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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 February 2010 (04.02.2010)

PCT

(10) International Publication Number
WO 2010/013098 A1

(51) International Patent Classification:
G06F 11/07 (2006.01) *H04L 12/26* (2006.01)
G06F 11/36 (2006.01)

(21) International Application Number:
PCT/IB2008/054460

(22) International Filing Date:
1 August 2008 (01.08.2008)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **ALCATEL LUCENT** [FR/FR]; 54, rue La Boétie, F-75008 Paris (FR).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **VERMA, Anish** [IN/IN]; #1683, East End A Main Road, 9th blockjayanagar, Bangalore, Karnataka 560003 (IN). **MCRAE, Andrew** [AU/AU]; 21, Glencoe Close, Berowra, Berowra, Nsw, 2081 (AU).

(74) Agent: **KORAKIS-MENAGER, Sophie**; Compagnie Financiere Alcatel-lucent, 54, rue La Boétie, F-75008 Paris (FR).

(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, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

— *with international search report (Art. 21(3))*

WO 2010/013098 A1

(54) Title: DATA PATH DEBUGGING

(57) Abstract: A method for debugging data paths in a network device is disclosed. The method includes inserting a debugging node in the data path, classifying the date packets passing "through the debugging nodes, generating debug logs for the data packet with rate limiting and removing the debugging node. The debugging nodes include complete debugging. rate limiting and one pass classification functionalities. The debugging node debugs the data packets and removes debug logs from uninterested interfaces. The debugging nodes may be placed at the ingress or egress of a particular component to debug the functionality of the component.

Data path debugging

BACKGROUND

Technical Field

[001] The embodiments disclosed herein generally relate to networks, and, more particularly, to data path debugging in networks.

Description of the Related Art

[002] Existing techniques to debug data path in network devices (for example, routers and switches) deteriorate the performance of the network devices and thus making online troubleshooting extremely difficult. In existing solutions, data path has to carry the debug code and the debug code has to be executed for every data packet that is processed by a network device. So the limitations of the existing debug architecture burden the data path code and hence adversely impact the network performance. Even if data logs are turned off, performance of network devices are adversely affected as the debugging code is present on data path.

[003] Furthermore, in existing solutions, rate limiting of debug logs occurs at debug logging server. Data path continuously generates debug logs, and debug logging server has to rate-limit the logs generated by data path. This will further impact performance of network devices adversely.

[004] Also, in existing solutions, debug architecture is not flexible enough in allowing debug code for particular component or a specified set of components in a network device. By default, debugging logs are generated for all interfaces in a network

device, resulting in a large number of debugging logs and therefore making troubleshooting more difficult.

SUMMARY

[005] In view of the foregoing, an embodiment herein provides a method and system for debugging a data path of a network device, for example, router/switch using dynamic method to selectively add and remove debugging nodes in the data path with no impact on the packet forwarding performance, the method comprising steps of inserting a debugging node in the data path, classifying the data packets passing through debugging node, generating debug logs for the data packets, and removing the debugging node from the data path on completion of debugging. A plurality of debugging nodes can be placed on the data path on per interface basis. The debugging node can be placed on the ingress and egress of a particular component in the data path thereby permitting the user to debug the functionality of a particular component in the data path. The debugging node includes complete debugging and rate limiting functionality. The debugging node classifies the data packets passing through the node using one pass classification and generates debug logs for each data packet with rate limiting. The debugging node classifies said data packets using one pass classification up to seven layers of the Open System Interconnection (OSI) model.

[006] Embodiments herein further disclose a debugging node adapted to perform debugging of data path comprises at least one means adapted to classify the data packets on data path, generate debug logs for the data packets and perform rate limiting of the debug logs for matching the data packets. The debugging node is adapted to classify said data packets using one pass classification and provides classification of data packets up to

seven layers of the OSI model.

