US 20180234473A1

(54) **DIRECT PACKET PLACEMENT**

(71) Applicant: **Mellanox Technologies, Ltd.**, Yokneam (IL)

(72) Inventors: **Dotan Levi**, Kiryat Motzkin (IL); **Idan Burstein**, Carmiel (IL); **Shlomi Museri**, Haifa (IL); **Richard Hastie**, Nantwich (GB)

(57) **ABSTRACT**

Communication apparatus includes a host interface and a network interface, which receives at least first and second redundant packet streams, each including a sequence of data packets, which include headers containing respective packet sequence numbers and data payloads of a predefined, fixed size containing respective slices of the data segment. Redundant first and second copies of each slice are transmitted in respective packets in the first and second packet streams. Packet processing circuitry receives the data packets from the network interface, maps the data packets in both the first and second packet streams, using the packet sequence numbers, to respective addresses in a buffer, and writes the data payloads to the respective addresses via the host interface while eliminating redundant data so that the buffer contains exactly one copy of each slice of the data segment, ordered in accordance with the packet sequence numbers.
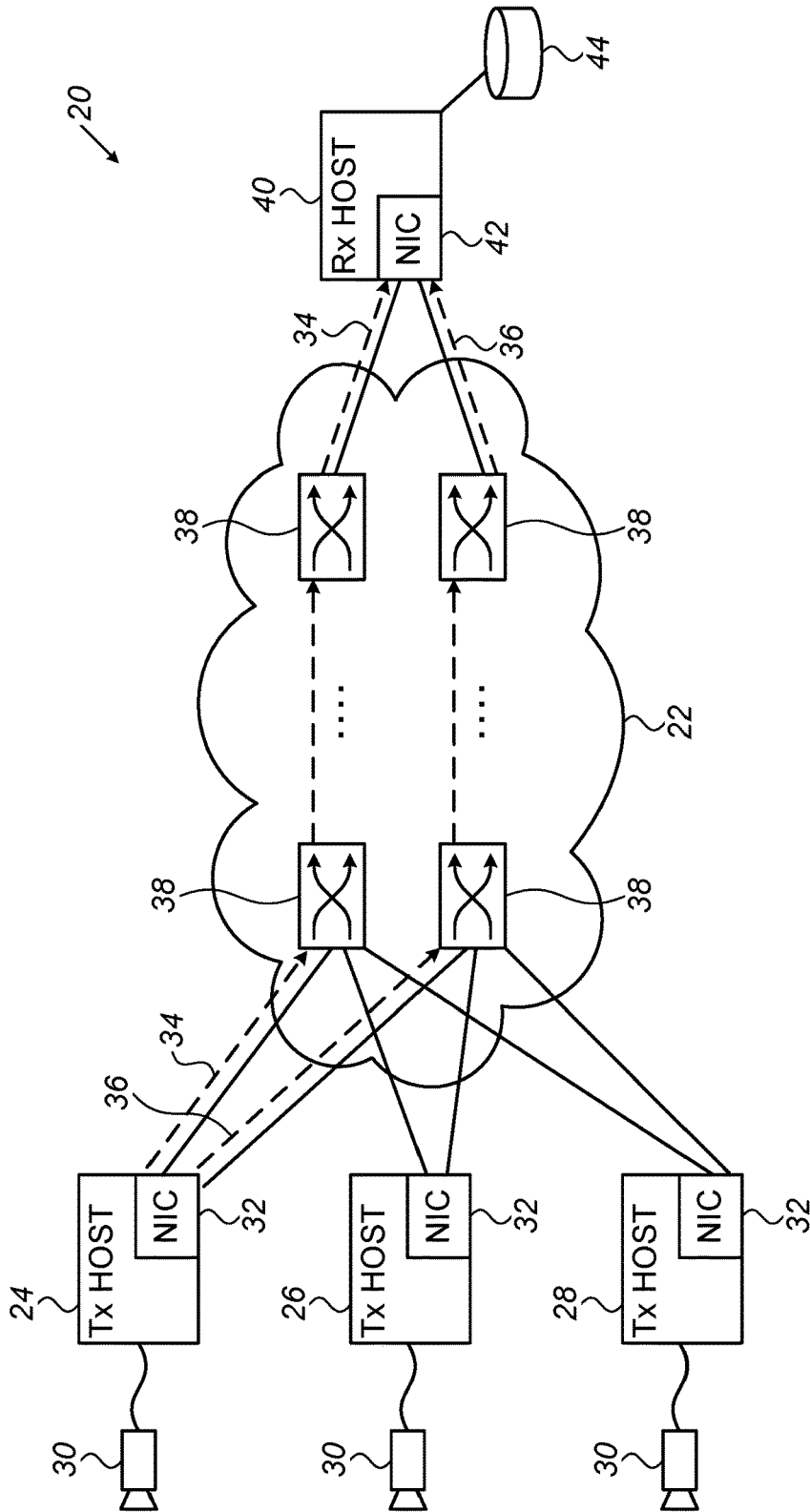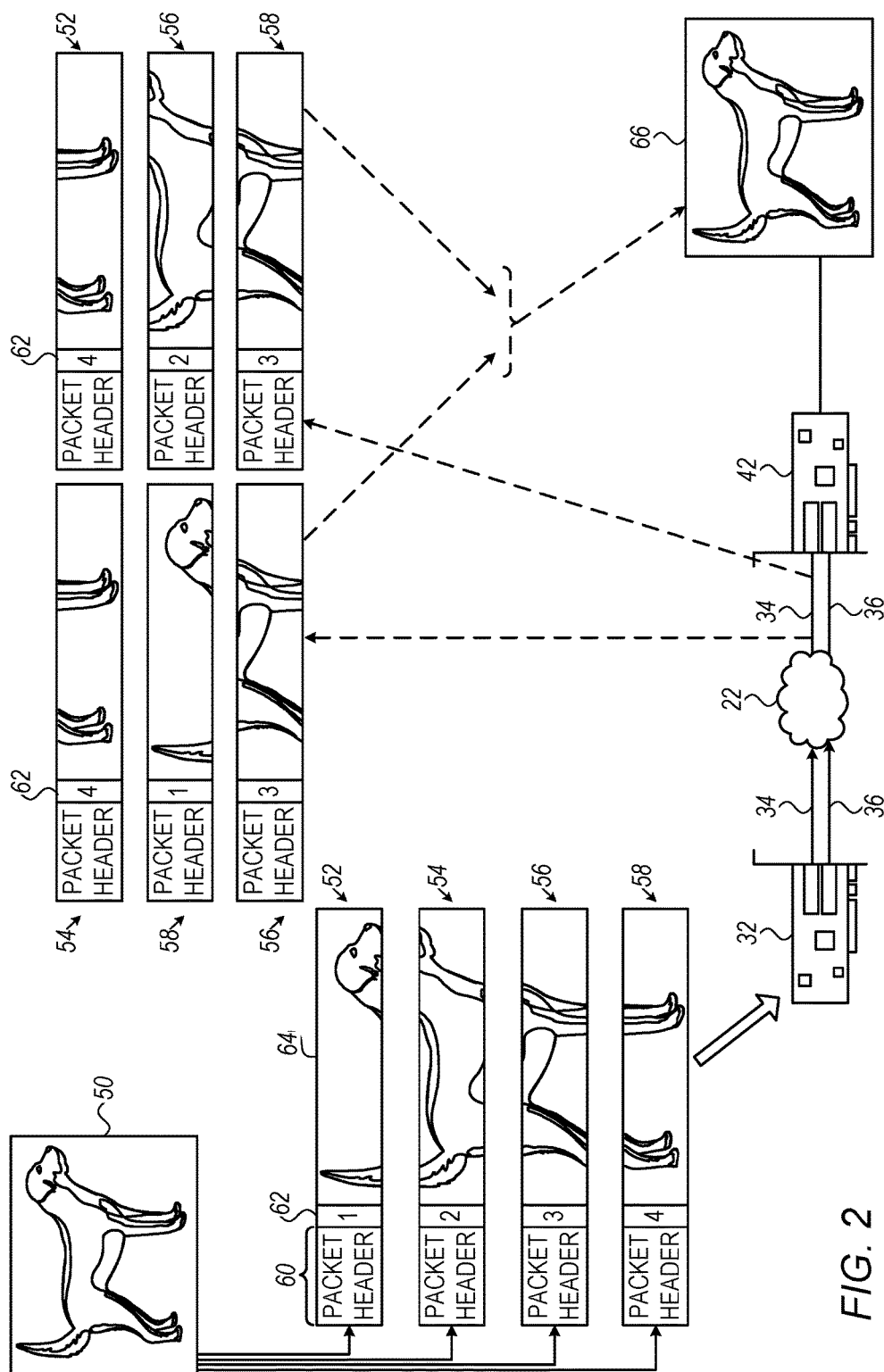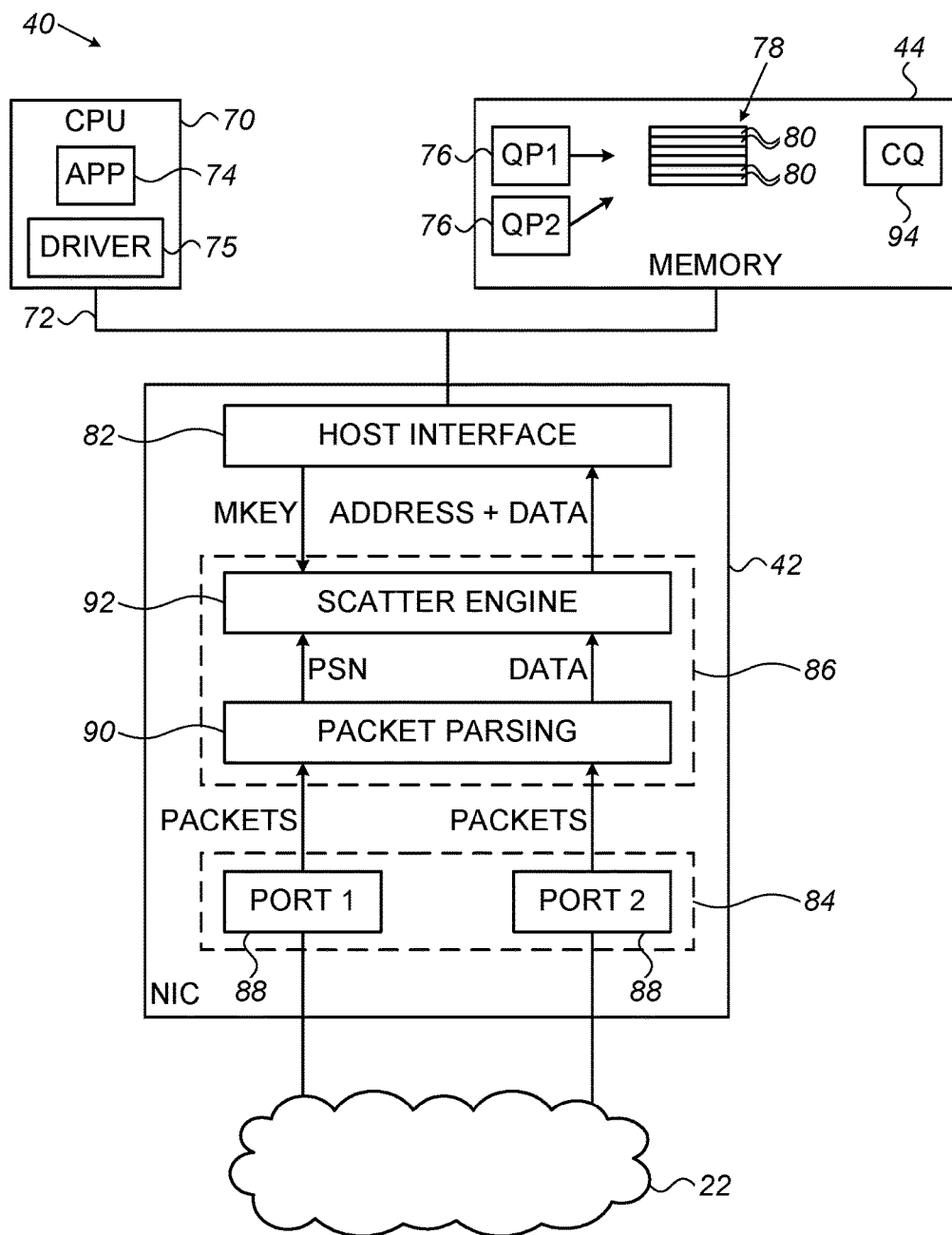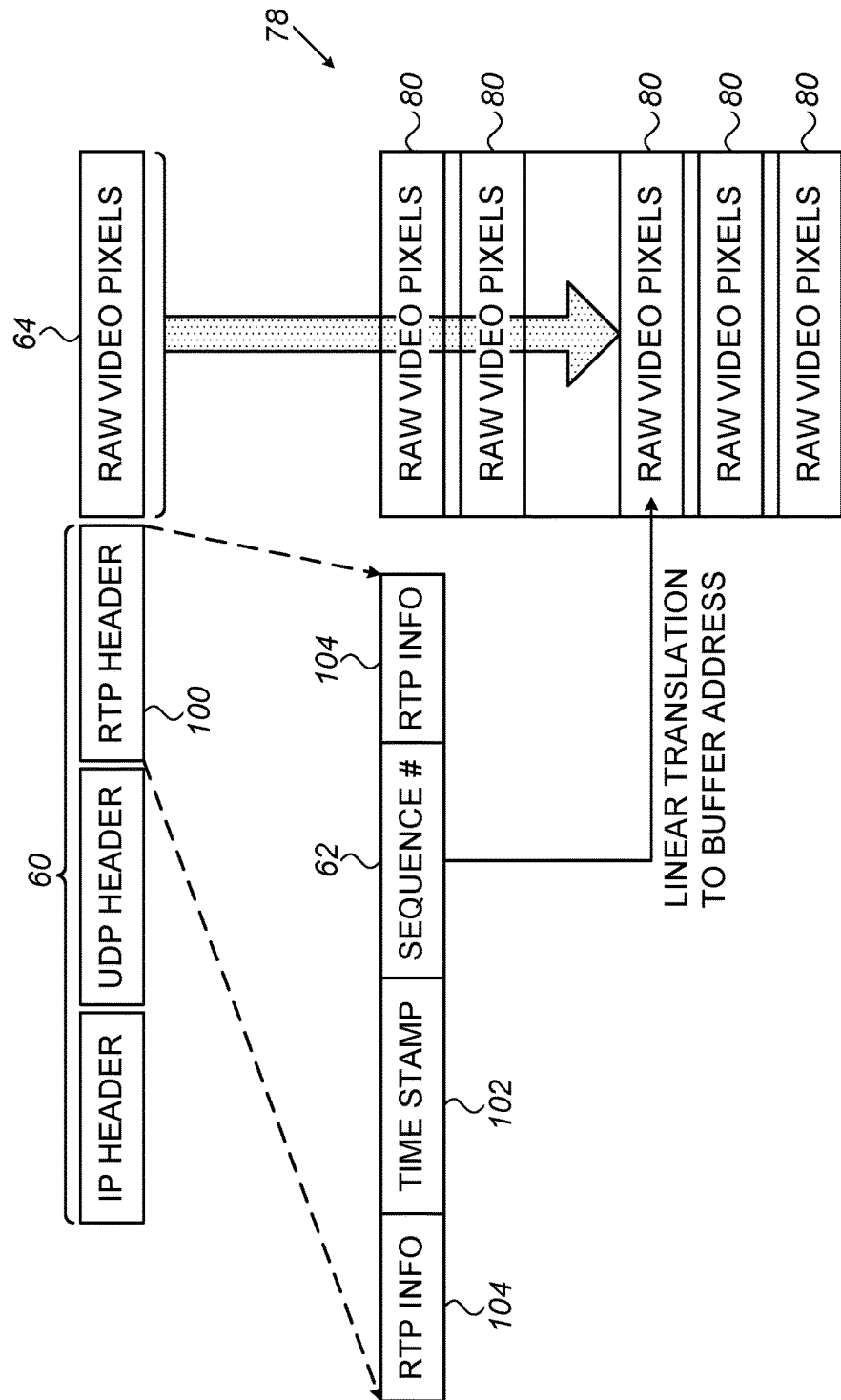
FIG. 1

