how layers are assigned to the high and low reliability channels linking various participants, eliminate one or more layers, scale the FGS/SNR enhancement layer or some participants.

In an exemplary scenario, a videoconference may have three participants, A, B, and C. Participants A and B may have access to a high-speed 500 Kbps channel that can guarantee a continuous rate of 200 Kbps. Participant C may have access to a 200 Kbps channel that can guarantee 100 Kbps. Participant A may use a coding scheme that has the following layers: a base layer ("Base"), a temporal scalability layer ("Temporal") that provides 7.5 fps, 15 fps, 30 fps video at CIF resolutions, and an SNR enhancement layer ("FPS") that allows increase of the spatial resolution at either of the three temporal frame rates. The Base and Temporal components each require 100 Kbps, and FGS requires 300 Kbps for a total of 500 Kbps bandwidth. Participant A can transmit all three Base, Temporal, and FPS components to MCU 110. Similarly, participant B can receive all three components. However, since only 200 Kbps are guaranteed to participant B in the scenario, FGS is transmitted through the non-guaranteed 300 Kbps channel segment. Participant C can receive only the Base and Temporal components with the Base component guaranteed at 100 Kbps. If the available bandwidth (either guaranteed or total) changes, then nt A's encoder (e.g., Terminal 140) can i mically change the t bitrate for any of the components. For example, if the guaranteed bandwidth is more than 200 Kbps, more bits may be allocated to the Base and Temporal components. Such changes can be implemented dynamically in real-time response since encoding occurs in real-time (i.e., the video is not pre-coded).

If both participants B and C are linked by channels with restricted capacity, e.g., 100 Kbps, then participant A may elect to only transmit the Base

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 215 of 830

component. Similarly, if participants B and C select to view received video only at QCIF resolution, participant A can respond by not transmitting the FGS component since the additional quality enhancement offered by the FGS component will be lost by downsampling of the received CIF video to QCIF resolution.

It will be noted that in some scenarios, it may be appropriate to transmit a single-layer video stream (base layer or total video) and to completely avoid the use of scalability layers.

In transmitting scalable video layers over HRCs and LRCs, whenever information on the LRCs is lost, only the information transmitted on the HRC may be used for video reconstruction and display. In practice, some portions of the displayed video picture will include data produced by decoding the base layer and designated enhancement layers, but other portions will include data produced by decoding only the base layer. If the quality levels associated with the different base layer and enhancement layer combinations are significantly different, then the quality differences between the displayed video picture that include or do not include lost LRC data may become noticeable. The visual effect may be more pronounced in the temporal dimension, where repeated changes of the displayed picture from base layer to 'base plus enhancement layer' may be perceived as flickering. To mitigate this effect, it may be desirable to ensure that the quality difference (e.g., in terms of PSNR) between the base layer picture and 'base plus enhancement layer' picture is kept low, especially on static parts of the picture where flickering is visually more obvious. The quality difference between the base layer picture and 'base plus enhancement layer' picture may be deliberately kept low by using suitable rate control techniques to increase the quality of the base layer itself. One such rate control technique may be to encode all or some of the L0 pictures with a lower QP value (i.e., a finer quantization value). For example, every L0 picture may be encoded with a QP lowered by a factor of 3. Such finer quantization may increase the quality of the base ıs minimizing any flickering effect or equ ırtifacts caused by ›ss of enhancement layer information. The lower QP value may also be applied every other L0 picture, or every four L0 pictures, with similar effectiveness in mitigating flickering and like artifacts. The specific use of a combination of SNR and spatial scalability (e.g., using HCIF coding to represent the base layer carrying QCIF quality) allows proper rate control applied to the base layer to bring static objects

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 216 of 830

close to HCIF resolution, and thus reduce flickering artifacts caused when an enhancement layer is lost.

While there have been described what are believed to be the preferred embodiments of the present invention, those skilled in the art will recognize that other and further changes and modifications may be made thereto without departing from the spirit of the invention, and it is intended to claim all such changes and modifications as fall within the true scope of the invention.

It also will be understood that in accordance with the present invention, the scalable codecs described herein may be implemented using any suitable combination of hardware and software. The software (i.e., instructions) for implementing and operating the aforementioned scalable codecs can be provided on computer-readable media, which can include without limitation, firmware, memory, storage devices, microcontrollers, microprocessors, integrated circuits, ASICS, on-line downloadable media, and other available media.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 217 of 830

WHAT IS CLAIMED IS:

1. A system for video communication between a plurality of endpoints over an electronic communications network and one or more servers, which network provides different quality of service and bandwidth channels linking the plurality of endpoints and servers, the channels including designated high reliability channels (HRCs) and low reliability channels (LRCs), the system comprising:,

transmitting and receiving terminals disposed at the endpoints,

wherein at least one transmitting terminal is configured to prepare at least one scalably coded video signal for transmission to other terminals in a base layer and enhancement layers format, and to transmit at least the base layer over a designated HRC, and

wherein at least one receiving terminal is configured to decode the scalably coded video signal layers that are received over network channels including a designated HRC, and to reconstruct a video for local use by combining the decoded video signal layers, and

wherein the servers are configured to mediate the transfer of the scalably coded video signal layers transmitted by the transmitting terminal to a receiving terminal over electronic communications network channels leading to the latter terminal.

2. The system of claim 1wherein at least one terminal is configured to access at least one of: a live video signal for encoding and transmission; a stored video signal for encoding and transmission; a synthesized video signal for encoding and transmission and a pre-coded video signal for transmission.

3. An endpoint terminal for video communication with other endpoints or more servers disposed in an electronics network, which ork provides different quality of service and bandwidth channels linking the plurality of endpoints, the channels including designated HRCs, the endpoint terminal comprising:

at least one scalable video encoder, which is configured to scalably encode at least one video signal in a base layer and enhancement layers format; and

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 218 of 830

a packet multiplexer, which is configured to multiplex layers of the video signal encoded in the base layer and enhancement layers format for transmission over the electronics communication network,

wherein the endpoint terminal is configured to designate at least the base layer from the base layer and enhancement layers of the video signal for transmission over the designated HRCs via a network interface controller.

4. The endpoint terminal of claim 3, further comprising an audio signal encoder whose output is connected to the packet multiplexer.

5. The terminal of claim 3, wherein the scalable video encoder is a motion-compensated , block-based codec comprised of:

a frame memory where one or more decoded frames are stored for future reference; and

a reference controller configured to select the picture type (I, P, or B) as well as the pictures in the frame memory that will be used as prediction references for the picture currently encoded,

wherein the codec is further configured to perform picture prediction using threads as a means of implementing temporal scalability layers.

6. The terminal of claim 5, wherein the scalable video encoder is configured to create one continuous prediction chain path for the base layer.

7. The terminal of claim 5 where the threads are picture threads comprising:

a base layer thread further comprised of pictures that are a number of pictures apart, and in which temporal prediction is performed using one or more previous pictures of the same thread; and

a temporal enhancement layer thread ·d of the remaining pictures, and in which prediction is performed from one or more preceding base layer pictures and/or one or more preceding temporal enhancement layer pictures.

8. The terminal of claim 5 where the picture threads are comprised of:

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 219 of 830

a base layer thread further comprised of pictures that are a fixednumber of pictures apart, and in which temporal prediction is performed using the immediately previous frame of the same thread;

a first temporal enhancement layer thread that is comprised of frames that are half way between those of the base layer thread, and in which prediction is performed from either the immediately preceding base layer picture or the immediately preceding first temporal enhancement layer pictures; and

a second temporal enhancement layer thread that is comprised of the remaining pictures, and in which prediction is performed from either the immediately preceding second temporal enhancement layer pictures, the immediately preceding first temporal enhancement layer picture, or the immediately preceding base layer thread picture.

9. The terminal of claim 5, wherein the scalable video encoder is configured to code the base temporal layer frames using a finer quantization than quantizations used for other temporal layers, whereby the said base layer is coded more accurately than the other layers.

10. The terminal of claim 5, wherein the scalable video encoder is configured to create at least one prediction chain which terminates at an enhancement layer.

11. The terminal of claim 5, wherein the temporally predictive codec further comprises an SNR quality scalability layer encoder.

12. The terminal of claim 11 wherein the SNR quality scalability enhancement layer encoder is configured with an input the residual coding error of the base layer obtained by subtracting the decoded base [illegible] om the original es and applying a positive offset and then encoding the difference in the same manner as the base layer encoder.

13. The terminal of claim 11, wherein the SNR enhancement layers encoder is further configured to use a prediction path which is different than the prediction path for the base layer or a lower enhancement layer.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 220 of 830

14. The terminal of claim 12, wherein the SNR enhancement layers encoder is further configured to drop the DC components of direct cosine transform (DCT) coefficients when predicting a coded video frame for a SNR quality enhancement layer.

15. The terminal of claim 12, wherein SNR enhancement layers encoder is further configured to quantize the DC and surrounding AC DCT coefficients at coarser level than the remaining DCT coefficients when encoding a video frame for an SNR quality enhancement layer.

16. The terminal of claim 11, wherein the SNR quality scalability layer decoder at the receiving endpoint is configured to display the decoded base layer frame in a desired reduced resolution by applying low-pass filtering and downsampling after decoding.

17. The terminal of claim 11, wherein the SNR quality scalability enhancement layer codec comprises an H.264 SVC FGS codec with threading.

18. The terminal of claim 17, wherein the spatial scalability layers codec comprises an H.264 SVC FGS codec that is configured to use a weighted average of previous enhancement layer pictures and a current base layer picture in a motion compensated prediction, wherein the weighting is dynamically varied to include the zero value at which value the prediction chain is terminated, whereby drift is eliminated..

19. The terminal of claim 11, wherein the SNR quality scalability layer encoder is configured to encode the differences in DCT coefficients before and after quantization by requantizing the differences and applying entropy coding to the requantized differences.

20. The terminal of claim 5, wherein the temporally predictive codec further comprises a spatial scalability layer encoder configured to low-filter and downsample the original input signal, wherein the lower resolution can be different than the intended display resolution, and use it as input to the base layer encoder.

21. The terminal of claim 20, wherein the spatial scalability layers encoder is configured so that a prediction path for an enhancement layer is different from that the prediction path for the base layer or a lower enhancement layer.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 221 of 830

22. The terminal of claim 20, where the spatial scalability layers encoder is configured to:

upsample the decoded low resolution signal to the original input signal resolution;

subtract the original input signal from the upsampled decoded low resolution signal to obtain a difference signal;

apply an offset to the difference signal; and

encode the offset difference signal.

23. The terminal of claim 22, wherein the spatial scalability layers encoder is configured so that the DC and surrounding DCT AC coefficients can be quantized more coarsely than the remaining DCT AC coefficients.

24. The terminal of claim 20 wherein the spatial scalability layer encoder is configured to use bi-predictive coding when predicting a high resolution video frame, and in which the first reference picture is a decoded past full resolution picture and the second reference picture is obtained by first encoding and decoding the downsampled base layer signal , and then upsampling it to the original resolution.

25. The terminal of claim 24 wherein the spatial scalability layer encoder is comprised of an H.264 AVC encoder with bi-predictive coding, in which the upsampled decoded base layer frames are inserted as additional reference frames, and in which the temporal and spatial direct modes of motion vector prediction are used to increase compression efficiency.

26. The terminal of claim 5 comprising a base layer encoder, which further comprises at least one of a SNR quality layers encoder, spatial scalability layers encoder, and temporal enhancement layers encoder.

27. An endpoint terminal for video communication with other endpoints one or more servers disposed in an electronics communication network, which network provides different quality of service and bandwidth channels linking the plurality of endpoints, the channels including designated HRCs, the endpoint terminal comprising:

a scalable video decoder or decoders, which is configured to scalably decode one or more video signals in a base layer and enhancement layers format; and

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 222 of 830

a packet demultiplexer, which is configured to demultiplex layers of the video signal encoded in the base layer and enhancement layers format after reception over the electronics communication network via a network interface controller.

28. The terminal of claim 27, wherein the decoder is comprised of an SNR quality scalability decoder.

29. The terminal of claim 28, where the SNR quality scalability decoder is configured to display the decoded base layer frame in a desired reduced resolution by applying low-pass filtering and downsampling after decoding.

30. The terminal of claim 28 wherein the SNR quality scalability enhancement layer decoder is configured to add to the decoded base layer frames the decoded residual error carried by the enhancement layer data after subtracting a positive offset.

31. The terminal of claim 27, wherein the decoder further comprises a spatial scalability decoder.

32. The terminal of claim 31, where the spatial scalability layers decoder is configured to:

upsample the decoded base resolution signal to the enhancement layer resolution;

decode the offset difference signal carried by the enhancement layer;

subtract an offset from the decoded enhancement layer signal and add the result to the upsampled decoded base resolution signal.

33. The terminal of claim 31, where the spatial scalability layer decoder comprises an H.264 AVC decoder with bi-predictive coding support, in which the upsampled decoded base layer frames are inserted as additional reference frames.

34. The terminal of claim 5, wherein the scalable video encoder is configured to encode the input signal in two or more spatial and/or quality resolutions that may be transmitted simultaneously.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 223 of 830

35. The terminal of claim 3, wherein the scalable coding structure may change dynamically depending on network conditions or indication of preferences by receiving endpoints, in any of the scalability dimensions.

36. A method for communication between a plurality of endpoints over an electronic communications network and one or more servers, which network provides different quality of service and bandwidth channels linking the plurality of endpoints, the channels including designated HRCs, the method comprising:

scalably coding a video signal in a base layer and enhancement layers format;

multiplexing layers of the video signal for transmission over the electronics communication network,

transmitting at least the base layer from the base layer and enhancement layers of the video signal for transmission over the designated HRCs.

37. The method of claim 36, wherein multiplexing layers of the video signal for transmission over the electronics communication network further comprising multiplexing an audio signal for transmission over the electronics communication network.

38. A method for communicating encoded video signal pictures, which are scalably coded in a base and enhancement layers format, between a plurality of endpoints over an electronic communications network, the method comprising:

selecting a picture type (I, P, or B) for the picture currently being encoded as well as a prediction reference picture from the decoded pictures stored in a frame memory; and

creating a temporal scalability layer by performing picture prediction using threads.

39. The method of claim 38 further comprising creating a continuous prediction chain path for the base layer.

40. The method of claim 38 wherein the picture threads include a base layer thread that includes pictures that are a number of pictures apart, and the method further comprises performing temporal prediction in each base layer picture using one or more of the preceding pictures of the said base layer picture,

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 224 of 830

wherein a temporal enhancement layer thread includes the remaining pictures, and the method further comprises performing prediction in each enhancement layer picture using either one or more of the preceding base layer pictures, or one or more preceding temporal enhancement layer pictures,

41. The method of claim 38 wherein the picture threads include a base layer thread that includes pictures that are a number of pictures apart, and the method further comprises performing temporal prediction using the immediately previous frame of the same,

wherein a first temporal enhancement layer thread includes that frames are half way between those of the base layer thread, and the method further comprises performing prediction from either the immediately preceding base layer frame or an immediately preceding first temporal enhancement layer thread picture, and

wherein a second temporal enhancement layer thread includes the remaining frames, and the method further comprises performing temporal prediction using a frame from either the immediately preceding second temporal enhancement layer thread picture, the immediately preceding first temporal enhancement layer thread picture, or the immediately preceding base layer thread picture.

42. The method of claim 38, further comprising coding the base temporal layer frames using a finer quantization than quantizations used for other temporal layers, whereby the said base layer is coded more accurately than the other layers.

43. The method of claim 38, further comprising creating at least one prediction chain, which terminates at an enhancement layer.

44. The method of claim 38, wherein scalably coding a temporal scalability layer by performing picture prediction using threads further comprises coding a SNR quality scalability enhancement layer

45. The method of claim 44, wherein coding a SNR quality scalability enhancement layer comprises applying a positive offset to a residual coding error of the base layer obtained by subtracting the decoded base layer frames from the original frames and then encoding the difference in the same manner as coding the base layer.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 225 of 830

46. The method of claim 44, wherein coding a SNR quality scalability enhancement layer comprises using a prediction path which is different than the prediction path for the base layer or a lower enhancement layer.

47. The method of claim 45, wherein coding a SNR quality scalability enhancement layer comprises dropping the DC components of Discrete Cosine Transform (DCT) coefficients when encoding a picture for the SNR quality scalability enhancement layer.

48. The method of claim 45, wherein coding a SNR quality scalability enhancement layer comprises quantizing the DC and surrounding AC DCT coefficients at coarser level than the remaining DCT coefficients when encoding a video frame for the SNR quality scalability enhancement layer.

49. The method of claim 44, wherein coding a SNR quality scalability enhancement layer further comprises at the receiving endpoint displaying the decoded base layer frame in a desired reduced resolution by applying low-pass filtering and downsampling after decoding.

50. The method of claim 44, wherein coding a SNR quality scalability enhancement layer further comprises an H.264 SVC FGS codec with threading.

51. The method of claim 50, further comprising using an H.264 SVC FGS codec that is configured to use a weighted average of previous enhancement layer pictures and a current base layer picture in a motion compensated prediction, wherein the weighting is dynamically varied to include the zero value at which value the prediction chain is terminated, whereby drift is eliminated.

52. The method of claim 44, wherein coding a SNR quality scalability enhancement layer comprises encoding the differences in DCT coefficients before and after quantization by requantizing the differences and applying entropy coding to the requantized differences.

53. The method of claim 38, wherein coding the temporal quality layers further comprises encoding a spatial scalability layer by applying a low-pass filter and downsampling the original input signal, wherein the lower resolution can be different than the intended display resolution, and encoding the downsampled signal in the same manner as the base layer.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 226 of 830

54. The method of claim 53, wherein encoding a spatial scalability layer comprises using a prediction path for an enhancement layer is different from the prediction path for the base layer or a lower enhancement layer.

55. The method of claim 53, wherein encoding a spatial scalability layer comprises:

upsampling the decoded low resolution signal to the original input signal resolution;

subtracting the original input signal from the upsampled decoded low resolution signal to obtain a difference signal;

applying an offset to the difference signal; and

encoding the offset difference signal.

56. The method of claim 55, wherein encoding a spatial scalability layer comprises:

quantizing the DC and surrounding DCT AC coefficients more coarsely than the remaining DCT AC coefficients.

57. The method of claim 53, wherein encoding a spatial scalability layer comprises:

using bi-predictive coding when predicting a high resolution video frame, and in which the first reference picture is a decoded past full resolution picture and the second reference picture is obtained by first encoding and decoding the the downsampled base layer signal and then upsamping it to the original resolution.

58. The method of claim 57, wherein encoding a spatial scalability layer comprises:

comprises using an H.264 AVC encoder with bi-predictive coding, in which the npled decoded base layer frames are insert reference frames, and in which the temporal and spatial direct modes of motion vector prediction are used to increase compression efficiency.

59. The method of claim 38 comprising using a base layer encoder, which further comprises at least one of a SNR quality layers encoder, spatial scalability layers encoder, and temporal enhancement layers encoder.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 227 of 830

60. A method for communicating encoded video signal pictures, which are scalably coded in a base and enhancement layers format, between a plurality of endpoints and one or more servers over an electronic communications network, the method comprising using:

a scalable video decoder or decoders, which is configured to scalably decode one or more video signals in a base layer and enhancement layers format; and

a packet demultiplexer, which is configured to demultiplex layers of the video signal encoded in the base layer and enhancement layers format after reception over the electronics communication network via a network interface controller.

61. The method of claim 60, wherein the decoder is comprised of an SNR quality scalability decoder.

62. The method of claim 61, further comprising using the SNR quality scalability decoder to display the decoded base layer frame in a desired reduced resolution by applying low-pass filtering and downsampling after decoding.

63. The method of claim 61, further comprising using the SNR quality scalability decoder to add to the decoded base layer frames the decoded residual error carried by the enhancement layer data after subtracting a positive offset.

64. The method of claim 60, wherein the decoder further comprises a spatial scalability decoder.

65. The method of claim 64, further comprising using the spatial scalability layers decoder to:

upsample the decoded base resolution signal to the enhancement layer resolution;

decode the offset difference signal carried by the enhancement layer;

subtract an offset from the decoded enhancement layer signal and add the result to the upsampled decoded base resolution signal.

66. The method of claim 64, wherein the spatial scalability layer decoder comprises an H.264 AVC decoder with bi-predictive coding support, the method

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 228 of 830

further comprising inserting upsampled decoded base layer frames as additional reference frames.

67. The method of claim 38, wherein scalably coding a video signal comprises encoding the signal in two or more spatial and/or quality resolutions that may be transmitted simultaneously.

68. The method of claim 38, wherein the scalable coding structure may change dynamically depending on network conditions or indication of preferences indicated by receiving endpoints, in any of the scalability dimensions.

69. A method for video communication between a plurality of endpoints over an electronic communications network and one or more servers, which network provides different quality of service and bandwidth channels linking the plurality of endpoints and servers, the channels including designated high reliability channels (HRCs) and low reliability channels (LRCs), the method comprising:

disposing transmitting and receiving terminals at the endpoints,

configuring at least one transmitting terminal to prepare at least one scalably coded video signal for transmission to other terminals in a base layer and enhancement layers format, and to transmit at least the base layer over a designated HRC,

configuring at least one receiving terminal to decode the scalably coded video signal layers that are received over network channels including a designated HRC, and to reconstruct a video for local use by combining the decoded video signal layers, and

configuring the servers to mediate the transfer of the scalably coded video signal layers transmitted by the transmitting terminal to a receiving terminal electronic communications network channe latter terminal

70. The method of claim 69, wherein configuring at least one transmitting terminal comprisies configuring the terminal to access at least one of: a live video signal for encoding and transmission; a stored video signal for encoding and transmission; a synthesized video signal for encoding and transmission and a pre-coded video signal for transmission.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 229 of 830

71. Computer readable media comprising a set of instructions to perform the steps recited in at least one of claims 36-70.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 230 of 830

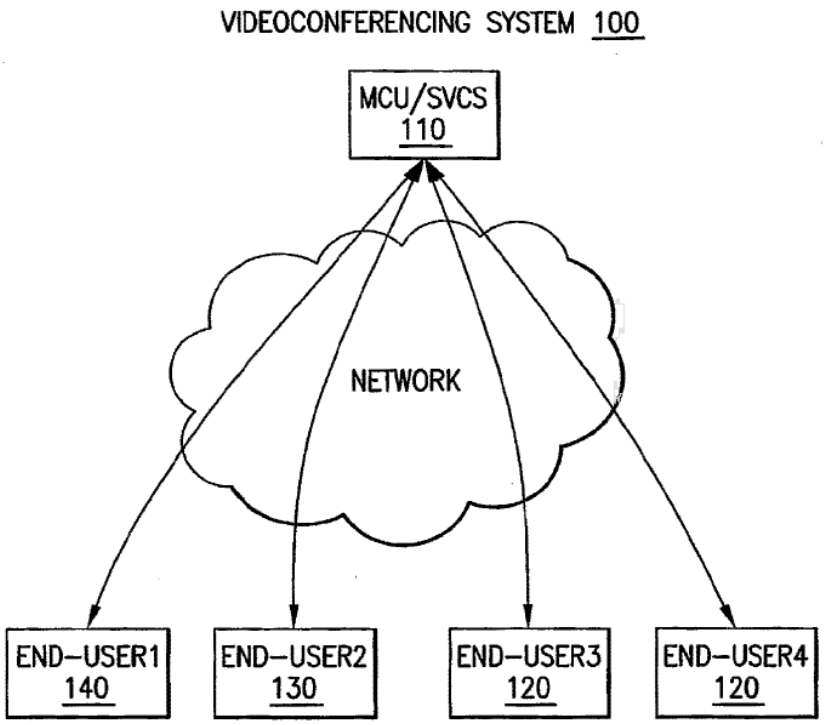


FIG.1A

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 231 of 830

SUBSTITUTE SHEET (RULE 26)

FIG.1B

SUBSTITUTE SHEET (RULE 26)

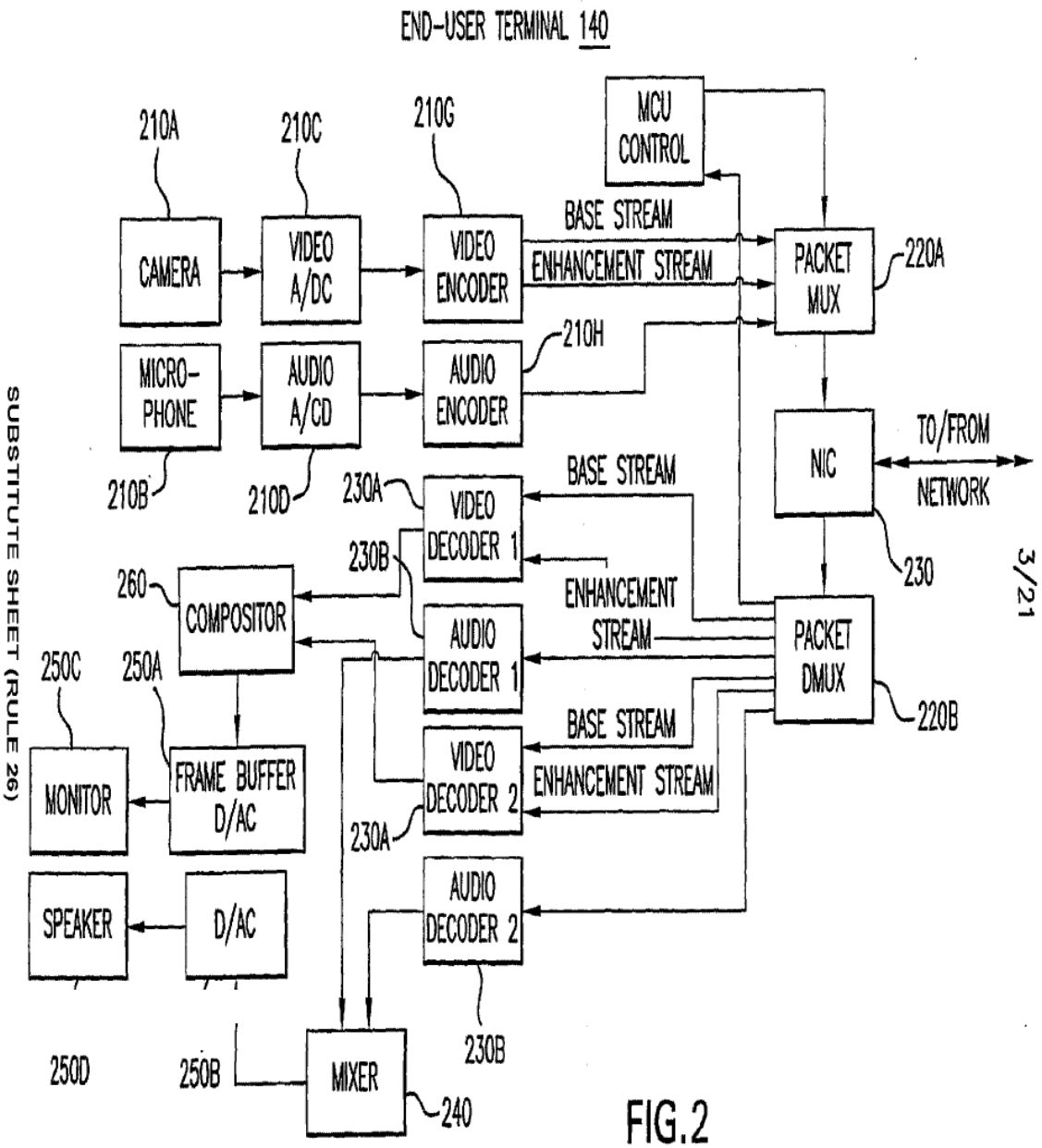


FIG.2

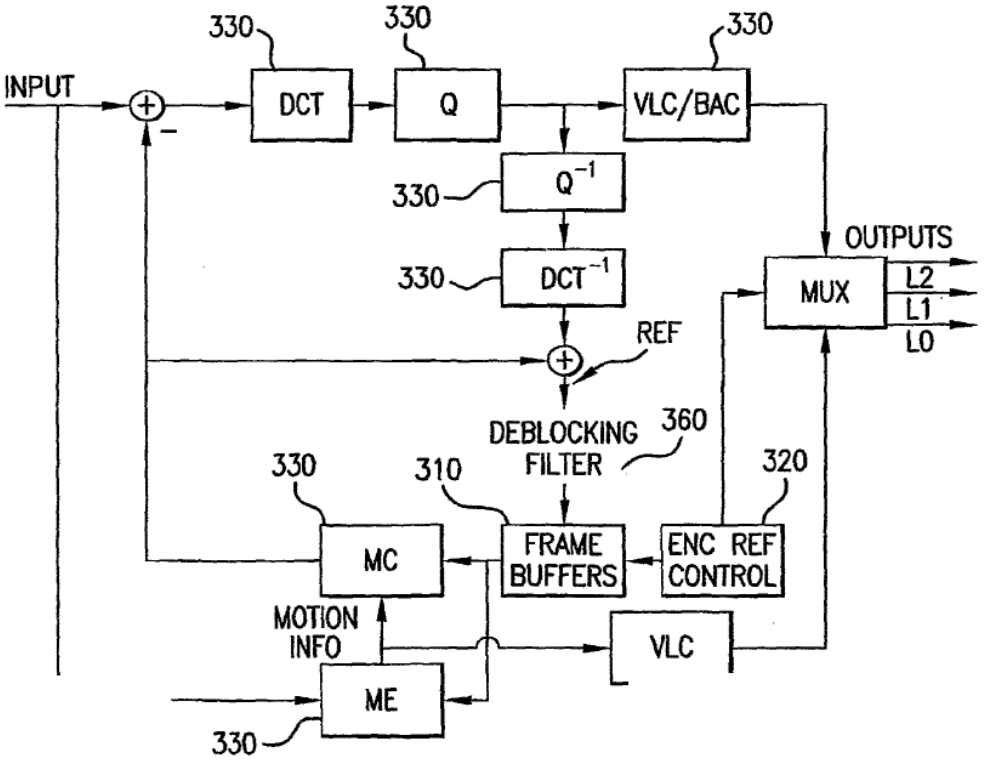

VIDEO ENCODER
(BASE AND TEMPORAL ENHANCEMENT LAYERS)
300
INPUT
330
DCT
330
Q
330
VLC/BAC
330
Q^{-1}
330
DCT^{-1}
OUTPUTS
MUX
L2
L1
L0
REF
DEBLOCKING FILTER
360
330
MC
310
FRAME BUFFERS
320
ENC REF CONTROL
MOTION INFO
VLC
ME
330

FIG.3

SUBSTITUTE SHEET (RULE 26)

FIG.4

SUBSTITUTE SHEET (RULE 26)

FIG.5

SUBSTITUTE SHEET (RULE 26)

FIG.6

SUBSTITUTE SHEET (RULE 26)

FIG.7

SUBSTITUTE SHEET (RULE 26)

FIG.8

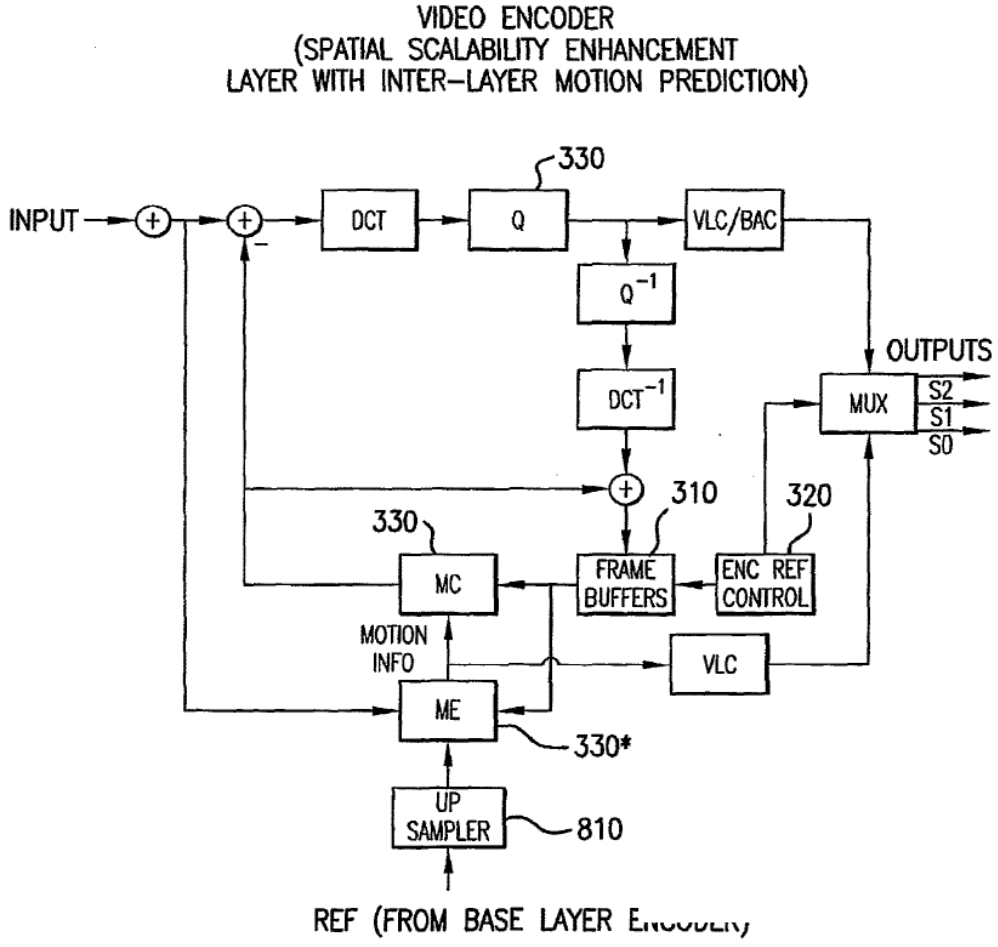

VIDEO ENCODER
(SPATIAL SCALABILITY ENHANCEMENT
LAYER WITH INTER-LAYER MOTION PREDICTION)
INPUT
DCT
Q
330
VLC/BAC
Q^{-1}
DCT^{-1}
OUTPUTS
MUX
S2
S1
S0
310
320
330
MC
FRAME BUFFERS
ENC REF CONTROL
MOTION INFO
VLC
ME
330*
UP SAMPLER
810
REF (FROM BASE LAYER ENCODER)

FIG.9

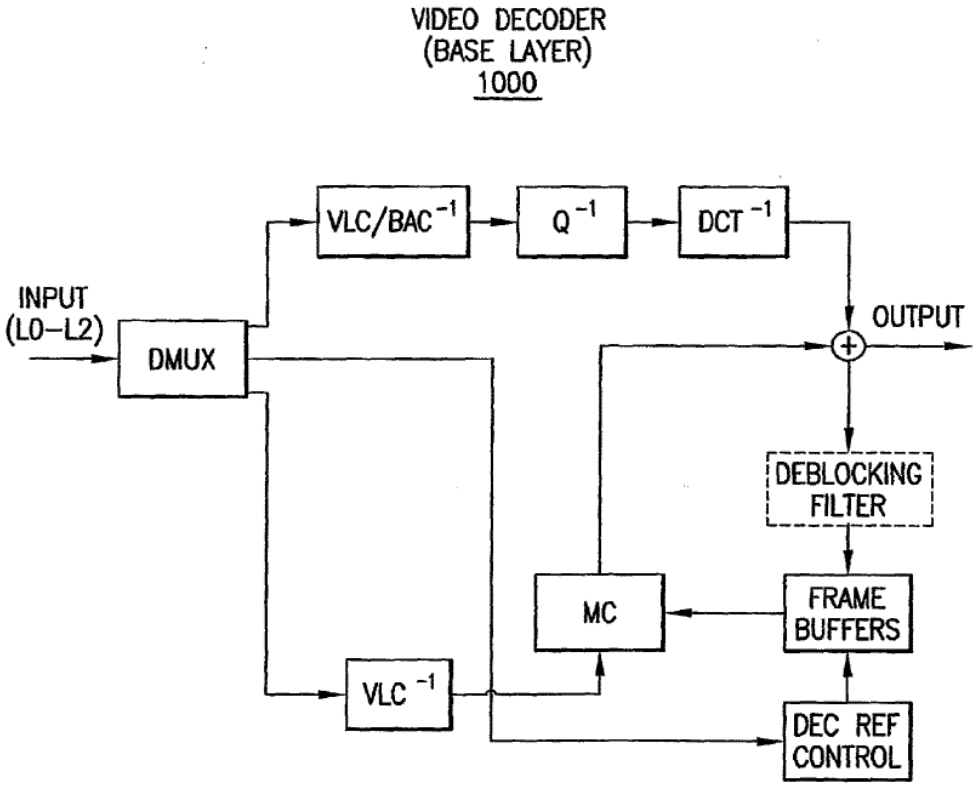

VIDEO DECODER
(BASE LAYER)
1000
INPUT
(L0–L2)
DMUX
VLC/BAC -1
Q -1
DCT -1
OUTPUT
DEBLOCKING
FILTER
MC
FRAME
BUFFERS
VLC -1
DEC REF
CONTROL