[007] Embodiments herein further disclose a system for debugging a data path of a network device using dynamic method to selectively add and remove debugging nodes in a data path with no impact on the packet forwarding performance, the system comprising at least one means adapted to insert a debugging node in the data path, classify the data packets passing through the debugging node, generate debug logs for the data packets with rate limiting and remove the debugging node once the user is done with debugging the data path. A plurality of debugging nodes can be placed on the data path where the debugging node can be placed on the ingress and egress of a particular component in the data path. The debugging node includes complete debugging and rate limiting functionality and is adapted to classify said data packets using one pass classification and provides classification of data packets up to seven layers of the OSI model.

[008]These and other aspects of the embodiments disclosed herein will be better appreciated and understood when considered in conjunction with the following description and the accompanying drawings. It should be understood, however, that the following descriptions, while indicating preferred embodiments and numerous specific details thereof, are given by way of illustration and not of limitation. Many changes and modifications may be made within the scope of the embodiments disclosed herein without departing from the spirit thereof, and the embodiments disclosed herein include all such modifications.

BRIEF DESCRIPTION OF THE DRAWINGS

[009] The embodiments disclosed herein will be better understood from the following detailed description with reference to the drawings, in which:

[0010] FIG. 1 illustrates a schematic diagram depicting a network device and components, according to an embodiment herein;

[0011] FIG. 2 illustrates a block diagram depicting the functional modules of a debugging node, according to an embodiment herein;.

[0012]FIG. 3 is a block diagram showing an exemplary illustration of a data path with debugging nodes inserted on an interface, according to an embodiment herein;

[0013] FIG. 4 illustrates a schematic diagram depicting the debugging nodes inserted on the ingress and egress of a Network Address Translation (NAT) node in the data path, according to an embodiment herein; and

[0014] FIG. 5 illustrates a flowchart depicting a method of debugging a data path by inserting debugging nodes, according to an embodiment herein.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0015] The embodiments disclosed herein and the various features and advantageous details thereof are explained more fully with reference to the non-limiting embodiments that are illustrated in the accompanying drawings and detailed in the following description. Descriptions of well-known components and processing techniques are omitted so as to not unnecessarily obscure the embodiments disclosed herein. The examples used herein are intended merely to facilitate an understanding of ways in which the embodiments disclosed herein may be practiced and to further enable those of skill in the art to practice the embodiments disclosed herein. Accordingly, the

examples should not be construed as limiting the scope of the embodiments disclosed herein.

[0016] Embodiments herein disclose a method of debugging the data path functionality of a network device by providing a dynamic method to selectively add and remove debugging nodes in a data path with little or no impact on the packet forwarding performance. Referring now to the drawings, and more particularly to FIGS. 1 through 4, where similar reference characters denote corresponding features consistently throughout the figures, there are shown preferred embodiments.

[0017] A debugging node contains complete debugging functionality, including rate limiting functionality. For debugging a path in a network device, user can insert a debugging node at the required point in the data path. The debugging node can be programmatically created and inserted in the data path. A user can place any number of debugging nodes simultaneously in the data path. More than one instance of a debugging node can be placed on the ingress and egress of a particular component in the data path, thereby permitting the user to debug the functionality of a particular component in the data path and narrow down a data path level problem to a component level problem in the data path. The debugging node classifies the data packets passing through the node and generates debug logs for each data packet with rate limiting. The debugging nodes have little or no impact on the packet forwarding performance of the path. Once the user is done with debugging the path, then the user can then remove the debugging node.

[0018] FIG. 1 illustrates a schematic diagram depicting a network device and the components, according to an embodiment herein. A network device 101 works as an intermediate system that mediates sending, receiving or forwarding data in a computer

network. The network device 101 can be a router, hub, bridge or switch. The network devices 101 allow computers on completely separate networks to communicate with one another. A network device 101 say router is used as gateway for other computers to access the Internet 102. The network device 101 is placed between the computers and the modem provided by the internet service provider and connects all computers to the network device by connecting the network device to the modem. The network device is connected to the computer 1 103, computer 2 104, server 105 and by a wireless connection to a laptop 106.