FIG. 2
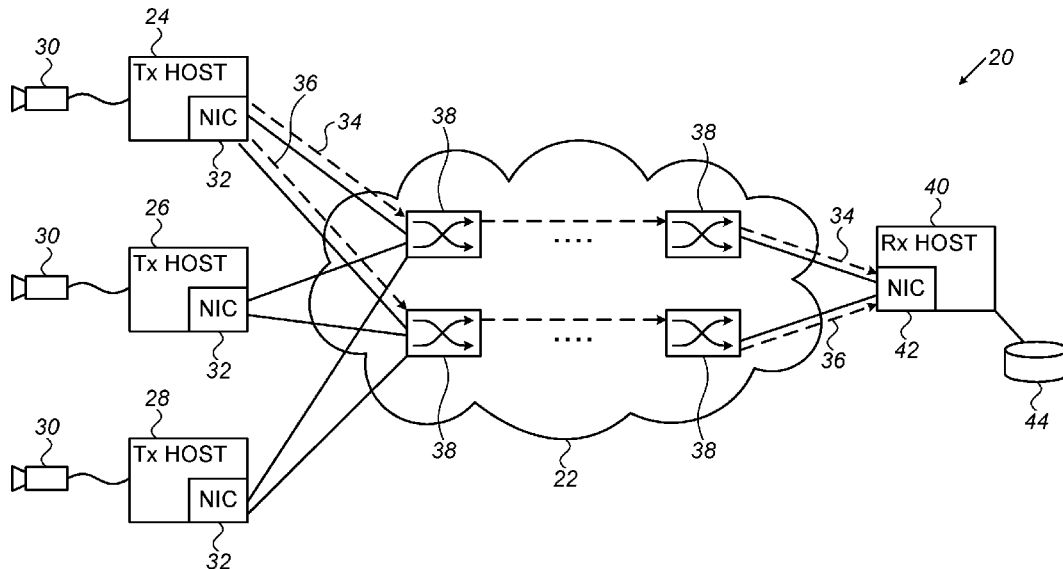
FIG. 3

FIG. 4

DIRECT PACKET PLACEMENT

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Patent Application 62/457,919, filed Feb. 12, 2017, which is incorporated herein by reference.

FIELD OF THE INVENTION

[0002] The present invention relates to apparatus and methods for transmission and reception of packet streams across a network with high speed and reliability.

BACKGROUND

[0003] High-speed packet streaming schemes are commonly used in transmitting real-time video and other digital media across a network. Because of the real-time nature of the data, the packet transmissions typically use a non-reliable protocol, without acknowledgment from the destination or retransmission by the source when packets are lost or corrupted. Therefore, in applications requiring high data availability, a source host may transmit multiple parallel, redundant streams of the data to the destination. Each packet is thus transmitted multiple times—once in each stream—over multiple different paths through the network—in order to increase the likelihood that at least one copy of each packet will be received intact at the destination.

[0004] In this sort of scheme, there is still no guarantee that all the packets in any given stream will reach the destination, nor can it be ensured that the packets will arrive at the destination in the order in which they were transmitted. Therefore, in many applications (such as broadcast or storage of the video data), the destination computer must store, reorder and interleave packets from the two (or more) received streams in order to reconstruct the data. This solution enables reliable video reconstruction, but at the cost of a substantial memory footprint and a heavy processing burden on the host processor, which increases data latency and can limit the data throughput.

[0005] Various schemes for handling data from redundant transmissions are known in the art. For example, U.S. Patent Application Publication 2009/0034633 describes a method for simultaneous processing of media and redundancy streams for mitigating impairments. The method comprises receiving a primary stream of encoded frames and a separate stream of redundant frames. The method further comprises decoding and reconstructing in parallel the frames in the primary stream and the separate stream of redundant frames, on a real-time basis, in accordance with a specified common clock reference. The method further comprises, upon determining that a frame in the primary stream exhibits an error or impairment, determining a decoded redundant frame in the separate stream that corresponds to the impaired frame, and substituting at least a portion of the information in the decoded redundant frame for a corresponding decoded version of the impaired frame.

SUMMARY

[0006] Embodiments of the present invention that are described herein provide efficient apparatus and methods for receiving and handling redundant data streams from a network.

[0007] There is therefore provided, in accordance with an embodiment of the invention, communication apparatus, including a host interface, which is configured to be connected to a bus of a host computer having a processor and a memory, in which a buffer is allocated for receiving a data segment. A network interface is configured to receive from a packet communication network at least first and second redundant packet streams. Each packet stream includes a sequence of data packets, which include headers containing respective packet sequence numbers and data payloads of a predefined, fixed size containing respective slices of the data segment, such that redundant first and second copies of each slice are transmitted respectively in at least a first data packet in the first packet stream and a second data packet in the second packet stream. Packet processing circuitry is configured to receive the data packets from the network interface, to map the data packets in both the first and second packet streams, using the packet sequence numbers, to respective addresses in the buffer, and to write the data payloads to the respective addresses via the host interface while eliminating redundant data so that the buffer contains exactly one copy of each slice of the data segment, ordered in accordance with the packet sequence numbers.

[0008] In some embodiments, the packet processing circuitry is configured to map the packet sequence numbers to the respective addresses, using a linear mapping defined so that the first and second copies of any given data slice are both mapped to a common address in the buffer. In a disclosed embodiment, the mapping is defined such that each packet sequence number PSN is mapped to an address equal to $A+(PSN-X) \times B$, wherein X is an initial sequence number, B is the fixed size of the data payloads, and A is a base address of the buffer.

[0009] In some embodiments, the data payloads include video data, including multiple data segments corresponding to frames of the video data. In one embodiment, the headers contain an indication of a start and end of each frame, and the packet processing circuitry is configured to identify the indication in the headers and to select the buffer to which the data payloads are to be written responsively to the identified indication.