FIG.10

SUBSTITUTE SHEET (RULE 26)

FIG.11

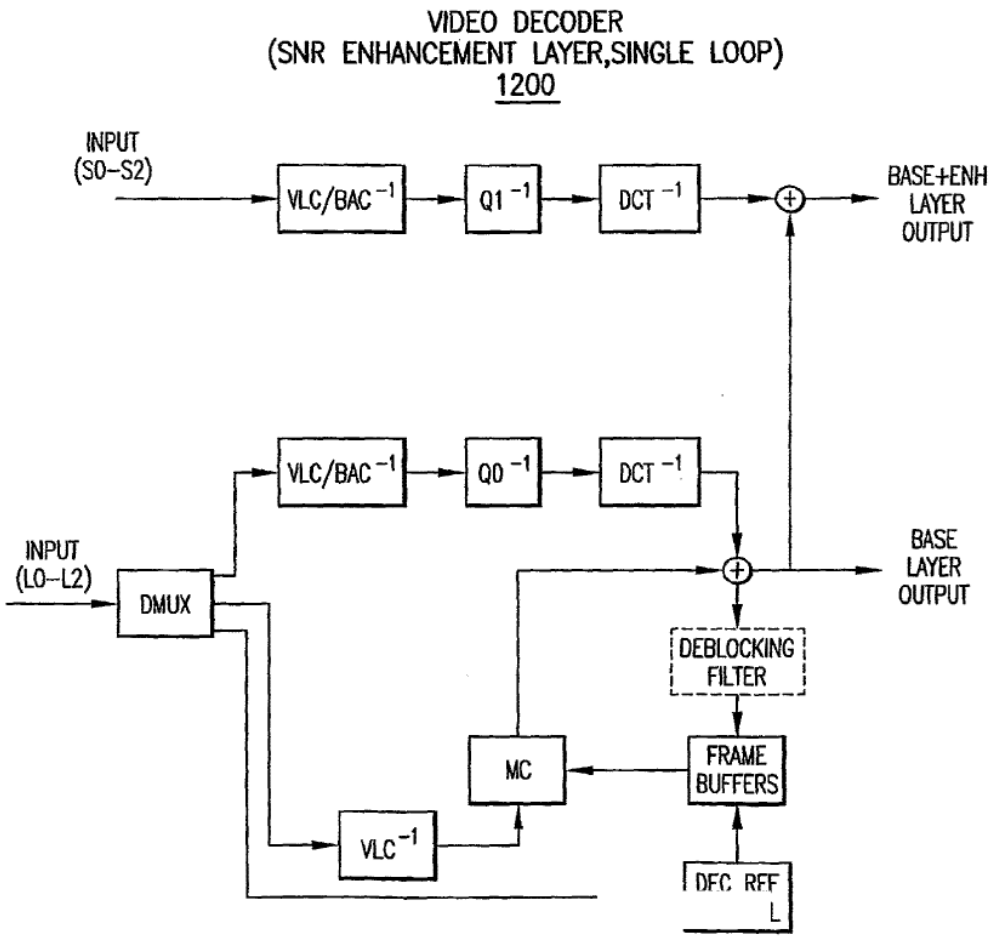

VIDEO DECODER
(SNR ENHANCEMENT LAYER,SINGLE LOOP)
1200
INPUT
(S0-S2)
VLC/BAC -1
Q1 -1
DCT -1
BASE+ENH
LAYER
OUTPUT
VLC/BAC -1
Q0 -1
DCT -1
INPUT
(L0-L2)
DMUX
BASE
LAYER
OUTPUT
DEBLOCKING
FILTER
MC
FRAME
BUFFERS
VLC -1
DFC RFF
L

FIG.12

SUBSTITUTE SHEET (RULE 26)

FIG.13

SUBSTITUTE SHEET (RULE 26)

FIG.14

SUBSTITUTE SHEET (RULE 26)

ALTERNATIVE LAYERED PICTURE CODING STRUCTURE 1580

FIG.15

ADDITIONAL THREADING STRUCTURES 1600

FIG.16

SUBSTITUTE SHEET (RULE 26)

FIG.17

SUBSTITUTE SHEET (RULE 26)

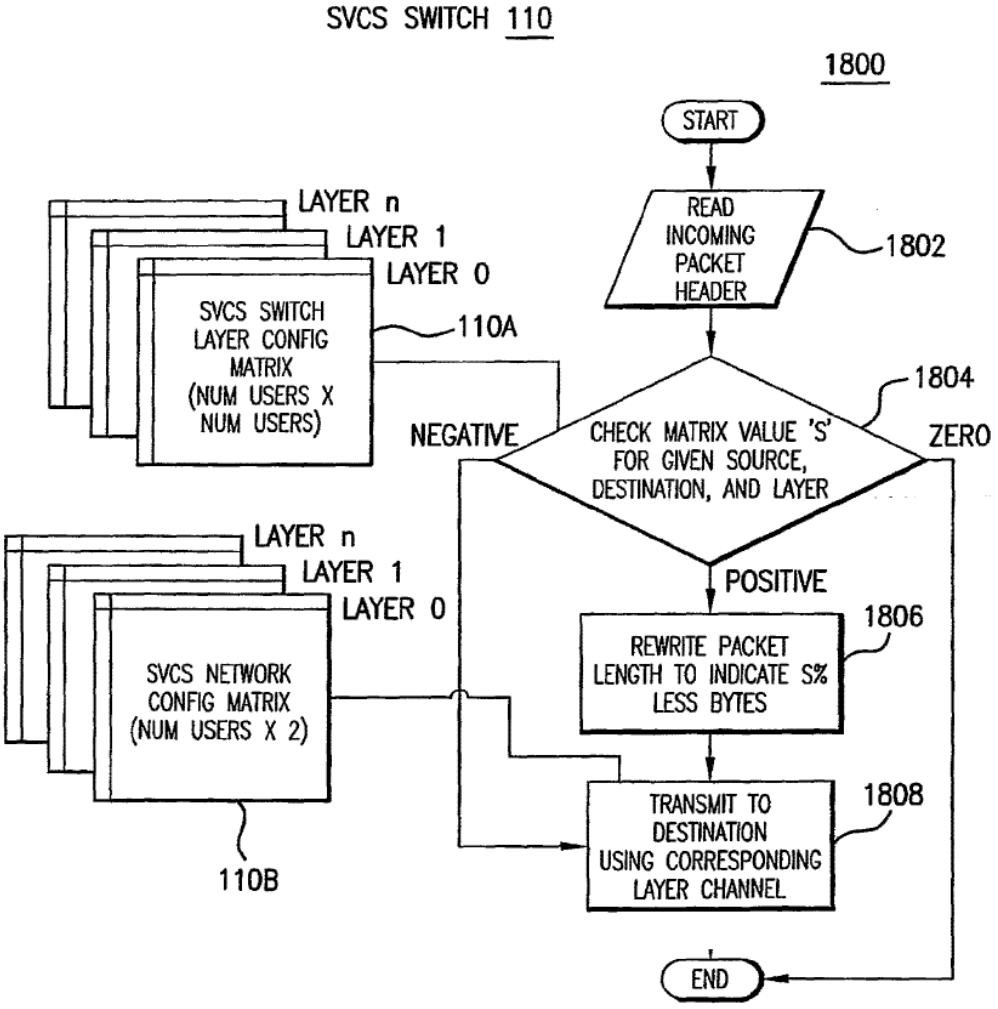

SVCS SWITCH 110
1800
START
READ INCOMING PACKET HEADER
1802
LAYER n
LAYER 1
LAYER 0
SVCS SWITCH LAYER CONFIG MATRIX (NUM USERS X NUM USERS)
110A
1804
NEGATIVE
CHECK MATRIX VALUE 'S' FOR GIVEN SOURCE, DESTINATION, AND LAYER
ZERO
POSITIVE
LAYER n
LAYER 1
LAYER 0
SVCS NETWORK CONFIG MATRIX (NUM USERS X 2)
110B
1806
REWRITE PACKET LENGTH TO INDICATE S% LESS BYTES
1808
TRANSMIT TO DESTINATION USING CORRESPONDING LAYER CHANNEL
END

FIG.18

SVCS SWITCH LAYER CONFIGURATION MATRIX

RECEIVED FROM \ TRANSMIT TO	USER 1	USER 2	...	USER n
USER 1	S11	S12	...	S1n
USER 2	S21	S22	...	S2n
...	...	...		...
USER N	Sn1	Sn2	...	Snn

FIG.19

SUBSTITUTE SHEET (RULE 26)

SVCS NETWORK CONFIGURATION MATRIX

CHANNEL ASSIGNMENTS

USERS	RECEIVE PORT	TRANSMIT PORT
USER 1	R1	T1
USER 2	R2	T2
⋮	⋮	⋮
USER N	Rn	Tn

FIG.20

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 251 of 830

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US06/28365

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - H04N 7/12 (2006.01)
USPC - 375/240.2
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8) - H04N 7/12 (2006.01)
USPC - 375/240.2

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
MicroPatent

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2004/0252758 A1 (KATSAVOUNIDIS et al) 16 December 2004 (16.12.2004) see whole document	1-71
A	EP 1 445 958 (FILIPPINI et al) 11 August 2004 (11.08.2004) see whole document	1-71
A	US 2004/0114817 A1 (JAYANT et al) 17 June 2004 (17.06.2004) see whole document	1-71
A	US 2004/0170395 A1 (FILIPPINI et al) 02 September 2004 (02.09.2004) see whole document	1-71

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
to an oral disclosure, use, exhibition or other
[illegible] prior to the international filing date but later than te claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with [illegible] cuments, such combination being obvious [illegible] irt
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
(Format 01 February 2006)	28 AUG 2007
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US06/28365

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fees must be paid.

Group I, claims 1-37 and 60-71, drawn to a system/device and method for video communication between a plurality of endpoints over an electronic communication network.
Group II, claims 38-59, drawn to a method for communicating encoded video signal pictures between a plurality of endpoints over an electronic communication network.

The inventions listed as Groups I and II do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons: the special technical feature of the Group I invention, mediating (multiplexing/demultiplexing) the transfer of scalably coded (encoded) video signals layers from a transmitting terminal to a receiving terminal and decoding (at a receiver at an endpoint) scalably coded (encoded) video signal layers as claimed therein, is not present in Group II; and the special technical feature of the Group II invention selecting a picture type as well as a prediction reference picture and creating a temporal scalability layer as claimed therein, is not present in Group I.

Since none of the special technical features of the Group I and II inventions is found in more than one of the inventions, unity of invention is lacking.

Form PCT/ISA/210 (extra sheet) (April 2005)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Erik Van ZIJST
Serial No.:	15/010,573
Filing Date:	January 29, 2016
Title:	Layered Multicast and Fair Bandwidth Allocation and Packet Prioritization
Confirmation No.:	9731
Examiner:	Not yet assigned
Art Unit:	2465
Attorney File No.:	GPB1-US9

INFORMATION DISCLOSURE STATEMENT

In accordance with 37 CFR §1.56, and in accordance with the provisions of 37 C.F.R. §§1.97 and 1.98, Applicant hereby makes disclosure of the patents, publications, and/or other information listed on the accompanying form PTO/SB/08A, which references are considered to be potentially material to the patentability of the invention disclosed in the above-referenced application. Copies of the non-US listed references are submitted herewith.

In accordance with 37 C.F.R. §1.97(h), the filing of this Information Disclosure Statement shall not constitute an admission that any information cited therein is, or is considered to be, material to patentability as defined in 37 C.F.R. §1.56(b). In the interest of full and complete disclosure to

[illegible] or all of the art cited herein may not be considered by Applicant(s) or the undersigned to be material under the new standards of materiality defined in 37 C.F.R. §1.56(b), enacted March 16, 1992, but may be material under the old standard of materiality defined in 37 C.F.R. §1.56(a), last amended on November 28, 1988, or may merely be technical background

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 254 of 830

which may be of interest to the Examiner. In accordance with 37 C.F.R. §1.97(g), the filing of this Information Disclosure Statement shall not be construed to mean that a search has been made.

This Information Disclosure Statement is being filed under 37 C.F.R. §1.97(b) within three months of the filing date of the application or before the mailing date of a first office action on the merits. No fee is required.

Written notification that these references have been considered in their entirety by return copy of the enclosed form, completed by the Examiner, is respectfully requested.

No fees are believed to be due at this time. However, the Commissioner is hereby authorized to charge any additional payment that may be due or credit any overpayment to Deposit Account 50-0392.

Respectfully submitted,

By: /Stephen Y. Liu/
Stephen Y. Liu
Registration No. 62,883
Attorney for Applicant

Date: February 11, 2016

CARSTENS & CAHOON, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001 Telephone
(972) 367-2002 Facsimile

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 255 of 830

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
15/010,573	01/29/2016	2465	800	GPB1-US9	14	2

CONFIRMATION NO. 9731

122200
Carstens & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380

FILING RECEIPT



Date Mailed: 02/17/2016

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Erik Van Zijst, Belmont, CA;

Applicant(s)

Blitz Stream Video, LLC, Plano, TX;

Power of Attorney: The patent practitioners associated with Customer Number 122200

Domestic Priority data as claimed by applicant

This application is a DIV of 13/619,969 09/14/2012
which is a CON of 12/486,656 06/17/2009 PAT 8514718
which is a DIV of 11/342,167 01/26/2006 PAT 7733868
which claims benefit of 60/647,601 01/26/2005

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: No

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 02/12/2016

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 15/010,573**

Projected Publication Date: 05/26/2016

Non-Publication Request: No

Early Publication Request: No
**** SMALL ENTITY ****
Title

LAYERED MULTICAST AND FAIR BANDWIDTH ALLOCATION AND PACKET PRIORITIZATION

Preliminary Class

370

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign
guide is available either by contacting the USI .er at 800-786-9199, or it
ıe USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for innovation, and commercialization of new technol ers tremendous resources for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 15/010,573

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	70		N/A	
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	300		N/A	
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	360		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	14 minus 20=	*	x 40 =	0.00	OR		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	2 minus 3 =	*	x 210 =	0.00			
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			0.00			
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				0.00			
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	730		TOTAL	

APPLICATION AS AMENDED - PART II

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)	OR	RATE($)	ADDITIONAL FEE($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

viously Paid For" IN THIS SPACE is less than 3, enter "3".

isly Paid For" (Total or Independent) is the highest found in the appropriate bo

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/010,573	01/29/2016	Erik Van Zijst	GPB1-US9

CONFIRMATION NO. 9731

122200
Carstens & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380

INFORMAL NOTICE

OC000000080622012

Date Mailed: 02/17/2016

INFORMATIONAL NOTICE TO APPLICANT

Applicant is notified that the above-identified application contains the deficiencies noted below. No period for reply is set forth in this notice for correction of these deficiencies. However, if a deficiency relates to the inventor's oath or declaration, the applicant must file an oath or declaration in compliance with 37 CFR 1.63, or a substitute statement in compliance with 37 CFR 1.64, executed by or with respect to each actual inventor no later than the expiration of the time period set in the "Notice of Allowability" to avoid abandonment. See 37 CFR 1.53(f).

The item(s) indicated below are also required and should be submitted with any reply to this notice to avoid further processing delays.

- A properly executed inventor's oath or declaration has not been received for the following inventor(s):
  Erik Van Zijst

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/hchin/

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

CARSTENS & CAHOON, LLP
P.O. BOX 802334
DALLAS TX 75380

MAILED
MAR 10 2016
OFFICE OF PETITIONS

Doc Code: TRACK1.GRANT

Decision Granting Request for Prioritized Examination (Track I or After RCE)	Application No.: 15/010,573

1. THE REQUEST FILED 1/29/16 IS **GRANTED**.

The above-identified application has met the requirements for prioritized examination

A. ☒ for an original nonprovisional application (Track I).
B. ☐ for an application undergoing continued examination (RCE).

2. **The above-identified application will undergo prioritized examination.** The application will be accorded special status throughout its entire course of prosecution until one of the following occurs:

A. filing a **petition for extension of time** to extend the time period for filing a reply;
B. filing an **amendment to amend the application to contain more than four independent claims, more than thirty total claims**, or a multiple dependent claim;
C. filing a **request for continued examination**;
D. filing a notice of appeal;
E. filing a request for suspension of action;
F. mailing of a notice of allowance;
G. mailing of a final Office action;
H. completion of examination as defined in 37 CFR 41.102; or
I. abandonment of the application.

Telephone inquiries with regard to this decision should be directed to Cheryl Gibson-Baylor at
fice of Petitions. In his/her absence, calls may an W. Brown,
).

Cheryl Gibson-Baylor
/Cheryl Gibson-Baylor/
[*Signature*]

Petitions Paralegal Specialist
(Title)

U.S. Patent and Trademark Office
PTO-2298 (Rev. 02-2012)

Doc code: Oath
Document Description: Oath or declaration filed

PTO/AIA/02 (07-13)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

SUBSTITUTE STATEMENT IN LIEU OF AN OATH OR DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION (35 U.S.C. 115(d) AND 37 CFR 1.64)

Title of Invention	Layered Multicast and Fair Bandwidth Allocation and Packet Prioritization

This statement is directed to:

☐ The attached application,

OR

☒ United States application or PCT international application number 15/010,573 filed on 1/29/2016

LEGAL NAME of inventor to whom this substitute statement applies:

(*E.g.*, Given Name (first and middle (if any)) and Family Name or Surname)

Erik VAN ZIJST

Residence (except for a deceased or legally incapacitated inventor):

City	State	Country
Belmont	CA	US

Mailing Address (except for a deceased or legally incapacitated inventor):

c/o Blitz Stream Video, LLC, P.O. Box 250649

City	State	Zip	Country
Plano	TX	75025	US

I believe the above-named inventor or joint inventor to be the original inventor or an original joint inventor of a claimed invention in the application.

The above-identified application was made or authorized to be made by me.

I hereby acknowledge that any willful false statement made in this statement is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

Relationship to the inventor to whom this substitute statement applies:

☐ Legal Representative (for deceased or legally incapacitated inventor only),

☐ Person to whom the inventor is under an obligation to assign,

☐ Person who otherwise shows a sufficient proprietary interest in the matter (petition under 37 CFR 1.46 is required), or

☐ Joint Inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 1 minute to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/AIA02 (07-13)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

SUBSTITUTE STATEMENT

Circumstances permitting execution of this substitute statement:

- ☐ Inventor is deceased,
- ☐ Inventor is under legal incapacity,
- ☐ Inventor cannot be found or reached after diligent effort, or
- ■ Inventor has refused to execute the oath or declaration under 37 CFR 1.63.

If there are joint inventors, please check the appropriate box below:

- ☐ An application data sheet under 37 CFR 1.76 (PTO/AIA/14 or equivalent) naming the entire inventive entity has been or is currently submitted.

OR

- ☐ An application data sheet under 37 CFR 1.76 (PTO/AIA/14 or equivalent) has not been submitted. Thus, a Substitute Statement Supplemental Sheet (PTO/AIA/11 or equivalent) naming the entire inventive entity and providing inventor information is attached. See 37 CFR 1.64(b).

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

PERSON EXECUTING THIS SUBSTITUTE STATEMENT:

Name: Thomas Crisman — Date (Optional): 03/16/2016

Signature: Thomas D. Crisman

APPLICANT NAME AND TITLE OF PERSON EXECUTING THIS SUBSTITUTE STATEMENT:

If the applicant is a juristic entity, list the applicant name and the title of the signer:

Applicant Name: Blitz Stream Video, LLC

Title of Person Executing This Substitute Statement: Member and Vice-President of Blitz Stream Video, LLC

The signer, whose title is supplied above, is authorized to act on behalf of the applicant.

nless provided in an application data sheet, PTO/A :

City | State TX | Country US

Mailing Address of the signer (unless provided in an application data sheet, PTO/AIA/14 or equivalent)

P.O. Box 250649

City Plano | State TX | Zip 75025 | Country US

Note: Use an additional PTO/AIA/02 form for each inventor who is deceased, legally incapacitated, cannot be found or reached after diligent effort, or has refused to execute the oath or declaration under 37 CFR 1.63.

Electronic Acknowledgement Receipt	
EFS ID:	25305135
Application Number:	15010573
International Application Number:	
Confirmation Number:	9731
Title of Invention:	LAYERED MULTICAST AND FAIR BANDWIDTH ALLOCATION AND PACKET PRIORITIZATION
First Named Inventor/Applicant Name:	Erik Van Zijst
Customer Number:	122200
Filer:	Stephen Y. Liu/Carolyn Bridwell
Filer Authorized By:	Stephen Y. Liu
Attorney Docket Number:	GPB1-US9
Receipt Date:	25-MAR-2016
Filing Date:	29-JAN-2016
Time Stamp:	11:18:53
Application Type:	Utility under 35 USC 111(a)

ion:

nt	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Oath or Declaration filed	GPB1-US9_Sub_Stmt_Van_Zijst_executed.pdf	3055742 b901241b61232041744c8efc9aefc2ca1263b047	no	2

Warnings:	
Information:	

Total Files Size (in bytes):	3055742

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (03-15)
Approved for use through 07/31/2016. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Not for submission under 37 CFR 1.99)

Application Number	15010573
Filing Date	2016-01-29
First Named Inventor	Van Zijst, Erik
Art Unit	2425
Examiner Name	Not yet assigned
Attorney Docket Number	GPB1-US9

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	6078564		2000-06-20	Lakshman	
	2	6650652		2003-11-18	Valencia	
	3	7370129		2008-05-06	Green	
	4	7626975		2009-12-01	Colban	
	5	7729719		2010-06-01	Bergstrom	
	6	7801142		2010-09-21	Jiang	
	7	7860120		2010-12-28	Wang	
	8	8224237		2012-07-17	Smith	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2425
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	9	7590131		2009-09-15	Ansorge	
	10	7746841		2010-06-29	Meiling	
	11	8514718		2013-08-20	Van Zijst	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20020071421		2002-06-13	Chiu	
	2	20130114594		2013-05-09	Van Zijst	
	3	20080126562		2008-05-29	Ludwig	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2425
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1		

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2425
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☒ See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Stephen Y. Liu/	Date (YYYY-MM-DD)	2016-05-02
Name/Print	Stephen Y. Liu	Registration Number	62883

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed . Time will vary depending upon the individual ents on the amount of time you m and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

EFS Web 2.1.17

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

system of records may be disclosed, as a rout... ...blic after either publication of rsuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

EFS Web 2.1.17

Electronic Acknowledgement Receipt	
EFS ID:	25659628
Application Number:	15010573
International Application Number:	
Confirmation Number:	9731
Title of Invention:	LAYERED MULTICAST AND FAIR BANDWIDTH ALLOCATION AND PACKET PRIORITIZATION
First Named Inventor/Applicant Name:	Erik Van Zijst
Customer Number:	122200
Filer:	Stephen Y. Liu/Carolyn Bridwell
Filer Authorized By:	Stephen Y. Liu
Attorney Docket Number:	GPB1-US9
Receipt Date:	02-MAY-2016
Filing Date:	29-JAN-2016
Time Stamp:	18:12:40
Application Type:	Utility under 35 USC 111(a)

ion:

nt	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	GPB1US9_IDS_Statement.pdf	66952 c79e0974874a02a5df83999be11ea397123 7bfca	no	2

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Form (SB08)	GPB1-US9_IDS.pdf	1035654 13f73cef9e5d05497ce48191ac6e9461bf0a54ae	no	5
Warnings:					
Information:					
			Total Files Size (in bytes):	1102606	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Blitz Stream Video, LLC
Serial No.:	15/010,573
Filing Date:	January 29, 2016
Title:	Layered Multicast and Fair Bandwidth Allocation and Packet Prioritization
Confirmation No.:	9731
Examiner:	Not yet assigned
Art Unit:	2465
Attorney File No.:	GPB1-US9

INFORMATION DISCLOSURE STATEMENT

In accordance with 37 CFR §1.56, and in accordance with the provisions of 37 C.F.R. §§1.97 and 1.98, Applicant hereby makes disclosure of the patents, publications, and/or other information listed on the accompanying form PTO/SB/08A, which references are considered to be potentially material to the patentability of the invention disclosed in the above-referenced application. Copies of the non-US listed references are submitted herewith.

In accordance with 37 C.F.R. §1.97(h), the filing of this Information Disclosure Statement shall not constitute an admission that any information cited therein is, or is considered to be, material to patentability as defined in 37 C.F.R. §1.56(b). In the interest of full and complete disclosure to the Office, some or all of the art cited herein may not be considered by Applicant(s) or the undersigned to be material under the new standards of materiality defined in 37 C.F.R. §1.56(b), , 1992, but may be material under the old s riality defined in 37 a), last amended on November 28, 1988, or may merely be technical background which may be of interest to the Examiner. In accordance with 37 C.F.R. §1.97(g), the filing of this Information Disclosure Statement shall not be construed to mean that a search has been made.

This Information Disclosure Statement is being filed under 37 C.F.R. §1.97(b) within three months of the filing date of the application or before the mailing date of a first office action on the

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 273 of 830

merits. No fee is required.

Written notification that these references have been considered in their entirety by return copy of the enclosed form, completed by the Examiner, is respectfully requested.

No fees are believed to be required. If, however, any other fees are required, I authorize the Commissioner to charge these fees to Carstens & Cahoon LLP, Deposit Account No. 50-0392.

Respectfully submitted,

By: /Stephen Y. Liu/
Stephen Y. Liu
Registration No. 62,883
Attorney for Applicant

Date: May 2, 2016

CARSTENS & CAHOON, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001 Telephone
(972) 367-2002 Facsimile

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 274 of 830

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Blitz Stream Video, LLC
Inventor:	Erik Van Zijst
Application No.:	15/010,573
Filing Date:	January 29, 2016
Confirmation No.	9731
Title:	Layered Multicast and Fair Bandwidth Allocation and Packet Prioritization
Examiner:	Hong, Michael Hyun
Art Unit:	2426
Attorney File:	GPB1-US9

PRELIMINARY AMENDMENT

Prior to examination of this application, please amend the application as shown below:

Amendments to the Specification begin on page 2 of this paper;

Amendments to the Abstract begin on page 3 of this paper; and

Remarks begin on page 4 of this paper.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 275 of 830

AMENDMENTS TO THE SPECIFICATION:

Please replace the entire specification with the Substitute Specification filed herewith. The specification has been amended to correct multiple typographical errors discovered during proofreading of the issued parent patent. The specification includes multiple subscript and superscript fonts that were not accurately captured during scanning; therefore, the substitute specification has been provided in larger font. Both a clean version and a marked version are being submitted herewith. No new matter has been added.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 276 of 830

AMENDMENTS TO THE ABSTRACT:

Please replace the Abstract with the following amended Abstract:

Abstract

Embodiments include an overlay multicast network. The overlay multicast network may provide a set of features to ensure reliable and timely arrival of multicast data. The embodiments include a congestion control system that may prioritize designated layers of data within a data stream over other layers of the same data stream. Each data stream transmitted over the network may be given an equal share of the bandwidth. Addressing in routing tables maintained by routers [[in the]] may utilize summarized addressing based on the difference in location of the router and destination address. Summarization levels may be adjusted to minimize travel distances for packets in the network. Data from high priority data stream layers may also be retransmitted upon request from a destination machine to ensure reliable delivery of data.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 277 of 830

REMARKS

Applicant has amended the specification and Abstract to correct multiple minor typographical errors. No new matter has been entered by the amendment. Entry of the above amendment is respectfully requested prior to examination of the application.

If there are any outstanding issues that the Examiner feels may be resolved by way of a telephone conference, the Examiner is cordially invited to contact Stephen Y. Liu or Vincent J. Allen at 972.367.2001.

Respectfully submitted,

By: /Stephen Y. Liu Reg. No. 62883/
Stephen Y. Liu
Registration No. 62,883
Attorney for Applicant

Date: May 20, 2016

CARSTENS & CAHOON, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001 Telephone
(972) 367-2002 Facsimile

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 278 of 830

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Blitz Stream Video, LLC
Inventor:	Erik Van Zijst
Application No.:	15/010,573
Filing Date:	January 29, 2016
Confirmation No.	9731
Title:	Layered Multicast and Fair Bandwidth Allocation and Packet Prioritization
Examiner:	Hong, Michael Hyun
Art Unit:	2426
Attorney File:	GPB1-US9

PRELIMINARY AMENDMENT

Prior to examination of this application, please amend the application as shown below:

Amendments to the Specification begin on page 2 of this paper;

Amendments to the Abstract begin on page 3 of this paper; and

Remarks begin on page 4 of this paper.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 279 of 830

AMENDMENTS TO THE SPECIFICATION:

Please replace the entire specification with the Substitute Specification filed herewith. The specification has been amended to correct multiple typographical errors discovered during proofreading of the issued parent patent. The specification includes multiple subscript and superscript fonts that were not accurately captured during scanning; therefore, the substitute specification has been provided in larger font. Both a clean version and a marked version are being submitted herewith. No new matter has been added.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 280 of 830

AMENDMENTS TO THE ABSTRACT:

Please replace the Abstract with the following amended Abstract:

Abstract

Embodiments include an overlay multicast network. The overlay multicast network may provide a set of features to ensure reliable and timely arrival of multicast data. The embodiments include a congestion control system that may prioritize designated layers of data within a data stream over other layers of the same data stream. Each data stream transmitted over the network may be given an equal share of the bandwidth. Addressing in routing tables maintained by routers [[in the]] may utilize summarized addressing based on the difference in location of the router and destination address. Summarization levels may be adjusted to minimize travel distances for packets in the network. Data from high priority data stream layers may also be retransmitted upon request from a destination machine to ensure reliable delivery of data.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 281 of 830

REMARKS

Applicant has amended the specification and Abstract to correct multiple minor typographical errors. No new matter has been entered by the amendment. Entry of the above amendment is respectfully requested prior to examination of the application.

If there are any outstanding issues that the Examiner feels may be resolved by way of a telephone conference, the Examiner is cordially invited to contact Stephen Y. Liu or Vincent J. Allen at 972.367.2001.

Respectfully submitted,

By: /Stephen Y. Liu Reg. No. 62883/
Stephen Y. Liu
Registration No. 62,883
Attorney for Applicant

Date: May 20, 2016

CARSTENS & CAHOON, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001 Telephone
(972) 367-2002 Facsimile

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 282 of 830

LAYERED MULTICAST AND FAIR BANDWIDTH ALLOCATION AND PACKET PRIORITIZATION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a divisional of pending U.S. Patent Application No. 13/619,969, filed September 14, 2012, which is a continuation of U.S. Patent Application No. 12/486,656, filed June 17, 2009, now U.S. Patent No. 8,514,718, which is a divisional of U.S. Patent Application No. 11/342,167, filed January 26, 2006, now U.S. Patent No. 7,733,868, which claims priority to U.S. Provisional Application No. 60/647,601, filed January 26, 2005, all of which are incorporated by reference as if set forth in full herein.

FIELD OF THE INVENTION

[0002] The invention relates to network management. Specifically, the invention relates to the management of data packets to support multicasting.

BACKGROUND

[0003] Despite the versatility in digital communication, interoperability and internationally accepted communication protocols of the Internet, its fundamental design has not changed much since its conception and does not excel in everything. Watching live TV for example is something which is not typically done over the Internet, even though television has been around almost twice as long as the tocol and represents a huge market. The reasons for this are based on the design of the Internet and Internet Protocol (IP).

[0004] The Internet is a packet-switching network where data is exchanged in small units or packets that are independently transported over the network and

concatenated again at the receiver into its original form. A strength of packet-switching is that it allows for very flexible use of the physical network wires. When two communicating parties have no data to exchange for a certain period of time, no packets are sent and the wires can carry packets from other parties. On the Internet, bandwidth is not reserved, but available to and shared by everyone. The consequence is that it cannot guarantee a minimum amount of end-to-end bandwidth, making live video streams often appear jerky because frames are skipped due to congestion that delays or prevents delivery.

[0005] Even though with help from specialized protocols such as Distance Vector Multicase Routing Protocol (DVMRP) or Protocol Independent Multicast (PIM), the Internet Protocol allows for data packets to be multicast to a large number of receivers simultaneously, using this feature to successfully realize a live video broadcast is a challenge. A video stream is transmitted at a fixed high rate and not all parts of the network are likely to have sufficient bandwidth available to forward the stream.

[0006] When a bandwidth bottleneck is reached, the router discards the packets that cannot immediately be forwarded. This causes two problems. The data stream that is eventually received by one or more receivers further down the network is corrupt and the congestion also has a negative impact on communication sessions of other nodes that communicate through the bottleneck router. The only way to avoid this problem using the Internet Protocol and standard multicast is to find a rate that is supported by all parts of the network. However, since the network is available to anyone, this rate will continuously change. A transmission rate is selected and the packet loss is accepted. However, when packets are dropped randomly by overloaded routers the data stream will suffer packet loss. If additional packets are sent through the bottleneck router, there is a larger chance

that the router will choose one of these packets when ready to send another packet, implicitly rewarding heavy streams during congestion. This encourages sending redundant data thereby exacerbating the problem.

[0007] A more fundamental problem of flow control using the Internet Protocol is that slowing down the data may not be an option for certain types of live data streams. However, packet loss is unavoidable using the Internet Protocol and while data types such as audio and video data can usually withstand some packet loss without becoming too corrupted to play, this does not apply to all types of live data. Real-time financial data, for example, will become useless and even dangerous to use if random packets of trades are lost.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Embodiments of the invention are illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to "an" or "one" embodiment in this discussion are not necessarily to the same embodiment, and such references mean at least one.

[0009] **Figure 1** is a diagram of one embodiment of an overlay multicast network.

[0010] **Figure 2** is a diagram of one embodiment of the basic components of a in the overlay multicast system.

[0011] **Figure 3A** is diagram of one example embodiment of a layered multicast system network divided into logical clusters.

[0012] **Figure 3B** is a diagram of a routing table for the network of **Figure 3A**.

[0013] **Figure 4** is a diagram of an example network with hierarchical structure.

[0014] **Figure 5A** is a diagram demonstrating summarization of an example network with an S2 domain.

[0015] **Figure 5B** is a diagram demonstrating the stretch factor for the example network of **Figure 5A** and shows the inner network of the S2 domain.

[0016] **Figure 6** is a diagram of an example embodiment of a multicast distribution or "sink" tree.

[0017] **Figure 7** is a diagram of an overlay multicast network with a failed link.

[0018] **Figure 8A** is a flowchart of one embodiment of a process or managing congestion in the overlay multicast system.

[0019] **Figure 8B** is a flowchart of one embodiment of a process for handling layer repair.

[0020] **Figure 9A** is a diagram of two router daemons connected by a link.

[0021] **Figure 9B** is a diagram of two router daemons connected by a link where a retransmission is requested.

[0022] **Figure 10** is a diagram that shows the process of selective packet repair and ordering.

Figure 11A is a diagram showing thinning of a stream of data.

[0024] **Figure 11B** is a diagram of a multi-level token ring structure.

[0025] **Figure 11C** is a diagram of a multi-level token ring structure showing thinning of data.

DETAILED DESCRIPTION

[0026] To provide multicast services a network needs to support one-to-many communication that can send data packets from a data source to more than one receiver, ideally without putting extra stress on the network or source when the number of receivers increases. Multicast routing can be offered by different methods. One method is to let receivers tell the network, but not necessarily the source, which data streams they want to receive and let the network compute data distribution paths that deliver just the right packets to each receiver. Multicasting can also be done by letting the source encode the list of receivers in each data packet, thereby freeing the network from the potentially computationally intensive task of maintaining multicast distribution paths. However, this method does not scale to handle a large number of receivers. A third method relies on logic at the receivers by letting the network apply a broadcast mechanism whereby each packet is delivered to every connected node and letting the receivers filter out only those packets that are interesting. This method may also generate a heavy load on a larger network but it is simple.

[0027] In one embodiment, a multicast network is constructed as an overlay network. In one embodiment, the overlay network includes a number of software implemented routers connected by normal IP or TCP connections. A mesh is created in which every software router is connected to one or more other software routers by means of virtual connections that appear to be direct connections to the are routers, but are likely implemented by a number of intervening traditional TCP/IP routers situated between the software routers. Two routers that are connected this way, are adjacent in the perspective of the overlay, but in reality are many physical hops away from one another. Also, a software router that has three software router neighbors has three independent virtual links. However, it is

possible that this software router only has a single physical network connection that is shared by the three virtual links.

[0028] To the underlying network, the overlay network is nothing more than a collection of applications that send data between static pairs. Beyond each router pair (identifiable by their TCP connection) there is no relation between the individual daemons. As such, the overlay network can easily work through firewalls, NAT (IP masquerading), proxies and VPN's. Firewalls cannot control which software router POPs can talk to each other. The system shares some principals with a HTTP proxy server tunneling traffic to and from web browsers. In an example HTTP proxy server system, an intranet has two web browser machines. They both browse the Internet using the proxy server also on the local intranet. Although the individual web browser machines can surf the net individually, they never make a direct connection with a remote webserver, but only with the local proxy server that acts like a relay point. As such, the firewall that sits between the proxy server and the remote webserver can only choose to either allow the proxy server to talk to the webserver or deny it. It is generally unable to enforce unique policies for individual browser machines. It generally cannot tell on behalf of which client the proxy server is fetching a webpage. Returning to the overlay network, because each software router is a relay point that tunnels data traffic for different senders and receivers similar to a HTTP proxy server, firewalls have no fine-grained control over communication over the overlay

ɔn as a firewall allows for only a sinɡ ction between an internal and an external POP, all software routers connected to the internal one can talk to all POPs connected to the external one, and vice versa, without restriction.