[0019] FIG. 2 illustrates a block diagram depicting the functional modules of a debugging node, according to an embodiment herein. The functional modules of the debugging node 201 include complete debugging module 202, rate-limiting module 203, and classifier module 204. The debugging node 201 can be any device connected to a network such as computers, personal digital assistants (PDAs), cell phones, switches, routers or other networked devices. The debugging node 201 acts as a connection point, either a redistribution point or an end point, for data transmission and has programmed or engineered capability such as debugging commands or modules to recognize and process data transmission to other nodes. Rate limiting is a security feature which disables a user's ability to send several instant messages at a time. The insertion and removal of debugging node 201 dynamically allows rate limiting of debug logs at data paths. The debugging node 201 uses a common classifier to classify the data packets on the data path. The classifier may be a collection of rules or policies. Packet classification requires matching each data packet against a database of filters (or rules), and forwarding the packet according to the highest priority filter. The classifier may use one pass

classification to classify the packets, where one pass classification is one method of packet classification where a single, flexible, extensible syntax defines a common classification and specifies policies for all services. The syntax also defines complex classifications for QoS, anti-virus, VoIP and other applications. With single-pass packet classification, a packet enters a firewall first, thus protecting all other services in a gateway. In the firewall, the IPSec service decrypts and classifies the packet using the common classification and attaches a tag that contains information about which services need to process the packet. The packet then passes to a filter in the services gateway that accepts or denies the packet based on information in the tag. In one pass classification 204, each data packet is classified only once in the data path and the rest of the nodes in the data path utilize the same classification. In one pass classification 204, data packets usually pass through a variety of security modules, such as firewalls and content filters before the packet is forwarded. One pass classification module 204 also provides classification upto seven layers of Open Systems Interconnection Basic Reference Model (OSI), where the seven layers are the Physical layer, Data link layer, Network layer, Transport layer, Session layer, Presentation layer, and the Application layer. In One pass classification the data packets are classified only once and rest of the module make use of this classification. Also, no extra load needs to be added to classify data packets in one pass classification. The debugging nodes 201 are dependent on the interface. The ingress and egress interfaces of the debugging nodes 201 can be modified according to the location of the node.

[0020] FIG. 3 is a block diagram showing an exemplary illustration of a data path with debugging nodes inserted on a data path, according to an embodiment herein. The

data path comprises of 3 nodes, 301, 302 and 303. Debugging nodes 201 have been inserted in the data path to debug the data path. The debugging node 201 classifies data packets passing through the node 201 and generates debug logs for packets, which match the criteria and available limits. The debug logs are generated only for packets which match pre-determined criteria. Also, the number of debug logs generated may also be limited, on a per second or a per minute basis. The debug logs generated are rate limited for matching each data packet. The debugging node 201 uses a common classifier to classify the data packets on the data path. The classifier module 204 may use one pass classification, where each data packet is classified only once in the data path and the rest of the nodes in the data path utilize the same classification. The classifier module 204 also provides classification upto seven layers. The user can also verify the functionality of node 2 302, using the debug logs which are generated by the debugging nodes 201 for a data packet before and after node 2 302.

[0021] FIG. 4 illustrates a schematic diagram depicting the debugging nodes inserted on the ingress and egress of a Network Address Translation (NAT) node 402 in the data path, according to an embodiment herein. The data path comprises of 3 nodes, decap node 401, NAT node 402 and IP node 403. Debugging nodes 201 have been inserted in the data path to debug the data path. The debugging node 201 classifies the data packets passing through the nodes 201 and generates debug logs for matching data packet if logs are within the configured rate limit. The debugging node 201 uses a common classifier to classify the data packets on the data path. The classifier module 204 may use one pass classification, where in one pass classification, data packet is classified only once in the data path and the rest of the nodes in the data path utilize the same

classification. One pass classification also provides classification upto seven layers. The user can verify the functionality of NAT node 402, using the debug logs which are generated by the debugging nodes 201 for a data packet before and after NAT node 402.

