US 20080317027A1

(19) **United States**

(12) **Patent Application Publication**

(10) **Pub. No.: US 2008/0317027 A1**

(43) **Pub. Date: Dec. 25, 2008**

BASSO et al.

(54) **SYSTEM FOR REDUCING LATENCY IN A HOST ETHERNET ADAPTER (HEA)**

(75) Inventors: **Claude BASSO**, Raleigh, NC (US); **Jean Louis Calvignac**, Raleigh, NC (US); **Chih-jen Chang**, Apex, NC (US); **Philippe Damon**, Raleigh, NC (US); **Natarajan Vaidhyanathan**, Carrboro, NC (US); **Fabrice Jean Verplanken**, La Gaude (FR); **Colin Beaton Verrilli**, Apex, NC (US)

Correspondence Address:
IBM RP-RPS
SAWYER LAW GROUP LLP
2465 E. Bayshore Road, Suite No. 406
PALO ALTO, CA 94303 (US)

(73) Assignee: **INTERNATIONAL BUSINESS MACHINES CORPORATION**, Armonk, NY (US)

(21) Appl. No.: **12/200,970**

(22) Filed: **Aug. 29, 2008**

Related U.S. Application Data

(63) Continuation of application No. 11/096,353, filed on Apr. 1, 2005.

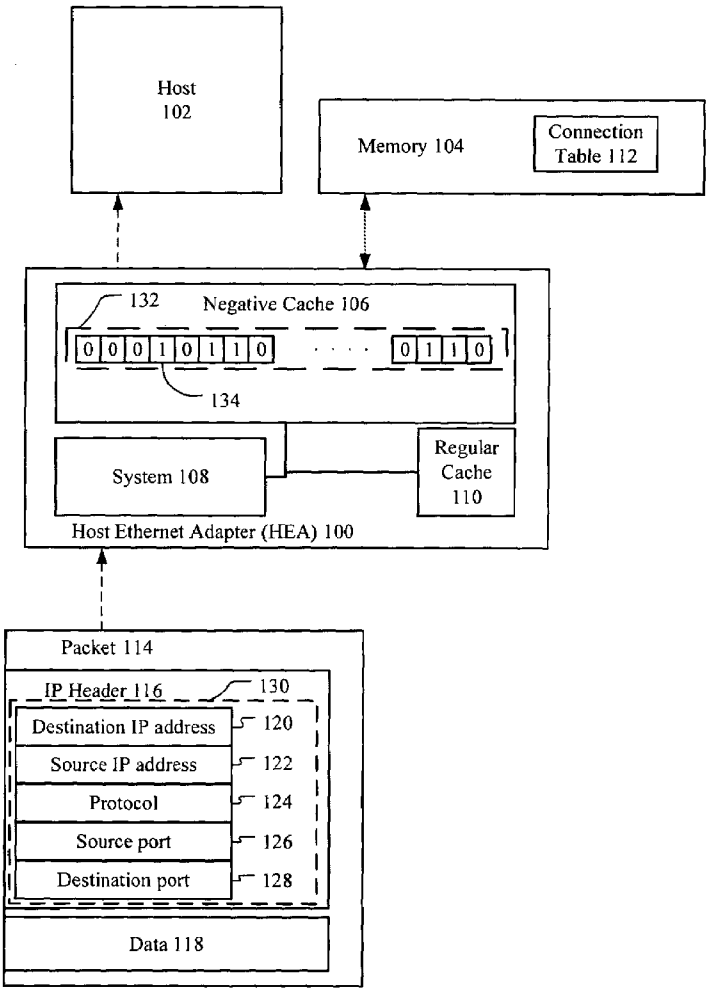
Publication Classification

(51) **Int. Cl.**
H04L 12/56 (2006.01)
H04L 12/28 (2006.01)

(52) **U.S. Cl.** **370/389**

(57) **ABSTRACT**

A system for reducing latency in a host Ethernet adapter (HEA) includes the following. First, the HEA receives a packet with an internet protocol (IP) header and data in the HEA. The HEA parses a connection identifier from the IP header and accesses a negative cache in the HEA to determine if the connection identifier is not in a memory external to the HEA. The HEA applies a default treatment to the packet if the connection identifier is not in the memory, thereby reducing latency by decreasing access to the memory.