[0029] In one embodiment, the overlay multicast routing system also manages flow control and timely delivery. Non interactive live data streams do not actively

anticipate network congestion. To manage congestion the network manages the available bandwidth to allow for fair or equal division among the streams. Without management, high volume streams are assigned a larger capacity percentage on overloaded links resulting in little benefit in keeping the bandwidth requirements of a stream low, as the packet loss percentage is determined by how much the network is overloaded by all streams combined and not by the requirements of the individual streams. An alternative to letting the network handle the flow control and congestion is to put the responsibility at the source and receivers. However, letting data streams anticipate network conditions requires a form of feedback information from the network or the receivers. In this case it is beneficial that the amount of feedback does not grow linearly with the size of the audience, as that would reduce the scalability of the multicast. Even when a scalable form of feedback information can be realized and the data stream adapts its transmission rate according to the network conditions, the problem remains that live streams lose their value when they are slowed down and delivered late.

[0030] In one embodiment, an overlay multicast system also implements or manages delivery of packets including an option for guaranteed delivery. It would be ideal if every receiver would receive the data stream without loss or corruption. However, when the content is 'live' and cannot be slowed down, but the network has insufficient capacity, packet loss is difficult to avoid. In fact, even when there is sufficient bandwidth at all times, store-and-forward packet-switched networks guarantee the delivery of all packet e, when a router crashes, all packets in its buffers may be irrecoverably lost. If a network uses dynamic routing, packets are also dropped when routing policies change or packets are trapped in an occasional, temporary routing loop. In cases where packets are 'accidentally' lost, an end-to-end mechanism of retransmissions can be applied that

can compensate for the loss. However, since this requires a form of feedback information, it is beneficial for reasons of scalability that the overhead involved with retransmissions is not linearly related to the size of the audience.

[0031] End-to-end retransmission feedback may be avoided in at least two ways. First, it is possible to let the network components keep a copy of the most recently forwarded packets and let them participate in retransmissions by intercepting the retransmission requests and servicing them locally. This approach often utilizes greater storage and increased processing power requirements at the network components.

[0032] The second alternative to end-to-end retransmission requests is that of encoding redundant information in the data packets. If enough redundancy is encoded, a lost packet's content may be entirely recovered from the extra information in the other packets. The downside of this system is that it comes with a constant level of bandwidth overhead that is related to the level of packet loss tolerance, regardless of whether packets are actually lost. Each of these approaches to packet loss fail in the case of a live data stream that produces more bytes than the network can forward. When local or end-to-end retransmission requests are used, the problem may even be increased as the retransmission requests use extra bandwidth, causing more data packets to be lost.

[0033] Embodiments of the overlay multicast system are designed to offer an ıtion to efficiently multicast live data ck market data to any machine connected to the Internet. The system is capable of giving certain hard guarantees over what is delivered to receivers. If packets must be dropped due to congestion or other irrecoverable problems, it is done in a fully deterministic way that does not corrupt the data. Where a receiver of data such as a viewer of a

film may accept the random loss of one or two video frames, this type of data loss may wreak havoc in financial data when the missed packet contains an important trade. The system supports deterministically delivery designated parts of a data stream when the network lacks sufficient capacity. The system utilizes a layered multicast with data streams subdivided into individual streams or layers. This allows receivers to subscribe to only those layers that the network can handle, so that random packet loss can largely be avoided.

[0034] In another embodiment, an enhanced form of layered multicast is used that guarantees complete delivery for certain layers to avoid random loss altogether, making it suitable for certain types of critical data such as market data. The system is characterized as controlling two primary activities. The first activity is running and managing a robust and scalable overlay network that uses its own routing algorithms and supports multicast, while the second activity is managing flow control and congestion when live streams overload the network and ensuring layered multicast can be offered with guarantees.

[0035] **Figure 1** is a diagram of one example embodiment of an overlay multicast system. In one embodiment, the system includes an overlay network 100 that includes a number of software router daemons 109, 115, that are interconnected by normal TCP connections 119 or similar reliable communication connections. The overlay multicast system forms an intricate web of routers and virtual links similar to the Internet, but operating on the application layer. This [illegible] aling at the application layer 'overlays' the physical network and other layers of the Internet providing its own system of routing, controlling the lower levels of the network. Any number of routers and client applications may be a part of the system. In the example, two of the routers 109, 115 are in communication with local client applications 101, 111.

[0036] In one embodiment, each of the routers have a unique, string-based addresses 117. In another embodiment, the routers may have other types of unique addresses such as numerical addresses. Each of the router daemons 109, 115 executes a routing algorithm that computes the shortest paths that allows each router to send data packets to any other router in the overlay multicast system as well as any machine in communication with these routers.

[0037] In one embodiment, the system includes a runtime client library 103, 113, application programming interface (API) or similar system for sending and receiving packets that can be utilized by user applications to communicate over the system. This library connects an application to a nearby system router daemon 109, 115 through a TCP connection 107, 121, native inter-process communication or similar communication method. The native inter-process communication method may be used if both the router and the client application run on the same physical machine. When connected to a router daemon, the client application 101, 111 can send and receive data packets from the network under the router's unique address. Router daemons have a unique address, while user applications connected to a router daemon are identified through logical ports.

[0038] In one embodiment, the topology of the overlay network may be configured at router daemon startup. The topology may be relatively static after configuration. In one embodiment, the relationship of router daemons to applications may be one to many, with a single router daemon serving multiple . In another embodiment, the relationship may be many to many.

[0039] In one embodiment, the data packets in the overlay network may be addressed to a single communication endpoint or to abstract multicast addresses. Single communication endpoints are network addresses used by user applications.

Access to them is exclusive such that only one user application can use them for sending and receiving packets at a time.

[0040] In one embodiment, a unique network address in the overlay network that is used by an application is the combination of a logical node address assigned to the overlay router and the port name chosen by the application. If an application that is connected to an overlay router with the logical node address "nl.msx.1" wants to use a port named "myport" for receiving unicast data packets, the fully qualified network address becomes "nl.mxs.1:myport." Other applications connected to the overlay network that want to send packets to this application can use this as a destination address. Ports may be referred to as *tports*. Overlay network addresses must be bound, prior to sending or receiving packets. When a data packet is sent from an endpoint address it will contain this information as its source, allowing other routers or the receiving application to send a response.

[0041] In one embodiment, the overlay network may support packets being sent to abstract multipoint destinations, or multicast addresses. An overlay network multicast address is a destination that each machine on the network may subscribe to. The overlay network software routers ensure that a copy of each data packet published to this multicast address is delivered to every user application that subscribed to it. In one embodiment, multipoint destinations are single sourced. Only one application can publish data packets to the multicast address, making the overlay network suitable for one to many, but not many to many communications. *Port* session of a multicast address can be freely chosen, each multicast address explicitly contains the location of the source on the overlay network. This makes multicast communication less flexible because publishing is not anonymous, but it greatly simplifies subscription management. In another

embodiment, multiple applications may publish to a multicast address and many to many communication is supported.

[0042] In one embodiment, both unicast and multicast addresses are syntactically equal. For example, the address "nl.mxs.1:timeservice" could be a unicast address used and bound by an application that provides the local date and time in response to any data packet it receives. However, it could also be a multicast address that anyone can subscribe to. For example, subscription to the address may provide periodic date and time broadcasts from the user application that bound this address as a multicast address. In one embodiment, each packet may contain a flag or similar indicator that indicates whether its destination address should be interpreted as a unicast or multicast address.

[0043] In one embodiment, the overlay network provides a set of higher layered protocols as services over the packet oriented base layer. These services may be referred to as protocol endpoints. Any number of protocol endpoints may be defined and supported. In one embodiment, five protocol endpoints may be supported. A unreliable unicast protocol (UUP), a reliable unicast protocol (RUP), unreliable multicast protocol (UMP), ordered layered multicast protocol (OLMP) and reliable ordered layered multicast protocol (ROLMP) may be supported. The UUP offers a best effort unicast datagram service to applications. The RUP offers reliable unicast communication between peers. The UMP offers best effort multicast datagram services to applications on the overlay network. The OLMP offers unreliable multicast communication with receiver driven rate control. Complete delivery is not guaranteed, but the packets that are received are guaranteed to be in their original order. The ROLMP offers reliable multicast communication with receiver driven rate control. Stream layering allows each subscriber to receive the

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 294 of 830

data stream in the highest possible quality, while the source never has to slow down.

[0044] **Figure 2** is a diagram of one embodiment of the basic components of a router daemon in the overlay multicast system. Each overlay network router 109 includes a packet switching core or kernel 201. In one embodiment, packets that are received, either from a connection to a neighbor router or from a connected application, pass through the kernel 201. The kernel 201 may not handle some specialized control packets. The task of the kernel 201 is to inspect the destination of each packet and use a set of routing tables to determine how to forward the packet.

[0045] In one embodiment, the kernel 201 forwards packets to neighbors through a set of interface modules 203, 205. In one embodiment, the kernel 201 may execute on its own thread and be event driven. The kernel 201 remains idle until it is notified by an interface module 203, 205 of an incoming packet or by an application 227 that is sending a packet. The kernel thread is woken up, reads the packet from the interface module or application and processes it. If the kernel decides that the packet must be sent out through an interface, it passes it to that interface and waits for the next notification.

[0046] In one embodiment, each router daemon in the network is connected to one or more neighbors. This is done by establishing connections between the embodiment, the connections may t TCP connections or similar communication connections. In one embodiment, a router daemon 109 runs one interface module 203, 205 instance for each configured neighbor or configured communication channel. In another embodiment, the router 109 may run multiple interface modules for a configured neighbor or communication

channel or a single interface module for multiple configured neighbors or communication channels. The responsibility of an interface module 203, 205 is to establish a connection, e.g., a TCP connection, and to pass packets from the kernel 201 to the connection and vice versa. Packets of the first kind are referred to as outbound packets, while the latter are inbound packets.

[0047] In one embodiment, the kernel 201 maintains a unicast routing table that is used for packet switching. To make it possible for the network to find shortest paths as well as adjusting these paths when the underlying physical network's characteristics change, each interface module may measure the quality of its virtual connection. These measurements are passed on to the kernel 201 when the link quality is found to have changed. Inside the kernel 201, the measurements may be fed to a routing algorithm to determine if the changed link quality alters any of the entries in the routing table. If the routing table is changed, it is advertised to the neighbors by encoding the entire table or a portion of the table in a data packet and passing it to each interface. In one embodiment, if this type of packet is received from a neighbor and propagated to the kernel 201 through the receiving interface, the kernel inspects the message and analyzes it through the routing algorithm. If the new information leads to changes in routing entries, the router sends its own related routing entries to all its neighbors.

[0048] In one embodiment, if a neighbor router crashes, the interface detects this through an error on the virtual link and passes an indicator to the routing [illegible] routing algorithm then changes the cost indication associated with the link to an infinite value or similar value indicating that the link should not be utilized. In one embodiment, the kernel 201 does not distinguish between a crashed neighbor and an infinitely slow link. The kernel 201 only detects that the link is not to be utilized when reading the entry in the routing table.

[0049] In one example, when a source application connected to router *s* wants to publish a live data stream to multicast group s:mygroup, where applications on node *p* and *q* want to receive it, the source may first bind the group in publish-mode. Binding a multicast group in publish-mode means that the group is owned by the binding process. Only the owner of the group is able to publish to it. The receiving or "sink" applications connected to routers *p* and *q* now bind group s:mygroup in subscribe-mode. Binding a multicast group in subscribe-mode results in the router node being connected to a virtual multicast distribution tree. The subscribers receive a copy of any packet published by the source.

[0050] In one embodiment, multicast groups do not necessarily need to be bound in publish-mode first. Any application may subscribe to any multicast group at any time. If there is no source, there will be no data to receive. Binding a multicast group either in publish-mode or subscribe-mode are distinct operations. When the source also wants to receive a copy of its own stream, it binds its group in subscribe-mode and uses the resulting *tsocket* to read the data back. Data packets carry both the address of their source application, as well as the network address of their destination.

[0051] For unicast packets, the router uses its unicast routing table to find the preferred next hop, while multicast packets are routed according to a subscription list in a multicast subscription table. In one embodiment, a unicast data packet contains the node address of the source router, the *tport* of the source application, the node address of the destination router and the *tport* of the destination application. A multicast data packet contains the multicast group it was published to, represented by the node address of the source router and the group identifier that was bound by the source application. It does not contain any other addressing information.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 297 of 830

[0052] In one embodiment, the overlay multicast system determines unicast routing tables and multicast subscription tables for each router. The system utilizes any type of routing algorithms or protocols to determine routing tables. Protocols that are utilized include distance-vector protocols and link-state protocols. Link-state protocols take the relatively simple approach of propagating the state of the entire network as a list of all known links in the network with their cost. Distance-vector protocols are based on the Bellman-Ford protocol. They work by letting every router periodically advertise its own routing table to its neighbors. In one embodiment, the Extended Bellman-Ford protocol, hereafter referred to as ExBF, is used as the basis of the overlay multicast network. For sake of convenience, the embodiments of overlay multicast system are described that utilize the ExBF, however, those of ordinary skill in the art would understand that other routing algorithms may also be utilized.

[0053] In the ExBF protocol, a slight increase in the amount of information is kept for every destination. Instead of storing just the distance to each destination for every neighbor, ExBF also stores the address of the pre-final node of each path. Hence, instead of storing the collection of distances $\{Di(j)\}$ where j represents a destination, D the distance between this node and j, while i ranges over the node's neighbors, each router also stores $\{Pi(j)\}$ where P is the pre-final, or so-called 'head-of-path' node in the shortest path to destination j. Now because the router knows the pre-final node of every destination, it can backtrack the full path to any ecursively looking at the head-of-pa tion and treating it as a new destination.

[0054] In one embodiment, the interface modules 203, 205 implement a simple algorithm to establish the communication connection to the neighbor machine or device. The interface module 203, 205 attempts to connect to the configured

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 298 of 830

neighbor router by actively trying to connect to its network address, e.g., its TCP/IP network address. If the connection is aborted with a connection refused or other error, it is assumed that the neighbor is not yet running and the interface module 203, 205 starts to listen on its configured IP port so that the neighbor can connect to it when it is started up. The interface module 203, 205 waits for an incoming connection request for a brief period of time. After this time period expires, the interface module 203, 205 returns to actively connecting to its neighbor. To avoid a situation where both neighbors continue to switch between the listen and active states at the same time, the duration of the listening state is influenced by a random factor. An advantage of allowing each neighbor to switch between the active connection and listening states when establishing a connection is that it allows the connection of routers even if one of the routers is on a masqueraded IP network. A router on a masqueraded IP network is unable to accept incoming connections such as TCP connections.

[0055] In one embodiment, interface modules 203, 205 can be configured at router deployment through configuration files. This will make the router daemon automatically create the specified interfaces at startup. If the configured neighbor routers are online, all long-lived connections will automatically be established. It is also possible to add new neighbor connections and interface module instances dynamically at runtime. This way the overlay network topology can be changed flexibly and new routers can be added to the network. If an interface module 203, to establish a connection with a r er, the interface modules of the routers exchange overlay network node address information to inform each other of their presence. When the interface module 203, 205 receives the node address of its neighbor, it passes this information, together with an

announcement that a connection has been made to the kernel 201. This information allows the kernel routing algorithms to build dynamic routing tables.

[0056] In one embodiment, a role of the interface modules 203, 205 is to establish the connection with a configured neighbor router and to send data packets from the router kernel 201 to the neighbor connection and vice versa. The interface module 203, 205 incorporates a framework that allows custom software plug-ins to influence the stream of packets that flows between the network and the kernel 201. This mechanism is referred to as the interceptor pipeline 251, 257. In one embodiment, each software plug-in component that needs to control the packet flow is a class that implements a simple programming interface. In another embodiment, the software plug-ins may have any implementation structure including objected oriented structures. This interface allows interceptor instances to be chained together, forming a packet processing pipeline. The contract of an interceptor is that it receives a packet, applies it operations and then passes the modified packet to the next interceptor in the chain.

[0057] In one embodiment, an interceptor pipeline 251, 257 sits between the router switching core and the network connection. When the router kernel 201 delivers a packet to the interface module 203, 205 for transmission to the neighbor router, the interface module runs the packet through the interceptor pipeline, giving the interceptors the chance to modify the packet. Each packet that comes out of the pipeline is transmitted to the neighbor router.

[0058] In one embodiment, each interface module 203, 205 has two interceptor pipelines 251, 257. The first interceptor pipeline 257 is used to process outbound packets. The second interceptor pipeline 251 is used for inbound packets. These pipelines are independent of one another, the ordering of the interceptors and the

number of processing steps can be different for inbound and outbound packets. Each pipeline can be configured uniquely. Interceptor pipelines may have any number of interceptors, provided they do not add too much latency.

[0059] In one embodiment, an example of an interceptor is one that filters packets from a specific overlay network router. When this interceptor is implemented as a manageable component that can be configured dynamically at runtime, it can be used to implement basic firewall functionality on the overlay network. If it receives a packet that matches its rejection pattern, it discards the packet by not passing it to the next interceptor in the pipeline. Another type of interceptor that may be used is a traffic monitor that counts the size of each packet that passes by and uses this to log traffic activity and compute bandwidth statistics. This plug-in mechanism allows an overlay network router to be extended with additional functionality without modifications to the underlying software.

[0060] In one embodiment, the interceptor pipelines 251, 257 act as a packet buffer between the router kernel 201 and the network. The interface modules 203, 205 temporarily buffer inbound and outbound packets. This ensures that the kernel 201 is not blocked when sending a packet to an interface. In one embodiment, the interface modules have a separate thread or set of threads that continuously dequeue and serialize packets from the buffer and write them to the communication connection or enqueue into buffer received packets.

e embodiment, the interceptor pipeli temporary packet buffer and offers inter-thread communication between the kernel thread and the interface thread. Interceptors may be divided into two categories: interceptors that block and interceptors that return immediately. The first category is referred to as blocking or synchronous interceptors. In one embodiment, to avoid situations

where a router kernel 201 is blocked for an arbitrarily long time, an interceptor pipeline may contain one non-blocking interceptor. A non-blocking interceptor guarantees immediate return of control to a caller by storing the packet in an internal buffer. The packets in the buffer may be discarded if the buffer exceeds a certain threshold size.

[0062] In one embodiment, a maximum size limit is placed on the packet buffers to prevent them from exhausting the router's memory. Storing packets before processing them means that their delivery will be delayed. The larger the buffer gets, the longer the packets are delayed. Because of this, the interceptor drops packets when the buffer reaches its maximum size. In one embodiment, the system uses reliable TCP or similar connections for transmitting packets between routers in which packets are only dropped inside the buffer interceptors of the interface modules.

[0063] In one embodiment, the interceptor plays a role in making packet loss deterministic. Packets are not explicitly dropped in any other part of the layered multicast system network, except the buffer interceptor in the interface pipelines. However, that may not guarantee that a packet that successfully makes it through all buffer interceptors of the network's routers is delivered at its destination. Aside from controllable packet loss, the network may also occasionally lose packets in a non-deterministic way, for example when a router crashes with pending packets in its buffers, or when a connection between adjacent routers is unexpectedly closed

mission of a packet.

[0064] In one example, an inbound interceptor pipeline 251 may be structured such that traffic throughput monitor interceptors 253 are positioned after a buffer interceptor 255 and a firewall interceptor 265 may be positioned before for the

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 302 of 830

buffer interceptor 255. In an example outbound interceptor pipeline 257, a throughput limiting interceptor 259 may be positioned before a buffer interceptor 261 and a traffic monitor interceptor 267 may be positioned after the buffer interceptor 261.

[0065] In one embodiment, the overlay network is accessible to applications through the use of the client-side programming library 221. This library connects to a router daemon 109 at application startup and communicates with it using remote procedure protocol or similar communication protocols. The communication between the library 221 and the router daemon 109 is driven by the application 227. The application 227 invokes functions in the router 109 through the library 221. In one embodiment, when the kernel 201 receives packets addressed to a *tport* that is bound by the application 227, it stores them until the client actively picks them up. The application 227 through the library 221 continuously polls the router for packets. To minimize the overhead of the polling mechanism, the router poll function does not return until there is at least one packet delivered by the kernel 201. If more than one packet is waiting, the poll function returns all waiting packets at the time it is invoked. In another embodiment, the router may send an indication such as an invoking a marshaled stub from the client, event notification or similar indicator to the application 227 through the library 221 to indicate the reception of a data packet for the application 227.

[0066] In one embodiment, a user application 227 uses a client library 221 ances of the overlay multicast system communication sockets 225. If a user wants to be able to receive packets, they reserve a network address. In one embodiment, a network address in the overlay multicast system is represented as a combination of the address of the router and a unique port identifier, reserved for the socket of the router. For sake of convenience, a port in the layered multicast

system will be referred to as a *tport* and a socket as a *tsocket*. When a user application creates a *tsocket* for receiving normal unicast packets, the *tsocket* automatically binds a local *tport* at the router daemon through a remote call to the client IO multiplexor 207 at the router. In one embodiment *tports* are bound exclusively and other clients may not use it concurrently.

[0067] In one example embodiment, communication sockets 225 communicate with a local input output (IO) multiplexor 229 that coordinates the handling of communication between the sockets and the router kernel through the formation of RPC calls, native inter-process communication or similar systems. A local IO multiplexor 229 utilizes an RPC stub 223 or similar program to communicate with the router via an RPC skeleton 219 and client adaptor 209. A client IO multiplexor 207 at the router manages the relay of these socket requests to the kernel 201.

[0068] In one embodiment, the router daemon 109 uses a packet buffer to temporarily store packets for each connected client application until they are picked up. An interceptor pipeline in the client adaptor 209 or similar process may be utilized for this buffering function.

[0069] To overcome the problem of growing routing tables, computation time and excessive advertisement overhead, large networks can be partitioned into smaller sub sections, connected by gateways. While the gateway routers maintain routing information necessary to route packets to nodes in other network sections, section or domain only maintain infc ıose nodes inside the same domain. By substituting logical ranges of hosts in the routing table by one single condensed entry, the size of the routing table is reduced. This is called address summarization. This mechanism introduces a form of hierarchy that allows the network as a whole to grow far beyond the practical limits of standard distance-

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 304 of 830

vector or link-state algorithms. The farther a destination host is away, the more efficient it can be condensed together with other remote hosts. The more summarization is applied, though, the less efficient the paths become on average. The factor by which the actual data paths on summarized networks differ from the optimal paths, is known as the *stretch-factor:* the maximum ratio between the length of a route computed by the routing algorithm and that of a shortest path connecting the same pair of nodes.

[0070] In one embodiment, the overlay multicast system takes a relatively straightforward approach to address summarization. An administrator decides at deploy time which nodes form clusters and which clusters form aggregations of clusters. This is done by encoding hierarchy in the layered multicast system node addresses using a dotted or similar notation. Node addresses may be ASCII strings. In one embodiment, the strings are limited to at most 127 characters. In one embodiment, only [a-z] and [0-9] are available. In another embodiment any characters, numbers of similar symbols may be utilized. Addressing may be case-sensitive or case-insensitive.

[0071] **Figure 3A** is diagram of one example embodiment of a layered multicast system network divided into logical clusters. The example illustrates a network of eight nodes, divided into three clusters. Assigning nodes to clusters may be based on geographical properties, administrative boundaries, wide area links and similar considerations. For example, nodes inside a corporate network are all assigned the u domain, whereas a network that connects nodes from different corporations, would usually assign a separate domain to each corporate network. Another criterion for assigning nodes is that nodes that often disconnect and reconnect again later, possibly because the overlay multicast system routers run on

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 305 of 830

personal computers, are placed in a subdomain, to avoid the routing updates triggered by their changing state to propagate far into the network.

[0072] Given the logical domains that cluster groups of nearby nodes, each node can treat domains other than its own as a single entity and use a wildcard address that matches every host inside that domain. This reduces the size of the routing table, as well as the amount of routing information that needs to be exchanged between the nodes when the topology changes inside a domain. For example, when a new host is added to domain S1, there is no need to propagate that information to the other domains, as they already have a wildcard entry that will match the new address.

[0073] **Figure 3B** is a diagram of a routing table for the network of **Figure 3A.** The example routing table shows the effect of address summarization in this network on the routing table of node S2.C. The cost value that is listed in the third column represents the cost of the shortest path to the nearest node inside that domain. In the fourth routing entry, the number in the cost column is the cost to reach S1.R from S2.C. If it is assumed that both interdomain links S2.C - S1.R and S2.A - S1.R have the same weight or cost, S2.C will route all traffic for domain S1 through neighbor S1.R.

[0074] In one embodiment, when nodes exchange distance vectors, either because of a link change, or as a normal, periodic exchange, the receiving node s the destination address of each dis elative to its own address. Summarization is done by matching the destination address with its own address field-by-field and when a field does not match, the rest of the address is substituted by a wildcard. For example, when S2.C receives a distance-vector from S1.R that contains a path to destination S1.P, it is immediately summarized to S1.*

upon arrival at node S2.C. This is because the first field differs from the first field of the local address and as such the remaining fields are replaced by a wildcard. This wildcard value is then fed to the distance vector algorithm that checks whether the new cost or path-length is shorter than the cost of the entry that was already in the routing table. In the present example there already is an S1.* wildcard entry in the routing table that was derived from neighbor destination S1.R. Since the path to S1.P runs through S1.R, the path to S1.R will always be shorter than the path to S1.P, so the entry in the routing table will not be changed and no further routing updates will be propagated to S2.C's neighbors.

[0075] In general, when a local address is 1.2.3.4 and an incoming distance-vector advertises destination 1.2.2.2, it will be summarized to 1.2.2.*. Destination 2.1.6 becomes 2.*, destination 1.2.3.4.5.6 becomes 1.2.3.4.5.*, destination 1.2.3.5 stays 1.2.3.5 and destination 1.2.3.4.5 also stays 1.2.3.4.5. Note that 1.2.3 and 1.2.3.* are two different addresses. The first only matches the exact address 1.2.3 while the second is a wildcard that matches everything that starts with 1.2.3 and has at least 4 address fields. This includes 1.2.3.4 and 1.2.3.4.5.6, but not 1.2.3. If this mechanism of address summarization is used in an ExBF implementation that carries tuples containing destination, cost and head-of-path attributes in its vectors, then the destination address is summarized, as well as the head-of-path address.

[0076] **Figure 4** is a diagram of an example network with a hierarchical structure. When the example network first starts to converge using the ExBF protocol, node B.X advertises the following routing information to neighbor A.S: $DV_{B.X,A.S}$: {(B.X, *, 0), (B.Y, B.X, 1), (B.Z, B.X, 1)}. When A.S receives the vectors, it summarizes its entries. What remains is the single vector $DV_{B.X,A.S}$: {(B.*, *, 0)}. The routing table of node A.S now contains $RT_{A.S}$: {(A.S, *, *, 0), (A.R, A.R, A.S, 1), (B.*, B.*, A.S, 1)}. All addresses are

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 307 of 830

summarized before processing. This includes neighbor addresses. The consequence of this is that when a node has more than one connection with a foreign domain, both neighbor addresses will be summarized into the same wildcard. This leads to ambiguities and nondeterministic routing when this wildcard is listed as the preferred hop in a routing entry, as it cannot identify a single outgoing link. This problem is solved by assigning a local identifier to each link and using these numbers in the preferred hop column, rather than the addresses of the links peers. The routing update A.S sends to A.S.Y contains $DV_{A.S,A.S.Y}$: {(A.S, *, 0), (A.R, A.S, 1), (B.*, A.S, 1)} and leads to A.S.Y's routing table $RT_{A.S.Y}$: {(A.S.Y, *, *, 0), (A.S.X, A.S.X, A.S.Y, 1), (A.S.Z, A.S.Z, A.S.Y, 1), (A.S, A.S, A.S.Y, 1), (A.R, A.S, A.S, 2), (B.*, A.S, A.S, 2)}. When A.S.Y has finished updating its routing table, it advertises $DV_{A.S.Y,A.S}$: {(A.S.Y, *, 0), (A.S.X, A.S.Y, 1), (A.S.Z, A.S.Y, 1), (B.*, *, *), (A.S, *, *), (A.R, *, *)} back to neighbor A.S where the asterisk indicates unreachable in the last three records to avoid long-lived loops. These loops were detected through the normal back trace mechanism of ExBF.

[0077] In one embodiment, summarization means that only a single path for a range of remote nodes is maintained. The consequence of this is that packets will not always be routed according to the real shortest path between source and destination. To illustrate this effect in the multicast network, consider the example network of **Figure 4.** Node S1.R receives the distance-vectors of both S2.A and S2.C. After summarization learns that both S2.A and S2.C offer a route for wildcard S2.*. If it is assumed that both interdomain links (S1.R - S2.A and S1.R - S2.C) have equals costs, then S1.R will choose S2.A to be the preferred hop for S2.* because of the fact that its address logically comes first. When S1.R needs to forward a data packet to S2.B, it uses the S2.* wildcard entry and forwards the

packet to neighbor S2.A. Unfortunately this is not the shortest path to S2.B, as S2. A first has to route the packet through S2.C. Instead, S1.R should have sent the packet directly through neighbor S2.C. This ratio between the length of the actual path and the length of the optimal path between the endpoints are called the stretch factor.

[0078] In one embodiment, path stretching on the overlay multicast system occurs when packets are forwarded between nodes that are in different logical domains. An entire subnet is treated as a single node with several outgoing links. Because a virtual node often contains a large number of nodes, connected by its own internal network structure, it is sometimes better to choose a different interdomain link when sending packets to the domain.

[0079] Since the overlay network runs its own adaptive routing algorithms, the content streams through the network are constantly rerouted to avoid the network's hot spots and congestion. This can be particularly useful on wide area networks that are used for very different types of applications and hot spots are dynamic (i.e., moving around). Another advantage of having custom routing algorithms is the freedom of substituting them with others in the future.

[0080] In one embodiment, the software routers implement load balancing inside the network. Traditionally routing algorithms seek for "best paths" through the network and send data streams over these. However, it is sometimes much not to just send streams over this c ɔut to also select several sub-optimal paths and divide the stream over all of them. This also avoids the optimal path from getting congested when the stream requires more bandwidth than this single path can provide.

[0081] **Figure 5A** is a diagram demonstrating summarization of an example network with an S2 domain. When node S1.P needs to forward a packet for destination S2.F, it will send the packet directly down its interdomain link to the S2 domain. In this example both interdomain links are assumed to have equal costs. **Figure 5B** is a diagram demonstrating the stretch factor for the example network of **Figure 5A** and shows the inner network of the S2 domain. **Figure 5B** clearly shows that the stretch-factor is quite high for packets from S1.P to S2.F, as the optimal path runs through S1.R instead. Although the overlay multicast system address summarization technique cannot guarantee a maximum upper bound on the stretch-factor, it can manipulate the stretch-factor by changing the summarization opacity.

[0082] By default, an address is summarized after the first field that differs from the local address. However, if that is changed to the second field, the overlay multicast system can look inside other domains for one level. A node with an address 1.2.3.4 will then summarize 2.3.4.5 into 2.3.*, rather than 2.* and 1.3.4.5 into 1.3.4.*, rather than 1.3.*. Doing this at least at the border nodes in the overlay multicast network that have the interdomain links reduces the stretch-factor under certain circumstances. As the overlay multicast network was designed to be administrated by independent parties, administrators are free to experiment with different summarization opacity levels without jeopardizing the other subnets or domains.

The advantage of the summarization of addresses in routing tables is that the number of links that are present between nodes from different logical domains is irrelevant with respect to the number of routing entries in the routing table of a distance-vector protocol or similar protocols. Since these tables only contain destinations with a single forwarding strategy, the number of interdomain links

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 310 of 830

does affect the size of the routing tables. If y is identified to be the number of entries in a node's routing table, x to be the total number of nodes in the entire network, n to be the number of entities (nodes or nested domains) inside a domain and m to be the depth of the hierarchy. From this it follows that the total number of nodes in the network can be calculated with:

$$\forall\, x = n^m \text{ where } n \in \mathbb{N} \wedge n > 1 \text{ and } m \in \mathbb{N} \wedge m > 0$$

[0084] The relation between domain depth, number of nodes and domain density can be expressed by the following formula for a network with a uniform topology:

$$y = \left(\frac{\ln(x)}{\ln(n)}\right) * (n-1) + 1$$

[0085] Given this formula of the address summarization's effectiveness, it can be shown that routing entries in any network node remains under 100 even when the network as a whole grows to well over 10 million nodes. It can also be shown that the domain size n has no spectacular effect on the scalability. While a small value n yields a relatively deeply nested network hierarchy, which implies more routing entries, it also means that each hierarchy level only contains a small number of entities. A large value n in a network of the same size yields a relatively flat hierarchy with few levels, each level contains a large number of entities. As a [illegible] an only contain a natural number (number remains constant as nodes are added to domains that are already captured by the existing wildcard entries.

[0086] An advantage of using a string based node addresses scheme is its large address space and flexible naming. New subdomains may be added at any time as

there is no need for predefined or fixed hierarchical levels. All nodes inside a flat domain can be reconfigured into nested subdomains without the need to reconfigure routers or compute netmasks. Also, as address fields are not bound to a limited number of characters, they contain location or company names to make administration of the network easier and more transparent. An example might be the address nl.amsterdam.level3.marketxs.node4, or simply nl.ams.13.mxs.n3 to keep the address relatively short. The drawback of these kinds of unrestricted addresses is the amount of bytes they require. Because every data packet always contains a source and a destination address, the amount of overhead per packet is excessive at times.

[0087] In one embodiment, the string based node address scheme is implemented by letting the first bytes of each address header specify the length of the address in bytes, followed by the address itself as a normal 8 bit ASCII string with a maximum of, for example, 256 bytes. In one alternative, null-terminated strings are used.

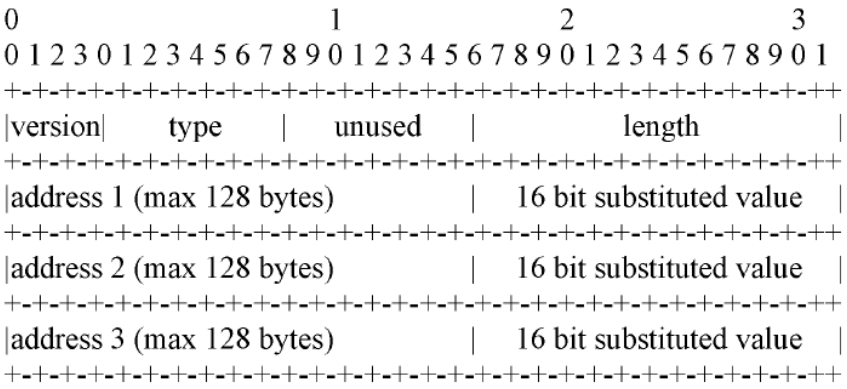
[0088] In another embodiment, a sufficiently large address space is used and divided into logical subsets, similar to IPv4 and IPv6 addresses. In one embodiment, a 64 bit address space is used and divided into 8 bit fields (identical to IPv6) used to identify logical subnets. In this embodiment, addresses might look like 120-23-61-201-43-146-128-132. To make distinct addresses, IPv4 uses the dot "." for separation, IPv6 uses the colon ":" and the present system uses a hyphen "-". address field represents exactly one logical subnet level, this scheme provides a simple, but at the same time somewhat limited, way of addressing. It means that the complete network can address as many individual hosts as IPv6, while grouping them in at most 8 levels of sub domains, where each sub domain can contain a maximum of 256 hosts or sub domains. The sub domains can be

utilized to indicate a network hierarchy in the topology, including a geographical or similarly based hierarchy.

[0089] In one embodiment, to reduce the amount of bandwidth wasted on this overhead, a mechanism is used for substituting the address strings for small 16 bit values or similar small sized value when transmitting a packet to a neighbor. Also, a translation table is kept at both sides of the connection that is used to put the original address string back in the packet before passing it to the router's packet switching kernel. This optimization works on a per-link basis and may be totally independent from the communication with other neighbors. When it is used on more than one interface, each interface maintains its own translation table.