[0022] FIG. 5 illustrates a flowchart depicting a method of debugging a data path by inserting debugging nodes, according to an embodiment herein. The user inserts (501) a debugging node 201 on a particular interface in the data path. The interface can have more than one instance for inserting a debugging node 201. The debugging node 201 classifies (502) the data packets passing through debugging node. The node 201 uses a common classifier to classify the data packets on the data path. Further, the debugging node 201 generates (503) debug logs for the packet with rate limiting. Thereafter, the user removes (504) the debugging nodes 201 from the data path. The debug node 201 is interface based and generates logs for data packets only from interested interface and removes debug logs from uninterested interface. The dynamic insertion and removal of debugging nodes 201 permits rate limiting of debug logs at the data path. The selective insertion and removal of debug nodes 201 has no impact on the packet forwarding performance of the data path while debugging is turned OFF and the selective insertion and removal of debug nodes 201 has minimal impact on the packet forwarding performance of the data path while debugging is turned ON. However, the impact does not diminish the performance of the data path as compared to the existing solutions. The various actions in method 500 may be performed in the order presented, in a different order, or simultaneously. Further, in some embodiments, some actions listed in FIG. 5 may be omitted.

[0023] The embodiments disclosed herein facilitate the debugging of the

functionality of a component by placing the debugging node ingress or egress of the component, thereby reducing the problems of debugging to component level in data path. Further, the debugging node is inserted on per interface basis, controls data path debugging and also removes the debug logs from uninterested interface.

[0024] As can be appreciated, the embodiments disclosed herein provides data path debugging functionality of a network device by inserting a debugging node containing complete debugging and rate limiting functionality. Also it is to be understood that the invention as described here is not limited to this precise embodiment and that various changes and modifications may be affected therein without departing from the original scope or spirit of present invention.

[0025] The list of structures corresponding to the claimed means is not exhaustive and that one skilled in the art understands that equivalent structures can be substituted for the recited structure without departing from the scope of the invention.

[0026] The embodiments disclosed herein can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment including both hardware and software elements. The embodiments that are implemented in software include but are not limited to, firmware, resident software, microcode, etc.

[0027] The foregoing description of the specific embodiments will so fully reveal the general nature of the embodiments disclosed herein that others can, by applying current knowledge, readily modify and/or adapt for various applications such specific embodiments without departing from the generic concept, and, therefore, such adaptations and modifications should and are intended to be comprehended within the meaning and range of equivalents of the disclosed embodiments. It is to be understood

that the phraseology or terminology employed herein is for the purpose of description and not of limitation. Therefore, while the embodiments disclosed herein have been described in terms of preferred embodiments, those skilled in the art will recognize that the embodiments disclosed herein can be practiced with modification within the spirit and scope of the appended claims.

CLAIMS

What is claimed is:

1. A method of debugging a data path of a network device, the method comprising steps of

inserting (501) at least one debugging node (201) in said data path,

classifying (502) data packets passing through said debugging node (201),

generating (503) debug logs for said data packets; and

removing (504) said debugging node (201).

2. The method, as claimed in claim 1, wherein said debugging node (201) includes debugging functionality.

3. The method, as claimed in claim 1, wherein said debugging node (201) includes rate limiting functionality for debug logs.

4. The method as claimed in claim 1, wherein said method permits a user to debug functionality of a component in said data path, wherein said debugging nodes (201) are placed on ingress and egress of said component.

5. The method, as claimed in claim 1, wherein said debugging node (201) classifies said data packets using one pass classification.

6. The method of claim 1, wherein said debugging node (201) classifies said data

packets using one pass classification up to seven layers of the Open System Interconnection (OSI) model.

7. A node (201) adapted to perform debugging of data path, wherein said node (201) comprises atleast one means adapted to

classifying (502) data packets on said data path;

performing rate limiting of debug logs for matching said data packets; and

generating (503) debug logs for said data packets.

8. A debugging node (201), as claimed in claim 7, wherein said debugging node (201) is adapted to classify (502) said data packets using one pass classification.

9. A system for debugging a data path in a network device, said system comprising atleast one means adapted to:

inserting (501) at least one debugging node (201) in said data path,

classifying (502) data packets passing through said debugging node (201),

generating (503) debug logs for said data packets; and

removing (504) said debugging node (201).

10. The system, as claimed in claim 9, wherein said debugging node (201) is adapted to perform debugging of data path, wherein said node (201) comprises atleast one means adapted to

classifying (502) data packets on said data path;

performing rate limiting of debug logs for matching said data packets; and generating (503) debug logs for said data packets.

