(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 July 2020 (30.07.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/154223 A1

(51) International Patent Classification:
G06F 15/16 (2006.01)

(21) International Application Number:
PCT/US2020/014269

(22) International Filing Date:
20 January 2020 (20.01.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/794,994 21 January 2019 (21.01.2019) US

(72) Inventor; and
(71) Applicant: RANKIN, John [US/US]; Federal Correctional Institution (73251-061), P.O. Box 1000, Morgantown, WV 20507 (US).

(74) Agent: STANDLEY, Jeffrey, S. et al.; Standley Law Group LLP, 6300 Riverside Drive, Dublin, OH 43017 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(54) Title: SYSTEMS AND METHODS FOR PROCESSING NETWORK TRAFFIC USING DYNAMIC MEMORY

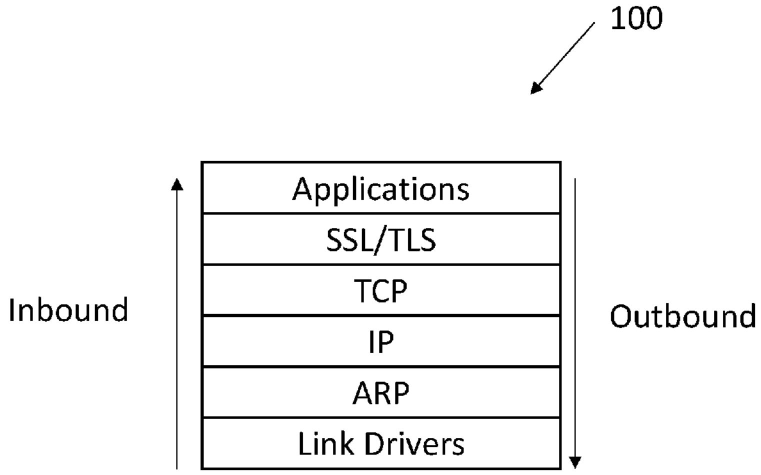

Figure 1
(Prior art)

(57) Abstract: Systems and method for processing network traffic are provided. The network traffic includes a number of data packets (300) representing a complete transmission which are located at a first electronic storage area. Each data packet (300) including a data payload (310). A data block (400) is generated by one or more processors according to software instructions for the received traffic. The data block (400) includes a series of header pointers (404) pointing to each of a series of headers (402) and a data pointer (404) pointing to the data payloads (406).

WO 2020/154223 A1

SYSTEMS AND METHODS FOR PROCESSING NETWORK TRAFFIC USING DYNAMIC MEMORY

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of US provisional patent application serial no. 62/794,994 filed January 21, 2019, the disclosures of which are hereby incorporated by reference as if fully restated herein.

TECHNICAL FIELD

[0002] Exemplary embodiments of the present invention relate generally to systems and methods for processing network traffic using dynamic memory.

BACKGROUND AND SUMMARY OF THE INVENTION

[0003] Two of the most important communication protocols used on the Internet and other similar networks are the Transmission Control Protocol (TCP) and the Internet Protocol (IP). Together, the TCP and IP protocols form core protocols of the larger Internet protocol suite used on packet-switched networks. That protocol suite is commonly referred to as the TCP/IP protocol because of the widespread adoption and implementation of the TCP and IP protocols.

[0004] The TCP/IP protocol was developed for the United States Advanced Research Projects Agency (ARPA). The TCP/IP protocol is a set of rules that enable different types of network-enabled or networked devices to communicate with each other. Those network devices communicate by using the TCP/IP standard, or format, to transfer or share data. TCP/IP rules are established and maintained by the Internet Engineering

Task Force (IETF). The IETF is an international community of network designers, operators, vendors, and researchers concerned with the Internet's architecture and operation. The IETF's mission is to produce technical and engineering documents that influence the way people design, use and manage the Internet with the goal of improving its operations and efficiencies. These documents include protocol standards, best current practices and information updates of various kinds, and are commonly referred to as Request for Comments (RFC).

[0005] TCP can be used to establish a bi-directional connection between two clients wherein activity begins with a request for information made by one client to another client. A "client" may be any program or application that initiates requests for or sends information from one remote location to another. As used herein, the term "client" may refer to such applications including, but not limited to, web browsers, web servers, file transfer protocol (FTP) programs, electronic mail programs, line printer (LPR) programs also known as print emulators, mobile phone apps, and telnet programs also known as terminal emulators, all of which operate conceptually in an application layer.

[0006] TCP software accepts requests and data streams directly from clients and other daemons, sequentially numbering the bytes, or octets, in the stream during the time the connection is active. When required, the TCP software breaks the data stream into smaller pieces called segments (sometimes referred to as datagrams or packets generally) for transmission to a requesting client. The protocol calls for the use of checksums, sequence numbers, timestamps, time-out counters and retransmission algorithms to ensure reliable data transmission.

[0007] The IP layer actually performs the communication function between two networked hosts. The IP software receives data segments from the TCP layer, ensures that the segment is sized properly to meet the requirements of the transmission path and physical adapters (such as Ethernets and CTCs). IP changes the segment size if necessary by breaking it down into smaller IP datagrams, and transmits the data to the physical network interface or layer of the host.