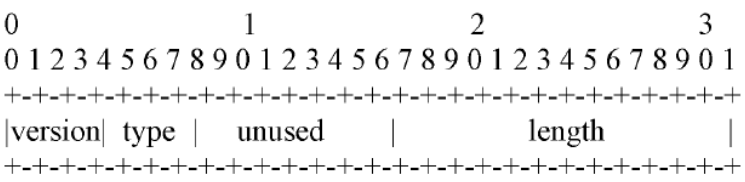
[0090] If a packet is transmitted to a neighbor, all address strings are stripped from the packet header. For each address string a 16 bit or similar value is generated and stored in the translation table. These 16 bits or similar values replace the original addresses and before the packet is transmitted, its protocol version field in the first header of its wire-level representation is changed to a higher version. This version number is reserved for optimized packets that would be unreadable for routers without support for address substitution. To ensure proper translation by the peer, the interface first sends a special packet that contains the new address substitutions. Such a packet is tagged with the higher protocol version number and contains one or more tuples of address strings and their substitute values. In one embodiment, such a packet takes the form:

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 313 of 830



Type 0x1 is used for issuing new substitutions and type 0x2 is used to invalidate an earlier or unknown substitution. The system attempts to ensure that this (or similar) packet is received by the peer prior to the data packet itself, which is something that cannot be guaranteed by every transport layer. TCP or similar guaranteed communication method may be used for communication between neighbor nodes in the overlay multicast network.

[0091] In one embodiment, configuring an interface for address substitution can be done manually, or automatically. In the latter case the interface uses a special handshake packet that is part of the higher protocol version. In one embodiment, the handshake packet takes the form:



field has the higher protocol version. The type for the handshake packet is 0x0 and the length is 4. A peer transmits such a (or similar) packet when the connection is first established. When the packet is echoed back by the peer, the interface knows that the neighbor also supports substitution and goes into substitution mode. After a connection with a neighbor has been lost and is re-

established, the handshake is performed again as the peer could have been replaced by software that does not support substitution.

[0092] In one embodiment, entries in the translation table all have a time-out or similar tracking mechanism to ensure timelines and accuracy of entries. After this period the entry is removed from the table and any packet still using the substituted value will not be processed and cause the receiving node to respond with a special packet containing the substituted values that weren't recognized, allowing the peer to synchronize translation tables by sending a packet containing the substitution tuples. Since the invalidation of timed-out entries is automatic, both peers use the same timeout or similar values in tracking mechanisms. In one embodiment, when the translation table is full, entries are removed according to a last recently used scheme. The address substitution is implemented in a substitution interceptor that is the lowest interceptor or a part of the lowest interceptor or other interceptor reconstructing packets immediately on arrival in an interface module of a router. Substitution introduces additional processing overhead and enabling it is optional for each node.

[0093] In one embodiment, the overlay multicast system may be used solely for routing. The overlay multicast system offers multicast support over existing IP networks. The overlay multicast system goes beyond offering plain multicast functionality and additionally focuses on supporting live multicast streams with limited jitter and delay through the combination of overlay multicast, multicast control and layered multicast.

[0094] In the overlay multicast system the ExBF protocol or similar routing algorithm implicitly contains information on which links in the network are child links. For example, this is a by-product of the long-lived loop avoidance

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 315 of 830

mechanism in ExBF. In ExBF, a node will inform its neighbors about the fastest route it offers to a destination S, except the neighbor that is in the path towards S. Instead, an infinite distance is advertised to this neighbor. This information is used by a router to conclude that each neighbor advertising that node S is unreachable, is actually using it as their parent node towards S and hence will immediately mark those links as child links in the shortest path tree rooted at S.

[0095] Neighbors that do advertise reachable routes to destination S are not flagged as child links and therefore not used when forwarding multipoint packets from S. In another embodiment, this implicit information is available through any distance-vector protocol that uses poisoned reverse or similar techniques. Both methods are used to quickly deliver a copy of any multipoint packet to all nodes in the network. A disadvantage of both mechanisms however is that a copy of every packet is delivered to every node, regardless of whether or not that node is actually interested in multipoint packets from that source.

[0096] In one embodiment, the overlay multicast system is equipped with a sparse forwarding mechanism. The overlay multicast system network is used for a single source multicast for high volume market data, video streams or similar data to be sent to large numbers receivers. In these types of uses it is not necessary that receivers themselves or other nodes be capable of publishing data packets to the multicast group. Instead, for security reasons it is better to know that only the real source can publish to the group. Given these requirements, together with the fact that multi-source protocols are more complex than their single-source alternatives, in one embodiment, the overlay multicast system preferably uses a custom single-source protocol for multicast distribution on the network. In another embodiment, the overlay multicast system supports multiple sources. For sake of convenience an embodiment of a single source multicast network is discussed. One of ordinary

skill in the art would understand that the principles of operation can also be applied to a multi-source application as well.

[0097] **Figure 6** is a diagram of an example embodiment of a multicast distribution or "sink" tree. In the example embodiment, the multicast system starts with a source q that is active, but in this example, a sparse-mode protocol, with no receiver yet subscribed, no distribution tree exists. When the first receiver application p_0 subscribes, its router initiates the reverse computation of one branch of the sink tree rooted at source q. It does so by first marking that p_0 has a local subscriber for q:G by setting the bit in LSp_0 [q,S] (LSp[] is a local array at node p that keeps track of all subscriptions of the local user applications connected to router p) to true and then sending a join (e.g., a <join (q:G)>) packet to the preferred hop towards source q. This neighbor p_1 receives the <join (q:G)> packet and marks the link with p_0 as a child link for multicast group q:G. It then forwards the packet to the next hop p_2 in the shortest path towards q. Eventually node p_n sends the <join (q:G)> packet to q. On receipt, q marks the link on which the packet was received as a child link for its local multicast group G and starts forwarding all packets addressed to G over the link to node p_n. The multipoint packets addressed to G are inserted into the overlay multicast network at node q by the user application that previously bound the local multicast group q:G in publish mode. Nodes p_n, p_{n-1}, ..., p_1 all forward the packets to their neighbors from which a <join (q:G)> packet was received earlier. Node p_0 has no child links for q:G, but $_0$[q:G] set, so it delivers the packets pplication. When another node p_1 becomes interested in q:G while LSp_i[q:G] is not already set, it sends a <join (q:G)> packet to its preferred hop towards q. Let p_{i+n} be in the path from p_i to q that receives the <join (q:G)> packet and suppose that p_{i+n} is also in the path between p_0 and q. In this case, p_{i+n} is already in the established part of the sink

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 317 of 830

tree of *q*, so it does not need to forward the <join (*q*:G)> packet further to *q*. Instead, it only marks the link on which it received the packet as a child link for *q*:G. In general, a node *u* only forwards a <join (S:G)> packet towards S if it is not already subscribed to S:G (hence, LS_u[S:G] is not set and the collection of child links for S:G is empty).

[0098] The example sparse-mode distribution tree generated when processes subscribe to a S:Q group originally equals a part or all of the optimal sink tree rooted at S. However, when the underlying unicast routing algorithm detects changes in the network performance and updates some of the routing table entries, the optimal sink tree changes accordingly and may no longer be matched by the multicast distribution tree, rendering the tree as well as the multicast performance suboptimal. An extreme example of this is when a link that is part of the multicast distribution fails entirely.

[0099] **Figure 7** is a diagram of an example overlay multicast network with a failed link. In the example a node u currently in the distribution tree for *q*:G only has a single route to *q* via *w*. The link between *u* and *w* fails. The node u concludes that it can no longer be part of the distribution tree (as *u* and *q* are no longer connected) and invalidates all child link subscription information. In case LS_u[*q*:G] is set (i.e., the subscription table indicates that *u* is subscribed to group G from node *q*), *u* will rejoin the distribution tree as soon as a new path to *q* is found. In another embodiment, the algorithm sends a notification to the locally subscribed .o indicate that *u* is no longer connected to *q*.

[00100] In the example, it is assumed that all links have weight 1, while the link between *p* and *w* has weight 100. The result is that the link connecting *p* and *w* is not in the optimal sink tree rooted at *q*. Since only *t* and *v* have a local subscriber

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 318 of 830

(LS_t[q:G] and LS_v[q:G] are set), the sparse distribution tree in this example equals the dotted arrows of **Figure 7.**

[00101] In the example, nodes w, u, r and s all have only a single route to q. Node p has a path through both s and w, while s is the preferred hop towards q. When the link between u and w fails, u becomes disconnected from q and invalidates the subscription information for link u-r. Node u informs neighbor r of this fact as part of an immediate distance-vector exchange. This routing update implicitly tells node r that its join for q:G through u is no longer valid and that it should look for an alternative neighbor to rejoin.

[00102] In the example, an explicit leave packet to u is unnecessary. Since r is also left with no alternative path to q and has no locally subscribed applications, it invalidates its q:G subscription and is no longer part of the distribution tree. Neighbors t and s are implicitly informed about this through the ExBF routing update or similar routing update sent by r. Eventually w, u, r and s all leave the distribution tree, while nodes t and v schedule a rejoin when a new path to q is found. Again, they inform their local subscriber applications about this. When p receives the routing update from s, it loses its preferred hop for q and switches to neighbor w. The new cost to q is now 100 plus distance (w, q) and neighbor s is informed. Upon receipt, s discovers the new path to q and informs its neighbors r and v. Node v then sends the <join (q:G)> packet to s. Nodes s, p and w then construct the first part of the new, optimal distribution tree rooted at q. When the packets reach r and t, t sends a join packet to r and r to s, reconnecting all receivers to the data stream.

[00103] In one embodiment, recovering from link failure or similar communication errors is divided in three steps. In the first step the underlying

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 319 of 830

ExBF or similar routing protocol starts a wave that propagates a link failure through the network. Depending on the actual topology, this takes up to N-1 packets, where N represents the number of nodes in the network. The worst case time complexity for recovering from a link failure is 3N when ExBF or similar algorithm is used as the underlying unicast routing algorithm. A best case time complexity would be 2N.

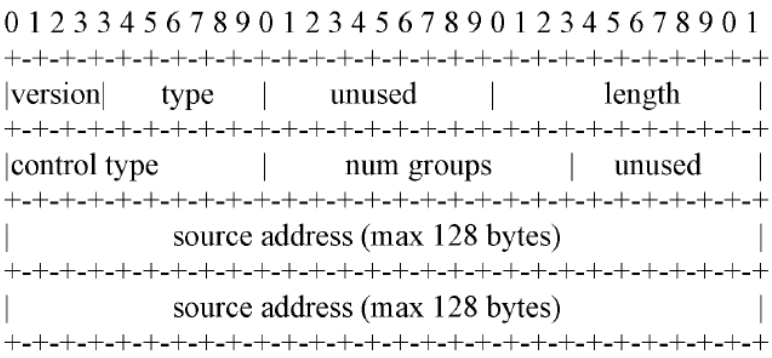
[00104] The example discusses a scenario where the system recovers from a failure of a link that is part of the multicast distribution tree. More often however, links will not fail completely, but rather fluctuate in quality, causing the underlying routing algorithm to reassign preferred neighbors. In this example case it is not a requirement that the distribution tree is changed to reflect these changes, as it is not partitioned. However, since the tree becomes suboptimal, it is changed.

[00105] In one embodiment, the decision whether or not to recompute a distribution tree after a routing table update is made according to the quality of the used path. When a node *p* in the tree detects that the neighbor in the optimal path towards the multicast source has a cost that is only marginally lower than the neighbor that is currently *p*'s parent in the actual distribution tree, the overhead of recomputation and the risk of packet loss outweighs the increased performance of the new tree. Also, since the quality of each link is continuously re-evaluated, the updated sink tree may only be temporal. Two straightforward solutions to this can be used. Either the algorithm sets a threshold on recomputation so that s are only moved to the neighbor with the currently optimal path towards the source if the decrease in cost is at least a ratio *x*, where $x > 1$. A larger value of *x* then postpone tree adjustments until a substantial improvement can be gained, while a smaller *x* makes the tree actively follow the changing characteristics of the underlying network.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 320 of 830

[00106] Another solution is to postpone recomputation of the tree for a period inversely proportional to the cost decrease ration *x*. The latter has the advantage that the optimal distribution tree is always guaranteed to be reached in finite time since the last routing table change.

[00107] In one embodiment, to run the tree building protocol, two new messages are introduced: the join message and the leave message. Both messages use a GroupData control message of the following format:



This message is used to send a list of subscriptions to a neighbor node or to cancel an aggregated list of subscriptions with a neighbor. The action field (9th byte) is used to distinguish between join and leave. The 10th and 11th byte indicate the number of (S, G) groups in the message. The action field is either "join" (0) or "leave" (1). In an alternative embodiment, this is extended to "stale", "dead", etc., to indicate the state of the publishing application.

[00108] Many multicast applications require some level of reliable ion. Examples of this are uploading files to several receivers simultaneously or replicating web server caches. Without guaranteed delivery, a multicast transport service has limited applicability. In one embodiment, the overlay multicast network utilizes standard delivery control based on moderating

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 321 of 830

the sending of data based on the lowest bandwidth available on the route to a destination. However, this delivery system is unsuitable for certain applications.

[00109] One example is financial data distribution, it needs a transport service that can multicast live data without packet loss or corruption and without substantial end-to-end delay when parts of the network are temporarily or permanently congested. While live market data cannot tolerate random packet loss, it can be thinned. Under certain circumstances it is acceptable to omit certain stock quote updates. An example is a desktop application that displays real-time updates for hundreds or thousands of financial instruments. If all quote updates were delivered to this application, this would require a substantial amount of bandwidth and would cause the value of the more volatile instruments to change faster than a human can read. The data may be thinned through a process that involves inspection of the individual stock quotes and encoding them in individual data packets, tagging each with a priority value. For this application, as long as all packets labeled with the highest priority number are received, the partial stream can be considered intact. Additionally, when all updates of the second highest priority are also received, the quality of the partial stream is increased usually in the form of less latency.

[00110] In one embodiment of the overlay multicast system, priority numbers are associated with data packets. The priority numbers represent a logical layer inside a data stream. Data provided by a multicast application may be in the form of a data. This data is subdivided into categories or priorities based on the nature of the data. When packets in a data stream are labeled with priorities in the range 0 to 3, the stream is said to have 4 layers. Also, the convention is to treat 0 as the highest priority and 3 as the lowest. Any other system of identifying priority

levels may be utilized including alpha numeric indicators or similar identifiers. If a stream only contains a single layer, all packets are labeled with priority 0.

[00111] To software router daemons, a priority value of a packet is relative to the stream and becomes relevant when a decision to discard data at a congested router is made. The priority value has no meaning other than as a criterion for discarding packets on congested links. When an outgoing link of a router has insufficient bandwidth to transmit all pending packets, it forwards only those packets with a designated priority or the highest priority. This technique is also applied to any data type, including audio/video data and financial data. Using this system, the packet priority numbers cannot be misused by sources to give their data packets a greater chance of prioritized transmission by giving them the highest priority to gain advantage over other sources. Packet priorities are only compared between packets that are part of the same data stream.

[00112] In one embodiment, knowing that routers will use the packet priority numbers when making forwarding selections on congested parts of the network, a source carefully divides its data packets over different layers or priorities, in a way that a subset of the layers still contain a useful, uncorrupted representation of the data. This system is especially useful when multicasting a live data stream with a high data rate to a large number of receivers, scattered over a heterogeneous wide area network. Receivers that are on congested parts of the network will then receive the highest priority parts of the data only. This eliminates the need to lower ing rate to match the slowest receiver, while still being able to offer live, uncorrupted, thinned data streams to clients suffering from insufficient bandwidth. Example applications of the system include audio and video codecs that divide live multimedia content over layers to enhance the user experience over wide area networks.

[00113] In one embodiment, incoming messages are sorted by unicast or multicast sender address. As described above, a sender address is the combination of source router address and application session. An example of a source address is n1.mxs.office.erik:video.an2 which could be used by a user application broadcasting a video channel. In this case, only the source address is relevant, not the (uni- or multicast) destination address. Each incoming message is added to the queue that holds messages sent by that particular sender. If this is the first message from a sender, a new queue is automatically created to store it.

[00114] **Figure 8A** is a flowchart of one embodiment of a process for managing congestion in the overlay multicast system. The congestion can be managed at each individual router. In one embodiment, the congestion is managed at the interface module level in each router. Each interface module has an inbound pipeline and outbound pipeline discussed above for processing inbound and outbound data. Each pipeline buffers data that is awaiting further processing. However, if either pipeline is unable to keep up with the pace of incoming data that needs to be processed some data must dropped.

[00115] In one embodiment, data is received as a set of data streams at each router (block 851). The data is then buffered in the inbound pipeline buffer (block 853). The same process applies to outbound data that is received from the kernel by the outbound pipeline. This data is stored in the outbound buffer. After the data has been stored a check is made of the inbound or outbound buffer to determine if

)ck 855). In one embodiment, data is stored in the data structures in the buffers that organize the data packets into a set of queues. Each source address, data stream or layer in a data stream has a separate queue. A queue is sorted with highest priority and oldest data packets at the front of the queue. If the buffer is full

then a decision is made to drop a designated amount of data in the form of packets from the buffer to make room for incoming data packets (block 857).

[00116] In one embodiment, a queue is chosen randomly or in a round robin to have data dropped. In another embodiment, the queue with the most data is chosen to have data dropped. In a further embodiment, a weighting factor is calculated to determine which queue is selected to have data dropped. The weighting factor is based on the amount of data in a queue, size of packets in a queue and similar factors. The weighting factor counteracts unfair distribution that is caused by selecting a queue by a round robin, random or similar method of selection. Data streams with large packets are unfairly affected by other methods because a disproportionate amount of data is dropped in comparison with other queues with smaller packets. Queue selection is also influenced by the size and the amount of data that is intended to be dropped. A large queue that can drop close to the amount of data desired is weighted for selection.

[00117] In one embodiment, data is dropped if a total amount of data stored in all queues exceeds a threshold value. This threshold value is set by an administrator or is a set value. These systems enforce the fair allotment of bandwidth between data streams. Also, this system implements the prioritization of logical layers by dropping lower priority level layers when congestion occurs. If the buffer is not full then the pipelines may continue to store data in the buffers (block 851).

ıe embodiment, while the queues ; as packets are received, a background thread constantly dequeues packets from the queues and transmits them over the network.

[00119] In one embodiment, the system that manages in- and output for the queues is divided in two parts. The first part is run by the background thread that

constantly dequeues packets from the queues and transmits them over the network, while the second part is in charge of queuing new packets that are to be transmitted. The latter also implements the logic that defines when and which packet should be discarded (due to a buffer overflow).

[00120] The dequeuing part of the system, when selecting a packet for transmission, only looks at the first packet (most urgent) of each queue. To make sure each stream gets an equal share of bandwidth, it takes the individual packet sizes into account. When all packets (note that only the first packet of each queue is observed) have equal size, the dequeuing thread simply selects a random queue with equal probability and dequeues one of its packets. Since each queue has the same probability of being selected, each queue will deliver an equal amount of packets per time unit. Hence, each source will transmit an equal amount of bytes per second.

[00121] Since individual packets can have any size between 1 and some determined maximum number of bytes, their size must be considered when the transmit thread selects queues. Queues with many large packets should generally have a smaller probability of getting selected than queues with lots of small packets in order to keep the bandwidth division fair.

[00122] In one embodiment, only the first packet of each queue is considered to represent its queue and give it a probability of setting selected that is reversely its size in bytes. In an example whe ıree queues, P, Q and R where the first packet in P is 100 bytes, the first packet from sender Q is 300 bytes, while the packet from R is 500 bytes, to calculate the selection probabilities, first their respective selection weights are defined. For example, the weight of the packet in P is computed by dividing the total size of P, Q and R by P's size: (100 +

300 + 500) / 100 = 9. Q's weight is 3 and R's weight is 9/5. The weights are converted into selection probabilities by dividing them by the sum of all weights. This gives P a probability of 9 / (9 + 3 + 1.8) = 0.65, Q a probability of 0.22 and R of 0.13. When a queue becomes empty after dequeuing, it is removed.

[00123] In one embodiment, to keep the delay introduced by buffering packets under control, the second part of the system enforces a maximum total queue size. The sum of all packets in all queues may never exceed this threshold. When a new packet comes in, it is always accepted at first. If necessary a new queue is created for it, or it is added to its designated, existing queue. After adding a new packet, the total size of all queues is checked. If it is larger than the configured maximum, the algorithm runs a removal round in which it first selects a queue and then tells that queue to remove one packet. If the queues are still too big after removal of one packet, the process of selecting a queue and removing a packet is repeated iteratively until the total queue size is smaller than or equal to the configured maximum.

[00124] In one embodiment, shrinking the queues is always done after a packet was added, never preemptively. The reason for this is to let the new packet immediately participate in the removal selection process, rather than discarding it or making room for it at the expense of the other queues. Contrary to de-queuing packets, where the algorithm tries to select the "most urgent" packet, we now need to select the "least urgent" one. This packet is determined by looking at the size of fat queues (the largest queue should generally be shrunk to keep resource division among the streams fair) and which queue can match the required size most accurately. This comes from the fact that individual packets can differ greatly in size, so removing the "least urgent" packet from queue P could result in freeing 8 kilobytes, while removing the "least urgent" packet from queue Q yields

40 bytes of space for example. Now if the queues in total exceed the maximum size by only a couple of bytes, it seems logical to remove the small packet from Q. This policy comes with a consequence, namely that it implicitly promotes the use of larger packets. After all, streams with lots of very small packets are more likely to accurately match the amount of buffer space that must be freed than stream queues with few very large packets. This property may encourage developers to use larger packets, increasing the efficiency and throughput of the network. In short, when selecting a queue for shrinking, the system favors queues that will expunge the least amount of bytes in order to match the total buffer capacity and are large compared to the others.

[00125] In one embodiment, before the selection can be made, the absolute weight factor for each queue is computed. The weight factor combines the queue's size and ability to accurately match the overcapacity of the buffer by removing packets. The latter is exposed in *v*. It represents the sum of the sizes of the packets that a queue must remove in order to eliminate the buffer's overcapacity. For example, if a queue P has 10 packets of 100 bytes each while the buffer currently has an overcapacity of 170 bytes (suppose the maximum is 10000 bytes, while all queues combined add up to 10170 bytes), P would need to remove at least 2 packets (2 * 100 bytes). In this case, v_P is 200. The formula to compute the absolute weight factor of queue *m* is defined:

$$w_m = \frac{\left(\sum_{i=0}^{n} v_i\right) \cdot s_m}{v_m}$$

In this formula, *n* represents the total number of queues, *v* represents the number of bytes that would be removed if the queue was selected and *s* represents the total size of a particular queue. The formula captures the direct relation to queue size

and the inverse relation to the amount of bytes that would be removed by the queue if it was selected.

[00126] When the absolute weight factors of all queues have been derived, they are converted to weighted selection probabilities. This is done by dividing the individual weight values by the sum of all weights. The selection probability of queue *m* is expressed in P_m.

$$P_m = \frac{w_m}{\sum_{i=0}^{n} w_i}$$

After the selection probabilities have been computed, a roulette-wheel algorithm is run to select a queue and let that queue remove its *v* bytes. If *v* is smaller than the current overcapacity of the buffer in total (note that this implies that the selected queue became empty and was thus removed automatically), another selection round is done until the buffers shrink sufficiently. Note that the queue selection process is implemented as an atomic operation. During selection rounds no new packets may be added to or removed from the queues.

[00127] In one embodiment, the overlay multicast system is a layered multicast in combination with a scalable, selective packet retransmission mechanism to offer a service that can meet the demands of real-time financial data and similar applications. Selective packet retransmissions are crucial so that the overlay [illegible] n is guaranteed that subscribers will [illegible] more layers that are completely intact. When an occasional packet in a high priority layer is missed, it is repaired. However, the receiver may decide not to attempt to recover missing packets from low priority layers that are missing due to network congestion. Trying to repair these layers would require additional bandwidth, resulting in

additional packet loss. Yet, as more end-to-end bandwidth becomes available, the receiver detects this and repairs additional layers, so more intact layers are delivered to the application.

[00128] **Figure 8B** is a flowchart of one embodiment of a process for handling layer repair. In one embodiment, the overlay multicast system supports a receiver-side multicast socket that receives the raw packets from the multicast stream and is in charge of repairing damaged stream layers before forwarding them to the user application (block 801). This method includes a retransmission request packet. When the receiver detects a missing packet in a layer that should be repaired (block 803), it sends a retransmission request for this packet towards the source (block 805).

[00129] In one embodiment, detecting packet loss is done in the conventional way by sequencing each packet with an incrementing number. In another embodiment, each layer has its own sequence number, so all packets in one layer are sequenced independent of the other layers. This way, packet loss is detected in each layer, regardless of the conditions and packet loss in other layers.

[00130] To avoid a cascade of retransmission requests when a packet is dropped close to the source, a tree-based negative acknowledgment is used. This technique provides localized repair without communication with the source, because an intermediate router that has a copy of the requested data packet responds by ı and suppressing the retransmission ın the next router closer in the distribution tree receives the retransmission request it checks its buffer to determine if the requested packet is still stored therein (block 807). If the requested packet is there, it is retransmitted (block 809). If the packet is not in the

buffer then the retransmission request is forwarded to the next router in the distribution tree toward the source (block 811).

[00131] In one embodiment, when layers are dropped due to congestion, the programming API that is exposed to user applications notifies the user by means of an exception or special return value when it is reading packets. These notifications may not be fatal and only serve to inform the user that the quality of the stream has changed.

[00132] In one embodiment, localized repair works by having each router daemon store a window of transmitted packets. Packets are either stored for a fixed period of time, after which they are discarded, or a fixed amount of buffer space is reserved to store the most recently forwarded packets. Since the overlay multicast network by default use reliable connections between router daemons, packets never get lost while they are in transit between nodes. The only place where packets are purposely discarded is in the interface buffer interceptors. Thus, there is no need to buffer packets that were actually transmitted over the reliable connection, as those packets are guaranteed to have reached the neighbor node. In one embodiment, a router daemon copies packets to its retransmission packet store before those packets reach a buffer interceptor that discards packets.

[00133] The router's buffer interceptors copy packets to be stored for retransmission. In one embodiment, localized packet repair in the overlay multicast lemented by adding an interceptor t ptor pipelines of each interface and letting those interceptors simply store a copy of each packet. Additionally, when a retransmission request passes through the router on its way to the multicast source, the interceptor inspects the request packet and if it has the requested packet in its temporal packet store, it does not forward the retransmission

request to the next interceptor. The requested data packet is injected into the network again.

[00134] **Figure 9A** is a diagram of two router daemons connected by a link. There are two interceptor pipelines 903, 905 inside each interface 907. Each interface contains both an outbound 903 and an inbound 905 interceptor pipeline. The latter processes packets received from the network, while the outbound pipeline processes packets just before they leave the router daemon. In one embodiment, the pipelines contain two different interceptors; the CCI (Congestion Control Interceptor) 909 and the PRI (Packet Retransmission Interceptor) 911. In another embodiment, a router daemon contains additional interceptor instances per pipeline.

[00135] The CCI 909 implements bandwidth allocation rules and is responsible for discarding packets. During normal operation the source 913 publishes its multicast packet stream to its router daemon 901 (path 1) that sends the packets to all interfaces that lead to interested receivers according to the multicast routing table. In the present example, the packets are passed to the first outbound interceptor (PRI) of the interface (path 2). The PRI is responsible for storing a copy of each recoverable packet that is transmitted. It stores the copies in the private temporary packet store 915 of its interface (path 3). Then the interceptor passes the packets to the next interceptor (CCI) (path 4) where it is buffered until the connection link with the adjacent router daemon has time to transfer it.

[00136] In one embodiment, if the packets are not dropped in the outbound CCI, they are received by the neighbor router daemon (paths 5 and 6) and sent through that router's inbound interceptor pipeline 921 (path 7). They first enter the PRI 923 (path 7), which stores copies in its inbound buffer 924 (path 8) and then passes

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 332 of 830

them on to the next interceptor (CCI) 925 (path 9) where they are temporarily parked until the router's kernel 927 thread picks them up (path 10) and delivers the packets to the subscribed multicast *tsocket* which forwards them to their next destination 929 (path 11).

[00137] **Figure 9B** is a diagram of two router daemons connected by a link where a retransmission is requested. If a packet is lost in the outbound packet interceptor pipeline of the source router because the connection between the routers was not fast enough to transmit all the packets, the receiving *tsocket* notices the loss and send a retransmission request (path 1). The destination address of this unicast retransmission request packet is that of the multicast group. The network packet switching kernels recognize the retransmission packets and use the multicast group to route the packet towards the source of the group. The retransmission request packet of the example passes the router kernel 927 and reaches the outbound packet interceptor pipeline 931 and goes into the PRI 933 (path 2). This interceptor scans specifically for retransmission requests and tries to answer them locally. To this end it inspects the packet, looks up the multicast group, the stream layer identifier and the layer's sequence number and then checks the inbound packet buffer 935 to see if it contains the packet (path 3). If this is not the case, the packet is passed on to the rest of the interceptors (path 4) and eventually transmitted to the adjacent router (paths 5 and 6) where it is fed to the inbound interceptor pipeline 941 (path 7).

In one embodiment, the PRI 943 checks the interface's outbound packet buffer 945 (path 8) and finds the multicast packet that was dropped by the congested outbound CCI 909 earlier. It is then re-inserted into the outbound packet stream via the outbound PRI 911 (path 9). The retransmission request packet is then dropped and not forwarded any further. Assuming the packet is not dropped

again, it travels the normal way towards the receiver (paths 10, 11, 12, 13). In one embodiment, the receiver's PRI sees the multicast packet for the first time and stores a copy in its outbound packet buffer (path 14). The packet then continues on to destination 929 (paths 15, 16, 17). This mechanism of unicast retransmission packets and interceptors that offer localized repair is flexible in the way that not all router daemons need to support local packet repair. A router that does not support it simply forwards the requests and lets the upstream routers handle them. The network as a whole becomes more efficient when more routers support it.

[00139] In one embodiment, because storing a copy of every multicast data packet requires storage capacity, the overlay multicast system links the packet buffers of the packet retransmission interceptors to a central data structure that will only store unique packets. This is done because in multicast transmissions a router that is a branch point in the multicast distribution tree otherwise ends up storing at least three copies of every data packet. This is because the packet is received on one interface and sent out over at least two other interfaces. If every interface would individually store its packets, the inbound packet buffer of the receiving interface as well as the outbound packet buffers of the forwarding interfaces contain the same packets.

[00140] In one embodiment, the central packet store eliminates duplicate packets by storing each packet only once and by keeping reference tables that point to packets for every interface. When an interface buffer stores a new packet in the :, while that same packet was already stored by another interface, the central packet store merely adds a pointer to that packet instance to the interface's packet reference table.

[00141] In one embodiment, each entry in the reference tables has a time-out attached to it. This is to ensure packets are only temporarily stored. In one embodiment, the same packet could pass different interfaces at different times. The time-outs are not attached to the packets in the central store but rather to the packet references in the interface reference tables. In one embodiment, packets are only expunged from the central store when all references to the packet have timed out. Another optimization that can substantially reduce the amount of required storage space is to only store packets that were explicitly discarded by the Congestion Control Interceptors (CCIs) because of bandwidth constraints. Storing other packets has little value, as those are guaranteed to have arrived at the next router. No retransmissions may be requested for them unless some non-deterministic packet loss occurs as a result of a crashing node.

[00142] In one embodiment, if a packet was lost close to the multicast source and because of this and all receivers simultaneously send a retransmission request, the upstream routers still apply conventional negative acknowledgement "nack" suppression techniques to combine all concurrent retransmission requests from its child links into a single request that is forwarded upstream towards the source. Applying local packet stores at every router reduces the latency of packet recovery.

[00143] In one embodiment, time to live values are adjustable per router, set system wide or similarly configured. A short time-out minimizes storage requirements, but will also lead to more overhead towards the source as more on requests need to be propagated furtner upstream. Large timeouts will be more tolerant to late retransmissions, but require more storage. When a time-out of 60 seconds is used by the routers that means a lost packet can still be recovered after a minute. However, when the data stream contains real-time data such as stock quotes, such a delay in delivery cannot be tolerated. Problems get

worse if the data packets also have to be delivered to the user applications in their original order. In that case the user does not receive any data for up to a minute while the missed packet is recovered and all newer packets are waiting for it. In one embodiment, a ten second timeout setting is utilized.

[00144] If all routers in the network use the same time-out value, there is no use in propagating these requests as the upstream routers are likely to have purged their copies as well. To avoid nack-implosions from slow receivers right after a packet is purged, each router interface remembers which packets it has purged. When a retransmission request is received for such a packet, the interface may respond by sending a special packet to the receiver that indicates that the requested packet had timed-out and cannot be recovered anymore. The retransmission request in this scenario is not propagated further upstream. When a receiver gets this timeout notification, it knows it will not have to wait any longer for the missed packet. It will notify the user application of the fact that data was lost and continue to deliver the next packets. Whether or not an application can tolerate packet loss will depend on the type of content. Audio and video will usually not be severely impacted by the loss of an occasional packet, while real-time stock quotes become dangerous to use when it is not known what quote was lost.

[00145] In one embodiment, where layered multicast is combined with localized packet retransmission, a service is realized that applies packet recovery only to those layers that can be delivered with the current bandwidth capacity. When the network has insufficient resources to deliver all layers or a stream, it is beneficial if the receivers know this and will not attempt to repair all missed packets from all layers. Instead, the receivers may use a mechanism that provides them feedback about the current network capacity and use that to decide for which layers it will recover missed packets. The algorithm monitors the status of the total stream and

from this information derives how many layers of the stream can be reliably delivered to the user without stressing the network. It then marks these layers as being intact, reliable or similarly labeled and sends retransmission requests when an occasional packet is lost from these layers.

[00146] In one embodiment, the algorithm only delivers packets from the reliable layers to the application, but not before they have been put back into their original global order. In order for the algorithm to decide which layers are safely marked as intact, it requires some status information about the reception of the stream as a whole as well as the network conditions. To provide this, every packet contains the number of the most recent packet from all other layers or similar sequence data. Aside from carrying its own sequence number, each packet contains the current sequence number of all the other layers as well. By inspecting these sequence numbers each time a packet arrives, the receiver determines whether it has missed any packets from the layers it marked as reliable. If that is the case, these packets may be recovered. Each time a missed packet is detected, a countdown timer is started for it. If the packet is recovered before the time-out expired, the timer is canceled and removed.

[00147] If however the timer manages to expire, the layer is considered impossible to repair and may be, together with all higher reliable layers (lower in priority), removed from the list of reliable layers. How long the time-out interval should be can be application or content dependent. The interval determines how is tolerated by the user. Setting the interval to a low value means a temporary congestion may corrupt a layer long enough to cause the receivers to drop it. Since packets occasionally get lost when unicast routing tables converge, changing the shape of multicast distribution trees, or when a router crashes that had

pending packets in its interface buffers, appropriate time-out values are adjustable by an administrator according to network conditions to be found experimentally.

[00148] In an overlay multicast network that uses packet prioritization, routers explicitly introduce out of order delivery when a burst of packets queues in the outbound interceptor pipeline of a router interface with a slow connection. The time-out may be large enough to allow for this. Setting the time-out to a long period makes the stream much more resilient to congestion, but also increases the time for the receiver to discover that a layer must be dropped due to bandwidth constraints. Until congestion is finally detected, delivery of the previously received packets that causally depend on missed packets is postponed. Starting timers when packet loss is detected allows for congestion detection. However, it cannot be used to guarantee low end-to-end latency.

[00149] When no packets are lost and all are received according to their global order, the receiver cannot measure the total transmission delay. As such, a receiver cannot distinguish between an idle and a crashed source. In one embodiment, aside from removing layers from the reliable-list, the algorithm is also able to detect when more bandwidth becomes available and new higher layers can be added to this list, so that a higher quality stream can be delivered to the user. This is done by passively or actively monitoring the layers that are not currently in the reliable-list and not under repair. Every time a packet is received from these layers, this fact is stored for a period that is equal to the repair time-out discussed earlier. When a a layer is received with sequence numbers that shows that a packet has been lost from a higher layer that is not in the reliable list, no retransmission request is sent, but the fact that this packet should have been received is recorded and stored for the same repair time-out period.

[00150] If during the time-out period the packet is received after all, possibly because it was delayed by a router, the stored record is marked as received. Because the state of each packet from every layer is recorded, the overlay multicast system builds a packet arrival history that is used to determine whether the reception quality was high enough to add a higher layer to the reliable-list and repair any further packet loss. In the overlay multicast system the reception history or similar data may be used to calculate a moving average that describes the amount of packet loss over the last *x* seconds, where *x* is equal to the repair time-out. With a small moving average history, the receiver will quickly respond to increased network capacity, while a longer history will only add layers when the network's capacity was sufficiently high for a longer period of time. The receiver may keep track of the reception quality of each layer by inspecting the list of sequence number that is attached to each data packet.

[00151] This causality information adds overhead to each packet, linearly related to the number of layers used in the stream. For example, when sequence numbers are 32 bits and the publisher uses all 256 available layers, each packet comes with a kilobyte of causality information, which contributes to at least 12.5% for an 8 Kb packet that is filled to the brim. To ease layer administration in the receiver socket, an overlay multicast stream only supports a static number of layers. When a publisher binds a multicast group address for reliable, layered communication, it may explicitly specify the number of layers that will be used.