11. The system, as claimed in claim 9, wherein said debugging node (201) is adapted to classify (202) said data packets using one pass classification up to seven layers of Open System Interconnection (OSI) model.

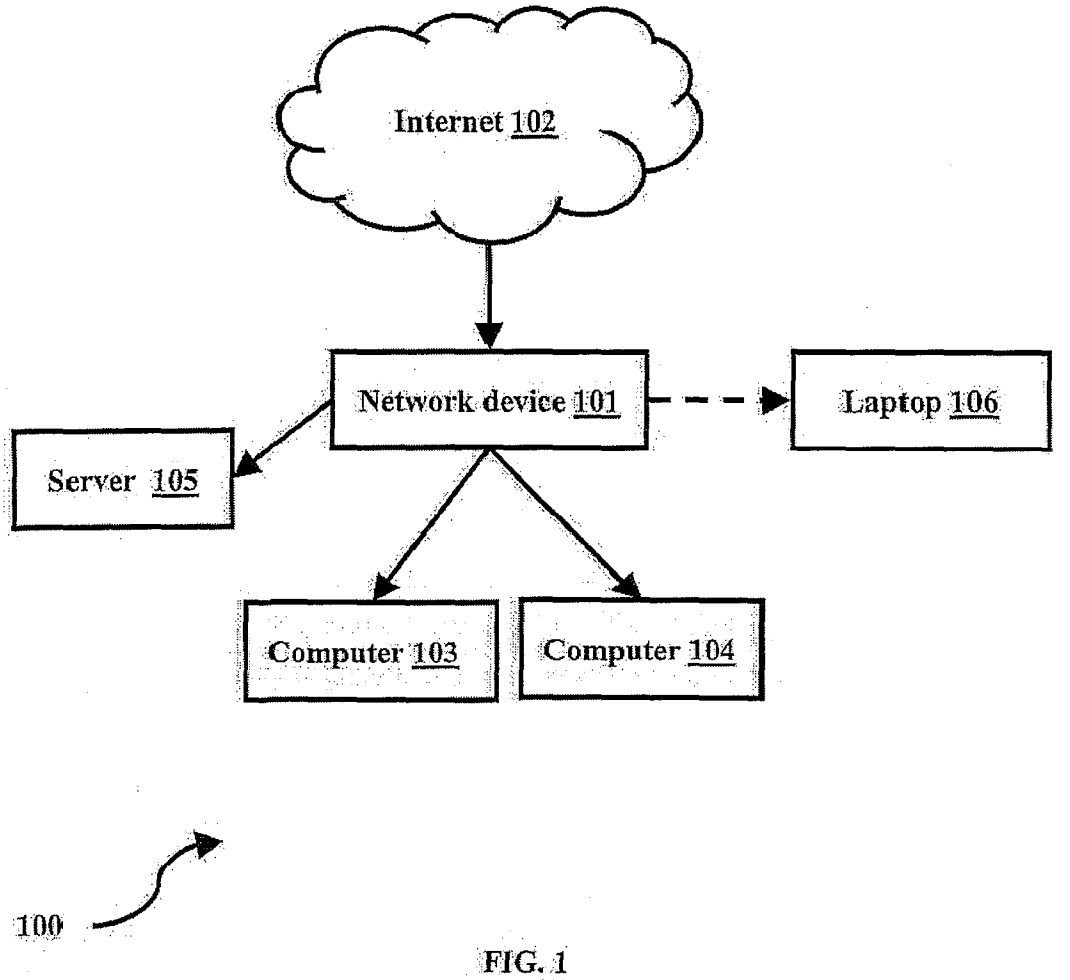
Internet 102
Network device 101
Laptop 106
Server 105
Computer 103
Computer 104
100