[0010] In a disclosed embodiment, the packet processing circuitry is configured to maintain a record of the packet sequence numbers for which the data payloads have been written to the buffer, and responsively to the record, to discard redundant copies of the data payloads so that each slice is transmitted over the bus via the host interface no more than once. Alternatively, the packet processing circuitry is configured to transmit all of the copies of each slice over the bus so that the redundant data in the buffer are overwritten.

[0011] In a disclosed embodiment, the packet processing circuitry is configured to receive a work item from the processor indicating, for each data segment, an address of the buffer and the size of the slices, and to submit a completion report to the processor when all slices of the data segment have been written to the buffer.

[0012] In one embodiment, the first and second packet streams are transmitted in accordance with a Real-time Transport Protocol (RTP), and the headers include an RTP header, which contains the packet sequence numbers.

[0013] There is also provided, in accordance with an embodiment of the invention, a method for data communication, including allocating a buffer in a memory of a host

computer for receiving a data segment. At least first and second redundant packet streams are received in a network interface controller (NIC) of the host computer from a packet communication network. Each packet stream includes a sequence of data packets, which include headers containing respective packet sequence numbers and data payloads of a predefined, fixed size containing respective slices of the data segment, such that redundant first and second copies of each slice are transmitted respectively in at least a first data packet in the first packet stream and a second data packet in the second packet stream. The data packets in both the first and second packet streams are mapped, using the packet sequence numbers, to respective addresses in the buffer. The data payloads are written to the respective addresses in the buffer while eliminating redundant data so that the buffer contains exactly one copy of each slice of the data segment, ordered in accordance with the packet sequence numbers.

[0014] There is additionally provided, in accordance with an embodiment of the invention, a method for data communication, which includes allocating a buffer in a memory of a host computer for receiving a data segment. A packet stream including a sequence of data packets, which include headers containing respective packet sequence numbers and data payloads of a predefined, fixed size containing respective slices of the data segment, is received in a network interface controller (NIC) of the host computer from a packet communication network. The data packets in the packet stream are mapped to respective addresses in the buffer using a linear mapping of the packet sequence numbers to the addresses. The data payloads are written to the respective addresses in the buffer, ordered in accordance with the packet sequence numbers.

[0015] The present invention will be more fully understood from the following detailed description of the embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] FIG. 1 is a block diagram that schematically illustrates a data communication system, in accordance with an embodiment of the invention;

[0017] FIG. 2 is a block diagram that schematically illustrates a process of data reconstruction that is carried out by a network interface controller (NIC), in accordance with an embodiment of the invention;

[0018] FIG. 3 is a block diagram that schematically illustrates a host computer with a NIC, in accordance with an embodiment of the invention; and

[0019] FIG. 4 is a block diagram that schematically illustrates a method for direct placement of packet data, in accordance with an embodiment of the invention.

DETAILED DESCRIPTION OF EMBODIMENTS

[0020] In many real time streaming protocols, such as the Real-time Transport Protocol (RTP), each of the packets in each transmitted stream contains a respective packet sequence number (PSN), which can be used at the destination in detecting missing packets and restoring the packet data to the transmit order. In the present embodiments, as explained in detail hereinbelow, these reordering and reconstruction functions are carried out by hardware logic in the network interface controller (NIC) of the destination computer, thus offloading these tasks from the host processor.

Although the disclosed embodiments relate to a single stream of video data (which is typically duplicated in at least one redundant stream), in general the destination computer may receive and process many such streams concurrently from different sources.

[0021] In the disclosed embodiments, the destination host processor allocates a single buffer to receive the reconstructed video data in each segment of an incoming data stream, for example, in each video frame, based on the expected segment size and data rate. These stream parameters may be preset or negotiated in each instance by exchange of control messages over the network between the source and destination hosts. The packet sizes can thus be assumed to be fixed and known to the host processors. Each transmitted packet is labeled with a successive PSN, as provided by RTP or another suitable protocol.

[0022] The receiving NIC performs the functions of data integrity checking, packet reordering, and elimination of redundant data, using the known packet sizes and PSNs. Thus the receiving NIC writes the data payloads to the proper locations in the allocated buffer, in accordance with the transmit order (and irrespective of the receive order), while discarding or overwriting the data from redundant packets. The receiving NIC notifies the destination host processor only when the data segment in memory is complete. The process of packet reception and reordering is thus entirely transparent to the destination host processor, which deals only with complete data segments. Assuming the receiving NIC is able to receive data from the network at wire speed, the latency and throughput of video data transfer at the destination are limited only by the processing and bus access rates of the receiving NIC. This solution not only maximizes data bandwidth, but also substantially reduces the processing load, memory footprint, and power consumption of the host processor.

[0023] Although the disclosed embodiments relate specifically to scenarios in which the source host transmits multiple redundant streams of packets to the destination, the principles of the present invention may similarly be applied to single streams of packets (transmitted without redundancy), as well as to schemes with higher degrees of redundancy.

[0024] FIG. 1 is block diagram that schematically illustrates a data communication system 20, in accordance with an embodiment of the invention. In the pictured scenario, multiple transmitting (Tx) host computers 24, 26, 28, . . . , are connected to a packet data network 22, such as an Internet Protocol (IP) network, for example. Each host computer receives a streaming data input, for example from a respective video camera 30. This data input comprises a series of data segments, such as video frame or fields, of a predefined size. The host computer encapsulates the data in a stream of data packets, each comprising a header, containing, inter alia, a respective PSN, and a data payload of a predefined, fixed size containing a respective slice of a given data segment. Although host computers 24, 26, 28 and respective video cameras 30 are shown in FIG. 1, for the sake of conceptual clarity, as separate physical entities, in some implementations the host computer is incorporated into the video camera (typically together with a NIC, as well).