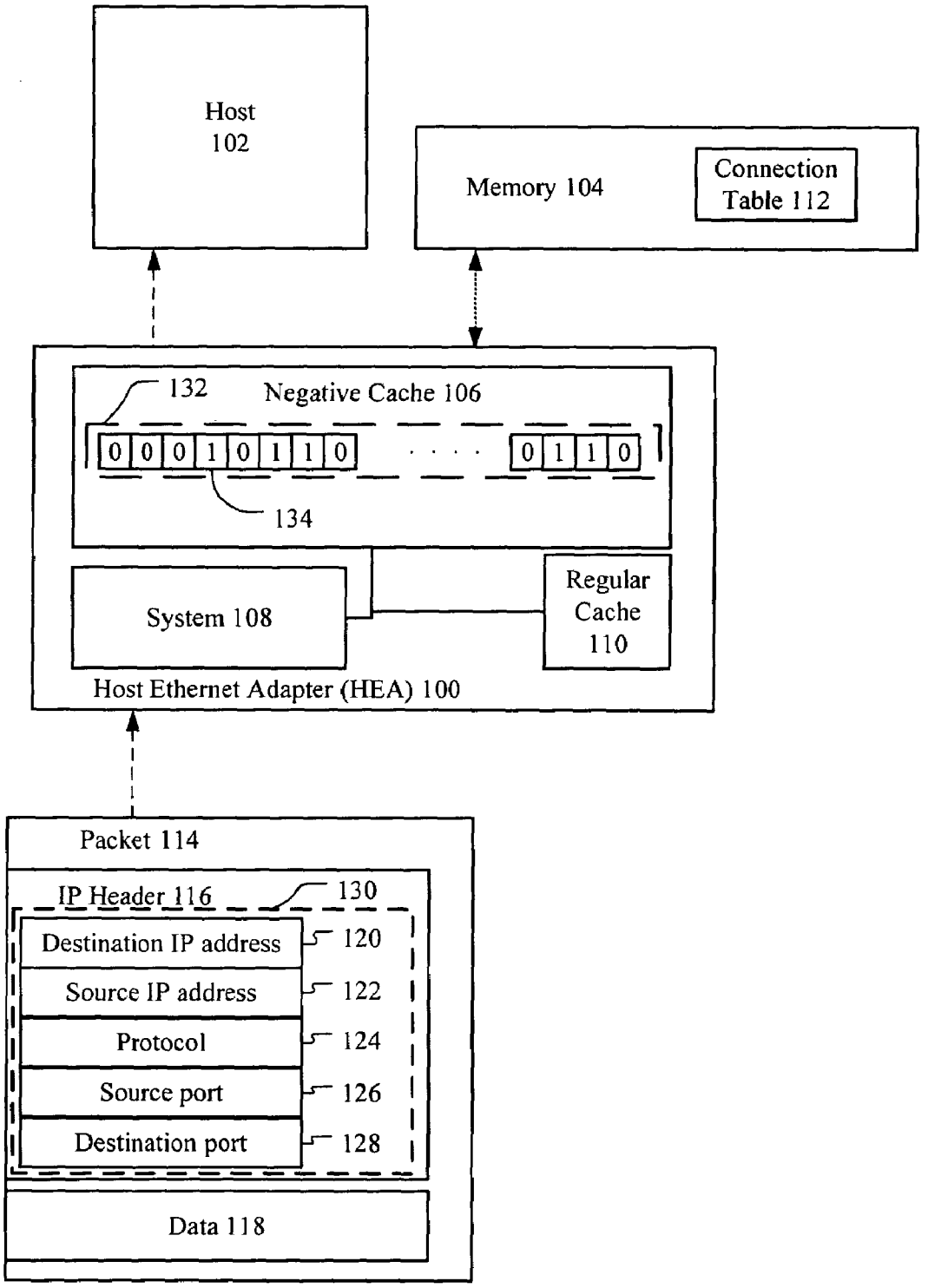

Host
102
Memory 104
Connection
Table 112
132
Negative Cache 106
0 0 0 1 0 1 1 0
0 1 1 0
134
System 108
Regular
Cache
110
Host Ethernet Adapter (HEA) 100
Packet 114
IP Header 116
130
Destination IP address
120
Source IP address
122
Protocol
124
Source port
126
Destination port
128
Data 118