[0008] The software that manages handling the TCP/IP RFCs may be referred to as a TCP/IP protocol stack. This software may be referred to as a stack because the different protocols are often layered into this type of organization. A TCP/IP stack may have, for example, the following layers: 1) Link drivers, 2) ARP, 3) IP, 4) TCP, 5) SSL/TLS, and 6) Applications. Inbound data may transfer from layer 1 to layer 6 in turn and outbound data may take the reverse direction going from layer 6 to 1.

[0009] Depending on the implementation, these different layers can operate under the same process, or can function independently. Within the IP layer, there are several separate operational functions: 1) Address resolution, 2) Routing, 3) Fragmentation, and 4) Protocol isolation. Again, inbound data may use these functions from 1 to 4 in secession and outbound data may go in reverse order from 4 to 1.

[0010] A frame of inbound data works its way through the stack, breaking the information down until the final payload that is destined for an application is exposed. Once the application completes operating upon the information, the response is once again passed to the stack. On the outbound operation, additional tags and headers are constructed that will be required to deliver the information back to its origin.

[0011] A TCP/IP stack processes incoming and outgoing IP traffic based upon the natural request of the network. Therefore, as transmissions arrive to be handled, the data is moved into buffers and passed from level to level within the system. This method of handling traffic utilizes a fixed series of buffers and eventually locates the user work areas defined by the socket layer that originates the ability to communicate. In other words, data flowing into a system using a traditional TCP/IP stack will populate fixed buffers that are created by the services in operation.

[0012] The common handling of TCP/IP traffic is managed through the population of fixed buffers. When a connection is opened by an application, there is a buffer provided that is sufficient to hold the network traffic related to the size of a TCP window. As the data flows across the network, data is moved in and out of this fixed buffer and there is a direct relationship between data buffers and flow control. Therefore, TCP window management is ultimately limited to the original established size of the buffer provided at connection open time.

[0013] There are a number of limiting issues that arise through the use of a predefined fixed buffer. The most significant is the failure to promote significant in bound and out bound traffic due to the unnecessary binding of the window flow control. To exacerbate this issue, implementation of TCP does not venture beyond traditional and expected behavior. TCP/IP stacks that have been developed are mostly available in C and open or license free source. These readily available versions function to limit the imagination and innovation that would occur without available source code. Furthermore, when such a volume of TCP/IP stack source code is available, it is most often used as a guide and thus retards exploration and innovation due to the common expectation. As a result, TCP/IP stacks have been implemented with the standard expectation of fixed buffers relating to application creation for the purpose of flow control.

[0014] Therefore, what is needed is a system and method for processing network traffic using dynamic memory. The present invention is a system and method for processing network traffic using dynamic memory.

[0015] A more effective system and method of handling this traffic is realized by dynamically creating a block of memory sufficient to hold the specific elements of network traffic. This block of memory may be created at the time the traffic occurs, rather than before the traffic occurs as conventionally required. This block of network data may be passed from layer to layer without regard to the volume or dimensions of the network traffic. Therefore, rather than predetermining the amount of traffic a system can process and populating predetermined buffers, this dynamic system and process allows for an expansive level of network processing by allowing the number of data containers to dynamically increase to match the network needs.

[0016] These disclosures provide systems and methods where network traffic, such as but not limited to a TCP/IP network, drive the creation and handling of blocks of memory that are utilized for the stack, such as but not limited to a TCP/IP stack, and connection processing. These dynamically allocated memory storage areas contain the incoming and outgoing network traffic and are managed in a way which allows complete utilization without movement or transfers within storage. These systems and processes innovate the handling of connection flow by removing limitations upon the volume and complexity of the network traffic. By allowing physical memory storage to contain network traffic, rather than act as a predetermined and fixed buffer, the network can respond with far more flexibility, reliability, and performance.

[0017] Further features and advantages of the systems and methods disclosed herein, as well as the structure and operation of various aspects of the present disclosure, are described in detail below with reference to the accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] In addition to the features mentioned above, other aspects of the present invention will be readily apparent from the following descriptions of the drawings and exemplary embodiments, wherein like reference numerals across the several views refer to identical or equivalent features, and wherein:

[0019] **FIGURE 1** is an exemplary TCP/IP stack;

[0020] **FIGURE 2** is an exemplary IP layer;

[0021] **FIGURE 3** is an exemplary layer of data in an exemplary TCP/IP transmission;

[0022] **FIGURE 4** is an exemplary dynamically allocated data block;

[0023] **FIGURE 5** is a simplified block diagram illustrating exemplary processing of network traffic using the data blocks of figure 4;

[0024] **FIGURE 6** is a simplified block diagram illustrating exemplary processing of network traffic using the data blocks of figure 4; and

[0025] **FIGURE 7** is a simplified block diagram illustrating exemplary processing of network traffic using the data block of figure 4.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENT(S)

[0026] Various embodiments of the present invention will now be described in detail with reference to the accompanying drawings. In the following description, specific details such as detailed configuration and components are merely provided to assist the overall understanding of these embodiments of the present invention. Therefore, it should be apparent to those skilled in the art that various changes and modifications of the embodiments described herein can be made without departing from the scope and spirit of the present invention. In addition, descriptions of well-known functions and constructions are omitted for clarity and conciseness.