[0025] A NIC 32 in each host computer 24, 26, 28, . . . , transmits the data packets over network 22 in two redundant packet streams 34 and 36, as indicated by the dashed arrows

3

in FIG. **1**. Thus, redundant copies of each slice are transmitted concurrently in two packets, one in each of the two packet streams. Streams **34** and **36** are typically transmitted through network **22** over different, respective paths, for example by transmitting the streams through different ports of NIC **32**, which are connected respectively to different switches **38** in the network. The packets are typically transmitted using an unreliable transport protocol, such as RTP over User Datagram Protocol (UDP) transport, which gives no guarantee that any given packet will reach its destination. Transmitting the two redundant streams ensures that at least one copy of each packet will, with few exceptions, reach its destination. For even higher confidence of delivery, three or even more redundant streams can be transmitted and handled at the receiver in the manner described herein. Alternatively, as noted above, the techniques described herein are useful in efficient handling of even a single packet stream by a receiving NIC, when redundancy is not required.

[0026] Streams **34** and **36**, as well as the packet streams transmitted by hosts **26, 28, . . .** , are addressed to a receiving (Rx) host computer **40**, which is connected to network **22** by a NIC **42**. Upon receiving the data packets from network **22**, NIC **42** maps the packets in both of packet streams **34** and **36**, using the packet sequence numbers, to respective addresses in a buffer in a memory **44** of host computer **40**. NIC **42** writes the data payloads to the respective addresses while eliminating redundant data so that the buffer contains exactly one copy of each slice of the data segment, ordered in accordance with the packet sequence numbers. The video frames are thus immediately available in memory for retransmission (over a television network, for example) or other access.

[0027] FIG. **2** is a block diagram that schematically illustrates a process of data reconstruction that is carried out by NIC **42**, in accordance with an embodiment of the invention. Video camera **30** captures an image frame **50**, which is divided into slices for transmission by NIC **32** in a sequence of data packets **52, 54, 56, 58**. Each packet comprises a header **60**, containing a packet sequence number (PSN) **62**, and a payload **64** containing a respective slice of image frame **50**. NIC **32** transmits two copies of each packet **52, 54, 56, 58**, i.e., one in each of streams **34** and **36**.

[0028] In the pictured scenario, certain packets are lost or corrupted in transmission through network **22**, and others reach NIC **42** out of order. NIC **42** detects and discards corrupted packets, for example by computing and checking packet checksums or other error detection codes, as provided by the applicable protocols. In the pictured example, NIC **42** has received only packets **54, 58** and **56** (in that order) from stream **34** and packets **52, 56** and **58** from stream **36**. Using packet serial numbers **62**, however, NIC is able to directly place data from the appropriate payloads **64** in the designated buffer in memory **44**, and thus reconstruct a complete image frame **66** in memory **44**. As noted earlier, NIC **42** eliminates redundant data so that the buffer contains exactly one copy of each slice of the frame, ordered in accordance with the packet sequence numbers. The process of reconstruction is transparent to software running on host computer **40** and requires no involvement by the software in data reordering or eliminating redundancies.

[0029] FIG. **3** is a block diagram that schematically shows details of host computer **40** and NIC **42**, in accordance with an embodiment of the invention. Host computer **40** comprises a processor **70**, typically comprising at least one central processing unit (CPU), and memory **44**, which are connected to NIC **42** by a host bus **72**, such as a PCI Express® bus, as is known in the art. Processor **70** runs various software programs, including, in this example, a video processing application **74** and a driver **75** for controlling operation of NIC **42**.

[0030] Driver **75** creates a respective work queue **76** (conventionally referred to as a queue pair, or QP) for each incoming stream of video packets that NIC **42** is to receive. For each segment in the incoming stream (such as a video frame) that NIC **42** is to receive, driver **75** posts a corresponding work item (referred to as a work queue element, or WQE) in the appropriate work queue **76** pointing to a buffer **78** that is to receive the data in the segment. In some embodiments, this functionality is implemented as follows:

[0031] Driver **75**, upon receiving a control message indicating that such a video stream is to be initiated, creates a QP for the flow, and specifies the following parameters, for example in the QP context in memory **44**, to be accessed by NIC **42** via bus **72**:

[0032] Packet size;

[0033] Header size;

[0034] Range of expected packet sequence numbers in the segment;

[0035] PSN offset within the packet header;

[0036] Size and offset of the payload data (excluding the frame checksum that follows the data).

[0037] In some embodiments, for every segment (which may comprise tens, hundreds, or even thousands of IP packets), the driver **75** posts a WQE in queue **76** containing the following information, to be accessed by NIC **42**:

[0038] Segment PSN start (X);

[0039] Segment buffer base address (A).

[0040] The WQE thus specifies the location and contents of buffer **78**, which is divided into slices **80** of the payload size indicated by the QP context. In some implementations, the WQE contains a memory key (MKEY), to be used by NIC **42** in accessing buffer **78**. Alternatively, the above information can be included in the QP context, which points to a cyclic buffer to which NIC **42** will write slices **80** in succession (thus relieving the software of the burden of continually allocating buffers and posting WQEs).

[0041] When another identical, redundant flow is to be sent in a second stream, driver **75** creates another QP, specifying the same parameters (packet size, header size, etc.)