FIG. 1

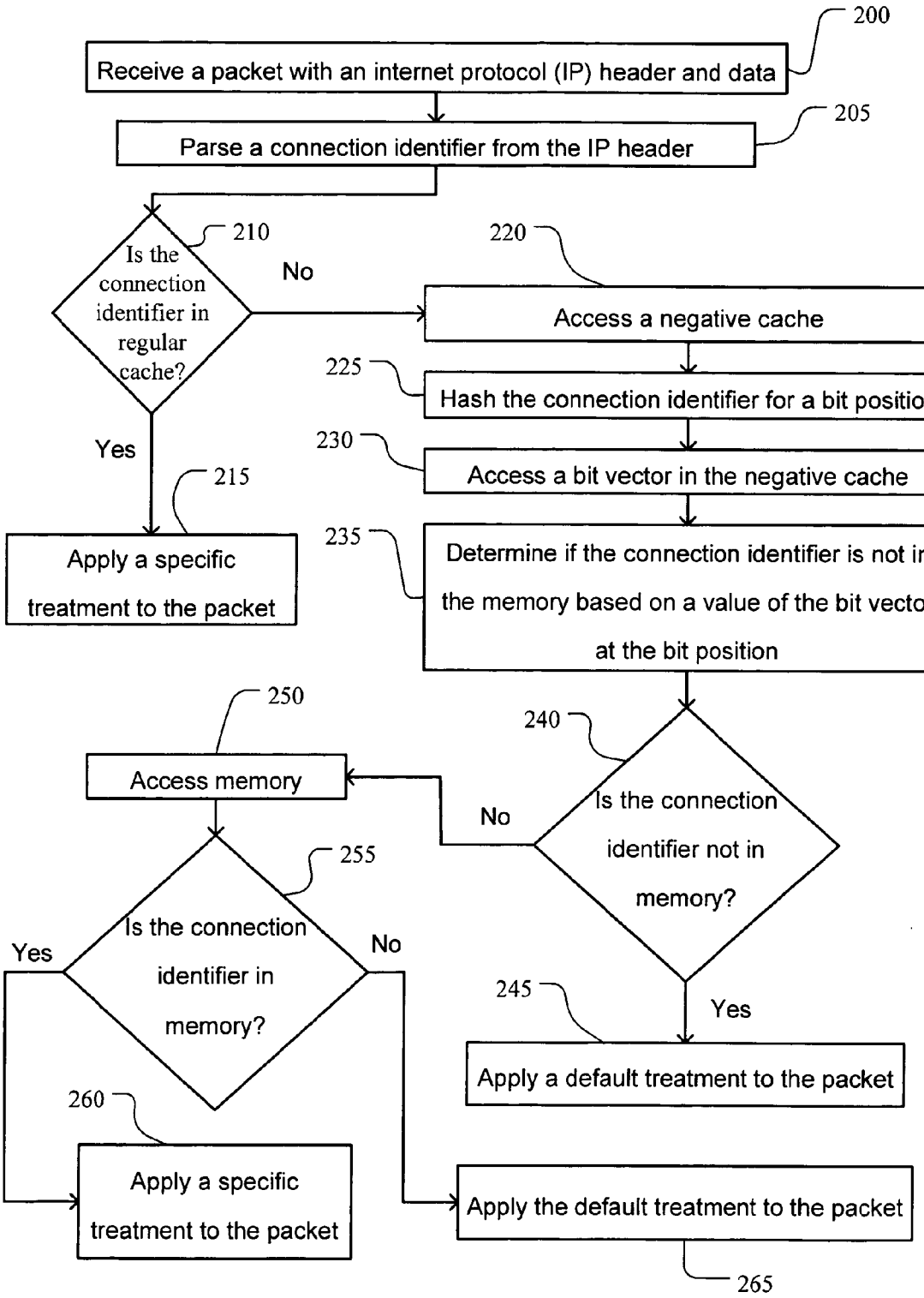

200
Receive a packet with an internet protocol (IP) header and data
205
Parse a connection identifier from the IP header
210
Is the connection identifier in regular cache?
No
220
Access a negative cache
225
Hash the connection identifier for a bit position
230
Access a bit vector in the negative cache
235
Determine if the connection identifier is not in the memory based on a value of the bit vector at the bit position
Yes
215
Apply a specific treatment to the packet
240
Is the connection identifier not in memory?
No
250
Access memory
255
Is the connection identifier in memory?
Yes
No
245
Yes
Apply a default treatment to the packet
260
Apply a specific treatment to the packet
Apply the default treatment to the packet
265

FIG. 2

SYSTEM FOR REDUCING LATENCY IN A HOST ETHERNET ADAPTER (HEA)

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation application under 35 U.S.C. §120 and claims priority to U.S. patent application Ser. No. 11/096,353, filed Apr. 1, 2005, entitled, "Method for Reducing Latency in a Host Ethernet Adapter (HEA)," all of which is incorporated herein by reference.

[0002] The present application is related to the following copending U.S. patent applications:

[0003] U.S. patent application Ser. No. 11/097,608, (Attorney Docket No. RPS920050059US1/3485P), entitled "Host Ethernet Adapter for Networking Offload in Server Environment", filed on even date herewith and assigned to the assignee of the present invention.