[0027] Embodiments of the invention are described herein with reference to illustrations of idealized embodiments (and intermediate structures) of the invention. As such, variations from the shapes of the illustrations as a result, for example, of manufacturing techniques and/or tolerances, are to be expected. Thus, embodiments of the invention should not be construed as limited to the particular shapes of regions illustrated herein but are to include deviations in shapes that result, for example, from manufacturing.

[0028] **FIGURE 1** is an exemplary TCP/IP stack 100, and **FIGURE 2** is an exemplary IP layer 200. Network traffic, such as that experienced in a TCP/IP network, is generally organized into layers of information that roughly relate to the processing that will occur within the stack 100. **FIGURE 3** is an exemplary layer of data 300 in an exemplary TCP/IP transmission. The data 300 from a user application may be appended with one or more headers, indicated generally at item 312, that may be required for transmission, routing, and delivery of data 310. The header(s) 312 may comprise a hardware header 302, an ethernet header 304, an IP header 306, and a TCP header 308, some combination thereof, or the like. Any number and type of header(s) 312 are contemplated. Any size data payload 310 may be utilized containing any number of types of data.

[0029] **FIGURE 4** is an exemplary dynamically allocated data block 400. The presentiment of TCP/IP traffic may result in the allocation of a single data block 400. The data block 400 may have its own header(s), as indicated generally at 402 associated with a data payload 406, and also contain the complete transmission as one cohesive unit. The header(s) 402 may comprise some or all of the headers 312 shown or described with regards to figure 3, such as but not limited to the hardware header 302, the ethernet header 304, the IP header 306, and the TCP header 308. The header(s) 402 may further comprise one or more pointers 404. With the complete transmission contained within a single unit of storage (i.e., data block 400), it may be possible to manage the use of the data through a series of pointers, as indicated generally at 404, to information stored within the data block header(s) 402. All other stack implementations have isolated the information into predefined containers rather than allow for the flexibility of a dynamic construction.

[0030] **FIGURE 5** is a simplified block diagram illustrating exemplary processing of network traffic using the data blocks 400 of figure 4. Since the data blocks 400 may each comprise all relevant information, the data blocks 400 may be passed from process to process without requiring movement or reallocation. Such processes may include, for example without limitation, physical I/O processing 502, decoding ethernet frame 504, handling IP data and fragments 506, managing TCP flow control 508, and user applications 510. As each process of the TCP/IP stack 100 analyzes the incoming or outgoing transmission, the data block header 402 may be updated to reflect realistic pointers 404 and observations that may be used by other processes. For example, data transmissions arrive by some method of physical distribution. This physical and electrical interaction is often different depending upon the physical connector, adapter, or linkage. Therefore, the unit specific process that handles the physical connection is in the best position to determine the time of arrival, the point of payload data, and the overall length of the transmission. All of these pieces of information may be recorded and updated within the data block header 402 so as to be available for the next process in the stack 100 to receive the data block 400 for processing.

[0031] **FIGURE 6** is a simplified block diagram illustrating exemplary processing of network traffic using the data blocks 400 of figure 4. The use of connected data blocks 400 into a single unit may also serve for effective management of a TCP connection flow. Normally, TCP connection flows are limited by the pre-allocated buffers created while the connection is open. However, by dynamically allocating a data block 400 that contains real world transmission data, the flow of information may be adapted with often unexpected flexibility and performance.

[0032] For example, without limitation, a chain of data blocks 400A, 400B, and 400C may represent the processing of information and may relate to the actual traffic flow rather than forced segmenting, moving, and parking of information within fixed buffers. Each of the data blocks 400A, 400B, and 400C in a chain may be connected by a connection block 502. Each of the data blocks 400A, 400B, and 400C may comprise one or more pointers, or other connections, pointing to the subsequent blocks 400A, 400B, and 400C in the chain. Any number of data blocks 400A, 400B, and 400C may be utilized. The blocks 400A, 400B, and 400C may be linked in linear or non-linear fashions.

[0033] **FIGURE 7** is a simplified block diagram illustrating exemplary processing of network traffic using the data blocks 400 of figure 4. This process of allocating a data block 400 to contain network traffic, rather than using predetermined and fixed buffers, provides for significant improvements over current TCP/IP stack implementations. These data blocks 400A, 400B, 400C, 400D, 400E, 400F, 400G, and 400H may serve as internal management structures that facilitate the operation of a TCP/IP stack. For example, without limitation, the connection block 502 may connect to outbound data blocks 400 (in the illustrated example, data blocks 400A, 400B, 400C, and 400D), which may connect to inbound data blocks 400 (in the illustrated example, data blocks 400E, 400F, 400G, and 400H) by way of a physical connection 504. The last inbound data block 400 (in the illustrated example, data block 400H) may connect with the first outbound data block 400 (in the illustrated example, data block 400A) by way of the connection block. Each of the data blocks 400A, 400B, 400C, 400D, 400E, 400F, 400G, and 400H may be connected to one another using pointers, though other connections may be utilized. Any number of data blocks 400A, 400B, 400C, 400D, 400E, 400F, 400G, and 400H may be utilized. The blocks 400A, 400B, 400C, 400D, 400E, 400F, 400G, and 400H may be linked in linear or non-linear fashions.