[0042] For every segment of this redundant flow, the software will post the following information to the NIC:

[0043] Segment PSN start (Y)—The two streams need not have the same PSNs, although it is simpler if Y=X. If not, driver **75** provides NIC **42** with the necessary context information to synchronize between the corresponding packet sequence numbers in the two streams.

[0044] Segment buffer base address (A).

[0045] The result will be that both copies of any given data slice are mapped to a common slice address in buffer **78**, and thus the corresponding, redundant data will be written to the same place in the host memory, regardless of the arrival order.

[0046] NIC **42** comprises a host interface **82**, connected to bus **72**, and a network interface **84**, which connects to network **22**. Network interface **84** in this example comprises two ports **88** with different addresses, serving as the respective destination addresses for streams **34** and **36**. Alternatively, both streams may be received through the same port.

[0047] Packet processing circuitry **86** is coupled between host interface **82** and network interface **84**. For the sake of simplicity, FIG. **3** shows only certain specific elements of packet processing circuitry **86**: packet parsing logic **90** and a scatter engine **92**, which are used in implementing the direct data placement functionality that is the subject of the present description. In a typical NIC, these elements are combined with other components for transmitting and receiving packets for other purposes and in accordance with other sorts of protocols. These latter components will be apparent to those skilled in the art and are beyond the scope of the present description. Packet processing circuitry **86**, including parsing logic **90** and scatter engine **92**, is typically implemented in hardware logic circuits within one or more integrated circuit chips. The logic may be hard-wired or programmable, or a combination of hard-wired and programmable elements. Additionally or alternatively, certain functions of packet processing circuitry **86** may be carried out in software or firmware by an embedded microprocessor or microcontroller.

[0048] Network interface **84** passes incoming packets in streams **34** and **36** to packet parsing logic **90**. To process these packets, packet processing circuitry **86** reads and makes use of the information posted in the appropriate work queue **76** in memory **44** by driver **75**. This information enables packet parsing logic **90** to locate and extract PSN **62** and payload **64** from each packet. Scatter engine **92** uses PSN **62**, together with the base address of buffer **78**, to map each slice **80** to the respective address in the buffer, and thus to write the data payloads to the appropriate addresses via host interface **82**.

[0049] FIG. **4** is a block diagram that schematically shows details of a method for direct placement of packet data that is performed by NIC **42**, in accordance with an embodiment of the invention. FIG. **4** illustrates how scatter engine **92** uses PSN **62** in an RTP header **100** of a video packet in placing the packet data in buffer **78**. Other examples of video networking protocols that can be handled in similar fashion include (without limitation):

[0050] SMPTE 2022 -5/6/7

[0051] VSF TR-03/IETF RFC 4175

[0052] intoPIX TICO

[0053] Sony IP Live RDD 34 and NMI

[0054] NewTek

[0055] GigE vision GVSP

[0056] The packet illustrated in FIG. **4** belongs to a stream in which the payload size of all packets is fixed and known in advance. Driver **75** running on host processor **70** assigns and informs NIC **42** of the base address of buffer **78** to which each data segment in the stream is to be written, as explained above. Within packet header **60**, RTP header **100** comprises a timestamp **102** and other protocol information **104**, in addition to PSN **62**. Based on the context information provided by driver **75**, packet parsing logic **90** advances to the known location of PSN **62** in each packet and extracts the PSN value, as well as extracting data slice **80** from payload **64**. On this basis, scatter engine **92** is able to write the data slices from payloads **64** to buffer **78** in the proper order by simple linear translation of the PSN.

[0057] In other words, for a packet stream beginning from an initial sequence number X, with a data payload of size B in each packet, and a buffer base address A, scatter engine **92** will write the payload of each packet to an address:

$$\text{Payload address} = A + (PSN - X) \times B.$$

[0058] When the payloads of two valid packets from different, respective streams in a redundant transmission scheme map to the same address (whether they have the same or different PSNs), only one of them will ultimately be written to the buffer. Scatter engine **92** may overwrite the payload of the packet that arrives first with that of the second copy of the packet, or it may write only one of the payloads to the buffer and discard the other.

[0059] For some protocols, the segment definition is also indicated in the packet headers, thus enabling alternative implementations that reduce the involvement of driver software even further. For example:

[0060] A segment in most video applications corresponds to a single frame (or a single field of interlaced video).

[0061] In some video protocols, a frame-end marker is entered in the packet transport header: The first and last packets in the segment are identified by a special field in the transport header indicating that they are the first or last packet.

[0062] When the frame size is fixed and the frame start and end are marked in the headers in this manner, the NIC can automatically start to write the data into the allocated host buffers when the start marker is received. Thus, the need to specify segment boundaries is also offloaded from the software to the NIC hardware logic.

[0063] For a NIC operating at very high speed (for example, receiving incoming video data at 400 Gbps), access from NIC **42** to memory **44** over bus **72** can become a bottleneck. To reduce the bus pressure, packet processing circuitry **86** can monitor PSNs and, when the payload from a given packet having a given PSN in one of the streams has already been written to the buffer, simply discard the corresponding packet from the other stream, rather than overwriting the data already in the buffer. For this purpose, processing circuitry **86** may maintain a record, such as a PSN vector and a rolling PSN window, for each flow. Each packet arriving causes processing circuitry **86** to flip a bit in the PSN vector. When the bit for a given PSN is set when corresponding packet from the other stream arrives, packet parsing logic **90** drops this latter packet without further processing.

[0064] It will be appreciated that the embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention includes both combinations and subcombinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art.