[0004] U.S. patent application Ser. No. 11/096,363, (Attorney Docket No. RPS920050060US1/3486P), entitled "Method and System for Accommodating Several Ethernet Ports and a Wrap Transmitted Flow Handled by a Simplified Frame-By-Frame Upper Structure", filed on even date herewith and assigned to the assignee of the present invention.

[0005] U.S. patent application Ser. No. 11/096,571, (Attorney Docket No. RPS920050061US1/3487P), entitled "Method and Apparatus for Providing a Network Connection Table", filed on even date herewith and assigned to the assignee of the present invention.

[0006] U.S. patent application Ser. No. 11/097,051, (Attorney Docket No. RPS920050062US1/3488P), entitled "Network Communications for Operating System Partitions", filed on even date herewith and assigned to the assignee of the present invention.

[0007] U.S. patent application Ser. No. 11/097,652, (Attorney Docket No. RPS920050073US1/3502P), entitled "Configurable Ports for a Host Ethernet Adapter", filed on even date herewith and assigned to the assignee of the present invention.

[0008] U.S. patent application Ser. No. 11/096,365, (Attorney Docket No. RPS920050074US1/3503P), entitled "System and Method for Parsing, Filtering, and Computing the Checksum in a Host Ethernet Adapter (HEA)", filed on even date herewith and assigned to the assignee of the present invention.

[0009] U.S. patent application Ser. No. 11/097,055, (Attorney Docket No. RPS920050076US1/3505P), entitled "Method and Apparatus for Blind Checksum and Correction for Network Transmissions", filed on even date herewith and assigned to the assignee of the present invention.