[0034] By using this innovative system and process, the following advantages, which are exemplary but do not represent a comprehensive list of advantages, may be achieved:

[0035] Encouragement of data flow may occur through a flexible expansion of data within the TCP window. In other words, in the traditional TCP window management the advertised size of the window may be based upon the available space in the connection buffer that has not been read by the application. However, with dynamic data blocks 400, it is possible to allow far more incoming information to queue in storage while waiting for the application to process the data blocks 400, thus advertising a larger window that the application would limit due to its operation.

[0036] Management of retransmissions may be far more easily controlled due to the one to one relationship between data blocks 400 to transmission units. The retransmission of TCP data is the most sensitive part of the data flow operation and improving its efficiency will improve overall network performance.

[0037] By limiting data block 400 creation to the actual needs of the network, far more efficient use of memory storage is achieved. In other words, if a connection process always allocates the maximum buffer size necessary for a connection, there will be an excess of available storage consumed, due to potentially small actual requirements. However, by allocating dynamically only what is necessary, a conservative approach to memory is achieved.

[0038] By implementing this flexible process, storage retransmission and control flow are managed with greater efficiency and performance.

[0039] Any embodiment of the present invention may include any of the features of the other embodiments of the present invention. The exemplary embodiments herein disclosed are not intended to be exhaustive or to unnecessarily limit the scope of the invention. The exemplary embodiments were chosen and described in order to explain the principles of the present invention so that others skilled in the art may practice the invention. Having shown and described exemplary embodiments of the present invention, those skilled in the art will realize that many variations and modifications may be made to the described invention. Many of those variations and modifications will provide the same result and fall within the spirit of the claimed invention. It is the intention, therefore, to limit the invention only as indicated by the scope of the claims.

[0040] Certain operations described herein may be performed by one or more electronic devices. Each electronic device may comprise one or more processors, electronic storage devices, executable software instructions, and the like configured to perform the operations described herein. The electronic devices may be general purpose computers or specialized computing device. The electronic devices may be personal computers, smartphone, tablets, databases, servers, or the like. The electronic connections and transmissions described herein may be accomplished by wired or wireless means.

CLAIMS

What is claimed is:

1. A method for processing TCP/IP network traffic comprising the steps of:

receiving network traffic at a TCP/IP stack;

generating one or more data blocks in memory for the received network traffic,

wherein each of the one or more data blocks comprise a header and at least a portion of the received network traffic;

organizing each of the one or more data blocks into a chain;

associating at least one of the one or more data blocks in the chain with a connection block;

passing the data block between stack-oriented processes;

updating the headers for each of the one or more data blocks to

reflect accumulated updates regarding the content of the received network traffic;

queuing each of the one or more data blocks from the connection block;

using each of the one or more data blocks to manage the flow of data;

holding an incoming queue of network traffic arriving from the TCP/IP network; and

holding an outgoing queue of network traffic leaving from the TCP/IP stack until a determination is made that retransmission is no longer necessary.

2. A system for processing network traffic comprising:

a first electronic storage area comprising one or more data packets received from network traffic or queued for transmission, wherein each of said data packets comprise a data payload, and wherein each of said received data packets represents a complete transmission;

one or more processors;

a second electronic storage area comprising software instructions, which when executed, configure the one or more processors to generate a data block comprising a number of pointers, and wherein one of said number of pointers points to said data payloads for each of said data packets and each remaining one of said number of pointers points to one of a number of headers.

3. The system of claim 2 wherein:

said network is a TCP/IP network.

4. The system of claim 3 wherein:

said complete transmission comprises all data necessary to generate a webpage.

5. The system of claim 3 wherein:

said complete transmission comprises each fragment of a previously fragmented transmission.

6. The system of claim 3 wherein:

said number of number headers comprise a hardware header, an ethernet header, an IP header, and a TCP header;

said number of pointers comprise a hardware pointer, an ethernet pointer, an IP pointer, a TCP pointer, and a data pointer;

said hardware pointer points to said hardware header;

said ethernet pointer points to said ethernet header;

said IP pointer points to said IP header;

said TCP pointer points to said TCP header; and

said data pointer points to said data payload.

7. The system of claim 6 wherein:

said hardware header comprises data for physical I/O processing;

said ethernet header comprises data for decoding ethernet frames;

said IP header comprises data for handling IP data and fragments;

said TCP header comprises data for managing TCP flow control; and

said data payload comprises data for a user application.

8. The system of claim 2 further comprising:

a number of said data blocks pointing to one another to form a chain.

9. The system of claim 8 further comprising:

a connection block, wherein at least one of said number of data blocks points to said connection block.

10. The system of claim 9 further comprising:

a second number of said data blocks, wherein said second number of data blocks point to one another to form a second chain, and wherein said connection block comprises a pointer which points to one of said second number of data blocks, wherein said number of data blocks of said chain comprise inbound data, and wherein said second number of data blocks in said second chain comprise outbound data.

11. The system of claim 2 wherein:

said first and second electronic storage area are located at one or more electronic storage devices.