1. Communication apparatus, comprising:

a host interface, which is configured to be connected to a bus of a host computer having a processor and a memory, in which a buffer is allocated for receiving a data segment;

a network interface, which is configured to receive from a packet communication network at least first and second redundant packet streams, each packet stream comprising a sequence of data packets, which comprise headers containing respective packet sequence numbers and data payloads of a predefined, fixed size containing respective slices of the data segment, such that redundant first and second copies of each slice are transmitted respectively in at least a first data packet in the first packet stream and a second data packet in the second packet stream; and

packet processing circuitry, which is configured to receive the data packets from the network interface, to map the data packets in both the first and second packet streams, using the packet sequence numbers, to respective addresses in the buffer, and to write the data payloads to the respective addresses via the host interface while eliminating redundant data so that the buffer contains exactly one copy of each slice of the data segment, ordered in accordance with the packet sequence numbers.

2. The apparatus according to claim 1, wherein the packet processing circuitry is configured to map the packet sequence numbers to the respective addresses, using a linear mapping defined so that the first and second copies of any given data slice are both mapped to a common address in the buffer.

3. The apparatus according to claim 2, wherein the mapping is defined such that each packet sequence number PSN is mapped to an address equal to $A+(PSN-X) \times B$, wherein X is an initial sequence number, B is the fixed size of the data payloads, and A is a base address of the buffer.

4. The apparatus according to claim 1, wherein the data payloads comprise video data, comprising multiple data segments corresponding to frames of the video data.

5. The apparatus according to claim 4, wherein the headers contain an indication of a start and end of each frame, and wherein the packet processing circuitry is configured to identify the indication in the headers and to select the buffer to which the data payloads are to be written responsively to the identified indication.

6. The apparatus according to claim 1, wherein the packet processing circuitry is configured to maintain a record of the packet sequence numbers for which the data payloads have been written to the buffer, and responsively to the record, to discard redundant copies of the data payloads so that each slice is transmitted over the bus via the host interface no more than once.

7. The apparatus according to claim 1, wherein the packet processing circuitry is configured to transmit all of the copies of each slice over the bus so that the redundant data in the buffer are overwritten.

8. The apparatus according to claim 1, wherein the packet processing circuitry is configured to receive a work item from the processor indicating, for each data segment, an address of the buffer and the size of the slices, and to submit a completion report to the processor when all slices of the data segment have been written to the buffer.

9. The apparatus according to claim 1, wherein the first and second packet streams are transmitted in accordance with a Real-time Transport Protocol (RTP), and wherein the headers comprise an RTP header, which contains the packet sequence numbers.

10. A method for data communication, comprising:

allocating a buffer in a memory of a host computer for receiving a data segment;

receiving in a network interface controller (NIC) of the host computer from a packet communication network at least first and second redundant packet streams, each packet stream comprising a sequence of data packets, which comprise headers containing respective packet sequence numbers and data payloads of a predefined, fixed size containing respective slices of the data segment, such that redundant first and second copies of each slice are transmitted respectively in at least a first data packet in the first packet stream and a second data packet in the second packet stream;

mapping the data packets in both the first and second packet streams, using the packet sequence numbers, to respective addresses in the buffer; and

writing the data payloads to the respective addresses in the buffer while eliminating redundant data so that the buffer contains exactly one copy of each slice of the data segment, ordered in accordance with the packet sequence numbers.

11. The method according to claim 10, wherein mapping the data packets comprises mapping the packet sequence numbers to the respective addresses, using a linear mapping defined so that the first and second copies of any given data slice are both mapped to a common address in the buffer.

12. The method according to claim 11, wherein the mapping is defined such that each packet sequence number PSN is mapped to an address equal to $A+(PSN-X) \times B$, wherein X is an initial sequence number, B is the fixed size of the data payloads, and A is a base address of the buffer.

13. The method according to claim 10, wherein the data payloads comprise video data, comprising multiple data segments corresponding to frames of the video data.

14. The method according to claim 13, wherein the headers contain an indication of a start and end of each frame, and wherein mapping the data packets comprises identifying the indication in the headers and selecting the buffer to which the data payloads are to be written responsively to the identified indication.

15. The method according to claim 10, wherein writing the data payloads comprises maintaining a record of the packet sequence numbers for which the data payloads have been written to the buffer, and responsively to the record, discarding redundant copies of the data payloads so that each slice is transmitted over a bus of the host computer no more than once.

16. The method according to claim 10, wherein writing the data payloads comprises transmitting all of the copies of each slice over a bus of the host computer so that the redundant data in the buffer are overwritten.

17. The method according to claim 10, wherein mapping the data packets comprises receiving a work item from a processor of the host computer indicating, for each data segment, an address of the buffer and the size of the slices, and submitting a completion report to the processor when all slices of the data segment have been written to the buffer.

18. The method according to claim 10, wherein the first and second packet streams are transmitted in accordance with a Real-time Transport Protocol (RTP), and wherein the headers comprise an RTP header, which contains the packet sequence numbers.

19. A method for data communication, comprising:

allocating a buffer in a memory of a host computer for receiving a data segment;

receiving in a network interface controller (NIC) of the host computer from a packet communication network a packet stream comprising a sequence of data packets, which comprise headers containing respective packet sequence numbers and data payloads of a predefined, fixed size containing respective slices of the data segment;

mapping the data packets in the packet stream to respective addresses in the buffer using a linear mapping of the packet sequence numbers to the addresses; and

writing the data payloads to the respective addresses in the buffer, ordered in accordance with the packet sequence numbers.

20. The method according to claim **19**, wherein the mapping is defined such that each packet sequence number PSN is mapped to an address equal to A+(PSN−X)×B, wherein X is an initial sequence number, B is the fixed size of the data payloads, and A is a base address of the buffer.

* * * * *