FIG. 1

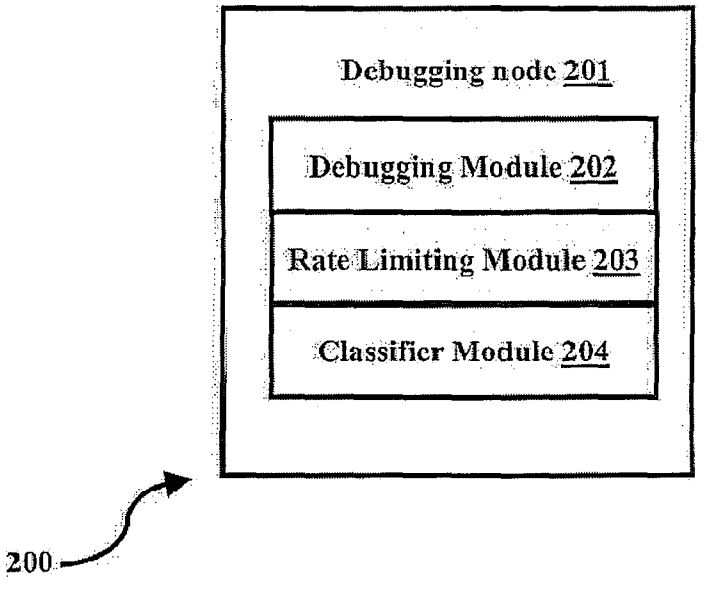
Debugging node 201
Debugging Module 202
Rate Limiting Module 203
Classifier Module 204
200

FIG. 2

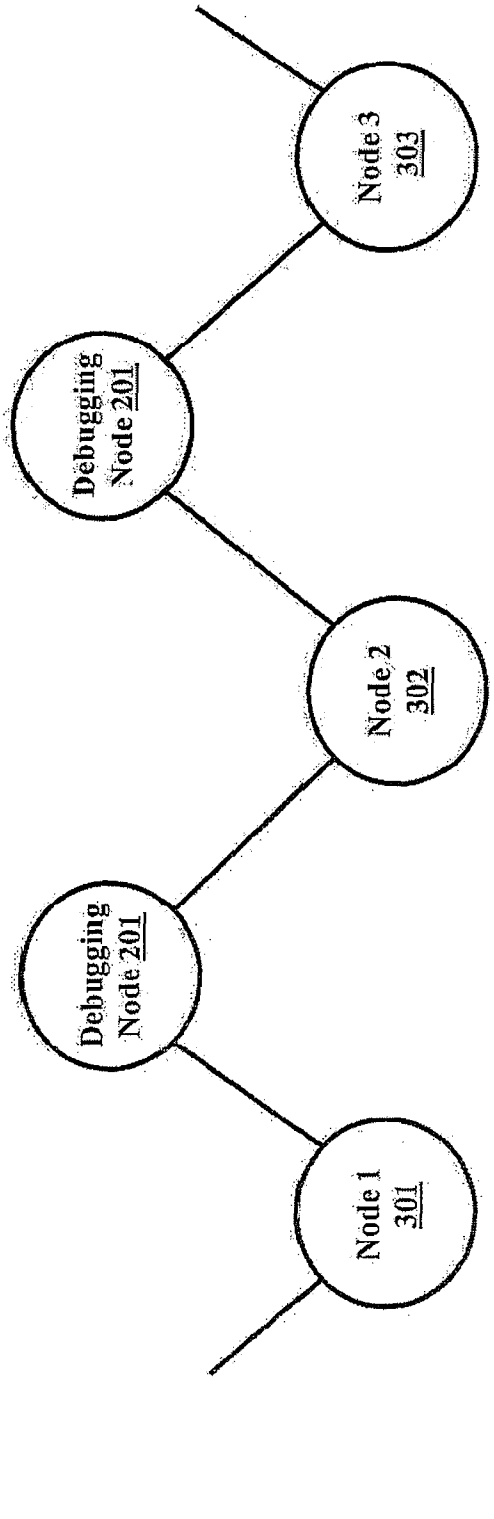
Node 1
301
Debugging
Node 201
Node 2
302
Debugging
Node 201
Node 3
303
300