12. A method for processing network traffic comprising the steps of:

receiving a request to transmit or receive network traffic at one or more electronic storage devices, wherein said network traffic comprises a number of data packets, wherein each of said number of data packets comprise a data payload, and wherein each of said number of data packets represent a complete data transmission; and

generating a data block for the network traffic, said data block comprising a series of header pointers pointing to each of a series of headers indicating processing instruction for said data block and one or more data pointers pointing to the data payloads for each of the number of data packets.

13. The method of claim 12 wherein:

said network comprises a TCP/IP network.

14. The method of claim 12 further comprising the steps of:

connecting each of said number of data blocks to one another by way of pointers.

15. The method of claim 14 wherein:

each of said number of data blocks are connected in a linear fashion by one or more pointers to form a chain.

16. The method of claim 15 further comprising the steps of:

providing a connection block, wherein at least one of said number of data blocks comprises a pointer to said connection block.

17. The method of claim 16 wherein:

the data payloads of each of said number of data blocks comprise inbound network traffic.

18. The method of claim 12 wherein:

said complete transmission comprises all data necessary to generate a webpage.

19. The method of claim 12 wherein:

said complete transmission comprises each fragment of a previously fragmented transmission.

20. The method of claim 12 wherein:

the step of generating a data block is not performed until each of said number of data packets representing a complete data transmission are received.

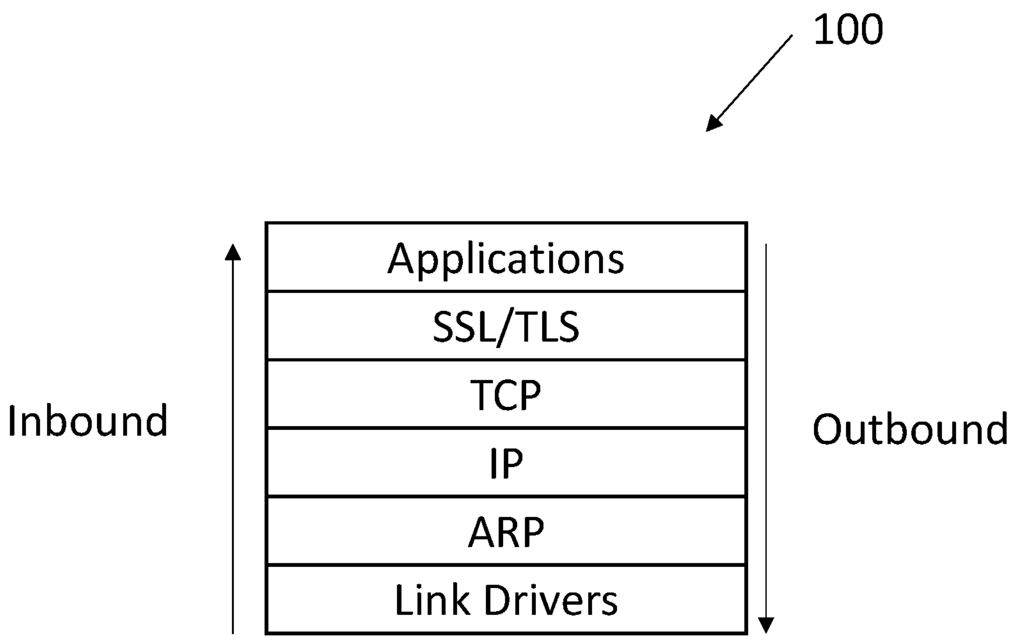

Figure 1
(Prior art)

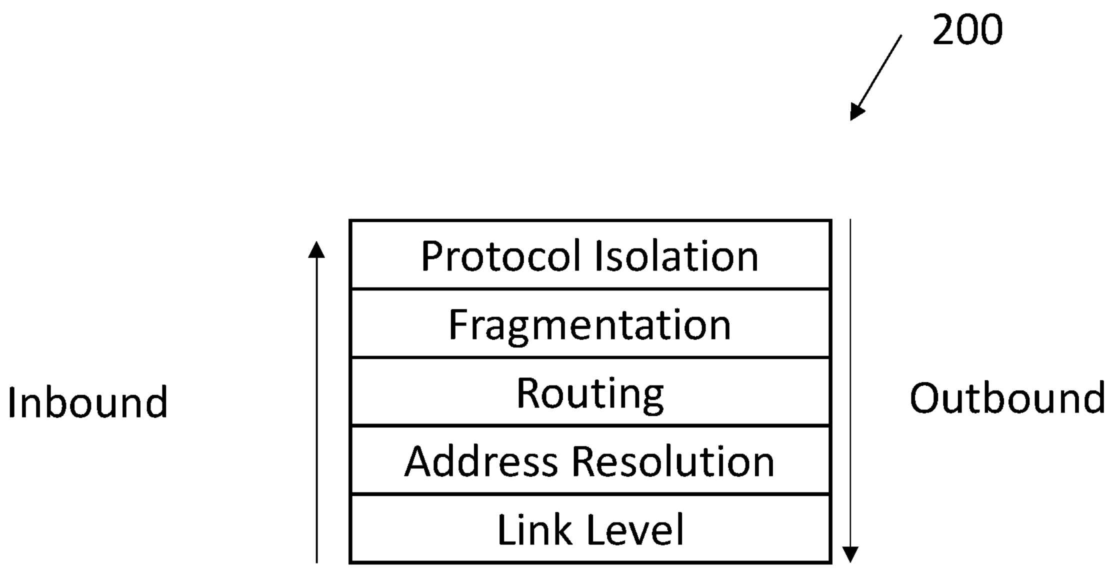

Figure 2
(prior art)