In one embodiment, by default, packets are delivered to the user application in their original order. Not only are the packets inside the individual layers restored to their natural order using their sequence numbers, but the total ordering across the layers may also be restored. When the source sends three packets with different priority (each in a different layer), all three are received by

the user application in the exact same order. A prioritized, layered packet is described as $P_{1\ (9,\ 4,\ 6)}$ where 1 represents the packet's priority or layer, 9 the sequence number of the last packet of layer 0 (the highest priority layer) that was sent at the time this packet was published, 4 represents the sequence number of this packet and 6 represents the sequence number of the last packet of layer 2. Also, the packet tells that the stream uses 3 layers in total: layer 0 up to and including layer 2. It is concluded that this packet causally depends on packet 9 from layer 0 and packet 6 from layer 2 and will only be delivered to the user application after those packets have been delivered. In another embodiment, the overlay multicast system does not reorder the packets, but may leave this task to the application using the network.

[00153] **Figure 10** is a diagram that shows one embodiment of a process of selective packet repair and ordering. For purposes of explanation the notation {0 (9, 2, 6); 1(9, 3, 6), ...} to describe a sequence of packets where $P_{0\ (9,\ 2,\ 6)}$ was sent prior to $P_{1\ (9,\ 3,\ 6)}$, is used for sake of convenience. The first segment 1001, shows the sequence of packets originally published by the source. It reads from left to right, so $P_{0\ (9,\ 2,\ 6)}$ was published first, followed by $P_{1\ (9,\ 3,\ 6)}$, etc. In this example the source has published eight packets, divided over three priority levels or layers (0, 1 and 2, where 0 is the lowest layer with the highest priority). The second segment 1003 shows the packet stream as received by one of the subscribers. It shows two lost packets and an incorrect ordering. Before the packets are delivered)lication, they are stored in internal g the repair and reordering process. This state is depicted in segment 1005. Here the packets are in separate buffers, each representing a layer. The packets are ordered inside the buffers. The illustration shows the missing packets $P_{1\ (9,\ 4,\ 7)}$ and $P_{2\ (9,\ 4,\ 8)}$.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 340 of 830

[00154] If the receiver currently only has layers 0 and 1 in its reliable-list, it will attempt to repair the hole in layer 1 by sending a retransmission request. Note that if the given sequence was just a snapshot of a running stream, the receiver would have detected the missing packet $P_{1\ (9,\ 4,\ 7)}$ when $P_{0\ (10,\ 4,\ 8)}$ was received, because this packet says it depends on packet #4 from layer 1. So even before $P_{1\ (10,\ 5,\ 9)}$ from layer 1 was received in our example, the receiver already detected loss in layer 1 and immediately scheduled a repair timeout for the missing packet and sent a retransmission request. In fact, if it is assumed that the receiver had received packet #2 from layer 1 prior to our snapshot of **Figure 10,** then the conclusion would be that packet #4 as well as packet #3 were lost.

[00155] In the example, shortly after $P_{0\ (10,\ 4,\ 8)}$ was received, packet $P_{1\ (9,\ 3,\ 6)}$ is received. The receiver places the delayed packet in the appropriate receiver buffer and cancels the repair time-out it started earlier when it detected that the packet was missing. This is illustrated in segment 1007. Note that the hole in layer 2 may also be detected when $P_{0\ (10,\ 4,\ 8)}$ is received, as that packet claims to be sent after packet #8 of layer 2 was published, so either packet #8 from layer 2 got lost in the network, or was delayed. However, since layer 2 is not in the reliable-list, a retransmission request is not sent. However, as for all packets, a timer is started for packet #8 of layer 2. When the user application is ready to read packets, the algorithm returns only packets from layers that are in the reliable-list. Even though some of the layer 2 packets were received, they are discarded and not delivered. tream of packets that is delivered to ual to the stream originally published by the source, with all layer 2 packets removed.

[00156] In this example, despite the fact that the network has insufficient capacity, having dropped packets from every layer and delivered the packets out of order, the user received a deterministic subset of the stream that is uncorrupted.

Because the overlay multicast system sources are live data streams that cannot slow down or pause and because real-time data are not buffered by the source too long, receivers will only attempt to recover lost packets for a limited period of time. Whether a packet is received, recovered or lost, its buffer slot will be freed after this timeout, resulting in a natural size limit of the receive buffer. How large the buffer can get is related to the average speed of the stream and the length of the repair time-out.

[00157] In one embodiment, the algorithm implementation does not enforce a hard size limit during packet repairs. A more troublesome situation occurs when the user application does not read data from the socket fast enough. When this happens, the amount of pending data builds up in the socket and both storage requirements and the transmission delay increases. In one embodiment, this may be handled by removing a layer from the reliable-list, causing those packets to be discarded from the buffers, while decreasing the amount of data that is delivered to the application. If a clean feedback algorithm that can keep the number of layers in balance with the application's reading speed is not used, the overlay multicast system throws a fatal exception to the user or provides a similar indicator to the user application and closes the socket when the receive buffer reaches a certain maximum size, or when the total time between arrival of packets and their actual delivery reaches a threshold.

[00158] Although restoring the original global packet ordering before delivering the user application is assumed to be appropriate for most types of data, there is content for which each packet invalidates all previous packets. This is the case for stock quotes and similar time sensitive data. For example, when a new stock quote update is received for a financial ticker symbol, it renders the previous updates useless. For most applications that process real-time financial data, only

the most recent information is interesting. When global ordering is restored, the algorithm will postpone the delivery of the most recent data until all prior packets have been received also. For a typical application, such as a market data terminal, that merely displays quote updates to the screen, this postponing adds little value. When the burst of pending updates is finally delivered by the *tsocket*, the application updates the symbol's last value on the screen, thereby leaving only the last and most recent update visible and overwriting all pending updates immediately.

[00159] In one embodiment, because the reordering process adds additional delay to the data delivery, it can be switched off by applications that do not benefit from it. Without reordering, global (causal relations between packets from different layers) and local ordering (order of individual packets inside a single layer) is ignored and packets are delivered to the application immediately after they have been received. Disabling reordering has no impact on the reliability. Lost or delayed packets may still be recovered for all layers the reliable-list, only the recovered packets may be delivered with additional delay. Whether or not this makes them useless is up to the application to decide.

[00160] For example, with market data it is useful to know whether an update is older or newer than the one previously received. This is because a stock quote may be overwritten by a newer one, but not the other way around. In this case, the source could add a logical time stamp or similar sequence indicator to each stock e receiver can decide how to handle the update. In one embodiment, aside from configuring a layered multicast *tsocket* to restore global ordering or no ordering at all, a receiver can configure a *tsocket* to only restore local ordering. Whether or not this is useful will depend on the type of application and the content,

but it offers the advantage that delivery of packets from lower (high priority) layers are not delayed during recovery of packets from higher, lower priority layers.

[00161] In one embodiment, an example market data publishing application tracks market financial updates in a linked list or similar data structure. Every time a new update is received, it is stored in the appropriate slot, overwriting the previous update. This way, the linked list always contains the most recent trade for every symbol. A virtual token traverses the list at a fixed speed. If the stream is to be thinned to one update per second, the token will visit one slot every second. When the token visits a slot, it removes the quote update and forwards it, leaving an empty slot. Empty slots will be skipped by the token without any delay. When one of the symbols has two updates per second in the incoming stream, the second incoming update after the last visit of the token will overwrite the update pending in the slot and the older pending update is dropped. This allows the leaky bucket algorithm to always provide the thinned clients with the most recent, live update of every symbol. While the virtual token visits the slots that have a recent quote update pending to be sent out, a thread receives the incoming market data stream and inserts new trades in the slots. To limit the outgoing transmission rate to one quote per second, the token thread sleeps for one second after it moved and sent an update from a slot. When the token is able to complete one full circle between two incoming updates of every symbol, there will be no data loss. Instead of limiting the outgoing bandwidth to a fixed maximum, the dequeuing thread could also be run as fast as possible, relying on dback such as a blocking write call from the network, to reduce publishing speed. This could be particularly useful when the algorithm is used on the server side of a point-to-point TCP connection to a client application. In one embodiment, the server uses the algorithm to send the data at original speed, until network saturation slows down

the links. A potential problem with this technique is that the original order of the updates is lost. In fact, when every symbol updates more than once while the token circles the list, the token will find every slot filled with a quote update, causing the order of the output stream to match the fixed order of the slots in the linked list. For quote updates from different symbols, this is usually not a problem, as they are not related to each other.

[00162] In another embodiment, an alternative algorithm may be used that provides another way of thinning a stream of data, such as live market data, with varying bandwidth to a fixed maximum rate, without the risk of providing stale data, for example stale market symbols. This algorithm may be used to produce enhancement layers that form a layered data stream, such as a layered market data stream. One example method of doing this is to have several instances of the algorithm running that each receive a copy of the input stream, letting their token run at different speeds and tagging their updates with a layer number. The layers may then be combined into a single stream. Unfortunately, this may introduce redundant information in the stream because the higher layers contain updates that are also available in the lower layers. This technique of encoding an entire stream in different qualities and sending them to clients concurrently to meet their individual bandwidth capacities may be referred to as simulcasting.

[00163] In one embodiment, the thinning algorithm may be modified to have each layer contain only information that is not already present in the lower layers, eliminating redundancy among the layers. To achieve this the leaky bucket algorithm may be extended to a multi-level ring as illustrated in **Figure 11B,** creating a virtual cylinder of vertically stacked, circular lists, where each ring represents a layer of the output stream. Each ring has its own token. In one embodiment, the tokens all circulate their ring at the same speed. In other

embodiments, the tokens in each ring may be set to circulate at different speeds. This data structure also identifies columns. A column is the collection of vertically stacked slots that can all store an update for the same symbol.

[00164] **Figure 11A** is a diagram of an example method of thinning a stream of live market data. The gray slots represent symbols for which a recent quote update was received, while the transparent slots are empty. **Figure 11B** is a diagram illustrating the token ring data structure. Multiple instances of the thinning algorithm and data structure may be conceived as being vertically stacked, each instance represents a stream layer and each symbol has a column that can store more than one update. **Figure 11C** is a diagram of one embodiment of a representation of the stacked thinning algorithm of **Figure 11A** at work. In the example, the gray slots contain a pending quote update, while the transparent slots are empty. The illustration shows how the updates of a volatile symbol are spread out over several higher layers.

[00165] In one embodiment, the thinning algorithm has a single input thread that receives the original stream. Newly received updates may be inserted in the lowest ring. If a new update arrives for a slot in the lowest ring that already contains an update for that symbol, this older update is vertically shifted into the next ring, so that the lower slot can always store the latest update. Once an update is shifted upwards, it cannot return to a lower ring anymore, even if all other slots beneath it were emptied.

[00166] In one example embodiment, symbols that have more than one trade during the time it takes the token to complete a full rotation of the ring will be spread out to two or more higher layers, while symbols that have very infrequent trades will only occupy a single slot in the bottom ring. An example result of this

process is depicted in **Figure 11C**. **Figure 11C** shows the thinning algorithm using three layers to thin a market data stream that contains updates for eight symbols. Four symbols are included in data structure and assigned to the four visible columns in the front. The figure also shows that all four symbols have pending updates in the bottom ring, while the 'MSFT' symbol is so volatile that it shifted updates into all three layers before the tokens could dequeue any of the pending updates.

[00167] In a case where a symbol is so actively traded that it manages to shift an update out of the top ring, that update is irrecoverably lost. If required, loss of updates can be avoided by letting the token in the top ring run without dequeuing delay, letting it empty every slot immediately after an update was queued. The result is that the transmission rate of the upper layer will equal the transmission rate of the original stream, minus the combined rate of all lower layers. Because the transmission rate of the lower layers may have a fixed maximum, the upper layer's rate may fluctuate and follow the variations of the original stream.

[00168] In one embodiment, if the tokens all rotate at the same speed, this does not imply that they visit the same symbol slots at the same time. Instead, they will usually be at different positions in their ring. This is a result of empty slots that are skipped without waiting.

[00169] In one embodiment, the layered thinning algorithm has out of order
d may cause additional delay. Every ain an update for
the same symbol at a certain time and because the tokens can have different column locations inside their ring, it is undefined which ring will dequeue and transmit its update first, which will be second and so on. This means that not only

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 347 of 830

the order of symbols is shuffled, but even the updates for a single symbol are no longer guaranteed to be transmitted chronologically.

[00170] While new updates for a symbol are queued, shifted and sent out, in some cases an update can survive in the rings substantially longer than its younger successors. This can occur when an update that is just about to be dequeued by the token, gets shifted to the next layer by a more recent update that entered through the lower ring. In this case it is likely that the more recent update is dequeued sooner than the older update that now has to wait for the token of its new ring to reach its slot. To address this issue, the system provides a way of putting the outgoing, layered updates back in the original, chronological order. This is done by labeling incoming updates with sequential numbers and having the tokens pass their dequeued, labeled updates to a transmit window that uses the sequence numbers to restore the chronological order. When updates are shifted out of the upper ring and therefore considered lost, the transmit window is notified not to wait for their sequence numbers.

[00171] When applying order reconstruction across all rings, the transmit window must have the same capacity as all rings combined, $size_{tw} = N \cdot j$, where $size_{tw}$ is the maximum capacity of the transmit window, N the number of symbols in the original stream (and therefore the number of slots per ring) and j the number of layers used. The potential size of the transmit window, together with its bursty behavior when a delayed quote update releases a substantial amount of pending updates in the window, may justify a mechanism that applies order reconstruction to achieve chronologic transmission per symbol only, similar to the output of the base layer encoder.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 348 of 830

[00172] Like the algorithm depicted in **Figure 11A**, the layered system introduces delay. The model is essentially a collection of instances of the first algorithm, thus, its worst case delay is the sum of each ring's worst case delay:

$$t_{\max} = \sum_{i=0}^{j} \frac{N}{f_{update}(i)}$$

[00173] where t_{max} is the maximum delay in seconds that the algorithm could introduce for a single quote update, N the number of symbols in the original stream, j the number of used layers and $f_{update}(i)$ the update frequency of the token in layer i.

[00174] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will however, be evident that various modification and changes can be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 349 of 830

MARKED VERSION

LAYERED MULTICAST AND FAIR BANDWIDTH ALLOCATION AND PACKET PRIORITIZATION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a divisional of pending U.S. Patent Application No. 13/619,969, filed September 14, 2012, which is a continuation of U.S. Patent Application No. 12/486,656, filed June 17, 2009, now U.S. Patent No. 8,514,718, which is a divisional of U.S. Patent Application No. 11/342,167, filed January 26, 2006, now U.S. Patent No. 7,733,868, which claims priority to U.S. Provisional Application No. 60/647,601, filed January 26, 2005, all of which are incorporated by reference as if set forth in full herein.

FIELD OF THE INVENTION

[0002] The invention relates to network management. Specifically, the invention relates to the management of data packets to support multicasting.

BACKGROUND

[0003] Despite the versatility in digital communication, interoperability and internationally accepted communication protocols of the Internet, its fundamental design has not changed much since its conception and does not excel in everything. Watching live TV for example is something which is not typically done over the Internet, even though television has been around almost twice as long as the Internet Protocol and represents a huge market. The reasons for this are based on the design of the Internet and Internet Protocol (IP).

nternet is a packet-switching network where data is exchanged in small units or packets that are independently transported over the network and concatenated again at the receiver into its original form. A strength of packet-switching is that it allows for very flexible use of the physical network wires. When two communicating parties have no data to exchange

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 350 of 830

for a certain period of time, no packets are sent and the wires can carry packets from other parties. On the Internet, bandwidth is not reserved[[;]] but available to and shared by everyone. The consequence is that it cannot guarantee a minimum amount of end-to-end bandwidth, making live video streams often appear jerky because frames are skipped due to congestion that delays or prevents delivery.

[0005] Even though with help from specialized protocols such as Distance Vector Multicase Routing Protocol (DVMRP) or Protocol Independent Multicast (PIM), the Internet Protocol allows for data packets to be multicast to a large number of receivers simultaneously, using this feature to successfully realize a live video broadcast is a challenge. A video stream is transmitted at a fixed high rate and not all parts of the network are likely to have sufficient bandwidth available to forward the stream.

[0006] When a bandwidth bottleneck is reached, the router discards the packets that cannot immediately be forwarded. This causes two problems. The data stream that is eventually received by one or more receivers further down the network is corrupt and the congestion also has a negative impact on communication sessions of other nodes that communicate through the bottleneck router. The only way to avoid this problem using the Internet Protocol and standard multicast is to find a transmission rate that is supported by all parts of the network. However, since the network is available to anyone, this rate will continuously change. A transmission rate is selected and the packet loss is accepted. However, when packets are dropped randomly by

ıters the data stream will suffer packet loss. If additional packets are sent through the bottleneck router, there is a larger chance that the router will choose one of these packets when ready to send another packet, implicitly rewarding heavy streams during congestion. This encourages sending redundant data thereby exacerbating the problem.

[0007] A more fundamental problem of flow control using the Internet Protocol is that slowing down the data may not be an option for certain types of live data streams. However, packet loss is unavoidable using the Internet Protocol and while data types such as audio and video data can usually withstand some packet loss without becoming too corrupted to play, this does not apply to all types of live data. Real-time financial data, for example, will become useless and even dangerous to use if random packets of trades are lost.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 352 of 830

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Embodiments of the invention are illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to "an" or "one" embodiment in this discussion are not necessarily to the same embodiment, and such references mean at least one.

[0009] **Figure 1** is a diagram of one embodiment of an overlay multicast network.

[0010] **Figure 2** is a diagram of one embodiment of the basic components of a router daemon in the overlay multicast system.

[0011] **Figure 3A** is diagram of one example embodiment of a layered multicast system network divided into logical clusters.

[0012] **Figure 3B** is a diagram of a routing table for the network of **Figure 3A**.

[0013] **Figure 4** is a diagram of an example network with hierarchical structure.

[0014] **Figure 5A** is a diagram demonstrating summarization of an example network with an S2 domain.

[0015] **Figure 5B** is a diagram demonstrating the stretch factor for the example network of **Figure 5A** and shows the inner network of the S2 domain.

[0016] **Figure 6** is a diagram of an example embodiment of a multicast distribution or "sink" tree.

[0017] **Figure 7** is a diagram of an overlay multicast network with a failed link.

[0018] **Figure 8A** is a flowchart of one embodiment of a process or managing congestion in the overlay multicast system.

[0019] **Figure 8B** is a flowchart of one embodiment of a process for handling layer repair.

[0020] **Figure 9A** is a diagram of two router daemons connected by a link.

[0021] **Figure 9B** is a diagram of two router daemons connected by a link where a retransmission is requested.

[0022] **Figure 10** is a diagram that shows the process of selective packet repair and ordering.

[0023] **Figure 11A** is a diagram showing thinning of a stream of data.

[0024] **Figure 11B** is a diagram of a multi-level token ring structure.

[0025] **Figure 11C** is a diagram of a multi-level token ring structure showing thinning of data.

DETAILED DESCRIPTION

[0026] To provide multicast services a network needs to support one-to-many communication that can send data packets from a data source to more than one receiver, ideally without putting extra stress on the network or source when the number of receivers increases. Multicast routing can be offered by different methods. One method is to let receivers tell the network, but not necessarily the source, which data streams they want to receive and let the network compute data distribution paths that deliver just the right packets to each receiver. Multicasting can also be done by letting the source encode the list of receivers in each data packet, thereby freeing the network from the potentially computationally intensive task of maintaining multicast distribution paths. However, this method does not scale to handle a large number of receivers. A third method relies on logic at the receivers by letting the network apply a broadcast mechanism whereby each packet is delivered to every connected node and letting the receivers filter out only those packets that are interesting. This method may also generate a heavy load on a larger network but it is simple.

[0027] In one embodiment, a multicast network is constructed as an overlay network. In one embodiment, the overlay network includes a number of software implemented routers connected by normal IP or TCP connections. A mesh is created in which every software router is connected to one or more other software routers by means of virtual connections that appear to be direct connections to the other software routers, but are likely implemented by a number of intervening "real" routers situated between the software routers. Two routers that are connected this way, are adjacent in the perspective of the overlay, but in reality are many physical hops away from one another. Also, a software router that has three software router neighbors has three

independent virtual links. However, it is possible that this software router only has a single physical network connection that is shared by the three virtual links.

[0028] To the underlying network, the overlay network is nothing more than a collection of applications that send data between static pairs. Beyond each router pair (identifiable by their TCP connection) there is no relation between the individual daemons. As such, the overlay network can easily work through firewalls, NAT (IP masquerading), proxies and VPN's. Firewalls cannot control which software router POPs can talk to each other. The system shares some principals with a HTTP proxy server tunneling traffic to and from web browsers. In an example HTTP proxy server system, an intranet has two web browser machines. They both browse the Internet using the proxy server also on the local intranet. Although the individual web browser machines can surf the net individually, they never make a direct connection with a remote webserver, but only with the local proxy server that acts like a relay point. As such, the firewall that sits between the proxy server and the remote webserver can only choose to either allow the proxy server to talk to the webserver or deny it. It is generally unable to enforce unique policies for individual browser machines. It generally cannot tell on behalf of which client the proxy server is fetching a webpage. Returning to the overlay network, because each software router is a relay point that tunnels data traffic for different senders and receivers similar to a HTTP proxy server, firewalls have no fine-grained control over communication over the overlay network. As soon as a firewall allows for only a single TCP connection between an internal and

[illegible], all software routers connected to the internal one can talk to all POPs connected to the external one, and vice versa, without restriction.

[0029] In one embodiment, the overlay multicast routing system also manages flow control and timely delivery. Non interactive live data streams do not actively anticipate network

congestion. To manage congestion the network manages the available bandwidth to allow for fair or equal division among the streams. Without management, high volume streams are assigned a larger capacity percentage on overloaded links resulting in little benefit in keeping the bandwidth requirements of a stream low, as the packet loss percentage is determined by how much the network is overloaded by all streams combined and not by the requirements of the individual streams. An alternative to letting the network handle the flow control and congestion is to put the responsibility at the source and receivers. However, letting data streams anticipate network conditions requires a form of feedback information from the network or the receivers. In this case it is beneficial that the amount of feedback does not grow linearly with the size of the audience, as that would reduce the scalability of the multicast. Even when a scalable form of feedback information can be realized and the data stream adapts its transmission rate according to the network conditions, the problem remains that live streams lose their value when they are slowed down and delivered late.

[0030] In one embodiment, an overlay multicast system also implements or manages delivery of packets including an option for guaranteed delivery. It would be ideal if every receiver would receive the data stream without loss or corruption. However, when the content is 'live' and cannot be slowed down, but the network has insufficient capacity, packet loss is difficult to avoid. In fact, even when there is sufficient bandwidth at all times, store-and-forward packet-switched networks are not able to guarantee the delivery of all packets. For example, crashes, all packets in its buffers may be irrecoverably lost. If a network uses dynamic routing, packets are also dropped when routing policies change or packets are trapped in an occasional, temporary routing loop. In cases where packets are 'accidentally' lost, an end-to-end mechanism of retransmissions can be applied that can compensate for the loss. However,

since this requires a form of feedback information, it is beneficial for reasons of scalability that the overhead involved with retransmissions is not linearly related to the size of the audience.

[0031] End-to-end retransmission feedback may be avoided in at least two ways. First, it is possible to let the network components keep a copy of the most recently forwarded packets and let them participate in retransmissions by intercepting the retransmission requests and servicing them locally. This approach often utilizes greater storage and increased processing power requirements at the network components.

[0032] The second alternative to end-to-end retransmission requests is that of encoding redundant information in the data packets. If enough redundancy is encoded, a lost packet's content may be entirely recovered from the extra information in the other packets. The downside of this system is that it comes with a constant level of bandwidth overhead that is related to the level of packet loss tolerance, regardless of whether packets are actually lost. Each of these approaches to packet loss fail in the case of a live data stream that produces more bytes than the network can forward. When local or end-to-end retransmission requests are used, the problem may even be increased as the retransmission requests use extra bandwidth, causing more data packets to be lost.

[0033] Embodiments of the overlay multicast system are designed to offer an end-to-end solution to efficiently multicast live data including stock market data to any machine connected to the Internet. The system is capable of giving certain hard g vhat is delivered to receivers. If packets must be dropped due to congestion or other irrecoverable problems, it is done in a fully deterministic way that does not corrupt the data. Where a receiver of data [[\ ]] such as a viewer of a film may accept the random loss of one or two video frames, this type of data loss may wreak havoc in financial data when the missed packet contains an important trade.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 358 of 830

The system supports deterministically delivery designated parts of a data stream when the network lacks sufficient capacity. The system utilizes a layered multicast with data streams subdivided into individual streams or layers. This allows receivers to subscribe to only those layers that the network can handle, so that random packet loss can largely be avoided.

[0034] In another embodiment, an enhanced form of layered multicast is used that guarantees complete delivery for certain layers to avoid random loss altogether, making it suitable for certain types of critical data such as market data. The system is characterized as controlling two primary activities. The first activity is running and managing a robust and scalable overlay network that uses its own routing algorithms and supports multicast, while the second activity is managing flow control and congestion when live streams overload the network and ensuring layered multicast can be offered with guarantees.

[0035] **Figure 1** is a diagram of one example embodiment of an overlay multicast system. In one embodiment, the system includes an overlay network 100 that includes a number of software router daemons 109, 115, that are interconnected by normal TCP connections 119 or similar reliable communication connections. The overlay multicast system forms an intricate web of routers and virtual links similar to the Internet, but operating on the application layer. This system operating at the application layer overlays the physical network and other layers of the Internet providing its own system of routing, controlling the lower levels of the network. Any number <u>of</u> [[or]] routers and client applications may be a part of the system. In the example, two 109, 115 are in communication with local client applications 101, 111.

[0036] In one embodiment, each of the routers have a unique, string-based addresses 117. In another embodiment, the routers may have other types of unique addresses such as numerical addresses. Each of the router daemons 109, 115 executes a routing algorithm that computes the

shortest paths that allows each router to send data packets to any other router in the overlay multicast system as well as any machine in communication with these routers.

[0037] In one embodiment, the system includes a runtime client library 103, 113, application programming interface (API) or similar system for sending and receiving packets that can be utilized by user applications to communicate over the system. This library connects an application to a nearby system router daemon 109, 115 through a TCP connection 107, 121, native inter-process communication or similar communication method. The native inter-process communication method may be used if both the router and the client application run on the same physical machine. When connected to a router daemon, the client application 101, 111 can send and receive data packets from the network under the router's unique address. Router daemons have a unique address, while user applications connected to a router daemon are identified through logical ports.

[0038] In one embodiment, the topology of the overlay network may be configured at router daemon startup. The topology may be relatively static after configuration. In one embodiment, the relationship of router daemons to applications may be one to many, with a single router daemon serving multiple applications. In another embodiment, the relationship may be many to many.

[0039] In one embodiment, the [[.]] data packets in the overlay network may be addressed to a single communication endpoint or to abstract multicast addresses. Single communication endpoints are network addresses used by user applications. Access to them is exclusive such that only one user application can use them for sending and receiving packets at a time.

[0040] In one embodiment, a unique network address in the overlay network that is used by an application is the combination of a logical node address assigned to the overlay router and the

port name chosen by the application. If an application that is connected to an [[a]] overlay router with the logical node address "nl.msx.1" wants to use a port named "myport" for receiving unicast data packets, the fully qualified network address becomes "nl.mxs.1:[[1 :]]myport." Other applications connected to the overlay network that want to send packets to this application can use this as a destination address. Ports may be referred to as *tports.* Overlay network addresses must be bound, prior to sending or receiving packets. When a data packet is sent from an endpoint address it will contain this information as its source, allowing other routers or the receiving application to send a response.

[0041] In one embodiment, the overlay network may support packets being sent to abstract multipoint destinations, or multicast addresses. An overlay network multicast address is a destination that each machine on the network may subscribe to. The overlay network software routers ensure that a copy of each data packet published to this multicast address is delivered to every user application that subscribed to it. In one embodiment, multipoint destinations are single sourced. Only one application can publish data packets to the multicast address, making the overlay network suitable for one to many, but not many to many communications. Because a tport session of a multicast address can be freely chosen, each multicast address explicitly contains the location of the source on the overlay network. This makes multicast communication less flexible because publishing is not anonymous, but it greatly simplifies subscription management. In another embodiment, multiple applications may publish to a multicast address

many communication is supported.

[0042] In one embodiment, both unicast and multicast addresses are syntactically equal. For example, the address "nl.mxs .1:timeservice" could be a unicast address used and bound by an application that provides the local date and time in response to any data packet it receives.

SCALE VIDEO CODING, LLC v. CISCO SYSTEMS, INC. **IPR2024-01371** **Ex. 1002, p. 361 of 830**

However, it could also be a multicast address that anyone can subscribe to. For example, subscription to the address may provide periodic date and time broadcasts from the user application that bound this address as a multicast address. In one embodiment, each packet may contain a flag or similar indicator that indicates whether its destination address should be interpreted as a unicast or multicast address.

[0043] In one embodiment, the overlay network provides a set of higher layered protocols as services over the packet oriented base layer. These services may be referred to as protocol endpoints. Any number of protocol endpoints may be defined and supported. In one embodiment, five protocol endpoints may be supported. A unreliable unicast protocol (UUP), a reliable unicast protocol (RUP), unreliable multicast protocol (UMP), ordered layered multicast protocol (OLMP) and reliable ordered layered multicast protocol[[.]] (ROLMP) may be supported. The UUP offers a best effort unicast datagram service to applications. The RUP offers reliable unicast communication between peers. The UMP offers best effort multicast datagram services to applications on the overlay network. The OLMP offers multicast communication with receiver driven rate control. Complete delivery is not guaranteed, but the packets that are received are guaranteed to be in their original order. The ROLMP offers reliable multicast communication with receiver driven rate control. Stream layering allows each subscriber to receive the data stream in the highest possible quality, while the source never has to slow down.

[0044] **Figure 2** is a diagram of one embodiment of the basic components of a router overlay multicast system. Each overlay network router 109 includes a packet switching core or kernel 201. In one embodiment, packets that are received, either from a connection to a neighbor router or from a connected application, pass through the kernel 201. The kernel 201 may not handle some specialized control packets. The task of the kernel 201 is to

inspect the destination of each packet and use a set of routing tables to determine how to forward the packet.

[0045] In one embodiment, the kernel 201 forwards packets to neighbors through a set of interface modules 203, 205. In one embodiment, the kernel 201 may execute on its own thread and be event driven. The kernel 201 remains idle until it is notified by an interface module 203, 205 of an incoming packet or by an application 227 that is sending a packet. The kernel thread is woken up, reads the packet from the interface module or application and processes it. If the kernel decides that the packet must be sent out through an interface, it passes it to that interface and waits for the next notification.

[0046] In one embodiment, each router daemon in the network is connected to one or more neighbors. This is done by establishing connections between the routers. In one embodiment, the connections may be long lived TCP connections or similar communication connections. In one embodiment, a router daemon 109 runs one interface module 203, 205 instance for each configured neighbor or configured communication channel. In another embodiment, the router 109 may run multiple interface modules for a configured neighbor or communication channel or a single interface module for multiple configured neighbors or communication channels. The responsibility of an interface module 203, 205 is to establish a connection, e.g., a TCP connection, and to[[·]] pass packets [[·]]from the kernel 201 to the connection and vice versa. Packets of the first kind are referred to as outbound packets, while the latter are inbound packets.

[0047] ... one embodiment, the kernel 201 maintains a unicast routing table that is used for packet switching. To make it possible for the network to find shortest paths as well as adjusting these paths when the underlying physical network's characteristics change, each interface module may measure the quality of its virtual connection. These measurements are passed on to the

kernel 201 when the link quality is found to have changed. Inside the kernel 201, the measurements may be fed to a routing algorithm to determine if the changed link quality alters any of the entries in the routing table. If the routing table is changed, it is advertised to the neighbors by encoding the entire table or a portion of the table in a data packet and passing it to each interface. In one embodiment, if this type of packet is received from a neighbor and propagated to the kernel 201 through the receiving interface, the kernel inspects the message and analyzes it through the routing algorithm. If the new information leads to changes in routing entries, the router sends its own related routing entries to all its neighbors.

[0048] In one embodiment, if a neighbor router crashes, the interface detects this through an error on the virtual link and passes an indicator to the routing algorithm. The routing algorithm then changes the cost indication associated with the link to an infinite value or similar value indicating that the link should not be utilized. In one embodiment, the kernel 201 does not distinguish between a crashed neighbor and an infinitely slow link. The kernel 201 only detects that the link is not to be utilized when reading the entry in the routing table.

[0049] In one example, when a source application connected to router s [[S]] wants to publish a live data stream to multicast group s:mygroup, where applications on node p and *q* want to receive it, the source may first bind the group in publish-mode . Binding a multicast group in publish-mode means that the group is owned by the binding process. Only the owner of the group is able to publish to it. The receiving or "sink" applications connected to routers p and

oup s:mygroup in subscribe-mode. Binding a multicast group in subscribe-mode results in the router node being connected to a virtual multicast distribution tree. The subscribers receive a copy of any packet published by the source.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 364 of 830

[0050] In one embodiment, multicast groups do not necessarily need to be bound in publish-mode first. Any application may subscribe to any multicast group at any time. If there is no source, there will be no [[.]]data to receive. Binding a multicast group either in publish-mode or subscribe-mode are distinct operations. When the source also wants to receive a copy of its own stream, it binds its group in subscribe-mode and uses the resulting tsocket to read the data back. Data packets carry both the address of their source application, as well as the network address of their destination.

[0051] For unicast packets, the router uses its unicast routing table to find the preferred next hop, while multicast packets are routed according to a subscription list in a multicast subscription table. In one embodiment, a unicast data packet contains the node address of the source router, the tport of the source application, the address of the destination router and the tport of the destination application. A multicast data packet contains the multicast group it was published to, represented by the node address of the source router and the group identifier that was bound by the source application. It does not contain any other addressing information.

[0052] In one embodiment, the overlay multicast system determines unicast routing tables and multicast subscription tables for each router. The system utilizes any type of routing algorithms or protocols to determine routing tables. Protocols that are utilized include distance-vector protocols and link-state protocols. Link-state protocols take the relatively simple approach of propagating the state of the entire network as a list of all known links in the network with their

-vector protocols are based on the Bellman-Ford protocol. They work by letting every router periodically advertise its own routing table to its neighbors. In one embodiment, the Extended Bellman-Ford protocol, hereafter referred to as ExBF, is used as the basis of the overlay multicast network. For sake of convenience, the embodiments of overlay multicast

system are described that utilize the ExBF, however, those of ordinary skill in the art would understand that other routing algorithms may also be utilized.

[0053] In the ExBF protocol, a slight increase in the amount of information is kept for every destination. Instead of storing just the distance to each destination for every neighbor, ExBF also stores the address of the pre-final node of each path. Hence, instead of storing the collection of distances *{Di(j)}* where j represents a destination, D the distance between this node and *j*, while *i* ranges over the node's neighbors, each router also stores {Pi(j)} where P is the pre-final, or so-called 'head-of-path' node in the shortest path to destination *j*. Now because the router knows the pre-final node of every destination, it can backtrack the full path to any destination by recursively looking at the head-of-path of a destination and treating it as a new destination.

[0054] In one embodiment, the interface modules 203, 205 implement a simple algorithm to establish the communication connection to the neighbor machine or device. The interface module 203, 205 attempts to connect to the configured [[·]]neighbor router by actively trying to connect to its network address, e.g., its TCP/IP network address. If the connection is aborted with a connection refused or other error, it is assumed that the neighbor is not yet running and the interface module 203, 205 starts to listen on its configured IP port so that the neighbor can connect to it when it is started up. The interface module 203, 205 waits for an incoming connection request for a brief period of time. After this time period expires, the interface module 203, 205 returns to actively connecting to its neighbor. To avoid a situation where both

tinue to switch between the listen and active states at the same time, the duration of the listening state is influenced by a random factor. An advantage of allowing each neighbor to switch [[·]]between the active connection and listening states when establishing a connection is that it allows the connection of routers even if one of the routers is on a masqueraded IP network.

A router on a masqueraded IP network is unable to accept incoming connections such as TCP connections.

[0055] In one embodiment, interface modules 203, 205 can be configured at router deployment through configuration files. This will make the router daemon automatically create the specified interfaces at startup. If the configured neighbor routers are online, all long-lived connections will automatically be established. It is also possible to add new neighbor connections and interface module instances dynamically at runtime. This way the overlay network topology can be changed flexibly and new routers can be added to the network. If an interface module 203, 205 manages to establish a connection with a neighbor router, the interface modules of the routers exchange overlay network node address information to inform each other of their presence. When the interface module 203, 205 receives the node address of its neighbor, it passes this information, together with an announcement that a connection has been made to the kernel 201. This information allows the kernel routing algorithms to build dynamic routing tables.

[0056] In one embodiment, a role of the interface modules 203, 205 is to establish the connection with a configured neighbor router and to send data packets from the router kernel 201 to the neighbor connection and vice versa. The interface module 203, 205 incorporates a [[.]]framework that allows custom software plug-ins to influence the stream of packets that flows between the network and the kernel 201. This mechanism is referred to [[.]]as the interceptor 257. In one embodiment, each software [[.]]plug-in component that needs to control the packet flow is a class that implements a simple programming interface. In another embodiment, the software plug-ins may have any implementation structure including objected oriented structures. This interface allows interceptor instances to be chained together, forming a

packet processing pipeline. The contract of an interceptor is that it receives a packet, applies it operations and then passes the modified packet to the next interceptor in the chain.