FIG. 3

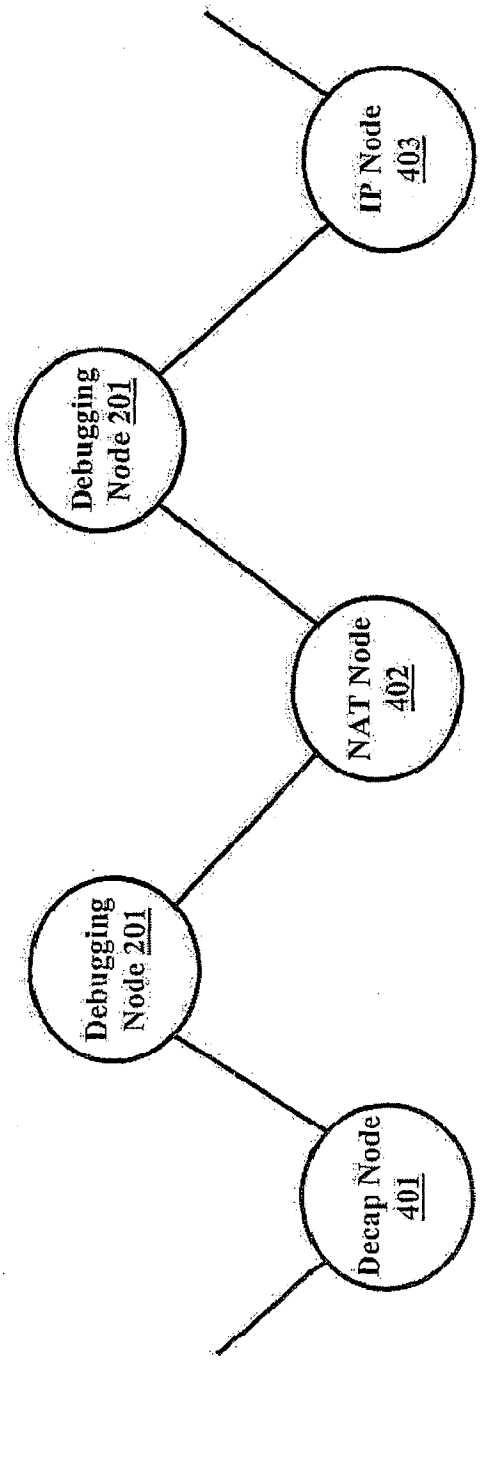
Decap Node 401
Debugging Node 201
NAT Node 402
Debugging Node 201
IP Node 403
400

FIG. 4

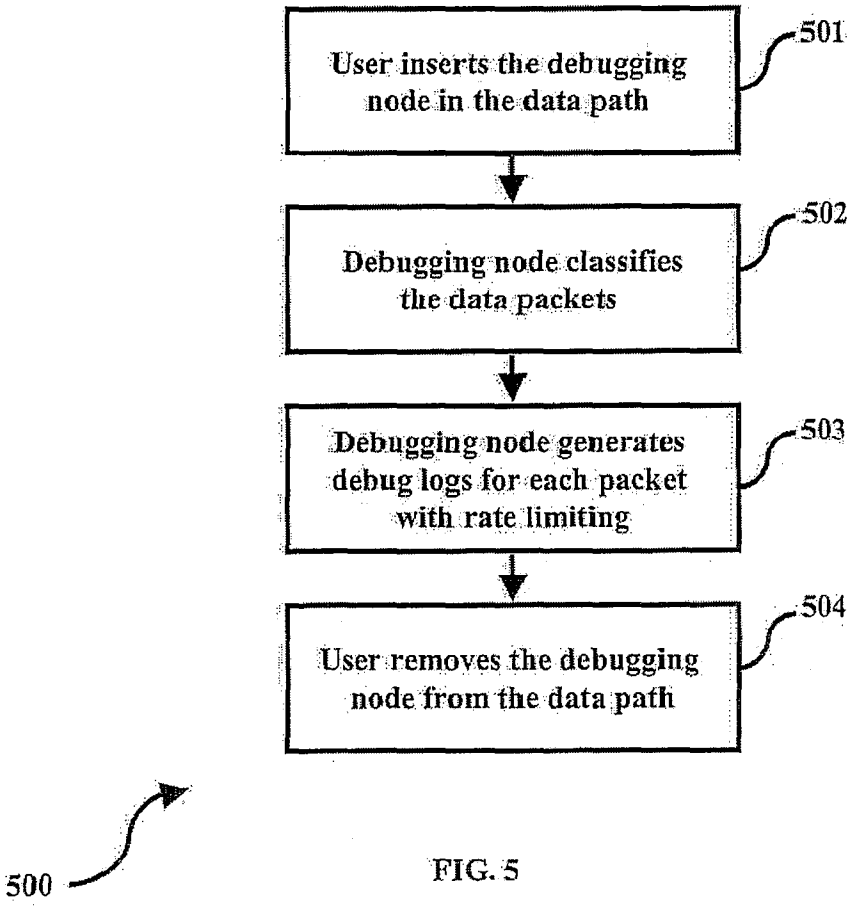
User inserts the debugging node in the data path
501
Debugging node classifies the data packets
502
Debugging node generates debug logs for each packet with rate limiting
503
User removes the debugging node from the data path
504
500