[0010] U.S. patent application Ser. No. 11/096,362, (Attorney Docket No. RPS920050082US1/3512P), entitled "Method and System for Performing a Packet Header Lookup", filed on even date herewith and assigned to the assignee of the present invention.

[0011] U.S. patent application Ser. No. 11/097,430, (Attorney Docket No. RPS920050089US1/3516P), entitled "System and Method for Computing a Blind Checksum in a Host Ethernet Adapter (HEA)", filed on even date herewith and assigned to the assignee of the present invention.

FIELD OF THE INVENTION

[0012] The present invention relates to adapters for parsing Internet packets generally, and specifically to a system and method for reducing latency in a host Ethernet adapter (HEA).

BACKGROUND OF THE INVENTION

[0013] A computer, or host, connects to a network through an adapter that parses, or separates, each packet received over the network. The adapter may be known as a host Ethernet adapter (HEA). A packet is composed of an Internet protocol (IP) header and some data. The IP header is composed of several fields, such as source address, destination address, ports, protocol, and some transport protocol information. These fields are known as the 5-tuple, or connection identifier (Id), and are used to identify how the packet should be handled.

[0014] After parsing out the connection Id, the HEA sends the connection Id to a memory external to the adapter. The external memory may have a lookup table or connection table for looking up the treatment protocol for the packet. One problem with this is that the process of sending the connection Id to the memory and having the memory look up the treatment protocol for each packet is time-consuming, resulting in an undesirable latency for the look up of the treatment protocol for packets in the adapter. Storing the connection table in the adapter is not practical due to the size of the connection table.

[0015] Accordingly, what is needed is a system and method for reducing latency in a host Ethernet adapter (HEA). The present invention addresses such a need.

BRIEF SUMMARY OF THE INVENTION

[0016] The present invention provides a system for reducing latency in a host Ethernet adapter (HEA) including the following. First, the HEA receives a packet with an Internet protocol (IP) header and data in the HEA. The HEA parses a connection identifier from the IP header and accesses a negative cache in the HEA to determine if the connection identifier is not in a memory external to the HEA. The HEA applies a default treatment to the packet if the connection identifier is not in the memory, thereby reducing latency by decreasing access to the memory.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[0017] FIG. **1** is a block diagram illustrating a host Ethernet adapter (HEA) according to one embodiment of the invention.

[0018] FIG. **2** is a flow diagram illustrating one embodiment of the invention implemented in the diagram of FIG. **1**.

DETAILED DESCRIPTION OF THE INVENTION

[0019] The present invention relates to a system and method for reducing latency in a host Ethernet adapter (HEA). The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiments and the generic principles and features described herein will be readily apparent to those skilled in the art. Thus, the present invention is not intended to be limited to the

embodiments shown, but is to be accorded the widest scope consistent with the principles and features described herein.

[0020] FIG. **1** is a block diagram illustrating a host Ethernet adapter (HEA) **100** connected to a host **102** and a memory **104**. The HEA contains a negative cache **106** connected to a system **108** and optionally may include a regular cache **110**. The memory **104** includes a connection table **112**.

[0021] FIG. **2** is a flow diagram illustrating one embodiment of the invention implemented in the diagram of FIG. **1**. FIGS. **1** and **2** will be discussed in conjunction with one another. For simplicity, reference numerals beginning with '100' will be found in FIG. **1** while reference numerals beginning with '200' will be found in FIG. **2**.

[0022] In block **200** of FIG. **2**, HEA **100** receives a packet **114** with an Internet protocol (IP) header **116** and data **118**. The IP header **116** includes a destination IP address **120**, a source IP address **122**, a protocol **124**, a source port **126**, and a destination port **128**, which are collectively known as a connection identifier (Id) **130**. The HEA **100** may receive many packets **114**.