[0057] In one embodiment, an interceptor pipeline 251, 257 sits between the router switching core and the network connection. When the router kernel 201 delivers a packet to the interface module 203, 205 for transmission to the neighbor router, the interface module runs the packet through the interceptor pipeline, giving the interceptors the chance to modify the packet. Each packet that comes out of the pipeline is transmitted to the neighbor router.

[0058] In one embodiment, each interface module 203, 205 has two interceptor pipelines 251, 257. The first interceptor pipeline 257 is used to process outbound packets. The second interceptor pipeline 251 is used for inbound packets. These pipelines are independent of one another, the ordering of the interceptors and the number of processing steps can be different for inbound and outbound packets. Each pipeline can be configured uniquely. Interceptor pipelines may have any number of interceptors, provided they do not add too much latency.

[0059] In one embodiment, an example of an interceptor is one that filters packets from a specific overlay network router. When this interceptor is implemented as a manageable component that can be configured dynamically at runtime, it can be used to implement basic firewall functionality on the overlay network. If it receives a packet that matches its rejection pattern, it discards the packet by not passing it to the next interceptor in the pipeline. Another type of interceptor that may be used is a traffic monitor that c[illegible] each packet that passes by and uses this to log traffic activity and compute bandwidth statistics. This plug-in mechanism allows an overlay network router to be extended with additional functionality without modifications to the underlying software.

[0060] In one embodiment, the interceptor pipelines 251, 257 act as a packet buffer between the router kernel 201 and the network. The interface modules 203, 205 temporarily buffer inbound and outbound packets. This ensures that the kernel 201 is not blocked when sending a packet to an interface. In one embodiment, the interface modules have a separate thread or set of threads that continuously dequeue and serialize packets from the buffer and write them to the communication connection or enqueue into [[a]] buffer received packets.

[0061] In one embodiment, the interceptor pipeline provides a temporary packet buffer and offers inter-thread communication between the kernel thread and the interface thread. Interceptors may be divided into two categories: interceptors that block and interceptors that return immediately. The first category is referred to as blocking or synchronous interceptors. In one embodiment, to avoid situations where a router kernel 201 is blocked for an arbitrarily long time, an interceptor pipeline may contain one non-blocking interceptor. A non-blocking interceptor guarantees immediate return of control to a caller by storing the packet in an internal buffer. The packets in the buffer may be discarded if the buffer exceeds a certain threshold size.

[0062] In one embodiment, a maximum size limit is placed on the packet buffers to prevent them from exhausting the router's memory. Storing packets before processing them means that their delivery will be delayed. The larger the buffer gets, the longer the packets are delayed. Because of this, the interceptor drops packets when the buffer reaches its maximum size. In one embodiment, the system uses reliable TCP or similar connections for transmitting packets

rs in which packets are only dropped inside the buffer interceptors of the interface modules.

[0063] In one embodiment, the interceptor plays a role in making packet loss deterministic. Packets are not explicitly dropped in any other part of the layered multicast system network,

SCALE VIDEO CODING, LLC v. CISCO SYSTEMS, INC. **IPR2024-01371** **Ex. 1002, p. 369 of 830**

except the buffer interceptor in the interface pipelines. However, that may not guarantee that a packet that successfully makes it through all buffer interceptors of the network's routers is delivered at its destination. Aside from controllable packet loss, the network may also occasionally lose packets in a [[ ]]non-deterministic way, for example when a router crashes with pending packets in its buffers, or when a connection between adjacent routers is unexpectedly closed during transmission of a packet.

[0064] In one example, an inbound interceptor pipeline 251 may be structured such that traffic throughput monitor interceptors 253 are positioned after a buffer interceptor 255 and a firewall interceptor 265 may be positioned before for the buffer interceptor 255. In an example outbound interceptor pipeline 257, a throughput limiting interceptor 259 may be positioned before a buffer interceptor 261 and a traffic monitor interceptor 267 may be positioned after the buffer interceptor 261.

[0065] In one embodiment, the overlay network is accessible to applications through the use of the client-side programming library 221. This library connects to a router daemon 109 at application startup and communicates with it using remote procedure protocol or similar communication protocols. The communication between the library 221 and the router daemon 109 is driven by the application 227. The application 227 invokes functions in the router 109 through the library 221. In one embodiment, when the kernel 201 receives packets addressed to a tport that is bound by the application 227, it stores them until the client actively picks them up.

n 227 through the library 221 continuously polls the router for packets. To minimize the overhead of the polling mechanism, the router poll function does not return until there is at least one packet delivered by the kernel 201. If more than one packet is waiting, the poll function returns all waiting packets at the time it is invoked. In another embodiment, the

router may send an indication such as an invoking a marshaled stub from the client, event notification or similar indicator to the application 227 through the library 221 to indicate the reception of a data packet for the application 227.

[0066] In one embodiment, a user application 227 uses a client library 221 through instances of the overlay multicast system communication sockets 225. If a user wants to be able to receive packets, they reserve a network address. In one embodiment, a network address in the overlay multicast system is represented as a combination of the address of the router and a unique port identifier, reserved for the socket of the router. For sake of convenience, a port in the layered multicast system will be referred to as a tport and a socket as a *tsocket.* When a user application creates a tsocket for receiving normal unicast packets, the tsocket automatically binds a local tport at the router daemon through a remote call to the client IO multiplexor 207 at the router. In one embodiment tports are bound exclusively and other clients may not use it concurrently.

[0067] In one example embodiment, communication sockets 225 communicate with a local input output (IO) multiplexor 229 that coordinates the handling of communication between the sockets and the router kernel through the formation of RPC calls, native inter-process communication or similar systems. A local IO multiplexor 229 utilizes an RPC stub 223 or similar program to communicate with the router via an RPC skeleton 219 and client adaptor 209. A client IO multiplexor 207 at the router manages the relay of these socket requests to the kernel 201.

[0068] ... one embodiment, the router daemon 109 uses a packet buffer to temporarily store packets for each connected client application until they are picked up. An interceptor pipeline in the client adaptor 209 or similar process may be utilized for this buffering function.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 371 of 830

[0069] To overcome the problem of growing routing tables, computation time and excessive advertisement overhead, large networks can be partitioned into smaller sub sections, connected by gateways. While the gateway routers maintain routing information necessary to route packets to nodes in other network sections, nodes inside a section or domain only maintain information for those nodes inside the same domain. By substituting logical ranges of hosts in the routing table by one single condensed entry, the size of the routing table is reduced. This is called address summarization. This mechanism introduces a form of hierarchy that allows the network as a whole to grow far beyond the practical limits of standard distance-[[ ]]vector or link-state algorithms. The farther a destination host is away, the more efficient it can be condensed together with other remote hosts. The more summarization is applied, though, the less efficient the paths become on average. The factor by which the actual data paths on summarized networks differ from the optimal paths, is known as the *stretch-factor:* the maximum ratio between the length of a route computed by the routing algorithm and that of a shortest path connecting the same pair of nodes.

[0070] In one embodiment, the overlay multicast system takes a relatively straightforward approach to address summarization. An administrator decides at deploy time which nodes form clusters and which clusters form aggregations of clusters. This is done by encoding hierarchy in the layered multicast system node addresses using a dotted or similar notation. Node addresses may be ASCII strings. In one embodiment, the strings are limited to at most 127 characters. In [illegible], only [a-z] and [0-9] are available. In another embodiment any characters, numbers of similar symbols may be utilized. Addressing may be case-sensitive or case-insensitive.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 372 of 830

[0071] **Figure 3A** is diagram of one example embodiment of a layered multicast system network divided into logical clusters. The example illustrates a network of eight nodes, divided into three clusters. Assigning nodes to clusters may be based on geographical properties, administrative boundaries, wide area links and similar considerations. For example, nodes inside a corporate network are all assigned the same logical domain, whereas a network that connects nodes from different corporations, would usually assign a separate domain to each corporate network. Another criterion for assigning nodes is that nodes that often disconnect and reconnect again later, possibly because the overlay multicast system routers run on personal computers, are placed in a subdomain, to avoid the routing updates triggered by their changing state to propagate far into the network.

[0072] Given the logical domains that cluster groups of nearby nodes, each node can treat domains other than its own as a single entity and use a wildcard address that matches every host inside that domain. This reduces the size of the routing table, as well as the amount of routing information that needs to be exchanged between the nodes when the topology changes inside a domain. For example, when a new host is added to domain S1, there is no need to propagate that information to the other domains, as they already have a wildcard entry that will match the new address.

[0073] **Figure 3B** is a diagram of a routing table for the network of **Figure 3A.** The example routing table shows the effect of address summarization in this network on the routing

S2.C. The cost value that is listed in the third column represents the cost of the shortest path to the nearest node inside that domain. In the fourth routing entry, the number in the cost column is the cost to reach S1.R from S2.C. If it is assumed that both interdomain links

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 373 of 830

S2.C_-_S1.R and S2.A_-_S1.R have the same weight or cost, S2.C will route all traffic for domain S1 through neighbor S1.R.

[0074] In one embodiment, when nodes exchange distance vectors, either because of a link change, or as a normal, periodic exchange, the receiving node first summarizes the destination address of each distance vector relative to its own address. Summarization is done by matching the destination address with its own address field-[[ ]]by-[[ ]]field and when a field does not match, the rest of the address is substituted by a wildcard. For example, when S2.C receives a distance-vector from S1.R that contains a path to destination S1.P, it is immediately summarized to .* upon arrival at node S2.C. This is because the first field differs from the first field of the local address and as such the remaining fields are replaced by a wildcard. This wildcard value is then fed to the distance vector algorithm that checks whether the new cost or path-length is shorter than the cost of the entry that was already in the routing table. In the present example there already is an S1.* wildcard entry in the routing table that was derived from neighbor destination S1.R. Since the path to S1.P runs through S1.R, the path to S1.R will always be shorter than the path to S1.P, so the entry in the routing table will not be changed and no further routing updates will be propagated to S2.C's neighbors.

[0075] In general, when a local address is 1.2.3.4 and an incoming distance-vector advertises destination 1.2.2.2, it will be summarized to 1.2.2.*. Destination 2.1.6 becomes 2.*, destination 1.2.3.4.5.6 becomes 1.2.3.4.5.*, destination 1.2.3.5 stavs 1.2.3.5 and destination stays 1.2.3.4.5. Note that 1.2.3 and 1.2.3.* are two different addresses. The first only matches the exact address 1.2.3 while the second is a wildcard that matches everything that starts with 1.2.3 and has at least 4 address fields. This includes 1.2.3.4 and 1.2.3.4.5.6, but not 1.2.3. If this mechanism of address summarization is used in an ExBF implementation that

SCALE VIDEO CODING, LLC v. CISCO SYSTEMS, INC. **IPR2024-01371** **Ex. 1002, p. 374 of 830**

carries tuples containing destination, cost and head-of-path attributes in its vectors, then the destination address is summarized, as well as the head-of-path address.

[0076] **Figure 4** is a diagram of an example network with a hierarchical structure. When the example network first starts to converge using the ExBF algorithm, node B.X advertises the following routing information to neighbor ~~A.S: DV$_{B.X,A.S:}$~~ A.S: DV$_{B.X,A.S:}$ {(B.X,_*,_0), (B.Y, B.X, 1), (B.Z, B.X,1)}. When A.S receives the vectors, it summarizes its entries. What remains is the single vector ~~DVax.As: {(~~ DV$_{B.X,A.S:}$~~{~~(B._*,_*,_0)}. The routing table of node A.S now contains RT$_{A.S}$ ~~RT$_{AS:}$~~ {(A.S, *, *, 0), (A.R, A.R, A.S, 1), (B.*, B.*, A.S, 1)}. All addresses are summarized before processing. This includes neighbor addresses. The consequence of this is that when a node has more than one connection with a foreign domain, both neighbor addresses will be summarized into the same wildcard. This leads to ambiguities and nondeterministic routing when this wildcard is listed as the preferred hop in a routing entry, as [[ ]]it cannot identify a single outgoing link . This problem is solved by assigning a local identifier to each link and using these numbers in the preferred hop column, rather than the addresses of the links peers. The routing update A. S sends to A.S.Y contains ~~DV$_{AS,A.S.Y:}$~~ DV$_{A.S.A.S.Y:}$ {(A.S, *, 0), (A.R, A.S, 1), (B.*, A.S, 1)} and leads to A.S.Y's routing table ~~RT$_{A.S.Y:}$~~ RT$_{A.S.Y:}$ {(A.S.Y, *, *, 0), (A.S.X, A.S.X, A.S.Y, 1), (A.S.Z, A.S.Z, A.S.Y, 1), (A.S, A.S, A.S.Y, 1), (A.R, A.S, A.S, 2), (B.*, A.S, A.S, 2)}. When A.S.Y has finished updating its routing table, it advertises ~~DV$_{A.S.Y,A.S:}$~~ DV$_{A.S.Y,A.S:}${(A.S.Y, *, 0), (A.S.X, A.S.Y, 1), (A.S.Z, A.S.Y, 1), (B.*, *, *), (A.S, *, *), (A.R, *, gnor [[.]]A. S where the asterisk indicates unreachable in the last three records to avoid long-lived loops. These loops were detected through the normal back trace mechanism of ExBF.

SCALE VIDEO CODING, LLC v. CISCO SYSTEMS, INC. **IPR2024-01371** **Ex. 1002, p. 375 of 830**

[0077] In one embodiment, summarization means that only a single path for a range of remote nodes is maintained. The consequence of this is that packets will not always be routed according to the real shortest path between source and destination. To illustrate this effect in the multicast network, consider the example network of **Figure 4.** Node S1.R receives the distance-vectors of both S2.A and S2.C. And after summarization learns that both [[bo.th]] neighbors offer a route for wildcard S2.*. If it is assumed that both interdomain links (S1.R - S2.A and S1.R- S2.C) have equals costs, then S1.R will choose S2.A to be the preferred hop for S2.* because of the fact that its address logically comes first. When S1.R needs to forward a data packet to S2.B, it uses the S2.* wildcard entry and forwards the packet to neighbor S2.A. Unfortunately this is not the shortest path to S2.B, as S2. A first has to route the packet through S2.C. Instead, S1.R should have sent the packet directly through neighbor S2.C. This ratio between the length of the actual path and the length of the optimal path between the endpoints are called the stretch factor.

[0078] In one embodiment, path stretching on the overlay multicast ~~multicas.t~~ system occurs when packets are forwarded between nodes that are in different logical domains. An entire subnet is treated as a single node with several outgoing links. Because a virtual node often contains a large number of nodes, connected by its own internal network structure, it is sometimes better to choose a different interdomain link when sending packets to the domain.

[0079] Since the overlay network runs its own adaptive routing algorithms, the content ɡh the network are constantly rerouted to avoid the network's hot spots and congestion. This can be particularly useful on wide area networks that are used for very different types of applications and hot spots are dynamic (i.e., moving around). Another advantage of

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 376 of 830

having custom routing algorithms ~~is. The~~ is the freedom of substituting them with others in the future.

[0080] In one embodiment, the software routers implement load balancing inside the network. Traditionally routing algorithms seek for "best paths" through the network and send data streams over these. However, it is sometimes much more desirable not to just send streams over this optimal path, but to also select several sub-optimal paths and divide the stream over all of them. This also avoids the optimal path from getting congested when the stream requires more bandwidth than this single path can provide.

[0081] **Figure 5A** is a diagram demonstrating summarization of an example network with an S2 [[82]] domain. When node S1.P needs to forward a packet for destination S2.F [[82.F]], it will send the packet directly down its interdomain link to the S2 [[82]] domain. In this example both interdomain links are assumed to have equal costs. **Figure 5B** is a diagram demonstrating the stretch factor for the example network of **Figure 5**A and shows the inner network of the S2 [[82]] domain. **Figure 5B** clearly shows that the stretch-factor is quite high for packets from S1.P to S2 [[82]].F, as the optimal path runs through S1.R instead. Although the overlay multicast system address summarization technique cannot ~~can not~~ guarantee a maximum upper bound on the stretch-factor, it can manipulate the stretch-factor by changing the summarization opacity.

[0082] By default, an address is summarized after the first field that differs from the local address. However, if that is changed to the second field, the overlay multicast system can look inside other domains for one level. A node with an address 1.2.3.4 will then summarize 2.3.4.5 into 2.3.*, rather than 2.* and 1.3.4.5 into 1.3.4.*, rather than 1.3.*. Doing this at least at the border nodes in the overlay multicast network that have the interdomain links reduces the

stretch[[ ]]-factor under certain circumstances. As the overlay multicast network was designed to be administrated by independent parties, administrators are free to experiment with different summarization opacity levels without jeopardizing the other subnets or domains.

[0083] One advantage of the summarization of addresses in routing tables is that the number of links that are present between nodes from different logical domains is irrelevant with respect to the number of routing entries in the routing table of a distance-vector protocol or similar protocols. Since these tables only contain destinations with a single forwarding strategy, the number of interdomain links does affect the size of the routing tables. If *y* is identified to be the number of entries in a node's routing table, *x* [[X]] to be the total number of nodes in the entire network, n to be the number of entities (nodes or nested domains) inside a domain and m to be the depth of the hierarchy. From this it follows that the total number of nodes in the network can be calculated with:

$$\forall\, x = n^m \text{ where } n \in \mathbb{N} \wedge n \geq 1 \text{ and } m \in \mathbb{N} \wedge m \geq 0$$

[0084] The relation between domain depth, number of nodes and [[ ]]domain density can be expressed by the following formula for a network with a uniform topology:

$$y = \left(\frac{\ln(x)}{\ln(n)}\right) * (n - 1) + 1$$

[0085] Given this formula of the address summarization's effectiveness, it can be shown that routing entries in[[ ]]any network node remains under 100 even when the network as a whole grows to well over 10 million nodes. It can also be shown that the domain size n has no spectacular effect on the scalability. While a small value n yields a relatively deeply nested network hierarchy, which implies more routing entries, it also means that each [[ ]]hierarchy level only contains a small number of entities. A large value *n* in a network of the same size

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 378 of 830

yields a relatively flat hierarchy with few levels, each level contains a large number of entities. As a routing table can only contain a natural number of entries, its number remains constant as nodes are added to domains that[[. A]]are already captured by the existing wildcard entries.

[0086] An advantage of using a string based node addresses scheme is its large address space and flexible naming. New subdomains may be added at any time as there is no need for predefined or fixed hierarchical levels. All nodes inside a flat domain can be reconfigured into nested subdomains without the need to reconfigure routers or compute netmasks. Also, as address fields are not bound to a limited number of characters, they contain location or company names to make administration of the network easier and more transparent. An example might be the address n1.amsterdam.level3.marketxs.node4, or simply nl.ams.13.mxs.n3 to keep the[[.]] address relatively short. The drawback of these kinds of unrestricted addresses is the amount of bytes they require. Because every data packet always contains a source and a destination address, the amount of overhead per packet is excessive at times.

[0087] In one embodiment, the string based node addresses scheme is implemented by letting[[·]] the first bytes of each address header specify the length of the address in bytes, followed by the address itself as a normal 8 bit ASCII string with a maximum of, for example, 256 bytes. In one alternative, null-[[ ]]terminated strings are used.

[0088] In another embodiment, a sufficiently large address space is used and divided [[it]] into logical subnets, similar to IPv4 and IPv6 addresses. In or a 64 bit address space is used and divided into 8 bit fields (identical to IPv6) used to identify logical subnets. In this embodiment, addresses might look like 120-23-61-201-43-146-128-132. To make distinct addresses, IPv4 uses the dot "."[[\\ *IIJ]* for separation, IPv6 uses the colon ":" and the present system uses a hyphen "-". When each address field represents exactly one logical subnet level,

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 379 of 830

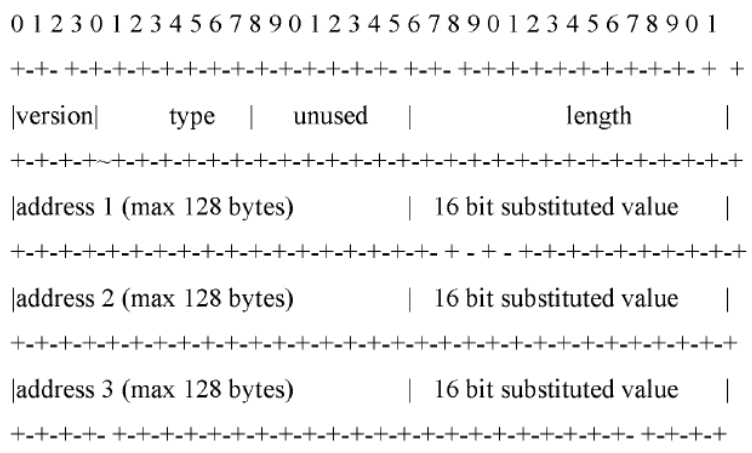
this scheme provides a simple, but at the same time somewhat limited, way of addressing. It means that the complete network can address as many individual hosts as IPv6, while grouping them in at most 8 levels of sub domains, where each sub domain can contain a maximum of 256 hosts or sub domains. The sub domains can be utilized to indicate a network hierarchy in the topology, including a geographical or similarly based hierarchy.

[0089] In one embodiment, to reduce the amount of bandwidth wasted on this overhead, a mechanism is used for substituting the address strings for small 16 bit values or similar small sized value when transmitting a packet to a neighbor. Also, a translation table is kept at both sides of the connection that is used to put the original address string back in the packet before passing it to the router's packet switching kernel. This optimization works on a per-[[ ]]link basis and may be totally independent from the communication with other neighbors. When it is used on more than one interface, each interface maintains its own translation table.

[0090] If a packet is transmitted to a neighbor, all address strings are stripped from the packet header. For each address string a 16 bit or similar value is generated and stored in the translation table. These 16 bits or similar values replace the original addresses and before the packet is transmitted, its protocol version field in the first header of its[[ ]] wire-level representation is changed to a higher version. This version number is reserved for optimized packets that would be unreadable for routers without support for address substitution. To ensure proper translation by the peer, the interface first sends a special packet that contains the new

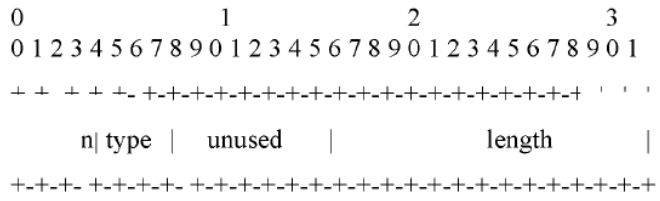
tutions. Such a packet is tagged with the higher protocol version number and contains one or more tuples of address strings[[ ]]and their substitute values. In one embodiment, such a packet takes the form:

0 1 2 3



Type 0[[O]]x1 is used for issuing new substitutions and type 0[[O]]x2 is used to invalidate an earlier or unknown substitution. The system attempts to ensure that this (or similar) packet is received by the peer prior to the data packet itself, which is something that cannot be guaranteed by every transport layer. TCP or similar guaranteed communication method may be used for communication between neighbor nodes in the overlay multicast network.

[0091] In one embodiment, configuring an interface for address substitution can be done manually, or automatically. In the latter case the interface uses a special handshake packet that is part of the higher protocol version. In one embodiment, the handshake ~~and shake~~ packet takes the form:



The version field has the higher protocol version. The type for the handshake packet is 0x0[[OxO]] and the length is 4. A peer transmits such a (or similar)[[}]] packet when the

connection is first established. When the packet is echoed back by the peer, the interface knows that the neighbor also supports substitution and goes into substitution mode. After a connection with a neighbor has been lost and is re-established, the handshake is performed again as the peer could have been replaced by software that does not support substitution.

[0092] In one embodiment, entries in the translation table all have a time-out or similar tracking mechanism to ensure timelines and accuracy of entries. After this period the entry is removed from the table and any packet still using the substituted value will not be processed and cause the receiving node to respond with a special packet containing the substituted values that weren't recognized, allowing the peer to synchronize translation tables by sending a packet containing the substitution tuples. Since the invalidation of timed-out entries is automatic, both peers use the same timeout or similar values in tracking mechanisms. In one embodiment, when the translation table is full, entries are removed according to a last recently used scheme. The address substitution is implemented in a substitution interceptor that is the lowest interceptor or a part of the lowest interceptor or other interceptor reconstructing packets immediately on arrival in an interface module of a router. Substitution introduces additional processing overhead and enabling it is [[.]]optional for each node.

[0093] In one embodiment, the overlay multicast system may be used solely for routing. The overlay multicast system offers multicast support over existing IP networks. The overlay multicast system goes beyond offering plain multicast functionality and additionally focuses on

e multicast streams with limited jitter and delay through the combination of overlay multicast, multicast congestion control and layered multicast.

[0094] In the overlay multicast system the ExBF protocol or similar routing algorithm implicitly contains information on which links in the network are child links. For example, this is

a by-product of the long-lived loop avoidance mechanism in ExBF. In ExBF, a node will inform its neighbors about the fastest route it offers to a destination S, except the neighbor that is in the path towards S. Instead, an infinite distance is advertised to this neighbor. This information is used by a router to conclude that each neighbor advertising that node S is unreachable, is actually using it as their parent node towards S and hence will immediately mark those links as child links in the shortest path tree rooted at S.

[0095] Neighbors that do advertise reachable routes to destination S are not flagged as child links and therefore not used when forwarding multipoint packets from S. In another embodiment, this implicit information is available through any distance-vector protocol that uses poisoned reverse or similar techniques. Both methods are used to quickly deliver a copy of any multipoint packet to all nodes in the network. A disadvantage of both mechanisms however is that a copy of every packet is delivered to every node, regardless of whether or not that node is actually interested in multipoint packets from that source.

[0096] In one embodiment, the overlay multicast system is equipped with a sparse forwarding mechanism. The overlay multicast system network is used for a single source multicast for high volume market data, video streams or similar data to be sent to large numbers receivers. In these types of uses it is not necessary that receivers themselves or other nodes be capable of publishing data packets to the multicast group. Instead, for security reasons it is better to know that only the real source can publish to the group. Given these requirements, together with the fact that multi-source protocols are more complex than their single-source alternatives, in one embodiment, the overlay multicast system preferably uses a custom single-[[ ]]source protocol for multicast distribution on the network. In another embodiment, the overlay multicast system supports multiple sources. For sake of convenience an embodiment of a single source

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 383 of 830

multicast network is discussed. One of ordinary skill in the art would understand that the principles of operation can also be applied to a multi-source application as well.

[0097] **Figure 6** is a diagram of an example embodiment of a multicast distribution or "sink" tree. In the example embodiment, the multicast system starts with a source q that is active, but in this example, a sparse-mode protocol, with no receiver yet subscribed, no distribution tree exists. When the first receiver application p_0 [[p_o]] subscribes, its router initiates the reverse computation of one branch of the sink tree rooted at source *q*. It does so by first marking that p_0 [[p_o]] has a local subscriber for q:G by setting the bit in LSp_0[[$_o$]] [q,S] (LSp[] is a local array at node p that keeps track of all subscriptions of the local user applications connected to router p) to true and then sending a join (e.g., a <join (q:G)>) packet to the preferred hop towards source *q*. This neighbor *p1* receives the <join (q:G)> packet and marks the link with p_0 [[p_o]] as a child link for multicast group q:G. It then forwards the packet to the next hop *p2* in the shortest path towards *q*. Eventually node p_n sends the <join (q:G)> packet to *q*. On receipt, *q* marks the link on which the packet was received as a child link for its local multicast group G and starts forwarding all packets addressed to G over the link to node p_n. The multipoint packets addressed to G are inserted into the overlay multicast network at node q by the user application that previously bound the local multicast group *q:G* in publish mode. Nodes p_n, p_{n-1}, ...,[[$\cdots$]] p_1 all forward the packets to their neighbors from which a <join (q:G)> packet was received earlier. Node p_0 [[p_o]] has no child links for *q:G*, but does have LSp_0[*q*:G] ~~LS$_\phi$[Q:G]~~ set, so it delivers

the local application. When another node p_1 becomes interested in q:G while LSp_1[*q*:G] ~~LS$_{pi}$[q:G)~~ is not already set, it sends a <join (q:G)> packet to its preferred hop towards ***q***. Let p_{i+n} be in the path from p_i to q that receives the <join (q:G)> packet and suppose that p_{i+n} [[$Pi+n$]] is also in the path between p_0 [[p_o]] and *q*. In this case, p_{i+n} *is* already in the established

SCALE VIDEO CODING, LLC v. CISCO SYSTEMS, INC. **IPR2024-01371** **Ex. 1002, p. 384 of 830**

part of the sink tree of *q*, so it does not need to forward the <join (q:G)> packet further to *q*. Instead, it only marks the link on which it received the packet as a child link for q:G. In general, a node u only forwards a <join (S:G)> packet towards S if it is not already subscribed to S :G (hence, LS_u[S:G] is not set and the collection of child links for S:G is empty).

[0098] The example sparse-mode distribution tree generated when processes subscribe to a S:Q group originally equals a part or all of the optimal sink tree rooted at S. However, when the underlying unicast routing algorithm detects changes in the network performance and updates some of the routing table entries, the optimal sink tree changes accordingly and may no longer be matched by the multicast distribution tree, rendering the tree as well as the multicast performance suboptimal. An extreme example of this is when a link that is part of the multicast distribution fails entirely.

[0099] **Figure 7** is a diagram of an example overlay multicast network with a failed link. In the example a node *u* currently in the distribution tree for q:G only has a single route to *q* via *w* [[*W*]]. The link between *u* and *w* fails. The node ***u*** concludes that it can no longer be part of the distribution tree (as u and q are no longer connected) and invalidates all child link subscription information. In case LS_u[q:G] is set (i.e., the subscription table indicates that *u* is subscribed to group G from node *q), u* will rejoin the distribution tree as soon as a new path to q is found. In another embodiment, the algorithm sends a notification to the locally subscribed application to indicate that *u* [[U]] is no longer connected to *q*.

[0100] ... the example, it is assumed that all links have weight 1, while the link between p and *w* has weight 100. The result is that the link connecting ***p*** and *w* is not in the optimal sink tree rooted at *q*. Since only t and ~~v~~V have a local subscriber (LS_t[*q*:G] ~~{LS_t[q:G]~~ and LS_v[q:G] are set), the sparse distribution tree in this example equals the dotted arrows of **Figure 7.**

SCALE VIDEO CODING, LLC v. CISCO SYSTEMS, INC. **IPR2024-01371** **Ex. 1002, p. 385 of 830**

[00101] In the example, nodes w [[*W*]], *u*, r and *s* all have only a single route to *q*. Node *p* has a path through both *s* and *w* ~~sand *W*~~, while *s* is [[Sis]] the preferred hop towards *q*. When the link between *u* and *w* fails, *u* becomes disconnected from *q* and invalidates the subscription information for link *u-r*. Node *u* informs neighbor r of this fact as part of an immediate distance-vector exchange. This routing update implicitly tells node *r* that its join for q:G through *u is* no longer valid and that it should look for an alternative neighbor to rejoin.

[00102] In the example, an explicit leave packet to *u* is unnecessary. Since r is also left with no alternative path to *q* and has no locally subscribed applications, it invalidates its q:G subscription and is no longer part of the distribution tree. Neighbors ~~(and *S*~~ *t* and *s* are implicitly informed about this through the ExBF routing update or similar routing update sent by r. Eventually w, u, r and s all leave the distribution tree, while nodes t and v schedule a rejoin when a new path to q is found. Again, they inform their local subscriber applications about this. When p receives the routing update from s, it loses its preferred hop for q and switches to neighbor w. The new cost to q is now 100 plus distance (w, q) and neighbor s is informed. Upon receipt, s discovers the new path to q and informs its neighbor-s r and v. Node v then sends the <join (q:G)> packet to s. Nodes s, p and *w* [[*2*]] then construct the first part of the new, optimal distribution tree rooted at q. When the new path reaches r and *t*, *t* sends a join packet to r and r to *s* [[*S*]], reconnecting all receivers to the data stream.

[00103] In one embodiment, recovering from link failure or similar communication errors is

:e steps. In the first step the underlying ExBF or similar routing protocol starts a wave that propagates a link failure through the network. Depending on the actual topology, this takes up to N-1 packets, where N represents the number of nodes *in* the network. The worst case

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 386 of 830

time complexity for recovering from a link failure is 3N when ExBF or similar algorithm is used as the underlying unicast routing algorithm. A best case time complexity would be 2N.

[00104] The example discusses a scenario where the system recovers from a failure of a link that is part of the multicast distribution tree. More often however, links will not fail completely, but rather fluctuate in quality, causing the underlying routing algorithm to reassign preferred neighbors. In this example case it is not a requirement that the distribution tree is changed to reflect these changes, as it is not partitioned. However, since the tree becomes suboptimal, it is changed.

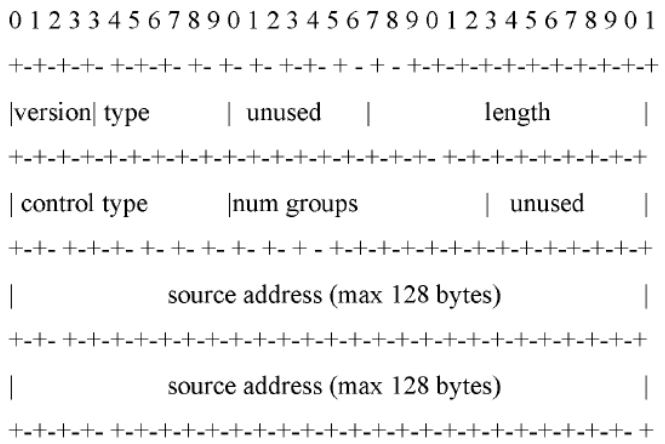
[00105] In one embodiment, the decision whether or not to recompute a distribution tree after a routing table update is made according to the quality of the used path. When a node *p* in the tree detects that the neighbor in the optimal path towards the multicast source has a cost that is only marginally lower than the neighbor that is currently *p's* parent in the actual distribution tree, the overhead of recomputation and the risk of packet loss outweighs the increased performance of the new tree. Also, since the quality of each link is continuously re-[[ ]]evaluated, the updated sink tree may only be temporal. Two straightforward solutions to this can be used. Either the algorithm sets a threshold on recomputation so that subscriptions are only moved to the neighbor with the currently optimal path towards the source if the decrease in cost is at least a ratio x [[*X*]], where x > 1. A larger value of *x* then postpone tree adjustments until a substantial improvement can be gained,[[,]] while a smaller x makes the tree actively follow the changing characteristics ing network.

[00106] Another solution is to postpone recomputation of the tree for a period inversely proportional to the cost decrease ration x [[*X*]]. The latter has the advantage that the optimal

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 387 of 830

distribution tree is always guaranteed to be reached in finite time since the last routing table change.

[00107] In one embodiment, to run the tree building protocol, two new messages are introduced: the join message and the leave message. Both messages use a GroupData control message of the following format:



This message is used to send a list of subscriptions to a neighbor node or to cancel an aggregated list of subscriptions with a neighbor. The action field (9th byte) is used to distinguish between join and leave. The 10th and 11th byte indicate the number of (S, G) groups in the message. The action field is either "join" (0) or "leave" (1) [[{1)]]. In an alternative embodiment, this is extended to "stale", "dead", etc., to indicate the state of the publishing application.

[00108] Many multicast applications require some level of reliable communication.

hıs are uploading files to several receivers simultaneously or replicating web server caches . Without guaranteed delivery, a multicast transport service has limited applicability. In one embodiment, the overlay multicast network utilizes standard delivery control based on

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 388 of 830

moderating the sending of data based on the lowest bandwidth available on the route to a destination. However, this delivery system is unsuitable for certain applications.

[00109] One example is financial data distribution, it needs a transport service that can multicast live data without packet loss or corruption and without substantial end-to-end delay when parts of the network are temporarily or permanently congested. While live market data cannot tolerate random packet loss, it can be thinned. Under certain circumstances it is acceptable to omit certain stock quote updates. An example is a desktop application that displays real-time updates for hundreds or thousands of financial instruments. If all quote updates were delivered to this application, this would require a substantial amount of bandwidth and would cause the value of the more volatile instruments to change faster than a human can read. The data may be thinned through a process that involves inspection of the individual stock quotes and encoding them in individual data packets, tagging each with a priority value. For this application, as long as all packets labeled with the highest priority number are received, the partial stream can be considered intact. Additionally, when all updates of the second highest priority are also received, the quality of the partial stream is increased usually in the form of less latency.

[00110] In one embodiment of the overlay multicast system, priority numbers are associated with data packets. The priority numbers represent a logical layer inside a data stream. Data provided by a multicast application may be in the form of a stream of data. This data is subdivided into categories or priorities based on the nature of the data. When packets in a data

eled with priorities in the range 0 to 3, the stream is said to have 4 layers. Also, the convention is to treat 0 as the highest priority and 3 as the lowest. Any other system of identifying priority levels may be utilized including alpha numeric indicators or similar identifiers. If a stream only contains a single layer, all packets are labeled with priority 0.