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2008/054460

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F11/07 G06F11/36 H04L12/26

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)
H04L G06F

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	C.K.ZÜWER, J.W.LOCKWOOD: "Debugging of an Internet Packet Scheduler Using the Identify® Software" THE SYNDICATED - A TECHNICAL NEWSLETTER FOR ASIC AND FPGA DESIGNERS, vol. 4, no. 4, December 2004 (2004-12), pages 5-6, XP002518744	1,2,4-11
Y	the whole document	3
Y	KEVIN DOOLEY ET AL: "Paragraph 18.14: Rate-Limiting Syslog Traffic" CISCO COOKBOOK, DOOLEY K, BROWN I J, O'REALLY, July 2003 (2003-07), pages 689-690, XP009113703 ISBN: 978-0-596-00367-8 the whole document	3

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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

"E" earlier document 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
16 March 2009	26/03/2009

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Di Felice, M

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2008/054460

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7 299 277 B1 (MORAN MIKE [US] ET AL) 20 November 2007 (2007-11-20) abstract column 1, lines 16-19 column 2, lines 9-54 column 4, lines 1-19 column 5, lines 39-55 column 9, lines 36-48 column 15, lines 29-45 column 16, lines 1-62 column 19, lines 36-52 column 32, line 64 - column 35, line 67 figures 3,1115-17	1,2,5-11
X	US 2005/060598 A1 (KLOTZ STEVE [US] ET AL) 17 March 2005 (2005-03-17) abstract paragraph [0002] paragraph [0012] - paragraph [0014] paragraph [0038] - paragraph [0045] figures 3,4,7,8	1,2,7,9
A	US 6 182 247 B1 (HERRMANN ALAN L [US] ET AL) 30 January 2001 (2001-01-30) abstract column 3, line 61 - column 4, line 67 column 5, lines 36-42 column 9, lines 44-54 column 11, lines 41-60 figures 1-6	1-11
A	US 5 611 044 A (LUNDEBY BRUCE A [US]) 11 March 1997 (1997-03-11) abstract column 1, lines 12-16 column 3, lines 11-55 column 4, lines 13-36 column 5, line 1 - column 6, line 62 figures 2-4	1-11
A	US 6 651 099 B1 (DIETZ RUSSELL S [US] ET AL) 18 November 2003 (2003-11-18) abstract column 1, lines 39-42 column 4, line 45 - column 7, line 24	1-11
A	EP 0 230 712 A (OUTRAM RES LTD [GB]) 5 August 1987 (1987-08-05) abstract column 1, lines 6-13 column 3, line 13 - column 4, line 10	3

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/IB2008/054460

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7299277 B1	20-11-2007	NONE	
US 2005060598 A1	17-03-2005	NONE	
US 6182247 B1	30-01-2001	US 6389558 B1	14-05-2002
US 5611044 A	11-03-1997	NONE	
US 6651099 B1	18-11-2003	US 2004083299 A1	29-04-2004
EP 0230712 A	05-08-1987	CA 1273119 A1	21-08-1990
		DE 3676477 D1	07-02-1991
		JP 62168285 A	24-07-1987
		US 4910692 A	20-03-1990