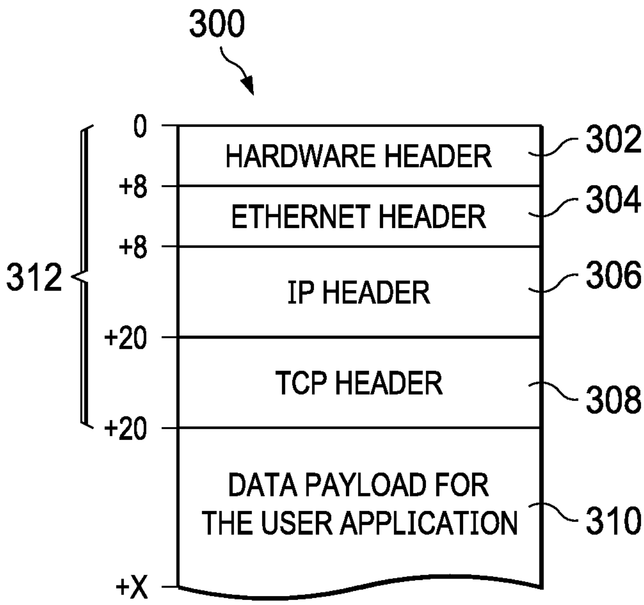

FIG. 3
(PRIOR ART)