SCALE VIDEO CODING, LLC v. CISCO SYSTEMS, INC. **IPR2024-01371** **Ex. 1002, p. 389 of 830**

[00111] To software router daemons, a priority value of a packet is relative to the stream and becomes relevant when a decision to discard data at a congested router is made. The priority value has no meaning other than as a criterion for discarding packets on congested links. When an outgoing link of a router has insufficient bandwidth to transmit all pending packets, it forwards only those packets with a designated priority or the highest priority. This technique is also applied to any data type, including audio/video data and financial data. Using this system, the packet priority numbers cannot be misused by sources to give their data packets a greater chance of prioritized transmission by giving them the highest priority to gain advantage over other sources. Packet priorities are only compared between packets that are part of the same data stream.

[00112] In one embodiment, knowing that routers will use the packet priority numbers when making forwarding selections on congested parts of the network, a source carefully divides its data packets over different layers or priorities, in a way that a subset of the layers still contain a useful, uncorrupted representation of the data. This system is especially useful when multicasting a live data stream with a high data rate to a large number of receivers, scattered over a heterogeneous wide area network. Receivers that are on congested parts of the network will then receive the highest priority parts of the data only. This eliminates the need to lower the publishing rate to match the slowest receiver, while still being able to offer live, uncorrupted, thinned data streams to clients suffering from insufficient bandwidth. Example applications of

[illegible] audio and video codecs that divide live multimedia content over layers to enhance the user experience over wide area networks.

[00113] In one embodiment, incoming messages are sorted by unicast or multicast sender address. As described above, a sender address is the combination of source router address and

application session. An example of a source address is n1.mxs.office. erik:video.an2 which could be used by a user application broadcasting a video channel. In this case, only the source address is relevant, not the (uni- or multicast) destination address. Each incoming message is added to the queue that holds messages sent by that particular sender. If this is the first message from a sender, a new queue is automatically created to store it.

[00114] **Figure 8A** is a flowchart of one embodiment of a process for managing congestion in the overlay multicast system. The congestion can be managed at each individual router. In one embodiment, the congestion is managed at the interface module level in each router. Each interface module has an inbound pipeline and outbound pipeline discussed above for processing inbound and outbound data. Each pipeline buffers data that is awaiting further processing. However, if either pipeline is unable to keep up with the pace of incoming data that needs to be processed some data must dropped.

[00115] In one embodiment, data is received as a set of data streams at each router (block 851). The data is then buffered in the inbound pipeline buffer (block 853). The same process applies to outbound data that is received from the kernel by the outbound pipeline. This data is stored in the outbound buffer. After the [[.]]data has[[·]] been stored a check is made of the inbound or outbound buffer to determine if it is full (block 855). In one embodiment, data is stored in the data structures in the buffers that organize the data packets into a set of queues. Each source address, data stream or layer in a data stream has a separate queue. A queue is sorted riority and oldest data packets at the front of the queue. If the buffer is full then a decision is made to drop a designated amount of data in the form of packets from the buffer to make room for incoming data packets (block 857).

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 391 of 830

[00116] In one embodiment, a queue is chosen randomly or in a round robin to have data dropped. In another embodiment, the queue with the most data is chosen to have data dropped. In a further embodiment, a weighting factor is calculated to determine which queue is selected to have data dropped. The weighting factor is based on the amount of data in a queue, size of packets in a queue and similar factors. The weighting factor counteracts unfair distribution that is caused by selecting a queue by a round robin, random or similar method of selection. Data streams with large packets are unfairly affected by other methods because a disproportionate amount of data is dropped in comparison with other queues with smaller packets. Queue selection is also influenced by the size and the amount of data that is intended to be dropped. A large queue that can drop close to the amount of data desired is weighted for selection.

[00117] In one embodiment, data is dropped if a total amount of data stored in all queues exceeds a threshold value. This threshold value is set by an administrator or is a set value. These systems enforce the fair allotment of bandwidth between data streams. Also, this system implements the prioritization of logical layers by dropping lower priority level layers when congestion occurs. If the buffer is not full then the pipelines may continue to store data in the buffers (block 851).

[00118] In one embodiment, while the queues grow in size as packets are received, a background thread constantly dequeues packets from the queues and transmits them over the network.

[00119] ... one embodiment, the system that manages in- and output for the queues is divided in two parts. The first part is run by the background thread that constantly dequeues packets from the queues and transmits them over the network, while the second part is in charge of queuing

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 392 of 830

new packets that are to be transmitted. The latter also implements the logic that defines when and which packet should be discarded (due to a buffer overflow).

[00120] The dequeuing part of the system, when selecting a packet for transmission, only looks at the first packet (most urgent) of each queue. To make sure each stream gets an equal share of bandwidth, it takes the individual packet sizes into account. When all packets (note that only the first packet of each queue is observed) have equal size, the dequeuing thread simply selects a random queue with equal probability and dequeues one of its packets. Since each queue has the same probability of being selected, each queue will deliver an equal amount of packets per time unit. Hence, each source will transmit an equal amount of bytes per second.

[00121] Since individual packets can have any size between 1 and some determined maximum number of bytes, their size must be considered when the transmit thread selects queues. Queues with many large packets should generally have a smaller probability of getting selected than queues with lots of small packets in order to keep the bandwidth division fair.

[00122] In one embodiment, only the first packet of each queue is considered to represent its queue and give it a probability of setting selected that is reversely proportional to its size in bytes. In an example where there are three queues, P, Q and R where the first packet in P is 100 bytes, the first packet from sender Q is 300 bytes, while the packet from R is 500 bytes, to calculate the selection probabilities, first their respective selection weights are defined. For example, the weight of the packet in P is computed by dividing the total size of P, Q and R by P's size: (100 + 300 + 500) /[[/]] 100 = 9. Q's weight is 3 and R's weight is 9/5. The weights are converted into selection probabilities by dividing them by the sum of all weights. This gives P a probability of 9 /[[/]] (9 + 3 + 1. 8) = 0.65, Q a probability of 0. 22 and R of 0.13. When a queue becomes empty after dequeuing, it is removed.

[00123] In one embodiment, to keep the delay introduced by buffering packets under control, the second part of the system enforces a maximum total queue size. The sum of all packets in all queues may never exceed this threshold. When a new packet comes in, it is always accepted at first. If necessary a new queue is created for it, or it *is* added to its designated, existing queue. After adding a new packet, the total size of all queues is checked. If it is larger than the configured maximum, the algorithm runs a removal round in which it first selects a queue and then tells that queue to remove one packet. If the queues are still too big after removal of one packet, the process of selecting a queue and removing a packet is repeated iteratively until the total queue size is smaller than or equal to the configured maximum.

[00124] In one embodiment, shrinking the queues is always done after a packet was added, never preemptively. The reason for this is to let the new packet immediately participate in the removal selection process, rather than discarding it or making room for it at the expense of the other queues. Contrary to de-queuing packets, where the algorithm tries to select the "most urgent" packet, we now need to select the "least urgent" one. This packet is determined by looking at the size of the individual queues (the largest queue should generally be shrunk to keep resource division among the streams fair) and which queue can match the required size most accurately. This comes from the fact that individual packets can differ greatly in size, so removing the "least urgent" packet from queue P could result in freeing 8 kilobytes, while removing the "least urgent" packet from queue Q yields 40 bytes of space for example. Now if total exceed the maximum size by only a couple of bytes, it seems logical to remove the small packet from Q. This policy comes with a consequence, namely that it implicitly promotes the use of larger packets. After all, streams with lots of very small packets are more likely to accurately match the amount of buffer space that must be freed than stream queues with

few very large packets. This property may encourage developers to use larger packets, increasing the efficiency and throughput of the network. In short, when selecting a queue for shrinking, the system favors queues that [[are]] will expunge the least amount of bytes in order to match the total buffer capacity and are large compared to the others.

[00125] In one embodiment, before the selection can be made, the absolute weight factor for each queue is computed. The weight factor combines the queue's size and ability to accurately match the overcapacity of the buffer by removing packets. The latter is exposed in *v*. It represents the sum of the sizes of the packets that a queue must remove in order to eliminate the buffer's overcapacity. For example, if a queue P has 10 packets of 100 bytes each while the buffer currently has an overcapacity of 170 bytes (suppose the maximum is 10000 bytes, while all queues combined add up to 10170 bytes), P would need to remove at least 2 packets (2 * 100 bytes). In this case, v_P [[*vp*]] is 200. The formula to compute the absolute weight factor of queue m is **defined**:

$$w_m = \frac{(\sum_{k=0}^{n} v_k) \cdot s_m}{v_m}$$

In this formula, *n* represents the total number of queues, *v* represents the number of bytes that would be removed if the queue was selected and *s* represents the total size of a particular queue. The formula captures the direct relation to queue size and the inverse relation to the amount of bytes that would be removed by the queue if it was selected.

[00126] When the absolute weight factors of all queues have been derived, they are converted to weighted selection probabilities. This is done by dividing the individual weight values by the sum of all weights. The selection probability of queue *m* is expressed in P_m.

SCALE VIDEO CODING, LLC v. CISCO SYSTEMS, INC. **IPR2024-01371** **Ex. 1002, p. 395 of 830**



After the selection probabilities have been computed, a roulette-wheel algorithm is run to select a queue and let that queue remove its v bytes. If v is smaller than the current overcapacity of the buffer in total (note that this implies that the selected queue became empty and was thus removed automatically), another selection round is done until the buffers ~~shrunk~~shrink sufficiently. Note that the queue selection process is implemented as an atomic operation. During selection rounds no new packets may be added to or removed from the queues.

[00127] In one embodiment, the overlay multicast system is a layered multicast in combination with a scalable, selective packet retransmission mechanism to offer a service that can meet the demands of real-time financial data and similar applications. Selective packet retransmissions are crucial so that the overlay multicast system is guaranteed that subscribers will receive one or more layers that are completely intact. When an occasional packet in a high priority layer is [[ ]]missed, it is repaired. However, the receiver may decide not to attempt to recover missing packets from low priority layers that are missing due to network congestion. Trying to repair these layers would require additional bandwidth, resulting in additional packet loss. Yet, as more end-to-end bandwidth becomes available, the receiver detects this and repairs additional layers, so more intact layers are delivered to the application.

[00128] **Figure 8B** is a flowchart of one embodiment of a process for handling layer repair.

ment, the overlay multicast system supports a receiver-[[ ]]side multicast socket that receives the raw packets from the multicast stream and is in charge of repairing damaged stream layers before forwarding them to the user application (block 801). This method includes a retransmission request packet. When the receiver detects a missing packet in a layer that should

be repaired (block 803), it sends a retransmission request for this packet towards the source (block 805).

[00129] In one embodiment, detecting packet loss is done in the conventional way by sequencing each packet with an incrementing number. In another embodiment, each layer has its own sequence number, so all packets in one layer are sequenced independent of the other layers. This way, packet loss is detected in each layer, regardless of the conditions and packet loss in other layers.

[00130] To avoid a cascade of retransmission requests when a packet is dropped close to the source, a tree-based negative acknowledgment is used. This technique provides localized repair without communication with the source, because an intermediate router that has a copy of the requested data packet responds by sending it again and suppressing the retransmission request. When the next router closer in the distribution tree receives the retransmission request it checks its buffer to determine if the requested packet is still stored therein (block 807). If the requested packet is there, it is retransmitted (block 809). If the packet is not in the buffer then the retransmission request is forwarded to the next router in the distribution tree toward the source (block 811).

[00131] In one embodiment, when layers are dropped due to congestion, the programming API that is exposed to user applications notifies the user by means of an exception or special return value when it is reading packets. These notifications m[illegible] nd only serve to inform the user that the quality of the stream has changed.

[00132] In one embodiment, localized repair works by having each router daemon store a window of transmitted packets. Packets are either stored for a fixed period of time, after which they are discarded, or a fixed amount of buffer space is reserved to store the most recently

forwarded packets. Since the overlay multicast network by default use reliable connections between router daemons, packets never get lost while they are in transit between nodes. The only place where packets are purposely discarded is in the interface buffer interceptors. Thus, there is no need to buffer packets that were actually transmitted over the reliable connection, as those packets are guaranteed to have reached the neighbor node. In one embodiment, a router daemon copies packets to its retransmission packet store before those packets reach a buffer interceptor that discards packets.

[00133] The router's buffer interceptors copy packets to be stored for retransmission. In one embodiment, localized packet repair in the overlay multicast system are implemented by adding an interceptor to both interceptor pipelines of each interface and letting those interceptors simply store a copy of each packet. Additionally, when a retransmission request passes through the router on its way to the multicast source, the interceptor inspects the request packet and if it has the requested packet in its temporal packet store, it does not forward the retransmission request to the next interceptor. The requested data packet is injected into the network again.

[00134] **Figure 9A** is a diagram of two router ~~r outer~~ daemons connected by a link. There are two interceptor pipelines 903, 905 inside each interface 907. Each interface contains both an outbound 903 and an inbound 905 interceptor pipeline. The latter processes packets received from the network, while the outbound pipeline processes packets just before they leave the router daemon. In one embodiment, the pipelines contain two different interceptors: the CCI ontrol Interceptor)[[}]] 909 and the PRI (Packet Retransmission Interceptor)[[}]] 911. In another embodiment, a router daemon contains additional interceptor instances per pipeline.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 398 of 830

[00135] The CCI 909 implements bandwidth allocation rules and is responsible for discarding packets. During normal operation the source 913 publishes its multicast packet stream to its router daemon 901 (path 1) that sends the packets to all interfaces that lead to interested receivers according to the multicast routing table. In the present example, the packets are passed to the first outbound interceptor (PRI) of the interface (path 2). The PRI is responsible for storing a copy of each recoverable packet that is transmitted. It stores the copies in the private temporary packet store 915 of its interface (path 3). Then the interceptor passes the packets to the next interceptor (CCI) (path 4) where it is buffered until the connection link with the adjacent router daemon has time to transfer it.

[00136] In one embodiment, if the packets are not dropped in the outbound ~~outboard~~ CCI, they are received by the neighbor router daemon (paths 5 and 6) and sent through that router's inbound interceptor pipeline 921 (path 7). They first enter the PRI 923 (path 7), which stores copies in its inbound buffer 924 (path 8) and then passes them on to the next interceptor (CCI) 925 (path 9) where they are temporarily parked until the router's kernel 927 thread picks them up (path 10) and delivers the packets to the subscribed multicast tsocket which forwards them to their next destination 929 (path 11).

[00137] **Figure 9B** is a diagram of two router daemons connected by a link where a retransmission is requested. If a packet is lost in the outbound packet interceptor pipeline of the source router because the connection between the routers was not fast enough to transmit all the

ceiving tsocket notices the loss and send a retransmission request (path 1). The destination address of this unicast retransmission request packet is that of the multicast group. The network packet switching kernels recognize the retransmission packets and use the multicast group to route the packet towards the source of the group. The retransmission request packet of

the example passes the router kernel 927 and reaches the outbound packet interceptor pipeline 931 and goes into the PRI 933 (path 2). This interceptor scans specifically for retransmission requests and tries to answer them locally. To this end it inspects the packet, looks up the multicast group, the stream layer identifier and the layer's sequence number and then checks the inbound packet buffer 935 to see if it contains the packet (path 3). If this is not the case, the packet is passed on to the rest of the interceptors (path 4) and eventually transmitted to the adjacent router (paths 5 and 6) where it is fed to the inbound interceptor pipeline 941 (path 7).

[00138] In one embodiment, the PRI 943 checks the interface's outbound packet buffer 945 (path 8) and finds the multicast packet that was dropped by the congested outbound CCI 909 earlier. It is then re-inserted into the outbound packet stream via the outbound PRI 911 (path 9). The retransmission request packet is then dropped and not forwarded any further. Assuming the packet is not dropped again, it travels the normal way towards the receiver (paths 10, 11, 12, 13). In one embodiment, the receiver's PRI sees the multicast packet for the first time and stores a copy in its outbound packet buffer (path 14). The packet then continues on to destination 929 (paths 15, 16, 17). This mechanism of unicast retransmission packets and interceptors that offer localized repair is flexible in the way that not all router daemons need to support local packet repair. A router that does not support it simply forwards the requests and lets the upstream routers handle them. The network as a whole becomes more efficient when more routers support it.

one embodiment, because storing a copy of every multicast data packet requires storage capacity, the overlay multicast system links the packet buffers of the packet retransmission interceptors to a central data structure that will only store unique packets. This is done because in multicast transmissions a router that is a branch point in the multicast

distribution tree otherwise ends up storing at least three copies of every data packet. This is because the packet is received on one interface and sent out over at least two other interfaces. If every interface would individually store its packets, the inbound packet buffer of the receiving interface as well as the outbound packet buffers of the forwarding interfaces contain the same packets.

[00140] In one embodiment, the central packet store eliminates duplicate packets by storing each packet only once and by keeping reference tables that point to packets for every interface. When an interface buffer stores a new packet in the central store, while that same packet was already stored by another interface, the central packet store merely adds a pointer to that packet instance to the interface's packet reference table.

[00141] In one embodiment, each entry in the reference tables has a time-out attached to it. This is to ensure packets are only temporarily stored. In one embodiment, the same packet could pass different interfaces at different times. The time-outs are not attached to the packets in the central store but rather to the packet references in the interface reference tables. In one embodiment, packets are only expunged from the central store when all references to the packet have timed out. Another optimization that can substantially reduce the amount of required storage space is to only store packets that were explicitly discarded by the Congestion Control Interceptors (CCIs) ~~(CC is)~~ because of bandwidth constraints. Storing other packets has little value, as those are guaranteed to have arrived at the next router. No retransmissions may be

:hem unless some non-deterministic packet loss occurs as a result of a crashing node.

[00142] In one embodiment, if a packet was lost close to the multicast source and because of this and all receivers simultaneously send a retransmission request, the upstream routers still

apply conventional negative acknowledgement "nack" suppression techniques to combine all concurrent retransmission requests from its child links into a single request that is forwarded upstream towards the source. Applying local packet stores at every router reduces the latency of packet recovery.

[00143] In one embodiment, time to live values are adjustable per router, set system wide or similarly configured. A short time-out minimizes storage requirements, but will also lead to more overhead towards the source as more retransmission requests need to be propagated further upstream. Large timeouts will be more tolerant to late retransmissions, but require more storage. When a time-out of 60 seconds is used by the routers that means a lost packet can still be recovered after a minute. However, when the data stream contains real-time data such as stock quotes, such a delay in delivery cannot be tolerated. Problems get worse if the data packets also have to be delivered to the user applications in their original order. In that case the user does not receive any data for up to a minute while the missed packet is recovered and all newer packets are waiting for it. In one embodiment, a ten second timeout setting is utilized.

[00144] If all routers in the network use the same time-out value, there is no use in propagating these requests as the upstream routers are likely to have purged their copies as well. To avoid nack-implosions from slow receivers right after a packet is purged, each router interface remembers which packets it has purged. When a retransmission request is received for such a packet, the interface may respond by sending a special packet to the receiver that indicates ıted packet had timed-out and cannot be recovered anymore. The retransmission request in this scenario is not propagated further upstream. When a receiver gets this timeout notification, it knows it will not have to wait any longer for the missed packet. It will notify the user application of the fact that data was lost and continue to deliver the next packets. Whether or

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 402 of 830

not an application can tolerate packet loss will depend on the type of content. Audio and video will usually not be severely impacted by the loss of an occasional packet, while real-time stock quotes become dangerous to use when it is not known what quote was lost.

[00145] In one embodiment, where layered multicast is combined with localized packet retransmission, a service is realized that applies packet recovery only to those layers that can be delivered with the current bandwidth capacity. When the network has insufficient resources to deliver all layers of a stream, it is beneficial if the receivers know this and will not attempt to repair all missed packets from all layers. Instead, the receivers may use a mechanism that provides them feedback about the current network capacity and use that to decide for which layers it will recover missed packets. The algorithm monitors the status of the total stream and from this information derives how many layers of the stream can be reliably delivered to the user without stressing the network. It then marks these layers as being intact, reliable or similarly labeled and sends retransmission requests when an occasional packet is lost from these layers.

[00146] In one embodiment, the algorithm only delivers packets from the reliable layers to the application, but not before they have been put back into their original global order. In order for the algorithm to decide which layers are safely marked as intact, it requires some status information about the reception of the stream as a whole as well as the network conditions. To provide this, every packet contains the number of the most recent packet from all other layers or similar sequence data. Aside from carrying its own sequence number, each packet contains the

ıce number of all the other layers as well. By inspecting these sequence numbers each time a packet arrives, the receiver determines whether it has missed any packets from the layers it marked as reliable. If that is the case, these packets may be recovered. Each time a

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 403 of 830

missed packet is detected, a countdown timer is started for it. If the packet is recovered before the time-out expired, the timer is canceled and removed.

[00147] If however the timer manages to expire, the layer is considered impossible to repair and may be, together with all higher reliable layers (lower in priority), removed from the list of reliable layers. How long the time-out interval should be can be application or content dependent. The interval determines how long delay is tolerated by the user. Setting the interval to a low value means a temporary congestion may corrupt a layer long enough to cause the receivers to drop it. Since packets occasionally get lost when unicast routing tables converge, changing the shape of multicast distribution trees, or when a router crashes that had pending packets in its interface buffers, appropriate time-[[ ]]out values are adjustable by an administrator according to network conditions to be found experimentally.

[00148] In an [[a]] overlay multicast network that uses packet prioritization, routers explicitly introduce out of order delivery when a burst of packets queues in the outbound interceptor pipeline of a router interface with a slow connection. The time-[[ ]]out may be large enough to allow for this. Setting the time-[[ ]]out to a long period makes the stream much more resilient to congestion, but also increases the time for the receiver to discover that a layer must be dropped due to bandwidth constraints. Until congestion is finally detected, delivery of the previously received packets that causally depend on missed packets is postponed. Starting timers when packet loss is detected allows for congestion detection. However, it cannot be used to guarantee

d latency.

[00149] When no packets are lost and all are received according to their global order, the receiver cannot measure the total transmission delay. As such, a receiver cannot distinguish between an idle and a crashed source. In one embodiment, aside from removing layers from the

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 404 of 830

reliable-list, the algorithm is also able to detect when more bandwidth becomes available and new higher layers can be added to this list, so that a higher quality stream can be delivered to the user. This is done by passively or actively monitoring the layers that are not currently in the reliable-list and not under repair. Every time a packet is received from these layers, this fact is stored for a period that is equal to the repair time-out discussed earlier. When a packet from a layer is received with [[a]] sequence numbers that shows that a packet has been lost from a higher layer that is not in the reliable list, no retransmission request is sent, but the fact that this packet should have been received is recorded and stored for the same repair time-out period.

[00150] If during the time-out period the packet is received after all, possibly because it was delayed by a router, the stored record is marked as received. Because the state of each packet from every layer is recorded, the overlay multicast system builds a packet arrival history that is used to determine whether the reception quality was high enough to add a higher layer to the reliable-list and repair any further packet loss. In the overlay multicast system the reception history or similar data may be used to calculate a moving average that describes the amount of packet loss over the last x seconds, where *x* is equal to the repair time-out ~~timeout~~. With a small moving average history, the receiver will quickly respond to increased network capacity, while a longer history will only add layers when the network's ~~network 's~~ capacity was sufficiently high for a longer period of time. The receiver may keep track of the reception quality of each layer by inspecting the list of sequence number that is attached to each data packet.

iis causality information adds overhead to each packet, linearly related to the number of layers used in the stream. For example, when sequence numbers are 32 bits and the publisher uses all 256 available layers, each packet comes with a kilobyte of causality information, which contributes to at least 12.5% for an 8 Kb packet that is filled to the brim. To

ease layer administration in the receiver socket, an overlay multicast stream only supports a static number of layers. When a publisher binds a multicast group address for reliable, layered communication, it may explicitly specify the number of layers that will be used.

[00152] In one embodiment, by default, packets are delivered to the user application in their original order. Not only are the packets inside the individual layers restored to their natural order using their sequence numbers, but the total ordering across the layers may also be restored. When the source sends three packets with different priority (each in a different layer), all three are received by the user application in the exact same order. A prioritized, layered packet is described as $P_{1\,(9,4,6)}$ ~~$P_{1(19,4,6)}$~~ where 1 represents the packet's priority or layer, 9 the sequence number of the last packet of layer 0 (the highest priority layer) that was sent at the time this packet was published, 4 represents the sequence number of this packet and 6 represents the sequence number of the last packet of layer 2. Also, the packet tells that the stream uses 3 layers in total: layer 0 up to and including layer 2. It is concluded that this packet causally depends on packet 9 from layer 0 and packet 6 from layer 2 and will only be delivered to the user application after those packets have been delivered. In another embodiment, the overlay multicast system does not reorder the packets, but may leave this task to the application using the network.

[00153] **Figure 10** is a diagram that shows one embodiment of a process of selective packet repair and ordering. For purposes of explanation the notation {0 (9,2,6) ; 1(9,3,6), .. . } to describe a sequence of packets where $P_{0\,(9,2,6)}$ ~~$P_{0<9.2.6J}$~~ was sent prior to $P_{1\,(9,3,6)}$ ~~$P_{1<9.3.6J}$~~, is

of convenience. The first segment 1001, shows the sequence of packets originally published by the source. It reads from left to right, so $P_{0\,(9,2,6)}$ ~~$P_{0<9.2.6J}$~~ was published first, followed by $P_{1\,(9,3,6)}$ ~~$P_{1<9.3.6J}$~~, etc. In this example the source has published eight packets, divided over three priority levels or layers (0, 1 and 2, where 0 is [[0 is]] the lowest layer with

the highest priority). The second segment 1003 shows the packet stream as received by one of the subscribers. It shows two lost packets and an incorrect ordering. Before the packets are delivered to the user application, they are stored in internal buffers during the repair and reordering process. This state is depicted in segment 1005. Here the packets are in separate buffers, each representing a layer. The packets are ordered inside the buffers. The illustration shows the missing packets $P_{1\,(9,\,4,\,7)}$ and $P_{2\,(9,\,4,\,8)}$ ~~$P_{1(9,4,7)}$ and $P_{2(9,4,8)}$~~.

[00154] If the receiver currently only has layers 0 and 1 in its reliable-list, it will attempt to repair the hole in layer 1 by sending a retransmission request. Note that if the given sequence was just a snapshot of a running stream, the receiver would have detected the missing packet $P_{1\,(9,\,4,\,7)}$ when $P_{0\,(10,\,4,\,8)}$ ~~$P_{1(9,4,7)}$ when $P_{0(9,4,8)}$~~ was received, because this packet says it depends on packet #4 from layer 1. So even before $P_{1\,(10,\,5,\,9)}$ ~~$P_{1(10,5,9)}$~~ from layer 1 was received in our example, the receiver already detected loss in layer 1 and immediately scheduled a repair timeout for the missing packet and sent a retransmission request. In fact, if it is assumed that the receiver had received packet #2 from layer 1 prior to our snapshot of **Figure 10,** then the conclusion would be that packet #4 as well as packet #3 were lost.

[00155] In the example, shortly after $P_{0(10,-4,-8)}$ was received, packet $P_{1(9,-3,-6)}$ is received. The receiver places [[p l aces]] the delayed packet in the appropriate receiver buffer and cancels the repair time-out it started earlier when it detected that the packet was missing. This is illustrated in segment 1007. Note that the hole in layer 2 may also be detected when $P_{0\,(10,\,4,\,8)}$ ~~$P_{0(10,4,8)}$~~ is

at packet claims to be sent after packet #8 of layer 2 was published, so either packet #8 from layer 2 got lost in the network, or was delayed . However, since layer 2 is not in the reliable-list, a retransmission request is not sent. However, as for all packets, a timer is started for packet #8 of layer 2. When the user application is ready to read packets, the algorithm

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 407 of 830

returns only packets from layers that are in the reliable-list. Even though some of the layer 2 packets were received, they are discarded and not delivered. The resulting stream of packets that is delivered to the user is equal to the stream originally published by the source, with all layer 2 packets removed.

[00156] In this example, despite the fact that the network has insufficient capacity, having dropped packets from every layer and delivered the packets out of order, the user received a deterministic subset of the stream that is uncorrupted. Because the overlay multicast system sources are live data streams that cannot slow down or pause and because real-time data are not buffered by the source too long, receivers will only attempt to recover lost packets for a limited period of time. Whether a packet is received, recovered or lost, its buffer slot will be freed after this timeout, resulting in a natural size limit of the receive buffer. How large the buffer can get is related to the average speed of the stream and the length of the repair time-out.

[00157] In one embodiment, the algorithm implementation does not enforce a hard size limit during packet repairs. A more troublesome situation occurs when the user application does not read data from the socket fast enough. When this happens, the amount of pending data builds up in the socket and both storage requirements and the transmission delay increases. In one embodiment, this may be handled by removing a layer from the reliable-list, causing those packets to be discarded from the buffers, while decreasing the amount of data that is delivered to the application. If a clean feedback algorithm that can keep the number of lavers in balance with

ı's reading speed is not used, the overlay multicast system throws a fatal exception to the user or provides a similar indicator to the user application and closes the socket when the receive buffer reaches a certain maximum size, or when the total time between arrival of packets and their actual delivery reaches a threshold.

[00158] Although restoring the original global packet ordering before delivering the data to the user application is assumed to be appropriate for most types of data, there is content for which each packet invalidates all previous packets. This is the case for stock quotes and similar time sensitive data. For example, when a new stock quote update is received for a financial ticker symbol, it renders the previous updates useless. For most applications that process real-time financial data, only the most recent information is interesting. When global ordering is restored, the algorithm will postpone the delivery of the most recent data until all prior packets have been received also. For a typical application, such as a market data terminal, that merely displays quote updates to the screen, this postponing adds little value. When the burst of pending updates is finally delivered by the tsocket, the application updates the symbol's last value on the screen, thereby leaving only the last and most recent update visible and overwriting all pending updates immediately.

[00159] In one embodiment, because the reordering process adds additional delay to the data delivery, it can be switched off by applications that do not benefit from it. Without reordering, global (causal relations between packets from different layers) and local ordering (order of individual packets inside a single layer) is ignored and packets are delivered to the application immediately after they have been received. Disabling reordering has no impact on the reliability. Lost or delayed packets may still be recovered for all layers the reliable-[[ ]]list, only the recovered packets may be delivered with additional delay. Whether or not this makes them

[illegible] the application to decide.

[00160] For example, with market data it is useful to know whether an update is older or newer than the one previously received. This is because a stock quote may be overwritten by a newer one, but not the other way around. In this case, the source could add a logical time stamp

SCALE VIDEO CODING, LLC v. CISCO SYSTEMS, INC. **IPR2024-01371** **Ex. 1002, p. 409 of 830**

or similar sequence indicator to each stock quote, so the receiver can decide how to handle the update. In one embodiment, aside from configuring a layered multicast tsocket to restore global ordering or no ordering at all, a receiver can configure a tsocket to only restore local ordering. Whether or not this is useful will depend on the type of application and the content, but it offers the advantage that delivery of packets from lower (high priority) layers are not delayed during recovery of packets from higher, lower priority layers.

[00161] In one embodiment, an example market data publishing application tracks market financial updates in a linked list or similar data structure. Every time a new update is received, it is stored in the appropriate slot, overwriting the previous update. This way, the linked list always contains the most recent trade for every symbol. A virtual token traverses the list at a fixed speed. If the stream is to be thinned to one update per second, the token will visit one slot every second. When the token visits a slot, it removes the quote update and forwards it, leaving an empty slot. Empty slots will be skipped by the token without any delay. When one of the symbols has two updates per second in the incoming stream, the second incoming update after the last visit of the token will overwrite the update pending in the slot and the older pending update is dropped. This allows the leaky bucket algorithm to always provide the thinned clients with the most recent, live update of every symbol. While the virtual token visits the slots that have a recent quote update pending to be sent out, a thread receives the incoming market data stream and inserts new trades in the slots. To limit the outgoing transmission rate to one quote

token thread sleeps for one second after it moved and sent an update from a slot. When the token is able to complete one full circle between two incoming updates of every symbol, there will be no data loss. Instead of limiting the outgoing bandwidth to a fixed maximum, the dequeuing thread could also be configured to run as fast as possible, relying on

external feedback such as a blocking write call from the network, to reduce publishing speed. This could be particularly useful when the algorithm is used on the server side of a point-to-point TCP connection to a client application. In one embodiment, the server uses the algorithm to send the data at original speed, until network saturation slows down the links. A potential problem with this technique is that the original order of the updates is lost. In fact, when every symbol updates more than once while the token circles the list, the token will find every slot filled with a quote update, causing the order of the output stream to match the fixed order of the slots in the linked list. For quote updates from different symbols, this is usually not a problem, as they are not related to each other.

[00162] In another embodiment, an alternative algorithm may be used that provides another way of thinning a stream of data, such as live market data, with varying bandwidth to a fixed maximum rate, without the risk of providing stale data, for example stale market symbols. This algorithm may be used to produce enhancement layers that form a layered data stream, such as a layered market data stream. One example method of doing this is to have several instances of the algorithm running that each receive a copy of the input stream, letting their token run at different speeds and tagging their updates with a layer number. The layers may then be combined into a single stream. Unfortunately, this may introduce redundant information *in* the stream because the higher layers contain updates that are also available *in* the lower layers. This technique of encoding an entire stream in different qualities and sending them to clients concurrently to meet [illegible] bandwidth capacities may be referred to as simulcasting.

[00163] In one embodiment, the thinning algorithm may be modified to have each layer contain only information that is not already present in the lower layers, thereby eliminating redundancy among the layers. To achieve this the leaky bucket algorithm may be extended to a

SCALE VIDEO CODING, LLC v. CISCO SYSTEMS, INC. **IPR2024-01371** **Ex. 1002, p. 411 of 830**

multi-level ring as illustrated in **Figure 11B ,** creating a virtual cylinder of vertically stacked, circular lists, where each ring represents a layer of the output stream. Each ring has its own token. In one embodiment, the tokens all circulate their ring at the same speed. In other embodiments, the tokens in each ring may be set to circulate at different speeds. This data structure also identifies columns. A column is the collection of vertically stacked slots that can all store an update for the same symbol.

[00164] **Figure 11A** is a diagram of an example method of thinning a stream of live market data. The gray slots represent symbols for which a recent quote update was received, while the transparent slots are empty. **Figure 11B** is a diagram illustrating the token ring data structure. Multiple instances of the thinning algorithm and data structure may be conceived as being vertically stacked, each instance represents a stream layer and each symbol has a column that can store more than one update. **Figure 11C** is a diagram of one embodiment of a representation of the stacked thinning algorithm of **Figure 11A** at work. In the example, the gray slots contain a pending quote update, while the transparent slots are empty. The illustration shows how the updates of a volatile symbol are spread out over several higher layers.

[00165] In one embodiment, the thinning algorithm has a single input thread that receives the original stream. Newly received updates may be inserted in the lowest ring. If a new update arrives for a slot in the lowest ring that already contains an update for that symbol, this older update is vertically shifted into the next ring, so that the lower slot can always store the latest an update is shifted upwards, it cannot return to a lower ring anymore, even if all other slots beneath it were emptied.

[00166] In one example embodiment, symbols that have more than one trade during the time it takes the token to complete a full rotation of the ring will be spread out to two or more higher

layers, while symbols that have very infrequent trades will only occupy a single slot in the bottom ring. An example result of this process is depicted in **Figure 11C. Figure 11C** shows the thinning algorithm using three layers to thin a market data stream that contains updates for eight symbols. Four symbols are included in data structure and assigned to the four visible columns in the front. The figure also shows that all four symbols have pending updates in the bottom ring, while the 'MSFT' symbol is so volatile that it shifted updates into all three layers before the tokens could dequeue any of the pending updates.

[00167] In a case where a symbol is so actively traded that it manages to shift an update out of the top ring, that update is irrecoverably lost. If required, loss of updates can be avoided by letting the token in the top ring run without dequeuing delay, letting it empty every slot immediately after an update was queued. The result is that the transmission rate of the upper layer will equal the transmission rate of the original stream, minus the combined rate of all lower layers. Because the transmission rate of the lower layers may have a fixed maximum, the upper layer's rate may fluctuate and follow the variations of the original stream.

[00168] In one embodiment, if the tokens all rotate at the same speed, this does not imply that they visit the same symbol slots at the same time. Instead, they will usually be at different positions in their ring. This is a result of empty slots that are skipped without waiting.

[00169] In one embodiment, the[[ ]] layered thinning algorithm has out of order transmission and may cause additional delay. Every ring can contain an update for the same symbol at a certain time and because the tokens can have different column locations inside their ring, it is undefined which ring will dequeue and transmit its update first, which will be second and so on. This means that not only the order of symbols is shuffled, but even the updates for a single symbol are no longer guaranteed to be transmitted chronologically.