[0023] In block **205**, the system **108** parses the connection Id **130** from the IP header **114**. If HEA **100** can determine that the connection Id **130** is stored in either regular cache **110** or connection table **112**, then HEA **100** will receive a specific treatment, or set of instructions, for directing packet **114**. Many packets **114** fall under a default treatment setting, which means they are all treated alike. Some packets **114** require special handling, and have specific treatments that differ from the default treatment. Some packets **114** may require special handling, but the specific treatment has not been entered into regular cache **110** or connection table **112**, so a default treatment is used.

[0024] In block **210**, the system **108** may determine if the connection Id **130** is in the optional regular cache **110**. If the regular cache **110** is not present in HEA **100**, this block is skipped. If the connection Id **130** is in the regular cache **110**, then the method of treating packet **114** may be looked up and applied in block **215**. Sometimes the specific treatment listed in the regular cache **110** is the same as the default treatment had the connection Id **130** not been listed.

[0025] Continuing from block **210**, if the connection Id **130** is not in the regular cache **110**, then in block **220**, system **108** accesses the negative cache **106**. The negative cache **106** includes a one-bit wide bit vector **132** lacking collision resolution, with any number of entries. The HEA **100** has better performance with a smaller bit vector **132**.

[0026] In block **225**, negative cache **106** hashes the connection Id **130** with a one-bit wide hash table (not shown) for a bit position. After hashing connection Id **130**, a bit position will result, for example bit position four. Then, in block **230**, negative cache **106** accesses the bit vector **132** and in block **235** determines if the connection Id **130** is not in the memory **104** based on the value of the bit vector **132** at the bit position (for example, bit position four).

[0027] At this time, an explanation of setting the bit vector **132** will aid in understanding the invention. The bit vector **132** is initially set to a predetermined value, for example all zero to indicate that no entries are in the memory **104**. When a specific treatment for a connection Id **130** is entered into memory **104**, the hash table of negative cache **106** (not shown) is used to hash the connection Id **130** being stored, resulting in a bit position. That bit position represents the connection Id **130**, and flipping the value from the default setting indicates the possible presence of a specific treatment for the connection Id **130** in the memory **104**. Negative cache **106** can only negatively determine the presence of a connection Id **130** in the memory **104** because the hash table (not shown) and bit vector **132** lack collision resolution, meaning some connection Ids **130** will have the same hash bit position.

[0028] For example when storing in memory **104** the specific treatment for a connection Id **130**, assume a bit vector **132** with all zeros. A hash of the connection Id **130** results in a bit position, for example bit position four, which currently has a value of zero (all the positions are currently zero). Bit position four is then switched in bit vector **132**, for example from zero to one. However, other connection Ids **130**, which do not receive the same treatment as the first, may also hash to position four. By changing bit position four from zero to one, the negative cache **106** establishes that at least one of possibly several connection Ids **130** with a hash value to bit position four are in the memory **104**. Only by checking the memory **104** will HEA **100** be able to determine if a particular connection Id **130** is in the memory **104**. However, if the value is zero, none of the connection Ids with a hash value to that bit position are in the memory **104**, and the memory **104** does not need to be checked. This is why cache **106** is a 'negative' cache.

[0029] Continuing after block **235**, in block **240** HEA **100** determines if the connection Id **130** is not in the memory **104**. As explained above, only the absence of the connection Id **130** from the memory **130** can be answered with authority, due to the lack of collision resolution in the bit vector **132**.

[0030] If the connection Id **130** is not in the memory **104**, then in block **245** the HEA **100** applies a default treatment to the packet **114**. The amount of time required to check the bit vector **132** is less than the amount of time required to check the connection table **112** in the memory **104**, so avoiding access to the memory **104** decreases latency in the HEA **100**.

[0031] In the earlier example, connection Id **130** hashed to bit position four. In bit vector **132**, bit position four **134** is one, so the absence of the connection Id **130** from memory **104** cannot be established without actually checking the memory **104**. Therefore, in block **250**, HEA **100** accesses the memory **104**. The memory **104** may access connection table **112** to determine if the connection Id **130** is present. If the connection Id **130** is present, the treatment is sent to HEA **100** and in block **260** the HEA **100** applies a specific treatment to the packet **114**.