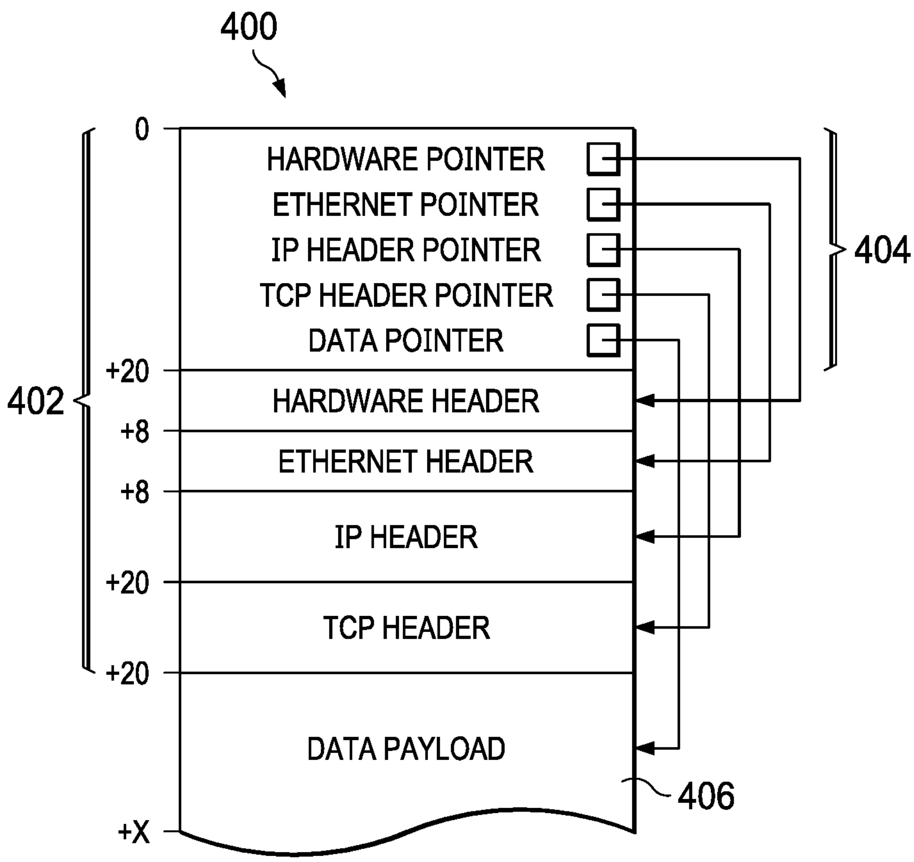

FIG. 4

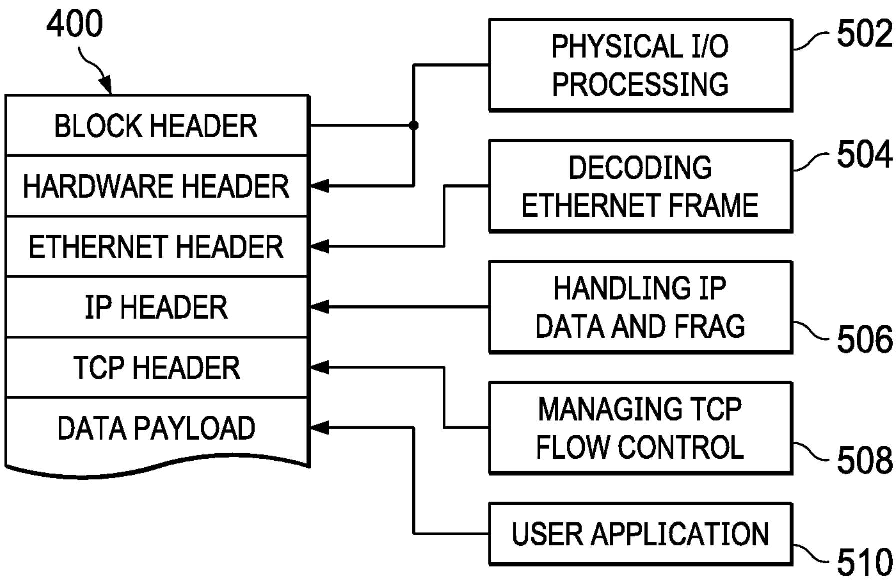
400
BLOCK HEADER
HARDWARE HEADER
ETHERNET HEADER
IP HEADER
TCP HEADER
DATA PAYLOAD
PHYSICAL I/O PROCESSING
502
DECODING ETHERNET FRAME
504
HANDLING IP DATA AND FRAG
506
MANAGING TCP FLOW CONTROL
508
USER APPLICATION
510

FIG. 5

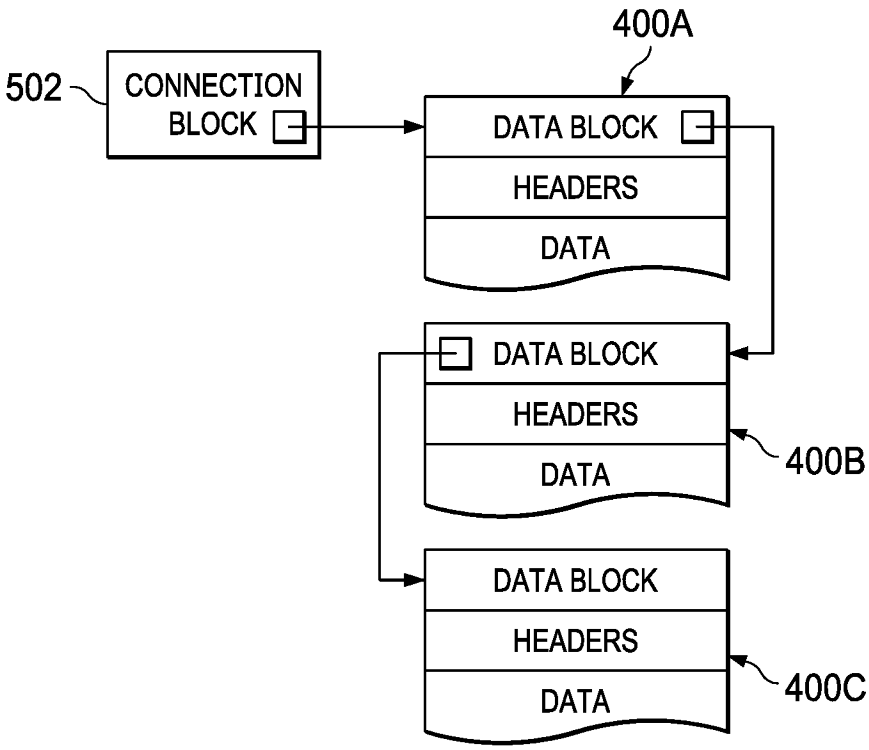
502
CONNECTION BLOCK
400A
DATA BLOCK
HEADERS
DATA
DATA BLOCK
HEADERS
DATA
400B
DATA BLOCK
HEADERS
DATA
400C