[00170] While new updates for a symbol are queued, shifted and sent out, in some cases an update can survive in the rings substantially longer than its younger successors. This can occur when an update that is just about to be dequeued by the token, gets shifted to the next layer by a more recent update that entered through the lower ring. In this case it is likely that the more recent update is dequeued sooner than the older update that now has to wait for the token of its new ring to reach its slot. To address this issue, the system provides a way of putting the outgoing, layered updates back in the original, chronological order. This is done by labeling incoming updates with sequential numbers and having the tokens pass their dequeued, labeled updates to a transmit window that uses the sequence numbers to restore the chronological order. When updates are shifted out of the upper ring and therefore considered lost, the transmit window is notified not to wait for their sequence numbers.

[00171] When applying order reconstruction across all rings, the transmit window must have the same capacity as all rings combined, $size_{tw} = N \cdot j$, where $size_{tw}$ is the maximum capacity of the transmit window, N the number of symbols in the original stream (and therefore the number of slots per ring) and j the number of layers used. The potential size of the transmit window, together with its bursty behavior when a delayed quote update releases a substantial amount of pending updates from the window, may justify a mechanism that applies order reconstruction to achieve chronologic transmission per symbol only, similar to the output of the base layer encoder.

ke the algorithm depicted in Figure 11A, the layered system introduces delay. The model is essentially a collection of instances of the first algorithm, thus, its worst case delay is the sum of each ring's worst case delay:

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 414 of 830

$$t_{max} = \sum_{i=1}^{j} \frac{N}{f_{update}(i)}$$

[00173] where tmax is the maximum delay in seconds that the algorithm could introduce for a single quote update, *N* the number of symbols in the original stream, *j* the number of used layers and $f_{update}(i)$ the update frequency of the token in layer *i*.

[00174] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will however, be evident that various modification and changes can be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded[[ing]] in an illustrative rather than a restrictive sense.

SCALE VIDEO CODING, LLC
v. CISCO SYSTEMS, INC.
IPR2024-01371
Ex. 1002, p. 415 of 830

Electronic Acknowledgement Receipt	
EFS ID:	25838329
Application Number:	15010573
International Application Number:	
Confirmation Number:	9731
Title of Invention:	LAYERED MULTICAST AND FAIR BANDWIDTH ALLOCATION AND PACKET PRIORITIZATION
First Named Inventor/Applicant Name:	Erik Van Zijst
Customer Number:	122200
Filer:	Stephen Y. Liu/Carolyn Bridwell
Filer Authorized By:	Stephen Y. Liu
Attorney Docket Number:	GPB1-US9
Receipt Date:	20-MAY-2016
Filing Date:	29-JAN-2016
Time Stamp:	13:54:51
Application Type:	Utility under 35 USC 111(a)

ion:

nt	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Preliminary Amendment	GPB1US9_PreliminaryAmendment.pdf	86904 ff9ed93dc9aac9cf5d9fff9a1fb0318744a6428	no	4
Warnings:					
Information:					

2	Preliminary Amendment	GPB1US9_Substitute_Specification_CLEAN.pdf	286720 9c6c3053da13474bc22ea7d07a64599ea983b0ac	no	67
Warnings:					
Information:					
3	Preliminary Amendment	GPB1US9_Substitute_Specification_MARKED.pdf	293656 088f1fa75e2160ca5447f3626ea49b1942bedb5d	no	66
Warnings:					
Information:					
		Total Files Size (in bytes):	667280		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/010,573	01/29/2016	Erik Van Zijst	GPB1-US9

CONFIRMATION NO. 9731

122200
Carstens & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380

PUBLICATION NOTICE

OC000000083215583

Title:LAYERED MULTICAST AND FAIR BANDWIDTH ALLOCATION AND PACKET PRIORITIZATION

Publication No.US-2016-0150256-A1
Publication Date:05/26/2016

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Office of Public Records. The Office of Public Records can be reached by telephone at (703) 308-9726 or (800) 972-6382, by facsimile at (703) 305-8759, by mail addressed to the United States Patent and Trademark Office, Office of Public Records, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently http://pair.uspto.gov/. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

ectronically accessing the publication, or about › by calling the Patent
; Center at 1-866-217-9197.

Office of Data Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/010,573	01/29/2016	Erik Van Zijst	GPB1-US9	9731

122200 7590 06/16/2016

Carstens & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380

EXAMINER	
HONG, MICHAEL HYUN	

ART UNIT	PAPER NUMBER
2426	

NOTIFICATION DATE	DELIVERY MODE
06/16/2016	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

panoptis@cclaw.com
docketing@panoptis.com

PTOL-90A (Rev. 04/07)

Office Action Summary	Application No. 15/010,573	Applicant(s) VAN ZIJST, ERIK	
	Examiner MICHAEL HONG	Art Unit 2426	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *1/29/16*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5)☒ Claim(s) *1-14* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1-14* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

ıpies of the priority documents have been receiı No. _____.

of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

The present application is being examined under the pre-AIA first to invent provisions.

DETAILED ACTION

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of pre-AIA 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1,2, 4, 8, 9, 11 are rejected under pre-AIA 35 U.S.C. 102b as being anticipated by Krishnamachari (US 2003/0072376).

Regarding claim 1, 8, Krishnamachari discloses A video signal distribution system comprising:

a source endpoint, wherein the source endpoint scalably encodes video signals into layers to form a layered data stream, wherein the layered data stream comprises a base layer and a set of enhancement layers ([0017]);

a scalable video coding router, wherein the scalable video coding router forwards om the source endpoint to a video rece and wherein the scalable video coding router selectively forwards one or more of the set of enhancement layers based upon information received by the scalable video coding router ([0018, 0019, 0021]); and

wherein the layered data stream is transmitted via a communications channel configured for a selected network protocol ([0021]).

Regarding claim 2, 9 Krishnamachari discloses wherein the information received by the scalable video coding router relates to throughput in a portion of the communications channel linking the scalable video coding router and the video receiver endpoint ([0019, 0020]).

Regarding claim 4, 11, Krishnamachari discloses wherein the scalable video coding router divides available bandwidth among a plurality of data streams, and wherein the scalable video coding router prioritizes forwarding of the set of enhancement layers ([0018-0021]).

Claim Rejections - 35 USC § 103

The following is a quotation of pre-AIA 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

s 3,10 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Krishnamachari (US 2003/0072376) in view of Parkkinen (US 2003/0206558).

Regarding claim 3, 10, Krishnamachari does not specifically disclose wherein the information received by the scalable video coding router is directly related to video processing capabilities of the video receiver endpoint.

However, Parkkinen discloses wherein the information received by the scalable video coding router is directly related to video processing capabilities of the video receiver endpoint ([0074]). It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the processing capability considerations of Parkkinen into the system of Krishnamachari in order to transmit streams that are capable of being processed by the receiver end.

Claim 5-7, 12-14 is rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over Krishnamachari (US 2003/0072376) in view of Li (US 2004/0071083).

Regarding claim 5, 12, Krishnamachari does not specifically disclose wherein each layer of the layered data stream comprises data packets, each of which is encoded with a sequence number and a priority identifier, and wherein the sequence number and the priority identifier for each data packet is based upon a layer to which

...igs.

However, Li discloses wherein each layer of the layered data stream comprises data packets, each of which is encoded with a sequence number and a priority identifier, and wherein the sequence number and the priority identifier for each

data packet is based upon a layer to which the packet belongs ([0027, 0028, 0030]). It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the packet identifiers of Li into the system of Krishna in order to assign specific packets to various layers in the encoding of data.

Regarding claim 6, 13, Krishnamachari in view of Li disclose wherein each layer of the layered data stream comprises data packets, each of which is encoded with a sequence number and a priority identifier, wherein the sequence number and the priority identifier for each data packet is based upon a layer to which the packet belongs, wherein the information received by the scalable video coding router is related to throughput in a portion of the communication channel linking the scalable video coding router and the video receiver endpoint, and wherein the information includes a request for retransmission of one or more packets from the layered data stream ([0027, 0028, 0030] of Li).

Regarding claim 7, 14, Krishnamachari in view of Li discloses wherein the selected network protocol is TCP/IP ([0049] of Li).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MICHAEL HONG whose telephone number is (571)270-1553. The examiner can normally be reached on M-F (10:30a-6:30p).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nasser Goodarzi can be reached on (571)272-4195. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/MICHAEL HONG/
Primary Examiner, Art Unit 2426

Notice of References Cited	Application/Control No. 15/010,573	Applicant(s)/Patent Under Reexamination VAN ZIJST, ERIK	
	Examiner MICHAEL HONG	Art Unit 2426	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-2003/0072376 A1	04-2003	Krishnamachari, Santhana	H04N21/234327	375/240.26
*	B	US-2003/0206558 A1	11-2003	Parkkinen, Teemu	G10L19/24	370/477
*	C	US-2004/0071083 A1	04-2004	Li, Qiong	H04N21/222	370/230
	D	US-				
	E	US-				
	F	US-				
	G	US-				
	H	US-				
	I	US-				
	J	US-				
	K	US-				
	L	US-				
	M	US-				

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	15010573	VAN ZIJST, ERIK
	Examiner	**Art Unit**
	MIKE HONG	2426

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	06/10/2016								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								
	9	✓								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (03-15)
Approved for use through 07/31/2016. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Not for submission under 37 CFR 1.99)

Application Number	15010573
Filing Date	2016-01-29
First Named Inventor	Van Zijst, Erik
Art Unit	2425
Examiner Name	Not yet assigned
Attorney Docket Number	GPB1-US9

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	6078564		2000-06-20	Lakshman	
	2	6650652		2003-11-18	Valencia	
	3	7370129		2008-05-06	Green	
	4	7626975		2009-12-01	Colban	
	5	7729719		2010-06-01	Bergstrom	
	6	7801142		2010-09-21	Jiang	
	7	7860120		2010-12-28	Wang	
	8	8224237		2012-07-17	Smith	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2425
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	9	7590131		2009-09-15	Ansorge	
	10	7746841		2010-06-29	Meiling	
	11	8514718		2013-08-20	Van Zijst	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20020071421		2002-06-13	Chiu	
	2	20130114594		2013-05-09	Van Zijst	
	3	20080126562		2008-05-29	Ludwig	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2425
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1		

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature	/Hyun Hong/	Date Considered	06/12/2016

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2425
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☒ See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Stephen Y. Liu/	Date (YYYY-MM-DD)	2016-05-02
Name/Print	Stephen Y. Liu	Registration Number	62883

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed . Time will vary depending upon the individual ents on the amount of time you and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

system of records may be disclosed, as a rout... ...blic after either publication of rsuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L3	343	@ad<"20050126" and (enhancement$1 adj layer$1) same bandwidth	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/12 11:48
L4	30	@ad<"20050126" and (enhancement$1 adj layer$1) same bandwidth same priority	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/12 11:48
L5	68	@ad<"20050126" and (enhancement$1 adj layer$1) same bandwidth same (scaling or scalable or scale) and priority	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/12 11:50
L6	15	@ad<"20050126" and (enhancement$1 adj layer$1) same bandwidth same (scaling or scalable or scale) same priority	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/12 11:51
L7	21	@ad<"20050126" and (enhancement$1 adj layer$1) same scale$1 same (processing)	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/12 12:25
L8	0	@ad<"20050126" and (enhancement$1 adj layer$1) same scale$1 same (receiver with power)	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/12 12:28
L9	0	@ad<"20050126" and (enhancement$1 adj layer$1) same scale$1 same (receiver)	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/12 12:28
L10	27	@ad<"20050126" and (enhancement$1 adj layer$1) same scale$1 same (display)	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/12 12:28
L11	71	@ad<"20050126" and (enhancement$1 adj layer$1) same bandwidth same packet$1	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/12 12:29
L12	7	@ad<"20050126" and (enhancement$1 adj layer$1) same bandwidth same packet$1 same priority	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/12 12:30
L14	2	@ad<"20050126" and (enhancement$1 adj layer$1) same bandwidth same packet$1 same priority and (retransmit or retransmission)	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/12 12:31
		20040198371-$ or US-20020006161-$ or US-20030072376-$ or US-20030206558-$ or US-20060164987-$).did.	US-PGPUB		FF	2016/06/12 12:33
L16	1	15 and tcp	US-PGPUB	OR	OFF	2016/06/12 12:33
S1	1053	(scalable with router)	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/03 15:09
S2	763	S1 and protocol	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/03 15:57

S3	0	WO-04010250-$.did.	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/09 16:34
S4	2	WO-2004010250-$.did.	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/09 16:34
S5	0	@ad<"20050126" and router same (enhancement$1 adj layer$1) same bandwidth same (scaling or scaleable or scale)	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/09 17:13
S6	4	@ad<"20050126" and router same (enhancement$1 adj layer$1) same (scaling or scaleable or scale)	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/09 17:20
S7	30	@ad<"20050126" and (enhancement$1 adj layer$1) same bandwidth same (scaling or scaleable or scale)	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/10 14:58
S8	11	S7 and priority	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/10 15:10
S9	3	S8 and TCP	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/10 15:10
S10	3	S8 and TCP$3	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/10 15:10
S11	0	@ad<"20050126" and h04n21/2662.cpc	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/10 15:15
S12	842	@ad<"20050126" and h04n21/2662.cpc.	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/10 15:15
S13	65	@ad<"20050126" and h04n21/234.cpc.	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2016/06/10 15:16

EAST Search History (Interference)

<This search history is empty>

6/12/2016 12:37:32 PM
C:\Users\hhong1\Documents\EAST\Workspaces\15010573.wsp

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 9731

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
15/010,573	01/29/2016 RULE	725	2426	GPB1-US9

APPLICANTS
Blitz Stream Video, LLC, Plano, TX;

INVENTORS
Erik Van Zijst, Belmont, CA;

**** CONTINUING DATA ***************************
This application is a DIV of 13/619,969 09/14/2012
which is a CON of 12/486,656 06/17/2009 PAT 8514718
which is a DIV of 11/342,167 01/26/2006 PAT 7733868
which claims benefit of 60/647,601 01/26/2005

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED **** ** SMALL ENTITY **
02/12/2016

Foreign Priority claimed	35 USC 119(a-d) conditions met	Verified and Acknowledged	Met after Allowance	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
☐ Yes ☑ No	☐ Yes ☑ No	/MIKE HYUN HONG/ Examiner's Signature	☐ Initials	CA	12	14	2

ADDRESS
Carstens & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
UNITED STATES

TITLE
LAYERED MULTICAST AND FAIR BANDWIDTH ALLOCATION AND PACKET PRIORITIZATION

FILING FEE RECEIVED		
800	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) _____ ☐ Credit

BIB (Rev. 05/07).

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Not for submission under 37 CFR 1.99)

Application Number	15010573
Filing Date	2016-01-29
First Named Inventor	Van Zijst, Erik
Art Unit	2465
Examiner Name	Not yet assigned
Attorney Docket Number	GPB1-US9

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	4799215		1989-01-17	Suzuki	
	2	5095480		1992-03-10	Fenner	
	3	5262906		1993-11-16	Mazzola	
	4	5331637		1994-07-19	Francis	
	5	5596574		1997-01-21	Perlman	
	6	5842224		1998-11-24	Fenner	
	7	5860136		1999-01-12	Fenner	
	8	5918074		1999-06-29	Wright	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	9	6222841		2001-04-24	Taniguchi	
	10	6330238		2001-12-11	Ooe	
	11	6347090		2002-01-12	Ooms	
	12	6496477		2002-12-17	Perkins	
	13	6577609		2003-06-10	Sharony	
	14	6614900		2003-09-02	Champoux	
	15	6691192		2004-02-10	Ajanovic	
	16	6757294		2004-06-29	Maruyama	
				2005-01-18	Olsson	
	18	6904015		2005-06-07	Chen	
	19	6907037		2005-06-14	Tsuchiya	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	20	6917984		2005-07-12	Tan	
	21	7076569		2006-07-11	Bailey	
	22	7085691		2006-08-01	Riess	
	23	7542417		2009-06-02	Babonneau	
	24	7733868		2010-06-08	Van Zijst	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20010014103		2001-08-16	Burns	
	2	20010048662		2001-12-06	Suzuki	
	3	20020009067		2002-01-24	Sachs	
	4	20020012667		2002-01-31	McMichael	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	5	20020028687		2002-03-07	Sato	
	6	20020114282		2002-08-22	MeLampy	
	7	20020120769		2002-08-29	Ammitzboell	
	8	20020118692		2002-08-29	Oberman	
	9	20020126671		2002-09-12	Ellis	
	10	20020143951		2002-10-03	Khan	
	11	20020176406		2002-11-28	Tsukada	
	12	20020191631		2002-12-19	Couty	
		6Z6Z68		2003-02-06	Navas	
	14	20030035424		2003-02-20	Abdollahi	
	15	20030035425		2003-02-20	Abdollahi	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	16	20030037132		2003-02-20	Abdollahi	
	17	20030041171		2003-02-27	Kobayashi	
	18	20030046425		2003-03-06	Lee	
	19	20030123479		2003-07-03	Lee	
	20	20030123438		2003-07-03	Li	
	21	20030123453		2003-07-03	Ooghe	
	22	20030133406		2003-07-17	Fawaz	
	23	20030161311		2003-08-28	Hiironniemi	
		:025 13		2003-10-30	Chen	
	25	20030212816		2003-11-13	Bender	
	26	20030233464		2003-12-18	Walpole	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	27	20040010616		2004-01-15	McCanne	
	28	20040015591		2004-01-22	Wang	
	29	20040052251		2004-03-18	Mehrotra	
	30	20040044791		2004-03-04	Pouzzner	
	31	20040081197		2004-04-29	Liu	
	32	20040133631		2004-07-08	Hagen	
	33	20040133697		2004-07-08	Mamaghani	
	34	20040165536		2004-08-26	Xu	
		:02164		2004-10-14	Hooper	
	36	20040258094		2004-12-23	Bashan	
	37	20040258002		2004-12-23	Tran	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	38	20050002363		2005-01-06	Cheng	
	39	20050102414		2005-05-12	Hares	
	40	20050190693		2005-09-01	Jinzaki	
	41	20060015624		2006-01-19	Smith	
	42	20060085828		2006-04-20	Dureau	
	43	20060164987		2006-07-27	Ruiz Floriach	
	44	20070121507		2007-05-31	Manzalini	
	45	20070206592		2007-09-06	Itakura	
		:09494		2008-08-28	Dravida	
	47	20100198913		2010-08-05	Garcia-Luna-Aceves	
	48	20130114443		2013-05-09	Van Zijst	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1	2004010250	WO		2004-01-29	Phillips Electronics		
	2	2006081454	WO		2006-08-03	Internet Broadcasting Company		

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	ADAMSON et al., "Negative-Acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Building Blocks", RFC-3941, November 2004, Sections (2.7 on page 7, 3.9 through 3.11 inclusive pp. 31-32), Internet Engineering Task Force of the Internet Society, Reston, VA, USA	
	2	ALLMAN et al., "TCP Congestion Control", RFC-2581, April 1999, Sections (1 through 4 on pages 1-10 inclusive, 6 on page 10), Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
	3	AVTECHNOLOGY, "Scalable Video Coding Demystified", AVTECHNOLOGY, http://www.avtechnologyonline.com/article/41538.aspx, printed March 27, 2012, 2 pages	
		U et al., "AIRMAIL: A link layer protocol for wireless networks", August 1994, Sections (1 pp. 4 7-49, 2.1 pp. 49-50, 2.2 pp. 50-51), Wireless Networks, Volume 1, Issue 1, pp. 47-60, Kluwer Academic Publishers	
	5	BALAKRISHNAN et al., "A comparison of mechanisms for improving TCP performance over wireless links", Dec 1997, Sections II and III inclusive of subsections on pp. 757-760, I EEE/ACM Transactions on Networking, vol.5, no.6, pp. 756-769, 2001 L Street NW. Suite 700, Washington, DC 20036-4910 USA.	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	6	BALAKRISHNAN et al., "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks", December 1995, Section 3 inclusive of subsections pp. 3-7, ACM Wireless Networks Journal (WI NET- 1995), Volume 1, Issue 4, Association for Computing Machinery Inc., Stanford, CA, USA.	
	7	BOLOT et al., "Scalable Feedback Control for Multicast Video Distribution in the Internet", August 1994, Sections (Abstract on pg. 58, Our contributions on pg. 59,2 pp. 59-60,4.2 pp. 63-64), Proceedings of ACM SIGCOMM 94, pp. 58-67, London, England, UK.	
	8	BRADEN et al., "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC-2309, April 1998, Sections 1 through 5 pp. 2-11 inclusive, Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
	9	BYERS et al., "A Digital Fountain Approach to Reliable Distribution of Bulk Data", SIGCOMM '98 Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication, pp. 56-67, Sections (3 pp. 4-5 of 22, 7 through 7.3 inclusive pp. 15-17 of 22), Association for Computing Machinery, New York, NY, USA	
	10	BYERS, J., et al., "FLID-DL: Congestion Control for Layered Multicast", In Proceedings NGC 2000, November 2000, 11 pages	
	11	CHAWATHE, "Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service", Dissertation for the Degree of Philosophy, Fall 2000, 161 pages, Berkeley, CA, USA	
	12	CHENG, C. et al., "A Loop-Free Extended Bellman-Ford Routing Protocol Without Councing Effect", ACM 089791-332-9/89/0009/0224, pg. 227, 228 section 3.2	
	13	CHEN-KHONG THAM et al., "CONGESTION ADAPTATION AND LAYER PRIORITIZATION IN A MULTICAST SCALABLE VIDEO DELIVERY SYSTEM", IFIP/IEEE International Conference on Management of Multimedia Networks and Services, 2002, Sections (1 through 3.2 inclusive pp. 1-6, 4.2 through 4.3 inclusive of sub-sections pp. 9-10), IEEE Communications Society, 10662 Los Vaqueros Circle, Los Alamitos, California 90720-1314 USA.	
		al., "An Adaptive Totally Ordered Multicast Protocol tions", Pages 237-246, ACM ute of Computer Science, The Hebrew University of Jerusalem, Israel, May (1998) Technical Report CS	
	15	CISCO, "Cisco Unified Videoconferencing 5100 & 5200 Series Products Cisco Unified Videoconferencing 7.0, High-Definition Video Conferencing Solution for Cisco Unified Communications", Cisco Data Sheet, Cisco Systems, Inc., 2009, 6 pages	
	16	CISCO, "Understanding Selective Packet Discard (SPD)", Document ID: 29920, Cisco Systems, Inc., July 7, 2005, 6 pages. ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./	

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	17	DAH MING CHIU, et al., "TRAM: Tree-based Reliable Multicast Protocol", SML Technical Report Series, Pages 24, Sun Microsystems, Inc. SML TR-98-66, Chelmsford, MA, USA.	
	18	FEAMSTER et al., "On the Interactions Between Layered Quality Adaptation and Congestion Control for Streaming Video", April 2001, Sections (1 through 2 inclusive pp. 1-2, 3.2 on pg. 3, 3.2.2 on pg. 4, 4 inclusive of subsections pp. 4-8), 11th International Packet Video Workshop, Kyongju, Korea.	
	19	FIDLER, "Differentiated Services Based Priority Dropping and Its Application to Layered Video Streams", Networking 2002, pp. 551-562, Sections 1 through 4 inclusive pp. 551-561), Springer-Verlag London, Springer Science Business Media UK Ltd., 236 Gray's Inn Road, Floor 6, London WC1X 8HB, United Kingdom.	
	20	FLOYD et al., "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management", August 2001, Section 4 inclusive of subsections pp. 5-8, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, USA.	
	21	FLOYD et al., "Link-sharing and Resource Management Models for Packet Networks", August 1995, Sections (5-6 pp. 8-14, 7 through 8 pp. 15-16), IEEE/ACM Transactions on Networking (1995), Vol. 3, No.4, 2001 L Street NW. Suite 700, Washington, DC 20036-4910 USA.	
	22	FLOYD et al., "Router Mechanisms to Support End-to-End Congestion Control", February 1997, Sections (Abstract on pg. 1, 3 pp. 4-7, 5 through 5.2 inclusive pp. 1 0-11, 7 pp. 12-13), Technical report, ftp://ftp.ee.lbl.gov/papers/collapse.ps.	
	23	GORINSKY, et al., "Addressing Heterogeneity and Scalability in Layered Multicast Congestion Control", Technical Report TR200-31, November 24, 2000, Pages 1-12, Austin, TX, USA.	
	24	GORINSKY, et al., "The Utility of Feedback in Layered Multicast Congestion Control", NOSSDAV'01, June 25-26, 2001, Port Jefferson, New York, USA, Pages 93-102.	
		New Terminology and Clarifications for Diffserv", RFI . . Sections (2-3 pp. 1-3,4pp. 3-4, Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
	26	HANDLEY et al., "The Reliable Multicast Design Space for Bulk Data Transfer", RFC-2887, August 2000, Section (2.1 through 2.5 pp. 3-6,3.3 pp. 7-8, 4.1.1 through 4.2.1 inclusive pp. 9-12,5 pp. 14-16), Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
	27	HARJU et al., "Distributing Layered Encoded Video through Caches", IEEE Transactions on Computers, Vol. 51, No. 6, June 2002, Sections (2 through 2.3 inclusive pp. 623-625, 3.1 on pg. 626, 4 on pg. 629), IEEE Communications Society, 10662 Los Vaqueros Circle, Los Alamitos, California 90720-1314 USA.	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	28	INTERNATIONAL TELECOMMUNICATION UNION, "An architecture for end-to-end QoS control and signalling", ITU-T Recommendation H.360, March 2004, Sections (1 on pg. 1, 5 through 5.1.2 inclusive on pg. 3, 5.3 inclusive of sub-sections pp. 4-6), International Telecommunication Union (ITU), Place des Nations, 1211 Geneva 20, Switzerland.	
	29	INTERNATIONAL TELECOMMUNICATION UNION, "Packet-based multimedia communications systems", ITU-T Recommendation H.323, July 2003, Sections (6.2.2 through 6.2.11 inclusive of sub-sections pp. 15-28, 10.3 inclusive of sub-sections pp. 127-130, 8.5 through 8.8 inclusive of sub-sections pp. 141-146), International Telecommunication Union (ITU), Place des Nations, 1211 Geneva 20, Switzerland.	
	30	INTERNATIONAL TELECOMMUNICATION UNION. "Call signalling protocols and media stream packetization for packet-based multimedia communication systems". ITU-T Recommendation H.225.0. July 2003, Sections (1 pp. 1-3, 6 inclusive of sub-section pp. 8-14, 7.6 pp. 45-56, 7.12.1 through 7.12.3 pp. 72-74, 7.15.1 through 7.15.4 pp. 80-83, 8 inclusive of sub-sections pp. 89-92, Annex A on pg. 92), International Telecommunication Union (ITU), Place des Nations, 1211 Geneva 20, Switzerland.	
	31	KIKUCHI et al., "RTP Paylod Format for MPEG-4 AudioNisual Streams", RFC-3016, November 2000, Sections (1.2 pp. 3-4, 4 inclusive of sub-sections pp. 1 0-12), Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
	32	MASON et al., "MarketScope for Telepresence and Group Video Systems", Research Note G00215756, Gartner, http://www.gartner.com/technology/streamReprints.do?id=1-174ZY5B&ct=110831&st=sb, August 30, 2011, 6 pages	
	33	MCCANNE et al., "Receiver-driven Layered Multicast", August 1996, Sections (1 through 2 pp. 1-3, 3.3 pp. 4-5, 4 on page 7, 5 through 6 pp. 10-11, 8 on page 12), ACM SIGCOMM (1996), Association for Computing Machinery Inc., Stanford, CA, USA.	
	34	NICHOLS et al., "Definition of the Differentiated Services Field (DS Field) in the 1Pv4 and 1Pv6 Headers", RFC-2474, December 1998, Sections (Abstract pp. 1-2, 1 through 3 pp. 3-9 inclusive, 4.2 through 5 and subsections pp. 10-14 inclusive), Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
	35	ONG et al., "An Introduction to the Stream Control Transmission Protocol (SCTP)", RFC-3286, May 2002, Sections (3 pp. 2-3, 6 on pg. 5, 9.1 on pg. 7), Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
		OS et al., "An Error Control Scheme for Large-Scale ıns", IEEE INFOCOM '98 ıs, pp. 1188-1196 Vol. 3, March 1998, Sections (3 through 4.3 inclusive pp. 1189-1192), IEEE Computer ...662 Los Vaqueros Circle, Los Alamitos, California 90720-1314 USA.	
	37	PARSA et al., "TULIP: A Link-Level Protocol for Improving TCP over Wireless Links", September 1999, Sections (II through III. B inclusive of subsections pp. 1-3 (1253-1255)), Wireless Communications and Networking Conference (1999) pp. 1253-1257, 2001 L Street NW. Suite 700, Washington, DC 20036-4910 USA.	
	38	PARTRIDGE et al., "Version 2 of the Reliable Data Protocol (RDP)", RFC-1151, April 1990, pp. 1-3, Internet Engineering Task Force of the Internet Society, Reston, VA, USA. ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./	

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	39	RAMAKRISHNAN et al., "The Addition of Explicit Congestion Notification (ECN) to IP", RFC-3168, September 2001, Sections (4 on page 6, 5 pp. 6-10, 13 on page 40, 19.1 pp. 50-52), Internet Engineering Task Force of the Internet Society, Reston, VA, USA	
	40	REJAIE et al., "Layered Quality Adaptation for Internet Video Streaming", December 2000, Sections (Abstract on pg. 2530, I. C and D through II. exclusive of subsections pp. 2531-2534, II. Con page 2535, II. Eon page 2536, III. Bon page 2537, VII pp. 2542-2543), IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 18, NO.12, pp. 2530-2543, 2001 L Street NW. Suite 700, Washington, DC 20036-4910 USA.	
	41	REY at al., "RTP Retransmission Payload Format", Internet Draft draft-ietf-avt-rtp-retransmission-10.txt, January 2004, Sections (Abstract through 7 inclusive pp. 1-13, 10.3 pg. 23), Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
	42	RIZZO et al., "The RLC multicast congestion control algorithm", 1998, Sections (II preface portion on pg. 2, II. B and Con pg. 3, III through IIIA3 inclusive pp. 3-5, III.B through III.B.2 inclusive pp. 5-6, V.A on pg. 9), Dipartimento di Ingegneria dell'Informazione of the Universita di Pisa, Italy.	
	43	SCHULZRINNE et al., "RTP: A Transport Protocol for Real-Time Applications", RFC-3550, July 2003, Sections (Abstract on pg. 1, Section 1. pp. 5-6, Sections 2.2 through 2.4 pp. 7-8, definition of SSRC and CSRC pp. 9-10, Section 5.1 pp. 12-14, Section 7.4 on pg 49, 8.3 on page 54, 11 pp. 56-57), Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
	44	SCHWARZ, Heiko et al., "Overview of the Scalable Video Coding Extension of the H.264/A VC Standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No.9, September 2007, pp. 1103-1120.	
	45	SPEAKMAN et al., "PGM Reliable Transport Protocol Specification", NWG RFC: 3208, December 2001, Cisco Systems, 112 pages	
	46	SPEER et al., "RTP usage with Layered Multimedia Streams", Internet Engineering Task Force draft-speer-av1-layered-video-02.txt, Dec 20, 1996, Abstract through Section 3 inclusive pp. 1-3, Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
		P Slow Start, Congestion Avoidance, Fast Retransm ry Algorithms", RFC-2001, 97, Sections 1 through 4 pp. 1-5 inclusive, Internet Engineering Task Force of the Internet Society, Reston,	
	48	STEWART et al., "Stream Control Transmission Protocol", RFC-2960, October 2000, Internet Engineering Task Force of the Internet Society	
	49	SUPRATIM DEB, et al., "Congestion Control for Fair Resource Allocation in Networks with Fulticas Flows", IEEE/ACM Transactions on Networking, Pages 274-285, Vol. 12, No. 2, April, 2004, IEEE ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./	

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	50	UNANUE, Iraide et al., "A Tutorial on H.264/SVC Scalable Video Coding and its Tradeoff between Quality, Coding Efficiency and Performance," www.intechopen.com, Computer and Information Science, Computer Graphics, Recent Advances on Video Coding, Book Published, July 5, 2011 under CC BY-NC-SA 3.0 License, 24 pages.

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature	/Hyun Hong/	Date Considered	06/12/2016

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☒ See attached certification statement.

Fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

None

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Stephen Y. Liu/	Date (YYYY-MM-DD)	2016-02-11
Name/Print	Stephen Y. Liu	Registration Number	62883

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed . Time will vary depending upon the individual ents on the amount of time you and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

system of records may be disclosed, as a rout... ...blic after either publication of rsuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

EFS Web 2.1.17

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Not for submission under 37 CFR 1.99)

Application Number	15010573
Filing Date	2016-01-29
First Named Inventor	Van Zijst, Erik
Art Unit	2465
Examiner Name	Not yet assigned
Attorney Docket Number	GPB1-US9

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
			WO		2008-05-22			

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	1	VELTEN et al., "Reliable Data Protocol (RDP)", RFC-908, July 1984, Sections (3.4 through 3.5 and subsections pp. 15-18, 3.7 and subsections pp. 20-29,4 and subsections pp. 31-43, 5.6 p. 50), Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
	2	VICISANO et al., "TCP-like congestion control for layered multicast data transfer", IEEE INFOCOM '98 Proceedings, pp. 996-1003 Vol. 3, March 1998, Sections (1 pp. 1-3 of 25, 2.3 through 3.1 inclusive pp. 4-7 of 25, 3.4 pp. 8-10 of 25), IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, California 90720-1314 USA.	
	3	WEI ZHAO et al., "Efficient Adaptive Media Scaling and Streaming of Layered Multimedia in Heterogeneous Environment", Sept. 1999, Sections (1 through 5 inclusive of subsections, pp. 377-381), IEEE Conference on Multimedia Computing and Systems (1999), vol.2, pp.377-381, 2001 L Street NW. Suite 700, Washington, DC 20036-4910 USA.	
	4	WEI ZHAO et al.. "Malleable Multimedia Presentations: Adaptive Streaming Tradeoffs for Best-Quality Fast-Response Systems". Multimedia Communications Part 2, pp 223-236, 1999, Sections (1 through 2 inclusive pp. 223-228, 3 first paragraph pp. 228-229,4 through 5 inclusive pp. 231-236), Springer-Verlag London, Springer Science+Business Media UK Ltd., 236 Gray's Inn Road, Floor 6, London WC1X 8HB, United Kingdom.	
	5	XUE LI, et al., "Layered Video Multicast with Retransmission (LVMR): Evaluation of Error Recovery Schemes", http://citeseer.ist.psu.edu/155541.html, 12 pages, Atlanta, GA, USA.	
	6	ZIJST, Erik Van "Address Summarization for Distance Vector Protocols," MarketXS, August 28, 2003, 5 pages.	
	7	ZIJST, Erik Van "Fair Bandwidth Allocation and Packet Prioritisation for TerraCast," October 7, 2004, 8 pages.	
	8	ZIJST, Erik Van "Introduction to the Internet's limitations regarding realtime data distribution," Internet Broadcast Corporation, December 6, 2004, 7 pages.	
		Van "Technical Overview of the TerraCast Network," May 24, 2004, 2 pages.	
	10	International Search Report PCT/US2006/002995, dated August 29, 2008, 3 pages	
	11	SCHULZRINNE et al., "Real Time Streaming Protocol (RTSP)", RFC-2326, April 1998, Section 14.2 pp 65-67, Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

	12	SCHULZRINNE et al., "RTP: A Transport Protocol for Real-Time Application", RFC-1889, January 1996, Sections (Abstract on pg. 1, Section 1. pp. 3-5, Sections 2.2 and 2.3 pp. 6-7, Section 5.1 pp. 10-12, Section 7.4 on pg 44), Internet Engineering Task Force of the Internet Society, Reston, VA, USA.	
	13	INTERNATIONAL TELECOMMUNICATION UNION, "Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services - Coding of moving video, Advanced video coding for generic audiovisual services," H.264, May 2003, 282 pages.	
	14	INTERNATIONAL TELECOMMUNICATION UNION, "Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services - Coding of moving video, Advanced video coding for generic audiovisual services," H.264, March 2005, 343 pages.	
	15	JOINT VIDEO TEAM OF ISO/IEC MPEG & ITU-T VCEG, "Joint Scalable Video Model JSVM-4," 17th meeting, Nice, France, October 2005, 206 pages.	
	16	WU et al., "Streaming Video over the Internet: Approaches and Directions," IEEE Transactions on Circuits and Systems for Video Technology, pages 282-300, March 2001, 20 pages.	

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature	/Hyun Hong/	Date Considered	06/12/2016

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	15010573
	Filing Date	2016-01-29
	First Named Inventor	Van Zijst, Erik
	Art Unit	2465
	Examiner Name	Not yet assigned
	Attorney Docket Number	GPB1-US9

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☒ See attached certification statement.

Fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

None

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Stephen Y. Liu/	Date (YYYY-MM-DD)	2016-02-11
Name/Print	Stephen Y. Liu	Registration Number	62883

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed . Time will vary depending upon the individual ents on the amount of time you m and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

system of records may be disclosed, as a rout... ...blic after either publication of rsuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /H.H./