[0032] If the connection Id **130** is not present in the connection table **112**, then in block **265** the HEA **100** applies the default treatment to the packet **114**. By properly identify which packets **114** should be given a default treatment and which have a specific treatment, the efficiency of HEA **100** increases.

[0033] After parsing, filtering and performing other tasks on the packet **114**, HEA **100** sends the data **118** on to host **102**.

[0034] According to the method and system disclosed herein, the present invention discloses a system and method for reducing latency in a HEA. One skilled in the art will recognize that the particular standards used are exemplary, and any bandwidth-limited network may apply the invention in the above manner. The present invention has been described in accordance with the embodiments shown, and one of ordinary skill in the art will readily recognize that there could be variations to the embodiments, and any variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of

ordinary skill in the art without departing from the spirit and scope of the appended claims.

We claim:

1. A host Ethernet adapter (HEA) for reducing latency comprising:

a system for receiving a packet with an internet protocol (IP) header and data, and for parsing a connection identifier from the IP header; and

a negative cache coupled to the system, for determining if the connection identifier is not in a memory external to the HEA, the system applying a default treatment to the packet if the connection identifier is not in the memory, thereby reducing latency by decreasing access to the memory,

the negative cache further for hashing the connection identifier for a bit position, accessing a bit vector in the HEA, and for determining if the connection identifier is not in the memory based on a value of the bit vector at the bit position.

2. The HEA of claim **1** further comprising:

a regular cache coupled to the system, the system further for determining if the connection identifier is in the regular cache, and for applying a specific treatment to the packet if the connection identifier is in the regular cache.

3. The HEA of claim **1**, the system for applying a specific treatment to the packet if the connection identifier is in the memory.

4. The HEA of claim **3**, the system for applying the default treatment to the packet if the connection identifier is not in the memory.

5. The HEA of claim **1**, a value of zero in bit vector at the bit position indicating that the connection identifier is not in the memory.

6. The HEA of claim **1**, the negative cache is a one-bit wide hash table lacking collision resolution.

7. The HEA of claim **1**, the connection identifier is a protocol, a source Internet protocol (IP) address, a destination IP address, a source port, and a destination port.

8. A computer readable medium encoded with computer executable instructions for reducing latency in a host Ethernet adapter (HEA), the computer executable instructions comprising:

receiving a packet with an internet protocol (IP) header and data;

parsing a connection identifier from the IP header;

accessing a negative cache in the HEA to determine if the connection identifier is not in a memory external to the HEA, the accessing a negative cache in the HEA further comprising:

hashing the connection identifier for a bit position;

accessing a bit vector in the HEA; and

determining if the connection identifier is not in the memory based on a value of the bit vector at the bit position; and

applying a default treatment to the packet if the connection identifier is not in the memory, thereby reducing latency by decreasing access to the memory.

9. The computer readable medium of claim **8**, the programming instructions further comprising:

determining if the connection identifier is in a regular cache in the HEA; and

applying a specific treatment to the packet if the connection identifier is in the regular cache.

10. The computer readable medium of claim **8**, the programming instructions further comprising:

determining if the connection identifier is in the memory;

applying a specific treatment to the packet if the connection identifier is in the memory; and

applying the default treatment to the packet if the connection identifier is not in the memory.

11. The computer readable medium of claim **8**, a value of zero in bit vector at the bit position indicating that the connection identifier is not in the memory.

12. The computer readable medium of claim **8** wherein the negative cache is a one-bit wide hash table lacking collision resolution.

13. The computer readable medium of claim **8** wherein the connection identifier is a protocol, a source Internet protocol (IP) address, a destination IP address, a source port, and a destination port.

* * * * *