FIG. 6

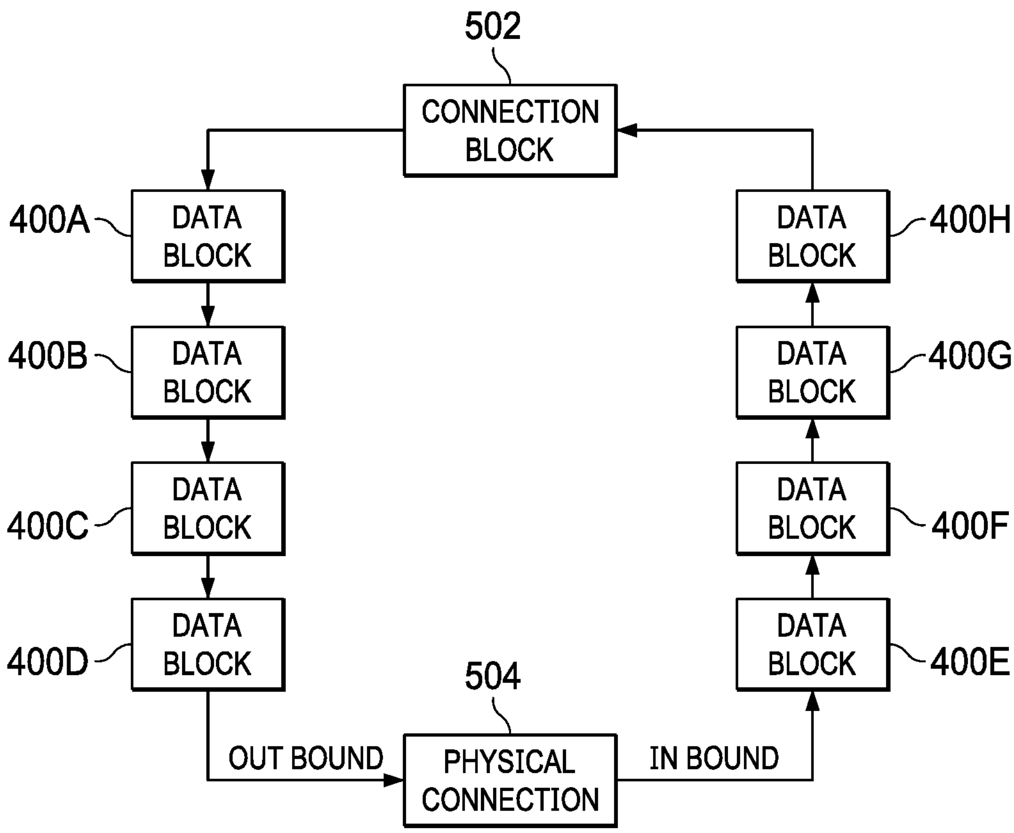
502
CONNECTION BLOCK
400A
DATA BLOCK
400B
DATA BLOCK
400C
DATA BLOCK
400D
DATA BLOCK
400H
DATA BLOCK
400G
DATA BLOCK
400F
DATA BLOCK
400E
DATA BLOCK
504
PHYSICAL CONNECTION
OUT BOUND
IN BOUND

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 20/14269

A. CLASSIFICATION OF SUBJECT MATTER
IPC - G06F 15/16 (2020.1)

CPC - H04L 69/16, H04L 69/165, H04L 69/12, H04L 69/161, H04L 12/5601, H04L 47/10, H04L 47/20

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)
See Search History document

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 8,090,866 B1 (Bashyam et al.), 03 January 2012 (03.01.2012), entire document, especially Abstract; col 4, ln 42-65; col 7, ln 50 to col 8, ln 5	1
Y	US 2008/0104313 A1 (CHU), 01 May 2008 (01.05.2008), entire document, especially Abstract; para [0024], [0036]-[0038]	1
Y	US 2007/0025388 A1 (Abhishek et al.), 01 February 2007 (01.02.2007), entire document, especially Abstract; para [0029], [0034], [0036]	1
A	US 2005/0105506 A1 (Birdwell et al.) 19 May 2005 (19.05.2005), entire document	1

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"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)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date 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 one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 May 2020	23 JUN 2020

Name and mailing address of the ISA/US	Authorized officer
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Lee Young Telephone No. PCT Helpdesk: 571-272-4300

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 20/14269

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

--- (See Continuation in Supplemental Box) ---

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:
Claim 1

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

Continuation of:
Box III. Observations where unity of invention is lacking

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 searched, the appropriate additional search fees must be paid.

Group I - Claim 1 is directed to a method for processing TCP/IP network traffic.

Group II - Claims 2-20 are directed to a method and system for storing network traffic.

The inventions listed as Groups I though 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:

Special Technical Features

Group I require the special technical feature of organizing each of the one or more data blocks into a chain; associating at least one of the one or more data blocks in the chain with a connection block; passing the data block between stack-oriented processes; updating the headers for each of the one or more data blocks to reflect accumulated updates regarding the content of the received network traffic; queuing each of the one or more data blocks from the connection block; using each of the one or more data blocks to manage the flow of data; holding an incoming queue of network traffic arriving from the TCP/IP network; and holding an outgoing queue of network traffic leaving from the TCP/IP stack until a determination is made that retransmission is no longer necessary, not required by group II.

Group II require the special technical feature of receiving a request to transmit or receive network traffic at one or more electronic storage devices; one or more processors; generate a data block comprising a number of pointers, and wherein one of said number of pointers points to said data payloads for each of said data packets and each remaining one of said number of pointers points to one of a number of headers, not required by group I.

Common Technical Features

Groups I-II share the features of receiving network traffic; a first electronic storage area comprising one or more data blocks/packets, each of the one or more data blocks/packets comprise a header and at least a portion of the received network traffic.

However, this shared technical feature does not represent a contribution over prior art, because the shared technical feature is being anticipated by US 2005/0105506 A1 (Birdwell et al.) (hereinafter 'Birdwell'), 19 May 2005 (19.05.2005).

Birdwell teaches receiving network traffic (para [0015]- data received over a data network); a first electronic storage area comprising one or more data blocks/packets (para [0044], [0065] - the network data packet received from the data network 72 is initially placed in a register. A header 134 and optional padding 136 are added to form the MPT frame 130, which is stored in another register; MPT packets are written into main memory in the order in which they arrive. The entire MPT packet 140' is written to memory); each of the one or more data blocks/packets comprise a header and at least a portion of the received network traffic (para [0063]- the IP packet 120' has intact the N-byte data payload 122, the 8-byte UDP header 124,).

As the common features were known in the art at the time of the invention, this cannot be considered a common technical feature that would otherwise unify the groups. Therefore, Groups I-II lack unity under PCT Rule 13.